

METHOD OF DETERMINING WEIGHTS OF TEMPORAL RULES IN MARKOV LOGIC NETWORK FOR BUILDING KNOWLEDGE BASE IN INFORMATION CONTROL SYSTEMS

Viktor Levykin

*Department of Information Control Systems
Kharkiv National University of Radio Electronics
16 Nauka ave., Kharkiv, Ukraine, 61166
levykinvictor@gmail.com*

Oksana Chala

*Department of Information Control Systems
Kharkiv National University of Radio Electronics
16 Nauka ave., Kharkiv, Ukraine, 61166
oksana.chala@nure.ua*

Abstract

The problem of constructing and expanding the temporal knowledge base for the information-control system is considered. This knowledge base is formally represented by the Markov logic network. It is shown that the behavior of the control object of a given class can be reflected in the form of a set of weighted temporal rules. These rules are formed on the basis of identifying links between events that reflect known variants of the behavior of the control object. A method is proposed for calculating the weights of temporal rules in a Markov logic network for a given level of detail of the control object. The level of detail is determined by the context for executing the sequences of control actions and for weighted temporal rules is specified by selecting subsets of the event attributes. The method includes such basic phases: preparation of a subset of temporal rules for a given level of detail; finding the weights of the rules taking into account the a priori probabilities of the event traces. The method creates conditions for supporting management decisions in information management systems at various levels of detail of complex management objects. Decision support is provided by predicting the probability of success in executing a sequence of actions that implement the management function in the current situation. These probabilities are determined using the weights of the temporal rules.

Keywords: temporal rule, temporal knowledge base, information control system, event log, event attributes.

DOI: 10.21303/2461-4262.2018.00713

© Viktor Levykin, Oksana Chala

1. Introduction

Modern methods of automated construction of knowledge bases [1, 2] use information from databases, sites on the Internet, and logs that reflect the functioning of various systems and processes, to isolate sets of dependencies in the format of rules and templates [3].

In contrast to the traditional paradigm of building knowledge bases like “creating knowledge by people – use of knowledge by people” [4], in this case the paradigm “automated creation of knowledge – use of knowledge in information systems” is realized. The key advantage of the paradigm of automated knowledge base construction is a significant reduction in the costs of allocating and integrating the required dependencies, which allows to continually add to the set of existing knowledge as we access new data [5, 6]. This advantage provides an opportunity to use the knowledge bases built by the automated method for a wide range of information systems – from information retrieval to information-control systems [7].

Currently used methods of automated knowledge base construction are focused primarily on solving information search problems, including reference information [8]. Therefore, the knowledge representation used primarily uses static dependencies between the facts of the subject domain. Changes in the subject area (the emergence of new facts) are reflected by completing the knowledge base. The obtained dependencies can take into account the temporal attributes of the facts. However, due to the peculiarities of the tasks solved by such information systems, the interpretation of temporal attributes by allocating temporal dependencies is not given enough attention.

At the same time, when solving managerial tasks, the temporal dependencies between facts, events and actions in the subject area are used. Each event reflects a change in the truth of the facts that reflect the state of the control object. Changing the state of a control object is usually associated with executing control actions or changing the properties of these actions. Accordingly, temporal dependencies reflect the influence of control actions on the state of the control object, which allows to predict its further behavior and thereby provide support for management decisions. The accuracy of this forecast depends on the importance of the used temporal rules: the more significant the rule, the less likely it is to be violated. When constructing knowledge bases on the basis of the formalism of a Markov logic network, the importance of a rule is given through the value of its weight [9, 10].

Thus, the problem of building and expanding the knowledge base for an information management system using weighted temporal rules is topical.

2. Literature review and problem statement

The application of the knowledge base to support decision-making in the information management system requires consideration of the multivariance of the behavior of the control object. To describe such multivariance, a logical-probabilistic representation of knowledge based on Markov logic networks is used [11–13]. The logic component specifies the patterns of possible states and actions for the control object, as well as the links between these actions and states. Based on these patterns, knowledge of the domain is formed in the form of a set of weighted predicates. Weights of predicates are used in constructing a probabilistic description in accordance with the rules of the Markov logic network [14]. The probabilistic component makes it possible to identify the most acceptable variant of the change in the states of the control object [15].

