Utilizing Case-Based Reasoning and Automatic Case Elicitation to Develop a Self-Taught Knowledgeable Agent

Jay H. Powell, Brandon M. Hauff, & John D. Hastings
Dept. of Computer Science and Information Systems
University of Nebraska at Kearney, USA
{hueljh@hotmail.com, brandon@genxian.com, hastingsjd@unk.edu}

Abstract

Traditionally case-based reasoning (CBR) systems have relied on information manually provided by domain experts to form their knowledge bases. Additional domain knowledge is often used to improve performance of such systems. A less costly method of knowledge acquisition is *automatic case elicitation*, a learning technique in which a CBR system acquires knowledge automatically during real-time interaction with its environment with no prior domain knowledge (e.g., rules or cases). For problems that are observable, discrete and either deterministic or strategic in nature, automatic case elicitation can lead to the development of a self-taught knowledgeable agent. This paper describes the use of automatic case elicitation in CHEBR, a CHEckers case-Based Reasoner that employs self-taught knowledgeable agents. CHEBR was tested using model-based versus non-model-based matching to evaluate its ability to learn without predefined domain knowledge. The results suggest that additional experience can substitute for the inclusion of precoded model-based knowledge.

Introduction

Knowledgeable agents are routinely used to enhance game play by providing challenging and adaptable opponents (Laird & van Lent 2001). Unfortunately, agents' knowledge of their gaming world is typically domain specific and therefore not adaptable to evolving or changing domains. Ideally, a game agent could be self-taught without access to predefined domain knowledge, by being introduced into a new domain and spending a relatively short amount of time acclimating itself to the new environment and learning the behavior needed to succeed. The agent would only need the ability to act within its environment (e.g., the abilities to move or manipulate objects, and observe the consequences of the actions). This paper argues for the automatic capture of agent knowledge using a technique called automatic case elicitation, and demonstrates and evaluates the performance of this approach in the system CHEBR (CHEckers case-Based Reasoner).

Case-Based Reasoning

Case-based reasoning (CBR) systems (Aamodt & Plaza 1994; Kolodner 1993) rely on a knowledge-base of cases (i.e., past or prototypical scenarios) that are most often captured manually through interview sessions with domain experts. While this manually-guided approach has demonstrated merit (Bergmann *et al.* 2003), it may not be applicable to all domains, such as continuous, partially observable, or time-dependent environments. In such domains:

- Human expertise may not provide an optimal solution,
- Eliciting an expert's knowledge and putting it into case form may be expensive, or
- Time constraints may not allow for human training of an agent.

Each of these situations could potentially benefit from an automated process in which the CBR system can learn with minimal or no human intervention to guide the learning process. For example, human expertise may not provide an adequate foundation for a CBR system because expert knowledge in the area is incomplete, perhaps because the domain is too complex for humans to fully comprehend and analyze. An example is online role playing and strategy games. These massively multiplayer online games use knowledgeable agents to interact with human players in a continuously evolving environment. In such cases, it might be beneficial for a learning system to forgo expert knowledge and allow an agent to learn and grow through its interaction with the game system.

Even when adequate expert knowledge can be acquired, the manual elicitation process can be expensive in terms of time. For complex problems this process could take years, or even a lifetime. However, it is possible for a properly initialized CBR system to rapidly and efficiently acquire this knowledge on its own, relying solely on cheap processor time to automatically train a knowledgeable agent about its environment.

Automatic Case Elicitation

Automatic case elicitation is a learning technique in which a CBR system acquires knowledge automatically during real-time interaction with its environment with no prior domain knowledge (e.g., rules or cases). CBR is particularly well suited to automated real-time learning due to its ability to acquire experiences in the form of cases and rapidly recall and apply these prior experiences to new situations. This is especially true for domains in which knowledge is incomplete or doesn't exist and is suggested as an alternate to brute force search techniques which do not transfer easily to new problem domains.

Our approach views a real-time environment as being comprised of a sequence of situations. For each situation encountered, the system takes a reasonable action by referring to prior experience, evaluates the outcome of the action, and commits the experience to memory along with a rating of the action. In a sense, the rating represents a memory of the usefulness of the action in the given context.

