

Supplemental Case Acquisition Using Mixed-Initiative Control

Michael W. Floyd and Babak Esfandiari

Department of Systems and Computer Engineering
 Carleton University
 1125 Colonel By Drive
 Ottawa, Ontario, Canada

Abstract

Learning by observation allows a software agent to learn by watching an expert perform a task. This transfers the burden of training from the expert, who would traditionally need to program the agent, to the agent itself. Most existing approaches to learning by observation perform their observation in a purely passive manner. We propose a case-based reasoning agent that is able to observe passively but can also use mixed-initiative control to request assistance from the expert for difficult input problems. Our agent uses mixed-initiative case acquisition in the game of Tetris. We show that the agent is able to obtain cases it would not have been able to with passive observation alone, is able to improve its performance and places less burden on the expert.

1 Introduction

When designing a software agent, the duty of training the agent traditionally falls upon an expert who acts as a teacher. However, transferring the expert's knowledge to a software agent can be a difficult and time-consuming task. This is especially true if the expert has difficulty modelling its knowledge, possibly if they lack computer programming skills, or they are not fully aware of all details related to how they perform a task. In order to overcome this knowledge-transfer burden, case-based reasoning has seen an interest in approaches that *learn by observation* (Ontañón et al. 2007; Romdhane and Lamontagne 2008; Floyd, Esfandiari, and Lam 2008; Rubin and Watson 2010; Gillespie et al. 2010). The agent learns by watching the expert perform a task and, when faced with the same task, aims to behave in a similar manner. This shifts the task of knowledge modelling from the expert to the agent.

Instead of making use of a case base that is completely authored by an expert, case-based reasoning systems that learn by observation automate (to varying degrees) case acquisition. The agent can observe the current state of the environment (*the problem*) along with how the expert reacts (*the solution*) and create a case. Automatic case acquisition is highly desirable because it greatly reduces the cost of generating each case. However, the downside of acquiring cases

automatically is that there is no control over *what cases* will be acquired.

Using passive observation (Floyd, Esfandiari, and Lam 2008), where the agent observes the expert without directly interacting with it, the case generation process is completely dependant on the behaviour of the expert and the state of the environment. If the expert never performs certain actions or encounters certain environment states it will be impossible for the agent to obtain cases related to those actions and states. To overcome this limitation, active case acquisition (Floyd and Esfandiari 2009) has been used to present the expert with problems the agent wishes to have solved. Instead of the expert interacting directly with the environment, the agent simulates the environment so a set of desired environment states can be presented to the expert. While active case acquisition allows the agent more control over what cases can be generated, its primary limitation is that the agent may need to provide the expert with a series of input problems in order to get a solution to one particular problem (Floyd and Esfandiari 2009).

To overcome the limitations of both passive and active case acquisition we propose using mixed-initiative control. In such an approach, the agent is controlled by the case-based reasoning (CBR) system unless it is unable to retrieve a solution to an input problem. In those situations the CBR system defers to the expert and can then observe the expert solving the problem and add the resulting case to the case base. This allows for the majority of the case acquisition to be done passively with small sessions of tutoring occurring afterwards.

This paper aims to demonstrate how mixed-initiative control can be used to guide the case acquisition process in order to add cases from poorly represented regions of the problem space. Section 2 describes our approach to case acquisition using mixed-initiative control and an experimental analysis, in the domain of Tetris, is presented in Section 3. Related work, in the areas of mixed-initiative case-based reasoning and case acquisition, is discussed in Section 4. Finally, concluding remarks and areas of future work are presented in Section 5.

2 Mixed-Initiative Case Acquisition

Mixed-initiative systems allow for the control of a single entity, in our case a software agent or robot, by several con-

trollers concurrently. At any time t , only one of the n controllers has initiative over the agent and may control the actions the agent performs. In our discussion we will limit the number of controllers sharing initiative to two¹: the case-based reasoning system and the expert.

