Language Support for Temporal Data Mining

Xiaodong Chen [1] and Ilias Petrounias [2]

[1] Department of Computing & Mathematics, Manchester Metropolitan University
John Dalton Building, Manchester M1 5GD, UK
X.Chen@doc.mmu.ac.uk
[2] Department of Computation, UMIST, P.O. Box 88, Manchester M60 1QD, UK
ilias@sna.co.umist.ac.uk

Abstract. Time is an important aspect of all real world phenomena. Any systems, approaches or techniques that are concerned with information need to take into account the temporal aspect of data. Data mining refers to a set of techniques for discovering previously unknown information from existing data in large databases and therefore, the information discovered will be of limited value if its temporal aspects, i.e. validity, periodicity, are not considered. This paper presents a generic definition of temporal patterns and a framework for discovering them. A query language for the mining of such patterns is presented in detail. As an instance of generic patterns, temporal association rules are used as examples of the proposed approach.

1. Introduction

Data mining has been regarded as a field, which potentially offers additional knowledge for a particular application domain. In real world applications, the knowledge that is used in order to aid decision making is always time-varying. Most existing data mining approaches, however, take a static view of an application domain so that the discovered knowledge is considered to be valid indefinitely on the time line. Temporal features of the knowledge are not taken into account in the mining models or processes, despite the fact that the time that something happened is also known and recorded (e.g. the date and time that a point-of-sale transaction took place, or a patient's temperature was taken). If data mining is to be used as a vehicle for better decision making, the existing approaches will in most cases lead into not very significant or interesting results. Consider, for example, a possible association - concerning a supermarket's transactions- between butter and bread (i.e. people who buy butter also buy bread). If someone looks at all the supermarket transactions that are available, let's say for the past ten years, that association might be -with a certain possibility- true. If, however, the highest concentration of people who bought butter and bread can be found up to five years ago, then the discovery of the association is not significant for the present and the future of the supermarket organisation.

Similarly, if someone looks at the rule "over a year 50% of people buy umbrellas" this might be true. However, if the periodic pattern "*during autumn* 85% of people buy umbrellas" is true, then it is certainly more interesting to a company making or selling umbrellas, since it also tells them when and how often the highest concentration of people buying umbrellas can be found. In fact, if one takes into account time components, there is a lot of time-related knowledge which may be discovered from databases in real world applications. For example, in stock market databases, we may find that "*over the last 6 months* some stocks rose by 5% when the financial index rose by 10%". In retail applications, one may also be interested in a rule which states that "*during the summer* customers who buy bread and butter, also buy milk", while in the medical domain, it may be discovered that "some patients experience nausea for about an hour followed by headache *after each meal*". It has been recognised recently ([8], [11]) that such temporal patterns or rules should be investigated and discovered from temporal databases since they can provide accurate information about an evolving business domain, rather than a static one. In this paper, temporal data mining is referred to as a set of techniques for extracting temporal knowledge hiding in temporal databases. The discovered temporal knowledge by temporal data mining can initiate business process change and redevelopment activities in order for an organisation to adopt to changes within its operating area.

The work presented in this paper is focused on the design of a temporal discovery language, which is a part of a framework for temporal data mining [2]. Since the process of knowledge discovery consists of several interactive and iterative stages [4], powerful languages should be used to express different ad hoc data mining tasks [6]. Because SQL has been used almost exclusively as a query language due to the extensive use of relational DBMSs in organisations, it is practically reasonable to develop SQL-like data mining languages ([5], [7]). Following this idea, an SQL-like temporal query and mining language is proposed in the paper, aiming to supply users with the ability to express any temporal data mining problem addressed within the framework. The rest of the paper is organised as follows. As an essential issue within the framework for temporal data mining, temporal mining problems are firstly addressed in section 2, based on the definition of temporal patterns. A temporal query and mining language (TQML) is presented in section 3. Implementation issues are discussed in section 4. Conclusions and future work are presented in section 5.

