Towards a Temporal Extension of Formal Concept Analysis

Rabih Neouchi, Ahmed Y. Tawfik, and Richard A. Frost

School of Computer Science, University of Windsor, Windsor,
Ontario N9B 3P4, Canada
{neouchi, atawfik, richard}@cs.uwindsor.ca

Abstract. This article presents a method for analyzing the evolution of concepts represented by concept lattices in a time stamped database, showing how the concepts that evolve with time induce a change in the concept lattice. The purpose of this work is to extend formal concept analysis to handle temporal properties and represent temporally evolving attributes.

1 Introduction

Formal concept analysis (FCA) [2, 7, 14] is a mathematical tool for analyzing data and formally representing conceptual knowledge. FCA helps forming conceptual structures from data. Such structures consist of units, which are formal abstractions of concepts of human thought allowing meaningful and comprehensible interpretation. FCA is a mathematical discipline whose features include:

- Visualizing inherent properties in data sets,
- Interactively exploring attributes of objects and their corresponding contexts, and
- Formally classifying systems based on relationships among objects and attributes through the concept of mathematical lattices.

FCA automatically generates hierarchies called concept lattices that characterize the relationships among objects and their attributes.

Concepts change and evolve with time. In fact, change seems to be constant in a continuously changing world. In many domains such as science, medicine, finance, population, and weather patterns, change is noticeable from one time to another. The extension of concepts (set of objects) and their intensions (set of related attributes) may change, affecting how the entities are related. As a consequence, the concept lattice characterizing the relationships among a set of entities (objects and attributes) evolves over time. Temporal concept lattice metamorphosis is the change in the concept lattice over time.

The rest of this section introduces some basic FCA definitions, terminology and states some assumptions. Section 2 presents temporal lattices and the evolution of concept structures in these lattices. Section 3 identifies the importance and applications of this work. In Section 4, the extension of FCA is developed and techniques for identifying temporal evolution patterns are presented. Section 5 discusses the problem of inferring temporal properties.

1.1 Terminology

Definition 1. A formal (or dyadic) context $\mathbb{K} = < G, M, I >$ is a triple consisting of two sets G and M and a relation I (also called dyadic relation) between G and M. The elements of G and M are called the formal objects and the formal attributes respectively. The relationship is written as gIm or $(g, m) \in I$ and is read as 'the formal object g has the formal attribute m'. A formal context can be represented by a cross table that has a row for each formal object g, a column for each formal attribute m and a cross in the row of g and the column of m if gIm.

Definition 2. Let $< G, M, I >$ be a context. The set of all formal attributes of a set $A \subseteq G$ of formal objects is denoted by A' and defined by: $A' = \{m \in M \mid gIm \text{ for all } g \in A\}$. Similarly, the set of all formal objects of a set $B \subseteq M$ of formal attributes is denoted by B' and defined by: $B' = \{g \in G \mid gIm \text{ for all } m \in B\}$. Note that if $A = \{\emptyset\}$ then $A' = A$. Similarly, if $B = \{\emptyset\}$ then $B' = B$.

Definition 3. A formal (or dyadic) concept c of the context $< G, M, I >$ is the pair (A, B) with $A \subseteq G$, $B \subseteq M$, $A' = B$ and $B' = A$. A is called the extent (denoted by Ext(c)) and B is called the intent (denoted by Int(c)) of the formal concept $c := (A, B)$.

Definition 4. Let $c_1 := (A_1, B_1)$ and $c_2 := (A_2, B_2)$ be two concepts of a formal context $< G, M, I >$. The formal concept c_1 is a formal subconcept of the formal concept c_2 if $Ext(c_1) \subseteq Ext(c_2)$ (or $A_1 \subseteq A_2$), which is equivalent to $Int(c_2) \subseteq Int(c_1)$ (or $B_2 \subseteq B_1$), and we write $c_1 \leq c_2$ (or $(A_1, B_1) \leq (A_2, B_2)$). In this case c_2 is a formal superconcept of c_1, and we write $c_2 \geq c_1$ (or $(A_2, B_2) \geq (A_1, B_1)$). It follows from this definition that a formal concept c_1 is formal subconcept of the formal concept c_2 if c_1 has fewer formal objects and more formal attributes than c_2. Similarly, a formal concept c_1 is formal superconcept of the formal concept c_2 if c_1 has more formal objects and fewer formal attributes than c_2.

