STATISTICAL QUERIES ON HISTORICAL RELATIONAL DATABASES

WIN MAUNG, CHIT SWE1 AND MEHMET A. ORGUN

Department of Computing, Macquarie University, Sydney, NSW 2109, Australia
E-mail: {winmg,cswe,mehmet}@comp.mq.edu.au

There has been much attention given recently to the task of extracting statistical information and interesting features from historical relational databases with the more pragmatic aim of
- Finding trends in the evolution of temporal data,
- Analysing the past values of time sequence data to detect changes in the behaviour of historical data,
- Measuring serial dependence in the data sequence, and
- Discovering repeating appearance of interesting similar patterns and their cycles.

Though there are many ways to model historical relational databases and associated query languages, the widely accepted approach is to extend the relational database concept and SQL language to temporal dimension. We follow this approach and augment some operators to the existing facilities for performing data analysis and statistical queries on the time sequence data generated from historical relational databases.

1 Introduction

Modelling, storing and manipulating time varying information in databases have been an intense area of research interest for some time. In this connection, semantics of time at the conceptual level, development of temporal database models, especially temporal extension of relational databases, and design of query languages particularly extension of languages such as SQL and QUEL are the main topics of discussion. Further along the development naturally follows providing facilities for querying statistical information and discovering interesting patterns that may exist in time varying data sequences stored in temporal databases.

In this paper, we explore the statistical dimension of a historical relational database (temporal database supporting valid-time only) by incorporating an inner product operator in a temporal relational algebra (TRA) which is endowed with temporal

1 Current address: Department of Mathematics and Statistics, Faculty of Humanities, Assumption University of Thailand, Bangkok 10204
operators that provide a close correspondence to temporal logic. There exists a few versions of such TRAs, [1,3,4,5,6]. Of all the temporal databases, the historical relational database model with discrete valid time representation has the simplest time structure. Valid time more closely represents changes in the real world being modeled.

Moreover, such a database could be analysed in the framework of the well-developed discipline of time series analysis [2]. Modelling temporal semantics that could facilitate some statistical data analysis has been studied by some authors including Sarda [6,7], Segev and Shoshani [8], Snodgrass et. al. [10], Navathe and Ahmed [3] and Orgun [4].

The paper is organized as follows. We first discuss the temporal structure we employ in our model. We next summarize the HRDB model studied by Sarda [6,7] and consider Time Normal Form [3] that ensures the consistency of time-invariant and time-variant relations. We then discuss Time Sequence Collections [8] that provide a central role in analyzing historical data in our model. We also outline an extension of SQL (TSQL with an inner product operator) which can be used to perform simple data analysis on scalar time sequences and time series [15]. We conclude the paper with a brief discussion.

2 Temporal Structure

To endow a temporal algebra with an inner product operator in a manner independent of the actual representation of temporal data, we use a temporal structure that can be described by any discrete linear bounded model [1,14]. Such a structure is very common in the real world time sequence data. However, if our interest goes beyond the time for which the data are stored as in the case of forecasting we could adopt an unbounded discrete linear model for the time domain, as studied in [11].

Thus, time is considered isomorphic to the natural numbers and the set of time we consider is a linear order; i.e., given two instants of time \( t_1 \) and \( t_2 \), we have \( t_1 = t_2 \), \( t_1 < t_2 \) or \( t_1 > t_2 \). The "current" moment in time refers to the latest clock tick and is referred to as NOW. NOW can be considered a moving variable, which changes its value with time.

Thus in a discrete linear bounded time, conceptually a historical relational database (HRDB) consists of a sequence of relational database instances,

\[
D_H = \{D_t : t \in T\} \quad \text{where} \quad T = \{0, 1, 2, \ldots, n\}
\]
where the index $t$ denotes the time associated with a particular database instance.

Here $D_t$ means the database instance at time $t$ and each $D_t$ has the same schema as the database $D_H$. $D_H$ could be seen as a collection of time-varying relations [1,4]. In our discussion, we will include time as a relation, or as time stamps in tuples of the relation.

3 Time Representation and Operations in HRDB

A valid schema $R_1 = \{A_1, ..., A_n \mid T\}$ is a finite set of explicit attribute names and a timestamp attribute $T$ representing valid time. Corresponding to each attribute name $A_i$, $1 \leq i \leq n$, we have a domain set $D_i$. We thus define $D = D_1 \times D_2 \times ... \times D_n$. If $T_v$ represents the set of all valid times, then the domain of $T$ is the power set $P(T_v)$.

