Games with Dynamic Difficulty Adjustment using POMDPs

Robby Goetschalckx ROBBY@COMPUTING.DUNDEE.AC.UK
School of Computing, University of Dundee, Dundee, UK

Olana Missura OLANA.MISSURA@IAIS.FRAUNHOFER.DE
Fraunhofer IAIS, Schloss Birlinghovenm Sankt Augustin, Germany

Jesse Hoey JESSEHOEY@COMPUTING.DUNDEE.AC.UK
School of Computing, University of Dundee, Dundee, UK

Thomas Gärtner THOMAS.GAERTNER@IAIS.FRAUNHOFER.DE
Fraunhofer IAIS, Schloss Birlinghovenm Sankt Augustin, Germany

Abstract

In this paper the approach of using a partially observable Markov model for games with dynamical difficulty adjustment is introduced. This approach leads implicitly to a strategy which balances gathering information about the player through his or her behavior with adjusting the game to the estimated player's abilities and preferences. We will show how this approach can be used in a stroke rehabilitation system, where a person plays a game in which the controller is a rehabilitation device. We show that the parameters of the model have a clear influence on the behavior of the system and that aspects of the player's abilities and characteristics can be measured by observing the behavior.

1 Introduction

Games which adapt themselves so as to maximize the enjoyment of the user are currently the topic of much research. The difficulty, the rules or specifications of items in the game could be changed, giving every user a personalized experience. A taxonomy for such games has recently been suggested by Togelius et al. (2010). In the ideal case, the game would measure the characteristics of the user during play and adapt dynamically.

There are many other domains where characteristics of a task can be adapted dynamically to the user. For example, in the work by Patricia Kan, Jesse Hoey, Alex Mihailidis (2008) the difficulty of a rehabilitation exercise is dynamically adapted to the estimated abilities and fatigue level of the person, while in the work by Boger et al. (2005) the user is a person suffering from dementia and the task is washing hands, with different types of aid from the system; here the choice of actions needs to be tuned to the actual cognitive capacities. In these specific instances, the whole of user, system and task was modeled as a *partially observable Markov decision process* or POMDP, a model for sequential decision making where certain aspects (e.g. abilities, fatigue) are not directly observable but only indirectly through the person's behavior. This approach of modeling the user and system as one POMDP and then solving it, which gives a strategy to dynamically find the user's abilities and the optimal behavior of the system, could also be used in the setting of adaptive games, where the ability level and specific interests of the player are not directly observable. This is the approach we will present in this paper. More specifically we will investigate the stroke rehabilitation system, where the rehabilitation is achieved by letting the person play a computer game, using a controller which has adaptable resistance. We will show some results of simulations using this system where the behavior of the system and the estimates of the capabilities of the person are measured for different settings.

The rest of the paper is structured as follows. In Section 2 related work in games with dynamic difficulty adjustment is discussed. Section 3 introduces the POMDP approach to dynamically adjusting systems (with some examples from other fields) and explains how this could be used for games. The variable aspects of a player in a game and how these can be estimated are discussed in Section 4. Section 5 shows some preliminary results of simulations in the stroke rehabilitation setting. We conclude with Section 6.

Appearing in *Proceedings of the ICML Workshop on Machine Learning and Games*.

2 Dynamic Difficulty Adjustment in Games

One particular way in which games could adjust themselves to their players is by dynamically changing the perceived difficulty of the challenges that the player is facing. There is a connection between the achievements of the player and the enjoyment that the player feels. Biederman and Vessel (2006) postulates that our brains are physiologically driven by a desire to learn something new: new skills, new patterns, new ideas. We have an instinct to play, and this instinct is inherently connected with acquiring new knowledge. There is an internal reward mechanism for each new mastered skill or gained knowledge: the feeling of joy. The games create additional rewards for their players such as new items available, new areas to explore. At the same time there are new challenges to overcome, new goals to achieve, and new skills to learn, which creates a loop of learning-mastery-reward and keeps the player involved and engaged.

