Temporal Data Obsolescence: Modelling problems

Marcelo Finger Flávio S. Corrêa da Silva

Department of Computer Science
Instituto de Matemática e Estatística, Universidade de São Paulo
05508-900 – São Paulo (SP) – Brazil
mfinger@ime.usp.br fcs@ime.usp.br

Abstract

Data and knowledge bases model certain aspects of the world. When the state of the world changes faster than our ability to discover these state changes and update the data repositories accordingly, the confidence on the validity of data decays with time, and software systems running in such environments have to cope with the decay of confidence in the data lest they run the risk of giving wrong answers and behaving erroneously.

This gradual loss of confidence on stored data is termed information obsolescence*, and it is inherently a temporal phenomenon. We have come across such problem when designing an information system for traffic monitoring and control in a large city, and in this paper we propose to investigate some problems related to this modelling task. We propose two approaches to deal with information obsolescence, coined here the* analytical approach *and the* algebraic approach*, and show how both can converge to a general, temporal treatment of obsolescence. Our immediate goal is to try to approximate and reduce the whole problem of monitoring and controlling obsolescence of information to a purely temporal phenomenon, in which case we can model the system using conventional temporal database technology.*

1. Introduction

Databases and knowledge bases are built to model a certain part of the world. Updates in such data repositories are intended to reflect changes in the state of the modelled world. However, in situations in which the state of the world changes faster than our ability to find out such alterations, the confidence on the validity of data decays with time.

This gradual loss of confidence on data stored in a database or a knowledge base we call *information obsolescence*, and it is inherently a temporal phenomenon. Situations in which there is intense state change and this phenomenon becomes non-negligible abound. Software systems running in such environments have to cope with the decay of confidence in the data lest they run the risk of giving wrong answers and behaving erroneously.

We have come across such problem when designing an information system for traffic monitoring and control in the city of Sao Paulo, Brazil (population: 10 million[1]), project SIDAM. This system has to cope with information that changes dynamically and is distributed over the whole area of the city and its surroundings. Additionally, the rate of change of information varies significantly with location and along time (for example, data is far more alterable during peak hours in weekdays than at late night hours or during weekends).

Furthermore, there are fixed and mobile (helicopter and car based) sources of information, which may not be available at all times, and which operate in a fault-prone environment. This all puts information obsolescence under the spotlight when modelling such a system.

In this paper we propose to investigate some of the unavoidable problems one is faced with in such a modelling task. We propose two approaches to deal with information obsolescence, namely the *analytical approach* – in which functions on real numbers are adopted as basis for the description of the processes of information obsolescence – and the *algebraic approach* – in which discrete values with the least algebraic structuring strictly required to characterise time and uncertainty are employed as descriptive tools for information obsolescence – and we compare them and show how both can converge to a general, temporal treatment of

[1] Source: `http://www.seade.gov.br/english`

obsolescence.

Our immediate goal is to try to approximate and reduce the problem of monitoring and controlling obsolescence of information to a purely temporal phenomenon, in which case we can model the system using conventional temporal database technology. In this way, we hope to achieve a fast first solution to the data obsolescence problem. We also hope to make clear what the limitations inherent in such an approximation are. More complex developments will hopefully be achieved in the future.

In section 2 we introduce our analytical approach for treating obsolescence of information. In section 3 we present the algebraic approach for the same problem. Finally, in section 4 we present some further discussion and proposed future work.

2. The Analytical Approach

Suppose there is a helicopter flying over the city and sending information about the average speed of cars in the streets below. (In fact, there are two or three such helicopters in operation in Sao Paulo during peak hours, but instead of carrying measuring equipment, only a reporter is on board broadcasting traffic news.)

Suppose it is 17h20[2] in the afternoon and the helicopter measurement equipment sends some data reporting that in Happiness St the average traffic speed is 50km/h. The helicopter keeps flying and we do not know if and when it is coming back to Happiness St.

Initially, we have good reasons to trust that information, but its credibility decays fast, and five minutes later it may no longer be considered trustworthy.

The average traffic speed in Happiness St can be part of a *traffic monitoring database* to help traffic engineers manage car traffic in the city. "Happiness St" is a geographical entity in this database, "average traffic speed" is one attribute of that entity and "50km/h" is one possible value for this attribute. Clearly, this value is associated with a certain *instant in time* and bound to a certain *credibility degree*.

Also, this small example indicates that in one such database credibility is dependent upon time. More precisely, the credibility of data is monotonically non-increasing (and most often strictly monotonically decreasing) with time, once it is input to the database.

