Available online at www.sciencedirect.com
SCIENCE DIRECT®
Information Sciences 172 (2005) 361–395
INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL
www.elsevier.com/locate/ins

From intra-transaction to generalized inter-transaction: Landscaping multidimensional contexts in association rule mining

Qing Li [a,*], Ling Feng [b], Allan Wong [c]

[a] *Department of Computer Engineering and Information Technology, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hongkong*
[b] *Department of Computer Science, University of Twente, P.O. Box 217, 7500 AE, Enschede, The Netherlands*
[c] *Department of Computing, Hong Kong Polytechnic University, China*

Received 11 November 2003; received in revised form 6 July 2004; accepted 6 July 2004

Abstract

The problem of mining multidimensional inter-transactional association rules was recently introduced in [ACM Trans. Inform. Syst. 18(4) (2000) 423; Proc. of the ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, Seattle, Washington, June 1998, p. 12:1]. It extends the scope of mining association rules from traditional single-dimensional intra-transactional associations to multidimensional inter-transactional associations. Inter-transactional association rules can represent not only the associations of items happening within transactions as traditional intra-transactional association rules do, but also the associations of items among different transactions under a multidimensional context. "After McDonald and Burger King open branches, KFC will open a branch two months later and one mile away" is

* Corresponding author.
E-mail addresses: itqli@cityu.edu.hk (Q. Li), ling@cs.utwente.nl (L. Feng), csalwong@comp.polyu.edu.hk (A. Wong).

0020-0255/$ - see front matter
doi:10.1016/j.ins.2004.07.006

an example of such rules. In this paper, we extend the previous problem definition based on context expansion, and present a more general form of association rules, named *generalized multidimensional inter-transactional association rules*. An algorithm for mining such generalized inter-transactional association rules is presented by extension of a priori. We report our experiments on applying the algorithm to both real-life and synthetic data sets. Empirical evaluation shows that with the generalized inter-transactional association rules, more comprehensive and interesting association relationships can be detected from data sets.

Keywords: Intra-transactional association rule; Generalized inter-transactional association rule; Point/scope-wise; Multidimensional context

1. Introduction

The problem of mining association rules from basket data was first introduced by Agrawal et al. in [2]. A formal statement of the problem is as follows: Let $\mathscr{I} = \{i_1, i_2, \ldots, i_m\}$ be a set of literals, called items. Let $\mathscr{D}$ be a set of transactions in the database, where each transaction T is a set of items such that $T \subseteq \mathscr{I}$. A transaction is said to contain X, a subset of items in $\mathscr{I}$, if $X \subseteq T$. An *association rule* is an implication of the form $X \Rightarrow Y$, where $X \subset \mathscr{I}$, $Y \subset \mathscr{I}$, and $X \cap Y = \emptyset$. The rule $X \Rightarrow Y$ holds in the transaction set $\mathscr{D}$ with *confidence c* if c% of transactions in $\mathscr{D}$ that contain X also contain Y. The rule $X \Rightarrow Y$ has *support s* in the transaction set $\mathscr{D}$ if s% of transactions in $\mathscr{D}$ contain $X \cup Y$ [2,5]. Association rules provide a useful mechanism for discovering correlations among the underlying data. Applications of association rules range from decision support to product marketing, alarm diagnosis and prediction. For example, we can apply the above association rule concept to discover such knowledge as:

R_1: "When the prices of IBM and SUN go up, 80% of the time the price of Microsoft increases on the same day."

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

Mining association rules from large databases has received considerable attention in recent years. Numerous studies have been carried out in various directions, including efficient, a priori-like mining methods [5,19,30,44–46, 50,56,65], mining without generating candidates [28,47], mining generalized, multilevel, or quantitative association rules [23,24,26,29,31,41,49,53,54], association rule mining query languages [40,57], constraint-based rule mining [7,27,42,55,57], incremental maintenance of discovered association rules [12], parallel and distributed mining [4,13,25], mining correlations and causal structures [10,51,52], cyclic, interesting and surprising association rule mining [1,11,14,43,48], mining association rules with multiple supports [34,60], and so on. Despite these efforts, there is an important form of association relationships which are useful, but could not be expressed with the traditional association rule framework.

1.1. Extension of classical association rules to inter-transactional association rules

Recently, the notion of *multidimensional inter-transactional association rules* was introduced in [36,37]. It is motivated by the observation that many real-world associations happen under certain *contexts*. However, in the traditional association mining, this contextual information has received less exploration due to the fact that such rule mining is *intra-transaction* in nature, i.e., only looking at associations happening within the same transaction, which could be the items bought by the *same customer*, the events happening at the *same time*, etc. On the other hand, 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 such as:

R_1': "When the prices of IBM and SUN go up, 80% of the time the price of Microsoft increases *the next day*."

R_2': "If there is an east wind direction and no rain *in 6 hours*, then there will also be no rain *in 24 hours*."

The inter-transactional association concept can be further extended to associate multiple contextual properties in the same rule, so that multidimensional inter-transactional association rules can be defined and discovered. A 2-dimensional inter-transactional association rule example is

R_3': "After McDonald and Burger King open branches, KFC will open a branch *two months later* and *one mile away*."

which involves a 2-dimensional context comprised by *time* and *space*.

Inter-transactional association rules provide a more detailed view of association relationships among items because they intend to capture richer contextual information for association relationships. In comparison, the context for traditional intra-transactional association rules is limited to single transaction. Thus, from both a conceptual and algorithmic point of view, traditional intra-transactional association rules can be viewed as a simple case of inter-transactional association rules. For mining inter-transactional association rules from large data sets, two kinds of algorithms, namely E/EH-a priori (Extended/

Extended Hash-based a priori) and FITI (First-Intra-Then-Inter), were described in [36,37,58], respectively. Feng et al. [21,22] described a constraint-based inter-transactional association mining approach which was template-guided. [20] reported the application of multidimensional inter-transactional associations to meteorological data study.

However, the previous problem definition of inter-transactional association rule mining as introduced in [36,37] has the limitation because the contexts of association relationships explored are very rigid. If we view each database transaction as a point in an *m*-dimensional space, each item in such an inter-transactional association rule must come from a single transaction and be located at a certain contextual point, although the rule itself may embrace many items from different transactions. For ease of explanation, we refer to this kind of inter-transactional association rules as *point-wise inter-transactional association rules*, since only items occurring at different *contextual points* are explored with their correlationships detected.

Nevertheless, in real situations, the context under investigation is not always uniform due to the presence of possible data holes, which are analogous to the "missing parts" of a jigsaw puzzle. These data holes may be meaningless contexts for the occurrence of any database transaction. Taking the above fast-food outlet rule for example, if one mile away from McDonald restaurant flows a wide river in some areas, then it would not be possible for any shop to be set up there. These areas thus give negative supports to the rule which in fact describes the reality that fast-food outlets usually gather together. When the mining context contains a number of holes like this, there is the risk that some rules reflecting regularities will receive unreasonably lower support/confidence compared to real situations, and some of them may be neglected by the data miners.

1.2. Generalized multidimensional inter-transactional association rules

The mining context problem of the point-wise inter-transactional association framework, however, can be rectified by patching data holes while performing the mining. In other words, we can expand rule contexts from *point-wise* (e.g., *one mile away*) to *scope-wise* (e.g., *within one mile and three miles away*). In fact, for many applications, it does not matter whether an item in an inter-transactional association rule is within a single transaction or a group of transactions, provided that the contextual scope where these transactions locate is meaningful and of interest to applications. By context expansion, we can enhance the flexibility and expressiveness of inter-transactional association framework to capture more comprehensive and general knowledge like:

R_1'': "When the prices of IBM and SUN go up, 80% of the time the price of Microsoft increases *within three days*."

R_2'': "If there is no rain *within 6 hours* and the wind direction continues to be moderate *during the following 24 hours*, then there will be no rain *for 2 days*."

R_3'': "After McDonald and Burger King open branches, KFC will open a branch *within two months* and *between one and three miles away*."

In this paper, we extend the previous problem definition of multidimensional inter-transactional association rules given in [36,37] based on context expansion. We call such extended association rules *generalized multidimensional inter-transactional association rules*, since they provide a *uniform* view for a number of association and sequence-related patterns defined before. An algorithm for mining generalized 1-dimensional inter-transactional association rules is presented by extension of a priori. We conduct experiments on both real-life and synthetic data sets to study the performance of the algorithm. Further extension of the algorithm to a multidimensional context is also discussed.

The remainder of the paper is organized as follows. Section 2 provides a brief review of point-wise multidimensional inter-transactional association rules, based on which a generalized multidimensional inter-transactional association rule framework is presented in Section 3. Section 4 describes an algorithm for mining such generalized association rules. Our experiments that evaluate the performance of the algorithm on both synthetic and real-life data sets are presented in Section 5. Further generalization to multidimensional inter-transactional association rule mining is given in Section 6. We review some closely related work in Section 7, and make concluding remarks in Section 8.

2. Point-wise multidimensional inter-transactional association rules

Inter-transactional association rules extend the traditional intra-transactional ones by incorporating contextual information into the association rule mining. To this end, a series of concepts in the traditional association rule framework are extended. These include *multidimensional context, extended transaction/item, normalized extended transaction/item set*, and *containing relationship.* The Inter-transactional association rule framework is built based on these extended notions.

2.1. Multidimensional contexts

In classical association rule mining, records in a transactional database contain only items and are identified by their transaction identifiers. 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 is intra-transaction in nature.

However, when we talk about inter-transactional associations across multiple transactions, the occurrence contexts of transactions become important and must be taken into account. Virtually, an *m-dimensional context* can be defined through m-dimensional attributes $a_1, a_2, \ldots, a_m$, whose domains $\mathrm{Dom}(a_1)$, $\mathrm{Dom}(a_2), \ldots, \mathrm{Dom}(a_m)$ are finite subsets of non-negative integers. Each dimension of the context is represented by a dimensional attribute, which can be of any kind as long as it is meaningful to applications. Time, distance, temperature, latitude, etc., are typical dimensional attributes. For a 1-dimensional stock movement database, the only dimensional attribute could be the *trading date*. For a meteorological database where each transaction records observations of various meteorological elements taken at a certain time in a certain region, there are 2-dimensional attributes, namely, *time* and *region*. The occurrence context of association rules to be examined later is constructed by such an m-dimensional space. When $m = 1$, we have a single-dimensional context.

Let $n_l = (a_{l,1}, a_{l,2}, \ldots, a_{l,m})$ and $n_u = (a_{u,1}, a_{u,2}, \ldots, a_{u,m})$ be two contextual points in an m-dimensional space, whose values on the m dimensions are denoted as $a_{l,1}, a_{l,2}, \ldots, a_{l,m}$ and $a_{u,1}, a_{u,2}, \ldots, a_{u,m}$, respectively. We define that

(1) $(n_l = n_u)$ if and only if $\forall s\ (1 \leqslant s \leqslant m)\ (a_{l,s} = a_{u,s})$;
(2) $(n_l \preceq n_u$, conversely $n_u \succeq n_l)$ if and only if $\forall s\ (1 \leqslant s \leqslant m)\ (a_{l,s} \leqslant a_{u,s})$;
(3) $(n_l \prec n_u$, conversely $n_u \succ n_l)$ if and only if $(n_l \preceq n_u) \wedge \exists s(1 \leqslant s \leqslant m)$ $(a_{l,s} < a_{u,s})$.

Along with the introduction of multidimensional contexts, the traditional concepts of *transaction* and *item* are extended accordingly.

