Behavioral Pattern Identification Through Rough Set Modelling

Jan G. Bazan[1], James F. Peters[2], and Andrzej Skowron[3]

[1] Institute of Mathematics, University of Rzeszów, Rejtana 16A, 35-959 Rzeszów, Poland
bazan@univ.rzeszow.pl

[2] Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, Manitoba R3T 5V6, Canada
jfpeters@ee.umanitoba.ca

[3] Institute of Mathematics, Warsaw University, Banacha 2, 02-097 Warsaw, Poland
skowron@mimuw.edu.pl

Abstract. This paper introduces an approach to behavioral pattern identification as a part of a study of temporal patterns in complex dynamical systems. Rough set theory introduced by Zdzisław Pawlak during the early 1980s provides the foundation for the construction of classifiers relative to what are known as temporal pattern tables. It is quite remarkable that temporal patterns can be treated as features that make it possible to approximate complex concepts. This article introduces what are known as behavior graphs. Temporal concepts approximated by approximate reasoning schemes become nodes in behavioral graphs. In addition, we discuss some rough set tools for perception modeling that are developed for a system for modelling networks of classifiers. Such networks make it possible to recognize behavioral patterns of objects changing over time. They are constructed using an ontology of concepts delivered by experts that engage in approximate reasoning on concepts embedded in such an ontology. This article also includes examples based on data from a vehicular traffic simulator useful in the identification of behavioral patterns by drivers.

Keywords: Behaviorial pattern, concept approximation, dynamical system, graph, rough sets.

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* [1,20], *autonomous multiagent systems* [15,11,13,20], or swarm intelligent systems (see, e.g., [11,13,14,18]). For example, one can consider *road traffic* as a dynamical system represented by road simulator (see,e.g. [16]). While

D. Ślęzak et al. (Eds.): RSFDGrC 2005, LNAI 3642, pp. 688–697, 2005.

driving on a road each vehicle can be treated as *an intelligent autonomous agent*. Each agent is "observing" the surrounding situation on the road, keeping in mind his destination and his own parameters, and makes an independent decision about further steps by performing some maneuver such as passing, overtaking, changing lane, or stopping.

We would like to investigate behavioral patterns of complex objects and their parts changing over time. An identification of some behavioral patterns can be very important for identification or prediction of behaviour of a dynamical system, e.g., some behavioral patterns correspond to undesirable behaviours of complex objects. If in the current situation some patterns are identified (i.e., they are satisfied to a satisfactory degree), then the control module can use this information to tune selected parameters to obtain the desirable behavior of the system. This can make it possible to overcome dangerous or uncomfortable situations. For example, if some behaviour of a vehicle that cause a danger on the road is identified, we can try to change its behaviour by using some suitable means such as road traffic signalling, radio message or police patrol intervention. Note also that the study of collective behavior in intelligent systems is now one of the more challenging research problems (see, e.g., [11,6,12,13]), especially if one considers the introduction of some form of learning by cooperating agents (see, e.g., [18,4,14,5,19]).

The prediction of behavioral patterns of a complex object evaluated over time is usually based on some historical knowledge representation used to store information about changes in relevant futures or parameters. This information is usually represented as a data set and has to be collected during long-term observation of a complex dynamic system. For example, in case of road traffic, we associate the object-vehicle parameters with the readouts of different measuring devices or technical equipment placed inside the vehicle or in the outside environment (e.g., alongside the road, in a helicopter observing the situation on the road, in a traffic patrol vehicle). Many monitoring devices play serve as informative sensors such as GPS, laser scanners, thermometers, range finders, digital cameras, radar, image and sound converters (see, e.g. [20]). Hence, many vehicle features serve as models of physical sensors. Here are some exemplary sensors: location, speed, current acceleration or deceleration, visibility, humidity (slipperiness) of the road. By analogy to this example, many features of complex objects are often dubbed sensors.

