CS378
Autonomous Multiagent Systems
Spring 2005

Prof: Peter Stone
TA: Nate Kohl

Department of Computer Sciences
The University of Texas at Austin

Week 14a: Tuesday, April 25th
Good Afternoon, Colleagues

Are there any questions?
Logistics

- Executable teams due next Tuesday
Logistics

- Executable teams due next Tuesday
- Final reports due on Thursday
Logistics

- Executable teams due next Tuesday
- Final reports due on Thursday
- Final tournament: Tuesday, May 16th, 1pm, ACES 2.402
The Tournament

- Nate and I will start running the tournament in the very near future using your executables
The Tournament

• Nate and I will start running the tournament in the very near future using your executables

• We’ll guard the results with our lives
  – Don’t even think of trying to find out…
The Tournament

- Nate and I will start running the tournament in the very near future using your executables

- We’ll guard the results with our lives
  - Don’t even think of trying to find out . . .

- We’ll show you the results on 5/13
The Tournament

• Nate and I will start running the tournament in the very near future using your executables

• We’ll guard the results with our lives
  – Don’t even think of trying to find out…

• We’ll show you the results on 5/13

• Come prepared to (informally) talk about your team for a few minutes
The Tournament

- Nate and I will start running the tournament in the very near future using your executables
- We’ll guard the results with our lives
  - Don’t even think of trying to find out…
- We’ll show you the results on 5/13
- Come prepared to (informally) talk about your team for a few minutes
- We may (or may not) have guests
Suggested Structure

- One group of 5 and one group of 4 with playoffs
  - Semifinals: 2 vs. 1, 1 vs. 2 → finals, 3rd place
  - 3 vs. 3
  - 4 vs. 4
  - 5 vs. loser of 4 vs. 4
Suggested Structure

- One group of 5 and one group of 4 with playoffs
  - Semifinals: 2 vs. 1, 1 vs. 2 → finals, 3rd place
  - 3 vs. 3
  - 4 vs. 4
  - 5 vs. loser of 4 vs. 4

- 2 seeded teams?
Suggested Structure

- One group of 5 and one group of 4 with playoffs
  - Semifinals: 2 vs. 1, 1 vs. 2 → finals, 3rd place
  - 3 vs. 3
  - 4 vs. 4
  - 5 vs. loser of 4 vs. 4

- 2 seeded teams?

- Random draw for the groups
Machine Learning

**Hypothesis space:** set of possible functions

**Training examples:** the data

**Learning method:** training examples $\mapsto$ hypothesis
Machine Learning

Hypothesis space: set of possible functions

Training examples: the data

Learning method: training examples $\rightarrow$ hypothesis

Agent Learning

Policy: how to act (generate training examples)
Machine Learning

Hypothesis space: set of possible functions

Training examples: the data

Learning method: training examples $\mapsto$ hypothesis

Agent Learning

Policy: how to act (generate training examples)

neural network training, decision tree training, clustering, genetic algorithms, genetic programming, reinforcement learning...
3 vs. 2 Keepaway  (joint with Rich Sutton)

- 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

- Performance measure: **average possession duration**

- Use **CMUnited-99 skills:**
  - HoldBall, PassBall(ₖ), GoToBall, GetOpen
Available Skills (from CMUnited-99)

**HoldBall()**: Remain stationary while keeping possession of the ball.

**PassBall(\(k\))**: Kick the ball directly to keeper \(k\).

**GoToBall()**: Intercept a moving ball or move directly towards a stationary ball.

**GetOpen()**: Move to a position that is free from opponents and open for a pass from the ball’s current position (using SPAR (Veloso et al., 1999))

**BlockPass(\(k\))**: Get in between the ball and keeper \(k\).
The Keepers’ Policy Space

Teammate with ball or can get there faster
GetOpen

Ball not kickable

Ball kickable

GoToBall

\{\text{HoldBall, PassBall}(k)\}
(k is another keeper)
The Keepers’ Policy Space

Teammate with ball or can get there faster

GetOpen

Ball not kickable

Ball kickable

GoToBall

{HoldBall, PassBall(k)}
(k is another keeper)

Example Policies

Random: HoldBall or PassBall(k) randomly

Hold: Always HoldBall

Hand-coded:

If no taker within 10m: HoldBall

Else If there’s a good pass: PassBall(k)

Else HoldBall
Mapping Keepaway to RL

Discrete-time, episodic, distributed RL

- Simulator operates in discrete time steps, $t = 0, 1, 2, \ldots$, each representing 100 msec

- Episode: $s_0, a_0, r_1, s_1, \ldots, s_t, a_t, r_{t+1}, s_{t+1}, \ldots, r_T, s_T$

- $a_t \in \{\text{HoldBall, PassBall}(k), \text{GoToBall, GetOpen}\}$

- $r_t = 1$

- $V^\pi(s) = E\{T \mid s_0 = s\}$

- Goal: Find $\pi^*$ that maximizes $V$ for all $s$
Representation

- Full soccer state
- Few continuous state variables (13)
- Sparse, coarse, tile coding
- Huge binary feature vector (about 400 1’s and 40,000 0’s)

Linear map → Action values

Peter Stone
$s$: 13 Continuous 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)
Policy Learning

- Learn $Q^\pi(s, a)$: Expected possession time
Policy Learning

- Learn $Q^\pi(s, a)$: Expected possession time

- Linear Sarsa($\lambda$) — each agent learns independently
  - On-policy method: advantages over e.g. Q-learning
  - Not known to converge, but works (e.g. (Sutton, 1996))
Main Result

1 hour = 720 5-second episodes
### Varied Field Size

<table>
<thead>
<tr>
<th>Keepers</th>
<th>Testing Field Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>15x15</td>
</tr>
<tr>
<td>Trained on field of size</td>
<td></td>
</tr>
<tr>
<td>15x15</td>
<td><strong>11.0</strong></td>
</tr>
<tr>
<td>20x20</td>
<td>10.7</td>
</tr>
<tr>
<td>25x25</td>
<td>6.3</td>
</tr>
<tr>
<td>Benchmarks</td>
<td></td>
</tr>
<tr>
<td>Hand</td>
<td>4.3</td>
</tr>
<tr>
<td>Hold</td>
<td>3.9</td>
</tr>
<tr>
<td>Random</td>
<td>4.2</td>
</tr>
</tbody>
</table>

- Single runs
- Learning specific to fields
  - mechanism generalizes better than policies


- Preliminary: taker learning successful as well

\[\text{Hours of Training Time (bins of 1000 episodes)}\]

\[\text{Episode Duration (seconds)}\]

- handcoded
- random
- always hold
Class Discussion

Marco Huerta on RL tasks