Coaching a Learning Soccer Agent in the RoboCup Simulator

Gregory Kuhlmann and Peter Stone
Outline

• In what environment will the agent act?
  – The RoboCup Soccer Simulator

• How will the advice be communicated?
  – Giving Advice in RoboCup
  – The Coach Competition
  – The UT Austin Villa Coach

• What task will the agent perform?
  – Keepaway

• How will the agent learn?
  – Reinforcement Learning for Keepaway

• How will the agent incorporate advice?
  – Advice for Keepaway
RoboCup Simulator

- **Distributed**: each player a separate client
- Server models dynamics and kinematics
- Clients receive *sensations*, send *actions*

- Parametric actions: *dash, turn, kick, say*
- *Abstract, noisy* sensors, hidden state
  - *Hear* sounds from limited distance
  - *See* relative distance, angle to objects ahead
- > $10^{923}$ states
- **Limited resources**: stamina
- Play occurs in *real time* ($\approx$ human parameters)
Motivation for Coaching

• **MAMSIG**
  – **Aim:** encourage research in opponent modeling
  – **Challenge:** create a simulated coach
    * autonomous agent that gives advice
    * improves performance of a team against a fixed opponent

• **Power of a coach:**
  – More a priori knowledge
  – Better view of world
  – More computational resources

• **Prerequisites:**
  – coachable players *(programmed by others)*
  – standardized coaching language
RoboCup Coach Competition

- **Sub-league** of RoboCup Simulator League
- **Coaching scenario:**
  - Access to log files ("game films") of fixed opponent
  - *Noise-free, omniscient* view of field
  - Limited communication (once every 300 cycles, 50 cycle delay)
    - can’t *micromanage*
  - Advice sent in standardized coach language
  - Players to follow advice *most of the time*
  - Performance measured by *goal difference*
RoboCup Coach Competition (contd.)

- **3 International Competitions** *(plus regional events)*
  - Previous years - best result worse than no advice
    * teams already coherent and competent
    * probably stuck in local maximum
  - 2003 - coaching helped
    * team of players from **several institutions** (UT, CMU, USTC)
    * little or no default strategy.
  - New for 2004 - rule changes
    * standardized **communication** language
    * new **scoring metric**
    * limited time to review logfiles
**CLang**

- **Standardized Coach Language**
  - independent of coachable player’s behavior representation

- If-then rules:
  \[ \{condition\} \rightarrow \{action\} \]

- Example:
  If our player 7 has the ball, then he should pass to player 8 or player 9

  (definerule pass789 direc
   ((bowner our {7})
    (do our {7} (pass {8 9}))))
Example: UT Austin Villa Coachable Player

- Candidate actions are assigned values using a heuristic
  - Based on probability and value of success
- Before advice:

- Action with highest value is chosen
Example: UT Austin Villa Coachable Player (contd.)

- Advice bumps values up (or down)
- When rule pass789 becomes active:
  - generally takes best advised action
  - possible to override advice
The UT Austin Villa Coach

- Opponent-specific advice
  - Learned **defensive positioning** advice
    * predict opponent passes
    * advise player to block pass
  - Learned **offensive action selection**
    * mimic successful team’s passing and shooting
  - Learned **formations**
    * mimic successful team’s positioning
    * average position + ball attraction
- Handcoded rules
  - encode general soccer strategy
The UT Austin Villa Coach (contd.)

- **Game analysis**
  - Given x and y coordinates
  - Detect high-level events: play-by-play
- **Offline learning**
  - Learn from logfiles
  - Online learning possible but difficult
  - All advice sent at start of game
Predicting Agent Behavior

- **Inputs**: features of current world state
  - Player locations, distances to ball and goal, current score, etc.
- **Classification**: $\text{PassFrom}_k$
  - **Example**: PassFrom7 stored in opponent 10’s training set
Model: Decision Trees

- Compile training instances
- Train decision tree for each modeled player
  - J48 algorithm (weka)
Generating Advice

- **Generate advice** for each leaf node in tree
  - Action to **counter** predicted opponent action
  - Example:
    * If opponent 10’s y-coordinate is greater than 17.7, then position our player 4 between opponent 10 and opponent 7

```
(definerule def4rule1 direc
  (((ppos opp {10} (rec (pt -52.5 34) (pt 52.5 17.7))))
   (do our {4} (pos (((pt opp 10) * (pt .7 .7)) +
                         (pt opp 7) * (pt .3 .3)))))))
```
Incorporating Advice

• Thanks to the advice, defender 4 is ready to intercept a pass from opponent 7 to 10.
## Competition Results