The action taken depends on the relationship of the situation to prior experience. For a situation with a similar past experience and a successful outcome (i.e., one with a good rating), the system applies the action taken in the prior experience to the current situation. For a situation that is novel, sufficiently distant from prior experience, or previously unsolved (i.e., the actions taken in previous experiences with the situation resulted in poor outcomes), the system creates and applies a new action (e.g., by selecting an action at random or by combining the actions of multiple successful experiences) which does not depend on prior knowledge of the model. Action in novel or previously unsolved situations results in the addition of new cases to memory, while the reuse of an existing case in the identical context results in an update to the case's success rating.

Over time, the system begins to encounter situations which are identical or similar to those previously seen. The system learns continuously by evaluating the application of retrieved cases to these new situations. Cases that provide good results suggest further reuse, while cases that provide poor outcomes suggest that alternate, previously unattempted solutions are required in given situations.

Related Work

Prior work has researched the use of CBR in games. For example, MAYOR (Fasciano 1996) is a player of the simulation game SimCity and is based on a predefined understanding of an incomplete world model. A case-based planner complements the world by using a library of plans manually built prior to game play. In automatic case elicitation, cases are gathered in real time and are used as the sole reasoning mechanism.

(Goodman 1993; 1994) uses offline built decision-tree induction projectors to predict the outcome of various actions during game play in Bilestoad. Automatic case elicitation differs in that agents learn in real time and projection is not coded as a separate step but is instead encapsulated within individual case ratings.

Previous work has also investigated the use of automatic case generation from predefined expert knowledge. For example, the planning system SHOP/CCBR (Mukkamalla & Muñoz-Avila 2002) automatically acquires cases from manually entered project plans. A related approach has been seen in chess games (Flinter & Keane 1995; Sinclair 1998) which use CBR for chess play by automatically generating case libraries from sets of preexisting grandmaster games. (Shih 2001) integrates CBR and the idea of sequential dependency to learn bridge play from a set of existing games. Automatic case elicitation does not compile cases from manually entered or existing data, but instead knowledge is acquired automatically through the experiences of the agents who learn completely from scratch.

GINA (DeJong & Shultz 1988) uses real-time experience-based learning to enhance the abilities of an existing Othello game-playing program. Automatic case elicitation differs in that an existing problem solver is not used as a basis for the system, instead CBR is used as the sole reasoning mechanism.

This research extends previous results by developing self-taught knowledgeable agents who learn in real time from scratch by automatically acquiring knowledge in the form of cases without the presence of preexisting model-based or expert knowledge.

CHEBR

We illustrate automatic case elicitation in CHEBR, a CBR system capable of allowing human and computer players to compete in the board game of checkers. Checkers was chosen as an initial test domain due its observable, real-time nature and its relative complexity due to the nondeterministic nature of the board (i.e., a specific board configuration is not mapped directly to a corresponding correct action). In CHEBR, computer players are represented by CBR agents that utilize automatic case elicitation and play against other computer and human players in order to learn and test their expertise.

At the beginning of a CHEBR agent's life, its only understanding of the checkers domain is a vision of the game board and the ability to move game pieces. Since the agent in its infancy has no prior experience with checkers, and thus lacks knowledge of valid game play, it learns valid moves through a process of trial-and-error interaction with the game manager (a module that processes agent requests and indicates whether or not the desired action is valid). This approach is used rather than implementing a move generation technique that would depend on prior knowledge of the domain. The agent repeatedly attempts a new move (i.e., one not previously seen in the case base) until a valid move is found. Once found, the valid move is made and committed to the case base along with a snapshot of a real-time board state and a rating of the move's performance. As the case library grows, the agent increasingly applies previously acquired knowledge (in the form of cases) to new scenarios. The rules and behavior which influence how the pieces are controlled are not manually coded into the system, but rather are acquired as an inherent part of the case library through interaction with its environment.

A case rating system is used to provide positive reinforcement so that the agent can discriminate between appropriate and inappropriate moves. The rating system uses the equation

$$\frac{c + Wins}{2 \times c + Wins + Losses}$$

where *c* is a constant value used to deemphasize moves made early on in an agent's life when the agent is simply learning valid game play. For example, if the value of *c* is ten, then for the first ten games the rating of every case in the agent's library will tend towards 0.5, the rating of an average move. As the agent gains more experience and plays more games, the rating of a move can tend towards 1.0 (an optimal move) or 0.0 (a completely ineffectual move). This rating approach is meant to simulate a general positive remembrance a player might have for a particular move and the general long-term consequences derived from repeated application of said move.

