Knowledge Discovery in Temporal Databases

Mohamad H. Saraee and Babis Theodoulidis

UMIST
Department of Computation
P.O. Box 88
Manchester M60 1QD
United Kingdom
email: {saraee, babis}@sna.co.umist.ac.uk

Abstract

The essence of data mining is the nontrivial extraction of implicit, previously unknown, and potentially useful information from data. Existing data mining tools consider snapshots of data and therefore unable to handle the complexity of a dynamic environment, such as financial applications which contain a huge amount of data that changes over time. The knowledge discovered has limited value since the temporal nature of data is not taken into account but only the current or latest snapshot. In this paper, we present our framework for data mining in temporal databases. We believe that the next-generation database systems and in particular those that accommodate temporal features are most appropriate platform for knowledge discovery.

Keywords: Data Mining, Knowledge Discovery, Temporal Databases, Mining Association Rules, Pattern Discovery.

1. Introduction

Data mining or knowledge discovery in databases is the process of applying statistical, machine learning and other techniques to classical databases [9]. Classical database model static aspects of data. Thus, the information in database consists of data currently true in the world. However in many dynamic applications such as financial applications information about the history of data is just as important as static information[1]. The knowledge discovered from classical databases has limited value since the temporal nature of data is not taken into account but only the current or latest snapshot. We believe that the next-generation database systems and in particular those that accommodate temporal features are the most appropriate platform for knowledge discovery. When history is taken into account the discovery technique also considers the evolution of data through time. Temporal database system has become increasingly popular and influential in the development of new generation database systems. This motivates the investigation of mechanisms for knowledge discovery in temporal databases.

The key issues addressed in this work include:

- Make use of database techniques to discover rules in databases.
- Taking in to account the temporal aspects of data to discover rules in databases.

Two important aspects that motivate our research are that first, in most application domains the information about history of data is just as important and useful as the data itself. It is essential to consider snapshots of data at one or several points of time and make predictions for future terms using the discovered regularities. Second aspect is to investigate if there is any change in patterns over time and if there is one then we could focus on that time period of data and conduct deeper search to see the various effects that had taken place.
Our survey in knowledge discovery systems has indicated that up to date there is no knowledge discovery system to deal with temporal databases. One of the main reasons for this is due to the fact that up to date there exists no commercially available Temporal Database Management System (TDBMS) [7]. The ORES TDBMS, an ESPRIT project, currently under development by UMIST as an extension of an existing commercial product namely the relational database management INGRES, allows us to create temporal databases [7],[17],[18],[19].

In section 2 data mining in database and its necessity is discussed. In particular we study the properties of association rule discovery in relations. We then present some examples to show how the quality and correctness of discovered patterns can be increased by taking into account the temporality of changing data. In section 3, overview of temporal databases is presented, and we describe how the ORES TDBMS could provide the necessary functionality to infer accurate and valuable knowledge from temporal databases. In section 4 data mining in temporal database is discussed and our overall framework for our data mining system under research is introduced. In particular, we study two common classes of database mining problems involving associations and sequences as the main components of this framework. Section 5 concludes with a summary of the paper and outline a research plan for the development of a powerful, hybrid data mining system for temporal databases.

2. Data Mining in Databases

The total amount of information in the world is estimated to be doubling every 20 months and the size of databases is probably growing even faster [9]. In 1989 the total number of databases in the world was estimated at five million. The automation of activities in all areas including business, engineering, science and government produces an ever-increasing stream of data. Everyday new applications of databases arise, and even simple transactions such as paying by credit card or using the telephone, are typically recorded.

This large amount of data typically sits in computer centres unused until an auditor or government regulation deems it safe to throw out [13]. This scenario is changing radically as corporations around the world take a second look at the power hidden in their databases. Instead of files figuratively gathering dust, historical data is now being viewed as an invaluable assets that can uncover patterns that can provide useful information about the past and help understand and plan for the future.

The need for supporting the process of querying and analysing databases has been widely recognised. Data mining was ranked one of the most promising topics of database research for the 90s[8]. It is an area of common interest for researchers in intelligent databases, machine learning, knowledge acquisition for expert systems and data visualisation.