We express the explicit attributes of $R_1$ by $R = \{A_1, ..., A_n\}$. A relation $r$ on schema $R$, denoted by $r(R)$ is a finite set of mappings $x_1, x_2, ..., x_k$ from $A_1, A_2, ..., A_n$ to $D$, where associated with each $x_i$, $1 \leq i \leq k$, is a nonempty time stamp attribute $t_i \in P(T_v)$. Hence $r = \{x_1 \mid t_1, x_2 \mid t_2, ..., x_k \mid t_k\}$. As in the snapshot model, tuples with identical explicit attribute values, so called value-equivalent tuples, are disallowed [11].

Consider a relation schema EMP (Name, Posn, Mgr, Sal, T) with time granularity month. An instance of the relation is given in table 1.

<table>
<thead>
<tr>
<th>Name</th>
<th>Posn</th>
<th>Mgr</th>
<th>Sal</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Jr.Engg</td>
<td>Bill</td>
<td>30K</td>
<td>{10, ..., 15, 20, ..., 25}</td>
</tr>
<tr>
<td>John</td>
<td>Jr.Engg</td>
<td>Al</td>
<td>35K</td>
<td>{26, ..., 44}</td>
</tr>
<tr>
<td>Jack</td>
<td>Jr.Engg</td>
<td>Al</td>
<td>25K</td>
<td>{35, ..., 45}</td>
</tr>
</tbody>
</table>

For a historical relation $R$, the attribute $K$ is a key if any time instant $t$ and for a given value $k$ of $K$, there is only one tuple in $R$ containing $k$ and whose interval contains $t$. 
Thus, if r and s are tuples in R and R satisfies certain integrity constraints and r.K = s.K, then intervals in r and s must be adjacent. This definition of a key basically indicates that its values are time unique (and not tuple unique as in the relational model).

Primitive operations of the standard relational model (Union (∪), set difference (\(\setminus\)), Cartesian product (×), projection (Π) and selection (σ)) are carried over to the historical relations and additional and more useful operators such as join are accordingly defined. If X represents a set of (time-varying) attributes, then a historical relation in the standard relation model is R (X, T)

And we can apply the relational operators to R with the usual meaning but the result may not be a historical relation. But our main interest is especially projection \(Π_Y(R)\) - project R on Y - where the result may or may not be a historical relation depending on whether Y includes the time attributes T. HRDB and associated algebra together with query language HSQL have been extensively studied by Sarda [6,7].

4 A Variant of Historical Relational Model and Time Normal Form

In the model thus far considered, the timestamps associated with each tuple is a set of chronons rather than contiguous sequence of time. We could replace T with time-start (\(T_s\)) and Time-end (\(T_e\)) pair for each interval in T and rearrange the tuples according to the temporal ordering. In the new format of the relation we have to use either \(T_e\) or \(T_s\) as part of the key. Such a historical relation has been proposed and studied by Navathe and Ahmed [2] under the name Temporal Relational Model (TRM). Employee (EMP) relation is now expressed in TRM format in Table 2.

<table>
<thead>
<tr>
<th>Name</th>
<th>Posn</th>
<th>Mgr</th>
<th>Sal</th>
<th>(T_s)</th>
<th>(T_e)</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Jr.Engg</td>
<td>Bill</td>
<td>30K</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>John</td>
<td>Jr.Engg</td>
<td>Bill</td>
<td>30K</td>
<td>20</td>
<td>25</td>
</tr>
<tr>
<td>John</td>
<td>Engg</td>
<td>Al</td>
<td>35K</td>
<td>26</td>
<td>44</td>
</tr>
<tr>
<td>Jack</td>
<td>Apprentice</td>
<td>Al</td>
<td>25K</td>
<td>35</td>
<td>45</td>
</tr>
</tbody>
</table>

Table 2. Employee (EMP) Relation in TRM format
Then we could define valid-time state operators as reported in [3].

The following additional constructs of TSQL are then available that will facilitate statistical data analysis [3,8].

- Conditional expression using the WHEN clause.
- Retrieval of time stamp values with or without computation.
- Retrieval of temporally ordered information.
- Specification of time domain using TIME-SLICE clause.
- Modified aggregate functions, and the GROUP-BY clause and the GROUP-TO clause.
- COMPOSITION of related data in two time sequences that occur as attribute values in a relation.