Thus, an important ingredient of the games that are fun to play is providing the players with the challenges corresponding to their skills. It appears that an inherent property of any challenge (and of the learning required to master it) is its difficulty level. Here the difficulty is a subjective factor that stems from the interaction between the player and the challenge. The perceived difficulty is also not a static property: It changes with the time that the player spends learning a skill.

Note that while the perceived difficulty depends on the abilities of the player, at the same time the ability to learn the skill and the speed of the learning process are also controlled by how difficult the player perceives the task. If the bar is set too high and the task appears too difficult, the player will end up frustrated and will give up on the process in favour of something more rewarding. Then again if the challenge turns out to be too easy (meaning that the player already possesses the skill necessary to deal with it) then there is no learning involved, which makes the game appear boring.

It becomes obvious that the game should provide the challenges for the player of the "right" perceived difficulty: The one that stimulates the learning without pushing the players too far or not far enough. Ideally then, the difficulty of any particular instance of the game should be determined by who is playing it at this moment.

Both artificial intelligence researchers and the game developers community display an interest in the problem of automatic difficulty scaling. Different approaches can be seen in the work of Hunicke and Chapman (2004), Herbrich et al. (2006), Danzi et al. (2003), and others. As can be seen from these examples the problem of dynamic difficulty adjustment in video games was attacked from different angles, but a unifying approach is still missing. Player modeling in computer games is a relatively new area of interest for the researchers. Nevertheless, existing work (Yannakakis & Maragoudakis, 2005; Togelius et al., 2006; Charles & Black, 2004; Missura & Gärtner, 2009) demonstrates the power of utilising the player models to create the games or in-game situations of high interest and satisfaction for the players.

3 Dynamically Estimating Player Abilities

3.1 Dynamical Systems Which Adapt to Users

Adapting the behavior of a system to a user is not only useful for games. Work of Boger et al. (2005); Hoey et al. (2007) shows how a system can be developed to assist people suffering from dementia in a task such as hand-washing. Some people might forget to rinse their hands after applying soap, and a visual or auditory cue might help to remind them or, in worse cases, the assistance of a carer might be required. If a prompt is not given when needed, the person will not succeed in the task; if a prompt is given when none was required the person will feel less independent. Here we see that adaptation to the specific case is important. To make the task even harder for the system, the cognitive abilities will change over time, so the system should be able to cope with this, ideally by having some kind of prediction of future abilities from the estimated current state.

Closer to the dynamic difficulty adjustment in games is the ongoing work (building on previous work (Patricia Kan, Jesse Hoey, Alex Mihailidis, 2008)), in which the same approach is used for rehabilitation for upper limb recovery for people who have suffered a stroke. Here the person has to perform a simple exercise, where the system can adapt the difficulty. Ideally the exercise should be of such difficulty that it should be manageable, but not too easily. The system keeps an estimate of the fatigue level, prompting the person to take a break when the estimated fatigue level is too high. The exercise system is a haptic controller for a computer game the person is playing. This was introduced to make the exercises more enjoyable, an important aspect of rehabilitation exercises. We will show some results of simulations with this system in Section 5.

The approach used in both these domains is to model the person together with the system as a POMDP, which is capable of dealing with sequential dynamic systems where some states are preferred to others and not everything relevant to the process is observable. In the given examples the actual abilities of the people are not directly observable, but their interactions with the system are.

When using a model represented as a POMDP, at each step a *belief state* (a probability distribution over all states) is kept. A policy for a POMDP tells us for any belief state which action the system should take next. When the action is taken and the observable variables are gathered, the belief state can be updated using the new evidence in a Bayesian approach.

A given POMDP can be solved by a number of approaches, but finding an exact solution is generally a very complex and time-consuming task (Aberdeen, 2003). The POMDPs for the described tasks were represented as *influence diagrams*, a method to reduce large POMDPs to a smaller and easier to handle form (Hoey et al., 1999). Given this representation, an adaptation of the PERSEUS algorithm (Spaan & Vlassis, 2005) can find an approximate solution.

