“As time goes by” - Using time series based decision tree induction to analyze the behaviour of opponent players

Christian Drücker, Sebastian Hübner,
Ubbo Visser, Hans-Georg Weland

TZI - Center for Computing Technologies
University of Bremen
D-28334 Bremen, Germany
grp-rcintern@tzi.de
WWW home page: http://www.virtualwerder.de/

Abstract. With the more sophisticated abilities of teams within the simulation league high level online functions become more and more attractive. Last year we proposed an approach to recognize the opponents strategy and developed the online coach accordingly. The coach was able to detect their strategy and then passed this information together with appropriate countermeasures to his team. However, this approach gives only information about the entire team and is not able to detect significant situations (e.g. double pass, standard situations, repeated patterns between two or three players). In this paper we describe a new method for time series which is able to describe the time series by qualitative abstraction and produces samples which then can be used for inductive learning methods such as decision trees.

1 Introduction

As the RoboCup 2000 in Melbourne showed, the differences between the technical abilities of the major teams aren't as big as in previous world-cups. Due to teams which share their progress with the RoboCup community by releasing their source code shortly after the event - a big thanks to CMU at this point -, every team is able to easily arrange eleven pretty good agents. Future teams will have to develop a superior team behaviour to win a tournament. This means a big step forward in terms of multiagent-system research.

Version 7.0 of the soccer server extended the abilities of the online coach further, so that its use becomes even more interesting. It is still the most effective instrument to analyze the opponent, because it possesses all the information about the simulated environment. Therefore, it was important for us to continue the development of the online coach which we used at the RoboCup 2000. In [Visser et al., 2001] we describe how the old coach determines the opponent tactical formation with a neural network and how it is able to change our team formation during a match.

This year our main target is the collection of more information about the opposing team. On one hand we want to recognize more aspects of the opponents global tactics. Beside of their formations, we are interested in their defense behaviour (e.g. the use of offside traps; cover or zone defense) and their preferred offensive plays (e.g. wing play or center breakthroughs; quick passing or long dribbling), but also in their ability to cope with such behaviours when they are used by our team.

On the other hand we want to analyze an individual opponent agent to detect which task in their system it fulfills. To determine this, we have to observe how a certain player reacts to certain situations. Under which circumstances does an agent pass the ball to a fellow player? Why does it pass the ball in this situation to just this player? When does an opponent player shoot at our goal? When does a defense player attack our ball carrier?

If we know how an opponent agent reacts in a certain situation, we can use this information to be always one step ahead of the opponent. For example, if we realize that the opponent forwards only try to score if they have enough space (as it seemed to be performed by the Brainstormers agents), we could prevent them from shooting at the goal by placing our defenders close enough to them.

Similar work has been done over the last couple of years by [Wünstel et al., 2000], [Frank et al., 2000], and [Raines et al., 1999].

Our objective is to find a method that can improve the behaviour of our team during the game and that not only shows us how we could do better next time. Every year new teams debut in RoboCup whose agents differ in their behaviours from all that existed before. Therefore, off-line analysis can prepare our team for existing agents only, whose binaries have been released during the year. Unfortunately only a few teams did that early enough in the past. But even if all would, we couldn't assume that our opponents didn't made major changes in their behaviour.

2 Qualitative abstraction of time series

A new method for the qualitative abstraction of multiple time series has been developed at the Center for Computing Technologies [Boronowsky, 2001]. This method seems especially suitable to us in order to deliver patterns which we then use as a basis for learning algortihms. The method has been developed to analyze continuous valued time series. It uses decision tree induction and therefore generates rules about the analyzed time series, in our case the observed players. The special feature of this method is the way the continuous valued attributes are discretized.

Certain rules, generated by the decision tree, can be evaluated automatically by our coach. This information can be used to improve our agents' behaviour in the running game. Additionally, all rules can can be evaluated by hand after the game has ended. The results of the evaluation by hand can be used to improve our agents' basic behaviour. A big advantage of the used method is, that the discretization of continuous valued attributes is done automatically by our system.