2.2. Extended transactions and normalized extended transaction sets

Let $\mathscr{I} = \{i_1, i_2, \ldots, i_\omega\}$ be a set of items. A traditional transactional database is a set of transactions $\mathscr{T} = \{t_1, t_2, \ldots, t_n\}$, where each transaction in $\mathscr{T}$ is a subset of $\mathscr{I}$. Such a database model is enhanced under an m-dimensional context by associating each transaction with an m-dimensional attribute value so that each transaction can be mapped to a point in the m-dimensional space, describing its occurrence context. We call a transaction $t \in \mathscr{T}$ happening at an m-dimensional contextual point n an *extended transaction*, and denote it as $t(n)$. Let $\mathscr{T}_{\mathrm{E}}$ be the set of all extended transactions in the database.

Example 2.1. Fig. 1 shows a simple transactional database under a 2-dimensional space. The domains of its 2-dimensional attributes X and Y have been discretized into five and four equal-sized intervals, respectively. There are six different items: a, b, c, d, e, f. Table 1 lists all the 20 extended transactions in the database.

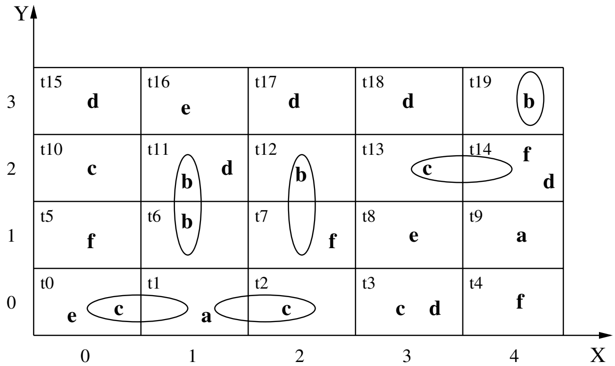


Fig. 1. A simple database under a 2-dimensional context.

Table 1
Extended transactions in the example database of Fig. 1

Transaction	Extended transaction
t_0	$t_0(0,0)$
t_1	$t_1(1,0)$
t_2	$t_2(2,0)$
t_3	$t_3(3,0)$
t_4	$t_4(4,0)$
t_5	$t_5(0,1)$
t_6	$t_6(1,1)$
t_7	$t_7(2,1)$
t_8	$t_8(3,1)$
t_9	$t_9(4,1)$
t_{10}	$t_{10}(0,2)$
t_{11}	$t_{11}(1,2)$
t_{12}	$t_{12}(2,2)$
t_{13}	$t_{13}(3,2)$
t_{14}	$t_{14}(4,2)$
t_{15}	$t_{15}(0,3)$
t_{16}	$t_{16}(1,3)$
t_{17}	$t_{17}(2,3)$
t_{18}	$t_{18}(3,3)$
t_{19}	$t_{19}(4,3)$

Given a set of extended transactions, $T_e = \{t_1(n_1), t_2(n_2), \ldots, t_r(n_r)\}$ where $n_x = (a_{x,1}, a_{x,2}, \ldots, a_{x,m})$ $(1 \leqslant x \leqslant r)$, T_e is called a *normalized extended transaction set* if within all the contextual points involved, the minimal dimensional value is 0 along each dimension, i.e., $\forall s\,(1 \leqslant s \leqslant m)\ \min(a_{1,s}, a_{2,s}, \ldots, a_{r,s}) = 0$.

Any non-normalized extended transaction set can be transformed into a normalized one through a normalization function called *Point-Norm*, whose intention is to re-position all contextual points in the extended transaction set based on its minimal dimensional value along every dimension.

Example 2.2. Let $T_1 = \{t_0(0,0), t_1(1,0), t_6(1,1), t_{11}(1,2)\}$ and $T_2 = \{t_1(1,0), t_2(2,0), t_7(2,1), t_{12}(2,2)\}$ be two extended transaction sets in the 2-dimensional context in Fig. 1. T_1 is a normalized extended transaction set, since it has minimal dimensional value 0 along both X and Y axes, i.e., $\min(0,1,1,1) = 0$ and $\min(0,0,1,2) = 0$.

But T_2 is not due to its non-zero minimal dimensional value along the X dimension, which is $\min(1,2,2,2) = 1$. We can, however, normalize T_2 by subtracting this minimal value 1 from each X-coordinate of the four points, and re-position the contextual points in T_2: $\textit{Point-Norm}(T_2) = \{t_1(1-1,0), t_2(2-1,0), t_7(2-1,1), t_{12}(2-1,2)\} = \{t_1(0,0), t_2(1,0), t_7(1,1), t_{12}(1,2)\}$. The obtained extended transaction set then becomes a normalized one.

2.3. Extended items and normalized extended item sets

By associating contextual information with items, we can come up with extended items. We call an item $i \in \mathscr{I}$ happening at an m-dimensional contextual point n an *extended item*, and denote it as $i(n)$. Two extended items $i(n)$ and $i'(n')$ are equal, denoted as $i(n) = i'(n')$, if and only if $(i = i')$ and $(n = n')$. The set of all possible extended items, $\mathscr{I}_E$, is defined as a set of $i(n)$ for any $i \in \mathscr{I}$ at all possible points n in the m-dimensional space.

The normalized extended item set is defined in a similar fashion as the normalized extended transaction set. Let $I_e = \{i_1(n_1), i_2(n_2), \ldots, i_k(n_k)\}$ be an extended item set, where $n_x = (a_{x,1}, a_{x,2}, \ldots, a_{x,m})$ $(1 \leqslant x \leqslant k)$. We call I_e a *normalized extended item set* if within all the contextual points involved, the minimal dimensional value is 0 along each dimension, i.e., $\forall s$ $(1 \leqslant s \leqslant m)$ $\min(a_{1,s}, a_{2,s}, \ldots, a_{k,s}) = 0$. Let $\mathscr{I}_{NE}$ denote the set of all possible normalized extended item sets in the database.

Similarly, we can transform a non-normalized extended item set into a normalized one through function *Point-Norm* by re-positioning all contextual points in the extended item set based on its minimal dimensional value along every dimension.

2.4. Containing relationship

We now define when an extended transaction set contains an extended item set. Let T_e be an extended transaction set, and let $I_{ne} = \{i_1(n_1), i_2(n_2), \ldots, i_k(n_k)\}$ be a normalized extended item set. T_e is said to *contain* I_{ne} if and only if (1) $\forall i_x(n_x) \in I_{ne}$, $\exists t(n_x) \in \textit{Point-Norm}(T_e)$, such that $(i_x \in t)$; and (2) $\forall t(n_x) \in \textit{Point-Norm}(T_e)$, $\exists i_x(n_x) \in I_{ne}$. The first condition states that for each extended item in I_{ne}, we can find an extended transaction with the same context location after normalizing T_e, which contains i_x. The second condition ensures the minimal consideration for the transaction set, i.e., only transactions after normalization having the same context locations as extended items in I_{ne} are included in T_e.

Example 2.3. Given a normalized extended item set $I = \{c(0,0), b(1,2)\}$, from the database shown in Fig. 1, we can only find one extended transaction set containing I, which is $T = \{t_0(0,0), t_{11}(1,2)\}$. It has two transactions located at (0,0) and (1,2) with item c and b included, respectively.

With the notions of multidimensional context, extended transaction/item, normalized extended transaction/item set, and containing relationship as described above, we can formally define point-wise multidimensional inter-transactional association rules as follows.

2.5. A formal definition of point-wise multidimensional inter-transactional association rules

Definition 2.1. *A point-wise multidimensional inter-transactional association rule* is an implication of the form $X \Rightarrow Y$, where $X, Y \subset \mathscr{I}_E$, $X \cup Y \subset \mathscr{I}_{NE}$, and $X \cap Y = \emptyset$.

Different from classical intra-transactional association rules, a point-wise inter-transactional association rule provides the occurrence context for associated items by means of a normalized extended item set $X \cup Y$. For example, a rule that predicts the stock price movement—"if stock '*a*' increases one day, and stock '*c*' increases the following day, then most probably stock '*e*' will increase on the fourth day", can be expressed by a point-wise 1-dimensional inter-transactional association rule "$a(0), c(1) \Rightarrow e(3)$".

Similar to intra-transactional association rules, we use *support* and *confidence* as two major measurements for inter-transactional association rules. 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 multidimensional inter-transactional association rules which may span different transactions, the traditional support concept must be extended accordingly from the original *single-transaction-based* to *transaction-set-based.*

Definition 2.2. Given a normalized extended item set X and an extended item set Y, let $|T_{xy}|$ be the total number of extended transaction sets that contain $X \cup Y$, $|T_E|$ be the total number of extended transaction sets in the database, and $|T_x|$ be the total number of extended transaction sets that contain X. The *support* and *confidence* of a point-wise multidimensional inter-transactional association rule $X \Rightarrow Y$ is defined as: *support*$(X \Rightarrow Y) = |T_{xy}| / |\mathscr{T}_E|$ and *confidence*$(X \Rightarrow Y) = |T_{xy}|/|T_x|$.

3. Generalized scope-wise multidimensional inter-transactional association rules

To enhance the flexibility and expressiveness of discovered association rules so as to cover broader real situations, the *point-wise* multidimensional inter-transactional association rule framework described in the previous section is further generalized into a *scope-wise* multidimensional inter-transactional association rule framework. In this section, we focus our discussion on the generalization parts, based on which a formal definition of the generalized multidimensional inter-transactional association rule framework, its properties, and related measurements are then described.

3.1. Expanding contextual points to contextual scopes

The contextual information carried by the point-wise inter-transactional association rules is explicated by contextual points. To overcome the limitation of such a rigid context expression, here, we landscape the mining context by expanding the *point-wise* contextual representation to *scope-wise* contextual representation. Basically, an *m-dimensional contextual scope*, denoted as $[n_l, n_u]$, is delimited by two m-dimensional contextual points n_l and n_u where $n_l \preceq n_u$. A point n_i lies within the scope $[n_l, n_u]$, denoted as $n_i \in [n_l, n_u]$, if and only if $(n_l \preceq n_i \preceq n_u)$. Given two contextual scopes $[n_l, n_u]$ and $[n'_l, n'_u]$ in an m-dimensional space, we define the following three boolean comparison operators:

(1) *inclusive*$([n_l, n_u], [n'_l, n'_u])$ is true, if and only if $(n_l \preceq n'_l) \wedge (n'_u \preceq n_u)$.
(2) *intersect*$([n_l, n_u], [n'_l, n'_u])$ is true, if and only if $\exists n_i (n_i \in [n_l, n_u]) \wedge (n_i \in [n'_l, n'_u])$.
(3) *precedence*$([n_l, n_u], [n'_l, n'_u])$ is true, if and only if $(n_u \prec n'_l)$.

Property 3.1. *The binary comparison operators defined on contextual scopes have the following properties*:

(1) *inclusive operator is reflexive and transitive, i.e.*,
  - *inclusive*$([n_l, n_u], [n_l, n_u])$;
  - *inclusive*$([n_l, n_u], [n'_l, n'_u]) \wedge$ *inclusive*$([n'_l, n'_u], [n''_l, n''_u]) \rightarrow$ *inclusive*$([n_l, n_u], [n''_l, n''_u])$.
(2) *intersect operator is reflexive and symmetric, i.e.*,
  - *intersect*$([n_l, n_u], [n_l, n_u])$;
  - *intersect*$([n_l, n_u], [n'_l, n'_u]) \rightarrow$ *intersect*$([n'_l, n'_u], [n_l, n_u])$.

(3) *precedence operator is transitive, i.e.,*
- $precedence([n_l, n_u], [n'_l, n'_u]) \wedge precedence([n'_l, n'_u], [n''_l, n''_u]) \rightarrow precedence([n_l, n_u], [n''_l, n''_u])$.