Generally, an influence diagram for a system adapting to the user's abilities would look like Figure 1. In an influence diagram, a node represents a variable aspect of a state. The probabilities of these values are dependent on only those variables from which a directed edge arrives at the variable node. For example, in Figure 1, the value of the STRETCH variable in the next time step depends only on the ACTION of the system and the current ABILITY of the person. Variables where the name ends in an apostrophe (e.g. ABILITY') are variables representing the state in the next time-step.

There is a single action per time step that the system can choose and four variables describing the current state. ABILITY indicates the current estimate of the abilities of the user. The variable TASK is a description of the task the person is performing; representing for example the type of game being played. STRETCH describes the difference in level between the selected action and the actual abilities of the person. If the stretch is high, the task is estimated to be very hard, if it is equal to zero the task is exactly at the current ability, if it is below zero the task is too easy. This variable is crucial as the entire expected behavior depends directly on this variable. The BEHAVIOR is what we are actually able to monitor, in the rehabilitation task the behavior is the time needed to perform the exercise and whether the person compensated using his or her upper body. The conditional probabilities of these variables are monotonic functions of the STRETCH variable, with high stretch giving high probabilities of compensating and lower control.

3.2 Modeling a Game and a Player as a POMDP

The adaptive models represented as POMDPs as described in the previous section are clearly usable for finding policies for adaptive games. Indeed, in such games we want the behavior of the system (the selection of the difficulty level, for example) to depend on the current estimates of the player abilities, her enjoyment and the type of the player.

In this case the hidden variables are the actual abilities of the player (which will change as the player learns to play the game better), the enjoyment (including boredom and frustration levels), and the player type, as well as the stretch (as in the previous section) which indicates how hard the previous level was, given the estimated abilities of the player. Observable behavior is the time the player needs to complete the level, how many lives the player lost during the level, how many sub-goals were accomplished, etc. An important extra observable parameter is whether the player decided to stop playing for now – this might indicate that he did not enjoy the game enough. Both observable and unobservable variables will be discussed more elaborately in the next section.

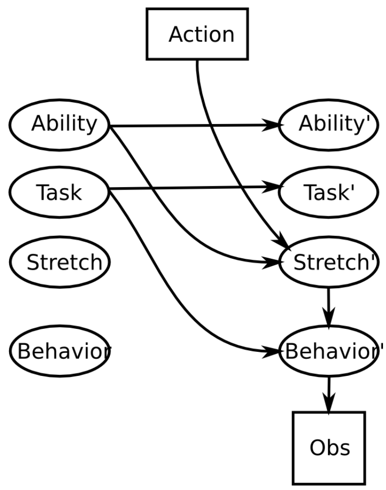

Figure 1. An influence diagram for adaptive systems.

4 Variables of the Player and System in Games

In this section we will discuss possible observable behavior of a player in a game, and the characteristics of a player which have to be estimated from this behavior. These can be modeled by an influence diagram, enabling us to model the combination of game and player as a POMDP.

4.1 Observable Behavior of a Player

The actual performance (whether the player is playing well or not) tells us a lot about her abilities. Depending on the game, this could be measured by the time to achieve the goal, the number of lives lost during the level, or the number of points scored. If the player has good performance on a level intended to be hard, it means she has a high ability for this type of game and will be expected to perform well in similar levels or games of the same type.

4.2 Unobservable Characteristics of a Player

There are many aspects of the physical and mental capabilities of the player in his current state which are relevant to his behavior and performance in the game but are not directly observable. In this section we will give a number of examples of such aspects and describe their possible influence on the behavior of the player, which indicates in what way these things could be estimated and how they should be taken into account for a general strategy.

Abilities: the user's expected success rate for specific kinds of tasks can be very diverse for different people. Some people have slower reflexes or have difficulty getting their timing right, which is critical for many games.

Boredom: if a player is not challenged enough for a period of time, she can be expected to get bored by the game. If a player is bored, general enjoyment is low and the player might decide to stop playing. In this case the game has clearly failed to hold the player's attention. Thus, making levels not too easy is an important part of the general game strategy.

Frustration: in contrast to overly easy levels boring the player, levels which are too hard may be frustrating for the player. If a level is not passable, the player will at some point give up trying and stop playing. Even if she does not, having to try every level for a large number of tries before the player gets through will not give much enjoyment to the player. The conclusion is that the levels should not be too hard given the player's estimated abilities.