Definition 5. The set of all formal concepts of the formal context $< G, M, I >$ is denoted by ß(G, M, I). The relation $\leq$ is a mathematical order relation called the hierarchical order (or simply order) of the formal concepts. It is also called formal conceptual ordering on ß(G, M, I). A concept lattice of the formal context $< G, M, I >$ is the set of all formal concepts of $< G, M, I >$ ordered in this way and is denoted by $\underline{ß}(G, M, I)$.

Definition 6. An ordered set (or partially ordered set) is a pair $(M, \leq)$, with $\leq$ being a set and $\leq$ an order relation on M.

1.2 Assumptions

In extending FCA over time, we prefer to index an object with a time variable, e.g. Obj_{ti}, rather than having time itself as an attribute for that object. The conventions we assume are as follows:

- Obj_{ti} represents object *Obj* at time t_i, where *t* is the time variable and *i* is an integer, and read as 'Object *Obj* at time *t i*'
- t_i precedes t_j if $i < j$ for any two integers *i* and *j*
- Given all the time intervals $t_{i \in n}$, where *i* is an integer and *n* is the number of observations made on a particular object over time, they form a partial order

To define the hidden evolution patterns that the concepts exhibit in a time stamped database, we assume the following:

- The database is fairly complete; that is, there is no missing stage in the evolution patterns for all individual objects.
- We start with the patterns that have more stages. Then we fit individual objects that do not show the complete evolution stages to these patterns using temporal matching [13].
- As we do not have a reliable way to distinguish among stages that could be missed and stages that have happened in reality but are missing in the dataset for some object, we assume here the latter. This assumption may be unreasonable in some domain and we propose treating such cases as defaults in nonmonotonic reasoning that should be retracted to resolve conflicts. Such conflicts may occur if persistence or causation properties are violated. As this may involve a complex knowledge engineering tasks, the knowledge engineer should critically examine the validity of this assumption.

2 Temporal Lattices

To handle the evolution phenomenon of concept structures and analyze temporal metamorphosis, a temporal extension of FCA is developed. This temporal extension involves the study of persistence, and other temporal properties implied by the data and concept lattices.

As an example, consider the simple database in Table 1, where some attributes change over time (juvenile, adult, senior) while others persist (dead). It can be easily seen from the concept lattice how the concepts manifest a change over different times (figure 1). Note that in our notation, *t* represents the time variable and t_i precedes t_j if $i < j$ for any two integers *i* and *j*. In addition, t_i and t_j form partial order.

To represent temporal evolutions in a concept lattice, we use two types of edges: temporal edges and non-temporal edges. The temporal edges allow the evolution of a particular object to be followed over time. A temporal precedence relation "<" is defined over time points. The direction of the arrow indicates this precedence. Non-temporal edges are undirected, as they are not governed by the temporal precedence. In fact, non-temporal edges describe a concept at a particular point in time.

The temporal concept lattice in Figure 1 shows that there are transient attribute and persistent ones. For example, juvenile is a transient attribute as a juvenile becomes an adult but dead is a persistent attribute.

