On Incomplete Data in A Fuzzy Temporal Database Model

Werasak Kurutach
Department of Computer Engineering
Mahanakorn University of Technology
51 Cheum-Sampan Rd., Nong Chok, Bangkok 10530, THAILAND
e-mail: werasak@mut.ac.th

ABSTRACT: This paper presents the concept of incomplete data arising in a fuzzy temporal database model and their approximation. Interactivity among the temporal attributes is important in the process of approximating the unknown values or reducing the space of incompleteness. In addition, the relationship between the current time and incomplete data has been investigated. Semantic constraints in an application domain can be used in enhancing such the relationship so that the incompleteness is less. This study is motivated by our contention that the understanding of these issues is important to evaluating queries in such a database model. However, to emphasise the point, only the temporal attributes are mainly discussed. Our approach is based on the theory of fuzzy set, and a fuzzy temporal data model has been presented as a basic framework for our discussion.

KEYWORDS: Incomplete data, Fuzzy Temporal Data Model, Fuzzy Time, Temporal Data, Database Systems

1. INTRODUCTION

Incomplete data have been an important issue in the database literature for around two decades [Codd (1979), Lipski (1979)]. Generally, there is no consensus on the meaning of the term “incomplete data”. It may mean either partially known values [Lipski (1979)] or missing values [Codd (1979)]. However, in this work, incomplete data will be used in the sense of missing values of attributes. In this paper, we will discuss issues on the incompleteness aspect of temporal data in an extended relational data model. These temporal data can be fuzzy values. This data model has been proposed by the author to allow a treatment of both time and fuzziness [Kurutach (1998)]. Until recently, temporal data models and fuzzy data models have been researched separately for several years [Petry (1996), Tansel (1993)]. During the last few years, researchers have paid their attention to methods of handling both temporal and fuzzy data [Kurutach (1995), De Tre (1997)]. However, most of them have concentrated on the issues of data models and query operators. It is our contention that incompleteness of fuzzy temporal data is also an important issue to enhance the expressive power of the model, and, additionally, being able to estimate its value will facilitate the capability of providing the meaningful and useful information from databases, especially when the notion of the current time (or “now” [Clifford (1997)]) is concerned.

The paper is organised as follows. Section 2 will provide the concepts of a discrete time model and fuzzy time. These notions will be used as a basis for the discussion in the sequel sections. Section 3 introduces the underlying data model that emphasises fuzziness in temporal data. In section 4, we discuss how incomplete data can arise in temporal knowledge and how to estimate their meanings. Section 5 shows the relationship between the value denoting the current time and incompleteness of data. It also indicates that a semantic constraint of an application can be used to reduce the space of incompleteness, but not be able to remove it. Finally, in the last section, we will conclude this work and discuss further investigation.

2. DISCRETE TIME MODEL AND FUZZY TIME: BACKGROUND CONCEPTS

In general, time is a continuous variable and can be represented by real numbers. However, most temporal database works have assumed the discrete time model because of its simplicity and relative ease of implementation. In such a model, the continuous time line is divided into nondecomposable and equivalent-length segments (called *chronons*). Each chronon is viewed as isomorphic to a natural number. Time in this sense is said to be *linearly ordered*. It is also assumed that a single granularity of time is used in defining the nondecomposable unit of time, and hence, a chronon can be denoted by an integer. To assume a discretization of the time axis implies a simplification of its computational handling and avoids the problem derived from differentiating between open and closed intervals.

A time point in the discrete time model can be specified by an integer which is corresponding to a chronon on the time axis. Hence, primarily, the notion of *a fuzzy number* [Kaufmann(1988)] can be employed in modeling a fuzzy time point as well as a fuzzy length of time. Basically, the length (or duration) of time can be regarded as the numbers of chronons from the starting time point to the ending time point inclusive. For computational efficiency and simplicity reasons, *a trapezoidal function* will be used in defining the grade of membership of a fuzzy number. It can be defined as follow.

$$T(x:a,b,c,d)=\begin{cases}0 & \text{if } x<a \text{ or } x>d,\\ \dfrac{(x-a)}{(b-a)} & a\le x<b,\\ 1 & \text{if } b\le x\le c,\\ \dfrac{(d-x)}{(d-c)} & \text{if } c<x\le d,\end{cases}$$