With this background in historical relational database, we now consider two fundamental concepts: Temporal Relational Model in Normal Form (Navathe & Ahmed [3]) and Time Sequence Collection (Segev and Shoshani [8])

5 Temporal Relational Model (TRM) in Time Normal Form

Temporal relational model in time normal form ensures the consistence of time invariant (static) and time-varying relations (TVR). Thus a relation schema for a TRM has the form R(A_1,...,A_n, T_s, T_e) and the attribute is not allowed to have multiple values at a particular instant of time. A time invariant key together with T_s, i.e (TIK,T_s) is usually designated as the primary key. Here TIK stands for "Time-Invariant Key". Each tuple of a TVR has a precise T_s value, but T_e may not be known when the tuple is created for which there is provided the following default value.

\[ T_e := \text{NOW, if } T_s < \text{NOW} \]
\[ T_e := \text{NOW, if } T_s > \text{NOW} \]

If a tuple in a time varying relation refers to a time point, then we have a degenerate interval where T_e = T_s will represent a point event.

TRM adopts tuple time stamping, although it can successfully capture the independent behavior of time-varying attributes by recognising asynchronism (temporal nonalignment) among attributes and incorporating the notion of time normalisation. In a way, TRM can be considered to support a variation of attribute time stamping. In fact, an attribute time-stamped relation could be transformed to tuple time stamping.
There are situations where there are no attributes as such to be time stamped. For example, Course-Offered (Course#, Instructor#, T_s, T_e) represents the history of courses offered in a department, but neither course number nor instructor number is a time-varying attribute.

A set of time-varying attributes in a given relation of TRM model is termed synchronous if every time varying attribute is uniformly associated and bound together as one whole with the time stamp values of the tuple. Thus, in the Employee relation of Table 2, promotion and salary raise occur together forming time-wise properly aligned or synchronous attributes. Synchronism could also arise because the time varying attributes collectively describe a certain event. Such a synchronism could occur in the house repair relation as given in Table 3.

**Table 3. House Repair Relation**

<table>
<thead>
<tr>
<th>Job No.</th>
<th>Type</th>
<th>Condition</th>
<th>Place</th>
<th>Cost</th>
<th>T_s</th>
<th>T_e</th>
</tr>
</thead>
<tbody>
<tr>
<td>52</td>
<td>roof</td>
<td>broken</td>
<td>Ryde</td>
<td>2000</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>70</td>
<td>drainage</td>
<td>blocked</td>
<td>Clyde</td>
<td>800</td>
<td>35</td>
<td>47</td>
</tr>
</tbody>
</table>

We could have situations where the attributes are not synchronous in time. Such attributes are said to be temporally dependent. A change in value of an attribute could occur in adjacent intervals while the values of the other attributes remain the same. Temporal dependency among attributes could lead to anomaly and discrepancy in the results of a query [3,12].

A relation in time normal form (TNF) means that it is in BCNF and there exists no temporal dependencies among nonkey attributes. It is always possible to decompose a relation, if a temporal dependency exists, into two or more time-normalised relations by appropriately partitioning the attributes and merging the relevant time intervals using set partitioning algorithms or compose operators that can perform the decomposition. The reverse process of getting back the original relations can be performed using Tnjoin operator [3].

Generally, the set of time varying attributes in a relation can be partitioned into a minimum number of subsets such that no two attributes within one subset are temporally dependent on each other. These subsets of attributes with TIK and Time start (T_s) form the relations in Time Normal Form.
In the Employee (EMP) relation (Table 4), attributes Posn, Sal and Mgr are asynchronous. We can therefore decompose the relation into three time-normalised forms as given in Table 5, Table 6 and Table 7.