When constructing the Markov logic network as the basis of the knowledge base of the information management system, it is necessary to take into account the eventual nature of the detected temporal dependencies. A general method for constructing a knowledge base using such dependencies is proposed in [16], and the approach to the formation of initial data in the form of a sequence of events is given in [17]. However, the issues of formation of weights of temporal rules taking into account the characteristics of the initial event description of the control object in this work have not been considered.

The existing methods for finding the weights of dependencies for a Markov logic network are based on the use of gradient descent. The criterion for completing the descent is to find such weights of the rules that make it possible to obtain the maximum correspondence between the calculated and a priori known probabilities of the appearance of sets of values of variables [18]. The main drawback of the known algorithms for finding weights is that they converge very slowly, the complexity of the computation grows exponentially in the general case with respect to the number of dependencies.

There are two groups of methods for finding weights: generative and discriminatory training. In the first case, weights are calculated for both the observed facts and for the derived rules [11]. In the second case, the weights of the deduced rules for given observable facts are determined [19], and to improve the efficiency of the output, auxiliary information is used, which allows optimizing the objective function [20].

Thus, the existing methods of searching for weights have considerable computational complexity, which makes it difficult to use them in the construction and use of knowledge bases in real-time information management systems. To overcome this, it is advisable to use additional information, which makes it possible to reduce many possible solutions and, as a consequence, accelerate the convergence of gradient descent.

The aim of this article is development of a method for calculating the weights of temporal rules in a Markov logic network for constructing temporal knowledge bases in order to provide a continuous refinement of knowledge about objects and management processes in information management systems.

3. Method for calculating the weights of temporal rules

The method of calculating weights uses temporal rules, which determine knowledge of the possible and admissible behavior of the control object.

The proposed rules link pairs of events. Each event is considered as a logical fact, corresponding to the results of the action in certain conditions of the domain. As arguments for a predicate formalizing a logical fact, the attributes of the event are used.

The temporal rules reflect the following properties of the sequences of events that are formed by the information management system and describe the state sequences of the control object:

- each event contains information about the action performed with the control object, as well as the context of the execution of this action; contextual information is represented by the attributes of the event;
- event attributes correspond to the attributes of artifacts included in the control object;
- each of the events associated with a single control object has the same fixed set of attributes;
- events are presented in discrete time; each event has a timestamp that reflects the moment of its appearance;
- events are ordered in time in accordance with a change in the behavior of the control object in the form of traces;
- there is a set of ordered sequences of events corresponding to different instances of each control object;
- set of sequences of events is a “sweep” of temporal knowledge of its behavior.

The proposed basic temporal rules are sequential rules that define pairs of process actions that execute one after another (directly or in the future). Since each action within the information management system is displayed as a separate event, the continuous action rule can be reduced to a continuous rule for a pair of events for one instance of the control object.

The rule of continuous following C_{Next} for a pair of events $e_{i,j}$ and $e_{i,n}$ define as a temporal relationship between two predicates $Af(e_{i,j})$ and $Af(e_{i,n})$ reflecting the logical facts of the occurrence of these events. Since each event $e_{i,j}$ is defined by a set of K attributes $\{a_{i,j}^k\}, k = 1, K$, the logical fact can be specified through attributes:

$$Af(e_{i,j}) = Af(\{a_{i,j}^k\}). \quad (1)$$

Using the values of the event attributes to determine the validity of logical facts allows to generalize these facts, as well as the corresponding rules. A generalization is performed by selecting a subset of key attributes.

The rule C_{Next} is specified by the temporal logic operator X (NeXt), which determines the successive appearance of two events for one instance of the control object:

$$C_{Next} = Af(e_{i,j})XAf(e_{i,n}). \quad (2)$$

The domain of truth Y_{Af} of predicates $Af(e_{i,j})$ and $Af(e_{i,n})$ on the set of all events $E = \{e_{i,j}\}$ is determined through the belonging of these events to any trace π_i describing the behavior of the instance of the control object.