2. Temporal Data Mining Problems

There are many temporal aspects which can be associated with patterns to depict temporal features of discovered knowledge. The major concerns in this paper are the valid period and the periodicity of patterns. The valid period shows the time interval during which the pattern is valid, while the periodicity conveys when and how often a pattern is repeated. The notion of periodicity [10] is an important temporal feature in the real world. A series of repeated occurrences of a certain type of event at regular intervals is described as a periodic event, which exists in many temporal applications. Both the valid period and the periodicity can be specified by calendar time

expressions, which are composed of calendar units in a specific calendar and may represent different time features such as a calendar interval, an arbitrary contiguous time interval, a periodic time, and a limited periodic time. The following is an example of the limited periodic expression:

Weeks·Days(2)·Hours(12:13) starts_from Years(1993)·Months(5)
and finishes_by Years(1996)·Months(8)

In this expression, "Weeks", "Days", "Hours", "Years", and "Months" are calendar units in the Gregorian calendar. "Weeks·Days(2)·Hours(12:13)" is a *periodic expression*, describing *a periodic time* which consists of a series of periodic intervals (lunchtime) on the repeated cycles (week). "starts_from Years(1993) ·Months(5) and Years(1996)·Months(8)" is an *interval expression*, describing a specific contiguous time framework (May 1993 to August 1996). The whole expression represents "the lunch time of every Monday during the period between May 1993 and August 1996".

According to the discrete, linear model of time, an instant can be represented by a chronon, which is a non-decomposable time interval of some fixed, minimal duration, in which an event takes place. Time line is a totally ordered set of chronons and an interval may be represented by a set of contiguous chronons. We use $\Phi(TimeExp)$ to denote the interpretation of a time expression. The interpretation of a periodic expression, $\Phi(PeriodicExp)$, is a set of periodic intervals. The interpretation of an interval expression, $\Phi(IntervalExp)$, is a contiguous time interval. The interpretation, $\Phi(Periodic \circ IntervalExp)$, of a limited periodic expression, consisting of a periodic expression and a interval expression, is also a set of periodic intervals, such that :

$\Phi(\ PeriodicExp \circ IntervalExp\ ) =$

$\{\ p \mid \exists\ p' \in \Phi(\ PeriodicExp),\ p = p' \cap \Phi(\ IntervalExp\ )$ and $p \neq \varnothing\ \}$

Definition 1: *A temporal pattern* is a triplet *<Patt, PeriodicExp, IntervalExp>*, where *Patt* is a general pattern which may be a trend, a classification rule, an association, a causal relationship, etc., *PeriodicExp* is a periodic time expression or a special symbol *p_null* with $\Phi(p\_null)$ being $\{\boldsymbol{T}\}$, and *IntervalExp* is a general interval expression or a special symbol *i_null* with $\Phi(i\_null)$ being $\boldsymbol{T}$. It expresses that *Patt* holds during each interval in $\Phi(PeriodicExp \circ IntervalExp)$. $\boldsymbol{T}$ is the time domain.

For any temporal pattern of the form < Patt, *PeriodicExp, IntervalExp* >, if *PeriodicExp* is *p_null* and *IntervalExp* is not *i_null*, the expression represents a pattern that refers to an absolute time interval $\Phi(IntervalExp)$; if *PeriodicExp* is not *p_null* and *IntervalExp* is *i_null*, it represents a periodic pattern without any time limitation; if neither *PeriodicExp* is *p_null* nor *IntervalExp* is *i_null*, it represents a periodic pattern which is valid during the time interval $\Phi(IntervalExp)$; otherwise, it represents a non-temporal pattern. The following are some examples of temporal patterns that refer to absolute time intervals or periodic times:

- < "(Emp,=),(Rank,<)⇒(Salary,≤)", *p_null*, starts_from Years(1992)·Months(3) and finishes_by Years(1993) ·Months(7) >: describes the fact that during the period between March 1992 and July 1993, if an employee's rank increases then his/her salary does not decrease.
- < "HikingBoots ⇒ Outerwear", Years·Months(4:6), starts_from Years(1990) and finishes_by Years(1995)] > : shows that every Spring, between 1990 and 1995, shoppers who buy hiking boots, also want outerwear at the same time;

- <"nausea→headache",Days·Hours(6:8),*i_null*>:indicates that patients with a certain disease feel nausea, followed by headaches from 6 to 8 o'clock every morning.