Since the motivation for using a mixed-initiative approach is for case generation, under most circumstances the case-based reasoning system will control the agent. By giving the majority of control to the CBR system there will be two primary benefits. Firstly, the CBR system will handle the majority of the problem solving. This minimizes the amount of work the expert must perform since they will be passive most of the time. Secondly, and more importantly, the CBR system will be attempting to behave in a similar manner to the expert but will likely make errors that the expert would never have made. For example, if the expert played a game in an optimum or near-optimum way, it might never perform actions that put it in a disadvantageous position (and the CBR system would never observe cases related to those disadvantageous positions). These errors in the ability of the CBR system to replicate the expert's behaviour can actually be advantageous because it allows for the exploration of previously inaccessible areas, due to the expert's optimum behaviour, of the problem space.

2.1 Agent Control

Since the agent can only be controlled by either the case-based reasoning system or the expert at a particular moment in time, the mechanisms for transferring control are important. Each controller has two control actions: they can *seize* control of the agent or *cede* control to another controller. Therefore, the current controller may either *cede* or do nothing and the other may either *seize* or do nothing. The following details the situations in which the controllers will seize or cede control:

Expert

- **Seize:** The expert may seize control at any time (although this will likely occur rarely). Seizing would generally be performed if the expert noticed the CBR system to be performing poorly and wanted to provide unsolicited assistance.
- **Cede:** The expert will automatically cede control after controlling the agent for a single turn. This is done to give the majority of the control to the CBR system.

Case-based Reasoning System

- **Seize:** The CBR system will never attempt to seize control. Since the expert will automatically cede control back to the CBR system there is no need to seize control.
- **Cede:** The CBR system will cede control to the expert when it determines it is unable to successfully solve the input problem. The system could also cede if it required more information, however our implementation is currently only failure-driven.

¹ Although the work could easily be extended to include multiple experts.

The case-based reasoning system determines which problems it is unable to solve based on the similarity of a problem to the cases in its case base. The input problem will be compared to each case in the CBR system's case base. If none of the cases in the case base have a similarity to the input problem greater than a threshold, τ , the CBR system will cede control of the agent to the expert (otherwise it will solve the problem itself). A flowchart of the process is shown in Figure 1. By setting an appropriate similarity threshold it is possible to control how often the CBR system will request help from the expert.

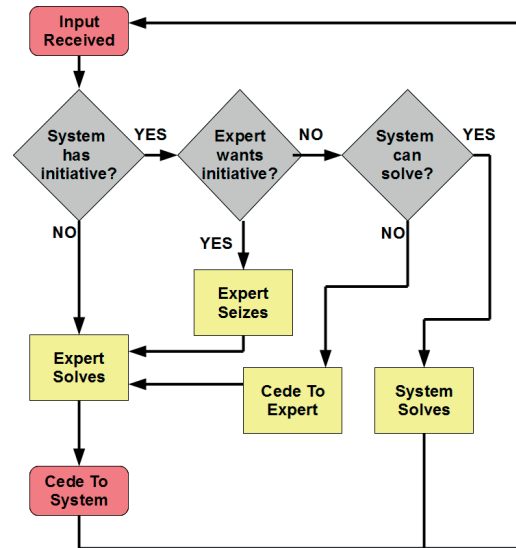


Figure 1: Flowchart of how initiative is seized and ceded by the expert and the system.

2.2 Case Acquisition

When the case-based reasoning system has ceded control (or the expert has seized control) of the agent it will then observe the expert and create a new case from the observation. During observation the CBR system will not have complete knowledge of the expert's reasoning process but must instead rely on the inputs to the expert and outputs from the expert. When controlling a software agent, the inputs to the expert will be the agent's sensory inputs, S and the expert will output the actions, A , it wishes the agent to perform. Each case, C , that is created by observation will contain the input problem (the sensory input S) and solution (the performed actions A):

$$C = \{S, A\} \quad (1)$$

While this approach to case generation is limited in that it does not contain knowledge related to the expert's internal reasoning or goals, it is advantageous because it can be formed completely autonomously without any annotation by the expert. The expert is only required to demonstrate their behaviour and does not need to model or describe it in any way. The burden of case acquisition falls entirely on the

case-based reasoning system and requires no manual authoring from the expert.

We have described how the CBR system cedes control of the agent to the expert during run-time but the initial learning by observation can also be thought of in a mixed-initiative context. During the initial training the system essentially sets the similarity threshold arbitrarily high ($\tau = \infty$) such that it is constantly allowing the expert to solve the input problems. The CBR system has a limited case base so it is focused on letting the expert solve problems so that it can observe and attain more cases. As the case base size grows, and the CBR system becomes better at solving problems on its own, the similarity threshold can decrease (either to a constant value or at a specific rate over time) to allow the CBR system to behave with more autonomy.