Table 1. Human context database

Objects/Attributes	Juvenile	Adult	Senior	Dead	Male	Female
$Adam_{t1}$	X				X	
$Adam_{t2}$		X			X	
$Steve_{t1}$		X			X	
$Steve_{t2}$			X		X	
$Nancy_{t1}$			X			X
$Nancy_{t2}$				X		X
$Mary_{t1}$				X		X
$Mary_{t2}$				X		X

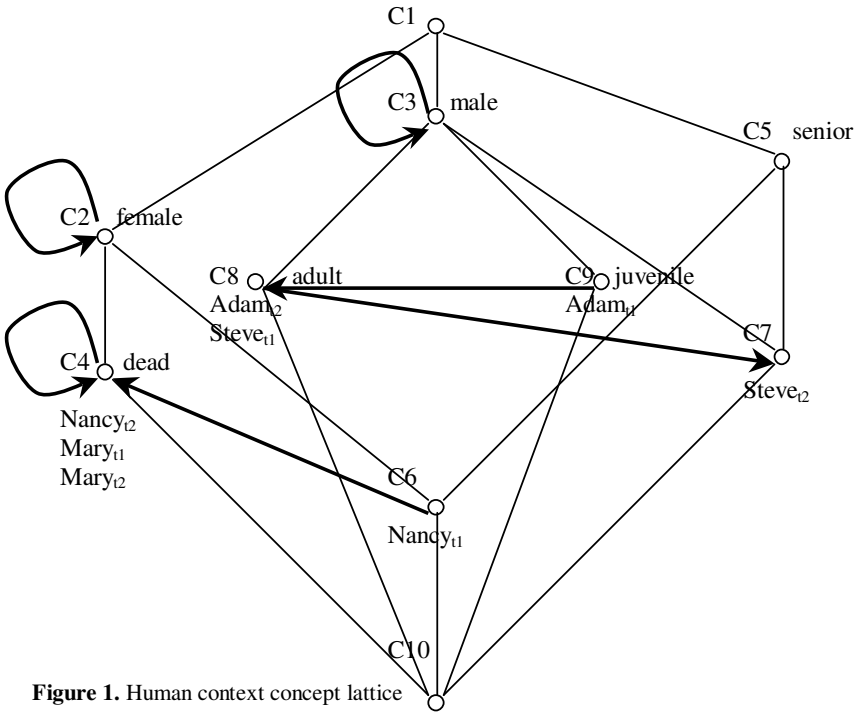

Figure 1. Human context concept lattice

3 Importance and Applications

Understanding concept evolutions can be useful in many applications including data mining, planning, and decision support systems. The application that motivated this work in particular is identifying the evolution of a set of concepts and related vocabulary for speech recognition in the Speech Web [6] as a conversation evolves.

In discovering useful patterns from a database researchers are increasingly relying on data visualization to complement data mining in the knowledge discovery process. Visualization helps developing insights and deduces the hidden regularities in the data. Animation seems to provide proper visualization for temporal evolution. However such animations can be easily generated from the proposed lattices. Moreover, the concept hierarchies provide a new tool for the study of the relationships between an interval and its subintervals.

FCA has been applied to static domains such as assessing the modular structure of legacy code [11] in software engineering, text analysis from different sources [4] in information retrieval, designing and exploring conceptual hierarchies of conceptual information systems [12] in knowledge acquisition, transforming object class hierarchies into normalized forms [10] in databases, performing structure-activity relationships [1] in environmental/chemical applications, structuring the design interface of some educational applications [5], analyzing individual preference judgments [9] in decision making tasks, implementing the TOSCANA system [8] in natural-language, improving the accuracy of speech recognizers [15] in speech recognition, and using medical-discharge summaries as training data sets [3] in medicine.

Adding further notations to FCA's representation is important to extend it over time. The suggested extension opens up new areas of applications to FCA such as representing the course of infection for a disease, the life cycle of a software project, the evolution of social, economic, and population trends.

4 Classification of Temporal Patterns

4.1 Unconditional Evolution Patterns

Unconditional patterns are any kind of a change in the attributes of an object over time that always happens in one unique direction. For example, during the process of getting older in humans, a person always evolves from being a child to an adult and then to a senior.