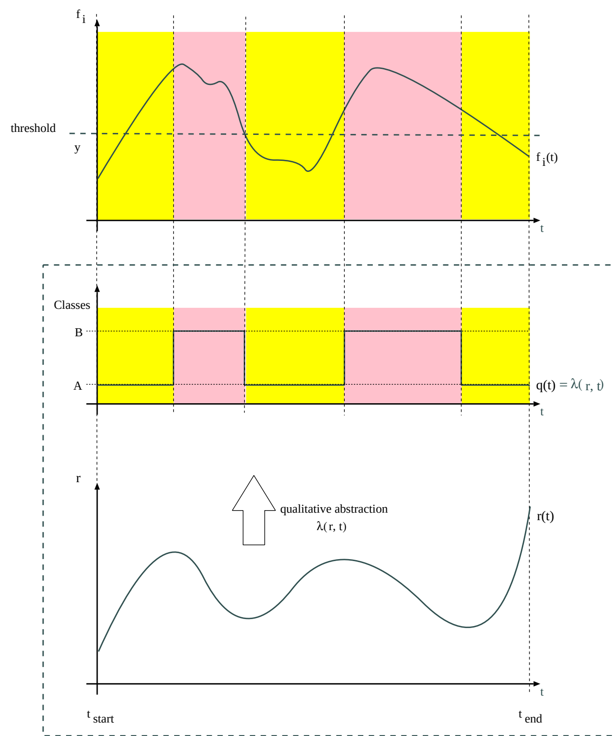

Fig. 1. Qualitative abstraction and horizontal splitting

Thus, there will be no problem with the decision tree induction not generating a proper result, if an adverse discretization has been chosen by the designer. The discretizations resulting from the method particularly hold valuable information about the opposing team. Discretization of continuous valued time series is the main task of the method. The method uses an very efficient algorithm to solve this problem.

To analyze our opponent we record several time series F. Supplementary mathematical transformed time series can be added in order to improve the results. One of the time series $r \in F$ must be qualitatively abstracted. This special series gives the classes to be learned, e.g. a time series containing: *goalie leaves goal*, *goalie stays in goal*, and *goalie returns to goal*. By the qualitatively abstracted time series time slices on all time series are assigned to the classes. One of the time series f_i is split horizontally at a threshold y (fig. 1). To determine this threshold all possible split points must be evaluated with a special heuristic. The best split point is the one which separates the different classes the best.

All the other time series are split vertically (fig. 2) at all points at which the time series f_i crosses the threshold y. With these separated graphs the process is recursively repeated.

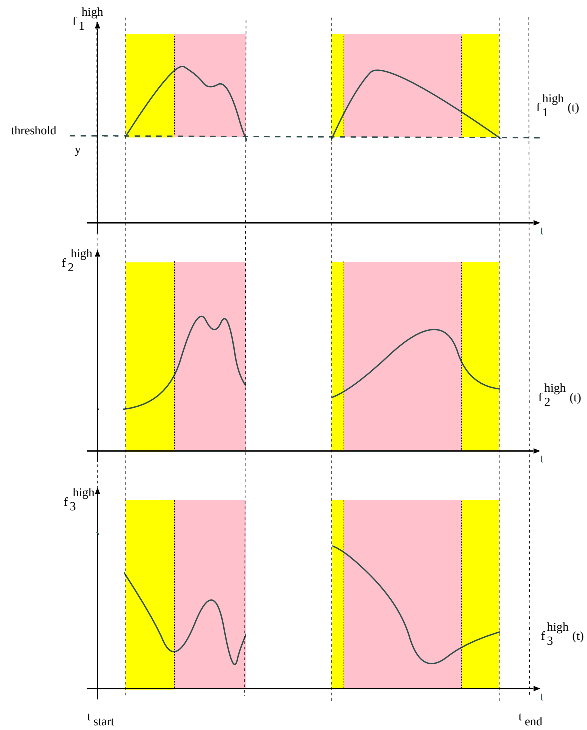

Fig. 2. vertical splitting

To apply the uniform distribution theorem of the entropy minimalization heuristic the time series must be approximated by partially linear functions.