3 Experimental Results

Our experiments will look to demonstrate the benefits of mixed-initiative control for case acquisition. We will use the game of Tetris as the domain and show how mixed-initiative case acquisition can generate cases that can not be obtained in a passive manner. Tetris was chosen because it has a large problem space (approximately 2^{216} using our representation) but the player has a significant amount of control over what problems are encountered based on how they play.

3.1 Experimental Domain

In our experiments the case-based reasoning system and the expert will share control of an agent that plays the game of Tetris. During a game of Tetris, a player must slide and rotate descending game pieces onto the rectangular game region (Figure 2). Each game piece is composed of four square blocks, arranged in various configurations, that stack on one another as they are placed in the game region. The goal of the player is to fill entire rows of the game region with blocks, making the row disappear, in order to keep blocks from being stacked to the top of the game region.

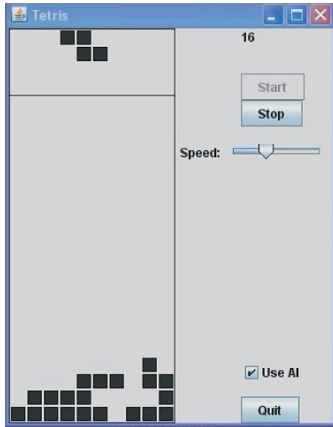


Figure 2: Game of Tetris

The sensory inputs, S_{Tetris} , received by the player during a Tetris game have two components: the game region and the game piece. The game region, which has 20 rows and 10

columns, is represented by a 20×10 matrix, R . Similarly, the game piece is represented by a 4×4 matrix, P . Each of these matrices contain binary values to represent if a cell is empty (a 0 value) or not (a 1 value).

$$S_{Tetris} = \{R, P\} \quad (2)$$

The similarity between two Tetris sensory inputs, A and B , is calculated as follows:

$$sim(A, B) = \frac{1}{2} [sim(A.R, B.R) + sim(A.P, B.P)] \quad (3)$$

And the similarity between two matrices, M_1 and M_2 , is calculated using their normalized Hamming distance:

$$sim(M_1, M_2) = 1 - normHamming(M_1, M_2) \quad (4)$$

The normalized Hamming distance was chosen for its generality in comparing two matrices rather than any benefit for comparing matrices that represent a Tetris board and piece. We looked to create a similarity function that could be used in a variety of domains without hardcoding any domain knowledge about the Tetris game or goals². It should also be noted that we arbitrarily chose to give both the game region and the piece equal weighting in the similarity function. However, this can easily be changed to give more influence to one or the other by adding weights.

The actions of the player, A_{Tetris} , also have two components: sliding and rotating. The player is able to slide the piece horizontally by a certain number of squares (with positive values representing sliding right and negative left), *slide*, and rotate the piece 90 degrees clockwise a certain number of times, *rotate*. An action with 0 slides and 0 rotations would represent a no-op.

$$A_{Tetris} = \{slides, rotate\} \quad (5)$$

There are other methods for representing and comparing Tetris cases, like only examining subregions of the game region, that can lead to better game playing performance (Romdhane and Lamontagne 2008). We have chosen to use our representation because it keeps the sensory information in the same form as it is received by the agent. No information is added, removed or transformed. This avoids introducing any knowledge we have about Tetris strategy in order to make the case-based reasoning system as domain-independent as possible (Floyd and Esfandiari 2010).

3.2 Results

The expert that is observed is a Tetris-playing software agent. The expert plays by placing each piece so as to minimize the resulting height of the stacked pieces and the number of holes in rows. Two case bases, each containing 100,000 cases, were generated by observing the expert. Both case bases use the same set of 90,000 initial cases, called the *seed cases*, that were generated passively. The

²Future work will examine how we can improve performance by learning a similarity function that is better suited for Tetris.

first case base, which we will call the *passive case base*, also contains an additional 10,000 cases that were generated through passive observation. The second case base, called the *mixed-initiative case base*, contains 10,000 additional cases were generated using our mixed-initiative approach (with $\tau = 0.8$). The case-based reasoning system we use only makes use of the case data when reasoning and has no encoding of the rules or goals of Tetris.

