MS&E 233
Game Theory, Data Science and AI
Lecture 17

Vasilis Syrgkanis

Assistant Professor

Management Science and Engineering

(by courtesy) Computer Science and Electrical Engineering

Institute for Computational and Mathematical Engineering

Course Evaluations

http://course-evaluations.stanford.edu/

Computational Game Theory for Complex Games

1

- Basics of game theory and zero-sum games (T)
- Basics of online learning theory (T)
- Solving zero-sum games via online learning (T)
- *HW1: implement simple algorithms to solve zero-sum games*
- Applications to ML and AI (T+A)
- *HW2: implement boosting as solving a zero-sum game*

2

- Basics of extensive-form games
- Solving extensive-form games via online learning (T)
- *HW3: implement agents to solve very simple variants of poker*

3

- General games, equilibria and online learning (T)
- Online learning in general games
- *HW4: implement no-regret algorithms that converge to correlated equilibria in general games*

Data Science for Auctions and Mechanisms

4

- Basics and applications of auction theory (T+A)
- Basic Auctions and Learning to bid in auctions (T)
- *HW5: implement bandit algorithms to bid in ad auctions*

5

- Optimal auctions and mechanisms (T)
- Simple vs optimal mechanisms (T)
- *HW6: implement simple and optimal auctions, analyze revenue empirically*

6

- Basics of Statistical Learning Theory (T)
- Optimizing Mechanisms from Samples (T)
- *HW7: implement procedures to learn approximately optimal auctions from historical samples*

Further Topics

7

- Econometrics in games and auctions (T+A)
- **A/B testing in markets (T+A)**
- *HW8: implement procedure to estimate values from bids in an auction*

Guest Lectures

- Mechanism Design for LLMs, Renato Paes Leme, Google Research
- Auto-bidding in Sponsored Search Auctions, Kshipra Bhawalkar, Google Research

What if all we want is to compare between auctions A and B in terms of revenue?

What I could potentially do is:
For each auction flip a coin;
If heads, then run auction A else run auction B

After many auctions compare average revenue from A auctions, vs., average revenue from B auctions

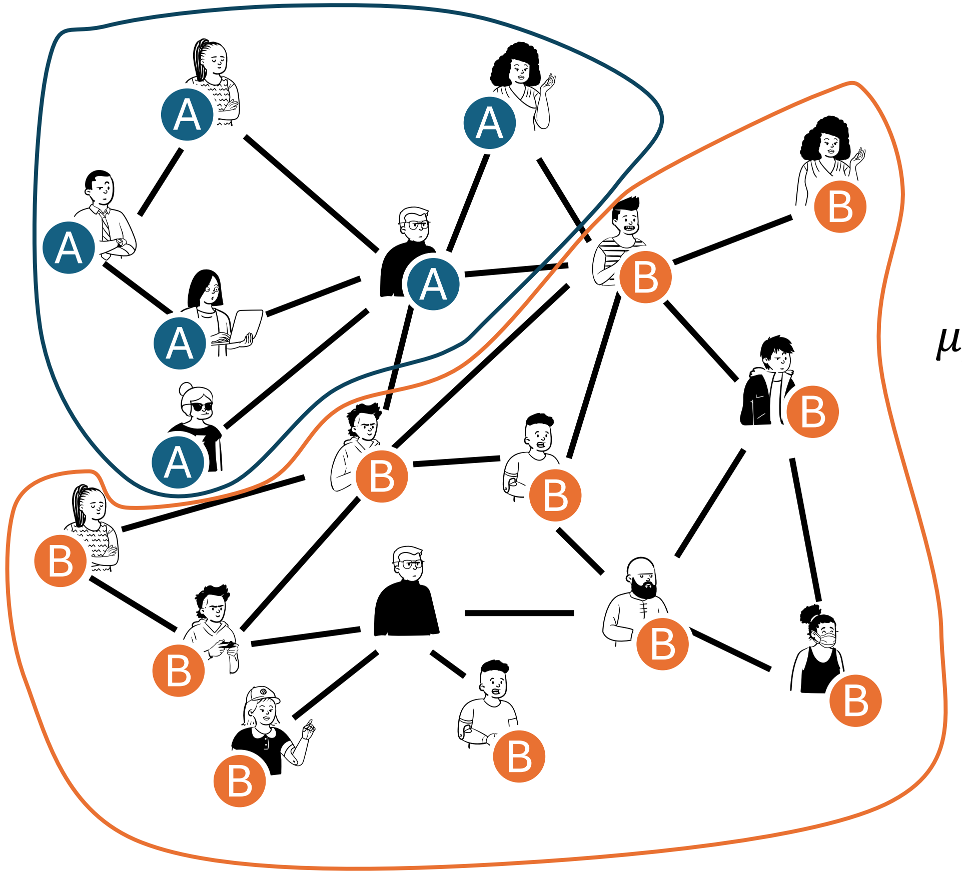
RCTs are the gold standard for measuring the “causal effect” of a “treatment” on an “outcome”

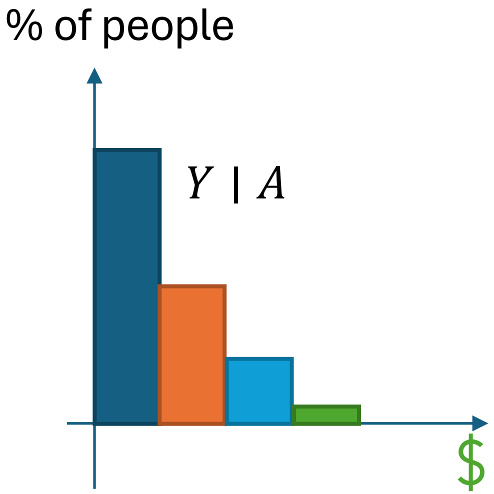
Interference!
The Big Challenge of A/B Testing in Markets and Platforms

Interference

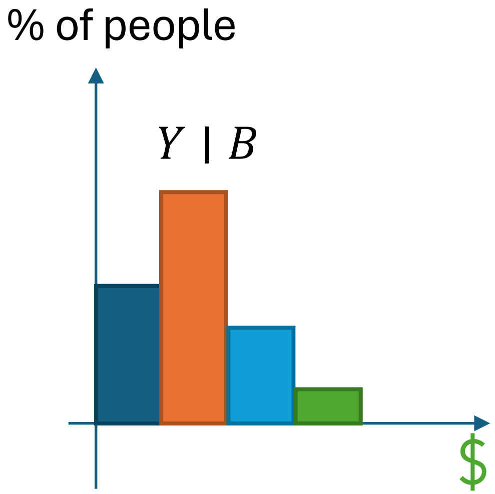
- Social Network interference
- Equilibrium effects
- Stateful systems and time effects

Approach: Clustering





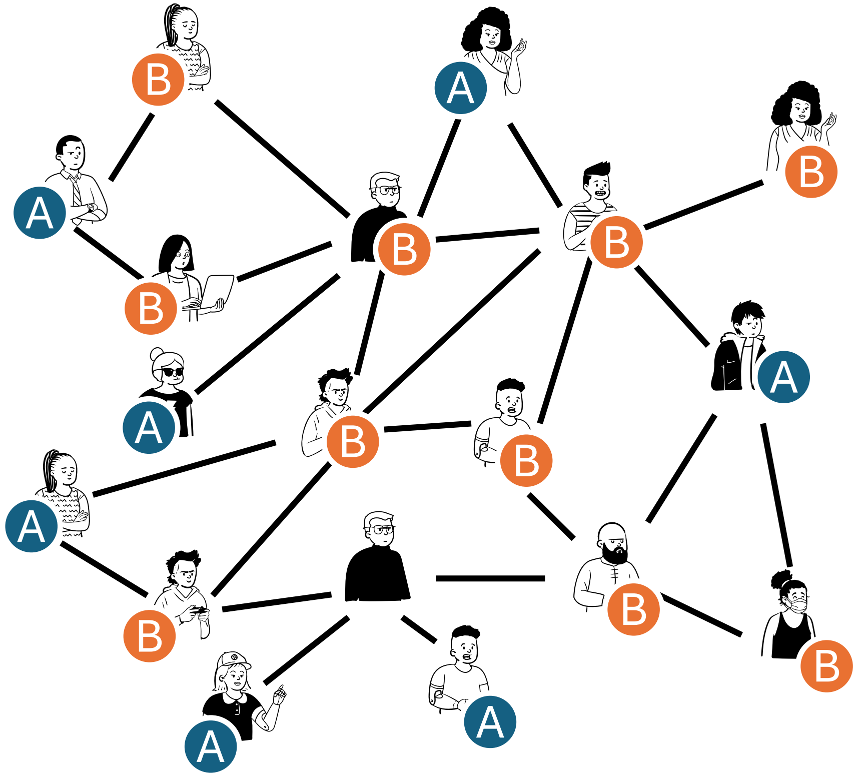
$\mu_A = 10\$$ (average spend)

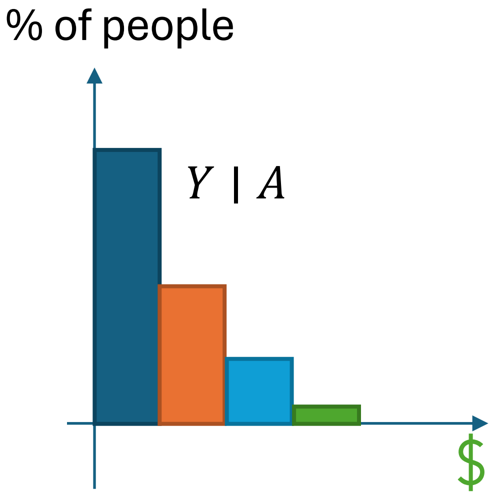


$\mu_B = 20\$$ (average spend)