The used method is based on the entropy minimalization heuristic as used in ID3 [Quinlan, 1986] and C4.5 [Quinlan, 1993]. An attribute is split at the point at which the heuristic is minimal.

Fayyad and Irani [Fayyad and Irani, 1992] have proved that this can only happen at class boundaries, we call these points boundary points (BP). By using this knowledge an important increase in efficiency can be reached. But it only works properly for non-overlapping classes. When two classes are overlapping the number of BPs can increase dramatically. In the worst case there can be BP between all examples in the overlapping area.

The method we use in this work allows an efficient splitting of continuous valued attributes in these cases. Boronowsky [Boronowsky, 2001] shows that this problem can be solved by using intervals of uniform distribution. This can be achieved by a linear approximation of the real distribution. He also explains that the entropy minimalization heuristic can only be minimal at the joints of the approximation.

So far the method was used in the system ExtraKT. ExtraKT is a system to aid humans in analyzing time series. It generates hypotheses about coherences in time series which can then be interpreted by a human analyst. The user can also manipulate the decision tree induction to enter his own knowledge into the process.

The system was used to analyze simulated technical systems. In these tests also sudden occurring failures in the technical systems were simulated. ExtraKT didn't know about these errors beforehand, but they were represented in the decision tree and could easily be found by the user. This aspect is very interesting for the use in the RoboCup competition. In a RoboCup match changes in an agent's behaviour can not only happen due to an error in the agent, but also because the team has changed its tactic.

3 New Coach

In order to improve our abilities to analyze the opposing team we integrated the system into the online coach. The coach continuously tracks certain aspects of the game, e.g. the positions of the players, their distance to each other and their distance to the ball. They are stored in memory forming several time series which are analyzed in regular intervals by the system mentioned above. As a result we receive rules about the opponents behaviour. Some of these rules can be used instantly within the game to improve the behaviour of our players and to adapt to the opponent. To achieve this the coach must have some a priori knowledge about the rules to expect and the suitable changes to our behaviour or playing style.

Additionally, there is the possibility to extract the collected data and rules at the end of the game. This information can be revised and corrected by a human expert. They can then be used for further development of the team or to improve the automatic extraction of rules by the coach.

In the first step we use the described method to analyze the opponents goal keeper. The goalie is particulary suitable, because his role in the team is fixed. He is the only player who's function is known from the beginning of the game; all the other players can only be classified as forwards, mid-fielders or defenders by their behaviour in the game, so we focus on the keeper for our initial work. Additionally, the goalie is a very important player. The knowledge of his strengths and weaknesses can be used to optimize the behaviour and configuration of our forwards, leading to better scoring abilities.

We also want to use the method to analyze our own goal keeper. This way we can test the quality of our system by cross checking the results with the details of our implementation. Besides that we may find some aspects in the behaviour of our goalie which could be improved further.

To prepare the analysis our coach stores some time series of game aspects which are related to the goal and the goal keeper. This includes

- the distance between the ball and the goal,

– the distance between the ball and the goal keeper,
– the distance between the goal keeper and the goal,
– the number of opponents and team mates within the penalty area,
– the number of team mates which may intercept the ball when kicked towards the goal.

For us the most important task is to determine, under which circumstances the goal keeper starts to leave the goal to get the ball and when he starts to return to it. According to this problem we have chosen a suitable qualitative abstraction. This abstraction uses the change in the time series corresponding to the distance of the goal keeper to the goal. This is used to determine whether the goal keeper is leaving the goal, is staying in the goal or is returning to the goal

The situations in which the keeper is leaving the goal or is returning to it are of special interest for us, because in these moments he can be taken by surprise more easily.

We plan to modify and extend the methods used to analyze the goal keeper to the other parts of the team in the next step. Then we will be able to improve the ability of all players of our team to adapt to the behaviour of the opponent. The forwards appear to be particular interesting to us. By finding special rules of attacks we can strengthen our defense by using suitable formations and behaviours.

4 Usage of the collected information

After collecting those information, we use them to improve our performance. The off-line usage is rather simple, if we store the data in a human readable form.

