Frequent Spatiotemporal Association Patterns Mining Based on Granular Computing

Gang Fang[1, 2] and Yue Wu[1]
[1] School of Computer Science & Engineering
University of Electronic Science and Technology of China, Chengdu, Sichuan, 611731, P. R. China
[2] School of Computer Science & Engineering
Chongqing Three Gorges University, Wanzhou, Chongqing, 404000, P. R. China
E-mail: gangfang07@sohu.com, ywu@uestc.edu.cn

Keywords: spatiotemporal association patterns, star association model, granular computing, mixed radix notation system, spatiotemporal data mining

Received: May 17, 2013

In order to discover multi-dimensional spatiotemporal association patterns, and improve the efficiency of traditional mining algorithms for spatiotemporal association patterns, this paper firstly constructs a star association model based on event, which can show more spatiotemporal information on the basis of the present star association model. Besides traditional attributes association with point, line and plane, the model also can fully and flexibly express temporal association, orientation association, and topology association, namely, it can quickly and simply form multi-dimensional spatiotemporal association patterns. And then for the star association model based on event, an algorithm of discovering frequent spatiotemporal association patterns based on granular computing is proposed, which is different from traditional association patterns mining algorithms. One is that the algorithm breaks traditional thinking of generating candidate frequent itemsets, namely, it generates candidate frequent itemsets by updating the mixed radix numeral. The method is quick and simple to avoid redundant complicated calculations for adopting complex FP-tree data structure or generating candidate by joining frequent itemsets. The other is that the algorithm for discovering frequent spatiotemporal association patterns only needs to read database once via granular computing, in other words, it discovers each frequent spatiotemporal association pattern via constructing a spatiotemporal information granule, where the intension can be mapped to the mixed radix numeral from the mixed radix notation system based on spatiotemporal information system. Finally, this paper further discusses the characteristics and the optimal application environments of the algorithm. Experimental results indicate that the algorithm is simpler and faster than these traditional frequent patterns mining algorithms on the optimal application environments.

Povzetek: Opisana je nova metoda za rudarjenje prostorsko-časovnih vzorcev.

1 Introduction

Discovering spatial association patterns from spatial database is one of important tasks for spatial data mining and knowledge and discovery. Spatial association patterns have been applied to some valuable domains, such as Urban Traffic [1], Bioscience [2], Social Security [3], Climate forecasting [4], and Demographic survey [5]. In recent decade, there is some research work for mining spatial association patterns. Reference [6] focuses on this specificity of spatial data mining by showing the suitability of join indices to this context. It describes the join index structure and shows how it could be used as a tool for spatial data mining; Reference [7] discusses the multiple level association rules mining, and further indicates spatiotemporal association rules mining should address issues of data integration, data classification, the representation and calculation of spatial relationships, and strategies for finding 'interesting' rules; Reference [8] has proposed a generalized framework to effectively discover different types of spatial and spatiotemporal patterns in scientific data sets, which can be used to capture a variety of interactions among objects of interest and the evolutionary behaviour of such interactions. Based on the feature of geographic elements, the research work can be divided into the following two groups.

One group is discovering frequent region association patterns for numeric geographic elements with point, line and plane, i.e. firstly, these numeric attributes are turned into Boolean attributes with geographic elements, and spatial association patterns are discovered by transaction frequent patterns mining methods. The group is suitable for mining spatial association patterns based on spatial location. Reference [7] uses association rules to discover spatiotemporal relationships among a set of variables that characterize socioeconomic and land cover changes, but it only refers to the region. Reference [9] proposes a robust geospatial multivariate association rules mining framework, where the attributes for geographic elements with point, line and plane can be turned into the region. Reference [10] proposes a novel framework to mine

regional association rules based on a given class structure.

The other is discovering frequent spatial association patterns for discrete geographic elements with spatial objects and layout relationships, i.e. firstly, these discrete geographic elements are turned into the category set, and transaction frequent patterns mining methods are used to extract spatial association patterns. Reference [8] and [11] discuss star association patterns, sequence association patterns and clique association patterns based on spatial distance for these spatial objects relationships and layout relationships.

However, these research have the following some shortcoming, firstly, these mining objectives are mainly from spatial database, where these algorithms do not fully regard temporal relationship with spatial association patterns; Secondly, their geographic elements in spatial association patterns are most one-dimensional, namely, the form of spatial association patterns is quite single. Finally, for these traditional frequent patterns mining algorithms, such as Apriori, FP-growth, and their improved algorithms have some disadvantages as follows:

One is the mining framework based on Apriori, i.e. these mining algorithms discover frequent patterns via the thinking of the algorithm Apriori, called the Apriori Framework. The mining framework needs to repeatedly read database for discovering frequent itemsets.

The other is the mining framework based on FP-growth, i.e. these mining algorithms discover frequent patterns via data structure FP-tree, called the FP-growth Framework. The mining framework uses complex data structure to save reading database, but it needs to cost much memory for discovering frequent patterns.

These mining frameworks have some disadvantages for more details seeing references [12-16].

The main contributions in our research work can be summarized as follows:

One is constructing a star association model based on event, which not only expresses traditional attributes association with point, line and plane; the model also can fully flexibly express multi-dimensional spatiotemporal association patterns including the orientation association, the temporal association and the topology association.

The other is proposing an algorithm of discovering frequent spatiotemporal association patterns based on granular computing. For discovering frequent spatio-temporal association patterns, it only needs to read the database once; and then it generates candidate frequent itemsets via updating the mixed radix numeral, where granular computing is introduced to save reading the spatiotemporal database.

The remainder parts are organized as follows:

In Section 2, we introduce the related research work; In Section 3, we construct a star association model based on event; In Section 4, we propose an algorithm of discovering frequent spatiotemporal association patterns based on granular computing; In Section 5, we use some experiments to verify the algorithm, and then discuss its the optimal application environments. In Section 6, we summary research results and discuss future work.

2 Related research work

Based on the notions of granularity [17] and abstraction [18], the ideas of granular computing have been widely investigated in artificial intelligence [19]. In this paper, we adopt a partition model of granular computing to construct information granule [19], which depends on rough set theory [20] and quotient space theory [21]. Here, we introduce the following related definitions.

Definition 2.1 An information table is a quintuple $S = (U, A, \{V_a \mid a \in A\}, L, \{I_a \mid a \in A\})$, where

U, called universe of discourse, is a finite nonempty set for objects;

A, called attributes set, is also a finite nonempty set for attributes;

V_a, called domain set, is a finite set of values for $a \in A$, where V_a is defined as a discrete category set;

L, called descriptive language, a language is defined by attributes in A;

For describing an object of U via the language, it can be denoted as $L = \{\ell \mid V_{a_*^1} \times V_{a_*^2} \times ... \times V_{a_*^{|A^*|}}, a_* \in A^* \subseteq A\}$;

I_a, called information function, is a total function that maps an object of U to exactly one value in V_a, namely $I_a: U \to V_a$.