Image Source: https://www.leadpages.com/blog/ab-testing-split-testing/

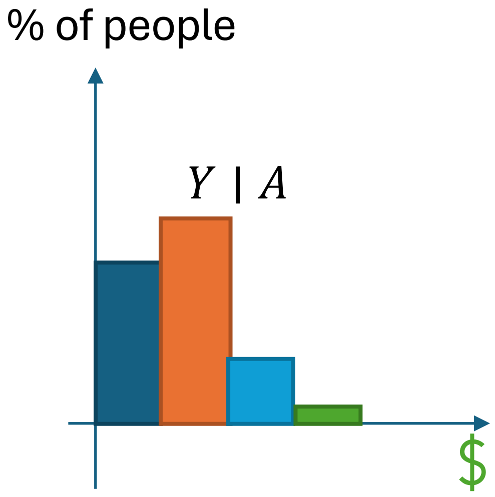
Approach: Structural Bias Correction





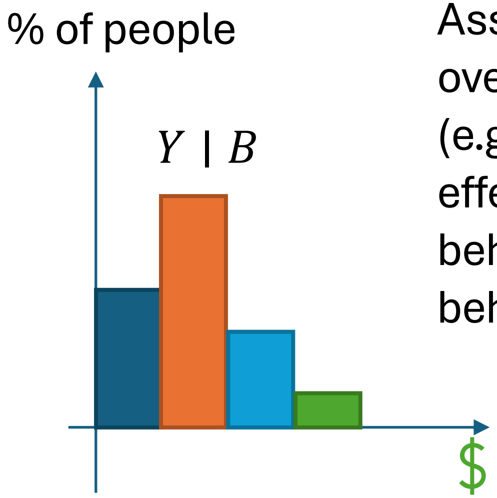
$\mu_A = 10\$$ (average spend)

Correct Spill-Over Bias

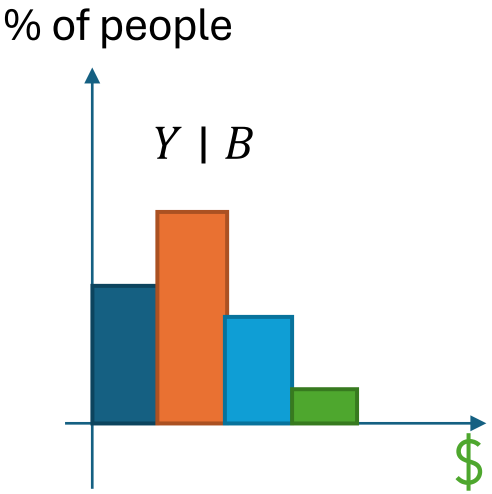


$\mu_A = 10\$$ (average spend)

Use Network Information + Assumptions on how spill-overs change outcome (e.g. additive homophily effects, market equilibrium behavior, Nash equilibrium behavior)



$\mu_B = 20\$$ (average spend)



$\mu_B = 20\$$ (average spend)

Image Source: https://www.leadpages.com/blog/ab-testing-split-testing/

A/B Testing in Auctions

A/B Testing over Position Auction Formats

We observe a bid distribution, described by the quantile function $b(q)$, from a randomized k-unit auction (which chooses each k with positive probability)

For any other randomized k-unit (with probabilities w_k) first-price auction among symmetric bidders, we have:

$$\text{Rev} = n \sum_{k \leq N} w_k \, E[b(q) \cdot f(q)]$$

for a function $f(q)$ known in closed form

With access to bidding data from a single randomized k-unit auction (which chooses each k with positive probability), we can estimate Rev of any other k-unit auction.

Estimate CDF of bids using the empirical CDF $\hat{G}$.
Then use $\hat{b} = \hat{G}^{-1}$ and

$$\widehat{\mathrm{Rev}} = n \sum_{k \le N} w_k \int_0^1 \hat{b}(q) \cdot f(q) dq$$

By convergence rates of empirical CDF, we can show:

$$\left|\widehat{\mathrm{Rev}} - \mathrm{Rev}\right| \lesssim 1/\sqrt{m}$$

A/B Testing across Many Keywords with Budgets

Budgets!

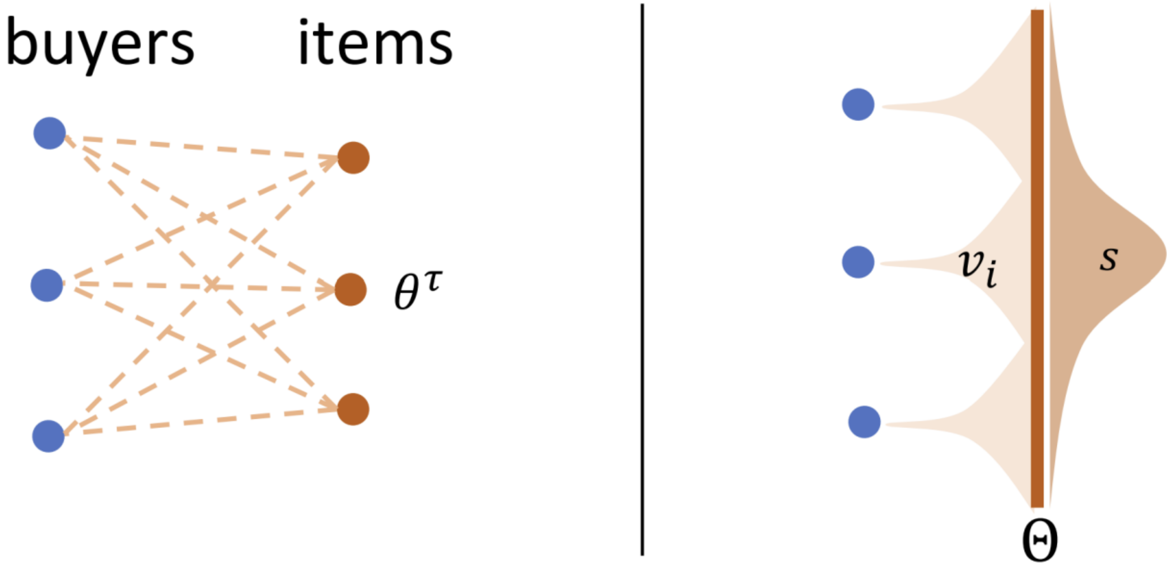
- So far we did not place any budget constraints on bidders
- In practice, budget constraints are very important
- Bidders participate in many auctions and have a budget limit
- Can only spend at most B_i in total across all the auctions

- This couples the bidding strategy across auctions
- Makes learning (e.g. no-regret learning hard)
- In its full generality a stochastic dynamic program

Simplified Budgets: Pacing Equilibria

Interference Among First-Price Pacing Equilibria: A Bias and Variance Analysis (arxiv.org)

- In practice, people use the following simplification
- We have n bidders and a continuum of items
- Items have type θ which follows some distribution with measure s
- $v_i(\theta)$ is bidder i's value for an item of type θ



Simplified Budgets: Pacing Equilibria

The multipliers $\beta = (\beta_1, \ldots, \beta_n)$ and price function $p(\theta)$ are a *pacing equilibrium* if there exists and allocation function $x(\theta)$ such that

- First-price payment: $p(\theta) = \max_i \beta_i v_i(\theta)$
- Highest-bidder wins: $x_i(\theta) \geq 0 \Rightarrow \beta_i v_i(\theta) = \max_k \beta_k v_k(\theta)$
- Budgets are respected

$$\int_\theta x_i(\theta) p(\theta) s(\theta) d\theta \leq B_i$$

- No-overselling: $\sum_i x_i(\theta) \leq 1$
- Full-allocation of competitive items: $p(\theta) > 0 \Rightarrow \sum_i x_i(\theta) = 1$
- No un-necessary pacing: $\int_\theta x_i(\theta) p(\theta) s(\theta) d\theta < B_i \Rightarrow \beta_i = 1$

Characterization of Pacing Equilibria

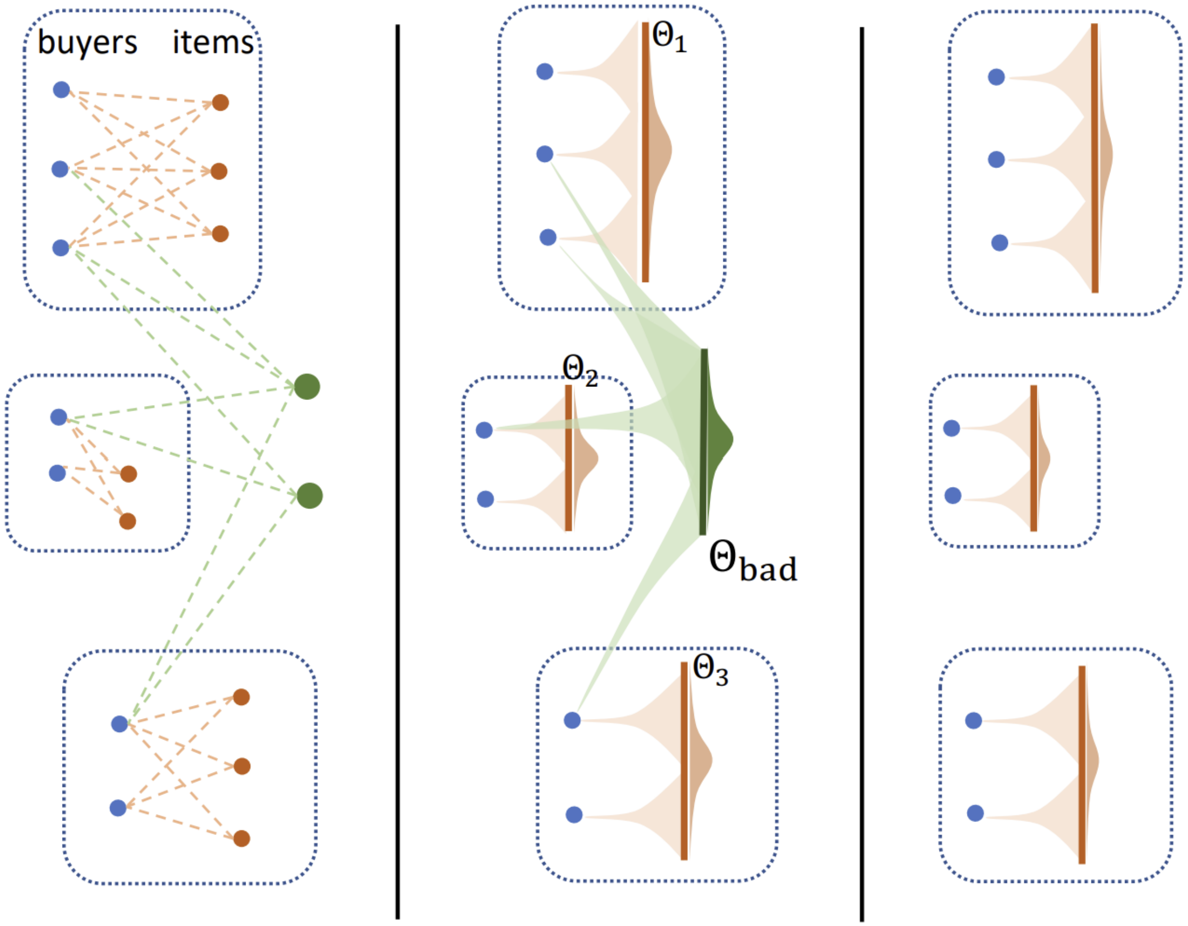
Multipliers in pacing equilibrium are characterized as solutions to a convex optimization problem (related to market equilibrium)

$$\beta_* = \underset{\beta \in (0,1]^n}{\operatorname{argmin}} E\left[\max_i \beta_i v_i(\theta)\right] - \sum_i B_i \log(\beta_i)$$