The agent makes informed decisions (i.e. those based on previous experience) by retrieving similar cases its case library as described in the following section. The retrieved cases are sorted in descending order by their rating. The agent attempts to apply the highest rated move by iterating through the set of retrieved cases until a valid move is found. If the agent finds that none of the moves from the retrieved cases are applicable to the current board state, it resorts to attempting a new move.

Matching in CHEBR

Two different matching algorithms were tested in CHEBR. The first algorithm, called CHEBR-Exact, requires retrieved cases to exactly match the current board. We will not detail CHEBR-Exact, but note that multiple cases can be retrieved because a variety of moves can typically be made for any one board position.

The second algorithm, called CHEBR-Region, uses a regional matching scheme (similar to chunking in chess (Flinter & Keane 1995)) and is summarized in Figure 1 by the function **FindRegionalMatchingCases**. CHEBR-Region begins by reading a case (C_i) from the agent's case library and extracting the original location of the piece moved (p_0) and the final position of the piece (p_f). The agent then begins its examinations of the retrieved case and the current board by constructing small grids (i.e., subsets of both the case and checkers board) centered around p_0 and p_f on the case as well as the board using the function `RegionCenteredAt`. Each grid contains the specified center position (p_0 or p_f) and all immediately adjacent locations. The purpose of these grids is to allow the agent to investigate how game pieces close to it will affect the move being made.

When grids are built, two scenarios can occur. The first scenario takes place when grids surrounding the original and final locations of the piece in the case overlap ($r_{c_0} \cap r_{c_1} \neq \phi$), meaning that the move made in the case was a relatively short one (i.e., not a multiple jump), and regions surrounding intermediate jump steps need not be considered when comparing the case to the current board.

The second scenario arises when the grids constructed by the agent do not overlap ($r_{c_0} \cap r_{c_1} = \phi$), meaning that the move made in the case was a long one (i.e., a multiple jump). In this situation, regions with an offset of $\pm\delta$ from the midpoint (p_m) of the multiple jump in both the case and current board are compared. These regions allow a better comparison of game pieces in the area that the agent may move through. By examining the area surrounding the origin and destination of the move, as well as the areas immediately adjacent to the midpoint of the move, a subset of the checkers board that is roughly shaped like a diamond is examined.

Once analogous regions on both the case and current board state have been constructed, they are compared to determine similarity. To do this, the average difference between corresponding regions is calculated, as well as the average size of each of the regions. The average region difference determines the number of pieces in each corresponding region that differ. If no more than α percent of the pieces in the case and the current board state differ ($avgDiff < \alpha \times avgRegionSize$), then the case is considered to sufficiently match the current board state. The algorithm proceeds in this manner and checks every case in the library until all matching cases are found and returned.

function FindRegionalMatchingCases()

C := case library
B := current game board
$M := \phi$
for each case $C_i \epsilon C$ **do**
  p_0 := original position of piece moved in C_i
  r_{c_0} := `RegionCenteredAt`(C_i, p_0)
  r_{b_0} := `RegionCenteredAt`(B, p_0)
  p_f := final position of piece moved in C_i
  r_{c_1} := `RegionCenteredAt`(C_i, p_f)
  r_{b_1} := `RegionCenteredAt`(B, p_f)

  if $(r_{c_0} \cap r_{c_1}) = \phi$ **then**
    $p_m := \frac{|p_0 - p_f|}{2}$
    r_{c_2} := `RegionCenteredAt`($C_i, p_m - \delta$)
    r_{b_2} := `RegionCenteredAt`($B, p_m - \delta$)
    r_{c_3} := `RegionCenteredAt`($C_i, p_m + \delta$)
    r_{b_3} := `RegionCenteredAt`($B, p_m + \delta$)

  $avgDiff := \frac{\sum_{k=0}^{n-1}(r_{c_k} - r_{b_k})}{n}$

  $avgRegionSize := \frac{\sum_{k=0}^{n-1} \mathbf{card}(r_{c_k})}{n}$

  if $avgDiff < (\alpha \times avgRegionSize)$ **then**
    $M := M \cup C_i$
return M ;matching cases

Figure 1: CHEBR-Region matching algorithm

Results

CHEBR was tested using the two matching algorithms described in the previous section to evaluate its ability to learn without the use of predefined domain knowledge. The two configurations, CHEBR-Exact and CHEBR-Region, each with initially empty knowledge bases, were pitted against one another for 2000 games (approximately two and one-half hours worth of game time on a machine running an AMD Athlon XP2000+ processor).