Player Type: as not every person is the same, with some people being fast learners and others taking more time, and some people having more experience with certain kinds of games, the prior expected abilities (before the player gets any experience playing this specific game) and the rate at which these change depend on the player type. Not only that, but the rate at which the player gets bored or frustrated depends on this too, as does the threshold at which the player would become too bored or frustrated and decide to stop playing. In the work by Missura and Gärtner (2009) players were clustered according to having similar behavior in a game, where new players could then be classified as belonging to one of these clusters. We can assume that each cluster represents a *type* of player with specific learning rates, prior abilities and enjoyment expectations. In Section 5.2 we describe an experimental simulation where the type of person (indicating his learning rate) is estimated from his behavior.

5 Simulations

As an illustration of the possibilities of the suggested approach we will show some results of simulations using the stroke rehabilitation system. All simulations were run using a simple simulator which will improve his abilities at a constant rate over time, reaching full ability after 100 time steps. The amount of control is deterministically determined by the value of the STRETCH variable, as are the ability to reach the target and the time it takes and whether there was compensation. We realize this is not a plausible simulator for a real person but include these results to illustrate the behavior of the system for varying settings of system parameters.

The influence diagram for the stroke rehabilitation system is shown in Figure 2.

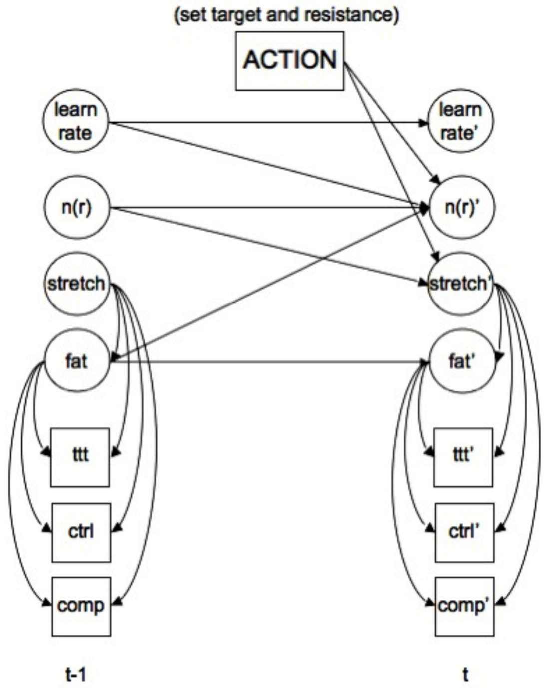

Figure 2. An overview of the influence diagram used for the stroke rehabilitation system.

The LEARN RATE indicates how fast a the increase of abilities is expected to be. The vector N(R) represents which distance the person is able to reach at each resistance setting, thus giving us his or her ability. The STRETCH variable indicates how much the given exercise was beyond the current ability. The fatigue level is represented by the FAT node. For the observations, TTT indicates the time needed to reach the target, CTRL represents whether the exercise was performed in a controlled way and COMP tells us whether the person compensated by using his or her upper body instead of just his arm. Rewards are given for succesful exercises, and a penalty is given whenever the system prompts the user to take a break using the STOP action.

5.1 The Effect of Stop Cost on Exercise Run Length

In a first experiment the cost of the STOP action was varied, measuring the effect on the average length of runs between breaks. The results are shown in Figure 3. This graph clearly shows that increasing the cost of stopping results in longer exercise runs. Important to note is that for very high costs, the system will select exercises slightly below the actual abilities. This prevents the person from becoming too fatigued. This implies that by changing the STOP cost the system can generate short runs of exercises which are a bit harder, or longer runs of easier exercises.

