CS344M
Autonomous Multiagent Systems

Patrick MacAlpine

Department of Computer Science
The University of Texas at Austin
Good Afternoon, Colleagues

Are there any questions?
Good Afternoon, Colleagues

Are there any questions?

- What agent could we use in a spectrum auction?
- What is open loop vs closed loop?
Logistics

- FAI talk on Friday at 11 GDC 6.302
  - Itsuki Noda: Multiagent Simulation for Designing Social Services
Logistics

- FAI talk on Friday at 11 GDC 6.302
  - Itsuki Noda: Multiagent Simulation for Designing Social Services
- Papers for next week finalized soon
Logistics

• FAI talk on Friday at 11 GDC 6.302
  – Itsuki Noda: Multiagent Simulation for Designing Social Services

• Papers for next week finalized soon

• Grades coming ASAP
3D Uniform Color Auction

- Auction off uniform colors: Black, Blue, Brown, Cyan, Green, Orange, Pink, Purple, Red, White, Yellow
3D Uniform Color Auction

- Auction off uniform colors: Black, Blue, Brown, Cyan, Green, Orange, Pink, Purple, Red, White, Yellow

- Sequential auction
3D Uniform Color Auction

• Auction off uniform colors: Black, Blue, Brown, Cyan, Green, Orange, Pink, Purple, Red, White, Yellow

• Sequential auction

• Everyone gets 100 points
3D Uniform Color Auction

- Auction off uniform colors: Black, Blue, Brown, Cyan, Green, Orange, Pink, Purple, Red, White, Yellow

- Sequential auction

- Everyone gets 100 points

- Single simultaneous bid - only bid integers unless bidding maximum points
  - Winner gets color, random tie breaker if necessary
  - Losing bids charged 50% of bid
3D Uniform Color Auction

• Auction off uniform colors: Black, Blue, Brown, Cyan, Green, Orange, Pink, Purple, Red, White, Yellow

• Sequential auction

• Everyone gets 100 points

• Single simultaneous bid - only bid integers unless bidding maximum points
  – Winner gets color, random tie breaker if necessary
  – Losing bids charged 50% of bid

• Secondary market - trade later if you want
3D Uniform Color Auction Discussion

- Who got first choice color, second choice, etc.?
3D Uniform Color Auction Discussion

- Who got first choice color, second choice, etc.?
- Pros and cons of auction mechanism?
3D Uniform Color Auction Discussion

• Who got first choice color, second choice, etc.?  
• Pros and cons of auction mechanism?  
• How can the auction mechanism be improved?
Trading Agent Competition

• Put forth as a **benchmark problem** for e-marketplaces (Wellman, Wurman, et al., 2000)

• Autonomous agents act as **travel agents**
Trading Agent Competition

- Put forth as a **benchmark problem** for e-marketplaces (Wellman, Wurman, et al., 2000)

- Autonomous agents act as **travel agents**
  - **Game**: 8 agents, 12 min.
  - **Agent**: simulated travel agent with 8 clients
  - **Client**: TACtown ↔ Tampa within 5-day period
Trading Agent Competition

- Put forth as a **benchmark problem** for e-marketplaces (Wellman, Wurman, et al., 2000)

- Autonomous agents act as **travel agents**
  - **Game**: 8 agents, 12 min.
  - **Agent**: simulated travel agent with 8 clients
  - **Client**: TACtown ↔ Tampa within 5-day period

- **Auctions** for flights, hotels, entertainment tickets
  - **Server** maintains markets, sends prices to agents
  - Agent sends bids to server **over network**
Flights: Inflight days 1-4, Outflight days 2-5 (8)

- Unlimited supply; prices tend to increase; immediate clear; no resale
28 Simultaneous Auctions

**Flights:** Inflight days 1-4, Outflight days 2-5 (8)
- Unlimited supply; prices tend to increase; immediate clear; no resale