**Table 4. Employee (EMP) Relation**

<table>
<thead>
<tr>
<th>Name</th>
<th>Posn</th>
<th>Mgr</th>
<th>Sal</th>
<th>Tₛ</th>
<th>Te</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Jr.Engg</td>
<td>Bill</td>
<td>30K</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>John</td>
<td>Jr.Engg</td>
<td>Bill</td>
<td>30K</td>
<td>20</td>
<td>25</td>
</tr>
<tr>
<td>John</td>
<td>Jr.Engg</td>
<td>Bill</td>
<td>32K</td>
<td>26</td>
<td>44</td>
</tr>
<tr>
<td>John</td>
<td>Engg</td>
<td>Al</td>
<td>35K</td>
<td>45</td>
<td>50</td>
</tr>
<tr>
<td>John</td>
<td>Engg</td>
<td>Al</td>
<td>37K</td>
<td>51</td>
<td>55</td>
</tr>
<tr>
<td>John</td>
<td>Engg</td>
<td>Bill</td>
<td>40K</td>
<td>56</td>
<td>58</td>
</tr>
<tr>
<td>John</td>
<td>Engg</td>
<td>Bill</td>
<td>45K</td>
<td>59</td>
<td>Now</td>
</tr>
<tr>
<td>Jack</td>
<td>Apprentice</td>
<td>George</td>
<td>25K</td>
<td>12</td>
<td>17</td>
</tr>
<tr>
<td>Jack</td>
<td>Apprentice</td>
<td>Al</td>
<td>25K</td>
<td>18</td>
<td>Now</td>
</tr>
</tbody>
</table>

**Table 5. Manager Relation**

<table>
<thead>
<tr>
<th>Name</th>
<th>Mgr</th>
<th>Tₛ</th>
<th>Te</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Bill</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>John</td>
<td>Bill</td>
<td>20</td>
<td>44</td>
</tr>
<tr>
<td>John</td>
<td>Al</td>
<td>45</td>
<td>55</td>
</tr>
<tr>
<td>John</td>
<td>Bill</td>
<td>56</td>
<td>Now</td>
</tr>
<tr>
<td>Jack</td>
<td>George</td>
<td>12</td>
<td>Now</td>
</tr>
</tbody>
</table>
Table 6. Position Relation

<table>
<thead>
<tr>
<th>Name</th>
<th>Posn</th>
<th>T_s</th>
<th>T_e</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Jr.Engg</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>John</td>
<td>Jr.Engg</td>
<td>20</td>
<td>44</td>
</tr>
<tr>
<td>John</td>
<td>Engg</td>
<td>45</td>
<td>Now</td>
</tr>
<tr>
<td>Jack</td>
<td>Apprentice</td>
<td>12</td>
<td>Now</td>
</tr>
</tbody>
</table>

Table 7. Salary Relation

<table>
<thead>
<tr>
<th>Name</th>
<th>Sal</th>
<th>T_s</th>
<th>T_e</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>30K</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>John</td>
<td>30K</td>
<td>20</td>
<td>25</td>
</tr>
<tr>
<td>John</td>
<td>32K</td>
<td>26</td>
<td>44</td>
</tr>
<tr>
<td>John</td>
<td>35K</td>
<td>45</td>
<td>50</td>
</tr>
<tr>
<td>John</td>
<td>37K</td>
<td>51</td>
<td>55</td>
</tr>
<tr>
<td>John</td>
<td>40K</td>
<td>56</td>
<td>58</td>
</tr>
<tr>
<td>John</td>
<td>45K</td>
<td>59</td>
<td>Now</td>
</tr>
<tr>
<td>Jack</td>
<td>25K</td>
<td>12</td>
<td>Now</td>
</tr>
</tbody>
</table>

6 Time Sequence Collections

Another important concept that plays a central role in analysing historical data is time sequence collection (TSC) [8]. Time sequence collections are implicitly embedded in a historical relational database and they serve as a framework for data analysis using time series techniques. A time series is a special kind of a time sequence of which the data values are observed at equidistant time points.

TSC modelling is endowed with a time data structure and associated data manipulation and analysis techniques that could serve more than the simple data analysis methods we have considered in this study.
Basis concept is a temporal data value defined for some object (e.g., sale outlet) at a certain point in time (e.g., first week of January, 1988) for some attribute of that object (e.g., total revenue). Thus a temporal data value is a triplet \(<s, t, a>\), where \(s\) is a specified object, which is referred to as a surrogate for the object [8]. For a given \(s\), the temporal values are totally ordered in time, forming a time sequence (TS). Such time sequences can be addressed using operators that can be expressed in terms of values (such as "Total revenue greater than $100K") as well as in terms of time (such as "Total revenue for every Thursday"). It is more interesting to consider the collection of TSs for the objects of the same class (or type). For example, we can study performance history regarding revenues for all the sales outlets, serial pattern in performance of each sales outlet, their similarities and other interesting features. Such a collection of time sequences is termed a time sequence collection (TSC). It should be observed that for the data collected at equi-distant time points, or over successive time intervals of equal length, we have the classical time series.