Rarity of Cases

The first experiments looked to examine whether mixed-initiative case acquisition is able to generate cases that are not obtainable through a purely passive approach. In the mixed-initiative case base 10,000 cases were generated using our mixed-initiative approach. It is possible that these cases were dissimilar to the 90,000 seed cases but would have occurred had more cases been generated passively.

To test this a much larger case base of 2 million cases was generated passively. Each of the 10,000 passively generated and 10,000 mixed-initiative generated cases were compared to the larger case base to find their most similar case. Of the 10,000 passive cases approximately 99% had a case in the larger case base with a similarity greater than 0.85 and approximately 44% had a similarity over 0.95 (the mean similarity was 0.93). Comparatively only 28% of the mixed-initiative cases had a case in the larger case base with a similarity greater than 0.85 and none had a similarity greater than 0.95 (the mean similarity was 0.82).

What these results demonstrate is that the vast majority of the cases generated using the mixed-initiative approach would not have been obtained even if a much larger case base was generated. This confirms our hypothesis that there are certain input problems that would not be observed in a purely passive manner. The errors made by the CBR system lead to unexplored areas of the problem space that can then be solved by the expert.

Game Performance

Having a case base with a more diverse collection of cases may be desirable but we also look to show it is beneficial for Tetris-playing performance. In order to examine this we measured the average length, in number of game pieces played, of a Tetris game. A longer game length implies the agent was better able to manage the height of the stacked blocks by using strategic piece placement or completing lines.

The CBR system used both case bases, the passive case base and the mixed-initiative case base, to play 350 games of Tetris. During these games the CBR system was not able to receive assistance from the expert and reasoned solely with the case base it was using. Table 1 shows the mean number of pieces played for each case base (and the 99% confidence interval). We can see that the mixed-initiative case base results in a significant increase (using a paired t-test with $\alpha < 0.001$) in the number of pieces played per game.

The increased number of pieces per game, while not a large increase, does show that using mixed-initiative case generation allows the case-based reasoning system to play the game of Tetris better. This is likely because the CBR

system is better able to recover from errors it makes. Since the expert has solved problems that represent the game state after these errors, the CBR system will possess cases that can help it during these situations.

	Passive	Mixed-Initiative
pieces	25.90 (+/- 0.67)	27.80 (+/- 0.81)

Table 1: Mean pieces played using each case base.

Case Generation Cost

One other area of interest is related to the number of problems, during mixed-initiative control, that the case-based reasoning system was able to solve itself. When generating the mixed-initiative case base the CBR system solved approximately 43,000 problems without the assistance of the expert. Since it required the expert's assistance for 10,000 problems (the portion of the case base generated using mixed-initiative control), the CBR system was able to solve 81% of the problems itself.

This is beneficial for two primary reasons. First, less of a burden is placed on the expert. The expert only needed to solve problems that the CBR system was unable to solve. Even if the initial case base was empty, and all cases were generated in a mixed-initiative manner, the expert would still only be required a portion of the time since the system would be able to solve more problems as more cases were added. Secondly, had the CBR system added the *first* 10,000 cases it observed it likely would have added many cases that were highly similar to cases it already had in its case base.

4 Related Work

The most similar work to our own is that of Grollman and Jenkins where they use mixed-initiative control to teach a robot soccer-related behaviours (Grollman and Jenkins 2007a). The robot can be controlled by both an autonomous system or a human controller. Under most circumstances the system will control the robot, unless the human chooses to take control. Their primary difference, compared to our own work, is that the autonomous system has no way of identifying that it needs assistance. Instead, the user must watch the robot to see when it is making mistakes and manually take control. Secondly, they provide background information, in the form of supplemental information about the task currently being demonstrated (Grollman and Jenkins 2007b), rather than demonstrating exclusively with the sensory inputs and performed actions.

Learning interface agents (Maes and Kozierok 1993) are also similar to our work. An agent acts as a user's personal assistant, for tasks like sorting e-mail messages, and learns by observing the user. The agent stores instance cases, containing information about the email and where the user moved it, and then compares new e-mail messages to these instances in order to determine where to recommend they should be moved. This work differs from our own in that it is designed to be assistive to a user rather than replace a user. Many of the features that are used are related to how the user interacts with the e-mail, like reading it or replying to it, so

the agent would be unable to operate without the user's involvement. Also, since the user is in control of actions the agent only passively observes and has no control over which situations are observed. This is in contrast to our approach which allows the agent to influence which input problems the expert solves.