Definition 2.2 Information granule is a two-tuple $IG = (\xi, \varphi(\xi))$, where

ξ, called the intension of information granule, consists of all attributes that are valid for all those objects to which information granule applies; in other words, the intension is an abstract description of common features or properties shared by elements in the extension, which is expressed as $\xi = (\xi_1, \xi_2, ..., \xi_{|\xi|})$, where

$\xi_k \in V_{a_*^k}$, $a_* \in A^* \subseteq A$, $k = 1, 2, ..., |\xi|$, $\xi \in L$;

$\varphi(\xi)$, called the extension of information granule, is the set of objects which information granule applies, in other words, the extension consists of concrete examples of information granule, which is expressed as follows:
$\varphi(\xi) = \{x \in U \mid I_{a_*^1}(x) = \xi_1, I_{a_*^2}(x) = \xi_2, ..., I_{a_*^{|\xi|}}(x) = \xi_{|\xi|}\}$;

Definition 2.3 Atomic information granule is a two-tuple $AIG = (\xi, \varphi(\xi))$, where

ξ, called the intension of $AIG = (\xi, \varphi(\xi))$, is expressed as $\xi = (\xi_a)$ $(\xi_a \in V_a, a \in A, \xi \in L)$;

$\varphi(\xi)$, called the extension of $AIG = (\xi, \varphi(\xi))$, is expressed as $\varphi(\xi) = \{x \in U \mid I_a(x) = \xi_a\}$.

Definition 2.4 Intersection operation of information granule is denoted by $\otimes$, which is described as follows:

Let two information granules be $IG_\alpha = (\xi_\alpha, \varphi(\xi_\alpha))$ and $IG_\beta = (\xi_\beta, \varphi(\xi_\beta))$, respectively; if $(\exists\, \xi_\alpha^i \in \xi_\alpha \wedge \xi_\alpha^i \in V_a) \wedge (\exists\, \xi_\beta^j \in \xi_\beta \wedge \xi_\beta^j \in V_a)$ then $\xi_\alpha^i = \xi_\beta^j$; and then the intersection operation $\otimes$ can be expressed as follows:
$IG = (\xi, \varphi(\xi)) = IG_\alpha \otimes IG_\beta = (\xi_\alpha \cup \xi_\beta, \varphi(\xi_\alpha) \cap \varphi(\xi_\beta))$.

Definition 2.5 Star association model is expressed as $M =< e_c, \{e_1, e_2,..., e_m\}, \prec >$, where

e_c, called the core element of star association model, is a sole core element;

$e_i (i = 1, 2,..., m)$, called the non-core element of star association model, at least there is a kind of association between each non-core element and the core element;

$\prec$, called time series relationship of star association model, this model only has two types of time series, namely, $\{e_c \prec e_1 \wedge e_c \prec e_2 \wedge ... \wedge e_c \prec e_m\}$ or $\{e_1 \prec e_c \wedge e_2 \prec e_c \wedge ... \wedge e_m \prec e_c\}$.

Definition 2.6 Star association pattern is denoted by $P = \{r_1, r_2,..., r_k\}$, where the association between the core element e_c and the non-core element $e_i (i \in [1, 2,..., k])$ can be denoted by $r_i = R < e_c, e_i > (i \in [1, 2,..., k])$, which consists of the temporal association, the orientation association and the topology association. And then, star association patterns mining is defined as discovering frequent star association patterns from spatiotemporal database for the given minimal support.

3 A star association model based on event

In this paper, on the basis of definition 2.5, we propose a star association model based on event. The model is applied to transform spatiotemporal events and discover frequent spatiotemporal association patterns in Section 4.

Definition 3.1 Star association model based on event is denoted as $EM =< e, e_c, A, E_s, F, P >$, where

e, called a spatiotemporal event, is from a spatiotemporal database, which consists of orientation factor, time factor and topology factor, besides these traditional attributes with point, line and plane;

e_c, called the core element of star association model based on event, is a subject object in the event;

A, called attributes set of the core element e_c, is also a finite nonempty set for the attribute, which can be traditional attribute with point, line and plane;

E_s, called non-core elements set of star association model based on event, is a set of spatial entity objects denoted by $E_s = \{e_1, e_2,..., e_m\}$. Here is only a kind of spatial location association between the core element e_c and each non-core element e_i $(e_i \in E_s)$;

F, called spatiotemporal factors set for describing this event e, is expressed as follows:

$F = \{time, orientation, topology\}$;

P, called predicates set for F, is a finite set of values for $f \in F$, denoted by $P = \{P_{time}, P_{orientation}, P_{topology}\}$.

In this paper, P_f $(f \in F)$ is defined as follows:

$P_{time}(e_c, e_t) = \{before(e_t), after(e_t), equal(e_t)\}$, where e_t is a temporal element;

$P_{orientation}(e_c, e_o) = \{east(e_c), south(e_c), southeast(e_c), west(e_c), southwest(e_c), northwest(e_c), northeast(e_c), north(e_c)\}$, where e_o is an orientation element;

$P_{topology}(e_c, e_s) = \{disjoint(e_c, e_s), coveredby(e_c, e_s), cover(e_c, e_s), contain(e_c, e_s), touch(e_c, e_s), inside(e_c, e_s), overlap(e_c, e_s)\}$, where e_s is a spatial entity objects;

For example, there are three spatiotemporal events from a spatiotemporal database, and then we use the star association model based on event to describe them as follows:

(1) e: a taxi (No.t2) with passenger eastward fast left the school (No.s1) along the riverside (No.r1) at 5 PM.

e_c: $taxi(2)$, is a subject object in this event;

A_1: $\{load, rate\} = \{true, fast\}$;

E_s: $\{school(1), river(1)\}$;

$time$: $\{at\ 5\ PM\}$;

$orientation$: $\{east\}$;

$topology$: $\{a\ taxi\ touch\ the\ river, a\ taxi\ disjoint\ the\ school\}$.

We can use traditional attributes and these predicates to describe the star association pattern for the event, which can be expressed as follows:

$P(1) = \{load = true, rate = fast, before(night), equal(afternoon), east(taxi(2)), disjoint(taxi(2), school(1)), touch(tax(2), river(1))\}$;

(2) e: a taxi (No.t5) with passenger southward slow droved into the school (No.s2), and parked at the gate of the bank (No.b3) at 11 AM.

e_c: $taxi(5)$, is a subject object in this event;

A_1: $\{load, rate\} = \{true, slow\}$;

E_s: $\{school(2), bank(3)\}$;

$time$: $\{at\ 11\ AM\}$;

$orientation$: $\{south\}$;

$topology$: $\{a\ taxi\ is\ inside\ the\ school, a\ taxi\ touch\ the\ bank\}$;

Via traditional attributes and these predicates, the star association pattern for the event can be expressed as follows:

$P(2) = \{load = true, rate = slow, before(afternoon), equal(morning), south(taxi(5)), touch(taxi(5), bank), inside(taxi(5), school(1))\}$;

(3) e: a taxi (No.t6) without passenger southwest slow left the bank (No.b4), and droved into the business street (No.b3) at 8 night.

e_c: $taxi(6)$, is a subject object in this event;

A_1: $\{load, rate\} = \{false, slow\}$;

E_s: $\{business\ street(3), bank(4)\}$;

$time$: $\{at\ 8\ night\}$;

$orientation$: $\{southwest\}$;

$topology$: {*a taxi is covered by the business street*, *a taxi disjoint the bank*} .