Example 3.1. Assume we have three contextual scopes $[n_1,n_2]$, $[n_3,n_4]$ and $[n_1,n_3]$ under a 2-dimensional space, where $n_1 = (0,0)$, $n_2 = (1,2)$, $n_3 = (3,3)$, $n_4 = (4,5)$. The following relations hold: *inclusive*$([n_1,n_3],[n_1,n_2])$ since $(n_1 \preceq n_1) \wedge (n_3 \succ n_2)$; *precedence*$([n_1,n_2],[n_3,n_4])$ since $(n_2 \prec n_3)$; and *intersect*$([n_1,n_3], [n_3,n_4])$ since $(n_3 \in [n_1,n_3]) \wedge (n_3 \in [n_3,n_4])$.

3.2. Widely-extended items and normalized widely-extended item sets

An extended item defined in Section 2 associates a contextual point with the item, indicating its occurrence context. We enlarge this occurrence context indicator further using a contextual scope instead of a contextual point. To distinguish from the previous definition, we call an item $i \in \mathscr{I}$ happening within an m-dimensional contextual scope $[n_l, n_u]$ a *widely-extended item*, and denote it as $i_{[n_l, nu]}$. Given two widely-extended items $i_{[n_l, nu]}$ and $i'_{[n'_l, n'_u]}$, we define that

(1) $(i_{[n_l,n_u]} = i'_{[n'_l,n'_u]})$ if and only if $(i = i') \wedge (n_l = n'_l) \wedge (n_u = n'_u)$;
(2) $(i_{[n_l,n_u]} < i'_{[n'_l,n'_u]})$ if and only if $(i < i') \vee (i = i' \wedge precedence([n_l, n_u], [n'_l, n'_u]))$.

The set of all possible widely-extended items, $\mathscr{I}_{\mathrm{WE}}$, is defined as a set of $i_{[n_l, nu]}$ for any $i \in \mathscr{I}$ within all possible scopes $[n_l, n_u]$ where $n_l \preceq n_u$ in the m-dimensional space.

Let $I_{\mathrm{we}} = \{i_{1[n_{1,l},n_{1,u}]}, i_{2[n_{2,l},n_{2,u}]}, \ldots, i_{k[n_{k,l},n_{k,u}]}\}$ be a widely-extended item set, where $n_{x,l} = (a_{x,l,1}, a_{x,l,2}, \ldots, a_{x,l,m})$ and $n_{x,u} = (a_{x,u,1}, a_{x,u,2}, \ldots, a_{x,u,m})$ $(1 \leqslant x \leqslant k)$. We call I_{we} a *normalized widely-extended item set*, if within all the contextual scopes involved, the minimal dimensional value is 0 along each dimension, i.e., $\forall s$ $(1 \leqslant s \leqslant m)$ $\min(a_{1,l,s}, a_{1,u,s}, \ldots, a_{k,l,s}, a_{k,u,s}) = 0$. Let $\mathscr{I}_{\mathrm{NWE}}$ denote the set of all possible normalized widely-extended item sets in the database.

In contrast to the normalization function *Point-Norm* defined on extended transaction/item sets, another normalization function called *Scope-Norm* can be introduced, whose task is to re-position all contextual scopes instead of points involved in an widely-extended item set based on its minimal dimensional value along every dimension.

Example 3.2. Let $I_1 = \{c_{[(0,0),(1,0)]}, b_{[(1,1),(1,2)]}\}$ and $I_2 = \{c_{[(1,0),(2,0)]}, b_{[(2,1),(2,2)]}\}$ be two widely-extended item sets in a 2-dimensional space. I_1 is a normalized extended item set, since it has minimal value 0 for both dimensions, i.e., $\min(0,1,1,1) = 0$ and $\min(0,0,1,2) = 0$.

However, I_2 is not because it has a non-zero minimal value min(1, 2, 2, 2) = 1 for the first dimension. We can normalize I_2 by subtracting this minimal value 1 from the four delimiting points' first dimensional values, i.e., $[(1-1,0), (2-1,0)] = [(0,0),(1,0)]$, $[(2-1,1),(2-1,2)] = [(1,1),(1,2)]$, and obtain a normalized widely-extended item set through the Scope-Norm function, i.e., $\mathit{Scope\text{-}Norm}(I_2) = \{c_{[(0,0),(1,0)]}, b_{[(1,1),(1,2)]}\}$.

Property 3.2. *Any superset of a normalized widely-extended item set is also a normalized widely-extended item set.*

Proof. Let $I_{\text{nwe}} = \{i_{1[n_{1,l},n_{1,u}]}, i_{2[n_{2,l},n_{2,u}]}, \ldots, i_{k[n_{k,l},n_{k,u}]}\}$ be a normalized widely-extended item set. Assume $I'_{\text{nwe}} = \{i_{1[n_{1,l},n_{1,u}]}, i_{2[n_{2,l},n_{2,u}]}, \ldots, i_{k[n_{k,l},n_{k,u}]}, i_{k+1[n_{k+1,l},n_{k+1,u}]}\}$ is a superset of I_{nwe} without loss of generality. According to the definition of normalized widely-extended item set, for $\forall s$ $(1 \leqslant s \leqslant m)$, $\min(a_{1,l,s}, a_{1,u,s}, \ldots, a_{k,l,s}, a_{k,u,s}) = 0$.

Since the domains of dimensional attributes are non-negative integers, i.e., $\forall s$ $(1 \leqslant s \leqslant m)$ $(a_{k+1,l,s} \geqslant 0) \wedge (a_{k+1,u,s} \geqslant 0)$, therefore, $\min(a_{1,l,s}, a_{1,u,s}, \ldots, a_{k,l,s}, a_{k,u,s}, a_{k+1,l,s}, a_{k+1,u,s}) = 0$, implying that I'_{nwe} is also a normalized widely-extended item set. □

Property 3.2 forms the basis for generating candidate widely-extended itemsets during the mining process to be discussed later.

3.3. Containing relationship revisited

As a widely-extended item may span several contextual points, it is necessary to re-define the containing relationship between an extended transaction set and a normalized widely-extended item set.

Let $I_{\text{nwe}} = \{i_{1[n_{1,l},n_{1,u}]}, i_{2[n_{2,l},n_{2,u}]}, \ldots, i_{k[n_{k,l},n_{k,u}]}\}$ be a normalized widely-extended item set. We define that an extended transaction set T_{e} *contains* I_{nwe}, if and only if

(1) $\forall i_{x[n_{x,l},n_{x,u}]} \in I_{\text{nwe}} \exists t(n_x) \in \mathit{Point\text{-}Norm}(T_{\text{e}})$, $n_x \in [n_{x,l}, n_{x,u}] \wedge (i_x \in t)$; and
(2) $\forall t\ (n_x) \in \mathit{Point\text{-}Norm}(T_{\text{e}}) \exists i_{x[n_{x,l},n_{x,u}]} \in I_{\text{nwe}}, n_x \in [n_{x,l}, n_{x,u}]$.

The first condition states that for each widely-extended item in I_{nwe}, there should exist an extended transaction in *Point-Norm*(T_{e}), which is located within the contextual scope of this item, and meanwhile contains the item. The second condition requires each extended transaction in *Point-Norm*(T_{e}) to be within one of the contextual scopes in I_{nwe}, ensuring the minimal consideration for the extended transaction set T_{e}.

3.4. Formal definition of generalized multidimensional inter-transactional association rules

Definition 3.1. A *generalized multidimensional inter-transactional association rule* is an implication of the form $X \Rightarrow Y$, which satisfies the following two conditions:

(1) $X \subset \mathscr{I}_{\mathrm{WE}}$, $Y \subset \mathscr{I}_{\mathrm{WE}}$, $X \cup Y \subset \mathscr{I}_{\mathrm{NWE}}$, $X \cap Y = \emptyset$;
(2) $\forall i_{[n_{s,l},n_{s,u}]} \in (X \cup Y)$, $\nexists i_{[n'_{s,l},n'_{s,u}]} \in (X \cup Y)$ where *intersect* $([n_{s,l}, n_{s,u}], [n'_{s,l}, n'_{s,u}])$ is true.

The first clause of the definition indicates that only a normalized widely-extended item set $X \cup Y$ is considered by a rule. The second clause requires that no two widely-extended items with the same item but intersected contextual scopes co-exist in one rule. This is to avoid verbose rules like "$\underline{a_{[(0),(0)]}}, \underline{a_{[(0),(2)]}} \Rightarrow b_{[(2),(4)]}$", "$a_{[(0),(0)]} \Rightarrow \underline{b_{[(2),(2)]}}, \underline{b_{[(2),(4)]}}$" or "$\underline{a_{[(0),(0)]}} \Rightarrow \underline{a_{[(0),(2)]}}, b_{[(2),(4)]}$", since the presence of $a_{[(0),(0)]}$ implies the presence of $a_{[(0),(2)]}$, and so does the pair of $b_{[(2),(2)]}$ and $b_{[(2),(4)]}$.

Based on Definition 3.1, a rule like "if there is no rain within 6 hours and the weather is medium wet during the following 24 hours, then there will be no rain for 2 days" can be expressed by a generalized 1-dimensional inter-transactional association rule "$\textit{no-rain}_{[(0),(1)]}, \textit{medium-wet}_{[(2),(5)]} \Rightarrow \textit{no-rain}_{[(2),(9)]}$". Here, each interval unit represents 6 h.

The support and confidence for generalized multidimensional inter-transactional association rules can be defined in a similar way as point-wise multidimensional inter-transactional association rules.

Definition 3.2. Let X and Y be two subsets of a normalized widely-extended itemset $\mathscr{I}_{\mathrm{nwe}}$. Further let T_{xy} be the set of extended transaction sets that contain $X \cup Y$, and T_x be the set of extended transaction sets that contain X. The *support* and *confidence* of a generalized multidimensional inter-transactional association rule $X \Rightarrow Y$ approximate to $\textit{support}(X \Rightarrow Y) = |T_{xy}|/|\mathscr{T}_{\mathrm{E}}|$ and $\textit{confidence}(X \Rightarrow Y) = |T_{xy}|/|T_x|$.

Note that the important monotonic property regarding the support of itemsets is still valid under the generalized multidimensional inter-transactional association mining framework. In other words, the support of a widely-extended itemset X will not be larger than the support of any of its subsets X', because $\textit{sup}(X) = \frac{|T_x|}{|\mathscr{T}_{\mathrm{E}}|} \leqslant \frac{|T_{x'}|}{|\mathscr{T}_{\mathrm{E}}|} = \textit{sup}(X')$. This property is desirable since it is the base for a large set of efficient association rule mining algorithms developed in the literature. The approach of mining generalized inter-transactional association relationships, to be described in the next section, is also an extension of the classic a priori algorithm, working on the basis of this monotonic property.

4. Mining generalized 1-dimensional inter-transactional association rules

Given a user-specified minimum support (called *minsup*) and minimum confidence (called *minconf*), our task is to discover from a contextual space a complete set of generalized inter-transactional association rules with $support \geqslant minsup$ and $confidence \geqslant minconf$. Like classical association rule mining, the problem of mining generalized inter-transactional association rules can be decomposed into two subproblems:

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

Of the two subproblems, subproblem 1 is of major concern as it is the bottleneck of the whole mining process mainly due to two reasons. First, compared to the previous intra-transactional and point-wise inter-transactional association rule mining, more candidate widely-extended itemsets are expected to be generated when mining generalized inter-transactional association relationships. Second, counting each candidate widely-extended itemset requires scanning a set of transactions instead of one, incurring a much larger search space than traditional association mining. On the other hand, subproblem 2 can be easily solved by making minor modifications to a fast algorithm given in [5]. As such, in this section, we focus our discussion on the first subproblem. Fig. 2 outlines a generalized 1-dimensional inter-transactional association mining algorithm by extension of a priori [5]. Further generalization to a multidimensional mining context is discussed in Section 6.