Figure 2 shows the winning percentage of the CHEBR-

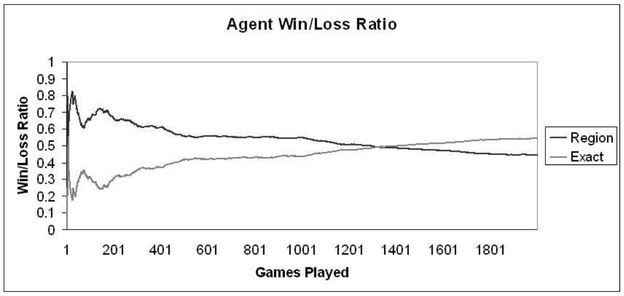

Figure 2: Win/loss ratio for CHEBR-Exact and CHEBR-Region

Exact and CHEBR-Region agents during a 2000 game match. These results were duplicated in repeated tests with a slight variability caused by the use of random move generation. CHEBR-Exact plays without prior domain knowledge, while CHEBR-Region takes advantage of non-learned precoded domain knowledge (i.e., a matching algorithm that assumes that checkers players might gain a strategic advantage by looking at regions within a game board). Notice that CHEBR-Region proved initially superior to CHEBR-Exact but the initial wide difference in skill decreased over time as CHEBR-Exact learned to adjust to the region matching strategies used by CHEBR-Region. The results suggest that experience gained through additional training (in CHEBR-Exact) can substitute for the addition of precoded model-based knowledge (in CHEBR-Region).

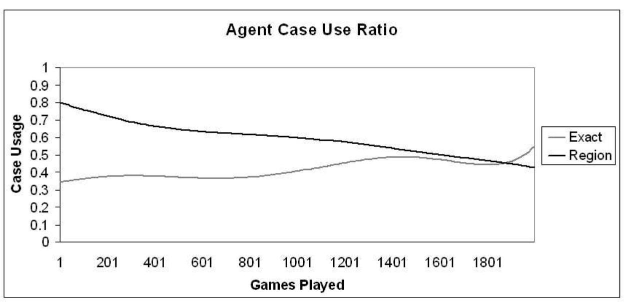

Figure 3: Case-use ratio for CHEBR-Exact and CHEBR-Region

Figure 3 illustrates the percentage of situations in which CHEBR-Exact and CHEBR-Region were able to reuse cases from their case library. The figure, which uses a sixth degree polynomial trend line to smooth the case ratio graph, shows that CHEBR-Region initially depended on prior experience around eighty percent of the time, with the ratio steadily decreasing throughout the testing. This means that over time CHEBR-Region was less able to apply previous experience to new situations. The early high case-use ratio for CHEBR-Region suggests that it is often incorrectly applying prior experiences, perhaps due to an overly general region matching scheme. As expected, CHEBR-Exact's case-use ratio grew at a much slower rate due to its requirement for exact case matches, but this rate grew for the majority of the training until a spike within the last 100 games.

Although not demonstrated in the figures, CHEBR-Exact's current checkers game play is competitive in the early to mid portion of each game played. CHEBR-Exact's quality of moves late in the game, especially with a small case library, deteriorates due to the great number of permutations of board configurations and a low ratio of these permutations represented in the case library. As CHEBR-Exact is trained longer and allowed to expand its case base, checker game play improves. These results are consistent with previous results (Sinclair 1998) that argue that larger case bases are required to fully represent diverse scenarios which are commonly encountered towards the end of game play.

Overfitting is a natural concern when training CBR systems for a long time. The results in Figure 2 do not seem to illustrate a convergence that could be caused by an overtraining effect in which an agent learns to repeatedly apply a small set of effective moves against a particular opponent. Note, however, the spike in the case-use ratio for CHEBR-Exact in Figure 3 within the last 100 games. This spike suggests either overtraining or that CHEBR-Exact is quickly finding a technique for winning against CHEBR-Region.

It appears that CHEBR-Exact requires more games of training to effectively learn checkers play. It is further believed that if CHEBR-Exact were allowed to compete against multiple opponents in the initial stages of its training, CHEBR-Exact might be better able to learn different game play strategies employed by a variety of opponents. This could be accomplished by allowing an inexperienced CHEBR-Exact agent to compete against a set of other computer players, each capable of utilizing their own distinct stratagems.