Via traditional attributes and these predicates, the star association pattern for the event can be expressed as follows:

$P(3) = \{load = true, rate = slow, after(afternoon), equal(night), south(taxi(6)), touch(taxi(6), bank(4)), inside(taxi(6), business\ street)\}$;

Definition 3.2 Spatiotemporal association patterns mining is defined as discovering frequent spatiotemporal association patterns from spatiotemporal database based on event, namely, frequent star association patterns, whose support is the same as traditional association rules.

In the course of mining spatiotemporal association patterns, there are two key problems as follows:

One is turning an event into a spatiotemporal association patterns, namely, star association patterns. We have solved the problem via definition 3.1;

The other is discovering frequent spatiotemporal association patterns. We use the algorithm as described in Section 4.2 to solve the problem.

4 Frequent spatiotemporal association patterns mining

In this section, firstly, we introduce granular computing based on the star association model, and then propose an algorithm of discovering frequent spatiotemporal association patterns based on granular computing; finally, we compare the algorithm with these traditional mining algorithms, particularly, the Apriori Framework and the FP-growth Framework

4.1 Granular computing based on the star association model

Definition 4.1 A spatiotemporal information system based on the star association model is a six-tuple $STIS = (U, F, A, \{V_a \mid a \in A\}, L, \{I_a \mid a \in A\})$, where

U , called universe of discourse, is a finite nonempty set of events, where each event has a sole core element;

F , called spatiotemporal factor set for describing the event e in U , is expressed as follows:

$F = \{time, orientation, topology\}$;

A , called joined attributes set of an event, is denoted by $A = A_t \cup E_t \cup \{orientation\} \cup E_s$,

Where

A_t , called traditional attributes with point, line and plane;

E_t , is a given group of time division; such as $E_t = \{morning, afternoon, night\}$ or $E_t = \{Monday, Tuesday, Wensday, Thursday, Friday, Saturday, Sunday\}$;

E_s , called non-core generalization set for U , is a finite nonempty set of non-core element category for the star association model based on event;

For example, there are three spatiotemporal events in Section 3; and we can get these spatial entities for them as follows:

$\{school(1), river(1), school(2), bank(3), business\ street(3), bank(4)\}$;

And then, we have the following E_s :

$E_s = \{school, river, bank, street\}$;

There are more details about A in table 1.

V_a , called domain set, is a nonempty finite set of values for attribute a ($a \in A$) , which can be expressed as follows:

$$V_a = \begin{cases} V*_a, & a \in A_t \\ P_{time}(e_c, a), & a \in E_t \\ P_{orientation}(e_c, a), & a = orientation \\ P_{topology}(e_c, a), & a \in E_s \end{cases}$$, where

$V*_a$, called domain of traditional attribute with point, line and plane, is defined as a discrete category set; the others are the predicate sets as described in definition 3.1.

L , called a kind of logical descriptive language, is defined to describe a spatiotemporal event through these traditional attributes and predicates; L can be expressed as $L = \{\ell \mid V_{a_*^1} \times V_{a_*^2} \times ... \times V_{a_*^{|A^*|}}, a_* \in A^* \subseteq A\}$;

I_a , called information function, is a total function that maps an event of U to exactly one value in V_a , namely $I_a: U \to V_a$.

Based on definitions 3.1 and 4.1, for the three spatiotemporal events in Section 3, and we let a time division be $E_t = \{morning, afternoon, night\}$, then we can construct a spatiotemporal information system based on the star association model as follows:

$STIS = (U, F, A, \{V_a \mid a \in A\}, L, \{I_a \mid a \in A\})$, where

$A_t = \{load, rate\}$, is an attribute set of the taxi (called a point entity);

$E_s = \{school, river, bank, street\}$;

So we can create the following mining database as described in table 1.

A / *T_ID*	Taxi(2)	Taxi(5)	Taxi(6)
Load	True	True	False
Rate	Fast	Slow	Slow
Morning	--	Equal	--
Afternoon	Equal	Before	After
Night	Before	--	Equal
Orientation	East	South	Southwest
School	Disjoint	Inside	--
River	Touch	--	--
Bank	--	Touch	Touch
Street	--	--	Inside

Table 1: Mining database.

Definition 4.2 Spatiotemporal information granule is a two-tuple $STIG = (\zeta, \psi(\zeta))$, where

ζ, called the intension of spatiotemporal information granule, is an abstract description of common values of joined attributes shared by events in the extension, which is expressed as $\zeta = (\zeta_1, \zeta_2, ..., \zeta_{|\zeta|})$ $(\zeta_k \in V_{a_*^k}, a_* \in A^* \subseteq A, k = 1, 2, ..., |\zeta|, \zeta \in L)$;

$\psi(\zeta)$, called the extension of spatiotemporal information granule, is a set of events which spatiotemporal information granule applies, which is expressed as follows:

$\psi(\zeta) = \{u \in U \mid I_{a_*^1}(u) = \zeta_1, I_{a_*^2}(u) = \zeta_2, ..., I_{a_*^{|\zeta|}}(u) = \zeta_{|\zeta|}\}$.

Definition 4.3 Atomic spatiotemporal information granule is a two-tuple $ASTIG = (\zeta, \psi(\zeta))$, where

ζ, called the intension of atomic spatiotemporal information granule, is denoted by $\zeta = (\zeta_a)$ $(\zeta_a \in V_a, a \in A, \zeta \in L)$;

$\psi(\zeta)$, called the extension of atomic spatiotemporal information granule, is denoted by the following:

$\psi(\zeta) = \{u \in U \mid I_a(u) = \zeta_a\}$.

Definition 4.4 Intersection operation of spatio- temporal information granule is denoted by Θ. Suppose two spatiotemporal information granules are $STIG_\alpha = (\zeta_\alpha, \psi(\zeta_\alpha))$ and $STIG_\beta = (\zeta_\beta, \psi(\zeta_\beta))$, respectively; if $(\exists \zeta_\alpha^i \in \zeta_\alpha \wedge \zeta_\alpha^i \in V_a) \wedge (\exists \zeta_\beta^j \in \zeta_\beta \wedge \zeta_\beta^j \in V_a)$ then $\zeta_\alpha^i = \zeta_\beta^j$; and so the intersection operation Θ can be expressed as $STIG = (\zeta, \psi(\zeta)) = STIG_\alpha \Theta STIG_\beta$

$= (\zeta_\alpha \cup \zeta_\beta, \psi(\zeta_\alpha) \cap \psi(\zeta_\beta))$.