**Hotels:** Tampa Towers/Shoreline Shanties days 1-4 (8)
- 16 rooms per auction; 16th-price ascending auction; quote is ask price; no resale
- Random auction closes minutes 4 – 11
28 Simultaneous Auctions

Flights: Inflight days 1-4, Outflight days 2-5 (8)

- Unlimited supply; prices tend to increase; immediate clear; no resale

Hotels: Tampa Towers/Shoreline Shanties days 1-4 (8)

- 16 rooms per auction; 16th-price ascending auction; quote is ask price; no resale
- Random auction closes minutes 4 – 11

Entertainment: Wrestling/Museum/Park days 1-4 (12)

- Continuous double auction; initial endowments; quote is bid-ask spread; resale allowed
Client Preferences and Utility

Preferences: randomly generated per client

- Ideal arrival, departure days
- Good Hotel Value
- Entertainment Values
Client Preferences and Utility

Preferences: randomly generated per client
  - Ideal arrival, departure days
  - Good Hotel Value
  - Entertainment Values

Utility: 1000 (if valid) – travel penalty + hotel bonus + entertainment bonus
Client Preferences and Utility

Preferences: randomly generated per client
- Ideal arrival, departure days
- Good Hotel Value
- Entertainment Values

Utility: 1000 (if valid) – travel penalty + hotel bonus + entertainment bonus

Score: Sum of client utilities – expenditures
Allocation

\[ G \equiv \text{complete allocation of goods to clients} \]

\[ v(G) \equiv \text{utility of } G - \text{cost of needed goods} \]

\[ G^* \equiv \text{argmax } v(G') \]
Allocation

\[ G \equiv \text{complete allocation of goods to clients} \]
\[ v(G) \equiv \text{utility of } G \text{ – cost of needed goods} \]
\[ G^* \equiv \arg\max v(G) \]

Given holdings and prices, find \( G^* \)
Allocation

\[ G \equiv \text{complete allocation of goods to clients} \]
\[ v(G) \equiv \text{utility of } G \text{ – cost of needed goods} \]
\[ G^* \equiv \text{argmax } v(G') \]

Given holdings and prices, find \( G^* \)

- General allocation NP-complete
  - Tractable in TAC: mixed-integer LP (ATTac-2000)
  - Estimate \( v(G^*) \) quickly with LP relaxation
Allocation

\[ G \equiv \text{complete allocation of goods to clients} \]
\[ v(G) \equiv \text{utility of } G - \text{cost of needed goods} \]
\[ G^* \equiv \text{argmax } v(G) \]

Given holdings and prices, find \( G^* \)

- General allocation NP-complete
  - Tractable in TAC: mixed-integer LP (ATTac-2000)
  - Estimate \( v(G^*) \) quickly with LP relaxation

Prices known \( \Rightarrow G^* \) known \( \Rightarrow \) optimal bids known
High-Level Strategy

- Learn model of expected hotel price
High-Level Strategy

- Learn model of expected hotel price distributions
High-Level Strategy

- Learn model of expected hotel price distributions
- For each auction:
  - Repeatedly sample price vector from distributions
High-Level Strategy

• Learn model of expected hotel price distributions

• For each auction:
  - Repeatedly sample price vector from distributions
  - Bid avg marginal expected utility: \( v(G_w^*) - v(G_l^*) \)
High-Level Strategy

- Learn model of expected hotel price distributions

- For each auction:
  - Repeatedly sample price vector from distributions
  - Bid avg marginal expected utility: $v(G_w^*) - v(G_l^*)$

- Bid for all goods — not just those in $G^*$
High-Level Strategy

- Learn model of expected hotel price distributions
- For each auction:
  - Repeatedly sample price vector from distributions
  - Bid avg marginal expected utility: $v(G_w^*) - v(G_l^*)$
- Bid for all goods — not just those in $G^*$

Goal: analytically calculate optimal bids
Hotel Price Prediction

- Current hotel and flight prices
- Current time in game
- Hotel closing times
- Agents in the game (when known)
- Variations of the above
Hotel Price Prediction