This rule specifies a pair of events e_j and e_n of trace π_i between which no other events occurred. Therefore, the truth area Y_{Next} of this rule is defined for events e_j and e_n with timestamps τ_j and τ_n , between which there is no intermediate timestamp τ^* for another event e^* .

Each trace reflects the execution of only one instance of the control object. Therefore, the same pairs of events can be repeated on several routes, for different instances of the same object. This allows to generalize the dependence (1) for all event traces of a single control object:

$$\begin{aligned} C_{Next} &= Af(e_j)XAf(e_n), \\ Y_{Af} &= \{e_j, e_n \in E \mid \exists \pi_i : e_j, e_n \in \pi_i\}, \\ Y_{Next} &= \{(e_j, e_n) \in \pi_i \mid \exists \pi_i : \forall e^* \neq e_j, e^* \in \pi_i, \tau^* < \tau_j \vee \tau^* \geq \tau_n\}. \end{aligned} \quad (3)$$

The rule of following in the future for events on the track π_i is given by the temporal logic operator F (Future). This operator determines the appearance of the event $e_{i,n}$ after several intermediate events after the event $e_{i,j}$. The appearance of these events $Af(e_{i,j})$ and $Af(e_{i,n})$ is given by predicates and. The truth area Y_{Future} of this rule is defined for events e_j and e_n from one trace, between which there is at least one intermediate event e_m :

$$C_{Future} = Af(e_j)FAf(e_n),$$

$$Y_{Future} = \{(e_j, e_n) \in \pi_i \mid \exists e_m \in \pi_i, e_m \neq e_j, e_m \neq e_n : Af(e_j)X Af(e_m) \vee Af(e_m)X Af(e_n)\}. \quad (4)$$

These rules reflect the preferences for processing the object in the temporal aspect. The indicated preferences are determined by:

- conditions for constructing a sequence of control actions;
- restrictions on the performance of actions.

The difference between constraints and conditions is that the constraints are satisfied for all instances of the control object. This allows a priori to set a large weight for the constraints and, thus, to exclude restrictions from further calculation.

The initial data for the method are: a set of sequences of events for the selected subset of instances of the control object; a set of temporal rules containing knowledge of admissible sequences for pairs of events; a subset of key event attributes for generalizing rules.

The method for calculating the weights of temporal rules includes the following steps.

Step 1. Generalization of rules for pairs of identical events for different traces. The generalization is performed in accordance with expression (1) based on the comparison of the complete set of event attributes. If both events from a pair have the same attribute values, then the rules are considered identical.

The result of the first step is a complete set of rules on a given set of events.

Step 2. Generalization of the rules for the types of artifacts (objects with which the actions are performed). At this step, the resulting complete set of rules is generalized using a subset of the event attributes represented in the input data. Attributes of events correspond to attributes of used artifacts. Therefore, to typify the rules, it is sufficient to select the target group of artifacts and generalize the rules that have the same attribute values for the selected group.

The aim of this step is ensuring the formation of rules for a given level of detail of the complex management object. This allows to build a hierarchy of rules to support the adoption of management decisions, taking into account key artifacts for each level.

Step 3. Elimination of restrictions for a given level of detail of temporal rules. As a rule, rules that are executed on all routes reflecting the behavior of this control object are considered. Obviously, the number of restrictions for a given set of rules largely depends on the level of detail of the rules: more general rules will be executed on more traces. Obtained restrictions are subject to unconditional implementation. Therefore, the weight ∞ is given for them. According to the rule for calculating the probability of implementing a route for a Markov logic network, this weight provides a unit probability for any trace associated with the constraint [11]. This allows to exclude restrictions from further consideration and thereby reduce the complexity of the task of calculating the weights of temporal rules.

The result of this step is a set of rules for which weights will be calculated.

Step 4. Calculation of the probability of realizing the traces. The criterion for selecting the weight of each rule in a Markov net in the event of an event description of the control object is that the total weight of the rules for the path should correspond to the probability of this path as much as possible [16].

Therefore, at this step, an a priori set of probabilities is created to find the weights. The probability of implementing a trace $P(\pi_i)$ is established traditionally, as the ratio of the number of identical traces π_i to the total number of traces Π :

$$P(\pi_i) = \frac{|\{\pi_i\}|}{|\Pi|}.$$

Step 5. Finding the weights of the obtained subset of temporal rules on the basis of gradient descent by the method presented in [20].