Clustered Experiment Designs and Debiasing

Interference Among First-Price Pacing Equilibria: A Bias and Variance Analysis (arxiv.org)

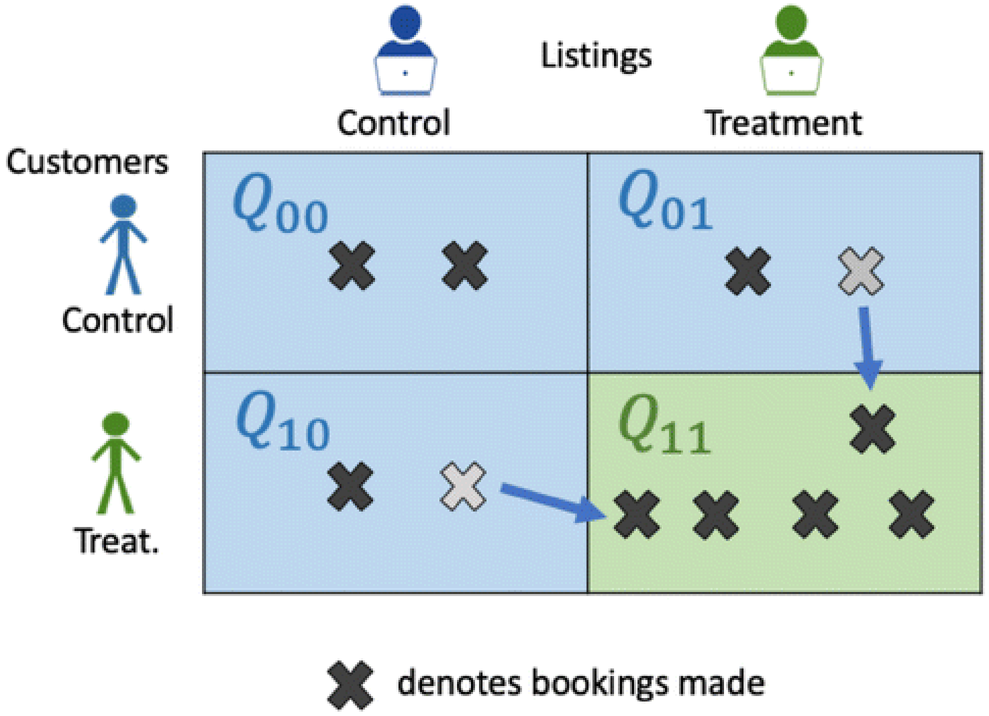
1. For each sub-market want pacing multipliers as if the bad items don't exist
2. With such multipliers, can estimate idealized revenue for each sub-market, as if isolated
3. Characterization of multipliers as minimizers of market equilibrium program ⇒ closed form first-order bias that bad items introduce
4. Subtract bias and measure revenue of A and B clusters using debiased multipliers



A/B Testing in Two-Sided Matching Markets

Two-Sided Randomized Designs

Experimental Design in Two-Sided Platforms: An Analysis of Bias | Management Science (informs.org)



Colab Notebook:
https://colab.research.google.com/drive/198fDKTPcXZm15cPwORoeYvDRrLfLHl9b?usp=sharing
Slide Version:
spring24/assets/presentations/AB_Testing_TwoSided_slides.html at master · stanford-msande233/spring24 (github.com)

Recap:
What did we learn?

Course Learning Objectives

- Learn the fundamentals of game theory
- Learn how game theory can be applied in many real-world settings (e.g. ad auctions, complex games)
- Learn the fundamentals of tools from data science and ML that are useful in game theoretic contexts (online learning theory, statistical learning theory, econometrics)
- Learn how these topics can be combined to
  - provide computational solutions to the design of agents that perform well in competitive environments
  - optimize and analyze markets, mechanisms and platforms from data
- Be able to implement and code up these solutions in Python

Computational Game Theory for Complex Games

1

- Basics of game theory and zero-sum games (T)
- Basics of online learning theory (T)
- Solving zero-sum games via online learning (T)
- *HW1: implement simple algorithms to solve zero-sum games*
- Applications to ML and AI (T+A)
- *HW2: implement boosting as solving a zero-sum game*

2

- Basics of extensive-form games
- Solving extensive-form games via online learning (T)
- *HW3: implement agents to solve very simple variants of poker*

3

- General games, equilibria and online learning (T)
- Online learning in general games (T+A)
- *HW4: implement no-regret algorithms that converge to correlated equilibria in general games*

Data Science for Auctions and Mechanisms

4

- Basics and applications of auction theory (T+A)
- Basic Auctions and Learning to bid in auctions (T)
- *HW5: implement bandit algorithms to bid in ad auctions*

5

- Optimal auctions and mechanisms (T)
- Simple vs optimal mechanisms (T)
- *HW6: implement simple and optimal auctions, analyze revenue empirically*

6

- Basics of Statistical Learning Theory (T)
- Optimizing Mechanisms from Samples (T)
- *HW7: implement procedures to learn approximately optimal auctions from historical samples*

Further Topics

7

- Econometrics in games and auctions (T+A)
- A/B testing in markets (T+A)
- *HW8: implement procedure to estimate values from bids in an auction, empirically analyze inaccuracy of A/B tests in markets*

Guest Lectures

- Mechanism Design for LLMs, Renato Paes Leme, Google Research
- Auto-bidding in Sponsored Search Auctions, Kshipra Bhawalkar, Google Research

Lecture 1

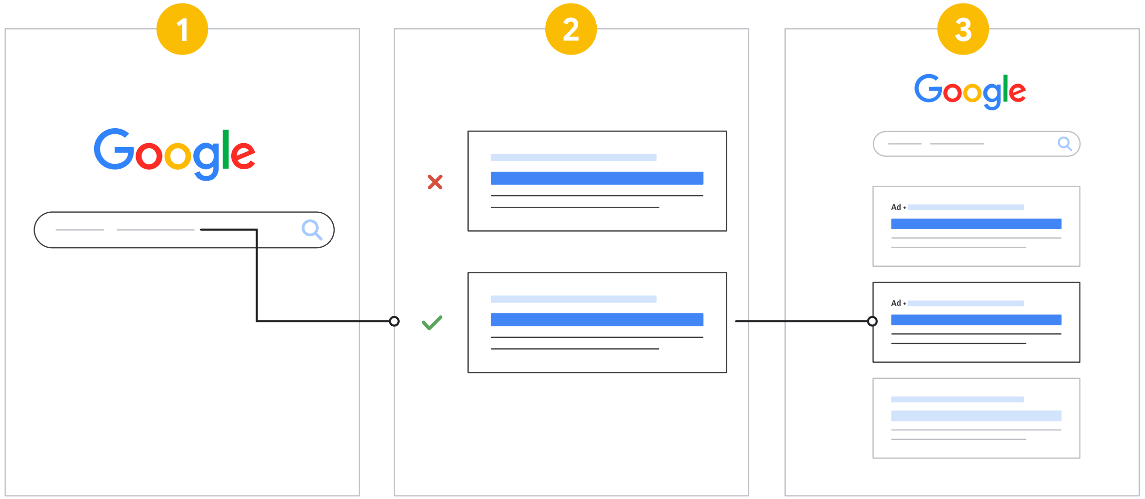
Example 1: Routing Games

- n drivers; each i wants to go from point a_i to point b_i on a road network
- Strategy space of player i: set of paths from $a_i \to b_i$
- When k_e users use road e it has latency $c_e(k_e)$
- Loss of a player: total latency on chosen path s_i

$$\ell_i(s) \coloneqq \sum_{e \in s_i} c_e\big(k_e(s)\big)$$

Image credits: chat.openai.com

Example 2: Sponsored Search Auctions



- n bidders; each bidder i has an ad to display under the search for a keyword
- Strategy space of bidder i: a bid $s_i \in R$
- Bidders allocated slots in decreasing order of bids; $j_i(s)$ is slot allocated to i
- Each slot j has a probability of click x_j
- When ad gets clicked, bidder pays bid s_i
- Utility of player is net expected gains

$$u_i(s) \coloneqq x_{j_i(s)} \cdot (v_i - s_i)$$

Image credits: https://support.google.com/google-ads/answer/142918?hl=en

Example 3: Recreational Games

- Simple two-player poker
- Each players strategy is an action plan on what to do at each possible decision point in the game
- Some decisions are also being taken by “nature” randomly and only partly announced to players
- Each leaf node is an end-result and contains a utility for P1
- Utility of P1 is expected value of the terminal node that will be reached
- Utility of P2 is negative of P1 (zero-sum)