Conclusion

The development of knowledgeable agents is an interesting subset of artificial intelligence. CBR is one approach to developing such agents. Unfortunately, most CBR systems rely upon previously and manually gathered expert knowledge. An alternate approach, automatic case elicitation, represents a less costly learning technique (for acquiring agent knowledge) in which a CBR system acquires knowledge automatically during real-time interaction with its environment with no prior domain knowledge (e.g., rules or cases). For problems that are observable, discrete and either deterministic or strategic in nature, automatic case elicitation can lead to the development of a self-taught knowledgeable agent. Automatic case elicitation is demonstrated within CHEBR, a CHEckers case-Based Reasoner that employs self-taught knowledgeable agents. CHEBR was tested using model-based versus non-model-based matching to evaluate its ability to learn without predefined domain knowledge. The results suggest that additional experience can substitute for the inclusion of precoded model-based knowledge.

Future Work

In the future, we hope to apply automatic case elicitation to massively multiplayer online games. In the meantime, we hope to learn more about automatic case elicitation within the context of CHEBR. We are currently looking at a variation of CHEBR-Exact that uses a *k*-nearest neighbor approach instead of exact matches. In addition, we plan to improve the move rating system used by CHEBR. Moves made early in training, when an agent is simply learning valid game play, are weighted equally with moves made in later games when actions are more informed. Early moves, which are based on little experience, tend to be low in quality and skew the case rating downward, eventually causing moves made in early games to be used less frequently. The case rating strategy will be revised to emphasize actions taken later in training.

We are developing more advanced model-based versions of CHEBR in order to provide a more rigorous test of CHEBR-Exact. Model-based refinements include improved model-based matching, and a model-based rating system that will select moves based on multiple goals including the current long-term goal of winning a game as well as short-term goals such as minimizing the number of pieces lost as the result of a single move.

Acknowledgements

Thanks to Matt Culver and Nolan Jurgens for their help in the initial development of CHEBR.

References

Aamodt, A., and Plaza, E. 1994. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications* 7(1):39–59.

Bergmann, R.; Althoff, K.-D.; Breen, S.; Göker, M.; Manago, M.; Traphöner, R.; and Wess, S. 2003. *Developing Industrial Case-based Reasoning Applications: The INRECA Methodology*. Number 1612 in Lecture Notes in Artificial Intelligence. Springer Verlag, 2nd edition.

DeJong, K. A., and Shultz, A. C. 1988. Using experience-based learning in game-playing. In *Proceedings of the Fifth International Conference on Machine Learning*, 284–290. San Mateo, California: Morgan Kaufmann.

Fasciano, M. J. 1996. Real-time case-based reasoning in a complex world. Technical Report TR-96-05, Computer Science Department, University of Chicago.

Flinter, S., and Keane, M. T. 1995. On the automatic generation of case libraries by chunking chess games. In *Proceedings of the First International Conference on Case Based Reasoning (ICCBR-95), LNAI 1010*, 421–430. Springer Verlag.

Goodman, M. 1993. Projective visualization: Acting from experience. In *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93)*, 54–59. Menlo Park, Calif.: AAAI Press.

Goodman, M. 1994. Results on controlling action with projective visualization. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, 1245–1250. Menlo Park, Calif.: AAAI Press.

Kolodner, J. 1993. *Case-based reasoning*. San Mateo, Calif.: Morgan Kaufmann.

Laird, J. E., and van Lent, M. 2001. Human-level ai's killer application: Interactive computer games. In *AI Magazine*, volume 2, 15–26. AAAI Press / The MIT Press.

Mukkamalla, S., and Muñoz-Avila, H. 2002. Case acquisition in a project planning environment. In *Proceedings of the Sixth European Conference on Case-based Reasoning (ECCBR-02), LNAI 2416*, 264–277. Springer-Verlag.

Shih, J. 2001. Sequential instance-based learning for planning in the context of an imperfect information game. In *Proceedings of the Fourth International Conference on Case-Based Reasoning (ICCBR-01), LNAI 2080*, 483–501. Springer-Verlag.

Sinclair, D. 1998. Using example-based reasoning for selective move generation in two player adversarial games. In *Proceedings of the Fourth European Workshop on Case-Based Reasoning (EWCBR-98), LNAI 1488*, 126–135. Springer-Verlag.