The result of this method is the set of pairs: (temporal rule, rule weight).

4. Experiments

The aim of the experiment is testing the possibility of constructing a hierarchy of weighted rules that segment knowledge of the behavior of the control object both over the hierarchy levels and the classification characteristics specified in the input data.

Let's consider the generalization of rules for a set of sequences of events with functional control, which reflects the behavior of three instances of the control object. The initial data for calculating the weights are given in **Table 1**.

Table 1

The initial data of the method

Name	Data
Traces	$\langle e_{1,1}, e_{1,2}, e_{1,4}, e_{1,5} \rangle, \langle e_{2,1}, e_{2,2}, e_{2,3}, e_{2,4}, e_{2,5} \rangle, \langle e_{3,1}, e_{3,2}, e_{3,6}, e_{3,5} \rangle$
Rules for the trace 1	$Af(e_{1,1})XAf(e_{1,2}), Af(e_{1,1})FAf(e_{1,4}), Af(e_{1,1})FAf(e_{1,5}),$ $Af(e_{1,2})FAf(e_{1,5}), Af(e_{1,2})XAf(e_{1,4}), Af(e_{1,4})XAf(e_{1,5})$
Rules for the trace 2	$Af(e_{2,1})XAf(e_{2,2}), Af(e_{2,1})FAf(e_{2,3}), Af(e_{2,1})FAf(e_{2,5}), Af(e_{2,2})FAf(e_{2,5}),$ $Af(e_{2,2})XAf(e_{2,3}), Af(e_{2,3})XAf(e_{3,4}), Af(e_{2,4})XAf(e_{2,5}), Af(e_{2,2})FAf(e_{2,4}),$ $Af(e_{2,3})XAf(e_{2,5}).$
Rules for the trace 3	$Af(e_{3,1})XAf(e_{3,2}), Af(e_{3,1})FAf(e_{3,5}), Af(e_{3,2})FAf(e_{3,5}),$ $Af(e_{3,1})FAf(e_{3,6}), Af(e_{3,2})XAf(e_{3,6}), Af(e_{3,6})XAf(e_{3,5})$
Event attributes	a_j^1 – department, a_j^2 – activity, a_j^3 – name
Detail attribute	a_j^1 – department
Detail attribute value	“11” – for events $e_{1,1}, e_{1,2}, e_{1,3}, e_{2,1}, e_{2,2}, e_{2,3}, e_{3,1}, e_{3,2}$ “15” – for events $e_{1,4}, e_{1,5}, e_{2,4}, e_{2,5}, e_{3,6}, e_{3,5}$

In accordance with the input data, it is necessary to generalize the rules at two levels of detail: the lower level corresponds to the actions of individual employees, and the upper level corresponds to activities at the departmental level. Two departmental codes are used, each associated with events.

The method was performed twice: (1) without taking into account the detail attribute, to obtain the lower-level rules: (2) taking into account the detail attribute, to obtain rules at the departmental level.

The probabilities of all these paths are 0.33.

The sets of weighted rules obtained as a result of the double execution of the method are shown in **Table 2**

Let's note that weight rules are satisfied on all routes, they are restrictions and therefore they were not taken into account in step 5 of this method. Preliminary allocation of such restrictions allowed in this example from three routes to reduce by 20 % the number of calculated weights of rules at the performer level and by 25 % at the departmental level.

Table 2

The results of the method

Name of level of rules	Weighted rules
Department «11»	$(Af(e_1)F Af(e_3), 0.03)$
Department «15»	$(Af(e_4)X Af(e_5), 8.5), (Af(e_6)X Af(e_5), 8.53)$
Limitations of the department level	$(Af(e_1)X Af(e_2), \infty)$
Performers (baseline level)	$(Af(e_1)F Af(e_4), 0.9), (Af(e_1)F Af(e_3), 0.45), (Af(e_1)F Af(e_6), 0.47),$ $(Af(e_2)X Af(e_4), 0.89), (Af(e_4)X Af(e_5), 0.11), (Af(e_2)X Af(e_3), 0.62),$ $(Af(e_3)X Af(e_4), 0.37), (Af(e_4)X Af(e_5), 0.11), (Af(e_2)F Af(e_4), 0.15),$ $(Af(e_3)F Af(e_5), 0.2), (Af(e_2)X Af(e_6), 0.92), (Af(e_6)X Af(e_5), 0.5)$
Limitations of the initial level	$(Af(e_1)X Af(e_2), \infty), (Af(e_1)F Af(e_3), \infty), (Af(e_2)F Af(e_3), \infty)$