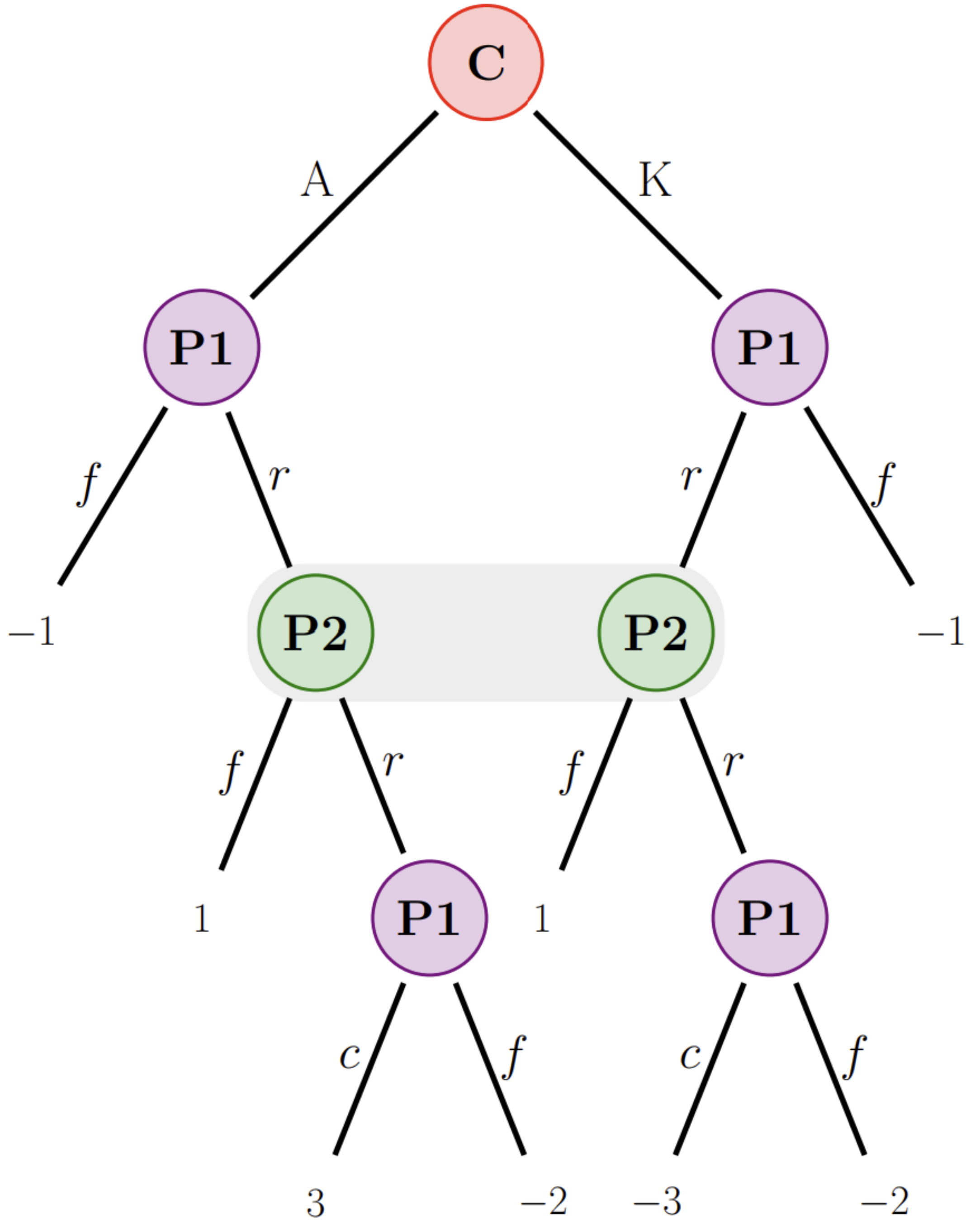
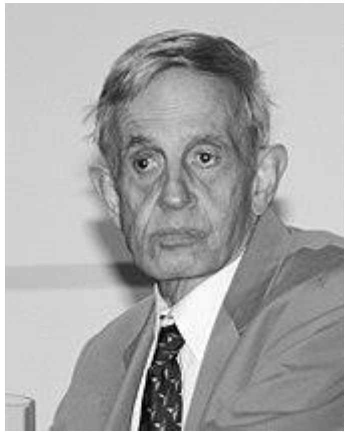


Image credits: https://www.columbia.edu/~ck2945/files/main_ai_games_markets.pdf

Pure Nash Equilibrium

- A strategy profile $s = (s_1, \dots, s_n)$ is a pure Nash equilibrium if no player is better off, by choosing some other strategy s_i'

$$\forall s_i' \in S_i \colon u_i(s_i, s_{-i}) \geq u_i(s_i', s_{-i})$$

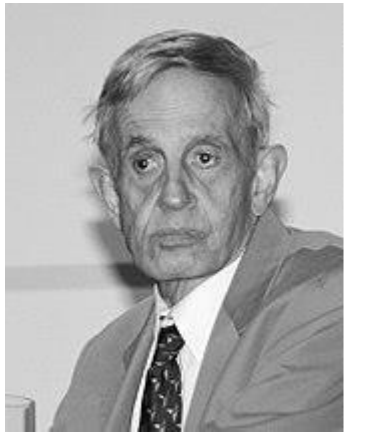


Mixed Nash Equilibrium

- A mixed strategy σ_i is a distribution over pure strategies
- At mixed strategy profile $\sigma = (\sigma_1, \ldots, \sigma_n)$, player i gets expected utility
$$U_i(\sigma) = E_{s_1 \sim \sigma_1, \ldots, s_n \sim \sigma_n}[u_i(s_1, \ldots, s_n)]$$
- Utility notation: $U_i(s_i', \sigma_{-i}) = E_{s_{-i} \sim \sigma_{-i}}[u_i(s_i', s_{-i})]$

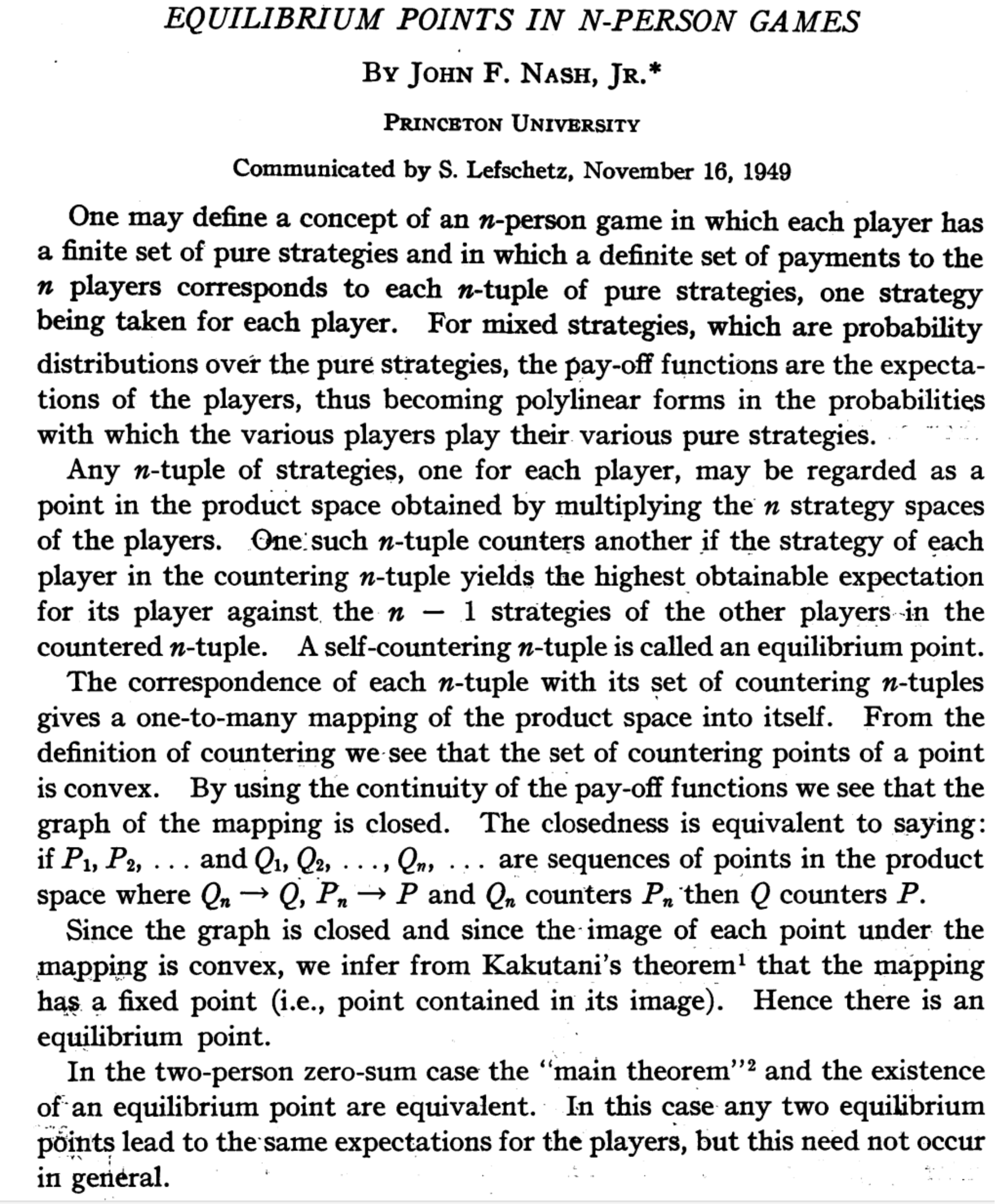
- A mixed strategy profile $\sigma = (\sigma_1, \ldots, \sigma_n)$ is a Nash equilibrium if no player is better off in expectation, by choosing another strategy s_i'

$$\forall s_i' \in S_i \colon U_i(\sigma) \geq U_i(s_i', \sigma_{-i})$$



Existence of Nash Equilibrium [Nash1950]

Every n player finite action game has at least one mixed Nash equilibrium



EQUILIBRIUM POINTS IN N-PERSON GAMES

By John F. Nash, Jr.*

Princeton University

Communicated by S. Lefschetz, November 16, 1949

One may define a concept of an n-person game in which each player has a finite set of pure strategies and in which a definite set of payments to the n players corresponds to each n-tuple of pure strategies, one strategy being taken for each player. For mixed strategies, which are probability distributions over the pure strategies, the pay-off functions are the expectations of the players, thus becoming polylinear forms in the probabilities with which the various players play their various pure strategies.

Any n-tuple of strategies, one for each player, may be regarded as a point in the product space obtained by multiplying the n strategy spaces of the players. One such n-tuple counters another if the strategy of each player in the countering n-tuple yields the highest obtainable expectation for its player against the $n - 1$ strategies of the other players in the countered n-tuple. A self-countering n-tuple is called an equilibrium point.

The correspondence of each n-tuple with its set of countering n-tuples gives a one-to-many mapping of the product space into itself. From the definition of countering we see that the set of countering points of a point is convex. By using the continuity of the pay-off functions we see that the graph of the mapping is closed. The closedness is equivalent to saying: if $P_1, P_2, \ldots$ and $Q_1, Q_2, \ldots, Q_n, \ldots$ are sequences of points in the product space where $Q_n \rightarrow Q$, $P_n \rightarrow P$ and Q_n counters P_n then Q counters P.

Since the graph is closed and since the image of each point under the mapping is convex, we infer from Kakutani's theorem[1] that the mapping has a fixed point (i.e., point contained in its image). Hence there is an equilibrium point.

In the two-person zero-sum case the "main theorem"[2] and the existence of an equilibrium point are equivalent. In this case any two equilibrium points lead to the same expectations for the players, but this need not occur in general.