Assume $M = T(x: a_m, b_m, c_m, d_m)$ and $N = T(x: a_n, b_n, c_n, d_n)$ be two fuzzy number. Then, two basic arithmetic operators can be defined as follows [4].

$$M + N = T(x: a_m+a_n, b_m+b_n, c_m+c_n, d_m+d_n)$$
$$M - N = T(x: a_n-d_m, b_n-c_m, c_n-b_m, d_n-a_m)$$

Also, note that $T(x: c, c, c, c)$ is a precise number c, and we will use both notations interchangeably.

3. A FUZZY TEMPORAL DATABASE MODEL

Basically, a fuzzy temporal database is defined as a collection of temporal relations (or tables) that can have fuzzy data, and each temporal relation contains tuples of the same types. To emphasise the point of discussion in this paper, a tuple will be viewed as composed of two main parts: *nontemporal* and *temporal* parts. Formally, given a temporal relation r, a tuple h in r can be defined by

$$h = \langle v, t \rangle$$

where $v \in dom(A_1) \times \ldots \times dom(A_n)$ ($dom(A_i)$, $1 \le i \le n$, is the domain of attribute A_i) and t is a *temporal specification* [Kurutach (1995)]. The set of n attributes $\{A_1, \ldots, A_n\}$ in the scheme of r are nontemporal ones. In general, the meaning of the tuple h is that the fact represented by the value v is valid in the real world during the time specified by the temporal specification t.

Primarily, a temporal specification defines a time interval which may be fuzzy itself. Conventionally, a time interval is represented by its starting time and its ending time, i.e. $[t_{start}, t_{end}]$, and its duration can be evaluated as $t_{end}-t_{start}+1$. However, when the feature of fuzziness is concerned, it has been argued that all three pieces of knowledge of time, the starting time, the ending time and the duration or the length of time, are required for a time interval representation [Kurutach (1995)]. As a consequence, a temporal specification t can be defined as

$$t \in dom(A_{start}) \times dom(A_{end}) \times dom(A_{lower}) \times dom(A_{upper}) \times dom(A_{version})$$

where A_{start} is the starting time attribute, A_{end} is the ending time attribute, A_{lower} is the lower bound attribute, A_{upper} is the upper bound attribute and $A_{version}$ is the reincarnation attribute[1]. The lower and upper bound attributes are used to restrict the duration of the time, and their values are the number of *chronons* in the time interval. The two attributes A_{start} and A_{end} can have the same domain which is the set of fuzzy numbers. Each of those fuzzy numbers is corresponding to a fuzzy set of smallest units defined on the time axis.

In the conventional data model, the concept of a key has been used to uniquely identify a tuple in a relation or, correspondingly, an object in the real world [Elmasri (1994)]. In what follows, the notion of a key in a fuzzy temporal data model will be defined so that this traditional property of the key is to be maintained.

Let $R = \{A_1, A_2, \ldots, A_n\} \cup \{A_{start}, A_{end}, A_{lower}, A_{upper}, A_{version}\}$ denote the schema of the relation r above. A set of attributes $K \subseteq \{A_1, A_2, \ldots, A_n\}$ is said to be *a time invariant key* if and only if

[1] The concept of this attribute will be made clear later.

$$\forall h, g \in r_x \text{ and } h \neq g, \qquad h[\mathrm{K}] \neq g[\mathrm{K}]$$

, where r_x is the time slid relation of r at a time point x. The value of K uniquely identify an object in the real world, but not a tuple in a relation r. Therefore, it cannot be a key of the relation. In fact, there could be more than one tuple in r that describe one object but at different states. Consequently, a key of the relation must consist of a time invariant key and a temporal attribute. When the temporal part is considered, it is obvious that neither of A_{start}, A_{end}, A_{lower} and A_{upper} can be a key attribute. This is because their values can be fuzzy. Fortunately, a value of the attribute $A_{version}$ of an object indicates the number of times that the object changed its state. In the other words, it will be updated or increased by 1 every time an attribute value of the object has been changed. The value of $A_{version}$ must be automatically generated by the system and cannot be modified by any user. Therefore, it is obvious that

$$\forall h, g \in r \text{ and } h \neq g, \qquad h[\mathrm{K}][A_{version}] \neq g[\mathrm{K}][A_{version}]$$

That means the set of attributes $\mathrm{K} \cup \{A_{version}\}$ can be defined as a key of the fuzzy temporal relation.