In this paper, we discuss some rough set [10] tools for perception modelling that have been developed in our project as part of a system for modelling networks of classifiers. Such networks make it possible to recognize behavioral patterns of objects and their parts changing over time. They are constructed using an ontology of concepts delivered by experts aiming to approximate reasoning on concepts embedded in such an ontology. In our approach we use the notion of *temporal pattern* (see Sect. 2) that express simple temporal features of objects or groups of objects in a given complex dynamic system. Temporal patterns can be used to approximate *temporal concepts* (see Sect. 3), that represent more complex features of complex objects. More complex behaviour of complex objects or groups of complex objects can be presented in the form of *behavioral graphs* (see Sect. 4 and Sect. 5). Any behavioral graph can be interpreted as a *behavioral pattern* and can be used as a complex classifier for recognition of complex behaviours (see Sect. 6). Finally, we present a complete approach to the perception of behavioral patterns that is based on behavioral graphs and so called *dynamic elimination of behavioral patterns* (see Sect. 6).

This paper is a continuation of a research project in which we investigate methods for complex concept approximation using a hierarchical approach (see, e.g., [3,4,2]), and is related to companion research on interactive behavior and learning by swarms of cooperating agents (see, e.g., [11,12,13,14]).

2 Temporal Patterns

In many complex dynamic systems, there are some *elementary actions* (performed by complex objects), that is easily expressed by a local change of object parameters, measured in a very short but a registerable period. So, an elementary action should be understood as a very small but meaningful change of some sensor values such as location, distance, speed. In case of the road traffic example, we distinguish the following elementary actions such as increase in speed, decrease in speed, lane change. However, a perception of composite actions requires analyses of elementary actions performed over a longer period called a *time window*. Therefore, if we want to predict composite actions or discover a behavioral pattern, we have to investigate all elementary actions that have been performed in the current time window. Hence, one can consider the frequency of elementary actions within a given time window and temporal dependencies between them. These properties can be expressed using so called *temporal patterns*. We define a temporal pattern as a function using parameters of an object observed over a time window. In this paper we consider temporal patterns of the following types:

- *sensory pattern*. A numerical characterization of values of selected sensor from a time window (e.g., the minimal, maximal or mean value of a selected sensor, initial and final values of selected sensor, deviation of selected sensor values).
- *local pattern*. A crisp (binary) characterization of occurrences of elementary actions (e.g., action A occurs within a time window, the action B occurs in the beginning of a time window, the action C does not occur within a time window).
- *sequential pattern*. A binary characterization of temporal dependencies between elementary actions inside a time window (e.g., action A persists throughout a time window or action A begins before action B, action C occurs after action D).

One can see that any sensory pattern is determined directly by values of some sensors. For example, in case of the road traffic one can consider sensory patterns such as minimal speed and estimated speed within a time window. The value of a local or sequential pattern is determined by elementary actions registered in a time window. Local or sequential patterns are often used in queries with binary answers such as Yes or No. For example, in case of road traffic we have exemplary local patterns such as “Did vehicle speed increase in the time window?” or “Was the speed stable in the time window?” and sequential patterns such as “Did the speed increase before a move to the left lane occurred?” or “Did the speed increase before a speed decrease occurred?”. We assume that any temporal pattern ought to be defined by a human expert using domain knowledge accumulated for the given complex dynamical system.

3 Approximation of Temporal Concepts

The temporal patterns mentioned in Sect. 2 can be treated as new features that can be used to approximate complex concepts. In this paper, we call them *temporal concepts*. We assume that temporal concepts are specified by a human expert. Temporal concepts are usually used in queries about the status of some objects in a particular temporal window. Answers to such queries can be of the form Yes, No or *Does not concern*. For example, in case of road traffic one can define complex concepts such as “Is a vehicle accelerating in the right lane?”, “Is a vehicle speed stable while changing lanes?”, or ”Is the speed of a vehicle in the left lane stable?”.