Under these considerations, we start by assuming that every piece of stored or derived data A in the database is associated with a *credibility degree* that changes with time t through a *credibility function* $\gamma(A, t)$. The only property we assume of such function is that it is monotonically non-increasing with time, that is,

$$t_1 \leq t_2 \Rightarrow \gamma(A, t_1) \geq \gamma(A, t_2)$$

We also assume there is a *credibility threshold*, γ_{th}, below which data A is no longer considered valid.

This assumption is going to be useful to release us from having to evaluate credibility degrees for each piece of information in the database, as we shall limit ourselves to identifying the moment in which the decreasing credibility of a piece of information reaches its threshold.

Only atomic data are updated, but when a query is posed to the database, we would like to derive the credibility degree associated to the answer, as well as information on how this credibility degree changes with time. Assuming that the credibility degree is a real value in the interval $[0, 1]$ and that queries are done in a first-order language, we can compute the credibility degree derived from complex queries e.g. in the following way:

$$\begin{gathered}\gamma(A \wedge B, t) = min(\,\gamma(A, t),\, \gamma(B, t)\,) \\ \gamma(A \vee B, t) = max(\,\gamma(A, t),\, \gamma(B, t)\,) \\ \gamma(\neg A, t) = 1 - \gamma(A, t) \\ \gamma(\forall x A(x), t) = min_y\{\gamma(A(y), t)\} \\ \gamma(\exists x A(x), t) = max_y\{\gamma(A(y), t)\}\end{gathered}$$

This is akin to Zadeh's fuzzy norm and conorm for fuzzy set operations, commonly used to characterise and to deal with vague data [DP88].

The rationale behind this definition is the following. The credibility in a conjunction cannot be greater than the credibility in each of the conjuncts, and it should not be smaller than the credibility in all conjuncts; similarly, the credibility in a disjunction cannot be smaller than the credibility in the disjuncts, and it should not be greater than the credibility in all disjuncts; quantifiers are treated as generalised conjunctions/disjunctions. As for negation, we take the view here that the smaller the credibility in some data, the greater the credibility in its negation. A "closed world assumption" in this setting would tell us that we have credibility 0 in any atomic data absent from the database.

These operations are also compatible with the generation of "coarse" constraints for belief measures, as implemented in [CdSRH94].

We have to face some immediate modelling problems that follow from this setting:

- We need credibility degrees, credibility functions and credibility thresholds. Such functions and values are domain dependent and we need domain

[2] it is notorious that traffic conditions may change drastically around this time.

experts to tell us what they are. However, for most domains, the experts do not usually deal with explicit evaluations for credibilities and decaying credibility functions, or pre-defined limiting values for credibility degrees on their data. Hence they may have difficulties to provide the system with these values and functions.

- Even for the simplest forms of decay, such as exponential and linear functions, which are supposed to be used as approximations for unknown domains, there are problems in their composition. For instance,
  - $min(k_A - m_A t, k_B - m_B t)$ is *not* of the form $k - mt$.
  - $min(k_A e^{-\beta_A t}, k_B e^{-\beta_B t})$ is *not* of the form $ke^{-\beta t}$.

Figure 1 shows a generic composition of two linear functions.

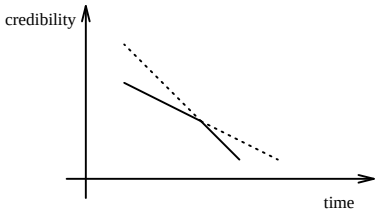

Figure 1 Min-composition of two linear functions

As a consequence, the computation of the temporal credibility degree derived from a query becomes quite costly. Instead of a fixed number of parameters for a given decay form (e.g. $\{k, m\}$ or $\{k, \beta\}$), the associated credibility function becomes a tree that grows in complexity proportionally to the complexity of the query. The leaves of such tree are the credibility functions associated to the atomic data. The intermediary nodes are of the form $min(Children)$ and $max(Children)$, where $Children$ represent the composed expression recursively computed by the subformulae of the query expression. The consequent overhead on storage space and computational efficiency is significant.

2.1. Homogeneous Decay

One way to try to solve the compositionality problem would be to postulate that the confidence in all data decays with the same "speed". With an exponential decay this would mean that for any data A and B:

$$\begin{aligned} \gamma(A,t) &= k_A e^{-\beta t} \\ \gamma(B,t) &= k_B e^{-\beta t} \end{aligned}$$

for the same value of β. The composition of $A \wedge B$ would give us

$$\gamma(A \wedge B, t) = min(k_A, k_B) e^{-\beta t}$$

which is still an exponential decay. Similarly for disjunction. However, computing

$$\gamma(A \wedge \neg B, t) = min(k_A e^{-\beta t}, 1 - k_B e^{-\beta t})$$

one sees it is not an exponential.

