CS344M
Autonomous Multiagent Systems

Prof: Peter Stone

Department of Computer Science
The University of Texas at Austin

Good Afternoon, Colleagues

Are there any questions?

Good Afternoon, Colleagues

Are there any questions?

- FCC: did they acheive “bundles”?
- FCC: what was the “optimal” strategy?

Good Afternoon, Colleagues

Are there any questions?

- FCC: did they acheive “bundles”?
- FCC: what was the “optimal” strategy?
- What’s new in TAC?
- Do algorithms scale with more clients?
- Was TAC SCM more successful?

Logistics

- FAI talk on Friday
  – Jeopardy agent (Fri., 11am 2.302)

Logistics

- FAI talk on Friday
  – Jeopardy agent (Fri., 11am 2.302)
- Final tournament: Monday 12/13, 9-noon

Logistics

- FAI talk on Friday
  - Jeopardy agent (Fri., 11am 2.302)
- Final tournament: Monday 12/13, 9-noon
- Peer review process — thoughts?

FCC Spectrum Auction #35

- 422 licences in 195 markets (cities)
  - 80 bidders spent $8 billion
  - ran Dec 12 - Jan 26 2001
  - licence is a 10 or 15 mhz spectrum chunk
- Run in rounds
  - bid on each licence you want each round
  - simultaneous; break ties by arrival time
  - current winner and all bids are known
- Allowable bids: 1 to 9 bid increments
  - 1 bid incr is 10% – 20% of current price
- Other complex rules

Model

- Agent goals
  - desire 0, 1, or 2 licences per market
  - desired markets have unique values
  - subject to budget constraint

Assumption: no inter-market value dependencies

- Utility is profit: $\Sigma_l (value - cost)$
- modeled 5 most important bidders
  - others served mainly to raise prices
  - modeled as several small bidders
  - lower valuations (75% $\rightarrow$ pessimistic)

Bidding Strategies

- Considering self only
  - Knapsack
  - best self-only approach
- Strategic bidding (consider others)
  - threats
  - budget stretching
  - Strategic Demand Reduction (SDR)

Explicit communication not allowed

Randomized SDR

- **Figure out allocations dynamically**
  - round 1: bid for everything you want
  - first big bidder winning bid owns licence
  - satisfaction = owned value / desired value
- **Random $\Rightarrow$ uneven allocation**
  - get small share $\Rightarrow$ incentive to cheat
  - fair: own satisfaction close to average
  - if unlucky, take licences until fair
- **Small bidders take licences from owners**
  - remember licence's owner
  - allocate while small bidders active

RSDR vs. Knapsack

Method	Agent	Profit ($M)	Ratio	Cost
Knapsack	0	980 (±170)	1.00	.82
	1	650 (±85)	1.00	.82
	2	830 (±91)	1.00	.84
	3	170 (±20)	1.00	.84
	4	550 (±96)	1.00	.86
RSDR	0	1240 (±210)	1.26	.76
	1	820 (±83)	1.25	.77
	2	1300 (±290)	1.58	.74
	3	300 (±44)	1.78	.79
	4	930 (±240)	1.68	.76

44% more profit; avg. ratio 1.51

Robustness

- What if someone cheats?
  - cheat: defect back to knapsack
  - others stay out of its way $\Rightarrow$ big win
- Solution: Punishing RSDR (PRSDR)
  - cheater takes your licence $\Rightarrow$ take it back
  - take it back first while still have money
  - aggressively punitive: skips optimizers

Simplification: pointing out cheaters by hand

Robustness

Method	*Ratio*	*Cost*
Knapsack	1.00	.84
RSDR	1.51	.76
RSDR Cheater	1.63	.76
RSDR Victim	1.22	.79
PRSDR Cheater	1.02	.83
PRSDR Enforcer	1.17	.81

Extensions

- **Change small bidder valuations**
  - test robustness
  - RSDR is optimal for preserving profit

- **Multiple cheaters**
  - current punishment too aggressive
  - collapse back to knapsack instead

Extentions

Method	*Ratio*	*Local Ratio*	*Cost*
Multiple Cheater	1.03	1.03	.84
Multiple Enforcer	1.01	1.01	.83
50% Knapsack	1.70	1.00	.74
50% RSDR	3.42	2.02	.51
75% Knapsack	1.00	1.00	.84
75% RSDR	1.51	1.51	.76
85% Knapsack	0.68	1.00	.89
85% RSDR	0.81	1.25	.87

Future Work

- Robustness enhancements
  - better punishment method
- More complex value functions
  - inter-market dependencies
- Automatic cheater detection
  - partial cheating vs. detection arms race
  - smack back into compliance
- Generalization to other auctions
  - more robust to tie-breaking procedure variations

Summary

- Communication-free coordination
- Enables much higher profits
- Works even uncertain knowledge
- Real-world functionality relies on simple assumptions:

Summary

- Communication-free coordination
- Enables much higher profits
- Works even uncertain knowledge
- Real-world functionality relies on simple assumptions:
  - bidders want more profit
  - bidders familiar with PRSDR and its benefits
  - bidders willing to try it risk-free

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 $\leftrightarrow$ 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 $\leftrightarrow$ 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**

28 Simultaneous Auctions

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

$$
\begin{aligned}
G &\equiv \text{complete allocation of goods to clients} \\
v(G) &\equiv \text{utility of } G - \text{cost of needed goods} \\
G^* &\equiv \text{argmax } v(G)
\end{aligned}
$$

Allocation

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

$\boxed{\text{Given holdings and prices, find } G^*}$

Allocation

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

$$\boxed{\text{Given holdings and prices, find } G^*}$$

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

Allocation

$$
\begin{aligned}
G &\equiv \text{complete allocation of goods to clients} \\
v(G) &\equiv \text{utility of } G - \text{cost of needed goods} \\
G^* &\equiv \text{argmax } v(G)
\end{aligned}
$$

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

- **Features:**
  - 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 $\mapsto$ actual prices

The Learning Algorithm

- $X \equiv$ feature vector $\in \mathbb{R}^n$
- $Y \equiv$ closing price $-$ current price $\in \mathbb{R}$

The Learning Algorithm

- $X \equiv$ feature vector $\in \mathbb{R}^n$
- $Y \equiv$ closing price $-$ current price $\in \mathbb{R}$
- Break Y into $k \approx 50$ cut points $b_1 \leq \cdots \leq b_k$

The Learning Algorithm

- $X \equiv$ feature vector $\in \mathbb{R}^n$
- $Y \equiv$ closing price $-$ 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

- $X \equiv$ feature vector $\in \mathbb{R}^n$
- $Y \equiv$ closing price $-$ 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

- $X \equiv$ feature vector $\in \mathbb{R}^n$
- $Y \equiv$ closing price $-$ 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

- $X \equiv$ feature vector $\in \mathbb{R}^n$
- $Y \equiv$ closing price $-$ 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 **BoosTexter** (boosting (Schapire, 1990))

The Learning Algorithm

- $X \equiv$ feature vector $\in \mathbb{R}^n$
- $Y \equiv$ closing price $-$ 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 **BoosTexter** (boosting (Schapire, 1990))
- Can convert to estimated distribution of $Y|X$

The Learning Algorithm

- $X \equiv$ feature vector $\in \mathbb{R}^n$
- $Y \equiv$ closing price $-$ 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 **BoosTexter** (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 ith copy is 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

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

- 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

- *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

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

- *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
  - Immediately bids high for G^* (with *SimpleMean*$_{ns}$)
  - Goes to sleep

Stability

- 7 *EarlyBidder*'s with 1 *ATTac*

Agent	Score	Utility
ATTac	2431 ± 464	8909 ± 264
EarlyBidder	-4880 ± 337	9870 ± 34

Stability

- 7 *EarlyBidder*'s with 1 *ATTac*

Agent	Score	Utility
ATTac	2431 ± 464	8909 ± 264
EarlyBidder	-4880 ± 337	9870 ± 34

- 7 *ATTac*'s with 1 *EarlyBidder*

Agent	Score	Utility
ATTac	2578 ± 25	9650 ± 21
EarlyBidder	2869 ± 69	10079 ± 55

Stability

- 7 *EarlyBidder*'s with 1 *ATTac*

Agent	Score	Utility
ATTac	2431 ± 464	8909 ± 264
EarlyBidder	-4880 ± 337	9870 ± 34

- 7 *ATTac*'s with 1 *EarlyBidder*

Agent	Score	Utility
ATTac	2578 ± 25	9650 ± 21
EarlyBidder	2869 ± 69	10079 ± 55

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

Agent	*Relative Score*	
	Phase I	*Phase III*
ATTac$_{ns}$	105.2 ± 49.5 (2)	166.2 ± 20.8 (1)
ATTac$_{s}$	27.8 ± 42.1 (3)	122.3 ± 19.4 (2)
EarlyBidder	140.3 ± 38.6 (1)	117.0 ± 18.0 (3)
SimpleMean$_{ns}$	-28.8 ± 45.1 (5)	-11.5 ± 21.7 (4)
SimpleMean$_{s}$	-72.0 ± 47.5 (7)	-44.1 ± 18.2 (5)
ConditionalMean$_{ns}$	8.6 ± 41.2 (4)	-60.1 ± 19.7 (6)
ConditionalMean$_{s}$	-147.5 ± 35.6 (8)	-91.1 ± 17.6 (7)
CurrentPrice	-33.7 ± 52.4 (6)	-198.8 ± 26.0 (8)

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 $\Longrightarrow$ 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.