In our discussion, we confine to a simple TSC, which has a single surrogate as its identifier and a single temporal attribute. If the attribute is a vector, then we have what is termed quasi-synchronous attribute in the TRM model.

Relationship between the three models (Historical Relational Database Model (HRDM) [6,7], Temporal Relational Model (TRM) [3], Time sequence Collection Model [8]) is best explained through their relational schema as in Table 8.

### Table 8. Summary of The Three Models

<table>
<thead>
<tr>
<th>Model</th>
<th>Schema</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>HRDM</td>
<td>( R(A_1, \ldots, A_n \mid T) ),</td>
<td>where (T) is a set of time points over which the attributes are valid.</td>
</tr>
<tr>
<td>TRM</td>
<td>( R(A_1, \ldots, A_n \mid T_s, T_e) )</td>
<td>where (T_s) and (T_e) represent the start time and the end time of the interval over which the attributes are valid.</td>
</tr>
<tr>
<td>Simple TSC</td>
<td>( R(S, A \mid T) )</td>
<td>where (T) is a sequence of contiguous time points</td>
</tr>
</tbody>
</table>
It can be observed that a TSC can be seen as TRM in time normal form and a TRM is a particular case of HRDM. A simple TSC thus inherits all the facilities available for statistical analysis of HRDM and TRM.

In the following, we assume that a time sequence or a time series is made available from a historical database.

In data analysis of a time series, there is a need to consider aggregation operations such as count, maximum and minimum regarding feature extraction and pattern recognition. Inner product plays a fundamental role in discovering the interdependence of values along the time series as well as among the time series (through auto-covariance and cross-covariance functions).

7 Analysis of Time Sequence Data

The modest objective of analysing any time sequence data (which is also referred to as time series if the data were collected at equi-distant time points) is to provide a concise description of the historical series [2]. Such a description may consist of a few summary statistics and perhaps one or more graphical representations of the data. The graphs facilitate how the observed data evolves over time.

Another interesting problem of time series analysis is to forecast future values of the series. Much of the analysis has been developed to this end and it has served well in many economic, management and environment applications. Time sequences are monitored on the basis of information provided by the historical data to detect changes in behaviour as they occur. From the past data, we could detect whether the situation is improving or deteriorating. Having an estimate of serial dependence from the historical data is an invaluable source of information about the structure of the data generating process.

Pragmatic aspect of time sequence data analysis is to extract the salient features of the sequence that would help us to attain the above objectives.

Simple descriptive method of analysis is time plot. The most basic graphical representation of a time sequence is

\[ \{y(t) : t = 1, 2, \ldots, n\}. \]

This is a time plot of the observed data values \( y(t) \) against the time of observations \( t \). Usually the display is enhanced in the simplest way by joining successive points by straight lines. Though an effective visual aid, it is not the best. Graphical output
commands such as TIME-PLOT and LINE-GRAH are very handy but a very effective facility for preliminary data analysis. It should be observed that by changing the scale of measurement of y values relative to the t-axis, we could heighten the asymmetry of the rise and fall within each cycle in a time sequence data. Shortening y-values can affect the interpretation.

Another widely used simple technique is to find the smoothed out function that can be fitted to the data using local averaging method. Here namely, smooth component \('s_t'\) and rough component \('r_t'\). Thus we have:

\[ y_t = s_t + r_t \]

The smooth component \('s_t'\) emphasises the major features of the data notably the seasonal pattern, while de-emphasising the apparently random fluctuations. But the residuals \(y_t - s_t\) re-emphasises the random aspects of the data without the distraction of the seasonal pattern.

Smoothing is fundamentally an exploratory operation, a means of gaining insight into data without precisely formulated models or hypothesis - a non-parametric method. The simplest of the smoothing operation is taking moving averages along the series.