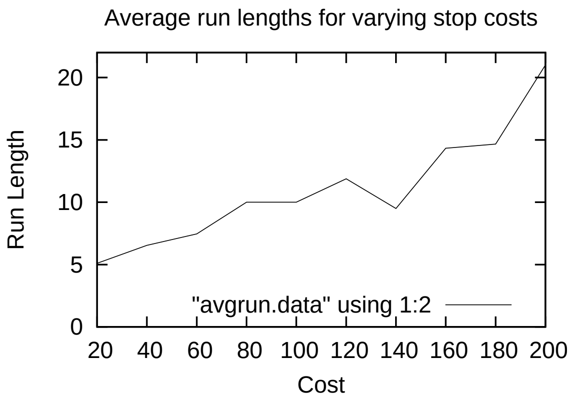

Figure 3. Average run length of exercise sessions for varying costs of the STOP action.

5.2 Estimating the Learning Rate from Observed Behavior

A second experiment using the simulator was performed to measure how the model can be used to estimate the learning rate of a given person. The simulated person was the same as in the previous experiment, with abilities improving every n steps with $10 \leq n \leq 1000$. The model was slightly extended to have three different types of people, with learning rates equal to $0.1, 0.01$ and 0.001.[1] The *a priori* probability of a person being a specific type is equal for each type.

We ran the simulation for 2000 steps for every simulated person, keeping track of the estimated learning rate. The results are shown in Figure 4 (note the logarithmic scale on both x and y-axis). In this graph we plot, for a wide range of actual improvement rates (x-axis) the expected value of the learning rate ($\sum_r rP(r), r \in \{0.1, 0.01, 0.001\}$) after varying numbers of time steps. Here, it is clear that for the fastest learning people (improving every 50 steps or faster) the correct learning rate is found already after a small number of steps. For values of the improvement frequency near 0.001, the estimates converge to the slowest learning type, while values between 0.008 and 0.01 converge to the middle learning rate. For the values between these ranges is not directly clear which learning rate they correspond to. The simulated person does not clearly belong to any of the given types, given the behavior, but is a mix of different types (the belief state has non-zero probabilities for more than one value of the learning rate).

For the lower ability update frequencies, the system takes a long time to converge. We assume this is due to the very low evidence for any increase, leading to the assumption that the user may be getting fatigued. Indeed, for these values of the update frequency the system's estimate of the fatigue level were very high throughout the entire run, which the system takes as a likely explanation for the lack of success in more difficult tasks than the very easiest. The system will then only select very easy exercises, getting little or no examples of slightly more taxing tasks.

The stepwise curve the graph converges to (as can be seen from the graph after 2000 steps) is the behavior as it was expected. The system believes every person to be of one of the three different types, and as the runs progress more evidence is gathered. If a higher resolution of learning rates is required to adequately choose a rehabilitation policy for a user, the model could be easily changed to have more basic learning rates as possibilities.

The fact that the system needs a large number of steps to converge to the correct learning rate for some frequencies does not mean that the model is not useful after a limited number of steps, however. When the system has not converged with high certainty to a single type, the estimate of the person type is a weighted combination of two or three different types, leading to a strategy which is both acceptable to the person and which will try to get more evidence to decide the type. If the system receives a penalty for overstretching, a safe exercise regime would be selected in case of uncertainty, until there is enough evidence that the learning rate is high enough for a heavier exercise scheme.

6 Conclusion

Partially observable Markov decision process present a good approach to find a strategy for games with dynamic difficulty adjustment, where the characteristics and abilities of the player are estimated on-line (during the actual run of the game, not in a separate training phase).

This kind of system has already been implemented in a setting for upper limb rehabilitation for stroke victims. The person plays a computer game where the controller is a haptic device with adaptable resistance and sensors to de-

[1] Unfortunately, due to the complexity of the dynamics of the system, a clear correspondence between the ability improvement frequency and the learning rate is not readily available.

