On-Line Elimination of Non-relevant Parts of Complex Objects in Behavioral Pattern Identification

Jan G. Bazan[1] and Andrzej Skowron[2]

[1] Institute of Mathematics, University of Rzeszów,
Rejtana 16A, 35-959 Rzeszów, Poland
bazan@univ.rzeszow.pl
[2] Institute of Mathematics, Warsaw University,
Banacha 2, 02-097 Warsaw, Poland
skowron@mimuw.edu.pl

Abstract. We discuss some rough set tools for perception modelling that have been developed in our project for a system for modelling networks of classifiers for compound concepts. Such networks make it possible to recognize behavioral patterns of objects and their parts changing over time. We present a method that we call *a method for on-line elimination of non-relevant parts* (ENP). This method was developed for on-line elimination of complex object parts that are irrelevant for identifying a given behavioral pattern. Some results of experiments with data from the road simulator are included.

1 Introduction

Many real life problems can be modelled by systems of complex objects and their parts changing and interacting over time. The objects are usually linked by some dependencies, can cooperate between themselves and are able to perform flexible autonomous complex actions (operations). Such systems are identified as *complex dynamical systems* [2] or *autonomous multiagent systems* [9]. As an example of such a dynamical system one can consider *road traffic*. For experiments we have developed a road simulator, see [15] and [4] for more details.

The identification of behavioral patterns of complex dynamical systems can be very important for identification or prediction of global behavior of the investigated system, e.g., behavioral patterns can be used to identify some undesirable behaviors of complex objects (see [4] for more details).

Studying cognition, and in particular, perception based computing [1,7,8,6] [10,17] is becoming now one of the very active research directions for methods of complex concept approximation [3,4,11,12,14,18] and as a consequence for building intelligent systems.

We discuss an exemplary rough set [13] tool for perception modelling that was developed in our project for modelling networks of classifiers for compound concepts. Such networks make it possible to recognize behavioral patterns of

S.K. Pal et al. (Eds.): PReMI 2005, LNCS 3776, pp. 720–725, 2005.

complex objects and their parts changing over time. We use *behavioral graphs* (see Section 2) for representing *behavioral patterns* of complex objects. The behavioral pattern identification for a part of complex object is performed by testing properties of registered behavior of this part, often during a quite long period of time. However, in many applications, faster (often in real-time) testing if parts of complex objects are matching the given behavioral pattern is necessary. Hence, we have developed the ENP method for on-line elimination of non-relevant for a given behavioral pattern parts of complex object in a short observational period. In analysis of complex dynamical systems, we usually have to investigate very many parts of complex objects. Therefore, the fast verification of parts of complex objects can save time necessary for searching among parts matching the given behavioral pattern. In Section 3 we show that the testing of parts of complex objects can be speeded up by using some special decision rules. The presented ENP method makes it possible to achieve very fast elimination of many irrelevant parts of a given complex object in identification of a given behavioral pattern. To illustrate the method and to verify the effectiveness of classifiers based on behavioral patterns, we have performed several experiments with the data sets recorded in the road simulator [15]. We present results of our experiments in Section 4.

2 Behavioral Patterns

In many complex dynamic systems, one can distinguish some *elementary actions* (performed by complex objects), that can be expressed by a local change of object parameters, measured in a very short but a registerable period [4]. However, the perception of more complicated actions or behaviors requires analysis of elementary actions performed over a longer period. This longer period we often call as a *time window*. Therefore, if we want to predict more compound actions or discover a behavioral pattern we have to investigate all elementary actions, that have been performed by the investigated complex object in the current time window. Hence, one can, e.g., consider the frequency of elementary actions in a given time window and temporal dependencies between them. These properties can be expressed using *temporal patterns*. Any temporal pattern is a function defined by features parameters of an observed object over the time window [4]. We assume that any temporal pattern is defined by a human expert using domain knowledge accumulated for the given complex dynamic system.

The temporal patterns can be treated as new attributes (features) that can be used for approximation of more complex concepts. In this paper we call such concepts as *temporal concepts*. We assume that temporal concepts are specified by a human expert and are usually used in queries about the status of some objects in the given temporal window. The approximation of temporal concepts is defined by classifiers [4].

The temporal concepts defined over objects from some complex dynamical system and approximated by classifiers, are used as nodes of a graph, that we call as *a behavioral graph*. The branches in the behavioral graph represent temporal dependencies between nodes. The behavioral graph can be constructed for different kinds of objects appearing in the investigated complex dynamic system (e.g., for a single vehicle, for a group of vehicle, for a short vehicle like a small car, for long vehicle like a truck with a trailer) and it is usually defined for some kind of behavior of a complex object (e.g., driving on the strength road, driving through crossroads, overtaking, passing). For example, if one would like to investigate the overtaking maneuver, it is necessary to observe at least two vehicles: an overtaking vehicle and an overtaken vehicle.