To simplify expressions, we omit bracket ( ) surrounding coordinates of points under 1-dimension, and use $[l, u]$ for $[(l), (u)]$. Also, *itemset* and *widely-extended itemset* are used interchangeably in the following discussions.

Like a priori, our algorithm performs in a level-wise manner. Let C_k represent the set of candidate k-itemsets, and L_k represent the set of large k-itemsets. 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. The algorithm then scans the database.

Input: a database DB containing an extended transaction set $\mathcal{T}_E$ under an 1-dimensional context; a *minsup* threshold and a maximal contextual scope *maxscope* of interest to applications.
Output: a set of large normalized widely-extended itemsets L discovered from the database.

k=1

$L_1 = \emptyset$;
$C_1 = \{\{i_{[l,\ u]}\} \mid (i \in \mathcal{I}) \ \wedge \ (l \le u) \ \wedge \ (0 \le l \le maxscope) \ \wedge (0 \le u \le maxscope)\}$;
foreach extended transaction $t_s(s) \in \mathcal{T}_E$ **do**
    $T_s = \{t_{s+d}(s+d) \mid (t_{s+d}(s+d) \in \mathcal{T}_E) \ \wedge \ (0 \le d \le maxscope)\}$;
    foreach extended transaction $t_{s+d}(s+d) \in T_s$ **do**
        foreach item $i \in t_{s+d}$ **do**
            for $(u = d;\ u \le maxscope;\ u++)$ **do**
                $i_{[d,\ u]}.count$++;   //increase along arrow ↑ shown in Figure 3
                for $(l = d$-1; $l \ge 0$; l- -) **do**
                    if $(i \notin t_{s+l})$ **then** $i_{[l,\ d]}.count$++; //increase along arrow ↖
                    else break;
            endfor
    Transform-Record-TranSet $(T_s,\ DB')$;
endfor
$L_1 = \{\{i_{[l,\ u]}\} \mid (i_{[l,\ u]} \in C_1) \ \wedge \ (i_{[l,\ u]}.count/|\mathcal{T}_E| \ge minsup)\}$

k>1

for $(k = 2;\ L_{k-1} \ne \phi;\ k++)$ **do**
    C_k=E-Apriori-Gen(L_{k-1});
    foreach record $r \in DB'$ **do**
        C_r = E-Subset $(C_k\ , r)$;   // candidates contained in record
        foreach candidate $X : \{i_{1[l_1,\ u_1]}, \ldots, i_{k[l_k,\ u_k]}\} \in C_r$ **do** $X.count$++;
    endfor
    $L_k = \{X : \{i_{1[l_1,\ u_1]}, \ldots, i_{k[l_k,\ u_k]}\} \mid (X \in C_k) \ \wedge \ (X.count/|\mathcal{T}_E| \ge minsup)\}$;
endfor
$L = \bigcup_k L_k$.

Fig. 2. A generalized 1-dimensional inter-transactional association mining algorithm.

From every extended transaction located at a certain point in the 1-dimensional space, it examines an extended transaction set nearby that possibly contains candidates, determines which candidates in C_k are actually contained, and increments their counts. At the end of the pass, C_k is examined to check which of the candidates are large, yielding L_k. The algorithm terminates when L_k becomes empty, as no candidate itemsets can be further generated.

4.1. Pass 1

4.1.1. Generation of candidate set C_1

Considering that users are usually interested in associations happening within certain limited scopes, here, we introduce a *maxscope* threshold to specify the maximal scope. Given *maxscope* = 3, Fig. 3 illustrates all the contextual scopes considered in our mining task. To generate candidate set C_1, for each item in $\mathcal{I}$, we attach all these possible contextual scopes, and obtain

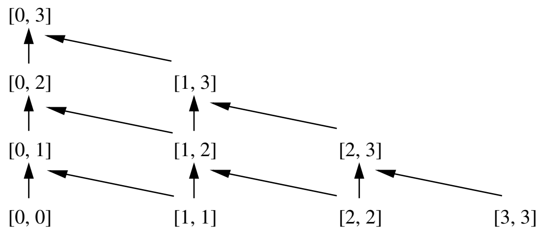


Fig. 3. 1-Dimensional contextual scopes considered when *maxscope* = 3.

$C_1 = \{\{i_{[l,u]}\}|(i \in \mathscr{I}) \wedge (l \leqslant u) \wedge (0 \leqslant l \leqslant maxscope) \wedge (0 \leqslant u \leqslant maxscope)\}$. Hence, $|C_1| = |\mathscr{I}| * |\sum_{u=l}^{maxscope}(\sum_{l=0}^{maxscope} i_{[l,u]})| = |\mathscr{I}| * (maxscope + 1) * (maxscope + 2)/2$.

4.1.2. Counting candidates in C_1

As candidate 1-itemsets (e.g., $a_{[0,3]}, a_{[1,1]}, a_{[2,3]}$) may span a set of transactions within contextual scopes (from 0 to *maxscope* at most), to count supports of candidates, we examine from each extended transaction $t_s(s)$ in the database, [1] a set of normalized extended transactions T_s instead of only itself (cf. line 4 of Fig. 2).

Property 4.1. *Let* $T_s = \{t_{s+d}(s+d)|(t_{s+d}(s+d) \in \mathscr{T}_{\mathrm{E}}) \wedge (0 \leqslant d \leqslant maxscope)\}$ *be an extended transaction set. For any two extended* 1*-itemsets* $i_{[l,u]}$ *and* $i_{[l',u']}$, *where* $(i \in \mathscr{I})$ *and inclusive*$([l',u'],[l,u])$ *is true, if* T_s *contains* $i_{[l,u]}$, *then* T_s *also contains* $i_{[l',u']}$.

Proof. T_s contains $i_{[l,u]}$, implying that there exists $t_{s+d}(s+d) \in T_s$, such that $(d \in [l,u]) \wedge (i \in t_{s+d})$. Because of *inclusive*$([l',u'],[l,u])$, $d \in [l,u]$ implies $d \in [l',u']$. Thus, T_s also contains $i_{[l',u']}$. □

When an item i appears in a transaction $t_{s+d}(s+d) \in T_s$ (cf. line 5), based on Property 4.1, the algorithm (cf. lines 6–12) increases not only the counter of $i_{[d,d]}$, but also the counters of all those 1-itemsets $i_{[l,u]}$ where *inclusive* $([l,u],[d,d])$ holds along vertical and slanted arrows in Fig. 3. Note that for a given extended transaction set T_s, line 10 ensures each counter increases at most by 1.

[1] As each contextual position has at most one transaction, here we use the position as subscript to name each transaction.

4.1.3. Database transformation

Compared with intra-transactional association mining, from each transaction, we need to scan (*maxscope* + 1) times the number of items. In order to save this extra search effort for the following passes, another work conducted during pass 1 is to transform and record every extended transaction set T_s into a new database DB'. This is performed by the function Transform-Record-TranSet (T_s, DB') (cf. line 13). Each record in DB' has the format:

$$(ID_s, item_1, occurNum_1, posi_1, \ldots, posi_{occurNum_1},$$
$$\ldots,$$
$$item_r, occurNum_r, posi_1, \ldots, posi_{occurNum_r}),$$

where ID_s is the new record identifier, $item_1, \ldots, item_r$ are the item identifiers in T_s, each of which is followed by $occurNum_1, \ldots, occurNum_r$ giving how many transactions (in T_s) the corresponding item appears in, and the normalized positions of these transactions (within $[0, maxscope]$) after normalization. All items in the new record ID_s are sorted in the ascending order of the item identifiers.

4.2. Pass $k > 1$

4.2.1. Candidate generation

Given L_{k-1}, the candidate generation function E-a priori-Gen(L_{k-1}) returns a superset C_k of L_k (cf. line 17). This procedure has two parts: the join phase and prune phase. In the join phase, a candidate k-itemset X'' is generated from two large $(k-1)$-itemsets X and X', where $X = \{x_{1[l_1 u_1]}, \ldots, x_{k-2[l_{k-2}, u_{k-2}]}, x_{k-1[l_{k-1}, u_{k-1}]}\}$, $X' = \left\{x'_{1[l'_1, u'_1]}, \ldots, x'_{k-2[l'_{k-2}, u'_{k-2}]}, x'_{k[l'_k, u'_k]}\right\}$, $X'' = \{x_{1[l_1, u_1]}, \ldots, x_{k-2[l_{k-2}, u_{k-2}]}, x_{k-1[l_{k-1}, u_{k-1}]}, x'_{k[l'_k, u'_k]}\}$, [2] satisfying the following three requirements:

(1) $X, X' \in L_{k-1}$;
(2) $(x_{1[l_1, u_1]} = x'_{1[l'_1, u'_1]}), \ldots, (x_{k-2[l_{k-2}, u_{k-2}]} = x'_{k-2[l'_{k-2}, u'_{k-2}]})$, $(x_{k-1[l_{k-1}, u_{k-1}]} < x_{k[l_k, u_k]})$;
(3) $X'' \in \mathscr{I}_{NE}$ (i.e., X'' is a normalized widely-extended itemset).

To obtain normalized candidate 2-itemsets, we have either $(l_1 = 0)$ in $X = \{x_{1[l_1, u1]}\}$ or $(l'_1 = 0)$ in $X' = \{x'_{1[l'_1, u'_1]}\}$ when $k = 2$. When $k > 2$, according to Property 3.2, since X, X' are normalized widely-extended itemsets, the superset of them $X'' = X \cup X'$ is also a normalized widely-extended itemset.

[2] The widely-extended items in an itemset are listed in an ascending order. Recall that $(i_{[n_l, n_u]} < i'_{[n'_l, n'_u]})$ iff $(i < i') \vee (i = i' \wedge precedence([n_l, n_u], [n'_l, n'_u]))$.

Next, in the prune phase, we delete all those itemsets in C_k which have some ($k-1$)-subsets whose normalization forms are not in L_{k-1}.

Example 4.1. Let $L_2 = \{\{a_{[0,0]}, b_{[0,1]}\}, \{a_{[0,0]}, c_{[1,3]}\}, \{b_{[0,1]}, c_{[1,3]}\}, \{b_{[0,1]}, d_{[2,3]}\}\}$. After the join step, $C_3 = \{\{a_{[0,0]}, b_{[0,1]}, c_{[1,3]}\}, \{b_{[0,1]}, c_{[1,3]}, d_{[2,3]}\}\}$. The prune step will delete the itemset $\{b_{[0,1]}, c_{[1,3]}, d_{[2,3]}\}$, because its subset $\{c_{[1,3]}, d_{[2,3]}\}$ (i.e., $\{c_{[0,2]}, d_{[1,2]}\}$ after normalization) is not in L_2. We will then be left with $C_3 = \{\{a_{[0,0]}, b_{[0,1]}, c_{[1,3]}\}\}$.

4.2.2. Correctness

We need to show that $C_k \subseteq L_k$. Since any subset of a large widely-extended itemset is also large, we will have a superset of L_k if we extend each itemset in L_{k-1} with all possible widely-extended items and then delete all those whose normalized ($k-1$)-subsets are not in L_{k-1}. The join performed by E-a priori-Gen (L_{k-1}) in the algorithm is equivalent to extending L_{k-1} with each widely-extended item in the database and dropping out k-itemsets for which the ($k-1$)-itemset obtained by deleting the ($k-1$)th widely-extended item is not in L_{k-1}. The requirement $x_{k-1[l_{k-1}, u_{k-1}]} < x_{k\,[l_k, u_k]}$ ensures no two *intersected* contextual scopes, associated with the same item, exist in a candidate itemset, as required by the rule in Definition 3.1. Thus, after the join step, $C_k \subseteq L_k$. Similarly, the prune step which deletes from C_k all widely-extended itemsets whose normalized ($k-1$)-subsets are not in L_{k-1}, also does not delete any widely-extended itemset that could be in L_k.