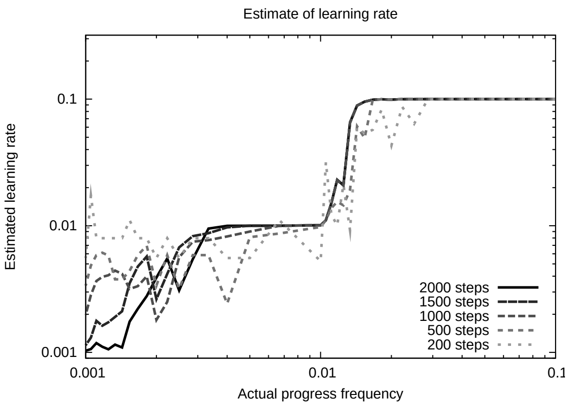

Figure 4. Estimate of the learning rate after given number of steps for simulators with varying ability improvement frequencies.

tect whether the person has good control over his movement or needs to compensate by using upper body strength. The adjustable difficulty in this system lies primarily in the resistance of the controller and the distance the person needs to reach, but if the game itself is too frustrating or boring it will not be enjoyable and the person might be less inclined to do the necessary exercises.

We presented simulated results for this system, showing that changing the values of the parameters directly influences the behavior of the system and showing how aspects of the characteristics of the user (e.g. learning rate) can indeed be measured dynamically.

Acknowledgments

This publication has been supported by FONCICYT contract number 000000000095185. The content of this document reflects only the author's views. FONCICYT is not liable for any use that may be made of the information contained therein. Part of this work was supported by the German Science Foundation (DFG) under the reference number 'GA 1615/1-1'.

References

Aberdeen, D. (2003). A (revised) survey of approximate methods for solving partially observable markov decision processes. *National ICT Australia, Canberra, Australia, Tech. Rep.*

Biederman, I., & Vessel, E. (2006). Perceptual pleasure and the brain. *American Scientist*, *94*.

Boger, J., Poupart, P., Hoey, J., Boutilier, C., Fernie, G., & Mihailidis, A. (2005). A decision-theoretic approach to task assistance for persons with dementia. *International Joint Conference on Artificial Intelligence* (p. 1293).

Charles, D., & Black, M. (2004). Dynamic player modeling: A framework for player-centered digital games. *Proc. of the International Conference on Computer Games: Artificial Intelligence, Design and Education* (pp. 29–35).

Danzi, G., Santana, A. H. P., Furtado, A. W. B., Gouveia, A. R., Leitão, A., & Ramalho, G. L. (2003). Online adaptation of computer games agents: A reinforcement learning approach. *II Workshop de Jogos e Entretenimento Digital*, 105–112.

Herbrich, R., Minka, T., & Graepel, T. (2006). Trueskill$^{\mathrm{tm}}$: A bayesian skill rating system. *NIPS* (pp. 569–576).

Hoey, J., St-Aubin, R., Hu, A., & Boutilier, C. (1999). SPUDD: Stochastic planning using decision diagrams. *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence* (pp. 279–288).

Hoey, J., Von Bertoldi, A., Poupart, P., & Mihailidis, A. (2007). Assisting persons with dementia during handwashing using a partially observable Markov decision process. *Proceedings of the International Conference on Vision Systems*.

Hunicke, R., & Chapman, V. (2004). AI for dynamic difficulty adjustment in games. *Proceedings of the Challenges in Game AI Workshop, Nineteenth National Conference on Artificial Intelligence*.

Missura, O., & Gärtner, T. (2009). Player Modeling for Intelligent Difficulty Adjustment. *Discovery Science* (pp. 197–211).

Patricia Kan, Jesse Hoey, Alex Mihailidis (2008). Automated upper extremity rehabilitation for stroke patients using a partially observable Markov decision process. *Proceedings of the AAAI 2008 Fall Symposium on AI in Eldercare*.

Spaan, M., & Vlassis, N. (2005). Perseus: Randomized point-based value iteration for POMDPs. *Journal of Artificial Intelligence Research*, *24*, 195–220.

Togelius, J., Nardi, R., & Lucas, S. (2006). Making racing fun through player modeling and track evolution. *SAB06 Workshop on Adaptive Approaches for Optimizing Player Satisfaction in Computer and Physical Games* (pp. 61–70).

Togelius, J., Yannakakis, G., Stanley, K., & Browne, C. (2010). Search-based Procedural Content Generation. To appear.

Yannakakis, G., & Maragoudakis, M. (2005). Player Modeling Impact on Player's Entertainment in Computer Games. *Lecture notes in computer science*, *3538*, 74.