Intractability of Mixed Nash Equilibrium

- The assumption of knowing the supports was crucial
- For games with many actions, we cannot enumerate all possible supports (combinatorial explosion)
- Turns out there is no easy way to side-step this

- Computing a mixed NE in two player games is “intractable”

- It is provable as hard as computing a “fixed point” ($f(x) = x$) of an arbitrary function f, which is considered an intractable problem

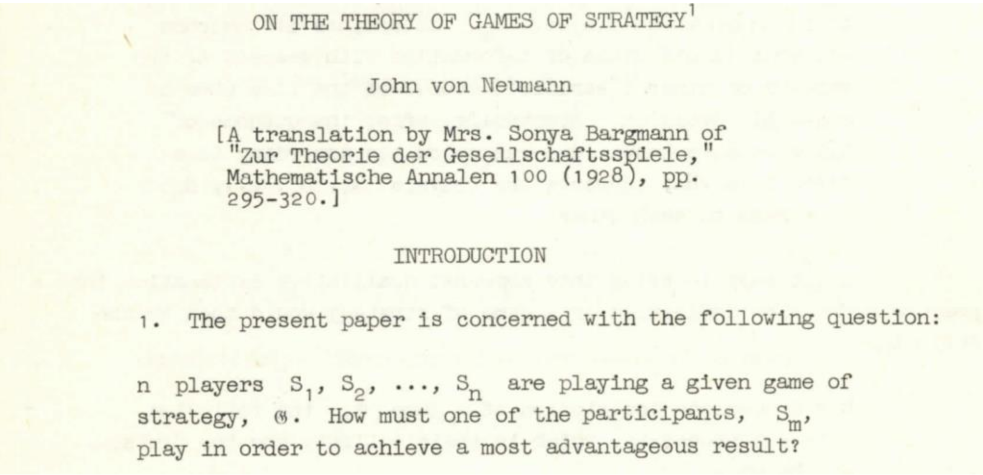
Two Player Zero-Sum Games

- Player one (“min” player or “row” player)
- Player two (“max” player or “column” player)
- Player one has n possible actions
- Player two has m possible actions

- If player one chooses action i and player two chooses action j then player one incurs loss $A[i, j]$ and player two gains utility $A[i, j]$

Von-Neuman’s Min-Max Theorem [1928]

$$\min_x \max_y x'Ay = \max_y \min_x x'Ay$$



ON THE THEORY OF GAMES OF STRATEGY[1]

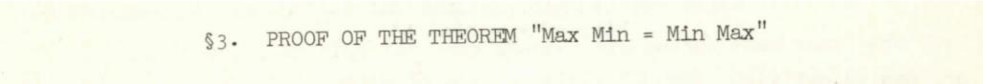
John von Neumann

[A translation by Mrs. Sonya Bargmann of
"Zur Theorie der Gesellschaftsspiele,"
Mathematische Annalen 100 (1928), pp.
295-320.]

INTRODUCTION

1. The present paper is concerned with the following question:

n players S_1, S_2, ..., S_n are playing a given game of strategy, $\mathfrak{G}$. How must one of the participants, S_m, play in order to achieve a most advantageous result?



§3. PROOF OF THE THEOREM "Max Min = Min Max"

Are there dynamics that will lead to a mixed Nash equilibrium?

Lecture 2

Example in Math

- Device a choice picking algorithm $\boldsymbol{i_t}$
- **Goal.** At end of the year, looking back, *not regret much* either "always taking Bay" or "always taking Dumbarton"

$$\text{Regret}(\ell_{1:T}) = \boxed{\frac{1}{T}\sum_{t=1}^{T} \ell_t^{i_t}} - \min_{i\in\{1,2\}} \boxed{\frac{1}{T}\sum_{t=1}^{T} \ell_t^{i}}$$

Short-hand notation for sequence of loss vectors $(\ell_1, \dots, \ell_T)$

Average # of jams you encountered

Average # of jams you would have encountered had you always chosen bridge i

Image credits: chat.openai.com

A choice picking algorithm is called a *no-regret learning algorithm* if the *worst-case regret* over any sequence of losses

$$R(T) = \sup_{\ell_{1:T}} \operatorname{Regret}(\ell_{1:T})$$

vanishes to zero with the number of periods

$$R(T) \to 0$$

The n action case

Short-hand for n-dimensional simplex $\Delta(n) := \{x \in R^n: x_i \geq 0, \sum_{i=1}^n x_i = 1\}$

At each period choose a distribution $p_t \in \boxed{\Delta(n)}$ over n actions

Observe a loss vector $\ell_t \in [0,1]^n$ and incur loss $\boxed{\langle p_t, \ell_t \rangle}$

Short-hand for inner product between two vectors

(FTRL) $$p_t = \min_p L_{t-1}(p) + \frac{1}{\eta}\mathcal{R}(p) = \min_p \sum_{\tau<t} \langle p, \ell_t \rangle + \frac{1}{\eta}\mathcal{R}(p)$$

For the *negative entropy* regularizer, leads to the simple EXP algorithm

$$p_t^i \propto \boxed{p_{t-1}^i \exp\left(-\eta \ell_{t-1}^i\right)}$$

Play each action with probability proportional to the exponential of its historical performance

The negative entropy is *1-strongly convex* and now takes values in $[-\log(\mathrm{n}), 0]$

$$R(T) \leq 2\eta + \frac{\log(n)}{\eta T} \boxed{\leq \sqrt{\frac{2\log(n)}{T}} \to 0}$$

For $\eta = \sqrt{\frac{\log(n)}{2T}}$

Punchline

(Linearized FTRL) $$p_t = \underset{p}{\operatorname{argmin}} \boxed{\bar{L}_{t-1}(p)} + \frac{1}{\eta}\boxed{\mathcal{R}(p)}$$

1-strongly convex function of p that stabilizes the minimizer

Linearized historical performance of always choosing vector p

Theorem. Assuming the linearized loss function at each period

$$\bar{\ell}_t(p) = \langle p, \nabla \ell_t(p_t) \rangle$$

is L-Lipschitz with respect to some norm $||\cdot||$ and the regularizer is 1-strongly convex with respect to the same norm then

$$\text{Regret} - \text{FTRL}(T) \leq \boxed{\eta L} + \boxed{\frac{1}{\eta T}\left(\max_p \mathcal{R}(p) - \min_p \mathcal{R}(p)\right)}$$

Average stability induced by regularizer

Average loss distortion caused by regularizer

Punchline: The Master Algorithms of our Times

(Linearized FTRL) $p_t = \underset{p}{\operatorname{argmin}}\, \bar{L}_{t-1}(p) + \frac{1}{\eta}\mathcal{R}(p)$

$$\mathcal{R}(p) = \sum_{i=1}^{n} p_i \log(p_i)$$

$$p_t \propto p_{t-1} \exp(-\eta\, \ell_{t-1})$$

Exponential weight updates algorithm!
(aka Hedge, Multiplicative Weight Updates, EXP, ….)

$$\mathcal{R}(p) = \frac{1}{2}\|p\|^2$$

$$p_t = p_{t-1} - \eta \nabla \ell_{t-1}(p_{t-1})$$

Online/Stochastic Gradient *Descent* Algorithm
(aka SGD)

Lecture 3

Main Takeaway: Equilibrium via No-Regret

Theorem. If two players play repeatedly a *convex-concave zero-sum game* and each player uses any no-regret algorithm to pick their vector (x_t, y_t), then the average vector of each player

$$\bar{x} = \frac{1}{T}\sum_{t=1}^{T} x_t\,, \qquad \bar{y} = \frac{1}{T}\sum_{t=1}^{T} y_t$$

are a 2ϵ-approximate Nash equilibrium (where ϵ is the regret at of each algorithm after T periods). Hence,

$$(\bar{x}, \bar{y}) \rightarrow \text{equilibrium as } T \rightarrow \infty$$

Minimax Theorem via No-Regret

Theorem. Existence of no-regret algorithms implies (as $\epsilon \to 0$) that

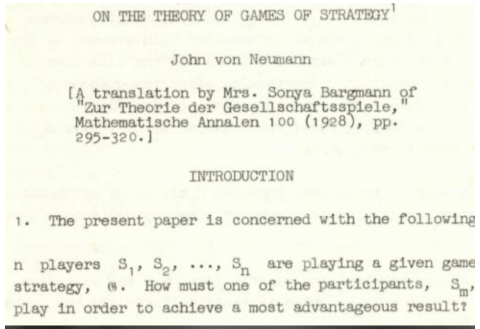
$$\max_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} \ell(x, y) \geq \min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \ell(x, y)$$

The other direction is trivial *(why?)*

$$\max_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} \ell(x, y) \leq \min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \ell(x, y)$$

Thus

$$\max_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} \ell(x, y) = \min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \ell(x, y)$$



ON THE THEORY OF GAMES OF STRATEGY[1]

John von Neumann

[A translation by Mrs. Sonya Bargmann of "Zur Theorie der Gesellschaftsspiele," Mathematische Annalen 100 (1928), pp. 295-320.]

INTRODUCTION

1. The present paper is concerned with the following

n players $S_1, S_2, \ldots, S_n$ are playing a given game strategy, $\mathfrak{G}$. How must one of the participants, S_m, play in order to achieve a most advantageous result?

(an alternative to von Neuman's original proof)

Can we do better in terms of rate?

Optimistic FTRL: Last Period Predictor

Optimism: predict that the next period loss will be the same as last period loss

$$\begin{pmatrix}\text{FTRL}\\ \text{w. Predictors}\end{pmatrix} \qquad p_t = \operatorname*{argmin}_p \boxed{\sum_{\tau<t}\langle p,\ell_\tau\rangle} + \boxed{\langle p,\ell_{t-1}\rangle} + \frac{1}{\eta}\boxed{\mathcal{R}(p)}$$

1-strongly convex function of p that stabilizes the minimizer

Historical performance of always choosing p

$$\mathcal{R}(p) = \sum_{i=1}^{n} p_i \log(p_i) \quad \begin{pmatrix}\text{Negative}\\ \text{Entropy}\end{pmatrix}$$

$$p_t \propto p_{t-1} \exp\!\big(\eta\,(2\ell_{t-1} - \ell_{t-2})\big)$$

Optimistic Exponential Weight Updates!