The approximation of temporal concepts is defined by classifiers, which are usually constructed on the basis of decision rules. However, if we want to apply classifiers for approximation of temporal concepts, we have to construct a suitable decision table called a *temporal pattern table*. A temporal pattern table is constructed from a table T that stores information about objects occurring in a complex dynamical system. Any row of table T represents information about parameters of a single object registered in a time window. Such a table can be treated as a data set accumulated from observations of the behavior of a complex dynamical system. Assume, for example, that we want to approximate a temporal concept C using table T. Initially, we construct a temporal pattern table PT as follows.

– Construct table PT with the same objects contained in table T.
– Any condition attribute of table PT is computed using temporal patterns defined by a human expert for the approximation of concept C.
– Values of the decision attribute (the characteristic function of concept C) are proposed by the human expert.

Next, we can construct a classifier for table PT that can approximate temporal concept C. Notice that many temporal concepts can be approximated using this approach. Some of these concepts are in some sense more complex than others. Therefore, usually a concept ontology for particular temporal concepts should be provided. The resulting ontology makes it possible to construct approximate reasoning (AR) schemes that can be used to approximate temporal concepts instead of simple (traditional) classifiers (see e.g. [2]).

4 Behavioral Graph for an Object

Temporal concepts defined for objects from a complex dynamical system and approximated by AR-schemes, can be treated as nodes of a graph called a *behavioral graph*, where connections between nodes represent temporal dependencies. Such connections between nodes can be defined by an expert or read from a data set that has been accumulated for a given complex dynamical system. Figure 1 presents a behavioral graph for a single object-vehicle exhibiting a behavioral pattern of vehicle while driving on a road.

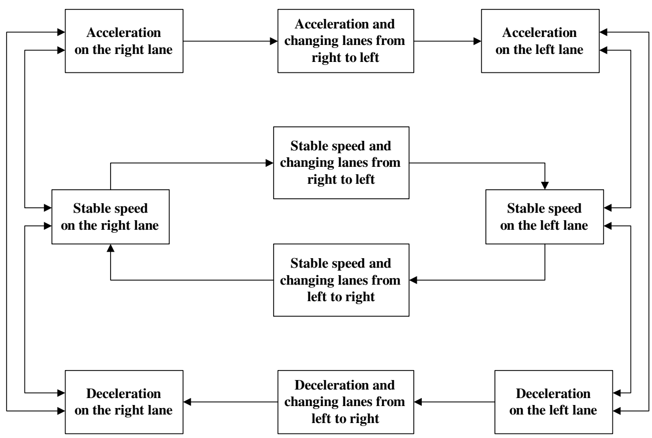

Fig. 1. A behavioral graph for a single object-vehicle

In this behavioral graph, for example, connections between node "Acceleration on the right lane" and node "Acceleration and changing lanes lanes from right to left" indicates that after an acceleration in the right lane, a vehicle can change to the left lane (maintaining its acceleration during both time windows). In addition, a behavioral graph can be constructed for different kinds of objects such as single vehicles or groups of vehicles and defined for behaviors such as driving on the strength road, driving through crossroads, overtaking, and passing. Therefore one can consider any behavioral graph as a model for behavioral patterns.

5 Behavioral Graph for a Group of Objects

In this paper, we introduce a method for approximation of temporal concepts for a group of objects based on what is known as a *group temporal pattern table*, which is constructed using the methodology presented in the Sect. 3 for the construction of the temporal pattern table, but with some important differences. To construct such a table, assume that behavioral graphs for all objects belonging to a group of objects have been constructed. For each behavioral graph we can define new temporal patterns using only two kinds of temporal patterns, namely, local patterns and sequential patterns, since information about sensors is not directly accessible.

5.1 Construction of Group Temporal Pattern Tables

A group temporal pattern table GT is constructed in the following way. Any row-object in GT is created using for any object in group G a path of nodes observed in a given behavioral graph of the object. The path in any behavioral graph should be understood as a sequence of graph nodes (temporal concepts) registered over some period, i.e., over a number of time windows. For any such path, a vector of values is computed (a vector of values of condition attributes of table GT) using temporal patterns provided by the expert for the approximation of a concept C. Additional attributes are defined using behavioral graphs of all objects simultaneously. These attributes describe temporal relations between objects in group G.