Different aspects of data mining have been explored in the literature both in the context of AI[15] and in the context of databases[2,10,14,9]. In classification, tuples are grouped together based on some common characteristic, and rules are generated to describe this grouping. Work is also has been carried out to search for sequential patterns[2]. Finding association rules is described as trying to discover frequently occurring patterns within database[15,3,4,5]. Our interest is in the problem of finding association rules and similar sequences in temporal databases. None of reported works in the literature have considered temporal database as their platform for data mining. Applications of association rules include customer behaviour analysis for example in a supermarket or banking environment, and predictions.
2.1 Association Rules

Association rules introduced by Agrawal, et. al [2], are an interesting, and simple class of database regularities, and useful in various analysis and prediction tasks. Association rule is in a form of \[ X \& Y \rightarrow Z \] and a confidence factor attached to it. If we consider a database of supermarket transactions with \( X, Y \) and \( Z \) to be the attributes, then we say that the rule \( X \& Y \rightarrow Z \) has 90% confidence, if when a customer bought \( X \) and \( Y \), in 90% of cases, the customer also bought \( Z \). A rule must also have some minimum user defined support, for example; \( X \& Y \rightarrow Z \) should hold in some minimum percentage of transactions to have business values.

2.2 Sequential Patterns

Another major source for database mining is ordered data, such as temporal data related to stock and point of sales data[2]. For example, we are given a database of customer transactions over a period of time, each transaction is a list of items in a visit and all transactions of a particular customer are temporally ordered. Now we look for all the sequential buying patterns supported by a specified minimum fraction of customers e.g.”What item sold in sequence?”, to which the response could be “dress followed by matching shoes”. Also an example of rule over stock market data could be “when BT stock goes up on 2 consecutive days and Mercury stock does not fall during this period, Orange stock goes up the next day 75% of the times.” Sequence rules are handled in a similar way to the association rules. The temporal nature of relationships between antecedents and consequents can be handled by the TDBMS.

3. Temporal Databases and ORES

Time is the fourth dimension of the world after three spatial dimensions and it is the most important one, since any kind of information happened sometime. Classical databases model static aspects of data, and therefore the information in the database consists of data currently true in the world. On the other hand, temporal databases are capable of storing data and their evolution in terms of different time dimensions such as valid and transaction time.

If for example, we wish to model and manipulate information about the content of the database through time, conceptually we can associate to each time \( t \) the state \( I_t \), of the database at time \( t \). Thus the database appears as a sequence of states (snapshots) indexed by some time domain. In temporal database two possible meanings of \( I_t \) have been defined. First is that \( I_t \) represents the data that was true in the world at time \( t \), this view of time is referred to as valid time. The second possibility is that time represents the moment when the information was recorded in the database, and this is called transaction time[1].

There is also user-defined time which is the time linked with each individual user's perception of the real world. It is an uninterpreted attribute domain of date and time. The SQL2 standard has explicit support for user-defined time in its datetime and interval types. User-defined time is parallel to domains such as “money” and “integer”. Unlike transaction time and valid time, it has no special query language support. It may used for attributes such as “birthday” and “hiring date”[12].

There are no restrictions on how valid times may be incorporated in to the tuples. The valid times may be incorporated by including one or more additional valid time attributes in the relation schema, or by including the valid times as a component of the values of the application specific attributes. The transaction time of a database fact is the time when fact is current in the database and may be retrieved. A transaction time relation is a relation with exactly one system supported transaction time.
Temporal database support both valid time and transaction time. A temporal database can view information valid at some moment in time, seen as of some other moment, and, also the time concept occurs at different levels during the specification of the components of the application.

ORES TDBMS is based on ERT (Entity Relationship Time) data model which is an extension of ER data model with the notions of complex objects and time. An example ERT schema is shown in figure 1. The domain structure produced by the ERT contains objects (such as employee, department, etc.), the attributes of these objects (such as salary, name, etc.) and the relationships among objects and between these objects and their associated attributes (employee has salary, employee works_for department, etc.). Apart from these basic concepts, ERT represents features such as time over which the objects, attributes and relationships hold and also, cardinality constraints [19]. For example, in figure 1 Employee and Department are time varying objects and as such their evolution is explicitly kept.

![Figure 1. An Example ERT Schema](image)

Besides the temporal dimension, ERT differs from the entity-relationship model in that it regards any association between objects in the unified form of a relationship, thus avoiding the unnecessary distinction between attributes and relationships. It also accommodates taxonomic hierarchies and complex objects and for the purpose of modelling complex database applications it offers abstraction mechanisms such as classification, generalisation, aggregation and grouping.