In case of perception of a complex behavior (of parts or groups of objects), when we have to observe the behavior of dynamic systems over a long period of time, we can construct a behavioral graph for the investigated complex behavior. Next, using this graph, we can investigate a complex behavior of objects or a group of objects during some period. It is possible, by observing of transitions between nodes of behavioral graph and registering a sequence of nodes, that make some path of temporal patterns. If the path of temporal patterns (registered for an investigated object or group of objects) is matching a path in the behavioral graph, we conclude, that the behavior of this object or group is compatible with the behavioral graph. So, we can use the behavioral graph as a complex classifier for perception of the complex behavior of objects or groups of object. Therefore, the behavioral graph constructed for some complex behavior we call as *a behavioral pattern*.

3 Discovering Perception Rules for Fast Elimination of Behavioral Patterns

Let us assume, that we have a family of behavioral patterns $BP = \{b_1, ..., b_n\}$ defined for groups of objects (or parts of a given object). For any pattern b_i from the family BP one can construct a complex classifier based on a suitable behavioral graph (see Section 2) that makes it possible for us to answer the question: "Does the behavior of the investigated group (or the part of a complex object) match the pattern b_i?". The identification of behavioral patterns of any group is performed by investigation of a time window sequence registered for this group during some period (sometimes quite long). This registration of time windows is necessary if we want to avoid mistakes in identification of the investigated object group. However, in many applications, we are forced to make a faster (often in real-time) decision if some group of objects is matching the given behavioral pattern. In other words, we would like to check the investigated group of objects at once, that is, using the first or second temporal window of our observation only. This is very important from the computational complexity point of view, because if we investigate complex dynamic systems, we usually have to investigate very many groups of objects. Hence, the faster verification of groups can help us to optimize the process of searching among groups matching the given behavioral pattern.

The verification of complex objects consisting of some groups of objects can be speeded up by using some special decision rules. Such rules are making it possible to exclude very fast many parts (groups of objects) of a given complex object as irrelevant for identification of a given behavioral pattern. This is possible because these rules can be often applied at once, that is after only one temporal window of our observation.

Temporal patterns are constructed over temporal windows. At the beginning we define a family of temporal patterns $TP = \{t_1, ..., t_m\}$ that have influence on matching of investigated groups to behavioral patterns from family BP. These temporal patterns should be defined on the basis of information from temporal windows (for the verification of single object) or from information in behavioral graphs (for the verification of group of objects). Next, we construct classifiers for all defined temporal patterns (see [4] for more details).

For any temporal pattern t_i from the family TP we create a decision table DT_i that has only two attributes. Any object-row of the table DT_i is constructed on the basis of information registered during a period that is typical for the given temporal pattern t_i. The second attribute of the table DT_i (the decision attribute of this table) is computed using the classifier for t_i. The condition attribute registers the index of behavioral pattern from the family BP. This index can be obtained by using complex classifiers created for behavioral patterns from the family BP, because any complex classifier from the family BP can check for a single temporal window (and its time neighborhood) whether the investigated group of objects is matching the given behavioral pattern.

Next, we compute decision rules for DT_i using methods of attribute values grouping that have been developed in the RSES system [16]. Any computed decision rule expresses a dependency between a temporal pattern and the set of bahavioral patterns that are not matching this temporal pattern. Let us consider a very simple illustrative example. Assume we are interested in the recognition of overtaking that can be understand as a behavioral pattern, defined for the group of two vehicles. Using the methodology presented above, we can obtain the following decision rule: `If the vehicle A is overtaking B then the vehicle B is driving on the right lane`. After applying of the transposition low, we obtain the following rule: `If the vehicle B is not driving on the right lane then the vehicle A is not overtaking B`. The last rule allow us for fast verification whether the investigated group of objects (two vehicle: A and B) is matching the behavioral pattern of overtaking. Of course, in case of the considered complex dynamic system, there are many other rules that can help us in the fast verification of groups of objects related to the overtaking behavioral pattern. Besides, there

Table 1. Results of experiments for the overtaking pattern

Decision class	Method	Accuracy
YES (overtaking)	BP	0.923
	BP-E	0.883
NO (no overtaking)	BP	0.993
	BP-E	0.998
All classes (YES + NO)	BP	0.989
	BP-E	0.992

are many other behavioral patterns in this complex dynamic system and we have to calculate rules for them using the methodology presented above.

The ENP method is not a method for behavioral pattern recognition. However, this method allows us to eliminate some behavioral patterns that are not matched by a given group (or a part) of objects. After such elimination the complex classifiers based on a suitable behavioral graphs should be applied to the remaining parts of the complex object.

4 Experiments with Data

To verify the effectiveness of classifiers based on behavioral patterns, we have implemented our algorithms in the *Behavioral Patterns* programming library (BP-lib). This is an extension of the RSES-lib 2.2 programming library forming the computational kernel of the RSES system [16].

