CASE BASED REASONING & GAME AI

Aprendizagem Simbólica e Sub-Simbólica – 2010

Samuel Mascarenhas
Outline

- Introduction
- Definition of CBR
- Challenges of Implementing CBR
- CBR for Game AI
- Conclusions
Introduction
Introduction

- In most AI approaches:
  - The aim is to perform as well as humans (or even better).

- In the many techniques developed, the parallel with humans becomes unimportant.

- One of the foundations of Case-Based Reasoning is that it gives a strong importance to this parallel
  - CBR attempts to mimic how humans often perform reasoning and learning, from a psychological perspective.
Introduction

- CBR has been successfully applied as an alternative to expert rule-based systems in several domains:
  - Examples:
    - JUDGE (1986) - Criminal Sentencing
    - CHEF (1986) - Create Recipes
    - CASEY (1989) - Heart Failure Diagnosis
    - JULIA (1992) - Meal Planning
    - CADET (1992) - Mechanical Parts Design

- It has also been used to improve decision-making in games.
Definition of CBR
What is CBR?

A methodology that combines

Problem-Solving

Learning

New problems are solved by remembering and adapting the solutions found for similar cases that happened in the past.

When a problem is successfully solved, the experience is retained as a new case, so it can be used in the future.
Origin of CBR 1/2

- CBR appeared in the 80’s following Roger Schank’s ideas on human reasoning and memory.
  - “humans are not rational planners”
  - “human experts are not systems of rules”
  - “human memory is story-based”
Origin of CBR 2/2

- CYRUS [Kolodner-83] was the first implemented CBR system.
  - Based on Schank’s dynamic memory theory.
  - A Q&A system about travels and meetings of an US former Secretary of State.

>Was Vance in Europe last year?
  YES, MOST RECENTLY IN MADRID.
>Was he welcomed there?
  YES, BY FOREIGN MINISTER MARCELINO OREJA AGUIRRE.
>Who did he talk to?
  A GROUP OF SPANISH OFFICIALS.
>Did he also go to Holland?
  YES, AFTER THE TRIP TO SPAIN.
What is a case?

- A case is a representation of an experienced situation often containing its relevant features, the occurring context, and the applied solution.

What is a problem?

- A problem is simply a new case without a solution.
Is it the same as Anological Reasoning?

- Sometimes used as synonyms

- However,
  - Analogical research usually focus on establishing analogies across domains.

- Case-based reasoning can be considered a form of *intra-domain analogy*
The CBR Process
Challenges of Implementing CBR
Representation

- How cases should be defined?
  - Feature vectors,
  - Structured frames,
  - Text-based
Indexing

How should the case memory be structured?

- Cases should be indexed in a way that facilitates retrieval
  - Trivial approach – use all features of the case.
  - Dynamic Memory [Schank-82] - generalize cases which share similar features.
  - Use a tree structure (K-D trees)
    - Splits the search space according to the similarity measure
Retrieve Step

- The goal is to find the most similar cases

- Similarity?
  - K-Nearest neighbors approach:
    - Uses a weighted sum of features.
    - How to determine the weights of the features?
  - Template retrieval approach:
    - Return all cases that fit within certain parameters
Reuse Step

- Adapt the retrieved solution to the new problem.
  - 2 types of methods:
    - Transformational
      - Modifies the previous solution using transformation operators that are domain specific.
    - Derivational:
      - Reuses the algorithms, methods or rules that generated the original solution to produce a new solution to the current problem
Revise Step

- Evaluates the adapted solution.
  - Feedback from “teacher”
  - Feedback from the environment

- If the solution is unsuccessful
  - Repair the solution using domain-specific knowledge.
Retain Step

- The new experience is retained as new case
  - Select the information to be stored in the case.
  - Create a new index in the memory structure for the new case.
## Example – Printer Diagnosis

### Case 1

<table>
<thead>
<tr>
<th>Features</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem: Printer doesn’t work</td>
<td>Replace Toner</td>
</tr>
<tr>
<td>Model: HP 460</td>
<td></td>
</tr>
<tr>
<td>Year: 2000</td>
<td></td>
</tr>
<tr>
<td>Type: Laser</td>
<td></td>
</tr>
<tr>
<td>Power Light: Green</td>
<td></td>
</tr>
<tr>
<td>Paper Light: Green</td>
<td></td>
</tr>
</tbody>
</table>