4. INCOMPLETENESS OF TEMPORAL DATA

Primarily, fuzziness can be represented in the structure of a temporal specification as discussed in the previous section. However, in the real world, some part of temporal knowledge (corresponding to some component of the temporal specification) may not be known. This unknown information results in what is called *incompleteness* of temporal knowledge. There are fifteen possible cases of incomplete data in a temporal specification. These cases arise in only four temporal attributes (i.e. A_{start}, A_{end}, A_{lower} and A_{upper}) in combination. Incompleteness cannot appear in the attribute $A_{version}$, because its value is generated by the system. However, in those cases, any unknown component can be estimated by using the knowledge of the known components. This estimation is important because it reduces the scope or the space of possible values of the unknown components. In what follow, each case will be shown with the symbol '*' to denote the unknown data of an attribute.

Case 1 $h = < v, <*, e, d_l, d_u, version\# >>$
In this case, the starting time point where the value v becomes valid is not known. However, the estimation of the unknown starting time point can be achieved by using the other known temporal attributes as follows

$$m_{h[A_{start}]}(x) = \sup_{x=y-z+1} \min(m_e(y), m_{d_1 \bar{\cup} d_2}(z))$$

where $m_{d_1 \bar{\cup} d_2}(z)$ is defined by

$$m_{d_1 \bar{\cup} d_2}(z) = \sup\{\min(m_{d_1 \cup d_2}(x), m_{d_1 \cup d_2}(y)) \mid x \le z \le y\}$$

That is, in general,

$$h[A_{start}] = h[A_{end}] - (h[A_{lower}] \bar{\cup} h[A_{upper}]) + 1$$

Case 2 h = < v, <s, *, d_l, d_u, version# >>
In this case, the ending time point where the value v becomes invalid. However, the estimation of the unknown ending time point can be achieved by using the other known temporal attributes as follows

$$m_{h[A_{end}]}(x) = \sup_{x=y+z-1} \min(m_s(y), m_{d_1 \bar{\cup} d_2}(z))$$

That is, in general,

$$h[A_{end}] = h[A_{start}] + (h[A_{lower}] \bar{\cup} h[A_{upper}]) - 1$$

Case 3 $h = < v, <s, e, *, d_u, version\# >>$

In this circumstance, the value of the lower bound attribute is not known. Generally, the difference between *s* and *e* is the duration which must lie between * and d_u. That means

$$* \leq (e - s + 1) \leq \mathrm{d}_u$$

Therefore, there are two extreme values, 1 and $(e - s + 1)$, that can be chosen as the estimated value of $h[\mathrm{A}_{lower}]$. If the latter is selected, the scope of incompleteness of the lower bound attribute is smaller. Hence, the value of the unknown lower bound attribute can be

$$h[\mathrm{A}_{lower}] = h[\mathrm{A}_{end}] - h[\mathrm{A}_{start}] + 1$$

Case 4 $h = < v, <s, e, d_l, *, version\# >>$
This is similar to Case 3, except that the two extreme values of $h[\mathrm{A}_{upper}]$ are $(e - s + 1)$ and u, where u is the upper end of the time axis. The former should be chosen for the same reason as in Case 3. Thus, the value of the unkown upper bound attribute can be

$$h[\mathrm{A}_{upper}] = h[\mathrm{A}_{end}] - h[\mathrm{A}_{start}] + 1$$

Case 5 $h = < v, <s, e, *_l, *_u, version\# >>$
When the values of the two bound attributes are unknown, they can be estimated with the same values as follows.

$$h[\mathrm{A}_{lower}] = h[\mathrm{A}_{upper}] = h[\mathrm{A}_{end}] - h[\mathrm{A}_{start}] + 1$$