The temporal concepts defined for group of objects and approximated by AR-schemes, can be treated as nodes of a new graph, that we call as *a behavioral graph for a group of objects*. One can say, that the behavioral graph for a group of objects expresses temporal dependencies on a higher level of generalization. On lower level behavioral graphs are expressing temporal dependencies between single objects (or simpler groups of objects).

In Figure 2 we present exemplary behavioral graph for group of two objects-vehicles: vehicle A and vehicle B, related to the standard overtaking pattern. There are 6 nodes in this graph, that represent following behavioral patterns: vehicle A is driving behind B on the right lane, vehicle A is changing lanes from right to left, vehicle A is moving back to the right lane, vehicle A is passing B (when A is on the left lane and B is on the right lane), vehicle A is changing lanes from left to right and vehicle A is before B on the right lane. There are 7 connections represented spatio-temporal dependencies between behavioral patterns from nodes. For example after the node "Vehicle A is driving behind B on the right lane" the behaviour of these two vehicles can much to the pattern "Vehicle A is changing lanes from right to left and B is driving on the right lane".

6 Behavioral Patterns

In perceiving complex behaviour by individual objects or by a group of objects over a long period of time, it is possible to construct behavioral graphs to codify our observations. Such graphs facilitate observations about transitions between nodes of behavioral graph and registering a sequence of nodes that form paths in temporal patterns. If the path of temporal patterns matches a path in a behavioral graph, we conclude that the observed behaviour is compatible with the behavioral graph. In effect, we can use a behavioral graph as a complex classifier for perception of the complex behaviour of individual objects or groups

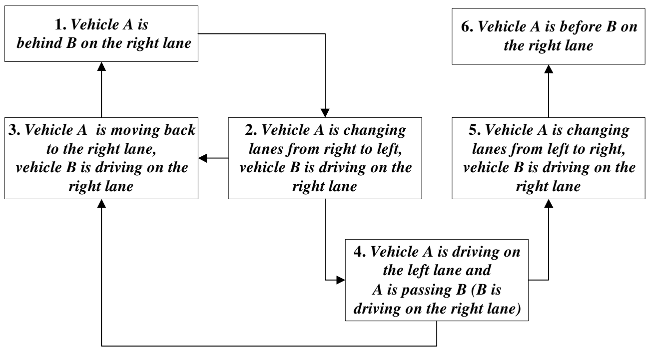

Fig. 2. A behavioral graph for the maneuver of overtaking

of objects. For this reason, a behavioral graph constructed for some complex behavior is called a *behavioral pattern*.

As an example, let us study the behavioral graph presented in Figure 2 for a group of two objects-vehicles (vehicle A and vehicle B) related to the standard overtaking pattern. We can see that the path of temporal patterns with indexes "1, 2, 3, 1, 2, 4" is matching a path from this behavioral graph, while the path with indexes: "6, 5, 4" is not matching any path from this behavioral graph (this path can match some other behavioral patterns).

A path of temporal patterns (that makes it possible to identify behavioral patterns) should have a suitable length. In the case where the length is too short, it may be impossible to discern one behavioral pattern from another pattern. For example, we can make a mistake between an overtaking a passing by a vehicle in traffic.