The experiments have been performed on the data sets obtained from the road simulator (see [15]). We have applied the "train and test" method for estimating accuracy. A training data set consists of 17553 objects generated by the road simulator during one thousand of simulation steps. Whereas, a testing data set consists of 17765 objects collected during another (completely different) session with the road simulator.

In our experiments, we compared the quality of two classifiers: BP and BP-E. The classifier BP is based on behavioral patterns (see Section 2) and the classifier BP-E is based on behavioral patterns too but with application of the ENP method (see Section 3). We compared BP and BP-E using accuracy of classification [3]. Table 1 shows the results of the considered classification algorithms for the concept related to the overtaking behavioral pattern.

One can see that in the case of perception of the overtaking maneuver (decision class YES) the accuracy of algorithm BP-E is 4% lower than the accuracy of the algorithm BP. However, the algorithm BP-E allows us to reduce the time of perception, because during perception we can usually identify the lack of overtaking much earlier than in the algorithm BP. This means that we do not need to collect and investigate the whole sequence of time windows (that is required in the BP method) but only an initial part of this sequence. In our experiments with the classifier BP-E it was only necessary to check on the average 47% of the whole window sequence for objects from the decision class NO (the lack of overtaking in the time window sequence).

5 Summary

We discussed some rough set tools for perception modelling. They make it possible to recognize behavioral patterns of objects and their parts changing over time. The presented approach is based on behavioral graphs and the ENP method. Note, that the ENP method is only one of the examples of perception methods that we have developed to speed up the identification of complex patterns (e.g., one can consider rules based on the hierarchical domain knowledge representation used for elimination of non-relevant parts of complex objects).

Acknowledgement. The authors gratefully acknowledge the suggestions made by Jim Peters concerning this paper. The research has been supported by the grant 3 T11C 002 26 from Ministry of Scientific Research and Information Technology of the Republic of Poland.

References

1. Anderson J.R. (1993). *Rules of the mind*. Lawrence Erlbaum, Hillsdale, NJ.
2. Bar-Yam Y. (1997). *Dynamics of Complex Systems*. Addison Wesley
3. Bazan J., Skowron A. (2005). Classifiers based on approximate reasoning schemes. In: Dunin-Keplicz B., Jankowski A., Skowron A., and Szczuka M. (Eds.), *Monitoring, Security, and Rescue Tasks in Multiagent Systems MSRAS*, Advances in Soft Computing, Springer, Heidelberg, 191–202.
4. Bazan J., Peters J.F., Skowron, A. (2005). Behavioral pattern identification through rough set modelling. Proceedings of the Tenth International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC 2005), August 31st - September 3rd, 2005, University of Regina, Canada (to appear).
5. Bazan J., Nguyen S. Hoa, Nguyen H. Son, Skowron A. (2004). Rough set methods in approximation of hierarchical concepts. *Proc. of RSCTC'2004*, LNAI **3066**, Springer, Heidelberg, 346–355
6. Kieras D., Meyer D.E. (1997). An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human-Computer Interaction* **12**: 391-438.
7. Langley P., Laird J.E. (2002). Cognitive architectures: Research issues and challenges. *Technical Report*, Institute for the Study of Learning and Expertise, Palo Alto, CA.
8. Laird J.E., Newell A., Rosenbloom P.S. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence* **33**: 1-64.
9. Luck M., McBurney P., Preist Ch. (2003). *Agent Technology: Enabling Next Generation. A Roadmap for Agent Based Computing*. Agent Link.
10. Newell A. (1990). *Unified Theories of Cognition*. Cambridge, Harvard University Press, MA.
11. Nguyen S. Hoa, Bazan J., Skowron A., Nguyen H. Son (2004). Layered learning for concept synthesis. LNAI **3100**, *Transactions on Rough Sets* **I**: 187–208, Springer, Heidelberg,
12. Pal S.K., Polkowski L., Skowron A. (Eds.) (2004). *Rough-Neural Computing: Techniques for Computing with Words*. Cognitive Technologies, Springer, Heidelberg
13. Pawlak Z. (1991). *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, Dordrecht
14. Peters J.F. (2005). Rough ethology: Towards a biologically-inspired study of collective behavior in intelligent systems with approximation spaces. LNAI **3400**, *Transactions on Rough Sets* **III**: 153–174, Springer, Heidelberg
15. The Road simulator Homepage – `logic.mimuw.edu.pl/∼bazan/simulator`
16. The RSES Homepage – `logic.mimuw.edu.pl/∼rses`
17. Veloso M.M., Carbonell J.G. (1993). Derivational analogy in PRODIGY: Automating case acquisition, storage, and utilization. *Machine Learning* **10**: 249-278.
18. Zadeh L.A. (2001). A new direction in AI: Toward a computational theory of perceptions. *AI Magazine*, **22**(1): 73–84.