# System Decomposition for Temporal Concept Analysis

David Luper[1], Caner Kazanci[2], John Schramski[3], and Hamid R. Arabnia[4]

[1] Department of Computer Science, University of Georgia, Athens, GA, USA
luper.david@gmail.com
[2] Department of Mathematics, University of Georgia, Athens, GA, USA
caner@uga.edu
[3] Department of Engineering, University of Georgia, Athens, GA, USA
jschrams@uga.edu
[4] Department of Computer Science, University of Georgia, Athens, GA, USA
hra@cs.uga.edu

**Abstract.** Temporal concept analysis is an extension of formal concept analysis (FCA) that introduces a time component to concept lattices allowing concepts to evolve. This time component establishes temporal orderings between concepts represented by directional edges connecting nodes within a temporal lattice. This type of relationship enforces a temporal link between concepts containing certain attributes. The evolution of concepts can provide insight into the underlying complex system causing change, and the concepts evolving can be seen as data emission from that complex system. This research utilizes models of complex systems to provide frequency histograms of activity in well-defined sub-networks within a system. Analyzing systems in this way can provide higher levels of contextual meaning than traditional system analysis calculations such as nodal connectedness and throughflow, providing unique insight into concept evolution within systems.

**Keywords:** Data Mining, Systems Analysis, Knowledge Extraction, Graph Mining, Sequence Mining.

## 1 Introduction

FCA is a principled way of deriving ontological structures from a set of data containing objects and attributes [1]. It establishes concepts from collections of objects exhibiting a certain group of attributes. In a database void of time, these concepts appear without change, however, in temporal concept analysis [2][3][4] time is taken into account and concepts can evolve to take on different meaning. As an example take a database where people are objects possessing the attributes of either young or old. If time steps are present in this database a person *p* could have entries at different time steps, *t1* and *t2*, where *p t1* is labeled young and *p t2* is labeled old. This would highlight that people objects can morph from young to old over sequential time steps. This serves to establish a temporal link from time step *t* to *t + 1* between the attributes young and old. This is a simple example where a one way transition from young to

S. Andrews et al. (Eds.): ICCS 2011, LNAI 6828, pp. 323–330, 2011.

old occurs, but temporal relationships between attributes can be far more complex involving a sophisticated network of transitions encompassing very complex systems of interaction. The underlying system causing the evolution can be modeled in an adjacency matrix of transition probabilities from one attribute to another. This adjacency matrix can be seen as a kind of Markov model outlining the attribute transition probabilities for objects in a concept lattice. A temporal concept attribute model (TCAM) is a modeling of a complex system where nodes in the system are attributes and flows in the system are probabilities that objects possessing an attribute at time $t$ will possess another attribute at time $t + 1$.

Modeling complex systems occurs across a wide variety of scientific disciplines [5][6][7][8][9] including economics, computer science, ecology, biology, sociology, etc. Models (networks) help understand systems that are too complex for deterministic behavior to be recognized, like a person's movement (i.e. tracking a person's GPS data)[10][11], ecosystem food webs [12], rhythm patterns within music [13] or financial volatility within economic systems [14]. Network analysis historically involves the evaluation of network structure and function through the calculation of such metrics as nodal connectedness or compartmental and total system throughflow. This approach is helpful, but lacks the ability to analyze groupings of connected nodes interacting with each other. Perhaps a more complete method for analyzing a network includes taking into account a node's sphere of influence within defined sub-networks. This approach can serve to contextualize the behavior of specific nodes providing a more complete understanding of their role in the network. The goal of this research is to quantify a measure of flow for nodal groupings, signifying levels of importance in a network. The two main obstacles include structurally decomposing a network into groupings of nodes (sub-networks) and calculating the amount of flow that passes through these derived subsets.