### Case 2

<table>
<thead>
<tr>
<th>Features</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem: Printer doesn’t work</td>
<td>Add paper to tray</td>
</tr>
<tr>
<td>Model: Olivetti 200</td>
<td></td>
</tr>
<tr>
<td>Year: 1998</td>
<td></td>
</tr>
<tr>
<td>Type: Inkjet</td>
<td></td>
</tr>
<tr>
<td>Power Light: Green</td>
<td></td>
</tr>
<tr>
<td>Paper Light: Red</td>
<td></td>
</tr>
</tbody>
</table>
Example – New Problem

<table>
<thead>
<tr>
<th>Problem</th>
<th>Features</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem: Printer doesn’t work</td>
<td>Model: Epson 1830</td>
<td>?</td>
</tr>
<tr>
<td>Year: ?</td>
<td>Type: Inkjet</td>
<td></td>
</tr>
<tr>
<td>Power Light: Green</td>
<td>Paper Light: Green</td>
<td></td>
</tr>
</tbody>
</table>

Example – Match with Case 1

<table>
<thead>
<tr>
<th>Problem</th>
<th>Case 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Features</strong></td>
<td><strong>Features</strong></td>
</tr>
</tbody>
</table>
| Problem: Printer doesn’t work  
Model: Epson 1830  
Year: ?  
Type: Inkjet  
Power Light: Green  
Paper Light:  Green | Problem: Printer doesn’t work  
Model: HP 460  
Year: 2000  
Type: Laser  
Power Light: Green  
Paper Light:  Green |
| **Solution** | **Solution** |
| ? | Replace Toner |

*Red* means higher weight than *blue*
## Example – Match with Case 2

<table>
<thead>
<tr>
<th>Problem</th>
<th>Case 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Features</strong></td>
<td><strong>Features</strong></td>
</tr>
<tr>
<td>Problem: Printer doesn’t work</td>
<td>Problem: Printer doesn’t work</td>
</tr>
<tr>
<td>Model: Epson 1830</td>
<td>Model: Olivetti 200</td>
</tr>
<tr>
<td>Year: ?</td>
<td>Year: 1998</td>
</tr>
<tr>
<td>Type: Inkjet</td>
<td>Type: Inkjet</td>
</tr>
<tr>
<td>Power Light: Green</td>
<td>Power Light: Green</td>
</tr>
<tr>
<td>Paper Light: Green</td>
<td>Paper Light: Red</td>
</tr>
<tr>
<td><strong>Solution</strong></td>
<td><strong>Solution</strong></td>
</tr>
<tr>
<td>?</td>
<td>Add paper to tray</td>
</tr>
</tbody>
</table>

Case 1 is **more** similar than Case 2.
Example – Apapt Case 1 Solution

Problem

Features

Problem: Printer doesn’t work
Model: Epson 1830
Year: ?
Type: Inkjet
Power Light: Green
Paper Light: Green

Solution

Replace Ink Cartridges

Case 1

Features

Problem: Printer doesn’t work
Model: HP 460
Year: 2000
Type: Laser
Power Light: Green
Paper Light: Green

Solution

Replace Toner

Apply a Transformation using domain knowledge
# Example – Retain

<table>
<thead>
<tr>
<th>Problem</th>
<th>Case 3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Features</strong></td>
<td><strong>Features</strong></td>
</tr>
<tr>
<td>Problem: Printer doesn’t work</td>
<td>Problem: Printer doesn’t work</td>
</tr>
<tr>
<td>Model: Epson 1830</td>
<td>Model: Epson 1830</td>
</tr>
<tr>
<td>Year: ?</td>
<td>Year: ?</td>
</tr>
<tr>
<td>Type: Inkjet</td>
<td>Type: Inkjet</td>
</tr>
<tr>
<td>Power Light: Green</td>
<td>Power Light: Green</td>
</tr>
<tr>
<td>Paper Light: Green</td>
<td>Paper Light: Green</td>
</tr>
<tr>
<td><strong>Solution</strong></td>
<td><strong>Solution</strong></td>
</tr>
<tr>
<td>?</td>
<td>Replace <strong>Ink Cartridges</strong></td>
</tr>
</tbody>
</table>
CBR vs. Rule-Based Knowledge Systems

- Advantages:
  - Improves with time.
  - Does not require a comprehensive theory of the domain.
  - Easier maintenance and can be implemented faster [Simoudis et al., 93]

- Disadvantages:
  - Cases may be insufficient to cover the domain well
  - Needs representation, similarity, adaptation, and revise semantics
CBR in Game AI
Why Game AI?