Definition 4.5 A mixed radix notation system based on a spatiotemporal information system is a triple $M = \{STIS, m, < w_1, w_2, ..., w_m >\}$, where

$STIS$, called a spatiotemporal information system, is expressed as follows:

$STIS = (U, F, A, \{V_a \mid a \in A\}, L, \{I_a \mid a \in A\})$;

m, called the number of bit for the mixed radix notation system, is denoted by $m = |A|$;

$< w_1, w_2, ..., w_m >$, called the weight set of bit for the mixed radix notation system, each element w_i is defined as $w_i = |V_{a_i}| + 1$ $(i = 1, 2, ..., m)$, and $V_{a_i}^k \leftrightarrow k$ $(k = 1, 2, ..., |V_{a_i}|)$, and $|M| = \prod_{i=1}^{m} w_i - 1$.

For example, for the spatiotemporal information system of table 1, we can get the following mixed radix notation system based on a spatiotemporal information system $M = \{STIS, 10, < 3, 3, 4, 4, 4, 9, 8, 8, 8, 8 >\}$.

Definition 4.6 Combinatorial number ratio based on a spatiotemporal information system is defined as $\rho = log_{|U|}^{|M|} > 0$, where

$|U|$, is the number of spatiotemporal events in the spatiotemporal database, which is mapped to the spatiotemporal information system $STIS$;

$|M|$, is a combinatorial number for attribute values in the spatiotemporal database, which is mapped to the spatiotemporal information system $STIS$.

4.2 Discovering frequent spatiotemporal association patterns

In this section, we propose an algorithm of discovering frequent spatiotemporal association patterns based on granular computing, which is denoted by DFSTAP, and then we use the following pseudo code to describe the algorithm DFSTAP.

STD, is a spatiotemporal database based on event;

s, is the given minimal support;

F: saving these maximal frequent spatiotemporal association patterns;

NF: saving these non frequent spatiotemporal association patterns;

Input: STD and s;

Output: F;

(1) $F = \Phi$;
(2) $NF = \Phi$;
(3) Read STD; //reading once database
(4) Create $STIS_{STD} = (U, F, A, \{V_a \mid a \in A\}, L, \{I_a \mid a \in A\})$; //def. 4.1, creating a $STIS$ by the STD
(5) Compute each $ASTIG = (\zeta, \psi(\zeta))$; // def. 4.3
(6) Create $M = \{STIS, m, < w_1, w_2, ..., w_m >\}$; // def. 4.5
(7) For $\forall i \in [1, |M|]$ do {
(8) $M(i) = (\omega_m \omega_{m-1} ... \omega_1)_M$; //a decimal integer i is turned into a mixed radix numeral $(\omega_m \omega_{m-1} ... \omega_1)_M$
(9) $\zeta_{M(i)} = \zeta_m \cup \zeta_{m-1} \cup ... \cup \zeta_1$; // $\zeta_{M(i)}$ is a set of items, each ζ_k is mapped to the ω_k
(10) If $(\forall \varpi \in NF, \varpi \not\subset \zeta)$ then {
(11) Construct $STIG = (\zeta_{M(i)}, \psi(\zeta_{M(i)}))$; // def. 4.4
(12) If $|\psi(\zeta_{M(i)})| \geq s$ then {
(13) Delete σ $(\forall \sigma \in F, \sigma \subset \zeta_{M(i)})$; //deleting all subsets of $\zeta_{M(i)}$ in F
(14) Write $\zeta_{M(i)}$ to F;} //saving frequent itemset
(15) Else
(16) Write $\zeta_{M(i)}$ to NF; //saving non frequent itemset
(17) }
(18) $i++$;}
(19) Output F;

The interval $[1, |M|]$ in the algorithm is the search range of candidate frequent patterns. In other word, the algorithm updates the mixed radix numeral to generate candidate frequent itemsets.

The algorithm discovers frequent spatiotemporal association patterns through constructing spatiotemporal information granule.

4.3 Performance comparison

Based on the introduction in Section 4.2, we know the algorithm DFSTAP is different from traditional frequent patterns mining algorithms, particularly, the Apriori Framework and the FP-growth Framework.

For discovering frequent association patterns, the Apriori Framework is a representative algorithm with candidate, and the FP-growth Framework is a typical algorithm without candidate, and then we compare the algorithm DFSTAP with the Apriori Framework and the FP-growth Framework. The comparative results can be expressed as the following table 2.

Based on the comparison as described in table 2, we can draw the following conclusions:

The Apriori Framework needs to read the database repeatedly, and it joins two frequent itemsets to generate candidate; and so there are lots of calculated amount for discovering frequent patterns. However, the algorithm DFSTAP updates the mixed radix numeral to generate candidates; the speed of which for the latter is faster than the former; additionally, the DFSTAP only needs to read the database once. Hence, the computational complexity of the algorithm DFSTAP is lower than the Apriori Framework. In other words, the algorithm avoids these disadvantages of the Apriori Framework.

In addition, the algorithm DFSTAP uses simple data structure as array to express single format of candidate, and traverses an interval to discover frequent association patterns; so it uses less memory; and it is easy to program and maintain the algorithm. Namely, the DFSTAP has these advantages of the Apriori Framework.

However, for mining frequent association patterns, the FP-growth Framework only needs to read database twice, its advantage is saving reading database, the DFSTAP also has the advantage. But the FP-growth Framework needs to traverse a complex FP-tree; so its computational complexity is higher than the DFSTAP, meanwhile, it also needs to cost more memory, and it is no picnic to program and maintain it. Obviously, the algorithm DFSTAP avoids these disadvantages of the FP-growth Framework.

Comparative items	DFSTAP	Apriori Framework	FP-growth Framework
Reading Database	Once	Many times	Twice
Data structure	Simple	Simple	Complex
Programming	Simple	Simple	Complex
Computational complexity	Low	High	High
Memory usage	Less	Less	More
Generating candidate	Yes	Yes	No
Format of candidate	Digit	Itemset	--
Speed of generating candidate	Fast	Slow	--

Table 2: Performance comparison.

In conclusion, this algorithm is better than traditional mining algorithm in theory.

5 Experimental result

In this section, we design two types of experiments as follows:

One is evaluating the performances of the proposed mining algorithm for discovering frequent spatiotemporal association patterns on different datasets.

The other is discussing the application environments for the proposed mining algorithm.

The first data set is from the GPS data of taxi in a city, for the GPS interval point with a taxi, an event is made of speed, loading, time, and space layout. There are 323080 spatiotemporal events after data filtering; the dataset can be mapped to a spatiotemporal information system $STIS_1$, and then we can create a mixed radix notation system based on the spatiotemporal information system $STIS_1$, which can be expressed as follows:

$M_1 = \{STIS_1, m, < w_1, w_2, ..., w_m >\}$, where

$STIS_1$, is mapped to the spatiotemporal database for the taxi;

$m = 7$, there are seven attributes in the database;

$< w_1, w_2, w_3, w_4, w_5, w_6, w_7 >=< 4, 4, 4, 4, 3, 5, 9 >$.

And we have $\rho = log_{|U|}^{|M|} = log_{323080}^{34559} = 0.824$.