• **Features:**
  - Current hotel and flight prices
  - Current time in game
  - Hotel closing times
  - Agents in the game (when known)
  - Variations of the above

• **Data:**
  - Hundreds of seeding round games
Hotel Price Prediction

- **Features:**
  - Current hotel and flight prices
  - Current time in game
  - Hotel closing times
  - Agents in the game (when known)
  - Variations of the above

- **Data:**
  - Hundreds of seeding round games
  - Assumption: similar economy
Hotel Price Prediction

- **Features:**
  - Current hotel and flight prices
  - Current time in game
  - Hotel closing times
  - Agents in the game (when known)
  - Variations of the above

- **Data:**
  - Hundreds of seeding round games
  - Assumption: similar economy
  - Features $\rightarrow$ actual prices
The Learning Algorithm

• $X \equiv \text{feature vector } \in \mathbb{R}^n$

• $Y \equiv \text{closing price} - \text{current price } \in \mathbb{R}$
The Learning Algorithm

• $X \equiv \text{feature vector } \in \mathbb{R}^n$

• $Y \equiv \text{closing price} - \text{current price } \in \mathbb{R}$

• Break $Y$ into $k \approx 50$ cut points $b_1 \leq \cdots \leq b_k$
The Learning Algorithm

- \( X \equiv \text{feature vector} \in \mathbb{R}^n \)
- \( Y \equiv \text{closing price} - \text{current price} \in \mathbb{R} \)
- Break \( Y \) into \( k \approx 50 \) cut points \( b_1 \leq \cdots \leq b_k \)
- For each \( b_i \), estimate probability \( Y \geq b_i \), given \( X \)
The Learning Algorithm

- \( X \equiv \text{feature vector} \in \mathbb{R}^n \)
- \( Y \equiv \text{closing price} - \text{current price} \in \mathbb{R} \)
- Break \( Y \) into \( k \approx 50 \) cut points \( b_1 \leq \cdots \leq b_k \)
- For each \( b_i \), estimate probability \( Y \geq b_i \), given \( X \)
  - Say \( X \) belongs to class \( C_i \) if \( Y \geq b_i \)
The Learning Algorithm

- \( X \equiv \text{feature vector} \in \mathbb{R}^n \)
- \( Y \equiv \text{closing price} - \text{current price} \in \mathbb{R} \)
- Break \( Y \) into \( k \approx 50 \) cut points \( b_1 \leq \cdots \leq b_k \)
- For each \( b_i \), estimate probability \( Y \geq b_i \), given \( X \)
  - Say \( X \) belongs to class \( C_i \) if \( Y \geq b_i \)
  - \( k \)-class problem: each example in many classes
The Learning Algorithm

- $X \equiv \text{feature vector} \in \mathbb{R}^n$
- $Y \equiv \text{closing price} - \text{current price} \in \mathbb{R}$
- Break $Y$ into $k \approx 50$ cut points $b_1 \leq \cdots \leq b_k$
- For each $b_i$, estimate probability $Y \geq b_i$, given $X$
  - Say $X$ belongs to class $C_i$ if $Y \geq b_i$
  - $k$-class problem: each example in many classes
  - Use BoostTexter (boosting (Schapire, 1990))
The Learning Algorithm

- \( X \equiv \text{feature vector} \in \mathbb{R}^n \)
- \( Y \equiv \text{closing price} - \text{current price} \in \mathbb{R} \)
- Break \( Y \) into \( k \approx 50 \) cut points \( b_1 \leq \cdots \leq b_k \)
- For each \( b_i \), estimate probability \( Y \geq b_i \), given \( X \)
  - Say \( X \) belongs to class \( C_i \) if \( Y \geq b_i \)
  - \( k \)-class problem: each example in many classes
  - Use BoostTexter (boosting (Schapire, 1990))
- Can convert to estimated distribution of \( Y|X \)
The Learning Algorithm