4.2 Conditional Evolution Patterns

Conditional patterns are any kind of change in the attributes of an object over time that might happen in more than one direction depending on a certain condition that controls the order of the changing attributes. For example, depending on the external temperature, a piece of ice might evolve from being solid to liquid or vice versa. Note that both conditional and unconditional patterns are forms of evolution and that some unconditional patterns become conditional once the data set has enough information about causes and effects.

4.3 Persistence

Persistence is a state in which an object maintains its attributes throughout its lifetime without any change in these attributes. For example, once a person is born male, he will always remain male. We say that gender is a persistent property.

4.4 Transitions

As opposed to persistence, transitions are any kind of change in the attributes of an object over time that does not follow any specific direction. For example, a person might eat then sleep, or sleep then eat. In this case, we say that eating and sleeping are transient properties. Note that transitions do not constitute evolution patterns for our purpose here.

5 Inferring Temporal Properties

Let B be the set of formal attributes, and b_j be the attribute number j, where i and j are integers.

Definition 7. The intension of an object *Obj* at time t_i is the set of all attributes of that particular object at time t_i. It is written as:

$$i\{Obj_{ti}\} = \{b_j \in B\} . \quad (1)$$

Definition 8. An evolution of an object, *Ev(Obj)*, is an ordered set of sets containing all the intensions that this particular object from an arbitrary initial time t_0 to a certain time t_i. The intension of the object at time t_{i+1} is added to the resulting evolution set when we consider the evolution up to time t_{i+1}:

$$\forall\, t<t_i,\ i\{Obj_{ti}\} \in Ev_{t0\text{-}ti}(Obj) . \quad (2)$$

For example, according to Table 1 above, we describe the evolution of *Adam* to be:

$$Ev(Adam) = \{\{male, juvenile\}, \{male, adult\}\} . \quad (3)$$

Definition 9. The evolution interval (the subscript of the evolution) of an object, $Ev_{t0\text{-}ti}(Obj)$, specifies an interval from time t_0 to time t_i, where t_0 is the time at which the observation of the object started and t_i is the time at which the observation ended. In other words, the evolution depends on the interval we allow for the object to change. For example, according to Table 1, we describe the evolution of *Adam* at time t_1 to be:

$$\mathrm{Ev}_{t0\text{-}t1}(\mathrm{Adam}) = \{\{\mathrm{male}, \mathrm{juvenile}\}\} . \quad (4)$$

Definition 10. An evolution pattern of an object, *EvPat(Obj)*, is an ordered set containing the elements of the conditional or unconditional patterns that this particular object might exhibit throughout its lifetime. For example, according to Table 1, we describe the evolution pattern of *Adam* to be:

$$\mathrm{EvPat}(\mathrm{Adam}) = \{\mathrm{juvenile}, \mathrm{adult}\} . \quad (5)$$

One way to determine the evolution pattern is to examine the intensions of a particular object over different points in time. For example, suppose that the intension of *Ann* includes at time t_1 the *child* attribute, at time t_2 the *adult* attribute, and at time t_3 the *senior* attribute, such that $t_1 \prec t_2 \prec t_3$ and $\prec$ is a precedence relation that defines a partial order. That is

$$\mathrm{i}\{\mathrm{Ann}_{t1}\} \supset \{\mathrm{child}\} . \quad (6)$$

$$\mathrm{i}\{\mathrm{Ann}_{t2}\} \supset \{\mathrm{adult}\} .$$

$$\mathrm{i}\{\mathrm{Ann}_{t3}\} \supset \{\mathrm{senior}\} .$$

Then an evolution pattern from *child* to *adult* to *senior* exists. Usually we are interested in evolution patterns that are consistent for all the objects or for a class of objects. For example, the pattern of evolution from child to adult to senior is applicable to any object *x* of type p*erson* in the database. This evolution is written as:

$$\forall\, \mathrm{person}(\mathrm{x} \in \mathrm{G}), \exists\, \mathrm{EvPat}(\mathrm{x}) = \{\mathrm{child}, \mathrm{adult}, \mathrm{senior}\} . \quad (7)$$

To determine if such a pattern holds, we form a temporal matching problem [13]. Temporal matching can be formulated as a special constraint satisfaction problem that tries to find a consistent assignment of states such as child, adult, and senior to individuals whose state is only known at particular points in time. This assignment has to be consistent with temporal constraints representing the persistence properties of each state.