7 Perception of Behavioral Patterns

The construction of temporal windows is based on the notion of a temporal pattern. 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 time period that is typical for a given temporal pattern t_i. The second attribute of the table DT_i (the decision attribute of this table) is computed using the temporal pattern t_i. By contrast, a condition attribute registers the index of some behavioral pattern from the family BP. This index can be obtained by using some complex classifier created for some behavioral pattern from the family BP, because any complex classifier from the family BP can check single temporal window (and its time neighbourhood) whether the investigated group of objects matches to a 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 (see [17]). Any computed decision rule expresses a dependence between information about matching to some behavioral pattern and information about matching to some temporal pattern. For example, consider the problem of recognition of overtaking that can be understand as a behavioral pattern defined for a 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 in the right lane`. After usage of a transposition low, we can obtain the following rule: `If the vehicle B is not driving in the right lane then the vehicle A is not overtaking B`. The last rule serves an aid to fast verification whether the behavior of a group of objects (e.g., vehicles A and B) matches the behavioral pattern of overtaking. This method only allows us to eliminate some behavioral patterns. After this elimination the complex classifiers based on a suitable behaviour graphs should be applied.

8 Experimental Results

To verify the effectiveness of classifiers based on behavioral patterns, we have implemented the algorithms in a *Behavioral Patterns* library (BP-lib), which is an extension of the RSES-lib 2.1 library forming the computational kernel of the RSES system [17]. The experiments have been performed on the data sets obtained from the road simulator (see [16]). We have applied the "train and test" method for estimating accuracy. A training set consists of 17553 objects generated by the road simulator during one thousand of simulation steps. Whereas, a testing set consists of 17765 objects collected during another(completely different) session with the road simulator.

In our experiments, we compared the quality of three classifiers: *Rough Set classifier with decomposition* (RS-D), *Behavioral Pattern classifier* (BP) and *Behavioral Pattern classifier with the dynamic elimination of behavioral patterns* (BP-E). For induction of RS-D, we employed RSES system generating the set of minimal decision rules that are next used for classification of situations from the testing data. However, we had to use the method of generating decision rules joined with a standard decomposition algorithm from the RSES system. This was necessary because the size of the training table was too large for the directly generating decision rules (see [17]). The classifiers BP and BP-E are based on behavioral patterns (see Sect. 6) but with application of dynamic elimination of behavioral patterns related to the investigated group of objects. We compared RS-D, BP and BP-E using accuracy of classification. Table 1 shows the results of applying these classification algorithms for the concept related to the *overtaking* behavioral pattern.

One can see that in case of perception of the overtaking maneuver (decision class Yes) the accuracy and the real accuracy (*real accuracy* = *accuracy* × *coverage*) of algorithm BP are higher than the accuracy and the real accuracy of algorithm RS-D for the analyzed data set. Besides, we see that the accuracy of algorithm BP-E is only 4 percent lower than the accuracy of algorithm BP. Whereas, the algorithm BP-E allows us to reduce the time of perception, because during perception we can usually identify the lack of overtaking earlier than in the algorithm BP. This means that we have not to collect and investigate the

Table 1. Results of experiments for the overtaking pattern

Decision class	Method	Accuracy	Coverage	Real accuracy
Yes (overtaking)	RS-D	0.800	0.757	0.606
	BP	0.923	1.0	0.923
	BP-E	0.883	1.0	0.883
No (no overtaking)	RS-D	0.998	0.977	0.975
	BP	0.993	1.0	0.993
	BP-E	0.998	1.0	0.998
All classes (Yes + No)	RS-D	0.990	0.966	0.956
	BP	0.989	1.0	0.989
	BP-E	0.992	1.0	0.992

whole sequence of time windows (that is required in the BP method) but only some first part of this sequence. In our experiments with the classifier BP-E it was at an average 47 percent of the whole sequence for objects from the decision class No (the lack of overtaking in the time window sequence).

9 Conclusion

In this paper, we discussed some rough set tools for perception modelling that make it possible to recognize behavioral patterns of objects and their parts changing over time. We have presented the complete approach to the perception of behavioral patterns, that is based on behavioral graphs and the dynamic elimination of behavioral patterns. Some of the tools mentioned in the paper are already implemented in our system and may be tested (see e.g. [17]). In our further work we would like to develop a complete software environment to model the perception of behavioral patterns.