A moving average of a time sequence \(\{y_t : t = 1, 2, \ldots, n\}\) is a time sequence \(\{s_t\}\), defined by

\[ s_t = \Sigma w_j y_{t+j} ; \quad t = p+1, p+2, \ldots, n-p \]

where \(p\) is a positive integer and the \(w_j\) are weights with \(\Sigma w_j = 1\). Usually each \(w_j\) is positive and \(w_j = w_{-j}\). \(2p+1\) is termed the order of the moving average. Since the definition leaves \(s_t\) undefined near the ends we could extend the definition by letting the summation range from \(j = \max (p, 1-t)\) to \(j = \min (p, n-t)\) and divide by the corresponding sum of the included weights. Usually, \(p\) is very much smaller than \(n\) and little is then lost by leaving the observations near the ends unsmoothed. The restriction to moving average of odd order is not necessary, but is imposed to preserve an unambiguous correspondence between \(y_t\) and \(s_t\). There are other smoothing techniques available such as polynomial smoothing, spline smoothing, etc., but moving average is the simplest to compute and it serves the purpose in extracting interesting features including trend and cycles.

To get the three-week moving average of wheat price time sequence, we could use the following query:
ACCUMULATE INTO AVG-PRICE AVG QUANTITY
FROM WHEAT PRICE
GROUP T-LAST 3

In the following we provide some statistical queries that could be asked in analysing a total revenue time series data using the facilities already available and the inner product operator that we will introduce.

Example 1. Find the weekly total revenue.

AGGREGATE INTO WEEKLY-REVENUE SUM QUANTITY
FROM TOTAL-REVENUE
GROUP T BY WEEK

For this query, the time hierarchy and time-unit keywords are known to the system.

Example 2. Find all weekly revenues of sales-outlet #9 after the first four weeks

SELECT INTO R1-SEQUENCE REVENUE
FROM OUTLET-REVENUE
WHERE S=9 AND T IN (5 TO END)

Similar question can be asked for the weekly total revenue before the last four weeks.

SELECT INTO R2-SEQUENCE REVENUE
FROM OUTLET-REVENUE
WHERE S=9 AND T IN (BEGIN TO N-4)

Here $N$ is the total length of the time series.

The next example requires the inner product operator to compose the two series of equal length in getting the answer.

Example 3. Find the auto-covariance of order 4 in the outlet-revenue data series.

FIND AUTOCOVARIANCE OF ORDER 4 QUANTITY
FROM OUTLET-REVENUE

The required autocovariance is obtained using inner product operator.
AUTOCOVARIANCE = INNER-PRODUCT R1-SEQUENCE
AND R2-SEQUENCE

We next explore finding interesting features of a fluctuating time sequence, including similar patterns and cycle of their occurrences along the time sequence in a simple manner. We may first code the local behaviour of the sequence by observing whether the time varying values are increasing, decreasing or stationary at each time point. A reasonable procedure to code these local features is the following:

**Step 1.** Find the local trend at each time point based on gradient at the point and code them as:
- **u** for (upward trend)
- **d** for (downward trend)
- **C** for (local maximum downturn-reversal)
- **D** for (local minimum upturn-reversal)

**Step 2.** Find the local maxima among the C’s and code them as A’s

**Step 3.** Find the local minima among the D’s and code them as B’s

A sequence of codes is thus generated. An example of such a code sequence is

\[
D \ d \ u \ u \ u \ C \ d \ D \ u \ A \ d \ B \ u \ u \ u \ u \ C \ C \ u \ u \ A \ d \ d \ B \ u \ u \ u \ C
\]

We can now perform search for similarity and cycle and other pattern search using available techniques. Using a string matching algorithm (Karp-Rabin) we identify sharpturn-folowed-by-downturn pattern in the above sequence as

"A d B" and "A d d B" at positions 10 and 22 respectively.

8 Conclusions

Relational normal forms played a fundamental role in relational database methodology. Temporal normal form, a concept that has been extended to the temporal relational model (TRM) has provided a means to express historical relational database model in a more manageable and understandable format that is
appropriate for statistical analysis of time sequence data. If attribute values were observed along the successive time points and recorded accordingly in order, then time sequence (TS) model is more attractive. If in addition, the time points are equidistant, then we have a time series data. In fact, time series are special time sequences which in turn are special temporal relation. A temporal relational model is a historical relational model in special format. Observing the relationship hierarchy, we can make use of the available facilities while performing data analysis on a particular model. In the present study we can explore only simple data analysis on a scalar time sequence and a scalar time series only. We plan to pursue our study on complex time sequences - constructed as well as constrained - both from the deterministic and stochastic point of view.

9 Acknowledgements

This work has been supported in part by an Australian Research Council (ARC) Grant and a Macquarie University Research Grant (MURG).

References