- \( X \equiv \text{feature vector} \in \mathbb{R}^n \)
- \( Y \equiv \text{closing price} - \text{current price} \in \mathbb{R} \)
- Break \( Y \) into \( k \approx 50 \) cut points \( b_1 \leq \cdots \leq b_k \)
- For each \( b_i \), estimate probability \( Y \geq b_i \), given \( X \)
  - Say \( X \) belongs to class \( C_i \) if \( Y \geq b_i \)
  - \( k \)-class problem: each example in many classes
  - Use BoostTexter (boosting (Schapire, 1990))
- Can convert to estimated distribution of \( Y|X \)

New algorithm for conditional density estimation
Hotel Expected Values

- Repeat until time bound, for each hotel:
  1. Assume this hotel closes next
Hotel Expected Values

- Repeat until time bound, for each hotel:
  1. Assume this hotel closes next
  2. Sample prices from predicted price distributions
Hotel Expected Values

- Repeat until time bound, for each hotel:
  1. Assume this hotel closes next
  2. Sample prices from predicted price distributions
  3. Given these prices compute $V_0, V_1, \ldots V_8$
     - $V_i = v(G^*)$ if own exactly $i$ of the hotel
     - $V_0 \leq V_1 \leq \ldots \leq V_8$
Hotel Expected Values

- Repeat until time bound, for each hotel:
  1. Assume this hotel closes next
  2. Sample prices from predicted price distributions
  3. Given these prices compute $V_0, V_1, \ldots, V_8$
     - $V_i = v(G^*)$ if own exactly $i$ of the hotel
     - $V_0 \leq V_1 \leq \ldots \leq V_8$

- Value of $i$th copy is $\text{avg}(V_i - V_{i-1})$
Other Uses of Sampling

**Flights:** Cost/benefit analysis for postponing commitment
Other Uses of Sampling

**Flights:** Cost/benefit analysis for postponing commitment

**Cost:** Price expected to rise over next $n$ minutes

**Benefit:** More price info becomes known

- Compute expected marginal value of buying some different flight
Other Uses of Sampling

**Flights:** Cost/benefit analysis for postponing commitment

- **Cost:** Price expected to rise over next $n$ minutes
- **Benefit:** More price info becomes known
  - Compute expected marginal value of buying some different flight

**Entertainment:** Bid more (ask less) than expected value of having one more (fewer) ticket
## Finals

<table>
<thead>
<tr>
<th>Team</th>
<th>Avg</th>
<th>Adj</th>
<th>Institution</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTac</td>
<td>3622</td>
<td>4154</td>
<td>AT&amp;T</td>
</tr>
<tr>
<td>livingagents</td>
<td>3670</td>
<td>4094</td>
<td>Living Systems (Germ.)</td>
</tr>
<tr>
<td>whitebear</td>
<td>3513</td>
<td>3931</td>
<td>Cornell</td>
</tr>
<tr>
<td>Urlaub01</td>
<td>3421</td>
<td>3909</td>
<td>Penn State</td>
</tr>
<tr>
<td>Retsina</td>
<td>3352</td>
<td>3812</td>
<td>CMU</td>
</tr>
<tr>
<td>CaiserSose</td>
<td>3074</td>
<td>3766</td>
<td>Essex (UK)</td>
</tr>
<tr>
<td>Southampton</td>
<td>3253*</td>
<td>3679</td>
<td>Southampton (UK)</td>
</tr>
<tr>
<td>TacsMan</td>
<td>2859</td>
<td>3338</td>
<td>Stanford</td>
</tr>
</tbody>
</table>

- ATTac improves over time
- livingagents is an open-loop strategy
Controlled Experiments

- $ATTac_s$: “full-strength” agent based on boosting
Controlled Experiments

- $ATTac_s$: "full-strength" agent based on boosting
- $SimpleMean_s$: sample from empirical distribution (previously played games)
Controlled Experiments