Acknowledgement

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. This research has also been supported by Natural Sciences and Engineering Research Council of Canada (NSERC) grant 185986 and grant T247 from Manitoba Hydro.

References

1. Bar-Yam, Y.: Dynamiocs of Complex Systems. Addison Wesley (1997)
2. Bazan, J., Skowron, A.: 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 (2005) 191–202

3. Bazan, J., Nguyen, S.H., Nguyen, H.S., Skowron, A.: Rough set methods in approximation of hierarchical concepts. In: Proc. of RSCTC'2004. Lecture Notes in Artificial Intelligence 3066, Springer, Heidelberg (2004) 346–355
4. Nguyen, S.H., Bazan, J., Skowron, A., Nguyen, H.S.: Layered learning for concept synthesis. Lecture Notes in Artificial Intelligence 3100, Transactions on Rough Sets, **I**, Springer, Heidelberg (2004) 187–208
5. Birattari, M., Di Caro, G., Dorigo, M.: Toward the formal foundation of ant programming. In: M. Dorigo, G. Di Caro, M. Sampels (Eds.), Ant Algorithms. Lecture Notes in Computer Science 2463, Springer-Verlag, Berlin (2002) 188–201
6. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence. From Natural to Artificial Systems. Oxford University Press, UK (1999)
7. Fahle, M., Poggio, T. (Eds.): Perceptual Learning. The MIT Press, Cambridge, MA (2002)
8. Harnad, S. (Ed.): Categorical Perception. The Groundwork of cognition. Cambridge University Press, UK (1987)
9. Pal, S.K., Polkowski, L., Skowron, A. (Eds.): Rough-Neural Computing: Techniques for Computing with Words. Cognitive Technologies. Springer, Heidelberg (2004)
10. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Volume 9 of System Theory, Knowledge Engineering and Problem Solving. Kluwer Academic Publishers, Dordrecht, The Netherlands (1991)
11. Peters, J.F.: Rough Ethology: Towards a Biologically-Inspired Study of Collective Behavior in Intelligent Systems with Approximation Spaces. Transactions on Rough Sets, **III**, LNCS 3400 (2005) 153–174
12. Peters, J.F.: Approximation spaces for hierarchical intelligent behavioral system models. In: Dunin-Keplicz, B., Jankowski, A., Skowron, A., and Szczuka, M. (Eds.), Monitoring, Security and Rescue Techniques in Multiagent Systems. Advances in Soft Computing, Heidelberg, Physica-Verlag (2004) 13–30
13. Peters, J.F., Henry, C., Ramanna, S.: Rough Ethograms: Study of Intelligent System Behavior. In: New Trends in Intelligent Information Processing and Web Mining (IIS05), Gdańsk, Poland, June 13-16 (2005)
14. Peters, J.F., Henry, C., Ramanna, S.: Reinforcement learning with pattern-based rewards. In: Computational Intelligence (CI), Calgary, Alberta, Canada, July (2005) [to appear]
15. Luck, M., McBurney, P., Preist, Ch.: Agent Technology: Enabling Next Generation. A Roadmap for Agent Based Computing. Agent Link (2003)
16. Road simulator Homepage at `http://logic.mimuw.edu.pl/∼bazan/simulator`
17. RSES Homepage at `http://logic.mimuw.edu.pl/∼rses`
18. Dorigo, M., Birattari, M., Blum, C., Gambardella, L.M., Monada, F., Stutzle, T. (Eds.): Ant Colony Optimization and Swarm Intelligence. Lecture Notes in Computer Science 3172, Springer-Verlag, Berlin (2004)
19. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. Cambridge, MA: The MIT Press (1998)
20. Urmson, C., et al.: High speed navigation of unrehearsed terrain: Red team technology for Grand Challenge 2004. Report CMU-RI-TR-04-37, The Robotics Institute, Carnegie Mellon University (2004)
21. Zadeh, L.A.: A new direction in AI: Toward a computational theory of perceptions. AI Magazine, 22(1) (2001) 73–84