## 2 Temporal Concept Attribute Models (TCAM)

An attribute transition model can be constructed from a time stamped database by isolating all instances of object transition from one attribute to another over a given window of time in the database. Strategies for constructing this model include, but are not limited to, the following method. First a group of attributes $A$ must be defined where $A$ contains all the attributes being modeled in the TCAM over a defined time window $T$. For completeness $A$ may need a null attribute value representing an object having an attribute in the model at time step $t$ and then having no attribute from the model at time step $t + 1$. Once $A$ and $T$ are defined a set of objects $O$ must be assembled that will be used to build the TCAM. $O$ can be any logical grouping of objects. With $A$, $T$ and $O$ defined all entries it the database for each object in $O$ over the time window $T$ must be enumerated in time step order. For any object being labeled with attribute $a_1$ at time step $t$ and $a_2$ at time step $t + 1$ a frequency of occurrence value for the edge between $a_1$ and $a_2$ in the TCAM is incremented by one. In this example we are seeking only transitions and constrain the methodology by saying an attribute is not permitted to have an edge looping back to itself. If an object stays in possession of a particular attribute for multiple time steps nothing is modified in the transition matrix. Once every time step in $T$ for every object in $O$ is enumerated the transition

matrix can be normalized to reflect the probability of transition between attributes. As an important note the set of attributes in $A$ must never appear in any combination at the same time step for a single object. If attributes $a_1$ and $a_2$ both appear at time step $t$ for object $o$ a new element must be added to $A$ called $a'$. This new element represents the occurrence of both attributes at the same time step. This maintains consistency in $A$ such that elements of $A$ are attribute state groupings that objects transition in an out of.

## 3 Network Decomposition

This research will pursue a computational algorithm to determine the partial through-flow for meaningful groupings of weakly connected nodes (sub-networks) in a complex system (network). Decomposing a network involves both structural and functional steps that will now be introduced, but flow vectors, sub-network vectors, and the sub-network matrix must be introduced first. A flow vector can be constructed from a network if every edge in the network is labeled with a sequential integer and an edge's magnitude of flow is stored at its respective index (Fig. 1). A sub-network vector (Fig. 2) is a binary vector with a size equal to the flow vector, where for any edge used in the sub-network a 1 is stored at the corresponding vector index and

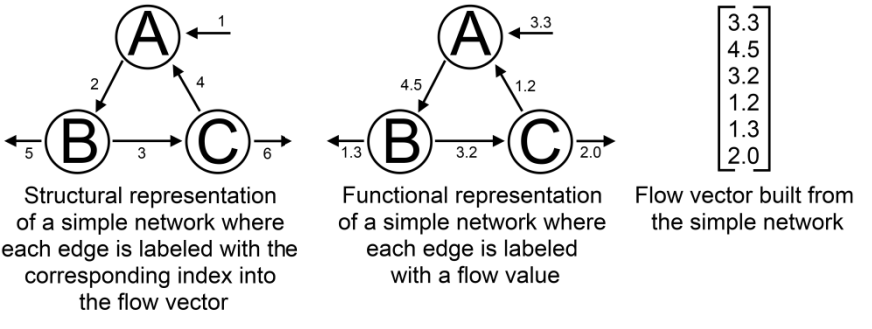


**Fig. 1.** Flow vector

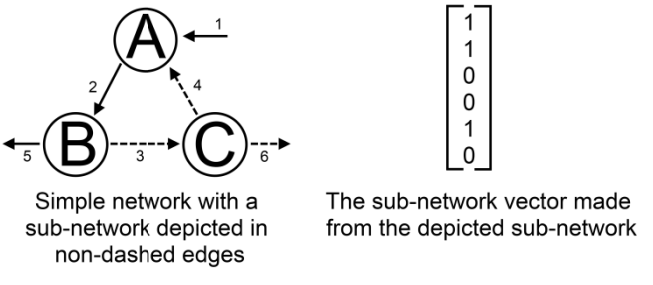


**Fig. 2.** Sub-network vector

all other elements are zero. A sub-network matrix is a matrix of $n$ rows and $m$ columns where $n$ equals the number of edges in the network and $m$ equals the number of decomposed sub-networks. Each column of the sub-network matrix is a sub-network vector.

### 3.1 Structural Decomposition

Structural decomposition is the process of finding all sub-networks within a network. The definition of a sub-network relies on understanding key concepts of network decomposition and throughflow.