Definition 11. A persistent property of an object, *Persist(Obj)*, is a set containing all that object's attributes that persisted in all of the intension sets of that object over time, including its evolution set if that object happened to have an evolution. For example, according to Table 1, we describe the persistent property of *Adam* to be:

$$\mathrm{Persist}(\mathrm{Adam}) = \{\mathrm{male}\} . \quad (8)$$

To determine the persistent property, we examine the intensions of a particular object over different points in time. For example, if the intension of *John* includes at

times t_1, t_2 and t_3 the *male* attribute then *male* is a persistent property of *John* in particular and to all objects *x* of type p*erson* in the database, such that *x* has the *male* property:

$$i\{John_{t1}\} \supset \{male\} . \quad (9)$$

$$i\{John_{t2}\} \supset \{male\} .$$

$$i\{John_{t3}\} \supset \{male\} .$$

$$\forall t\ (person(x \in G) \wedge male(x)),\ i\{x\} \supset \{male\} . \quad (10)$$

Similarly, the *female* property is persistent to all objects *y* of type p*erson* in the database, such that *y* has the *female* property:

$$\forall\ t(person(x \in G) \wedge female(x)),\ i\{x\} \supset \{female\} . \quad (11)$$

Definition 12. A transient property of an object, *Transient(Obj)*, is a set containing all that object's attributes that exhibited a change in all of the intension sets of that object over time, including its evolution set if that object happened to have an evolution. For example, according to Table 1, we describe the transient property of *Adam* to be:

$$Transient(Adam) = \{juvenile, adult\} . \quad (12)$$

Note that a change that happens in any order is not considered to be an evolution pattern. For example, suppose that the intension of *Bob* includes at time t_1 the *eating* attribute, at time t_2 the *sleeping* attribute while the intension of *Sam* includes at time t_1 the *sleeping* attribute, at time t_2 the eating attribute. That is:

$$i\{Bob_{t1}\} \supset \{eating\} . \quad (13)$$

$$i\{Bob_{t2}\} \supset \{sleeping\} .$$

$$i\{Sam_{t1}\} \supset \{sleeping\} .$$

$$i\{Sam_{t2}\} \supset \{eating\} .$$

Then no evolution pattern from *eating* to *sleeping*, or from *sleeping* to *eating* exists, and that this rule is applicable to any object *x* of type p*erson* in the database.

$$\forall\ person(x \in G),\ !\exists\ EvPat(x) = \{eating, sleeping\} . \quad (14)$$

$$\forall\ person(x \in G),\ !\exists\ EvPat(x) = \{sleeping, eating\} .$$

Note also that the transient property set of an object is not always identical to its evolution pattern set. For example in the case we just mentioned about *Bob* and *Sam*, the evolution pattern set of both *Bob* and *Sam* would be empty, while it is not the case for the transient property set of either one:

$$\text{EvPat(Bob)} = \{\varnothing\} . \quad (15)$$

$$\text{EvPat(Sam)} = \{\varnothing\} .$$

$$\text{Transient(Bob)} = \{\text{eating, sleeping}\} .$$

$$\text{Transient(Sam)} = \{\text{sleeping, eating}\} .$$

Note that the temporal matching exercise detects such transients as failure to match.

6 Conclusion

In artificial intelligence, there is a great need to represent temporal knowledge and to reason about models that capture change over time. Since changing the time variable causes a change in the intension (attributes) of concepts, we end up having different concept lattices representing the same context at different times, which we call a temporal concept lattice metamorphosis. Finding the order of the changing attributes therefore defines the evolution itself in the concept lattice.