Automatically generating cases is central to much of the work on using CBR to learn by observation. This includes generating cases while observing an agent playing the game of Tetris (Romdhane and Lamontagne 2008), building case-based planning libraries from traces of real-time strategy games (Ontañón et al. 2007) and extracting cases from logs of chess (Flinter and Keane 1995) or poker (Rubin and Watson 2010) games. The difference between these works and our own is that they all generate cases passively and do not attempt to guide the case acquisition process through identification of poorly covered regions of the problem space. Instead of observing an expert, Powell et al. (Powell, Hauff, and Hastings 2005) randomly generate novel cases and then evaluate them using reinforcement learning.

Passive case acquisition has also been used in domains that require extracting information from text. Yang et al. (Yang, Farley, and Orchard 2008) use data from two sources, human authored maintenance reports and computer generated fault messages, to create cases in an aviation maintenance domain. Similarly, Asiimwe et al. (Asiimwe et al. 2007) extract information from reports to create cases related to determining home upgrades for people with disabilities. More active approaches to identifying poorly covered regions of the problem space have used measures of complexity (Massie, Craw, and Wiratunga 2005) and coverage (McSherry 2000). These approaches analyze the case base offline, rather than while the CBR system is solving problems, and therefore requires some prior knowledge about the problem space.

A significant amount of mixed-initiative case-based reasoning work has been done in the area of query formulation (Gupta and Aha 2003). A human user is assisted in formulating a query by a system that determines the most useful attribute constraints to add to the query (Bridge 2002) or gathers query information autonomously (Carrick et al. 1999). McSherry and Aha (McSherry and Aha 2007) describe a recommender system where the user provides critiques of recommendations in order to add constraints to their search. Generally, in situations where the search has become over-constrained the system will determine which constraints to relax. However they propose mixed-initiative constraint relaxation where the user can propose which constraints should be relaxed. These systems tend to operate in a conversational manner with the CBR system posing questions and the user providing responses. This results in the initiative being transferred after each interaction whereas our approach allows initiative to be transferred at any time.

Mixed-initiative case-based planning has been used to integrate two stand-alone planning systems (Veloso, Mulvehill, and Cox 1997). The user can search and manually modify plans using one system, ForMAT, while the other system, Prodigy/Analogy, performs automated plan adaptation and recommends modifications for the user to make.

5 Conclusions and Future Work

In this paper we have described an approach to case acquisition that uses mixed-initiative control to allow for supplemental assistance from an expert. Unlike passive observation, which gives the agent no control over the observed input problems, mixed-initiative observation allows the agent to direct the input problems toward ones that it wishes to have solved by the expert (this can either be intentional or due to errors made by the agent).

We have shown, in the game of Tetris, that there are certain cases that can not be observed in a purely passive manner and that this is especially true if the expert behaves in an optimum or near-optimum way. This means there are regions of the problem space that are not represented in a passively generated case base. Using our mixed-initiative case acquisition method a case base was generated that allowed our CBR agent to play Tetris better than when using a passively obtained case-base.

The primary limitation of our approach is that it requires a cooperative expert. The expert must be willing to take control of the agent when requested. If the expert does not know it is being observed or is unwilling to assist then a mixed-initiative approach would not be possible. However, if the expert is willing to help then our mixed-initiative approach helps reduce the burden on the expert. The case-based reasoning system will only ask the expert to solve problems that it is unable to solve itself so the expert will not need to solve numerous highly similar problems.

Our future work will look to optimize the parameters used during mixed-initiative case acquisition (like the similarity threshold and percentage of cases to generate using mixed-initiative control). Even without optimizing the parameters we were still able to demonstrate the benefit of mixed-initiative case acquisition and obtain performance results that are similar to other Tetris-playing CBR systems that use a global view of the Tetris board (Romdhane and Lamontagne 2008). We have not directly compared our results to those of Romdhane and Lamontagne because our goal was not to optimize Tetris playing but instead to demonstrate an approach to case acquisition. Future work will also look to learn human-like strategies and case representations using only observed cases and without having to provide any explicit domain knowledge.