Case 6 $h = < v, <*, *, d_l, d_u, version\# >>$
When we have only knowledge of the two bound attributes, the values of the starting time point and the ending time point can be evaluated by

$$h[\mathrm{A}_{start}] = T[x : 0,0,0,0] \overline{\cup} (T[x : u,u,u,u] - h[\mathrm{A}_{lower}] + 1)$$

$$h[\mathrm{A}_{end}] = (T[x : 0,0,0,0] + h[\mathrm{A}_{lower}] - 1) \overline{\cup} h[\mathrm{A}_{lower}]$$

Case 7 $h = < v, <*, e, *_l, d_u, version\# >>$
Basically, the value $(e–d_u)$ means the starting time point of the time interval with the ending time point *e* and the duration d_u. However, any time point cannot precede the starting time point of the time axis (i.e. 0). Therefore, the unknown value of attribute A_{start} can be evaluated as

$$^{m}h[\mathrm{A}_{start}](x) = \begin{cases} 0 & \text{if } x < 0, \\ ^{m}(h[\mathrm{A}_{end}] - h[\mathrm{A}_{upper}] + 1)(x) & \text{otherwise} \end{cases}$$

, and the unknown value of attribute A_{lower} is

$$h[\mathrm{A}_{lower}] = 1$$

Because of the space limitations, we will leave Case 8 $h = < v, <*, e, \mathrm{d}_l, *, version\# >>$, Case 9 $h = < v, <s, *, *, d_u, version\# >>$ and Case 10 $h = < v, <s, *, d_l, *, version\# >>$ here. They can be evaluated using a similar approach to Case 7.

Case 11 $h = < v, <s, *, *, *, version\# >>$

$$h[\mathrm{A}_{end}] = h[\mathrm{A}_{start}] \overline{\cup} T[x : u,u,u,u]$$

$$h[\mathrm{A}_{lower}] = 1 \text{ and } h[\mathrm{A}_{upper}] = T[x : u,u,u,u] - h[\mathrm{A}_{start}] + 1$$

Case 12 $h = < v, <*, e, *, *, version\# >>$

$$h[\mathrm{A}_{start}] = h[\mathrm{A}_{end}] \overline{\cup} T[x : 0,0,0,0]$$

$$h[\mathrm{A}_{lower}] = 1 \text{ and } h[\mathrm{A}_{upper}] = h[\mathrm{A}_{end}] - T[x : u,u,u,u] + 1$$

Case 13 $h = < v, <*, *, d_l, *, version\# >>$

$$h[\mathrm{A}_{start}] = T[x : 0,0,0,0] \overline{\cup} (T[x : u,u,u,u] - h[\mathrm{A}_{lower}]) + 1$$

$$h[\mathrm{A}_{end}] = T[x : u,u,u,u] \overline{\cup} (T[x : 0,0,0,0] + h[\mathrm{A}_{lower}] - 1)$$

$$h[\mathrm{A}_{lower}] = u$$

Case 14 $h = < v, <*, *, *, d_u, version\# >>$

$$h[\mathrm{A}_{start}] = h[\mathrm{A}_{end}] = T(x : 0,0,u,u)$$

$$h[\mathrm{A}_{lower}] = 1$$

Case 15 $h = < v, <*, *, *, *, version\# >>$
This is the case of total lack of temporal knowledge. Therefore, the space of incompleteness cannot be reduced. That is,

$$h[\mathrm{A}_{start}] = h[\mathrm{A}_{end}] = T(x : 0,0,u,u)$$

$$h[\mathrm{A}_{lower}] = 1 \text{ and } h[\mathrm{A}_{upper}] = u$$

5. SEMANTICS OF THE CURRENT TIME

The current time is a moving time. The value *now* has been employed to denote such a time in temporal databases, and it can be considered as a time variable [Clifford (1997)]. The validity of time-varying information sometimes depends on the current-time value. For example, assume that the current time is June 7, 1999 and the granularity of time is *day*. Then, a tuple