During a match the coach has to broadcast its recognitions. As some of the information surely can't be expressed in the standardized language of the coaches, we'll still have to wait for an interruption of the game, to send a message. The communication works just like the change of our formation, which we described in Visser et al. [2001] . The coach generates an announcement, which every agent can parse.

If the coach recognizes that the opponent goalie leaves his goal every time when there is none of his defenders between our ball carrier and him, he could instruct our forwards to pull him to the wings and pass the ball to an unguarded teammate in the center. If the goalie only leaves the goal if the forward achieves a certain distance to his goal, our agents should try to score from about this distance. If the goalie extremely shifts towards the ball during our attacks, quick wing plays should be effective.

Fig. 3 shows an 2-vs-1 situation (the three defenders beneath don't really affect the play). The goalie decides to attack the ball carrier, so as the forward passes the ball to his teammate, he gets a big opportunity to score. If we assume that the decision of the goalie is deterministic, he should react in the same way

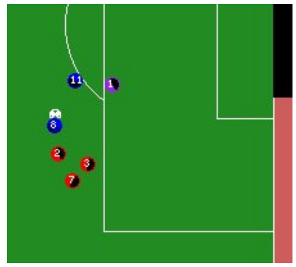

Fig. 3. 2-vs-1 situation

during the next 2-vs-1 situation. The coach should instruct the agents, that the supporting player should depart a little more, to get in an even better scoring position.

If the goalie had suspected and intercepted the pass, the coach would have told the agents that the ball carrier should dribble a little longer next time.

The counter actions of the coach have to be taught by a human instructor. They have to be parameterizable, so not every possible situation has to be taught manually. E.g. a goalie, which leaves his goal at a little different distance to the ball, shouldn't cause a complete different behaviour of our agents.

5 Results

As the described approach is a brand new method and has just been released we have to admit that there are only a few results so far. Currently, we are developing an interface for the RoboCup environment such that the method is provided with the time series information during a game. First test have shown that the approach is very promising, especially because of the ability to discretize automatically. We expect the method to provide the online coach with information such as propositional rules, e.g. *if the forward is closer than 10m and the forward has the ball then the keeper is approaching the ball*. In the final version of this paper we will show the results and discuss them in detail.

Acknowledgement

We would like to thank Michael Boronowsky for the work he has done helping us getting started with his method.

References

[Boronowsky, 2001] Boronowsky, M. (2001). Diskretisierung reellwertiger attribute mit gemischten kontinuierlichen gleichverteilungen und ihre anwendung bei der zeitreihenbasierten entscheidungsbauminduktion. In *DISKI*, volume 246, St. Augustin. infix Verlag.

[Fayyad and Irani, 1992] Fayyad, U. and Irani, K. (1992). On the handling of continuous-valued attributes in decision tree generation. In *Machine Learning*, volume 8, pages 87–102.

[Frank et al., 2000] Frank, I., Tanaka-Ishii, K., Arai, K., and Matsubara, H. (2000). The statistics proxy server. In *The Fourth International Workshop on RoboCup*, pages S.199–204.

[Quinlan, 1986] Quinlan, J. (1986). Induction of decision trees. In *Machine Learning*, volume 1.

[Quinlan, 1993] Quinlan, J. (1993). *C4.5 Programs for Machine Learning*. Morgan Kaufmann.

[Raines et al., 1999] Raines, T., Tambe, M., and Marsella, S. (1999). Automated assistants to aid humans in understanding team behabiors. In *Proceedings of the Third International Workshop on Robocup*, pages S.85–104.

[Visser et al., 2001] Visser, U., Drcker, C., Hbner, S., Schmidt, E., and Weland, H.-G. (2001). Recognizing formations in opponent teams. In *RoboCup-00, Robot Soccer World Cup IV*, Lecture Notes in Computer Science, Melbourne, Australia. Springer-Verlag. to appear.

[Wünstel et al., 2000] Wünstel, M., Polani, D., Uthmann, T., and Perl, J. (2000). Behavior classification with self-organizing maps. In *The Fourth International Workshop on RoboCup*, pages S.179–188.