The second data set is from the GPS data of bus in a city, for the GPS interval point with a bus, an event is made of speed, time, grade of service, and space layout. We deal with the dataset to form 40600 spatiotemporal events; the dataset can be mapped to a spatiotemporal information system $STIS_2$, and then we also can create a mixed radix notation system based on the $STIS_2$, which can be expressed as follows:

$M_2 = \{STIS_2, m, < w_1, w_2, ..., w_m >\}$, where

$STIS_2$, is mapped to the spatiotemporal database for the bus;

$m = 6$, there are six attributes in the database;

$< w_1, w_2, w_3, w_4, w_5, w_6, w_7 >=< 3, 5, 4, 6, 7, 8 >$.

And we also have $\rho = log_{|U|}^{|M|} = log_{40600}^{20159} = 0.934$.

Experimental environment is Microsoft Window XP Professional with Intel (R) Core (TM)2 Duo CPU (T6570 @) 2.10 GHz 1.19GHz) and 1.99 GB memory. The software development environment is based on C# with Microsoft Visual Studio 2008.

5.1 The experiments of performance comparison

Here, for discovering frequent spatiotemporal association patterns on the two datasets, we compare the algorithm DFSTAP with the Apriori Framework and the FP-growth Framework. Based on the performance comparison in Section 4.3, we respectively design three groups of experiments on the two datasets.

1. Testing on the first dataset

For the first dataset, we compare the performance as the number of frequent association pattern increases, and the test results are expressed as figure 1; as the maximal length of frequent association pattern increases, and the test results are expressed as figure 2; as the minimal support of frequent association pattern increases, and the test results are expressed as figure 3.

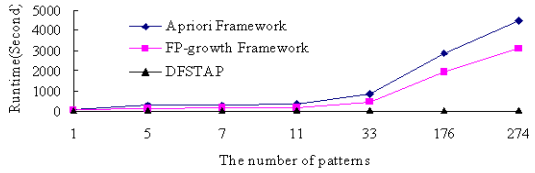

Figure 1: Performance comparison as the number of frequent association pattern increases.

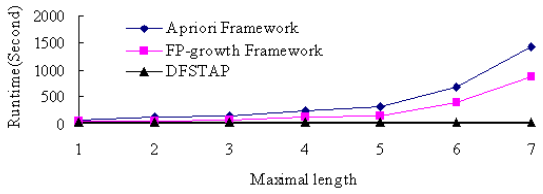

Figure 2: Performance comparison as the maximal length of frequent association pattern increases.

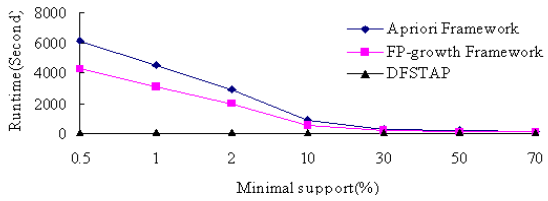

Figure 3: Performance comparison as the minimal support of frequent association pattern increases.

2. Testing on the second dataset

For the second dataset, we compare the performance from three aspects also; in other words, with the number of frequent association pattern, the maximal length, and the minimal support; and their experimental results are expressed as figures 4, 5, and 6, respectively.

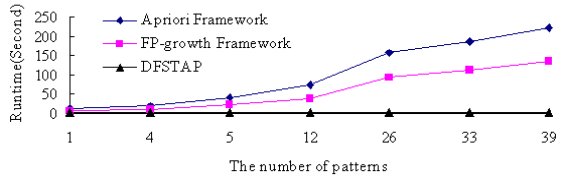

Figure 4: Performance comparison as the number of frequent association pattern increases.

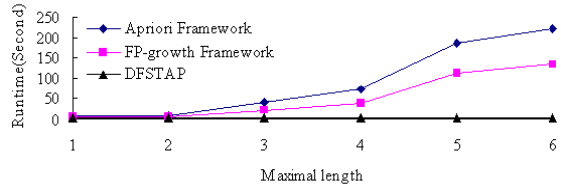

Figure 5: Performance comparison as the maximal length of frequent association pattern increases.

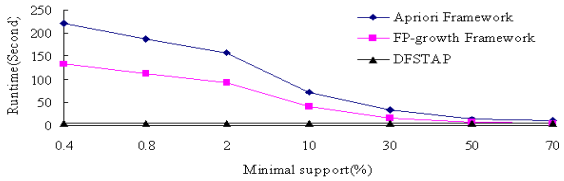

Figure 6: Performance comparison as the minimal support of frequent association pattern increases.

Based on these comparison results from figure 1 to 6, we can draw two conclusions as follows:

One is that the algorithm DFSTAP is better than the Apriori Framework and the FP-growth Framework on the type of mining dataset ($\rho \le 1$).

The other is that the performance of the algorithm DFSTAP does not depend on the number of frequent association pattern, the maximal length, and the minimal support parameter.

5.2 The experiments of discussing the optimal application environments

In this part, we mainly discuss the relationships between the performance and the following parameters:

$|U|$, is the number of spatiotemporal events;

$|M|$, is the combinatorial number for attribute values;

ρ, is the combinatorial number ratio.

1. Testing on the first dataset

For the first dataset, we change database events or database structure to create eight new datasets as table 3.

Name	Weight set	ρ
Data_T 1	<4,4,4,5,9>	$log_{403850}^{2879} = 0.617$
Data_T 2	<4,4,4,5,9>	$log_{323080}^{2879} = 0.628$
Data_T 3	<4,4,4,4,3,5,9>	$log_{403850}^{34559} = 0.810$
The first dataset	<4,4,4,4,3,5,9>	$log_{323080}^{34559} = 0.824$
Data_T 4	<4,4,4,4,3,5,9,3>	$log_{403850}^{103679} = 0.895$
Data_T 5	<4,4,4,4,3,5,9,3>	$log_{323080}^{103679} = 0.910$
Data_T 6	<4,4,4,5,9>	$log_{3231}^{2879} = 0.986$
Data_T 7	<4,4,4,4,3,5,9>	$log_{3231}^{34559} = 1.293$
Data_T 8	<4,4,4,4,3,5,9,3>	$log_{3231}^{103679} = 1.429$

Table 3: Changing description of the first dataset.

As we all know, the performance of the FP-growth Framework is better than the Apriori Framework, so we do not directly compare them in these experiments.

Here, if the minimal support is less than 1%, then we regard it as the lower support; if the minimal support is greater than 30%, then we regard it as the higher support.

(1) The relationship between the performance and ρ (the combinatorial number ratio)

As the minimal support increases, we compare the algorithm DFSTAP with the Apriori Framework and the

FP-growth Framework on the eight datasets. Their results are respectively expressed as figures 7-14.