Optimistic EXP

Corollary. Optimistic EXP is 3η-stable and has regret

$$R(T) \leq \frac{\eta}{T} \boxed{\sum_{t=1}^{T} \|\ell_t - \ell_{t-1}\|_\infty} + \frac{\log(n)}{\eta\, T}$$

Average stability of the loss vector

Optimistic EXP Dynamics

Larger step size than if we were playing against an adversary
$T^{-1/3}$ vs. $T^{-1/2}$
(e.g. if $T = 1000$, then 0.1 vs. 0.032)

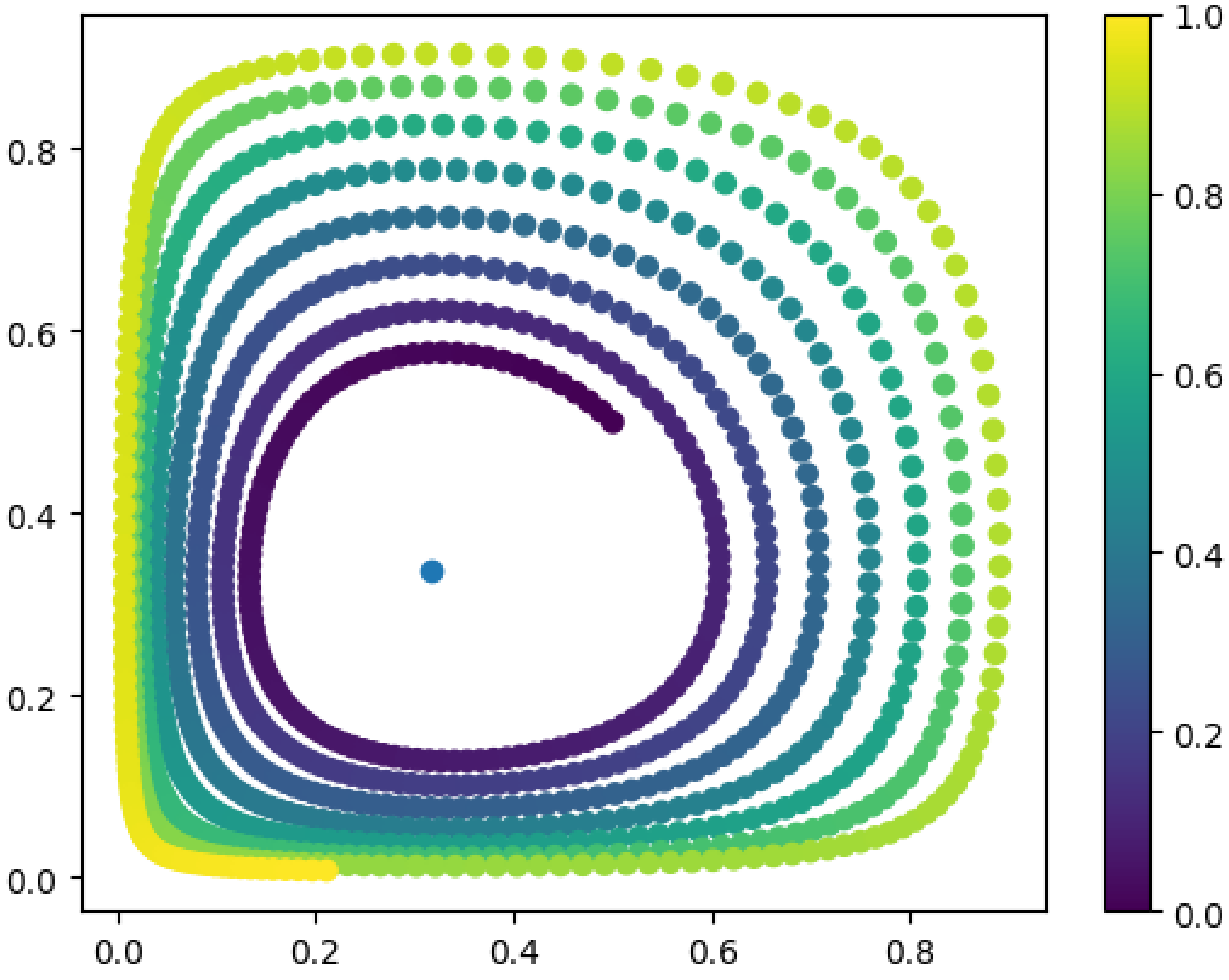
Corollary. If all players use Optimistic EXP with $\boxed{\eta = \left(\frac{\log(n \vee m)}{T}\right)^{1/3}}$ then each player's regret is at most $\boxed{\epsilon = 4\left(\frac{\log(n \vee m)}{T}\right)^{2/3}}$ and the average vectors $(\bar{x}, \bar{y})$ are an 2ϵ-approximate equilibrium

Order of magnitude smaller regret than playing against an adversary
$T^{-2/3}$ vs. $T^{-1/2}$
(e.g. if $T = 1000$, then 0.01 vs. 0.032)

Do the dynamics actually converge?

$(\bar{x}, \bar{y}) \to \text{equilibrium}$ vs. $(x_T, y_T) \to \text{equilibrium}$

"average iterate convergence" "last-iterate convergence"



A simple example

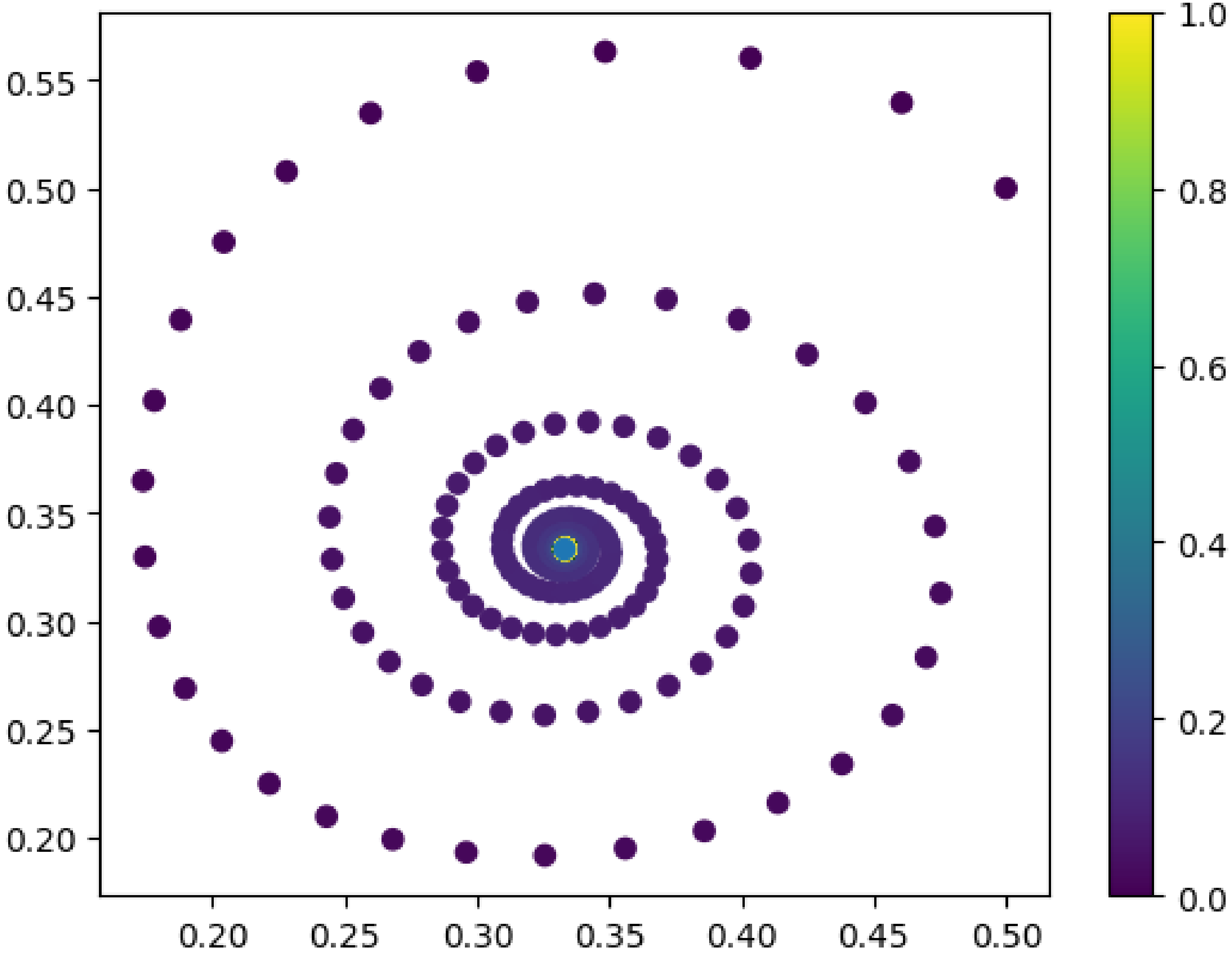
Consider the game defined by loss matrix

$$A = \begin{pmatrix} .5 & 0 \\ 0 & 1 \end{pmatrix}$$

EXP dynamics:

$$x_t \propto x_{t-1} \exp(-\eta A y_{t-1})$$

$$y_t \propto y_{t-1} \exp(\eta A^\top x_{t-1})$$



A simple example

Consider the game defined by loss matrix

$$A = \begin{pmatrix} .5 & 0 \\ 0 & 1 \end{pmatrix}$$

Optimistic EXP dynamics:

$$x_t \propto x_{t-1} \exp\big(-\eta(2Ay_{t-1} - Ay_{t-2})\big)$$

$$y_t \propto y_{t-1} \exp\Big(\eta\big(2A^\top x_{t-1} - A^\top x_{t-2}\big)\Big)$$

Lecture 4

Applications of Learning in Zero-Sum Games to ML and AI

Boosting, Distributional Robustness, Generative Learning, Learning from Human Feedback, Causal ML, Fair ML

The boosting problem

Given “weak” classification oracle, can we construct in a computationally efficient manner a “strong” classifier that achieves accuracy on D arbitrarily close to 1?

Major open problem among the ***tiny*** *ML community in late 80s-early 90s*

Resolved by Robert Schapire and further developed by Freund-Schapire

Punchline: Solving Large Games with Oracles

Theorem. Suppose we have Best-Response oracle over J for the max player for each distribution w over actions of the min player. Repeat for T iterations the process:

(EXP) $$w_t \propto w_{t-1} \exp\left(-\eta \ell_{j_{t-1}}\right)$$

(Best-Response) $$j_t = \mathrm{BR}(w_t)$$

Then $w_* = \frac{1}{T}\sum_{t=1}^{T} w_t$ and $P_* = \mathrm{Uniform}(\{j_1, \dots, j_T\})$ is a $\sqrt{\frac{2\log(n)}{T}}$-approximate equilibrium ⇒ P_* is $2\sqrt{\frac{2\log(n)}{T}}$-solution to max-min.