References

1. Bartel, H.-G., Bruggemann, R. 1998. Application of Formal Concept Analysis to Structure-Activity Relationships. Fresenius Journal of Analytical Chemistry (5 Jan.), vol. 361, no. 1. Springer-Verlag, Berlin, Germany, 23-28
2. Birkhoff, G. 1948. Lattice Theory. American Mathematical Society Colloquium Publications, Vol. XXV, Second (Revised) Edition. Providence, Rhode Island, Library of Congress Catalog Card Number 66-23707
3. Cole, R., Eklund, P. W. 1999. Scalability in Formal Concept Analysis. Computational Intelligence (Feb.), vol. 15, no. 1. Blackwell Publishers, USA, 11-27
4. Eklund, P. W., Wille, R. 1998. A Multimodal Approach to Term Extraction Using a Rhetorical Structure Theory Tagger and Formal Concept Analysis. In Proceedings of the 2nd International Conference on Multi-modal Communication, CMC/98, (Tilburg). 171-175, ISBN 90-9011386-X
5. Fernandez-Manjon, B., Fernandez-Valmayor, A. 1998. Building educational tools based on formal concept analysis. Education and Information Technologies (Dec.), vol. 3, no. 3-4. Kluwer Academic Publishers, Netherlands, 187-201
6. Frost, R. A., Chitte, S. 1999. A New Approach for Providing Natural-Language Speech Access to Large Knowledge Bases. In Proceedings of the Pacific Association of Computational Linguistics Conference, PACLING'99. University of Waterloo, Canada, August 1999, 82-89
7. Ganter, B., Wille, R. 1999. Formal Concept Analysis: Mathematical Foundations. Springer-Verlag, Berlin, Heidelberg, New York. 284 pp. ISBN 3-540-62771-5
8. Groh, B., Strahringer, S., Wille, R. 1998. TOSCANA-Systems Based on Thesauri. Conceptual Structures: Theory, Tools and Applications. In Proceedings of the 6th International Conference on Conceptual Structures, ICCS'98, (Montpellier, France, 10-12

Aug.). Lecture Notes in Artificial Intelligence, no. 1453. Springer-Verlag, Berlin, Germany, 127-38
9. Luksch, P., Wille, R. 1988. Formal Concept Analysis of Paired Comparisons. Classification and Related Methods of Data Analysis. In Proceedings of the 1st Conf. on the Intl. Federation of Classification Societies, IFCS, (Aachen, West Germany, 29 June-1 July 1987). North-Holland, Amsterdam, Netherlands, 567-75
10. Schmitt, I., Conrad, S. 1999. Restructuring Object-Oriented Database Schemata Using Formal Concept Analysis. Informatik Forschung und Entwicklung, vol. 14, no. 4. Springer-Verlag, Germany, 218-26
11. Snelting, G. 2000. Software Reengineering Based on Concept Lattices. In Proceedings of the Fourth European Conference on Software Maintenance and Reengineering (Zurich, Switzerland, 29 Feb.-3 March). IEEE Comput. Soc, Los Alamitos, CA, USA, 3-10
12. Stumme, G. 1999. Acquiring Expert Knowledge for the Design of Conceptual Information Systems. In Proceedings of the 11th European Workshop on knowledge Acquisition, Modeling, and Management, (Dagstuhl, 26-29 May), (submitted). Lecture Notes in Artificial Intelligence, no. 1621. Springer-Verlag, Berlin-Heidelberg, 271-286
13. Tawfik, A. Y., Scott, G. 2001. Temporal Matching under Uncertainty. In Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics.
14. Wille, R. 1982. Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts. Ordered Sets. D. Reidel Publishing Company, Dordrecht-Boston, 445-470
15. Willett, D., Neukirchen, C., Rottland, J., Rigoll, G. 1999. Refining Tree-Based State Clustering by Means of Formal Concept Analysis, Balanced Decision Trees and Automatically Generated Model-Sets. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP99, (Phoenix, AZ, USA, 15-19 March), vol. 2. IEEE, Piscataway, NJ, USA, 565-8