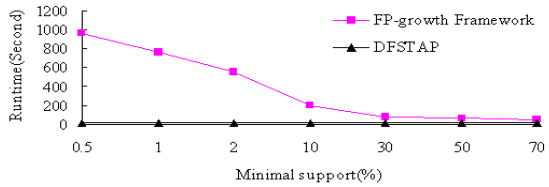

Figure 7: Performance comparison on Data_T 1 ($\rho = 0.617$)

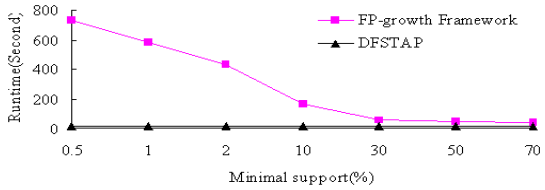

Figure 8: Performance comparison on Data_T 2 ($\rho = 0.628$)

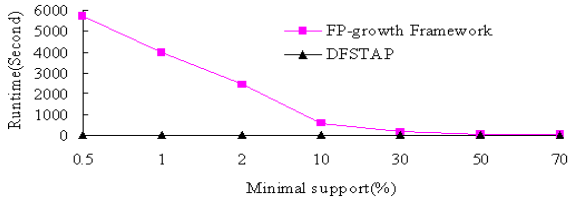

Figure 9: Performance comparison on Data_T 3 ($\rho = 0.810$)

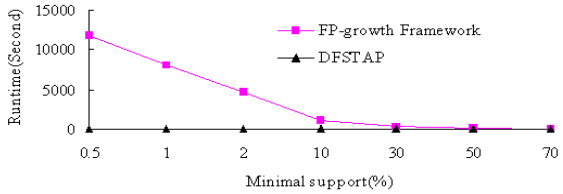

Figure 10: Performance comparison on Data_T 4 ($\rho = 0.895$)

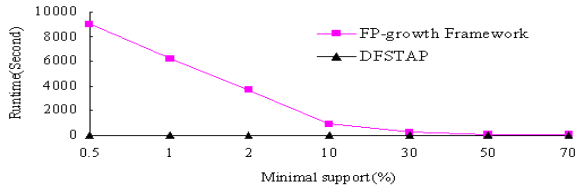

Figure 11: Performance comparison on Data_T 5 ($\rho = 0.910$)

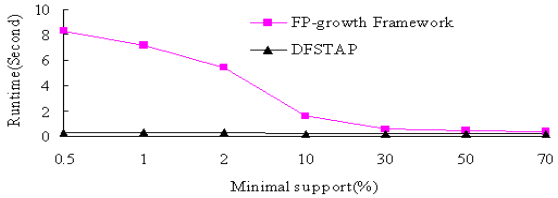

Figure 12: Performance comparison on Data_T 6 ($\rho = 0.986$)

Based on figures 7-12, when $\rho \le 1$, we can know that the performance of the algorithm DFSTAP is better than the Apriori Framework and the FP-growth Framework.

Based on figures 13 and 14, when $\rho > 1$, we can know that the performance of the algorithm DFSTAP is better than the Apriori Framework and the FP-growth Framework for the lower support; but for the higher support, it is not better than them.

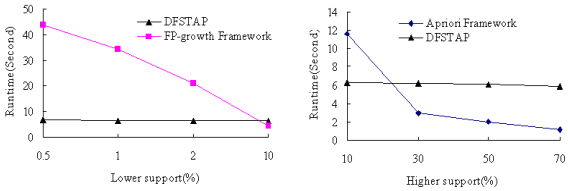

Figure 13: Performance comparison on Data_T 7 ($\rho = 1.293$)

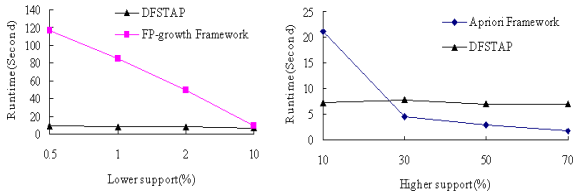

Figure 14: Performance comparison on Data_T 8 ($\rho = 1.429$)

(2) The relationship between the performance and $|M|$ (the combinatorial number for attribute values)

Here, we discuss the variation trend of runtime as the combinatorial number $|M|$ increases when the number of events $|U|$ is invariant. These experimental results can be expressed as figure 15.

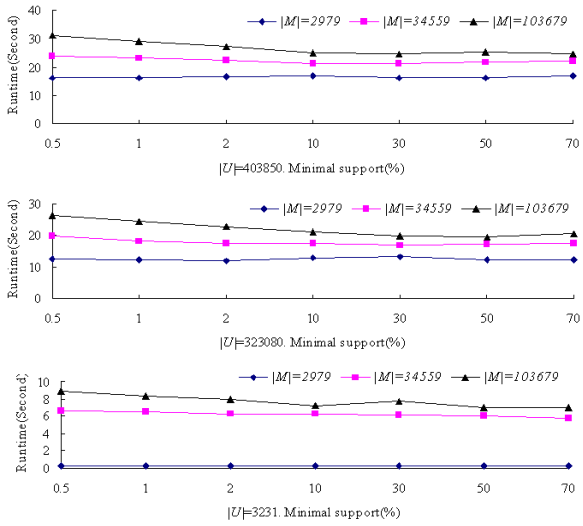

Figure 15: Performance comparison with $|M|$ varying

Based on these results from figure 15, we can know that the runtime of the algorithm DFSTAP is ascending as $|M|$ increases when $|U|$ is invariant.

(3) The relationship between the performance and $|U|$ (the number of events)

When $|M|$ is invariant, we discuss the variation trend of runtime as $|U|$ increases. These experimental results are expressed as figure 16.

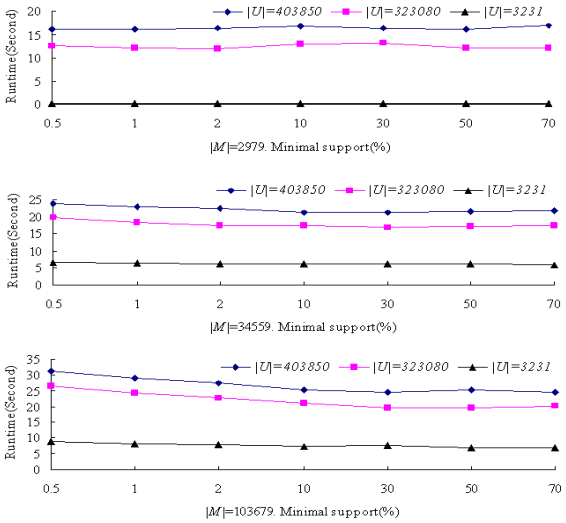

Figure 16: Performance comparison with $|U|$ varying

Based on these results from figure 16, when $|M|$ is invariant, we can know that the runtime of the algorithm DFSTAP is ascending as $|U|$ increases.

2. Testing on the second dataset

For the second dataset, we also use the same method to create eight new datasets as table 4, and compare the performance on these datasets.