The above examples express a trend appearing during a specific period, a periodic association happening within a relevant period, and a periodic causal relationship, respectively.

Definition 2: Given a set $\boldsymbol{D}$ of time-stamped data over a time domain $\boldsymbol{T}$, we use $\boldsymbol{D}$(p) to denote a subset of $\boldsymbol{D}$, which contains all data with timestamps (whatever they may be: user-defined, valid, decision or transaction time according to the temporal databases literature [10]) belonging to time interval p. We define:

- *<Patt, PeriodicExp, IntervalExp>* holds during interval p, $p \in \Phi(PeriodicExp \circ IntervalExp)$, if *Patt* satisfies all relevant thresholds in $\boldsymbol{D}$(p).
- *<Patt, PeriodicExp, IntervalExp>* satisfies all relevant thresholds with respect to the frequency f % in the dataset $\boldsymbol{D}$ if *<Patt, PeriodicExp, IntervalExp>* holds during no less than f % of intervals in $\Phi(PeriodicExp \circ IntervalExp)$.

In the above definition, the relevant thresholds are given in terms of the forms of interested patterns. The notion of frequency is introduced for measuring the proportion of intervals, during which *Patt* satisfies all relevant thresholds, when compared to the intervals in $\Phi(PeriodicExp \circ IntervalExp)$. It is required that the frequency of any discovered temporal pattern < *Patt, PeriodicExp, IntervalExp* > should not be smaller than the user-specified minimum frequency which is a fraction within [0,1]. In case that $|\Phi(PeriodicExp \circ IntervalExp)| = 1$, $\Phi(PeriodicExp \circ IntervalExp)$ just includes a single interval, so that any non-zero minimum frequency has the same meaning, that is, *Patt* must satisfy all the relevant thresholds during this single interval. Ideally, people might expect that all possibly hidden patterns of a certain type could be discovered without any known temporal features of patterns.

Definition 3 (General Mining Problem): Given a time-stamped dataset $\boldsymbol{D}$ over a time domain $\boldsymbol{T}$, the problem of mining temporal patterns is to discover all patterns of the form *<Patt, PeriodicExp,IntervalExp >* in $\boldsymbol{D}$ that satisfy all the user-specified thresholds with respect to the user-specified minimum frequency *min_f %*.

From a practical point of view, however, people might ask for something with some known temporal features: the mining of all patterns of a certain type during a specific time interval, the mining of all patterns of a certain type with a specific periodicity or the mining of all patterns of a certain type with a specific periodicity during a specific time interval. Additionally they might also be interested in looking for temporal features of a specific pattern: finding all contiguous time intervals during which a specific pattern holds, finding all periodicities of a specific pattern during a specific time interval or finding all limited periodicities of a specific pattern.

3. Temporal Query and Mining Language (TQML)

To a greater or lesser extent, data mining is application-dependent and user-dependent, and knowledge discovery is the process of interactively and iteratively querying patterns. It is important that KDD systems supply users with flexible and

powerful descriptive languages to express data mining tasks. Other languages have been proposed in the literature in order to aid the mining task [5], [7]. The second one of those is only intended to discover association rules, while both of them do not offer any support for discovering temporal patterns and/or features. Here we focus on the requirements for querying a database for the mining of temporal patterns and present TQML which offers the ability -in an SQL-like format- to achieve this task.