- The network, and each sub-network, are assumed to be at steady state (input in equals input out, compartmental storage is ignored).
- Edges in network decompositions are unweighted. The sub-network matrix is the output from a network decomposition, and any flow across the edges is abstracted into coefficient terms pursued later.
- For a sub-network to be dissected from the rest of the network it must be self-sustaining, ensuring that any agent traversing a sub-network will remain in that sub-network without getting lost to some other sub-network. This effectively binds an agent solely to a particular sub-network.
- A constraint is placed on the decomposition of a network, that once decomposed the network (including flows) must be able to be recomposed. This constraint can be met by applying the derived coefficients (discussed later) to their respective columns in the sub-network matrix. Then the rows of the sub-network matrix can be summed to produce the original network flow vector.
- Because a sub-network is unweighted and at steady state, each node in a sub-network must have only one input and one output. If nodes in a sub-network had multiple inputs or multiple outputs an agent traversing the sub-network would have to choose which path to take and the sub-network could not remain unweighted.

Accounting for these concepts defines a sub-network as any path through the network that starts where it ends, has no duplicate nodes and each node has exactly one input and one output. This is the definition of a simple cycle.

A structural decomposition algorithm can now be outlined. Let $M$ be an adjacency matrix for a network. If there are inputs to or outputs from the network, an additional start/stop state (compartment) must be added and its edges must be listed in $M$. The decomposition algorithm takes $M$ as input and places every network compartment into a path of length 1. All the paths are placed in a queue. Until the queue is empty path $p$ is removed from the queue and inspected to see if it is a simple cycle. If $p$ meets this criterion it is output as a sub-network. After inspection, paths of length $len(p) + 1$ are constructed using every edge stored in $M$ for the last node in $p$. Any new simple path (one that has not been created prior to this and has no duplicate nodes) is placed on the end of the queue and the loop continues.

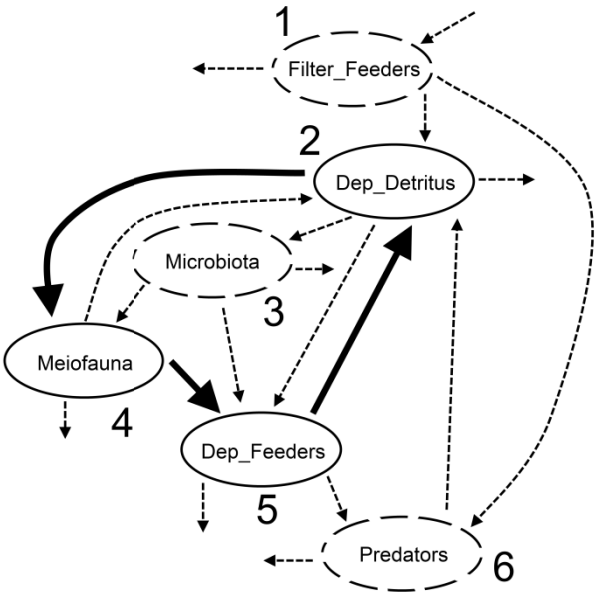


**Fig. 3.** Sub-network (compartments 2-4-5) of an ecological model depicting energy flow in an oyster reef habitat (Dame and Patten 1981)

### 3.2 Functional Decomposition

Functional decomposition is the process of assigning magnitude values to the sub-networks identified in the structural decomposition phase. These magnitude values can represent frequency of occurrence values for sub-networks, or portions of total system throughflow each of sub-networks is responsible for. This research presents a computational framework that analyzes simulated data from a network model and computes a coefficient for each sub-network in a system. It requires as input a weighted adjacency matrix of transition probabilities between compartments in a network. The output is a coefficient vector.

After structural decomposition, a data distribution containing pathways agents took through the system can be simulated using the transition probability matrix, and subsequently analyzed. This allows a histogram to be computed tracking the sub-networks used to interpret each of the data instances. Interpreting a pathway means viewing it as a combination of sub-networks rather than a combination of individual nodes [15], and an interpretation vector is the result of interpreting a pathway. It is a vector of length $n$, where $n$ is the number of sub-networks in the decomposed system. Each index in the interpretation vector represents a particular sub- network, and each value in the vector represents the number of times a sub-network was used in an interpretation. This methodology calculates an interpretation vector for each pathway in a distribution and adds it to a histogram to keep track of how many times each sub-network is used throughout interpretation of the entire distribution of data.