4.2.3. Counting candidates in C_k

After generating candidate k-itemsets, the function E-Subset(C_k, r) (cf. line 19) checks which k-itemsets in C_k are supported by a new database record r. To do this, we extract all the item IDs of these itemsets and store them in a hash tree similar to that in [5]. The contextual scopes associated with corresponding item IDs are stored uniformly in *leaf* nodes only.

Starting from the root node, we find all the candidates contained in the record r as follows. If we reach a leaf node, we first find those itemsets which have their item IDs present in r, and then further check whether the occurrence positions of these item IDs are within the specified contextual scopes indicated by the widely-extended itemsets. If so, we add the itemsets to C_r. Considering the situation that one item ID may appear consecutively several times in one widely-extended itemset (but with different contextual scopes of *precedence* relations, e.g., $\{a_{[0,0]}, a_{[1,2]}, b_{[1,1]}, b_{[3,3]}\}$), if we are at an interior node and have reached it by hashing the ith item in r, we hash on each item from the ith item *again* (rather than from the $(i+1)$th item as a priori does in [5]) and recursively apply this procedure to the node in the corresponding bucket. The subset function returns a set of k-itemsets, $C_r \subseteq C_k$, that are supported by r in the new database. We increase all the counters of k-itemsets in C_r by 1 (cf. line 20).

By scanning the transformed database DB' once, we can obtain L_k as desired (cf. line 22).

5. Performance study

To assess the performance of the proposed algorithm, we have conducted a series of experiments on both synthetic and real-life data. The method used to generate synthetic data is described in Section 5.1, while Section 5.2 presents some experimental results from this. Results obtained from real data are described in Section 5.3.

5.1. Generation of synthetic data

The method used by this study to generate synthetic transactions is similar to the one used in [5], with some modifications noted below. Table 2 summarizes the parameters used and their settings.

Transaction sizes are typically clustered around a mean and a few transactions have many items. Typical sizes of large itemsets are also clustered around a mean, with a few large itemsets having a large number of items across different transactions. In this study, a potentially large k-itemset is of the form $\{i_{1[l_1,u_1]}, i_{2[l_2,u_2]}, \ldots, i_{k[l_k,u_k]}\}$, where each item i_j is associated with a contextual scope $[l_j, u_j]$. We first generate a set L of the potentially large itemsets, which may span different transactions, and then assign large itemsets in L to corresponding transactions.

The number of potentially large itemsets is set to $|L|$. A potentially large itemset is generated by first picking the size of the itemset from a Poisson distribution with mean equal to $|I|$. The maximum size of potentially large itemsets is $|MI|$. Items and their contextual scopes in the first large itemset are chosen randomly in the following way. We pick up an item i_j randomly from 1 to N. The contextual scope $[l_j, u_j]$ of i_j is determined by choosing the lower

Table 2
Values of parameters in the experiments

Parameter	Meaning	Setting
$\|D\|$	Number of transactions	20K–100K
$\|T\|$	Average size of the transactions	3–7
$\|MT\|$	Maximum size of the transactions	5–9
$\|L\|$	Number of potentially large widely-extended itemsets	1000
$\|I\|$	Average size of the potentially large widely-extended itemsets	3
$\|MI\|$	Maximum size of the potentially large widely-extended itemsets	5
N	Number of items	800–1200
maxscope	Maximal contextual scope	0–11

bound l_j from 0 to *maxscope*, and the *scope* between l_j and u_j from 1 to (*maxscope* + 1). The upper bound u_j is equal to $l_j + scope - 1$. When $u_j > maxscope$, we subtract l_j from both l_j and u_j to ensure that $u_j \leqslant maxscope$.

To model the phenomenon that large itemsets often have common items and contextual scopes, some fraction of items and their contextual scopes in subsequent large itemsets are chosen from the previous itemset generated. We use an exponentially distributed random variable with mean equal to the *correlation level* to decide this fraction for each itemset. The remaining items and their contextual scopes are picked at random. In the datasets used in the experiments, the correlation level is set to 0.5. Having generated all the items and associated contextual scopes for a large itemset, we normalize the itemset by subtracting its minimum lower bound value from all the contextual scopes of the itemset.

After generating the set L of potentially large itemsets, we then generate transactions in the database. Each transaction is assigned a series of potentially large itemsets. However, upon the generation of a transaction, we must consider a list of consecutive ones starting from it, as items in a large itemset may span different transactions. For example, after selecting the large itemset $\{a_{[0,0]}, b_{[0,2]}, c_{[2,4]}\}$ for current transaction $t_s(s)$, we should assign item a to $t_s(s)$, item b to the transaction $t_x(x)$ (where $x \in [s, s+2]$ is determined by a random function) and item c to the transaction $t_y(y)$ (where $y \in [s+2, s+4]$ is determined randomly).

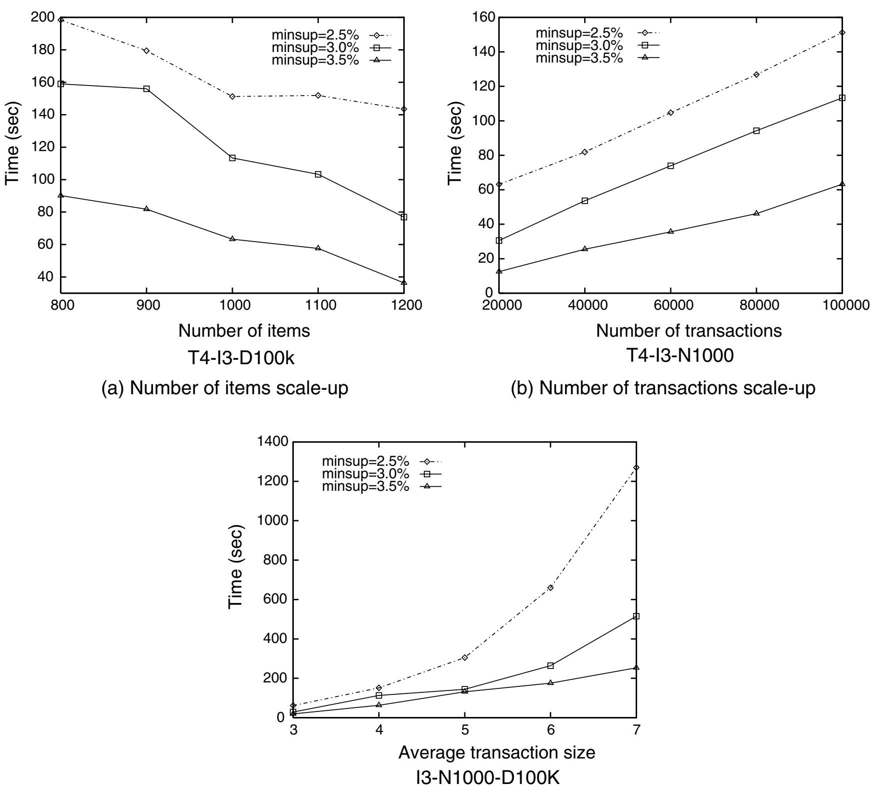
Before assigning items to a list of consecutive transactions, we should determine the sizes of those transactions. The size of each transaction is picked from a Poisson distribution with mean equal to $|T|$. The maximum size of transactions is $|MT|$. Each potentially large itemset has a weight associated with it, which corresponds to the probability that this itemset will be picked. The weight is picked from an exponential distribution with unit mean, and is then normalized so that the sum of the weights for all the itemsets in L is 1. The next itemset to be put in the transaction is chosen from L by tossing an $|L|$-sided weighted coin, where the weight for each side is the probability of picking the associated itemset.

If the large itemset picked on hand does not fit in the current, or any one of its successive transactions, it is put in these transactions anyway in half the cases, and enters an *unfit* queue for the next transaction in the rest of the cases. Each time, we pick itemsets from this queue first, according to the first-in-first-out principle. Only when the queue is empty, do we perform random selection from the set L. As in [5], we use a corruption level during the transaction generation to model the phenomenon that all the items in a large itemset do not always occur together. This corruption level for an itemset is fixed and is obtained from a normal distribution, with mean being 0.5 and variance 0.1. Detailed parameter settings in the experiments are given in Table 2.

5.2. Experiments with synthetic data

We study the scalability of the proposed algorithm using synthetic generated data. All the experiments were conducted on a Sun Ultra Sparc Workstation with a CPU clock rate of 270 MHz and 64 MB main memory. We measure the execution performance based on wall-clock time, i.e., the total elapsing time including CPU and I/O time.

Fig. 4(a) shows how the algorithm behaves as the number of items in a database increases from 800 to 1200. It is interesting to note that, when the number of items increases, the execution time of the algorithm decreases. This is due to the fact that with more items, each transaction under a certain average and maximum transaction size condition is more likely to have different items assigned, resulting in less number of large 1-itemsets that meet the *minsup* threshold. The smaller $|L_1|$ leads directly to less candidate itemsets, especially



(a) Number of items scale-up

(b) Number of transactions scale-up

(c) Average transaction size scale-up

Fig. 4. Results of scale-up experiments (*maxscope* = 2).

candidate 2-itemsets; being generated further, thus less time for counting them. For instance, when the database has 800 items, from 142 large 1-itemsets, 9759 candidate 2-itemsets are generated (at *minsup* = 3.0%). When the item number increases to 1200, we are left with only 77 large 1-itemsets and 2849 candidate 2-itemsets, cutting down about 46% and 70%, respectively.

Next, we examine how the algorithm scales up with the number of transactions in the database. We increase the number of transactions from 20K to 100K. The result in Fig. 4(b) coincides with our expectation: the execution time of the algorithm increases when more transactions in a database need to be scanned and checked. As shown in Fig. 4(b), the mining time scales linearly.

We further investigate the scale-up as we increase the average transaction size from 3 to 7. From the result presented in Fig. 4(c), the more items per transaction, the more time needed to process. The reason is obvious: given a minimum support and a set of items, when the average transaction size is large, there is more $|L_1|$ generated, hence more $|C_2|$ needs to be counted. Also, the time needed to scan every transaction in the database becomes longer, resulting in higher processing costs. For example, at average transaction size 5, the execution time is around 144 s (at *minsup* = 3.0%), but at average transaction size 7, it increases dramatically to 515 s.

Our last experiment with synthetic datasets is to study the effect of maximal scopes *maxscope* on the mining performance. In Fig. 5, when the maximum scope is 0, mining generalized inter-transactional association rules degrades to mining traditional intra-transactional association rules. When the maximum scopes are enlarged, much more candidates sitting at different transactions such as $\{a_{[0,0]}, a_{[1,3]}\}, \{a_{[0,2]}, b_{[3,3]}\}$ are added into C_2. The algorithm thus has to spend more time to scan database records and count candidate C_k. For

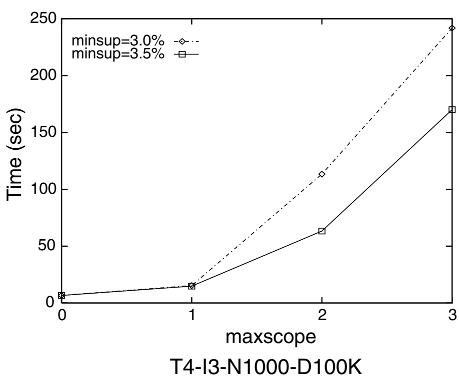


Fig. 5. Maximum scope versus execution time.

instance, at *minsup* = 3.5%, from maximal scope 0–2, the mining time increases from 6.6 to 63.27 s, about 9.6 times longer.