Punchline: AdaBoost Theorem

Theorem. Suppose we have a weak δ-classification oracle WEAK. For every hypothesis h, let ℓ_h be vector of 0-1 accuracies on each sample.

Repeat for T periods, such that $\sqrt{\frac{2\log(n)}{T}} < \delta$

(EXP) $$w_t \propto w_{t-1} \exp\left(-\eta \ell_{h_{t-1}}\right)$$

(Weak-oracle) $$h_t = \mathrm{WEAK}(w_t)$$

Then the following majority classifier classifies all samples correctly

$$h_* = \mathrm{Majority}(h_1, \dots, h_T) = 1\left\{\frac{1}{T}\sum_{t=1}^{T} h_t(\cdot) > \frac{1}{2}\right\}$$

Distributional Robustness

Image credits: chat.openai.com

Group Distributional Robustness; Group-DRO

[1611.02041] Does Distributionally Robust Supervised Learning Give Robust Classifiers? (arxiv.org)

[1909.02060] Distributionally Robust Language Modeling (arxiv.org)

- We pre-define a set of groups G (race, gender, sensitive attributes)
- At train time, we know the group identity of each sample
- We want to learn a single model θ (that does not use the group attribute as input) that performs well on distribution of each group

$$\min_{\theta \in \Theta} \max_{g \in G} E_{(x,y) \sim D_g} [\ell(y, h_\theta(x))]$$

Group DRO as a Zero-Sum Game

[1911.08731] Distributionally Robust Neural Networks for Group Shifts: On the Importance of Regularization for Worst-Case Generalization

- The learner player chooses $\theta \in \Theta$
- The adversary player chooses a distribution w_t over G
- If loss is convex in θ and Θ is convex set, solve via no-regret

(OGD) $$\theta_t = \theta_{t-1} - \eta \sum_{\mathrm{g}} w_{t-1}^{g}\, E_{(x,y)\sim D_g}\left[\nabla_\theta \ell\left(y, h_{\theta_{t-1}}(x)\right)\right]$$

(EXP) $$w_t^g \propto w_{t-1}^g \exp\left(E_{(x,y)\sim D_g}\left[\ell\left(y, h_{\theta_{t-1}}(x)\right)\right]\right)$$

- Even when loss is not convex in θ, the above translates to a practical training algorithm for neural network parameters
- Expectations are typically approximated by averages over small batches of samples

Note: typically, last iterate and not average iterate is used despite theory...

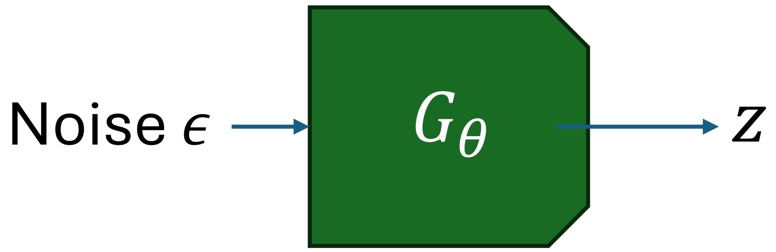
Image credits: chat.openai.com

Generative Adversarial Networks

GANs as a Zero-Sum Game

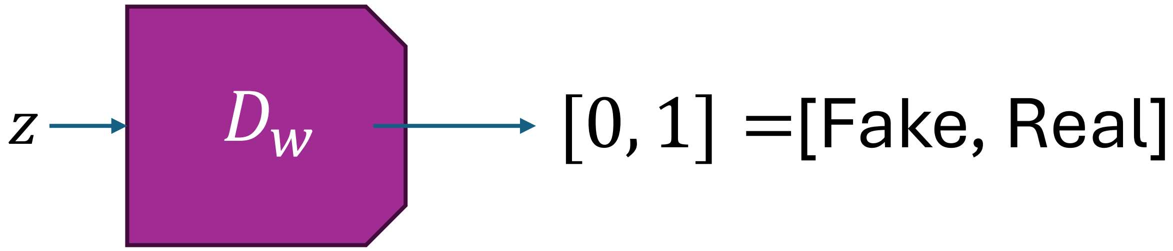
[1701.00160] NIPS 2016 Tutorial: Generative Adversarial Networks (arxiv.org)

- Learn a neural sample generator (max player)



Noise ϵ → G_θ → z

- Learn a discriminator (min player)



z → D_w → $[0, 1]$ =[Fake, Real]

- Discriminator minimizes classification error/Generator maximizes

$$\max_{\theta} \min_{w} -E_{z\sim D}\left[\log\left(D_w(z)\right)\right] + E_{\epsilon}\left[\log\left(D_w\left(G_\theta(\epsilon)\right)\right)\right]$$

$D_w(z)$ close to 1 when z is real

$D_w(z)$ close to 0 when fake

GANs as a Zero-Sum Game

[1701.00160] NIPS 2016 Tutorial: Generative Adversarial Networks (arxiv.org)

- We are trying to find a generator that fools the discriminator
- Solve max-min problem by finding equilibrium of zero-sum game

$$\max_{\theta} \min_{w} \ell(\theta, w) \coloneqq -E_{z\sim D}\left[\log\left(D_w(z)\right)\right] + E_{\epsilon}\left[\log\left(D_w\left(G_\theta(\epsilon)\right)\right)\right]$$

- Compute via no-regret dynamics (online gradient descent/ascent)

(OGD) $$\theta_t = \theta_{t-1} + \eta \nabla_\theta \ell(\theta_{t-1}, w_{t-1})$$

(OGD) $$w_t = w_{t-1} - \eta \nabla_w \ell(\theta_{t-1}, w_{t-1})$$

- Even though non-convex/non-concave!
- Last-iterate used, though theory says average (*optimism can help*)

[1711.00141] Training GANs with Optimism (arxiv.org)

Learning from Human Feedback

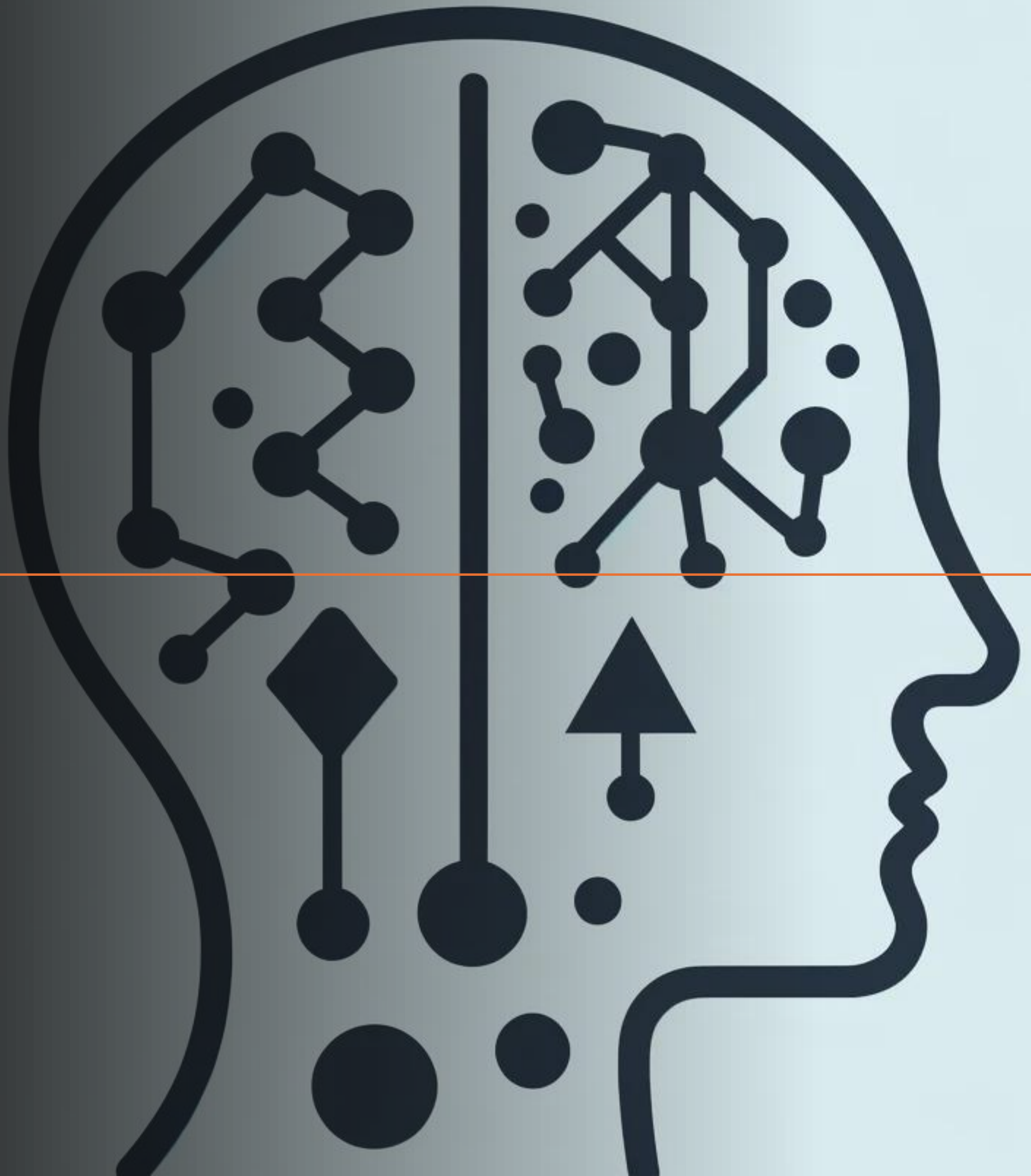


Image credits: chat.openai.com

Learning from Human Feedback

We have space of policies Π that given context x produce $y = \pi(x)$

AI Alignment Goal. Want to find a policy that produces output y that is typically more "aligned" with people's preferences

Human Feedback. We elicit pair-wise preferences over outputs

- We show people pairs of outputs $y_1 = \pi_1(x)$ and $y_2 = \pi_2(x)$
- We collect preference feedback, $1\{y_1 > y_2\} - 1\{y_2 < y_1\}$
- Our cumulative data provide a (*anti-symmetric*) preference function P

$$P(\pi, \pi') \in [-1,1], \qquad P(\pi, \pi') = -P(\pi', \pi)$$

  i.e. fraction of people with $\pi > \pi'$ minus fraction of people with $\pi' > \pi$