The structure of TQML is briefly defined as follows in the BNF grammar:

1) <TQML> ::=
2) Mine <Pattern-Form-Descriptor> (ALL | <specific-pattern>)
3) With Periodicity (ALL | OMISSION | <periodic-expression>)
4) During Interval (ALL | T_DOMAIN | <interval-expression>)
5) [Having Thresholds <threshold-expression-list>]
6) [Shown As <display-form>]
7) In
8) Select <relevant-attribute-list> [, <time-attribute>]
9) From <relation-list>
10) [Where <condition-expression>]
11) [Group By <group-attribute-list> [Having <condition-expression>]]

A mining task in TQML consists of the mining target part (lines 2-6) and the data query part (lines 8-11). In the above definition, <pattern-form-descriptor> points out the form of patterns which users may be interested in and it may be "Trend", "Classification", "Association", "Causality", etc. The option that follows that indicates whether all of the possible patterns should be found or the temporal features of a specific pattern should be extracted. The periodicity of patterns can be expressed by With-Periodicity (option), where "ALL" expresses that all possible periodic patterns or the periodicities of the specific pattern are expected, "OMMISION" shows that the periodicity of patterns is not of interest, and <periodic-expression> gives the periodicity of the expected patterns. The During-Interval(option) can be used for describing the valid period of patterns, where "ALL" expresses that all contiguous time intervals during which patterns (periodic or non-periodic) may exist are expected to be extracted, "T_DOMAIN" makes the assumption that the expected patterns are valid indefinitely, and <interval-expression> indicates the specific time period that users are interested in. Thresholds relevant to different forms of expected patterns can be stated in the Having-Threshold clause and presentation demands can be stated in the Shown-As clause. Note that the granularity of the time interval should be considered as one of criteria if people want to extract the contiguous time intervals during which a specific pattern exists. The data relevant to the data mining task can be stated in the Select-From-Where-Group clause. The Select subclause indicates attributes which are relevant to the mining task. The attribute in <relevant-attribute-list> may be an attribute that exists in the tables appearing in the From clause, an aggregate function (such as Max, Sum, etc.), or a set function which forms an attribute of a nested relation. Each attribute in <relevant-attribute-list> may also be followed by a descriptor, which is relevant to a specific mining model for certain types of patterns that the mining users are interested in. The <time-attribute> indicates the time dimension that users are concerned with in the database, such as transaction-

time, valid-time, decision-time or user-defined-time, as defined in the temporal databases literature [9]. The From-Where clause has the same syntax and semantics as SQL92, constituting a basic query to collect the set of relevant data. The Where subclause can also contain time-related conditions. The selected data may also be grouped by <group-attribute-list>, being presented in the form that mining algorithms expect. The Having clause can be used to filter groups which users want to consider.

3.1 An Example: Mining Temporal Association Rules

The problem of finding association rules was introduced in [1]. Given a set of transactions, where each transaction is a set of items, an association rule is an expression of the form $X \Rightarrow Y$, where X and Y are sets of items, and indicates that the presence of X in a transaction will imply the presence of Y in the same transaction. Consider a supermarket database where the set of items purchased by a customer is stored as a transaction. An example of an association rule is: "60% of transactions that contain bread and butter also contain milk; 30% of all transactions contain both of these items." Here, 60% is called the confidence of the rule and 30% the support of the rule. The meaning of this rule is that customers that purchase bread and butter also buy milk. The problem of mining association rules is the attempt to find all association rules satisfying the user-specified minimum support and minimum confidence. This problem has been extended for discovering temporal association rules in [3].

Let $\boldsymbol{I}=\{i_1, i_2, \ldots, i_l\}$ be a set of literals which are called items. A set of items $X \subset \boldsymbol{I}$ is called an itemset. Let $\boldsymbol{D}$ be a set of time-stamped transactions. Each time-stamped transaction S is a triplet <tid, itemset, timestamp>, where S.tid is the transaction identifier, S.itemset is a set of items, such that S.itemset $\subseteq \boldsymbol{I}$, and S.timestamp is an instant which the transaction S is stamped with, such that S.timestamp $\in \boldsymbol{T}$. We say a transaction S contains an itemset X if $X \subseteq$ S.itemset.