5.3. Application to real-life data

To investigate potential applications of generalized inter-transactional association rules, we performed two sets of experiments with meteorological data obtained from the Hong Kong Observatory headquarters, which takes meteorological observations, including *wind direction*, *wind speed*, *dry bulb temperature*, *relative humidity*, *rainfall* and *mean sea level pressure*, etc., every 6 hours each day.

Our first test is to detect generalized inter-transactional association rules from the 1996 meteorological data, and use the 1997 meteorological data from the same area in Hong Kong to examine their predictive capability, measured by *Pred-Rate* $(X \Rightarrow Y) = sup(X \cup Y)/sup(X)$, i.e., the ratio of extended transaction sets containing $X \cup Y$ to those containing only X.

The mining context in this test is 1-dimension with *time* as its dimensional attribute. Considering seasonal changes of weather, we extract records from the first day of May to the last day of October, and there are totally 736 records (total-days $*$ 4 = (31 + 30 + 31 + 31 + 30 + 31) $*$ 4 = 736) for each year. These raw data sets, containing continuous atmospheric elements, are further converted into appropriate formats with which the algorithm can work. (1) *Wind direction* values measured in continuous degrees are discretized into eight wind directions—*north-east*, *east*, *south-east*, *south*, *south-west*, *west*, *north-west* and *north*; (2) *wind speed* values are classified as *light, moderate, fresh,* or *strong*; (3) *rainfall* recorded in the unit of centimeter is discretized into *no-rain*, *trace*, *light*, *moderate*, or *heavy*; (4) *relative humidity* is characterized into *very-dry*, *dry*, *medium-wet* or *wet*; (5) *temperature* is represented by *very-cold*, *cold*, *mild*, *warm* or *hot*; (6) *mean sea level pressure* values are discretized into *very-low*, *low*, *moderate*, *slightly-high*, *high*, or *very-high*. After transformation, we obtain 32 kinds of items in total, and each database record contains six different items. The interval of every two consecutive records is 6 h.

By setting *maxscope* as 3, 7 and 11, we can detect associated meteorological relationships happening within one day ((3 + 1)/4 = 1), two days ((7 + 1)/4 = 2) and three days ((11 + 1)/4 = 3). Some generalized inter-transactional association rule examples found from the data set under *minsup* = 90% and *minconf* = 99% are as follows:

- "If there is *no rain* within 6 hours and the weather is *medium wet* during the following 24 hours, then there will be *no rain* for 2 days" ($13_{[0,1]}, 20_{[2,5]} \Rightarrow 13_{[2,7]}$, *Pred-Rate* = 96%).
- "If it is *warm* within 2 days, then within 3 days there will be *no rain*" ($25_{[0,7]} \Rightarrow 13_{[0,11]}$, *Pred-Rate* = 76%).

- “If the wind speed continues to be *moderate* for 2 days, then there will be *no rain* during the third day” ($10_{[0,7]} \Rightarrow 13_{[7,11]}$, *Pred-Rate* = 83%).

Another interesting observation made from this test is that, when we keep *minsup* and *minconf* unchanged and enlarge the mining context *maxscope*, the predictive rate of discovered rules on average decreases. As shown in Fig. 6, *Avg-Pred-Rate* = 94% at *maxscope* = 3 (1 day), *Avg-Pred-Rate* = 91% at *maxscope* = 7 (2 days), and *Avg-Pred-Rate* = 83% at *maxscope* = 11 (3 days). This is because with a large *maxscope*, the contextual constraints on the co-occurrence of association relationships are less strict than under a small *maxscope*. Hence, more *large* itemsets and generalized inter-transactional association rules are returned, and some of them have poor predictive rates when applied to the test data.

As a reference, in the second test with the meteorological data, we compare our generalized inter-transactional association mining with the other two association mining methods, i.e., *traditional intra-transactional association mining* [2,5] and *point-wise inter-transactional association mining* [36,37]. Table 3 summarizes the mining results. Under *minsup* = 60% and *minconf* = 90%, we found only one intra-transactional association rule and four point-wise inter-transactional association rules, but 113 generalized inter-transactional association rules (which reduce to 27 under *minconf* = 99%). This is expected as much more candidate and large itemsets are generated when mining generalized inter-transactional association mining. Note that such an increase from intra-transactional to generalized inter-transactional association rules is much greater than that from intra-transactional to point-wise inter-transactional association rules, due to the relaxation of contextual scopes in detecting association relationships. Inevitably, mining generalized inter-transactional association rules demands much more time in scanning the database and counting candidates than mining the other two association rules, as illustrated in Fig. 7.

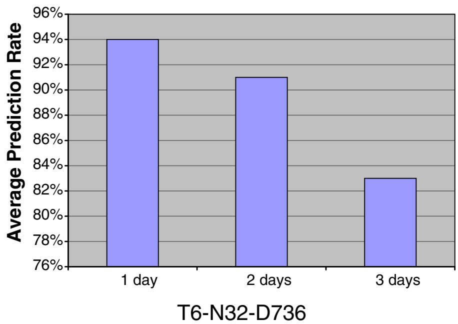


Fig. 6. Maxscope versus average predictive rate.

Table 3
Mining result comparison (T6-N32-D736, *maxscope* = 3)

	#Candidate-itemset	#Large-itemset	*minconf* = 90%		*minconf* = 99%	
			#Rule	Avg-Pred-Rate	#Rule	Avg-Pred-Rate
minsup = 60%						
Classic intra-trans. AR	33	3	1	94%	0	– –
Point-wise inter-trans. AR	332	16	4	89%	0	– –
Generalized inter-trans. AR	801	252	113	83%	27	92%
minsup = 80%						
Classic intra-trans. AR	32	1	0	–	0	– –
Point-wise inter-trans. AR	323	5	1	89%	0	– –
Generalized inter-trans. AR	379	52	27	84%	5	91%

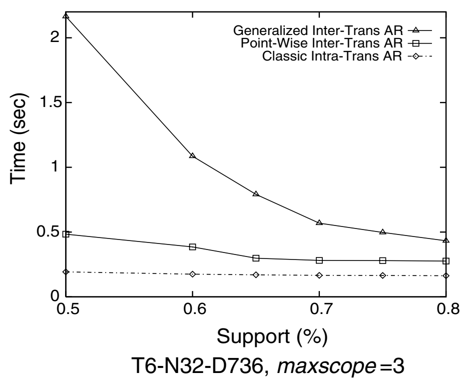


Fig. 7. Minimum support versus execution time.

6. Extension to generalized multidimensional inter-transactional association rule mining

The method of mining generalized 1-dimensional inter-transactional association rules described in Section 4 can be further extended to a multidimensional context, which involves multiple dimensional attributes.

Fig. 8 shows a 3-dimensional space, where each point is located by a 3-dimensional coordinate $n = (n.x, n.y, n.z)$ on the axes X, Y and Z. A generalized 3-dimensional inter-transactional association rules is like "$a_{[(0,0,0),(1,1,0)]} \Rightarrow$

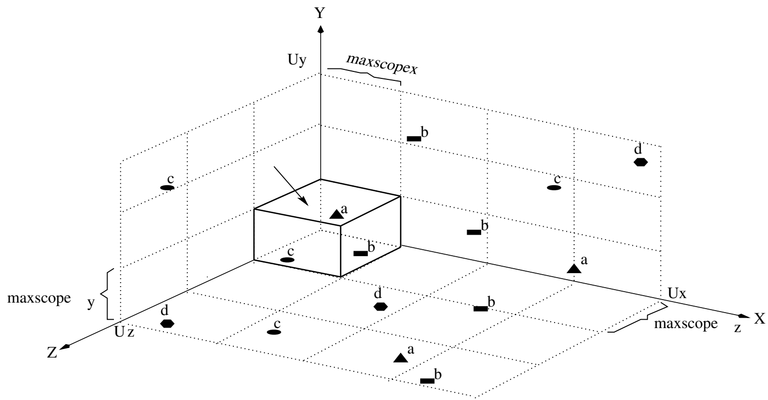


Fig. 8. A 3-dimensional mining context.

$b_{[(2,1,1),(4,1,1)]}$", stating that if *a* happens within the scope of [(0,0,0),(1,1,0)], then *b* will happen within [(2,1,1),(4,1,1)]. Here, every interval unit along the three dimensions can be assigned a specific meaning depending on real applications. In this example, suppose *X* dimension represents *area* measured in mile, *Y* dimension represents *undersea depth* measured in inch, *Z* dimension represents *temperature* in centigrade degree, and the database items denote *plants*. The above rule example expresses that "if plant 'a' grows within the area of 1 mile and undersea between 0 and 1 in., then plant 'b' will grow in the area which is 2–4 miles away, undersea 1 in. with a temperature 1 degree higher."

To discover such a generalized 3-dimensional inter-transactional association rule, same as 1-dimensional association mining, we can let users specify three values $maxscope_x$, $maxscope_y$, $maxscope_z$ as the maximal scopes along dimensions *X*, *Y* and *Z*, so as to limit the search space for interesting association rules. These three values constitute a 3-dimensional unit cube (as the arrow indicates in Fig. 8). Only the association relationships between items that are covered within the unit cube will be detected and returned. After that, starting from each point in the space, we slide this cube along each of the three dimensions to count all candidate large itemsets. If an itemset is contained by some transactions whose occurrence contextual points are inside the cube, its support will be increased by one. For example, to count itemset $\{a_{[(0,0,0),(1,1,0)]}, b_{[(2,1,1),(4,1,1)]}\}$, with (0,0,0) as the reference point of the cube, we need to check whether *a* and *b* are contained by the transactions located in the scopes of [(0,0,0),(1,1,0)] and [(2,1,1),(4,1,1)], respectively. Assume that $inclusive([(0,0,0),(maxscope_x, maxscope_y, maxscope_z)], [(0,0,0),(1,1,0)]) = true$ and $inclusive([(0,0,0),(maxscope_x, maxscope_y, maxscope_z)], [(2,1,1),(4,1,1)]) = true$. In other words, we need to scan every transaction at the point $n \in [(0,0,0),(1,1,0)]$ to examine whether it contains item *a*, and also each transaction at the point $n' \in [(2,1,1),(4,1,1)]$ to examine whether it contains item b. If we can find such a pair of transactions, then the count of $\{a_{[(0,0,0),(1,1,0)]}, b_{[(2,1,1),(4,1,1)]}\}$ is added by 1. Moving the cube one step further along X dimension to take $(1,0,0)$ as the reference point, we then check whether a and b are contained by transactions within $[(0+1,0,0),(1+1,1,0)]$ and $[(2+1,1,1),(4+1,1,1)]$, respectively. In the same way, we can slide the cube along Y and Z dimensions, each time taking different contextual points as the reference point, from which to count the itemset. In total, all $(U_x+1)*(U_y+1)*(U_z+1)$ points in the 3-dimensional space can be traversed as the reference point of the cube.