References

- Asiimwe, S.; Craw, S.; Taylor, B.; and Wiratunga, N. 2007. Case authoring: From textual reports to knowledge-rich cases. In *7th International Conference on Case-Based Reasoning*, 179–193.
- Bridge, D. G. 2002. Towards conversational recommender systems: A dialogue grammar approach. In *Workshop on Mixed-Initiative Case-Based Reasoning at the 6th European Conference on Case Based Reasoning*, 9–22.
- Carrick, C.; Yang, Q.; Abi-Zeid, I.; and Lamontagne, L. 1999. Activating CBR systems through autonomous information gathering. In *3rd International Conference on Case-Based Reasoning*, 74–88.

- Flinter, S., and Keane, M. T. 1995. On the automatic generation of cases libraries by chunking chess games. In *1st International Conference on Case-Based Reasoning*, 421–430.
- Floyd, M. W., and Esfandiari, B. 2009. An active approach to automatic case generation. In *8th International Conference on Case-Based Reasoning*, 150–164.
- Floyd, M. W., and Esfandiari, B. 2010. Toward a domain-independent case-based reasoning approach for imitation: Three case studies in gaming. In *Workshop on Case-Based Reasoning for Computer Games at the 18th International Conference on Case-Based Reasoning*, 55–64.
- Floyd, M. W.; Esfandiari, B.; and Lam, K. 2008. A case-based reasoning approach to imitating RoboCup players. In *21st International Florida Artificial Intelligence Research Society Conference*, 251–256.
- Gillespie, K.; Karneeb, J.; Lee-Urban, S.; and Muñoz-Avila, H. 2010. Imitating inscrutable enemies: Learning from stochastic policy observation, retrieval and reuse. In *18th International Conference on Case-Based Reasoning*, 126–140.
- Grollman, D. H., and Jenkins, O. C. 2007a. Dogged learning for robots. In *IEEE International Conference on Robotics and Automation*, 2483–2488.
- Grollman, D. H., and Jenkins, O. C. 2007b. Learning robot soccer skills from demonstration. In *IEEE International Conference on Development and Learning*, 276–281.
- Gupta, K. M., and Aha, D. W. 2003. Framework for incremental query formulation in mixed-initiative case-based reasoning. In *Workshop on Mixed-Initiative Case-Based Reasoning at the 5th International Conference on Case-Based Reasoning*.
- Maes, P., and Kozierok, R. 1993. Learning interface agents. In *11th National Conference on Artificial Intelligence*, 459–465.
- Massie, S.; Craw, S.; and Wiratunga, N. 2005. Complexity-guided case discovery for case based reasoning. In *20th National Conference on Artificial Intelligence*, 216–221.
- McSherry, D., and Aha, D. W. 2007. Mixed-initiative relaxation of constraints in critiquing dialogues. In *7th International Conference on Case-Based Reasoning*, 107–121.
- McSherry, D. 2000. Automating case selection in the construction of a case library. *Knowledge-Based Systems* 13(2-3):133–140.
- Ontañón, S.; Mishra, K.; Sugandh, N.; and Ram, A. 2007. Case-based planning and execution for real-time strategy games. In *7th International Conference on Case-Based Reasoning*, 164–178.
- Powell, J. H.; Hauff, B. M.; and Hastings, J. D. 2005. Evaluating the effectiveness of exploration and accumulated experience in automatic case elicitation. In *6th International Conference on Case-Based Reasoning*, 397–407.
- Romdhane, H., and Lamontagne, L. 2008. Reinforcement of local pattern cases for playing Tetris. In *21st International Florida Artificial Intelligence Research Society Conference*, 263–268.
- Rubin, J., and Watson, I. 2010. Similarity-based retrieval and solution re-use policies in the game of Texas Hold'em. In *18th International Conference on Case-Based Reasoning*, 465–479.
- Veloso, M. M.; Mulvehill, A. M.; and Cox, M. T. 1997. Rationale-supported mixed-initiative case-based planning. In *9th Conference on Innovative Applications of Artificial Intelligence*, 1072–1077.
- Yang, C.; Farley, B.; and Orchard, R. 2008. Automated case creation and management for diagnostic CBR systems. *Applied Intelligence* 28(1):17–28.