- $ATTac_s$: “full-strength” agent based on boosting

- $SimpleMean_s$: sample from empirical distribution (previously played games)

- $ConditionalMean_s$: condition on closing time
Controlled Experiments

- \textit{ATTac}_s: "full-strength" agent based on boosting
- \textit{SimpleMean}_s: sample from empirical distribution (previously played games)
- \textit{ConditionalMean}_s: condition on closing time
- \textit{ATTac}_n_s, \textit{ConditionalMean}_n_s, \textit{SimpleMean}_n_s: predict expected value of the distribution
Controlled Experiments

- $ATTac_s$: “full-strength” agent based on boosting

- $SimpleMean_s$: sample from empirical distribution (previously played games)

- $ConditionalMean_s$: condition on closing time

- $ATTac_{ns}, ConditionalMean_{ns}, SimpleMean_{ns}$: predict expected value of the distribution

- $CurrentPrice$: predict no change
Controlled Experiments

- \( ATTac_s \): “full-strength” agent based on boosting
- \( SimpleMean_s \): sample from empirical distribution (previously played games)
- \( ConditionalMean_s \): condition on closing time
- \( ATTac_{ns}, ConditionalMean_{ns}, SimpleMean_{ns} \): predict expected value of the distribution
- \( CurrentPrice \): predict no change
- \( EarlyBidder \): motivated by TAC-01 entry livingagents
Controlled Experiments

- **\( \text{ATTac}_s \):** "full-strength" agent based on boosting
- **\( \text{SimpleMean}_s \):** sample from empirical distribution (previously played games)
- **\( \text{ConditionalMean}_s \):** condition on closing time
- **\( \text{ATTac}_{ns}, \text{ConditionalMean}_{ns}, \text{SimpleMean}_{ns} \):** predict expected value of the distribution
- **\( \text{CurrentPrice} \):** predict no change
- **\( \text{EarlyBidder} \):** motivated by TAC-01 entry livingagents
  - Immediately bids high for \( G^* \) (with \( \text{SimpleMean}_{ns} \))
  - Goes to sleep
Stability

7 EarlyBidder’s with 1 ATTac

<table>
<thead>
<tr>
<th>Agent</th>
<th>Score</th>
<th>Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTac</td>
<td>2431 ± 464</td>
<td>8909 ± 264</td>
</tr>
<tr>
<td>EarlyBidder</td>
<td>−4880 ± 337</td>
<td>9870 ± 34</td>
</tr>
</tbody>
</table>
Stability

• 7 *EarlyBidder’s* with 1 *ATTac*

<table>
<thead>
<tr>
<th>Agent</th>
<th>Score</th>
<th>Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTac</td>
<td>2431 ± 464</td>
<td>8909 ± 264</td>
</tr>
<tr>
<td>EarlyBidder</td>
<td>−4880 ± 337</td>
<td>9870 ± 34</td>
</tr>
</tbody>
</table>

• 7 *ATTac’s* with 1 *EarlyBidder*

<table>
<thead>
<tr>
<th>Agent</th>
<th>Score</th>
<th>Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTac</td>
<td>2578 ± 25</td>
<td>9650 ± 21</td>
</tr>
<tr>
<td>EarlyBidder</td>
<td>2869 ± 69</td>
<td>10079 ± 55</td>
</tr>
</tbody>
</table>
Stability

• 7 EarlyBidder’s with 1 ATTac

<table>
<thead>
<tr>
<th>Agent</th>
<th>Score</th>
<th>Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTac</td>
<td>2431 ± 464</td>
<td>8909 ± 264</td>
</tr>
<tr>
<td>EarlyBidder</td>
<td>−4880 ± 337</td>
<td>9870 ± 34</td>
</tr>
</tbody>
</table>

• 7 ATTac’s with 1 EarlyBidder

<table>
<thead>
<tr>
<th>Agent</th>
<th>Score</th>
<th>Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTac</td>
<td>2578 ± 25</td>
<td>9650 ± 21</td>
</tr>
<tr>
<td>EarlyBidder</td>
<td>2869 ± 69</td>
<td>10079 ± 55</td>
</tr>
</tbody>
</table>