Exactly the same behaviour is obtained in the case of linear decay:

$$\begin{aligned} \gamma(A,t) &= k_A - mt \\ \gamma(B,t) &= k_B - mt \end{aligned}$$

for the same value of m. Note that these are parallel lines, so the composition of $A \wedge B$ is

$$\gamma(A \wedge B, t) = min(k_A, k_B) - mt$$

which is still linear. However,

$$\gamma(A \wedge \neg B, t) = min(k_A - mt, 1 - (k_B - mt))$$

is not linear. So not even this simplification solves the compositionality problem.

2.2. An Analytic Alternative

One could further argument that what is going wrong here is the way we are computing the composed credibility function. One could propose as an alternative the composition rules:

$$\begin{aligned} \gamma(A \wedge B, t) &= \gamma(A,t)\gamma(B,t) \\ \gamma(A \vee B, t) &= \gamma(A,t) + \gamma(B,t) \\ &\quad - \gamma(A,t)\gamma(B,t) \\ \gamma(\neg A, t) &= 1 - \gamma(A,t) \\ &\text{etc.} \end{aligned}$$

Apparently, in this way, the compositional problem would be solved, for if γ is a polynomial, so will be those compositions. Ignore for the moment the fact that the degree of the composed polynomial will be the sum of the degrees of the polynomials involved, thus adding to the system the burden of computing and storing such

polynomials; it will become clear that no coefficients need to be computed.

We show that such a formulation is very restrictive to the analytic modelling of data confidence. Classically, A is equivalent to $A \wedge A$, but this formulation would attribute γ to A and γ^2 to $A \wedge A$; by making $\gamma = \gamma^2$, we are forced to have $\gamma = 0$ or $\gamma = 1$.

Which means that to achieve compositionality in this way are limited to a binary set of confidence values. This restriction goes against the very idea of "decay" that we are trying to model here. This extreme simplification to the binary case may not be acceptable in the analytic approach, but a different approach may shed new light on it.

To avoid the problems of the analytical approach, we refine this modelling method with an algebraic approach.

3. The Algebraic Approach

Despite the fact that a large proportion of the "real-life" systems and applications that have been built employing or dealing with data and knowledge bases have to cope with reasoning with uncertainties and reasoning along and about time, research reports envisaging the interaction between time and uncertainties are not so frequently found in the literature (one interesting exception to this remark can be found e.g. in [KS92]).

One common and successful approach to deal with reasoning with/about uncertainties and with reasoning along/about time is the so-called "parallel approach" (see e.g. [Coh86] for a characterisation and a critical review of this approach for uncertain reasoning, and [Gab96] for an ambitious generalisation of this idea to a variety of applications). In this approach, given a language $\mathcal{L}$ (representing data and knowledge about that data) and a language $\mathcal{A}$ (characterising time relations, degrees of belief, etc.), connections between the operations in $\mathcal{A}$ and the operations in $\mathcal{L}$ can be defined, so that to each operation step in $\mathcal{L}$ corresponds a "parallel" operation step in $\mathcal{A}$.

It has been argued for elsewhere [CdSRH94] that for many applications it can be convenient to have more than one language $\mathcal{A}_i$ connected to a single system $\mathcal{L}$, to encompass different facets of a single problem. This argument certainly holds for problems in which time and uncertainties are relevant. As theoreticians, we tend to feel more comfortable with this approach to description of information. As system engineers, however, we would rather have the simplest possible languages to implement and the least amount of information to collect, and hence we would be happier to have a single language to connect to a system than to have many languages *and* therefore to have also to characterise the interconnection among these many languages.

We have studied the relations between degrees of credibility of pieces of information and the time periods in which they are valid (i.e. time intervals before credibility degrees reach the threshold), to investigate the possibilities of expressing degrees of credibility as time intervals.

The minimum requirements we have identified for a family of credibility values $\mathcal{A}_i$ is as follows:

- It shall contain more than one value, otherwise ascribing credibility degrees to pieces of information becomes useless; on the other hand, the set of values can – and in most cases should – be finite and relatively small, since it is rarely the case that more than a finite set of values is taken into account within any "real" model.
- The values shall be at least partially ordered, so that different credibility degrees can be compared with each other.
- Operations on credibility degrees shall be homomorphic with respect to operations on pieces of information in $\mathcal{L}$.

