Automatically Evaluating the Status of an RTS game

Sander Bakkes S.BAKKES@MICC.UNIMAAS.NL
Universiteit Maastricht / MICC-IKAT, P.O. Box 616, NL-6200 MD Maastricht, The Netherlands

Philip Kerbusch P.KERBUSCH@STUDENT.UNIMAAS.NL
Pieter Spronck P.SPRONCK@MICC.UNIMAAS.NL
Jaap van den Herik HERIK@MICC.UNIMAAS.NL

Abstract

One of the most challenging tasks when creating an adaptation mechanism is to transform domain knowledge into an evaluation function that adequately measures the quality of the generated solutions. The high complexity of modern video games makes the task to create a suitable evaluation function for adaptive game AI even more difficult. Still, our aim is to fully automatically generate an evaluation function for adaptive game AI. This paper describes our approach, and discusses the experiments performed in the RTS game SPRING. TD-learning is applied for establishing a unit-based evaluation term. In addition, we define a term that evaluates tactical positions. From our results we may conclude that an evaluation function based on the defined terms is able to predict the outcome of a SPRING game reasonably well. That is, for a unit-based evaluation the evaluation function is correct in 76% of all games played, and when evaluating tactical positions it is correct in 97% of all games played.

1. Introduction

Modern video games present a complex and realistic environment in which game AI is expected to behave realistically (i.e., 'human-like'). One feature of human-like behaviour of game AI, namely the ability to adapt adequately to changing circumstances, is called 'adaptive game AI'. When implementing adaptive game AI, arguably the most important factor is the evaluation function that rates the quality of newly generated game AI. Due to the complex nature of modern video games, the determination of a suitable evaluation function is often a difficult task. This paper discusses our work on fully automatically generating a good evaluation function for real-time strategy (RTS) games. In the present research, we use the open-source RTS game SPRING.

Our approach to adaptive game AI is by allowing computer-controlled players to imitate human players. This approach can be particularly successful in games that have access to the Internet and that store and retrieve samples of gameplay experiences (Spronck, 2005). For this approach to be feasible, a central data store of gameplay samples must be created. Game AI can utilise this data store for two purposes: (1) to establish an evaluation function for games, and (2) to be used as a model by an adaptation mechanism.

2. Evaluating the Status of an RTS game

SPRING only has access to feature data of those parts of the environment that are visible to its own units (i.e. an 'imperfect-information environment'). When we allow game AI to access all information we call this a 'perfect-information environment'.

Our evaluation function for the game's status consists of two terms. One term represents the 'number of units observed of each type', the other term represents the 'safety of tactical positions'. The weight assigned to the first term is determined by a free parameter $p \in [0...1]$. For the weights of the second term $1 - p$ is used. To deal with the imperfect information inherent in the SPRING environment, imperfect feature-data is mapped to a prediction of the perfect feature-data. A straightforward approach to this effect is to scale linearly the number of observed units to the non-observed region of the environment. If enemy units are homogenously distributed over the environment, the evaluation function applied to an imperfect informa-

Proceedings of the 18th Benelearn
P. Adriaans, M. van Someren, S. Katrenko (eds.)
Copyright © 2007, The Author(s)

tion environment will produce results close to those in a perfect information environment.

We used TD-learning to establish appropriate weights for all unit types. Our application of TD-learning is similar to its application by Beal and Smith (1997) for determining piece values in chess. Unit-type weights were learned from feature data collected with perfect information of 700 games where an AI was playing against four different computer-controlled opponents.

3. Experiments

To evaluate the performance of the evaluation function, we determined to what extent it is capable of predicting the actual outcome of a SPRING game. We defined the performance measure '*absolute prediction*' as the percentage of games of which the outcome is correctly predicted just before the end of the game. In a perfect-information environment we obtained an average absolute-prediction performance of 76% for a unit-based evaluation ($p = 1$), 97% for an evaluation of tactical positions ($p = 0$), and 71% for a combined evaluation ($p = 0.5$). These results indicate that the evaluation function provides an effective basis for evaluating a game's status. In an imperfect-information environment, the evaluation function obtained a performance comparable to that in a perfect-information environment.

We defined the measure '*weak relative prediction*' (*normal* / *strong*) as the game time at which the outcome of all tested games is predicted correctly at least 50% (60% / 70%) of the time. We observe that for all values of p the weak relative prediction performance is on average 54% (51%) in a perfect-information (imperfect-information) environment. The normal relative prediction performances are on average 67% (81%) in a perfect-information (imperfect-information) environment, and the strong relative prediction performances are on average 78% (96%) in a perfect-information (imperfect-information) environment. This indicates that the evaluation function in a perfect-information environment manages to predict the outcome of a game considerably earlier than in an imperfect-information environment. We observed that the term that evaluates tactical positions is particularly effective in the final phase of playing the game, and less effective in earlier phases.

4. Conclusions and Future Work

We discussed an approach to automatically generating an evaluation function for game AI in RTS games. From our experimental results, we may conclude that

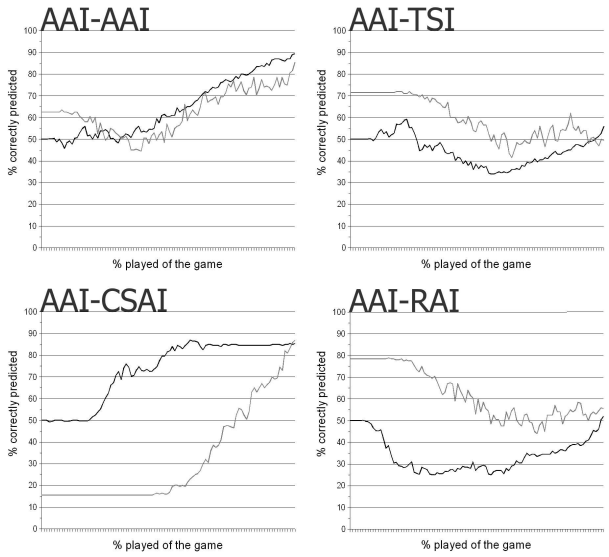

Figure 1. Outcomes correctly predicted as a function over time. The black line represents the prediction performance in a perfect-information environment, the gray line that in an imperfect information environment, for $p = 0.5$.

the generated evaluation function effectively predicts the outcome of a SPRING game, expressed in terms of the absolute-prediction performance. The relative prediction performance, which indicates how early in a game an outcome is predicted correctly, is lower (i.e., better) in a perfect-information environment. Two terms were defined to evaluate a game's status, namely a unit-based term and a term based on tactical positions. The term to evaluate tactical positions is capable of predicting the final outcome of the game almost perfectly. However, it only predicts accurately in the final phase of the game. The unit-based term, on the other hand, is only moderately accurate in predicting the final outcome of the game. However, it achieves a high prediction accuracy relatively early. This implies that the accuracy of outcome predictions is closely related to the phase of the game. Thus, to improve performance, the weights assigned to each term of the evaluation function should be made dependent on the phase of the game.

References

Beal, D., & Smith, M. (1997). Learning piece values using temportal differences. *International Computer Chess Association (ICCA) Journal, 20(3)*, 147–151.

Spronck, P. (2005). A model for reliable adaptive game intelligence. *Proceedings of the IJCAI-05 Workshop on Reasoning, Representation, and Learning in Computer Games.*