Definition 4: *A temporal association rule* is a triplet *<AssoRule, PeriodicExp,IntervalExp>*, where *AssoRule* is an implication of the form $X \Rightarrow Y$ such that $X \subset I$, $Y \subset I$ and $X \cap Y = \varnothing$, *PeriodicExp* is a periodic expression and *IntervalExp* is an interval expression. We define:

1) The rule has confidence *c*% during interval p_i, $p_i \in \Phi(PeriodicExp \circ IntervalExp)$, if not less than *c*% of transactions in $\boldsymbol{D}(p_i)$ that contain X also contain Y.
2) The rule has support *s*% during interval p_i, $p_i \in \Phi(PeriodicExp \circ IntervalExp)$, if no less than *s* % of transactions in $\boldsymbol{D}(p_i)$ contain $X \cup Y$.
3) The rule has confidence *c*% and support *s*% with respect to the frequency *f*% in the transaction set $\boldsymbol{D}$ (or saying the database $\boldsymbol{D}$) if it has confidence *c*% and support *s*% during not less than *f*% of intervals in $\Phi(PeriodicExp \circ IntervalExp)$.

Depending on user's interest, the mining problem may be finding all possible associations with a specific periodicity, or finding all possible periodicities of a specific association. In any case, the potential temporal association, $<X \Rightarrow Y$, *TimeExp*>, must satisfy the user-specified minimum support *min_s* % and confidence *min_c* % with respect to the user-specified minimum frequency *min_f* %.

The discussion about mining algorithms for temporal association rules is beyond the scope of this paper. Here, we only take the problem of mining temporal association rules as an example, in order to demonstrate the use of the temporal discovery language. Consider a Sales database containing two relations Items and Purchase as shown in Figure 1.

Example 1: Find all periodic association rules that convey purchase patterns every summer (assuming that summer starts from the sixth month of each year) between July 1990 and June 1996, with the thresholds of support, confidence and frequency being 0.6, 0.75 and 0.8, respectively. Here, we are only concerned with items whose retail prices are not less than £10 and transactions in which the number of purchased items is greater than 3:

Mine Association_Rules (ALL)
With Periodicity(Years·Months(6:8))
During Interval(starts_from Years(1990)·Months(7)
and finishes_by Years(1996)·Months(6))
Having Thresholds support = 0.6, confidence = 0.75, frequency = 0.8.
In
Select *trans-no: TID, Set(item-name): ItemSet, trans-time: TimeStamp*
From purchase, items
Where purchase.item-no = items.item-no and items.retail-price ≥ £10
Group By purchase.trans-no Having Count(*) > 3

Items

item-no	item-name	brand	category	retail-price

Purchase

trans-no	transaction-time	customer	item-no	qty	amount

Fig. 1. Relation Schemes in Sales Database

Example 2: Find all periodicities of the association, "H*iking Boots⇒Outerwear*", with the thresholds of support, confidence and frequency being 085, 0.90 and 0.75, respectively:

Mine Association_Rules ("H*ikingBoots⇒Outerwear*")
With Periodicity(ALL)
During Interval(T_DOMAIN)
Having Thresholds support = 0.85, confidence = 0.90, frequency = 0.75.
In
Select *trans-no: TID, Set(item-name): ItemSet, trans-time: TimeStamp*
From purchase, items
Where purchase.item-no = items.item-no
Group By purchase.trans-no.

In the above examples, the "Select-From-Where-Group" part expresses a basic data query demand. The result of the data query is a nested relation, forming the data set relevant to this mining task. "TID" and ItemSet are descriptors which indicate the

"trans-no" and "Set(item-name)" as the "tid" and "itemset", respectively, in the mining model for temporal associations. The "trans-time" in the database is chosen as the time-stamp.

4. Implementation Issues

There are various challenging works involved in the implementation of the temporal query and mining language. As space is limited, only three major issues are discussed below.

Query Generation: The first step in extracting interesting knowledge hidden in databases is querying the data relevant to the potential knowledge. Although the essential data for date mining is specified in the data query part, the mining target should also be considered in order to generate the actual query (expressed in a TSQL2-like intermediate language) for only necessary data. Consider example 1 in section 3, for instance, the temporal aspects in the mining target implicitly show that only those transactions that happened during June to August of every year between July 1990 and June 1996 will make contribution to the mining task.