Thus, the calculation of weights, taking into account the attributes of events, allows to obtain a generalized representation of temporal knowledge with a grouping of dependencies by separating criteria.

5. Results and discussion

The result of the work is a method for determining the weights of temporal rules in a Markov logic network, oriented to an eventual description of a control object functioning in real time.

The resulting weights of the individual rules allow to determine the likelihood of the actions associated with these rules being fulfilled. The set of weighted rules makes it possible to calculate the probability of implementing sequences of actions that lead the control object to the target state. Using weighted rules allows to support the adoption of management decisions by selecting one of the several most likely sequences of actions for the current state of the control object.

The difference between the proposed methods consists in the generalization of the components of the rule of logical facts by taking into account the values of the attributes of events. This generalization allows to combine events from different sequences, describing how the behavior of different instances of the same control object, and the behavior of similar objects. As a result let's determine the weights of general rules that reflect the basic temporal patterns in behavior for a given class of control objects.

The advantage of the method is the ability to find the weights of generalized rules for different levels of detail of the control object based on the selection of subsets of event attributes. This allows to obtain a hierarchy of weighted rules and provide support for functional management using a hierarchical decision-making scheme.

The disadvantage of the method is that the resulting weights of the rules depend to a large extent on the quality of the input data, which is represented by a subset of the sequence of events characterizing known variants of the behavior of the control object.

The method can also be used to form a hierarchy of weighted rules describing the normal, abnormal and faulty behavior of programs in computer systems.

6. Conclusions

The problem of constructing and expanding the temporal knowledge base for an information management system based on the use of weighted temporal rules is considered.

The expediency of finding the weights of temporal rules is justified by the fact that, according to the properties of the Markov neural network, these weights determine the probability of the control actions prescribed by these rules.

A method is proposed for calculating the weights of temporal rules in the Markov logic network, which includes the steps of preparing a set of temporal rules at a given level of detail, calculating the initial probabilities of the event traces, finding the weights of the rules.

The proposed method has the following differences from traditional approaches to finding the weights of dependencies in the Markov logic network.

First, the level of detail of the set of rules for which weights are determined is taken into account. On the one hand, this allows to reduce the computational cost by reducing the total number of rules as a result of their generalization for a given level of presentation. On the other hand, generalized rules operate with a selected subset of key artifacts, which avoids unnecessary detail and thereby increases the effectiveness of support for management decision making.

Secondly, the method allows to exclude from consideration limitations for the selected level of detail, thereby reducing the dimensionality of the problem and reducing the computational cost of finding the weights of the rules.

In practical terms, the developed method creates conditions for improving the management efficiency of complex objects and processes in non-standard situations by estimating the current state and determining the probabilities of feasible ways out of this state. The estimation of the current non-standard state is carried out on the basis of the total probability of all possible ways out of this state. The choice of the best way to achieve the target state is carried out taking into account the obtained probability of realizing this path. The definition of the weights of the rules is a necessary condition for calculating the probabilities of ways out of the non-standard situation.