Name	Weight set	ρ
Data_B 1	<3,5,6,7,8>	$log_{121800}^{5039} = 0.728$
Data_B 2	<3,5,6,7,8>	$log_{40600}^{5039} = 0.803$
Data_B 3	<3,5,4,6,7,8>	$log_{121800}^{20159} = 0.846$
The second dataset	<3,5,4,6,7,8>	$log_{40600}^{20159} = 0.934$
Data_B 4	<3,5,4,6,7,8,3,6>	$log_{121800}^{362879} = 1.093$
Data_B 5	<3,5,6,7,8>	$log_{2030}^{5039} = 1.119$
Data_B 6	<3,5,4,6,7,8,3,6>	$log_{40600}^{362879} = 1.206$
Data_B 7	<3,5,4,6,7,8>	$log_{2030}^{20159} = 1.301$
Data_B 8	<3,5,4,6,7,8,3,6>	$log_{2030}^{362879} = 1.954$

Table 4: Changing description of the second dataset.

(1) The relationship between the performance and ρ (the combinatorial number ratio)

We use the same method to test on the eight datasets. Their results are respectively expressed as figures 17-24.

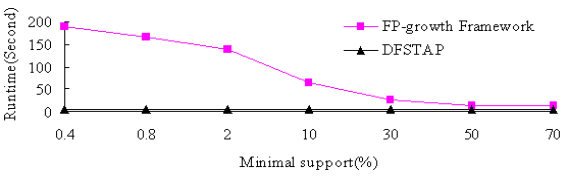

Figure 17: Performance comparison on Data_B 1 ($\rho = 0.728$)

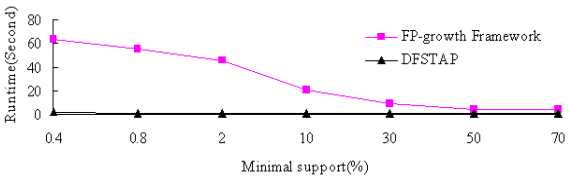

Figure 18: Performance comparison on Data_B 2 ($\rho = 0.803$)

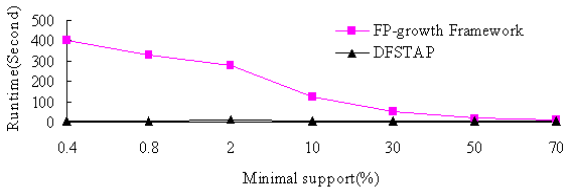

Figure 19: Performance comparison on Data_B 3 ($\rho = 0.846$)

Based on figures 17-19, when $\rho \le 1$, we can know the performance of the algorithm DFSTAP is better than the Apriori Framework and the FP-growth Framework.

Based on figures 20-24, when $\rho > 1$, we can know the performance of the algorithm DFSTAP is better than the Apriori Framework and the FP-growth Framework for the lower support; but for the higher support, it is not better than them.

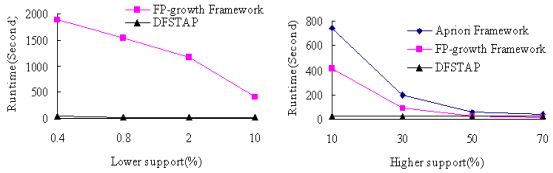

Figure 20: Performance comparison on Data_B 4 ($\rho = 1.093$)

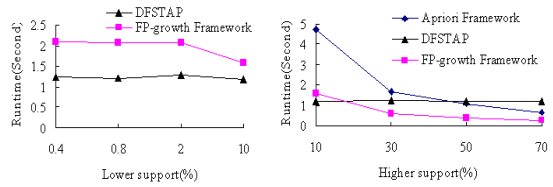

Figure 21: Performance comparison on Data_B 5 ($\rho = 1.119$)

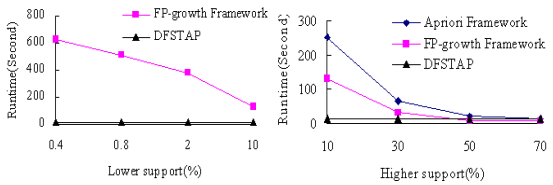

Figure 22: Performance comparison on Data_B 6 ($\rho = 1.206$)

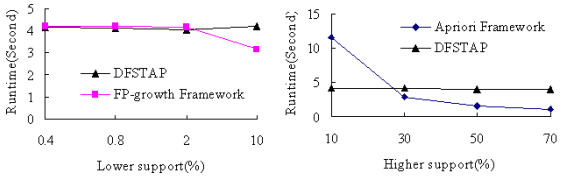

Figure 23: Performance comparison on Data_B 7 ($\rho = 1.301$)

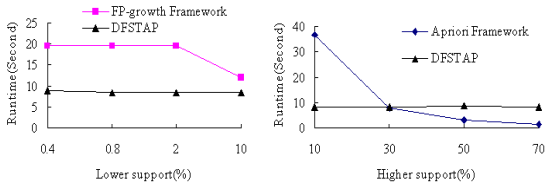

Figure 24: Performance comparison on Data_B 8 ($\rho = 1.954$)

(2) The relationship between the performance and $|M|$ (the combinatorial number for attribute values)

Here, when $|U|$ is invariant, we discuss the variation trend of runtime as $|M|$ increases. These experimental results are expressed as figure 25.

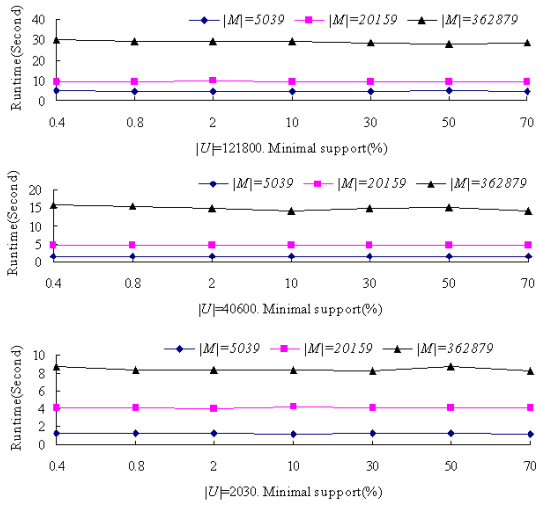

Figure 25: Performance comparison with $|M|$ varying

Based on figure 25, we can know that the runtime of the algorithm DFSTAP is ascending as $|M|$ increases when $|U|$ is invariant.

(3) The relationship between the performance and $|U|$ (the number of events)

When $|M|$ is invariant, we discuss the variation trend of runtime as $|U|$ increases. These experimental results are expressed as figure 26.

Based on figure 26, we can know that the runtime of algorithm DFSTAP is ascending as $|U|$ increases when $|M|$ is invariant.