Time Support: This is a key issue for supporting the processing of temporal aspects in data mining. Time support is required over the entire course of temporal data mining. Time support mechanisms are used to interpret any time aspects in the mining task in TQML while generating relevant temporal query demands and constructing internal representations for temporal mining models, and to evaluate periodicities and interval comparison operations when accessing data and searching for patterns. Currently, some commercial DBMSs provide support for dates and time. However, they usually support conventional calendar units, such as Hours, Days, Weeks, Months and Years, in the Gregorian calendar. A calendar knowledge base is needed for time support mechanisms. The definitions of all relevant calendars are maintained and managed in this knowledge base.

Search Performance: The performance of search algorithms for temporal patterns is another crucial issue in temporal data mining. In many cases, where only specific temporal features are interesting, the search performance is reasonable. However, the performance of search algorithms attempting to identify all possible temporal patterns (especially, periodic patterns) may be worse than the performance of one that does not take into account the time component. Special techniques for different mining problems and models need to be used for improving the search performance.

5. Conclusions and Future Work

With the large amount of temporal knowledge stored in databases and the growth of temporal database research, more and more attention is being paid to temporal data mining due to the value of temporal data. We argue that the temporal aspects of potential patterns hidden in temporal databases are important since they can provide

companies or organisations with accurate and evolving information for decision making. Until now little work involves the discovery of temporal patterns with temporal features. This paper presented a temporal discovery language with support for mining temporal patterns. Temporal patterns have been defined in this paper by associating general patterns with temporal features which are represented by calendar time expressions. Two different aspects of temporal features of patterns are identified in this paper: the valid period and periodicity of patterns. The presented mining language is an effective tool for expressing different kinds of temporal data mining tasks addressed within a framework for temporal mining.

Current work is concentrating on the implementation of the temporal discovery language and the extension of calendar support beyond that of the Gregorian calendar. Also work is being carried out in order to enhance the language so that it can be used both as a fully fledged temporal and data mining language. Finally, we are focusing on the effects of changes of business rules on temporal patterns.

References

1. Agrawal, R., Imielinski, T., and Swami, A., 1993, *"Mining Associations between Sets of Items in Massive Databases"*, Proceedings of the ACM SIGMOD International conference on Management of Data, Washington D.C., May 1993.
2. Chen, X. and Petrounias, I., 1998, "An Architecture for Temporal Data Mining", IEE Digest (The IEE Colloquium on Knowledge Discovery and Data Mining), No.98/310, PP.8/1-8/4, London, UK, May 1998.
3. Chen, X., Petrounias, I., and Heathfield, H., 1998, "Discovering Temporal Association Rules in Temporal Databases", in the Proceedings of the International Workshop on Issues and Applications of Database Technology (IADT'98), Berlin, Germany, July 1998.
4. Fayyad, U, Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R., 1996, *"Advances in Knowledge Discovery and Data Mining"*, The AAAI Press/The MIT Press, 1996.
5. Han, J., Fu, Y., Koperski, K., Wang W., and Zaiane, O., 1996, "DMQL: A Data Mining Query Language for Relational Databases", 1996 SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'96), Montreal, Canada, June 1996.
6. Imielinski, T., and Mannila H., 1996, "A Database Perspective on Knowledge Discovery", CACM, Vol.39, No.11, Nov. 1996.
7. Meo, R., Psaile, G., and Ceri, S., 1996, *"A New SQL-like Operator for Mining Association Rules"*, Proceedings of the 22nd International Conference on Very Large Data Bases (VLDB'96), Bombay, India, 1996.
8. Saraee, M., and Theodoulidis, B., 1995, *"Knowledge Discovery in Temporal Databases"*, Proceedings of IEE Colloquium on Knowledge Discovery in Databases, pp. 1-4, 1995.
9. Tansel, A., et al, 1994, *"Temporal Databases: Theory, Design, and Implementation"*, Benjamin/Cummings, Redwood City, CA, 1994.
10. Tuzhilin, A. and Clifford, J., 1995, *"On Periodicity in Temporal Databases"*, Information Systems, Vol.20, No.8, PP.619-639,1995.
11. Weiss, S. and Indurkhya, N., 1998, *"Predictive Data Mining"*, Morgan Kaufmann Publishers, Inc., San Francisco, USA, 1998.