- Games require players to learn and apply strategies that are adaptive.

- Is very challenging because:
  - Huge State Space
  - Non-Deterministic
  - Real-time (video games)
  - Multi-agents (video games)
Can CBR be used to improve decision-making in games?

- CASPER (Poker)
- CHEBR (Checkers)
- CBRetaliate (Team Based First Person Shooters)
- CaT (Wargus – Real-Time Strategy Game)
CASPER (Watson et al., 2008)

- CASPER (CASe-based Poker playER)
  - texas hold-em variation

- Uses a case-base of past poker experiences
  - Features include hand strength, number of opponents, amount of money,..
  - Retrieval is based on the k-nearest neighbour algorithm

- The case base was constructed by observing hands being played by other poker bots

- No knowledge engineering was performed.
CASPER (Watson et al., 2008)

It plays evenly against strong, adaptive poker-bots.

Plays profitably against non-adaptive poker-bots and against real opponents for play money.
CHEBR (Powell et al., 2004).

- CHEBR (CHEckers case-Based Reasoner)

- Uses Automatic Case Elicitation
  - No prior domain knowledge.
  - Valid moves are learned by a process of trial-and-error.

- Case rating formula:

\[
\frac{c + Wins}{2 \times c + Wins + Losses}
\]
CHEBR (Powell et al., 2004).

After 1300 games, the system taught itself to play better than an agent with domain knowledge.
CBRetaliate (Auslander et al., 2008)

- Game: Unreal Tournament Dominion Mode
  - Goal is to control certain points in the map
CBRetaliate (Auslander et al. 2008)

- RL algorithm is capable of quickly calculating a winning policy against a fixed strategy.

- But it takes time to adapt to switches in opponent strategy.

- Solution: store and retrieve Q-Tables for different strategies using CBR.
CBRetaliate (Auslander et al., 2008)

Game 1

Game 4
CaT (Aha et al. 2005)

- CaT (Case-based Tactician)
  - Implemented in Wargus
CaT (Aha et al. 2005)

- Uses CBR to select tactics (subplans) against randomly selected opponents.
- Was tested against 8 opponent scripts.
- In the first games it performs poorly.
  - Wins an average of 22.9%
- After 100 games it outperforms all other scripts, winning over 80% of games.
CBR & RL

- RL has also been applied to game AI
- RL doesn’t require any heuristics for similarity or adaptation that are domain dependent. However it also requires heuristics in state representation.
- CBR doesn’t try to achieve optimality.
- They can be combined together.
Other Works in CBR for Game AI

<table>
<thead>
<tr>
<th>Name</th>
<th>Genre</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chess</td>
<td>Board Game</td>
<td>(Kerner, 1995)</td>
</tr>
<tr>
<td>Othello</td>
<td>Checkers</td>
<td>(De Jong &amp; Schultz, 1994)</td>
</tr>
<tr>
<td>RoboCup Soccer</td>
<td>Team Sports</td>
<td>(Karol et al., 2003)</td>
</tr>
<tr>
<td>FreeCiv</td>
<td>Turn-based Strategy</td>
<td>(Ulam et al., 2004)</td>
</tr>
<tr>
<td>SimCity</td>
<td>Management</td>
<td>(Fasciano, 1996)</td>
</tr>
</tbody>
</table>
Conclusions

- CBR tries to mimic human reasoning and learning.
- The fundamental principle is to use previous experiences to solve new problems.
- It is a paradigm, not an algorithm.
  - There are many different implementations based on the challenges of the particular domain.
- Has been applied in several different domains replacing rule-based systems with success.
- Has also been used as a technique for improving decision-making in complicated game AI scenarios.
The End!

Questions?