During interpretation of data instances a problem can arise that certain pathways through a network can be interpreted using different sets of sub-networks. An example of a path with multiple interpretations is seen in Fig. 4. If all interpretations for a given pathway are added to the histogram, pathways containing multiple interpretations would have a greater impact on the histogram. Conversely, if only one interpretation is used, sub-networks can be viewed as being responsible for more or less flow depending on a pathway's chosen interpretation. Ultimately this causes multiple correct coefficient vectors to exist, and they constitute a solution space of possible coefficient vectors. This research uses an averaging technique to deal with pathways that contain multiple interpretations. Every interpretation vector for a pathway is added to the histogram after dividing each of them by the total number of interpretations for that pathway. This gives equal weight to every possible interpretation for a pathway, while adding the equivalent of a single interpretation to the histogram.

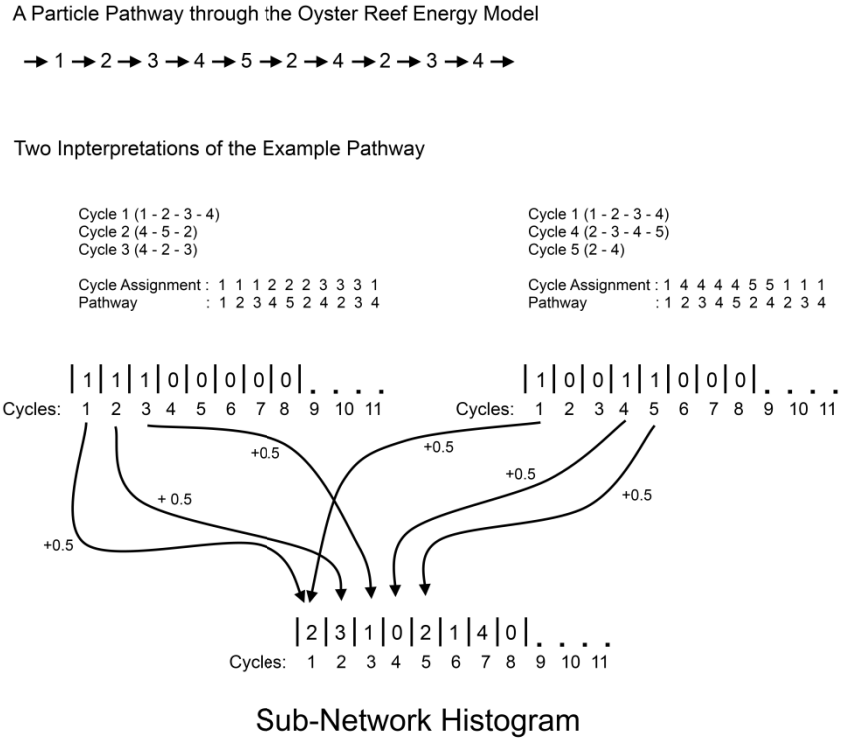


**Fig. 4.** An overview of how to determine coefficient vectors

## 4 Discussion

Decomposing a TCAM and computing coefficients for its sub-networks is a novel way to analyze complex systems behind conceptual evolution. However, much work

is needed to determine useful ways of applying this methodology. Using the information for data mining holds potential to be very transformative and is an interesting way of allowing machines to understand concept evolution. One interesting usage of this methodology would be to compare different windows of time within a time step database using the computed histogram in a distance metric. Complex systems in different states (i.e. they are modeled with different transition probability matrices) would have different coefficients, and the Euclidean distance between those coefficients in a particular feature space could be an invaluable source of information for data mining and knowledge extraction.

The coefficients computed by this methodology hold more information than the transition probabilities alone because they reflect all relationships each of the transition probabilities are involved in. Research needs to be devoted to finding ways to exploit the additional information embedded in this representation of system activity.

## 5 Conclusion

A TCAM is transition probability matrix that models attribute transition within temporal concept analysis. This work has shown how to construct a TCAM from a time stamp database. A TCAM models complex systems driving concept evolution within temporal concept analysis. A methodology was presented for decomposing a TCAM both structurally and functionally. First, an exhaustive set of unique groupings of sub-networks from the TCAM are found. After this, magnitude coefficient values are computed detailing the frequency of occurrence for each sub-network in simulated data from TCAM. This methodology can be seen as a transform that takes as input a transition probability matrix, and outputs a histogram of magnitude values representing frequency of occurrence for each simple cycle in the system. Stated another way, this transform takes a sequential grouping of compartment nodes in a time series (a pathway through a complex system) and maps it out of the temporal domain to a domain representing frequency of occurrence for each sub-network in the system. Viewing system activity in this different domain allows new information about the system to be ascertained specifically related to its decomposed set of sub-networks.