In general, we can view such a cube-sliding trace in the 3-dimensional space in a flattened 1-dimensional way. Recall that while mining generalized 1-dimensional inter-transactional association rules, the algorithm described in Section 4 transforms the original 1-dimensional database into a new one through the function *Transform-Record-Transet*, so that each record of the new database registers all items happening within a range (i.e., *maxscope*) of transactions instead of one transaction. In a similar way, here, we can transform the 3-dimensional database in Fig. 8 into an 1-dimensional database by merging items happening within a unit cube of transactions into one record. For instance, starting from a reference point, say (x_0,y_0,z_0) where $0 \leqslant x_0 \leqslant U_x$, $0 \leqslant y_0 \leqslant U_y$, and $0 \leqslant z_0 \leqslant U_z$, the unit cube $[(x_0,y_0,z_0),(maxscope_x+x_0,maxscope_y+x_0,maxscope_z+x_0)]$ covers transactions located at (x,y,z), which can be $(x_0,y_0,z_0),(x_0+1,y_0,z_0),\ldots,(x_0+maxscope_x,y_0,z_0),\ldots,$ or $(x_0+maxscope_x,y_0+maxscope_y,z_0+maxscope_z)$, i.e., $(x,y,z)\in[(x_0,y_0,z_0),(maxscope_x+x_0,maxscope_y+x_0,maxscope_z+x_0)]$. All the items contained by these transactions are put together into a record of the same format as in mining 1-dimensional inter-transactional association rules, i.e.,

$$(ID_{(x_0,y_0,z_0)}, item_1, occurNum_1, posi_1, \ldots, posi_{occurNum_1},$$
$$\ldots,$$
$$item_r, occurNum_r, posi_1, \ldots, posi_{occurNum_r}),$$

where $ID_{(x_0,y_0,z_0)}$ is the transformed new record identifier, $item_1,\ldots,item_r$ are the item identifiers in the unit cube each of which is followed by $occurNum_1,\ldots,occurNum_r$—the numbers of transactions the corresponding items appear in, and the normalized positions of these transactions within $[(0,0,0),(maxscope_x, maxscope_y, maxscope_z)]$ after normalization.

After the transformation of the original 3-dimensional database, the mining algorithm described in Section 4 can be directly applied to search and detect large widely-extended itemsets spanning within the scope of a unit cube from a 1-dimensional view. The discovered large itemsets can then be post-processed to give generalized 3-dimensional inter-transactional association rules. Along this line, the mining of generalized m-dimensional inter-transactional association rules (where $m > 3$) can be done in the same way as mining generalized 3-dimensional inter-transactional association rules. We leave the performance evaluation of m-dimensional inter-transactional association rule mining to a further study.

7. Related work

The problem of mining generalized multidimensional inter-transactional association rules looks similar to the problems of sequential pattern mining and time series analysis when there is only one dimensional attribute (e.g., time, space, etc.) involved. In this section, we compare such extended association rule mining with some closely related work conducted by the data mining, database and statistics communities.

7.1. Association rule mining

The traditional association rule mining proposed by Agrawal et al. [2] and point-wise inter-transactional association rule mining defined in [36,37] can be viewed as special cases of the generalized multidimensional inter-transactional association rule mining due to the following two points: (1) If we omit the *dimensional attributes* and *occurrence contexts* of database transactions, and set contextual scopes of all items to $[0, 0]$, the generalized inter-transactional association rule mining will degrade to the traditional intra-transactional association rule mining, i.e., looking for associated itemsets within the same transaction. For example, a rule "$a_{[0,0]}, b_{[0,0]} \Rightarrow c_{[0,0]}$" carries the same meaning as a traditional counterpart "$a, b \Rightarrow c$". (2) If we restrict contextual scopes of items to the format $[x, x]$, generalized inter-transactional association rules become rigid point-wise inter-transactional association rules. For instance, "$a_{[0,0]}, b_{[1,1]} \Rightarrow c_{[3,3]}$", which is equivalent to "$a(0), b(1) \Rightarrow c(3)$", is a point-wise inter-transactional association rule example.

7.2. Quantitative range-based association rule mining

Quantitative association rule mining intends to find associations among numeric and categorical attributes [54]. A quantitative rule example is like "10% of married people between age 50 and 60 have at least 2 cars". Wang et al. presented an interestingness-based interval merger method for fast discovery of numeric association rules [61]. Utilizing numeric association rules, Fukuda et al. described an approach of efficient construction of decision trees [23]. Liu et al. further proposed to integrate classification and association mining, and developed a classification-based association rule mining technique [35].

Each item in this work is of the form (*attribute*, *integer-value*), where an attribute can be either categorical or continuous.

At a first glance, the generalized multidimensional inter-transactional association rules resemble range-based quantitative association rules, if we view each dimensional attribute as a normal item. However, there are fundamental differences between them. First, each item in the former association rules are examined and thus attached with point/range-wise contextual information. This is not the case for the items in quantitative association rules. Second, the former inter-transactional association mining implies ordering of transactions. Its point/range-based dimensional attribute values denote *relative* rather than *absolute* contextual relationships of items. In comparison, each quantitative attribute in the latter must be of an absolute value.

7.3. Sequential pattern discovery

From transaction databases where each record contains items bought by a particular customer, Agrawal et al. coined the problem of mining sequential patterns in different transactions during a period of time [6]. One sequential pattern example is "80% of customers bought shoes *after* they bought shirts." For mining sequential patterns, transactions of each customer ordered by transaction-time are organized into one record. The problem of sequential pattern mining was further generalized to allow items to be present in a set of transactions whose transaction-times are within a user-specified time window [54]. Despite this, sequential pattern mining focuses on *successive/precedent* relationships of items. On the other hand, users may be interested in finding all associations across a set of transactions within *different* ranges. This part of contextual information can explicitly be captured within our generalized inter-transactional association framework.

7.4. Episode rule discovery

The problem of discovering frequent episodes from sequential events was introduced by Mannila et al. in [38,39]. An episode is a collection of events that occur relatively close to each other in a certain partial order, whose total span of time is constrained by a window. An episode rule has the format of $P[V] \Rightarrow Q[W]$ where P and Q are episodes; and V and W are real numbers representing time intervals, stating that if episode P has a minimal occurrence at interval $[t, t']$ with $t' - t \leqslant V$, then episode Q occurs at interval $[t, t'']$ for some t'' such that $t'' - t \leqslant W$. As time intervals in an episode rule are constrained to have the *same starting time* t, the order of events in an episode can only be roughly specified. Comparatively, the generalized inter-transactional association framework is more general and embrace the expressive power of episode rules. Another difference is that our mining algorithm aims to find, with reasonable amount of time, *all* association rules within different spans, regardless of the ordering of events. As mentioned by the authors, only certain types of episodes are easily detected using their mining algorithms. The efficient mining of more general episode rules with arbitrary time bounds from a large sequence remains an open problem.

7.5. Temporal relationship mining in time sequence

Compared to episode sequences, Bettini et al. looked for more complex event sequences from time sequential data [8,9]. Unlike episodes where only the order, but not the concise quantitative relationships among events can be expressed, Bettini's model allows temporal relationship among events to be quantitatively defined, even using different granularities. Their work differs from our generalized inter-transactional association rule mining in the following two aspects. First, they only considered the mining task where an event structure is given, and only some of its event variables, including the starting event variable, are instantiated. Therefore, the mining process can only discover possible event instances that match the given structure based on the frequence on which the corresponding events occur in the event sequence. No algorithms are given to discover all event structures with frequence that exceeds a threshold. Second, their work focused on event sequences. It is obvious that rules above a certain confidence threshold can show the connections between events more clearly than event sequences alone [39]. However, neither definitions nor mining algorithms regarding the rules were discussed in their context.

7.6. Rule discovery from time-series

To facilitate rule discovery from time-series, Das et al. presented an adaptive method to transform low-level signal data into a more abstract symbolic representation, from which rule induction can be performed using existing methods (such as episode rule discovery methods) [17]. As this work focuses more on data preparation, the most complex rules they studied have a rather simple format $A_1 \wedge \cdots \wedge A_h[V] \Rightarrow B[T]$, stating that *if* A_1 *and ... and* A_h *occur within* V *units of time*, *then* B *occurs within time* T. Furthermore, in contrast to our paper, no attention has been given to the issues of efficient and scalable mining.

7.7. Time-series analysis

Time-series analysis and forecasting has been an active research topic in statistics. The main purpose is to understand and model the stochastic mechanism that gives rise to an observed series, or to forecast future values of a series based on the history of that series [15,62]. DeCoste proposed a technique based on linear regression and neural network for automated detection of anomalies in sensor data [18]. Recently, Yi et al. presented a fast method called MUSCLES to analyze co-evolving time sequences to enable estimation of missing/delayed/future values and outlier detection [64]. The main theme of the analysis performed in this area is different from mining rules from a large amount of data under multidimensional contexts.

7.8. Similarity retrieval from sequences

Most of sequence-related work in the database community concerns similarity search and querying, i.e., finding similar sequences that match a given pattern in some error distances, or searching all pairs of similar sequences [3,16,33,59]. Various approaches have been suggested including using the discrete Fourier transform, interpolation approximation, or defining some shape querying languages [32,63]. Issues such as how to detect patterns efficiently from a huge database of sequences are not the focus in this body of work.

As a summary, the generalized multidimensional inter-transactional association rule mining presented in this paper provides a *unified* framework, under which a number of association and pattern related mining problems can be viewed and hence treated uniformly.

8. Conclusion

We have generalized the problem of mining association rules by incorporating multidimensional contexts into the mining framework. An extended and flexible form of association rules named *generalized multidimensional inter-transactional association rules* is presented. We described an algorithm for mining such extended association rules under 1-dimensional contexts by extension of a priori. Empirical evaluation of the algorithm on both real-life and synthetic data sets shows that, with such generalized association rules, we can detect more comprehensive and interesting association relationships. Further extension of the algorithm to a multidimensional context is also discussed.

Mining generalized inter-transactional association rules poses more challenges on efficient processing than mining classical intra-transactional and rigid point-wise inter-transactional association rules. Among the many possible future research issues, one immediate task is to provide a framework which enables users to declare what kinds of generalized inter-transactional associations are of interest, so that the mining can be focused and become more efficient. In addition, efficient discovery of generalized inter-transactional association rules in a distributed and parallel environment is also an interesting area of work we plan to explore.