It is worth noticing that these requirements already induce a rather precise structuring for $\mathcal{A}_i$. For example, if $\mathcal{L}$ is classical first-order logic, then $\mathcal{A}_i$ are finite lattices. Notice also that the analytical functions proposed in the previous section abide by the last two of these requirements.

Now let us assume that each piece of information in $\mathcal{L}$ is tagged with a timestamp, based on a linear discrete time description. Let us also assume that credibility degrees degenerate monotonically with time, i.e. that each piece of information has an associated monotonically non-increasing credibility function depending on timestamps.

Two additional requirements are implicit in the analytical formulation of the problem and shall be required in the algebraic formulation too, namely:

1. that the result of operating on credibility degrees is always present in one of the operands, i.e. in our algebraic formulation that the operations on elements $a_1, ..., a_n$ of the finite lattice $\mathcal{A}_i$ occurring in an expression must result in one of $a_1, ..., a_n$; and
2. that the uniform query "is the credibility of $l_i \in \mathcal{L}$ above the threshold value a_{th}?" can be replaced by "was l_i collected recently enough to be still considered trustworthy?" with no loss of generalisation.

One minimal way to comply with these two additional requirements is to require that:

- Each lattice of credibility degrees $\mathcal{A}_i$ becomes *totally* ordered instead of just partially ordered.
- Either a single credibility function or a single credibility threshold is assumed for every piece of information in the whole system. In our model, we assume the former.

These requirements constrain significantly the expressive power of credibility degrees. Luckily enough, our prospective design experiments have indicated that for traffic monitoring and control in large cities this shall suffice. As a simple illustration, let us assume there are only two elements in our lattice: 1 (credibility above the threshold) and 0 (credibility below the threshold).

Compare this state of things with the analytical alternative in Section 2.2, in which a binary credibility space was imposed on us through the need to compose analytic functions. Apparently we have arrived at the same point through very different ways.

The result is that valid-time is the collection of times at which the data are above the threshold. Traffic information is then stored in the valid-time database as, for example:

Street Name	*Speed (km/h)*	*Valid-Time*
Happiness St.	50	17h18–17h25

Valid-time for the data are estimated at insertion time. Roughly speaking, we implement these ideas as an intermediate layer between the sensors and the database, which "translates" credibility degrees, functions and thresholds into valid-time intervals. This is illustrated in Figure 2, which presents the system's proposed abstract architecture such that:

- The information collectors (sensors) provide the input data to the preprocessing layer.
- The preprocessing layer converts information on data credibility into temporal information; once this layer updates the database, all information about credibility is lost.
- An application accesses only the temporal data. No information about credibility is available to the application.

In our example, data is collected at 17h20 and the resulting interval is 17h18-17h25. The past is updated, for it is assumed that the speed has been at the measured levels for some time before the measurement (in this case, 2 minutes before 17h20).

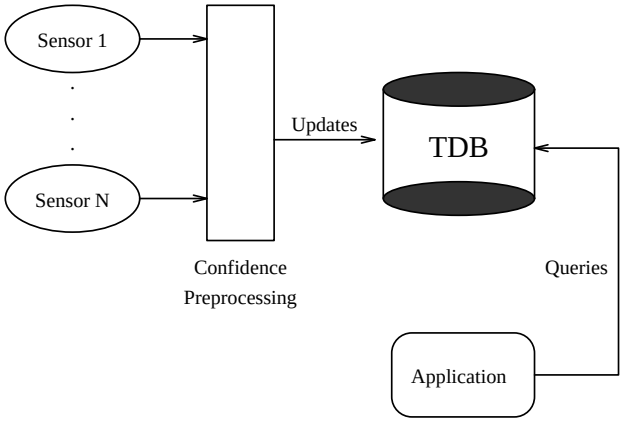

Figure 2 Abstract architecture of traffic monitoring system

Once the information goes through the layer between the sensors and the database and is added to the valid-time database, it is no longer possible to recover when the information was inserted (ie, its transaction time) and all we can refer to is the data valid-time $vt(A)$. Usual valid-time manipulations can be applied, and the usual treatment of valid-time databases can be employed [SA85, TCG+93, Sno95, Fin97]. With respect to the derivation of credibility intervals by queries, this amounts to the usual valid-time manipulation:

$$
\begin{aligned}
vt(A \wedge B) &= vt(A,t) \cap vt(B) \\
vt(A \vee B) &= vt(A,t) \cup vt(B) \\
vt(\neg A) &= complement(vt(A)) \\
vt(\forall x A(x)) &= \textstyle\bigcap_y \{vt(A(y))\} \\
vt(\exists x A(x)) &= \textstyle\bigcup_y \{vt(A(y))\}
\end{aligned}
$$

This is still not the end of our problems, for we have to decide what happens after 17h25 if no updates arrive. The fact that we do not know the average traffic speed at a given time does not mean that e.g. this speed is 0 km/h. We can either reject queries about such a time ("no reliable information available") or we can estimate such speed based on known past behaviour.