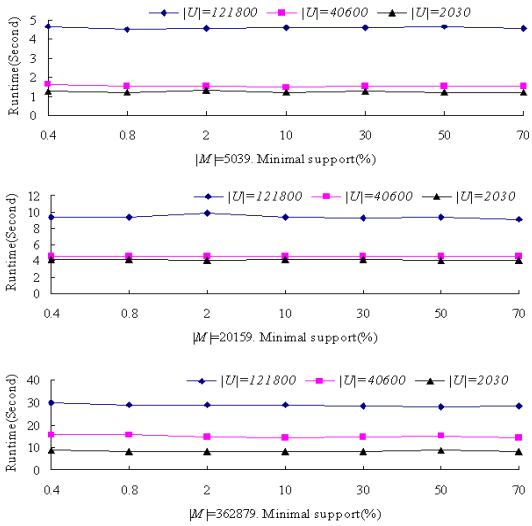

Figure 26: Performance comparison with $|U|$ varying

According to all these experimental results, we can draw the following conclusions:

(1) When the number of events $|U|$ is invariant, the runtime of the algorithm DFSTAP is ascending as the combinatorial number for attribute values $|M|$ increases. Namely, the performance is inversely proportional to the combinatorial number for attribute values $|M|$.

(2) When the combinatorial number for attribute values $|M|$ is invariant, the runtime of the DFSTAP is ascending as the number of events $|U|$ increases. Namely, the performance is inversely proportional to the number of events $|U|$.

(3) For mining frequent spatiotemporal association patterns, the performance of the DFSTAP is better than the Apriori Framework and the FP-growth Framework on the type of datasets ($\rho \le 1$).

On the type of datasets $\rho > 1$, the algorithm DFSTAP is suitable for mining frequent spatiotemporal association patterns with the lower support; but it is unsuitable for mining frequent spatiotemporal association patterns with the higher support.

(4) Since the computing environments generally has the performance bottleneck, when $|M| > \mu$ and $\rho \le 1$ (μ is a parameter with the computing environments), the performance of the DFSTAP also become much worse than the other. For our computing environments in this paper, if $|M| > \mu = 2^{25}$, the interval $[1,|M|]$ is too large, the performance will become much worse.

Hence, the optimal application environments for the algorithm DFSTAP is $|M| \le \mu, \rho \le 1$ (μ is a parameter with the computing environments).

6 Conclusion

In order to simply fast discovering multi-dimensional frequent spatiotemporal association patterns, in this paper, firstly, we construct a star association model based on event, the method of forming association patterns for the

model is very flexible, which can show more spatio-temporal information; and then propose an algorithm of discovering frequent spatiotemporal association patterns based on granular computing, which has two advantages; one is updating the mixed radix numeral to generate candidate; the method improves the speed of generating candidate. The other is adopting granular computing to discover frequent spatiotemporal association patterns to avoid repeatedly reading database. These experimental results indicate that the two key technologies improve the efficiency of algorithm. When $|M| \leq \mu$ (μ is a parameter with the computing environments), the algorithm is suitable for mining frequent patterns on the type of dataset ($\rho \leq 1$), and mining frequent association patterns with the lower support on the type of dataset ($\rho > 1$), but it is unsuitable for mining frequent association patterns with the higher support on the type of dataset ($\rho > 1$). Hence, we need to study the disadvantage in the future.

Acknowledgement

The authors would like to thank the anonymous reviewers for the constructive comment. This work was supported by the Chongqing education commission of science and technology research projects (#KJ121107, #KJ121111, KJ131108) in China.

References

[1] Cucchiara R., Piccardi M., Mello P. (2000). Image analysis and rule-based reasoning for a traffic monitoring system. IEEE Transactions on Intelligent Transportation Systems, IEEE Press, vol. 1, no. 2, pp.119-130.

[2] Pandey G., Atluri G., Steinbach M., et al (2009). An association analysis approach to biclustering. In Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Paris, France, pp. 677-686.

[3] Lee I., Phillips P. (2008). Urban crime analysis through areal categorized multivariate associations mining. Applied Artificial Intelligence, vol. 22, no. 5, pp.483-499.

[4] Huang Y., Kao L., Sandnes F. (2007). Predicting ocean salinity and temperature variations using data mining and fuzzy inference. International Journal of Fuzzy Systems, vol. 9, no. 3, pp. 143-151.

[5] Chang C., Shyue S. (2009). Association rules mining with GIS: An application to Taiwan census 2000. In Proceedings of the 6th international conference on Fuzzy systems and knowledge discovery, Tianjin, China, pp. 65-69.

[6] Zeitouni K., Yeh L., Aufaure M. (2000). Join indices as a tool for spatio data mining. In Proceedings of International Workshop on Temporal, Spatio and Spatiotemporal Data Mining (Berlin, Springer), pp. 102-114.

[7] Mennis J., Liu J. (2005). Mining association rules in spatio-temporal data: An analysis of urban socioeconomic and land cover change. Transactions in GIS, vol. 9, no. 1, pp, 5-17.

[8] Yang H., Parthasarathy S. (2006). Mining spatio and spatio-temporal patterns in scientific data. In Proceedings of 22nd International Conference on Data Engineering Workshops, Atlanta, GA, USA, pp. x146.

[9] Lee I. (2004). Mining multivariate associations within GIS environments. In Proceedings of 17th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Ottawa, Canada, pp. 1062-1071.

[10] Ding W., Eick C., Wang J., et al. (2006). A framework for regional association rule mining in spatio datasets. In Proceedings of the Sixth IEEE International Conference on Data Mining, IEEE Press, Hong Kong, pp. 851-856.

[11] Yang H., Parthasarathy S., Mehta S. (2005). Mining spatio object associations for scientific data. In Proceedings of the 19th International Joint Conference on Artificial Intelligence, Edinburgh, UK, pp.902-907.

[12] Jong S.P., Chen M.S., and Yu P.S. (1997). Using a hash-based method with transaction trimming for mining association rules. IEEE Transactions on Knowledge and Data Engineering, IEEE Press, vol. 9, no. 5, pp. 813-825.

[13] Han J.W., Pei J., and Yin Y.W. et al.(2004). Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. Data Mining and Knowledge Discovery, vol. 8, no. 1, pp.53-87.

[14] Lee H., Han J., Miller H., et al. (2007). Temporal and spatiotemporal data mining. IGI Publishing, New York.

[15] Tanbeer S., Ahmed C., Jeong B., et al. (2009). Efficient single-pass frequent pattern mining using a prefix-tree, Information Sciences, vol.179, no.5, pp.559-583.

[16] Lee A.J.T., Liu Y.H., Tsai H.M., et al. (2008). Mining frequent patterns in image databases with 9D-SPA representation. The Journal of Systems and Software, vol.82, no.4, pp. 603-618.

[17] Hobbs J. R. (1985). Granularity. In Proceedings of the 9th International Joint Conference on Artificial Intelligence, San Francisco, USA, pp. 432-435.

[18] Giunchglia F., Walsh T. (1992). A theory of abstraction. Artificial Intelligence, vol. 57, no. 2-3. pp. 323-389.

[19] Yao Y.Y. (2004). A partition model of granular computing. Lecture Notes in Computer Science Transactions on Rough Sets, vol. 3100, pp.232–253.

[20] Pawlak Z. (1998). Granularity of knowledge, indiscernibility and rough sets. In Proceedings of IEEE Int Conf on Fuzzy Systems, IEEE Press, Anchorage, AK, pp.106–110.

[21] Zhang L., Zhang B. (2003). The quotient space theory of problem solving. Lecture Notes in Computer Science, vol. 2639, pp. 11–15.