Research is being devoted towards two applications of the proposed methodology. First, effort is being made towards identifying important relationships within ecological models for differentiating seasonal variance. This research uses support vector machines to classify histograms computed from seasonal variations of ecological models. Second, this methodology is being utilized to provide impact analysis on an ecosystem. For this work an ecological model with two different sets of flow values (pre and post impact) are compared and a novel distance metric is outlined, applying the proposed methodology to find how much each of the sub-networks in the model is affected by some impact on the system.

This methodology has great potential, and the diverse range of problems to which it can be applied is a major strength of the work. It holds great potential for systems analysis because it provides a new domain in which system activity can be viewed.

## References


1. Priss, U.: Formal concept analysis in information science. Annual Review of Information Science and Technology 40, 521–543 (2006)
2. Neouchi, R., Tawfik, A.Y., Frost, R.A.: Towards a Temporal Extension of Formal Concept Analysis. In: Proceedings of the 14th Canadian Conference on Artificial Intelligence, Ottawa, Ontario (2001)
3. Wolff, K.E.: Interpretation of Automata in Temporal Concept Analysis. In: Priss, U., Corbett, D.R., Angelova, G. (eds.) ICCS 2002. LNCS (LNAI), vol. 2393, p. 341. Springer, Heidelberg (2002)
4. Wolff, K.E.: Temporal Concept Analysis. In: MephuNguifo, E., et al. (eds.) ICCS-2001 International Workshop on Concept Lattices-Based Theory, Methods and Tools for Knowledge Discovery in Databases, pp. 91–107. Stanford University, Palo Alto (2001)
5. Batagelj, V., Mrvar, A.: Pajek Program for large network analysis. Connections 21(2), 47–57 (1998), Project home page at, `http://vlado.fmf.uni-lj.si/pub/networks/pajek/`
6. Smith, D.A., White, D.R.: Structure and Dynamics of the Global Economy: Network Analysis of International Trade, 1965-1980. Social Forces 70, 857–893 (1992)
7. Wellman, B., Salaff, J., Dimitrova, D., Garton, L., Gulia, M., Haythronwaite, C.: Computer networks as social networks: collaborativework, telework and virtual community. Annu. Rev. Sociol 22, 213–238 (1996)
8. Ammann, P., Wijesekera, D., Kaushik, S.: Scalable, Graph-Based Network Vulnerability Analysis. In: Proceedings of CCS 2002: 9th ACM Conference on Computer and Communications Security, Washington, DC (November 2002)
9. Thibert, B., Bredesen, D.E., del Rio, G.: Improved prediction of critical residues for protein function based on network and phylogenetic analyses. BMC Bioinformatics 6, 213 (2005)
10. Luper, D., Chandrasekaran, M., Rasheed, K., Arabnia, H.R.: Path Normalcy Analysis Using Nearest Neighbor Outlier Detection. In: ICAI 2008, pp. 776-783 (2008); In: Proc. of International Conference on Information and Knowledge Engineering (IKE 2008), Las Vegas, USA, July 14-17, pp. 776-783 (2008) ISBN #: 1-60132-075-2
11. Luper, D., McClendon, R., Arabnia, H.R.: Positional Forecasting From Logged Training Data Using Probabilistic Neural Networks. In: Proc. of International Conference on Information and Knowledge Engineering (IKE 2009), Las Vegas, USA, July13-26, pp. 179–189 (2009) ISBN # for set: 1-60132-116-3
12. Rohr, R.P., Scherer, H., Kehrli, P., Mazza, C., Bersie, L.: Modeling Food Webs: Exploring Unexplained Structure Using Latent Traits. The American Naturalist 176(2), 170–177 (2010)
13. Temperley, D.: Modeling Common - Practice Rhythm. Music Perception: An Interdisciplinary Journal 27(5), 355–376 (2010)
14. Marcelo, C.: Medeiros and Alvaro Veiga, Modeling Multiple Regimes in Financial Volatility with a Flexible Coefficient GARCH(1, 1) Model. Econometric Theory 25(1), 117–161 (2009)
15. Luper, D., Kazanci, C., Schramski, J., Arabnia, H.R.: Flow Decomposition in Complex Systems. ITNG, Las Vegas, USA (2011)