*EarlyBidder* gets more utility; *ATTac* pays less
Results

• *Phase I*: Training from TAC-01 (seeding round, finals)
Results

- **Phase I**: Training from TAC-01 (seeding round, finals)
- **Phase II**: Training from TAC-01, phases I, II
Results

- **Phase I**: Training from TAC-01 (seeding round, finals)
- **Phase II**: Training from TAC-01, phases I, II
- **Phase III**: Training from phases I – III
## Results

- **Phase I**: Training from TAC-01 (seeding round, finals)
- **Phase II**: Training from TAC-01, phases I, II
- **Phase III**: Training from phases I – III

<table>
<thead>
<tr>
<th>Agent</th>
<th>Relative Score</th>
<th>Phase I</th>
<th>Phase III</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTac(_{ns})</td>
<td>105.2 ± 49.5 (2)</td>
<td>166.2 ± 20.8 (1)</td>
<td></td>
</tr>
<tr>
<td>ATTac(_{s})</td>
<td>27.8 ± 42.1 (3)</td>
<td>122.3 ± 19.4 (2)</td>
<td></td>
</tr>
<tr>
<td>EarlyBidder</td>
<td>140.3 ± 38.6 (1)</td>
<td>117.0 ± 18.0 (3)</td>
<td></td>
</tr>
<tr>
<td>SimpleMean(_{ns})</td>
<td>−28.8 ± 45.1 (5)</td>
<td>−11.5 ± 21.7 (4)</td>
<td></td>
</tr>
<tr>
<td>SimpleMean(_{s})</td>
<td>−72.0 ± 47.5 (7)</td>
<td>−44.1 ± 18.2 (5)</td>
<td></td>
</tr>
<tr>
<td>ConditionalMean(_{ns})</td>
<td>8.6 ± 41.2 (4)</td>
<td>−60.1 ± 19.7 (6)</td>
<td></td>
</tr>
<tr>
<td>ConditionalMean(_{s})</td>
<td>−147.5 ± 35.6 (8)</td>
<td>−91.1 ± 17.6 (7)</td>
<td></td>
</tr>
<tr>
<td>CurrentPrice</td>
<td>−33.7 ± 52.4 (6)</td>
<td>−198.8 ± 26.0 (8)</td>
<td></td>
</tr>
</tbody>
</table>
Last-minute bidding (R,O, 2001)

- eBay: first-price, ascending auction
- Amazon: auction extended if bid in last 10 minutes
- eBay: bots exist to incrementally raise your bid to a maximum

● Still people *snipe*. Why?
  - There’s a risk that the bid might not make it
  - However, common-value \[\rightarrow\] bid conveys info
  - Late-bidding can be seen as implicit collusion
  - Or . . . , lazy, unaware, etc. (Amazon and eBay)

● Finding: more late-bidding on eBay,
  - even more on antiques rather than computers

Small design-difference matters
Late Bidding as Best Response

- Good vs. incremental bidders
  - They start bidding low, plan to respond
  - Doesn’t give them time to respond

- Good vs. other snipers
  - Implicit collusion
  - Both bid low, chance that one bid doesn’t get in

- Good in common-value case
  - Protects information

Overall, the analysis of multiple bids supports the hypothesis that last-minute bidding arises at least in part as a response by sophisticated bidders to unsophisticated incremental bidding.
Other TAC competitions

- Supply Chain Management
- Ad Auctions
- Power
Discussion

- Are these agents useful for the real version of these tasks?
Discussion

• Are these agents useful for the real version of these tasks?
• What can we learn from these competitions?
Discussion

- Are these agents useful for the real version of these tasks?
- What can we learn from these competitions?
- General strategy that works well?
Discussion

- Are these agents useful for the real version of these tasks?
- What can we learn from these competitions?
- General strategy that works well?