References

- [1] Shin, J., Wu, S., Wang, F., De Sa, C., Zhang, C., Ré, C. (2015). Incremental knowledge base construction using DeepDive. *Proceedings of the VLDB Endowment*, 8 (11), 1310–1321. doi: <http://doi.org/10.14778/2809974.2809991>
- [2] Nguyen, D. B., Abujabal, A., Tran, N. K., Theobald, M., Weikum, G. (2017). Query-driven on-the-fly knowledge base construction. *Proceedings of the VLDB Endowment*, 11 (1), 66–79. doi: <http://doi.org/10.14778/3151113.3151119>
- [3] Re, C., Sadeghian, A., Shan, Z., Shin, J., Wang, F. Wu, S. et. al. (2014). Feature Engineering for Knowledge Base Construction. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 37, 26–40.
- [4] Kendal, S. L., Creen, M. (2007). *An introduction to knowledge engineering*. London: Springer, 290. doi: <http://doi.org/10.1007/978-1-84628-667-4>
- [5] Nakashole, N., Weikum, G. (2012). Real-time Population of Knowledge Bases: Opportunities and Challenges. *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX) AKBC Workshop*. Montreal, 41–45.
- [6] Kalynychenko, O., Chalyi, S., Bodyanskiy, Y., Golian, V., Golian, N. (2013). Implementation of search mechanism for implicit dependences in process mining. *2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS)*. doi: <http://doi.org/10.1109/idaacs.2013.6662657>
- [7] Galarraga, L., Heitz, G., Murphy, K., Suchanek, F. M. (2014). Canonicalizing Open Knowledge Bases. *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management – CIKM '14*. Shanghai, 1679–1688. doi: <http://doi.org/10.1145/2661829.2662073>
- [8] Niu, F., Zhang, C., Re, C. (2012). DeepDive: Web-scale Knowledge-base Construction using Statistical Learning and Inference. *VLDS*, 25–28.
- [9] Huynh, T. N., Mooney, R. J. (2010). Online Max-Margin Weight. *Proceedings of the 6th AAAI Conference on Statistical Relational Artificial Intelligence. AAAIWS'10-06*. Atlanta, 32–37.
- [10] Huynh, T. N., Raymond, J. (2009). MooneyMax-Margin Weight Learning for Markov Logic Networks. *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*. Bled, 564–579. doi: http://doi.org/10.1007/978-3-642-04180-8_54

- [11] Richardson, M., Domingos, P. (2006). Markov logic networks. *Machine Learning*, 62 (1-2), 107–136. doi: <http://doi.org/10.1007/s10994-006-5833-1>
- [12] Niu, F., Zhang, C., Re, C., Shavlik, J. (2012). Scaling inference for markov logic via dual decomposition. 12th International Conference on Data Mining (ICDM). Brussels, 1032–1037. doi: <http://doi.org/10.1109/icdm.2012.96>
- [13] Singla, P., Domingos, P. (2006). Entity resolution with Markov logic. 6 th International Conference on Data Mining (ICDM). Hong Kong. doi: <http://doi.org/10.1109/icdm.2006.65>
- [14] Sarkhel, S., Venugopal, D., Singla, P., Gogate, V. (2014). Lifted MAP Inference for Markov Logic Networks. 17th International Conference on Artificial Intelligence and Statistics (AISTATS), 859–867.
- [15] Gogate, V., Domingos, P. (2010). Formula-Based Probabilistic Inference. *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*.
- [16] Levykin, V., Chala, O. (2018). Method of automated construction and expansion of the knowledge base of the business process management system. *EUREKA: Physics and Engineering*, 4, 29–35. doi: <http://dx.doi.org/10.21303/2461-4262.2018.00676>
- [17] Chalyi, S., Levykin, I., Petrychenko, A., Bogatov, I. (2018). Causality-based model checking in business process management tasks. *IEEE 9th International Conference on Dependable Systems, Services and Technologies DESSERT'2018*. Kyiv, 478–483. doi: <http://doi.org/10.1109/dessert.2018.8409176>
- [18] Shewchuck, J. (1994). An introduction to the conjugate gradient method without the agonizing pain. *Technical Report CMU-CS-94-125*. Pittsburgh, 58.
- [19] Singla, P., Domingos, P. (2005). Discriminative Training of Markov Logic Networks. *Proceedings of the 20th national conference on Artificial intelligence*. Available at: <https://homes.cs.washington.edu/~pedrod/papers/aaai05.pdf>
- [20] Lowd, D., Domingos, P. (2007). Efficient weight learning for Markov logic networks. *European Conference on Principles of Data Mining and Knowledge Discovery. Knowledge discovery in databases: PKDD*, 200–211. doi: http://doi.org/10.1007/978-3-540-74976-9_21