References

[1] Y.-J. Tsay, Y.-W. Chang-Chien, An efficient cluster and decomposition algorithm for mining association rules, Information Sciences 160 (1–4) (2004) 161–171.
[2] R. Agrawal, T. Imielinski, A. Swami, Mining association rules between sets of items in large databases, in: Proc. of the ACM SIGMOD Intl. Conf. on Management of Data, Washington, DC, USA, May 1993, pp. 207–216.
[3] R. Agrawal, K.I. Lin, H.S. Sawhney, K. Shim, Fast similarity search in the presence of noise, scaling, and translation in time-series databases, in: Proc. of the 21st Intl. Conf. on Very Large Data Bases, Zurich, Switzerland, September 1995, pp. 490–501.
[4] R. Agrawal, J.C. Shafer, Parallel mining of association rules, IEEE Transactions on Knowledge and Data Engineering 8 (6) (1996) 962–969.
[5] R. Agrawal, R. Srikant, Fast algorithms for mining association rules, in: Proc. of the 20th Intl. Conf. on Very Large Data Bases, Santiago, Chile, September 1994, pp. 478–499.
[6] R. Agrawal, R. Srikant, Mining sequential patterns, in: Proc. of the Intl. Conf. on Data Engineering, Taipei, Taiwan, March 1995, pp. 3–14.
[7] E. Baralis, G. Psaila, Designing templates for mining association rules, Journal of Intelligent Information Systems 9 (1) (1997) 7–32.
[8] C. Bettini, X. Wang, S. Jajodia, Testing complex temporal relationships involving multiple granularities and its applications to data mining, in: Proc. of the Fifteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Montreal, Canada, June 1996, pp. 68–78.
[9] C. Bettini, X. Wang, S. Jajodia, Mining temporal relationships with multiple granularities in time sequences, Data Engineering 21 (1) (1998) 32–38.
[10] S. Brin, R. Motwani, C. Silverstein, Beyond market basket: generalizing association rules to correlations, in: Proc. of the ACM SIGMOD Intl. Conf. on Management of Data, Tucson, Arizona, USA, June 1997, pp. 265–276.
[11] S. Chakrabarti, S. Sarawagi, B. Dom, Mining surprising patterns using temporal description length, in: Proc. of the 24th Intl. Conf. on Very Large Data Bases, New York, USA, August 1998, pp. 606–617.
[12] D. Cheung, J. Han, V. Ng, C.Y. Wong, Maintenance of discovered association rules in large databases: an incremental updating technique, in: Proc. of the Intl. Conf. on Data Engineering, New Orleans, LA, USA, February 1996, pp. 106–114.
[13] D.W. Cheung, V.T. Ng, A.W. Fu, Y.J. Fu, Efficient mining of association rules in distributed databases, IEEE Transactions on Knowledge and Data Engineering 8 (6) (1996) 911–922.
[14] E. Cohen, M. Datar, S. Fujiwara, A. Gionis, P. Indyk, R. Motwani, J.D. Ullman, C. Yang, Finding interesting associations without support pruning, in: Proc. Intl. Conf. Data Engineering, California, USA, March 2000, pp. 489–499.
[15] J.D. Cryer, Time Series Analysis, Duxbury Press, 1985.
[16] G. Das, D. Gunopulos, H. Mannila, Finding similar time series, in: Proc. of the Intl. Conf. on Principles of Data Mining and Knowledge Discovery, Norway, June 1997, pp. 88–100.
[17] G. Das, K.-I. Lin, H. Mannila, G. Renganathan, P. Smyth, Rule discovery from time series, in: Proc. of the Second Intl. Conf. on Knowledge Discovery and Data Mining, New York, USA, August 1998, pp. 16–22.
[18] D. DeCoste, Mining multivariate time-series sensor data to discover behavior envelopes, in: Proc. of the 3rd Intl. Conf. on Knowledge Discovery and Data Mining, Newport Beach, CA, August 1997, pp. 151–154.
[19] M. Fang, N. Shivakumar, H. Garcia-Molina, R. Motwani, J.D. Ullman, Computing iceberg queries efficiently, in: Proc. of the 24th Intl. Conf. on Very Large Data Bases, New York, USA, August 1998, pp. 299–310.

[20] L. Feng, T. Dillon, J. Liu, Inter-transactional association rules for multi-dimensional contexts for prediction and their application to studying meteorological data, Data and Knowledge Engineering 37 (2001) 85–115.

[21] L. Feng, H. Lu, J. Yu, J. Han, Mining inter-transaction association rules with templates, in: Proc. ACM CIKM Intl. Conf. Information and Knowledge Management, November 1999, pp. 225–233.

[22] L. Feng, H. Lu, J. Yu, J. Han, A template model for multidimensional inter-transactional association rules, VLDB journal 11 (2) (2002) 153–175.

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

[24] T. Fukuda, Y. Morimoto, S. Morishita, T. Tokuyama, Mining optimized association rules for numeric attributes, in: Proc. of the 15th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Montreal, Canada, June 1996, pp. 182–191.

[25] E.-H. Han, G. Karypis, V. Kumar, Scalable parallel data mining for association rules, in: Proc. of the ACM SIGMOD Intl. Conf. on Management of Data, Tucson, AZ, USA, June 1997, pp. 277–288.

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

[27] J. Han, Y. Fu, Meta-rule-guided mining of association rules in relational databases, in: Proc. of the 1st Intl. Workshop on Integration of Knowledge Discovery with Deductive and Object-Oriented Databases, Singapore, December 1995, pp. 39–46.

[28] J. Han, J. Pei, Y. Yin, Mining frequent patterns without candidate generation, in: Proc. of the ACM SIGMOD Intl. Conf. on Management of Data, USA, May 2000, pp. 1–12.

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

[30] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, A.I. Verkamo, Finding interesting rules from large sets of discovered association rules, in: Proc. of the 3rd Intl. Conf. on Information and Knowledge Management, Gaithersburg, MD, November 1994, pp. 401–408.

[31] B. Lent, A. Swami, J. Widom, Clustering association rules, in: Proc. of the Intl. Conf. on Data Engineering, Birmingham, England, April 1997, pp. 220–231.

[32] C.S. Li, P.S. Yu, V. Castelli, HierachyScan: a hierachical similarity search algorithm for databases of long sequences, in: Proc. of the Intl. Conf. on Data Engineering, New Orleans, LA, USA, February 1996, pp. 546–553.

[33] L. Lin, T. Risch, Querying continuous time sequences, in: Proc. of the 24th Intl. Conf. on Very Large Data Bases, New York, USA, August 1998, pp. 170–181.

[34] B. Liu, W. Hsu, Y. Ma, Mining association rules with multiple minimum supports, in: Proc. ACM SIGKDD Intl. Conf. Knowledge Discovery and Dara Mining, California, USA, August 1999, pp. 125–134.

[35] B. Liu, W. Hsu, Y.M. Ma, Integrating classification and association rule mining, in: Proc. of the Intl. Conf. on Knowledge Discovery and Data Mining, New York, USA, August 1998, pp. 80–86.

[36] H. Lu, L. Feng, J. Han, Beyond intra-transactional association analysis: mining multi-dimensional inter-transaction association rules, ACM Transactions on Information Systems 18 (4) (2000) 423–454.

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

[38] H. Mannila, H. Toivonen, Discovering generalized episodes using minimal occurrences, in: Proc. of the Second Intl. Conf. on Knowledge Discovery and Data Mining, Portland, Oregon, August 1996, pp. 146–151.
[39] H. Mannila, H. Toivonen, A. Verkamo, Discovering frequent episodes in event sequences, Data Mining and Knowledge Discovery 3 (1) (1997) 259–289.
[40] R. Meo, G. Psaila, S. Ceri. A new SQL-like operator for mining association rules, in: Proc. of the 22th Intl. Conf. on Very Large Data Bases, Mumbai, India, September 1996, pp. 122–133.
[41] R.J. Miller, Y. Yang, Association rules over interval data, in: Proc. of the ACM SIGMOD Intl. Conf. on Management of Data, Tucson, AZ, USA, June 1997, pp. 452–461.
[42] 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 Intl. Conf. on Management of Data, Seattle, Washington, June 1998, pp. 13–24.
[43] B. Özden, A. Ramaswamy, A. Silberschatz, Cyclic association rules, in: Proc. of the Intl. Conf. on Data Engineering, Florida, USA, February 1998, pp. 412–421.
[44] J.-S. Park, M.-S. Chen, P.S. Yu, An effective hash based algorithm for mining association rules, in: Proc. of the ACM SIGMOD Intl. Conf. on Management of Data, San Jose, CA, May 1995, pp. 175–186.
[45] J.-S. Park, M.-S. Chen, P.S. Yu, Mining association rules with adjustable accuracy, Technical Report IBM Research Report, 1995.
[46] J.-S. Park, M.-S. Chen, P.S. Yu, Data mining for path traversal patterns in a web environment, in: Proc. of the 16th Conf. on Distributed Computing Systems, Hong Kong, May 1996, pp. 385–392.
[47] J. Pei, J. Han, R. Mao, CLOSET: an efficient algorithm for mining frequent closed itemsets, in: Proc. of the ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, USA, May 2000, pp. 21–30.
[48] S. Ramaswamy, S. Mahajan, A. Silberschatz, On the discovery of interesting patterns in association rules, in: Proc. of the 24th Intl. Conf. on Very Large Data Bases, New York, USA, August 1998, pp. 368–379.
[49] R. Rastogi, K. Shim, Mining optimized association rules with categorical and numerical attributes, in: Proc. of the Intl. Conf. on Data Engineering, Florida, USA, February 1998, pp. 503–512.
[50] A. Savasere, E. Omiecinski, S. Navathe. An efficient algorithm for mining association rules in large databases, in: Proc. of the 21st Intl. Conf. on Very Large Data Bases, Zurich, Switzerland, September 1995, pp. 432–443.
[51] C. Silverstein, S. Brin, R. Motwani, J.D. Ullman, Scalable techniques for mining causal structures, in: Proc. of the 24th Intl. Conf. on Very Large Data Bases, New York, USA, August 1998, pp. 594–605.
[52] C. Silverstein, S. Brin, R. Motwani, J.D. Ullman, Scalable techniques for mining causal structures, Data Mining and Knowledge Discovery 4 (2/3) (2000) 163–192.
[53] R. Srikant, R. Agrawal, Mining generalized association rules, in: Proc. of the 21st Intl. Conf. on Very Large Data Bases, Zurich, Switzerland, September 1995, pp. 409–419.
[54] R. Srikant, R. Agrawal, Mining quantitative association rules in large relational tables, in: Proc. of the ACM SIGMOD Intl. Conf. on Management of Data, Montreal, Canada, June 1996, pp. 1–12.
[55] R. Srikant, Q. Vu, R. Agrawal, Mining association rules with item constraints, in: Proc. of the 3rd Intl. Conf. on Knowledge Discovery and Data Mining, Newport Beach, CA, August 1997, pp. 67–73.
[56] H. Toivonen, Sampling large databases for association rules, in: Proc. of the 22th Conf. on Very Large Data Bases, Mumbai, India, September 1996, pp. 134–145.

[57] D. Tsur, J.D. Ullman, S. Abitboul, C. Clifton, R. Motwani, S. Nestorov, Query flocks: a generalization of association-rule mining, in: Proc. of the ACM SIGMOD Intl. Conf. on Management of Data, Seattle, Washington, June 1998, pp. 1–12.
[58] K.H. Tung, H. Lu, J. Han, L. Feng, Breaking the barrier of transactions: mining inter-transaction association rules, in: Proc. ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining, August 1999, pp. 297–301.
[59] J.T.-L. Wang, G.-W. Chirn, T.G. Marr, B. Shapiro, D. Shasha, K. Zhang, Combinatorial pattern discovery for scientific data: some preliminary results, in: Proc. of the ACM SIGMOD Intl. Conf. on Management of Data, Minnesota, USA, May 1994, pp. 115–125.
[60] K. Wang, Y. He, J. Han, Mining frequent itemsets using support constraints, in: Proc. 26th Intl. Conf. Very Large Data Bases, Cairo, Egypt, September 2000, pp. 43–52.
[61] K. Wang, S.H.W. Tay, B. Liu, Interestingness-based interval merger for numeric association rules, in: Proc. of the Intl. Conf. on Knowledge Discovery and Data Mining, New York, USA, August 1998, pp. 121–127.
[62] A. Weigend, N. Gerschenfeld, Time Series Prediction: Forecasting the Future and Understanding the Past, Addison Wesley, 1994.
[63] B.K. Yi, H.V. Jagadish, C. Faloutsos, Efficient retrieval of similar time sequences under time warping, in: Proc. of the 16th Intl. Conf. on Data Engineering, Florida, USA, February 1998, pp. 201–208.
[64] B.K. Yi, N.D. Sidiropoulos, T. Johnson, H.V. Jagadish, C. Faloutsos, A. Biliris, Online data mining for co-evolving time sequences, in: Proc. of the 16th Intl. Conf. on Data Engineering, California, USA, February 2000, pp. 13–22.
[65] M.J. Zaki, S. Parthasarathy, M. Ogihara, W. Li, New algorithms for fast discovery of association rules, in: Proc. of the 3rd Intl. Conf. on Knowledge Discovery and Data Mining, Newport Beach, CA, USA, August 1997, pp. 283–286.