4. Data Mining in Temporal Databases

Despite many Data Mining tools which have been developed to discover knowledge from classical databases, massive amounts of data are still largely unexplored and are either stored primary in an off-line store or are on the verge of being thrown away. The
main reason for this situation is that classical database systems are specialised only for
the creation, manipulation and processing of fixed format snapshot records, and unable
to handle the complexity of a dynamic environment, such as a production process, or a
financial application. These applications contain a huge amount of data that changes
over time. The knowledge which discovered from a snapshot of data has only some
significant value and sometimes can lead to erroneous discoveries, because the temporal
nature of data is not taken into account.

Temporal databases are the most appropriate platform for data mining since they
provide the necessary tools and techniques to store and reason about the history of data
and their time period of validity. Thus, we can reason not only about the current data
but also, about their evolution over time.

For example, if we consider a human resources database that records information about
employees, jobs and salaries, a snapshot of the data can provide information about the
current job and salary or in the best case, if provision was made to provide information
for jobs and salaries over a limited time period, say a year. On the other hand, temporal
databases can accommodate the complete history of jobs and salaries thus, increasing
both quantity and quality of the input to a data mining algorithm.

For scientific databases with timestamped on sensor data, for example in weather data,
geological, environmental, astrophysics databases etc., we could ask queries of the
form “find past days in which the solar magnetic wind showed patterns similar to
today’s pattern” to help in predictions of the earth’s magnetic field.

Now we give some examples that involve temporal reasoning:

• Example 1. Find a rule such as “65% of employees who take vacation between
  June 1 to August 1 of 1991, 1992 and 1994 have requested to extend their
  vacations.”

• Example 2. Find a rule such as “90% of patients who received medical attention
  between 9 AM to 3 PM on Monday 14/3/94 and between 3 PM to 9 PM on
  Wednesday 23/3/94 have come back for treatment of diarrhoea.”

• Example 3. Find a rule such as “ 67% of managers who make less than £25000 for
  the last two years have changed job”, while only 16% of employees who make
  more than £25000 for the last two years have changed jobs”

It can be concluded from the above examples that finding similar patterns in data would
require historical information.

As we mentioned, the available data mining tools are designed to work with classical
databases and are not capable to handle temporal data. Two approaches have been
considered. One is to change the temporal database structure to some sort of simple
table suitable to be inputted to the existing data mining tools. The other approach is to
develop new algorithms or to change and make enhancements to the existing algorithms
to directly work with temporal databases. We are aiming to work on developing new
algorithms and making enhancements to existing tools.

The temporal database can then be used as an input to our overall data mining
framework shown in Figure 2. The discovery techniques currently under investigation
are association rules and sequential patterns. The main system will interact with the
domain expert to get measures for rule confidence, rule support and time confidence to
rank and prioritise the discovered knowledge.

In the rest of this section we discuss the proposed extensions to existing mining
algorithms for association rules and sequential patterns.
First algorithm to solve the mining association rules developed by Agrawal in QUEST project known as AIS algorithm[2]. The AIS algorithm, later on enhanced to another algorithm known as Apriori also AprioriTid and AprioriHybrid. The idea behind this algorithm is that for a given set of transaction D, generate all association rules that have support and confidence greater than the user specified minimum support and minimum confidence or formally:

“An association rule is an implication of the form $X \Rightarrow Y$, where $X \subseteq I$, $Y \subseteq I$ and $X \cap Y = \emptyset$. The rule $X \Rightarrow Y$ holds in the transaction set $D$ with confident $c\%$ of transaction in $D$ that contain $X$ also contain $Y$. The rule $X \Rightarrow Y$ has support $s$ in the transaction set $D$ if $s\%$ of transactions in $D$ contain $X \cup Y$.”

In above definition $I = \{i_1, i_2, i_3, i_4, \ldots, i_m\}$ are a set of literal, called items and $D$ is a set of transactions, where each transaction $T$ is a set of items such that $T \subseteq I$ and also TID is a unique identifier which associated with each transaction.
In our framework, enhancements to the Apriori algorithm (from now we called it basic algorithm) for finding association rules are proposed in order to handle temporal data. These enhancements will be further investigated and refined into a set of temporal data mining techniques. In the sequel, these enhancements are further elaborated.