h = <<John, Manager, 50K>, <1 June 9, *now*, 7, 7, 1>>

means the time that John is a manager and has an income of 50K per annum starts from June 1, 1999 up until now. The value 1 of the last attribute shows that this is the first state of the object. That is, John started working as a manager with the salary 50K.
From the above, *now* is used to indicate that a fact is valid until the current time and *the future time of its validity is currently unknown*. In the other words, incomplete data are inherent in the notion of using *now* as a time variable denoting the current time. Another problem is that the lower and upper bounds of the time duration will encounter the difficulty of being updated when the ending time is moving. The problems can be treated into two situations.
Firstly, provided that the starting time *s* is certainly before the current time *now*, the fact can be simply represented as

h = <*v*, <s, *e*, *, *>>

and the ending time e is specified by

$$e = T(n: now + \Delta d, now + \Delta d, u, u)$$

where $s\text{-}now \leq \Delta d \leq u\text{-}now$. Δd is called an *offset span* whose semantics depending on applications [Jensen (1994)]. For example, in a database of the employment history, it may be a policy of the company that if an employee is either promoted or demoted, he/she must be noticed before its effective by 30 days. In this situation, the value of Δd is 30. Secondly, provided that the starting time s is not certainly before the present time, the ending time e is specified by

$$e = T(n: s + |\Delta d|, s + |\Delta d|, u, u)$$

For example, assume that today is August 15, 1999, if the company employs a new employee who will start working on September 1, 1999, then the temporal knowledge of the employment status of the new employee will be

$$t = \text{<1 Sept 1999, } e, *, *\text{>}$$

where e is a fuzzy time point characterized by

$$e = T(n: 1 \text{ Sept } 1999 + 14, 1 \text{ Sept } 1999 + 14, u, u)$$

6. CONCLUSION

In this work, we have proposed a fuzzy temporal data model based on a discrete time model. This data model has been employed as the framework to discussing issues on incomplete data in fuzzy temporal knowledge. We have shown that the space of incompleteness of data can be reduced by using interactivity among temporal attributes. Moreover, when the current time is concerned and the variable *now* is employed, incompleteness is inherent in its representation. Also, a semantic constraint of the application can be used to enhance such completeness.

Acknowledgement: This work has been supported by a grant from The Thailand Research Fund (PDF/50/2540).

REFERENCES.

Clifford J., Dyreson C., Isakowitz T., Jensen C. S. and Snodgrass R., 1997, “On the Semantics of “Now” in Databases”, ACM Transactions on Database Systems, Vol. 22, No. 2, pp. 171-214.

Codd E. F., 1979, “Extending the Database Relational Model to Capture More Meaning”, ACM Transactions on Databases, Vol. 4, No. 4, pp. 397-434.

De Tre G., De Caluwe R., Van der Cruyssen B., Van Gyseghem N., 1997, “Toward Temporal Fuzzy and Uncertain Object-Oriented Database Management Systems”, NAFIPS, pp. 63-67.

Jensen C. S. and Snodgrass R., 1994, Temporal Specialization and Generalization, IEEE Transactions on Knowledge and Data Engineering, Vol. 6, No. 6, pp. 954-974.

Kaufmann A. and Gupta M. M., 1988, “Introduction to Fuzzy Arithmetic”, Van Nostrand Reinhold Inc.

Kurutach W., 1995, “Modeling Fuzzy Interval-based Temporal Information: A Temporal Database Perspective”, The 4th IEEE Int. Conf. on Fuzzy Systems, Yokohama, Japan, pp. 741-748.

Kurutach W., 1998, “Handling Fuzziness in Temporal Databases”, IEEE Int. Conf. on SMC.

Lipski W., 1979, “On Semantic Issues Connected with Incomplete Information Databases”, ACM Transactions on Databases, Vol. 4, No. 3, pp. 262-296.

Elmasri R. and Navathe S. B., 1994, “Fundamentals of Database Systems”, The Benjamin/Cummings Publishing Company, Inc.

Petry F. E., 1996, “Fuzzy Databases: Principles and Applications”, Kluwer Academic Publishers.

Tansel A. U., Clifford J., Gadia S., Jajodia S., Segev A. and Snodgrass S., 1993, “Temporal Databases: Theory, Design and Implementation”, The Benjamin/Cummings Publishing Company.