<table>
<thead>
<tr>
<th>Team</th>
<th>1st Round</th>
<th>2nd Round</th>
<th>3rd Round</th>
</tr>
</thead>
<tbody>
<tr>
<td>UT Austin Villa</td>
<td>0:19</td>
<td>7th</td>
<td>0:2</td>
</tr>
<tr>
<td>FC Portugal</td>
<td>1:21</td>
<td>8th</td>
<td>0:8</td>
</tr>
<tr>
<td>Iranians</td>
<td>0:14</td>
<td>4th</td>
<td>0:5</td>
</tr>
<tr>
<td>Helli-Amistres</td>
<td>1:12</td>
<td>2nd</td>
<td>0:3</td>
</tr>
</tbody>
</table>

- **1st place** in 2003 RoboCup Coach Competition
- **Only one** other team used learning
- Statistical tie with second place
Experimental Results

<table>
<thead>
<tr>
<th>Opponent</th>
<th>w/ HC</th>
<th>None</th>
<th>Formation</th>
<th>Offensive</th>
<th>Defensive</th>
<th>Full</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoldHearts</td>
<td>N</td>
<td>-8.8</td>
<td>-3.3</td>
<td>-2.9</td>
<td>-2.9</td>
<td>-2.7</td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td>-6.8</td>
<td>-0.5</td>
<td>-1.4</td>
<td>-5.7</td>
<td>-6.5</td>
</tr>
<tr>
<td>Sirim</td>
<td>N</td>
<td>-4.1</td>
<td>2.6</td>
<td>1.2</td>
<td>0.9</td>
<td>1.7</td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td>-5.4</td>
<td>-1.6</td>
<td>-0.3</td>
<td>0.8</td>
<td>-0.4</td>
</tr>
<tr>
<td>EKA-PWr</td>
<td>N</td>
<td>-0.6</td>
<td>2.8</td>
<td>2.9</td>
<td>3.4</td>
<td>2.7</td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td>1.0</td>
<td>3.62</td>
<td>2</td>
<td>2.12</td>
<td>2.43</td>
</tr>
</tbody>
</table>

- Formation learning helps
- Handcoded sometimes hurts
- Offensive and defensive advice mixed
- Why?
Outline

- In what environment will the agent act?
  - The RoboCup Soccer Simulator
- How will the advice be communicated?
  - Giving Advice in RoboCup
  - The Coach Competition
  - The UT Austin Villa Coach
- What task will the agent perform?
  - Keepaway
- How will the agent learn?
  - Reinforcement Learning for Keepaway
- How will the agent incorporate advice?
  - Advice for Keepaway
Giving Advice to a Reinforcement Learner
Case Study: Keepaway

Department of Computer Sciences
The University of Texas at Austin
3 vs. 2 Keepaway

- Play in a **small area** (20m × 20m)
- **Keepers** try to keep the ball
- **Takers** try to get the ball
- **Episode:**
  - Players and ball reset randomly
  - Ball starts near a keeper
  - Ends when taker gets the ball or ball goes out of bounds
- Performance measure: average episode duration
Keeper Policy Space

- Basic skills from CMUnited-99 team
- **Example Policies**
  - Random
  - Hold
  - Hand-coded
Keeper’s State Variables

- 11 distances among players, ball, and center
- 2 angles to takers along passing lanes
Function Approximation: Tile Coding

- Form of sparse, coarse coding based on CMACs [Albus, 1981]
- Tiled state variables individually (13)

![Diagram showing the relationship between full soccer state, few state variables, sparse, coarse, tile coding, and huge binary feature vector.](image)
• Sarsa(λ) outperforms benchmarks

• Learns in 15 hours of simulator time
Advice in a Natural Language

• Possible Advice:
  – Do handcoded solution
  – Hold ball longer
  – etc.

• Convert advice to CLang
  – Example user input:
    If no opponents are within 10m then hold
  – Corresponding CLang:
    (definerule holdLonger1 direc
      ((not (ppos opp {0} (arc (pt our {1}) 0 10 0 360)))
       (do our {1} (hold))))
CLang to Behavior Representation

- Full soccer state
- Few state variables (continuous)
- Sparse, coarse, tile coding

Huge binary feature vector (about 400 1’s and 40,000 0’s)

- Bump up weights corresponding to advice
- Or graft on an additional network (e.g. KBANN)
Quicker Learning

[Graph showing episode duration over hours of training time for different methods: hand-coded, random, and always hold.]
Conclusion

- Advice-giving is well-established in RoboCup Soccer
  - coaching infrastructure in place
  - existing advice language

- 3 vs. 2 Keepaway is a good demo domain
  - simple enough that we know RL works
  - complex enough that advice will probably help
  - possible to scale up to 4 vs. 3, 5 vs. 4, etc.
  - infrastructure in place

- Left to do:
  - translate NL to CLang
  - represent and incorporate advice in learner