For temporal databases which support transaction time, the discovery of association rules is expressed as follows:

- Find all association rules that hold in the transaction set \( D' \) with confident \( c \) and support \( s \). \( D' \) is subset of \( D \) where transaction time is within the time interval of \( t \). \( t \) defined by user in the same way that he/she defines support and confidence factor. Formally, “An association rule is an implication of the form \( X \Rightarrow Y \), where \( X \subseteq I \), \( Y \subseteq I \) and \( X \cap Y = \emptyset \). The rule \( X \Rightarrow Y \) holds in the transaction set \( D \) with confident \( c \) if \( c\% \) of transaction in \( D \) that contain \( X \) also contain \( Y \). The rule \( X \Rightarrow Y \) has support \( s \) in the transaction set \( D \) if \( s\% \) of transactions in \( D \) contain \( X \cup Y \) during the time \( t \). \( t \) is transaction time and could be express in absolute or interval time. For example period of 2 weeks, or on 23/4/1995.

For temporal database with transaction and valid time support, we introduce “time confidence” a new time based measure of confidence. The enhancement to the basic algorithm for temporal database with time confidence help us in the selection of large itemsets and time involvement syntactic constraint as follow:

- In finding association rules, selection of large itemsets and candidate itemsets is not only based on rule support, but also on time confidence. The time confidence is calculated based on some user predefined terms. For example, rules discovered for the transactions in first half of the month are more important than the second half and therefore, the given time confidence value is greater. Determination of time confidence depends on many parameters and it could be a very complex task. This is an area of further research. In some cases a rule like “95% of shoppers, who shop at Arndale centre buy candies in Christmas time” may not be as important as “39% of shoppers who shop at Arndale centre buy candies in summer time”. The given time confidence for summer period is greater than Christmas period.

- Syntactic constraint further constrain the admissible combinations. For example, if only rules involving an item \( I_x \) in antecedent are of interest, or only rules involving an item \( I_x \) in antecedent with specified valid time are of interest.

- Finding Interesting rules from large sets of discovered association rules have been researched in some extent, however we would be interested to see if the measure of interest of the rule could be based on the history, valid time and interval of its components. The different between enhancement 2 and this one, is in here, we apply the time confidence factor to discovered rules to prune more interesting rules among discovered one, while the enhancement 1 is to consider time confidence to discover rules.

- Discovering frequent episodes in sequence is important. Sequences of events describing the behaviour and actions of users or systems can be collected in several domain. Most data mining techniques are adapted toward the analysis of unordered collections of data, however there are important application areas where data to be analysed has an inherent sequential structure. An episode is defined to be a collection of events that occurs within time intervals of a given size in a given partial order.

- Sequence rules can be viewed as a special case of association rules. We have a set of object \( O \) consists of timestamps (with different granularity such as days, minutes, hours etc.). In these rules, antecedents and consequents contain literals...
that are related through the temporal component. In most cases, antecedents and consequents come in sequence in time. This is a special case of syntactic constraint. Of course the support constraint also play a major role here as well. This means if the number of timestamps for which the antecedent of a rule is satisfied is too small, the statistical value of such a rule is insignificant.

The above enhancements to existing data mining techniques is the first attempt to define a set of usable and useful temporal data mining techniques for association rules and sequential patterns.

5. Summary

In this paper, a framework for data mining in temporal databases is introduced based on existing data mining techniques for association rules and sequential patterns. The problem of discovering large itemsets was addressed by constructing a candidate set, those itemsets that meet the large itemset requirement. We propose an number of enhancements to the basic algorithm to enable the discovery of temporal data.

There are a number of issues that need to be addressed in order to construct an effective data mining system for temporal databases. The first consideration should be given to testing the appropriateness and applicability of the proposed enhancements. This will take place by applying the new techniques to a temporal dataset. We have already started in this direction by experimenting with a large stroke and quality of hypertension control dataset.

We hope that this experiment will lead to a further refinement of the existing proposals and the incorporation of new proposals into the framework. Furthermore, we would be looking into the visualisation aspects of data mining to improve user involvement and the incorporation of other features such as hierarchies and episodes into the proposed techniques.

In summary, the work reported in this paper focuses on two areas and their integration. On one side, data mining as a technique to increase the quality of data and on the other side temporal databases as a technique to keep the evolution of data. We believe that their integration will lead to even higher quality data and discovered knowledge.

References


[11] Houtsma S., Swami A. “Set Oriented Mining for Association Rules in Relational Databases” Inter. Conf. on Data Engineering