The first alternative involves only treatment of information based on available data – updated accordingly based on the model in use for obsolescence of information. The second alternative requires an additional model coupled to the general model, to update *information itself* based on changes of credibility degrees due to obsolescence. Although this seems an interesting feature to add to our general model, it clearly requires a deeper understanding of the problem domain that must be incorporated into the model. Since our major interest is in information modelling, we may leave this second alternative for future versions of the system.

4. Discussion and Future Work

Obsolescence of information is an important issue for modelling information systems in which reliability on data may change dynamically with time.

We have proposed two approaches to deal with information obsolescence, which we have coined the *analytical approach* and the *algebraic approach*. Both approaches are presented in a "minimalistic" fashion, in the sense that we are interested in identifying the bare essentials for modelling the obsolescence of information.

This research has been directed towards building a specific system, namely a system for traffic monitoring and control in large cities. Nevertheless, we have explored methodological issues in the most generic terms we have been capable of, so that our experiences and results can bring up guidelines to system designers facing the same problems. In particular, we have studied the possibilities of expressing obsolescence relations as time intervals, which can be useful to simplify the implementation of the presented models.

Admittedly, the solution sketched here is just a fast first approximation, and was designed to reduce a complex problem into a known domain. The merits of this solution are:

- it is easy to implement;
- it presents a *principled way* of reducing the problem of data obsolescence into a temporal database application.

Its main disadvantage is that in this process the precious information about data credibility becomes unavailable to other applications.

Clearly, there is still much to be done in this area. In future articles we shall discuss the applicability of some particular algebraic structures of credibility degrees and their impact on the implementation of particular systems.

Acknowledgments: this work was partially supported by FAPESP grant 93/0603-01, and CNPq grants 300041/93-4, 300597/95-9.

References

[CdSRH94] F. S. Correa da Silva, D. S. Robertson, and J. Hesketh. Automated Reasoning with Uncertainties. In M. Masuch and L. Pawlos, editors, *Knowledge Representation and Uncertainty – ch. 5 – a preliminary version was presented at the APLOC – Applied Logic Conference (Logic at Work), 1992*. Springer Verlag LNAI 808, 1994.

[Coh86] P. R. Cohen. Numeric and Symbolic Reasoning in Expert Systems. In *ECAI '86 - Proceedings of the 7th European Conference on Artificial Intelligence*, 1986.

[DP88] D. Dubois and H. Prade. An Introduction to Possibilistic and Fuzzy Logics. In P. Smets, A. Mamdani, D. Dubois, and H. Prade, editors, *Non-standard Logics for Automated Reasoning*. Academic Press, 1988.

[Fin97] M. Finger. A Logical Reconstruction of Temporal Databases. *Journal of Logic and Computation*, 1997. Accepted (1997) for publication; also available as Technical Report RT-MAC-9703[3].

[Gab96] D. M. Gabbay. *Labelled Deductive Systems - Volume I*. Oxford University Press, 1996.

[KS92] M. Kifer and V. S. Subrahmanian. Theory of Generalized Annotated Logic Programs and its Applications. *Journal of Logic Programming*, 12:335–367, 1992.

[SA85] R. Snodgrass and I. Ahn. A Taxonomy of Time in Databases. In *ACM SIGMOD International Conference on Management of Data*, pages 236–246, Austin, Texas, May 1985.

[Sno95] R. T. Snodgrass, editor. *The TSQL2 Temporal Query Language*. Kluwer Academic Publishers, 1995.

[TCG+93] A. Tansel, J. Clifford, S. Gadia, S. Jajodia, A. Segev, and R. Snodgrass, editors. *Temporal Databases: Theory, Design and Implementaition*. Database Systems and Application Series. Benjamin/Cummings Pub. Co., 1993.

[3] ftp://ftp.ime.usp.br/pub/mfinger/rtmac9703.ps.gz