Social Choice Theory: Minimax Winner

[2401.04056] A Minimaximalist Approach to Reinforcement Learning from Human Feedback (arxiv.org)

- Choose a distribution p over options such that you prefer samples from that distribution than samples from any other distribution with probability at least ½

$$\min_{p'} E_{\pi\sim p,\pi'\sim p'}[P(\pi,\pi')] \geq 0$$

Lemma. The MW is the symmetric mixed Nash equilibrium of the zero-sum game defined by the preference matrix

	a	b	c	d
a	0	$+1$	$+1$	-1
b	-1	0	$+1$	-1
c	-1	-1	0	$+1$
d	$+1$	$+1$	-1	0

Computational Game Theory for Complex Games

1

- Basics of game theory and zero-sum games (T)
- Basics of online learning theory (T)
- Solving zero-sum games via online learning (T)
- *HW1: implement simple algorithms to solve zero-sum games*
- Applications to ML and AI (T+A)
- *HW2: implement boosting as solving a zero-sum game*

2

- Basics of extensive-form games
- Solving extensive-form games via online learning (T)
- *HW3: implement agents to solve very simple variants of poker*

3

- General games, equilibria and online learning (T)
- Online learning in general games (T+A)
- *HW4: implement no-regret algorithms that converge to correlated equilibria in general games*

Data Science for Auctions and Mechanisms

4

- Basics and applications of auction theory (T+A)
- Basic Auctions and Learning to bid in auctions (T)
- *HW5: implement bandit algorithms to bid in ad auctions*

5

- Optimal auctions and mechanisms (T)
- Simple vs optimal mechanisms (T)
- *HW6: implement simple and optimal auctions, analyze revenue empirically*

6

- Basics of Statistical Learning Theory (T)
- Optimizing Mechanisms from Samples (T)
- *HW7: implement procedures to learn approximately optimal auctions from historical samples*

Further Topics

7

- Econometrics in games and auctions (T+A)
- A/B testing in markets (T+A)
- *HW8: implement procedure to estimate values from bids in an auction, empirically analyze inaccuracy of A/B tests in markets*

Guest Lectures

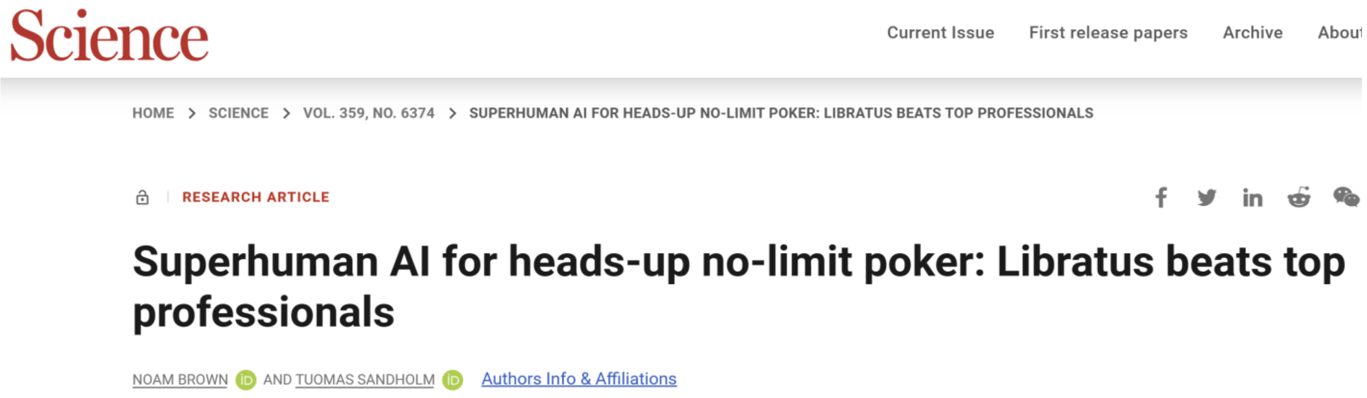
- Mechanism Design for LLMs, Renato Paes Leme, Google Research
- Auto-bidding in Sponsored Search Auctions, Kshipra Bhawalkar, Google Research

Lecture 5

Extensive Form Games

History and Progress

Many Recent Success Stories

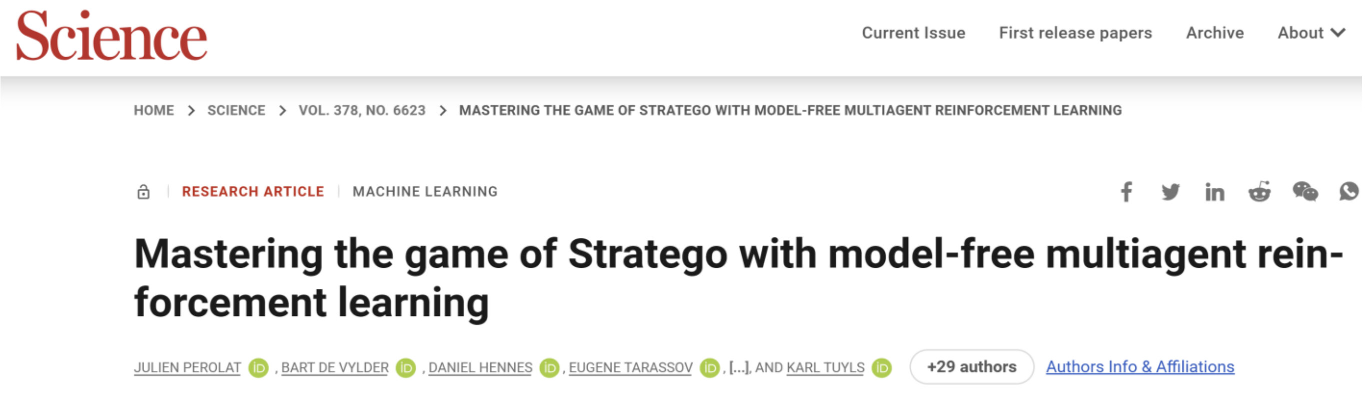


HOME > SCIENCE > VOL. 359, NO. 6374 > SUPERHUMAN AI FOR HEADS-UP NO-LIMIT POKER: LIBRATUS BEATS TOP PROFESSIONALS

RESEARCH ARTICLE

Superhuman AI for heads-up no-limit poker: Libratus beats top professionals

NOAM BROWN AND TUOMAS SANDHOLM Authors Info & Affiliations

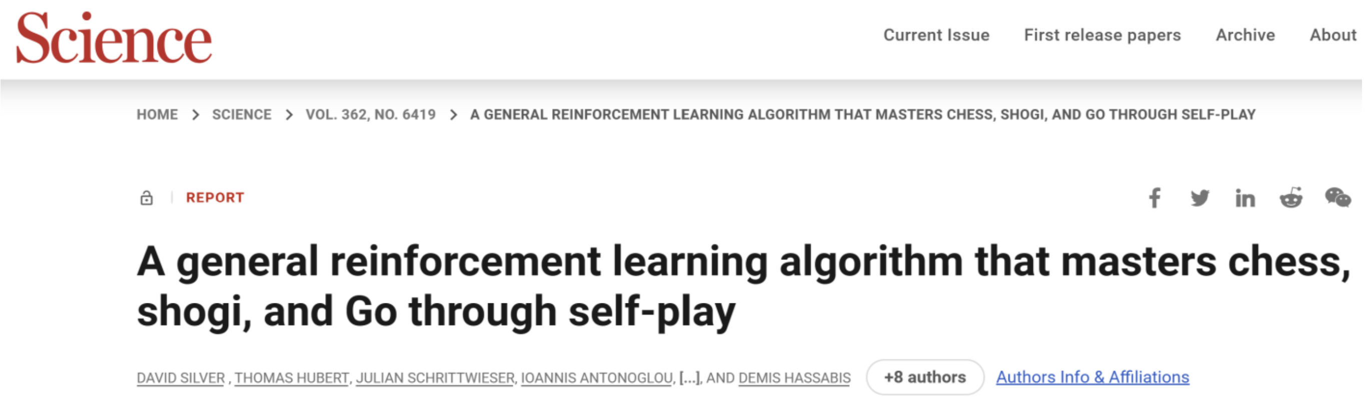


HOME > SCIENCE > VOL. 378, NO. 6623 > MASTERING THE GAME OF STRATEGO WITH MODEL-FREE MULTIAGENT REINFORCEMENT LEARNING

RESEARCH ARTICLE | MACHINE LEARNING

Mastering the game of Stratego with model-free multiagent reinforcement learning

JULIEN PEROLAT, BART DE VYLDER, DANIEL HENNES, EUGENE TARASSOV, [...], AND KARL TUYLS +29 authors Authors Info & Affiliations



HOME > SCIENCE > VOL. 362, NO. 6419 > A GENERAL REINFORCEMENT LEARNING ALGORITHM THAT MASTERS CHESS, SHOGI, AND GO THROUGH SELF-PLAY

REPORT

A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play

DAVID SILVER, THOMAS HUBERT, JULIAN SCHRITTWIESER, IOANNIS ANTONOGLOU, [...], AND DEMIS HASSABIS +8 authors Authors Info & Affiliations

Game Representations Convenient for Computing a Nash Equilibrium

Recap: Sequence Form Representation

- The strategies of the player can be represented as $\tilde{x} \in X, \tilde{y} \in Y$
- $\tilde{x}_{\mathrm{a}}$: product of probabilities of all actions of P1 on the path to *a*
- $\tilde{y}_{\mathrm{a}}$: product of probabilities of all actions of P2 on the path to *a*

$$X := \left\{ \forall j \in \mathcal{J}_1 : \sum_{a \in A_j} \tilde{x}_a = \tilde{x}_{p_j} \right\}, \qquad Y := \left\{ \forall j \in \mathcal{J}_2 : \sum_{a \in A_j} \tilde{y}_a = \tilde{y}_{p_j} \right\}$$

- The payoff to P1 under sequence strategies $\tilde{x} \in X, \tilde{y} \in Y$ is

$$\tilde{x}^\top A \tilde{y}$$

- $A_{a,a'} =$ **if** a was the last action of P1 and a' the last action of P2 before some leaf z, **then** payoff to P1 at z times product of chance probabilities on path to z **else** zero

No-Regret Learning in Sequence Form

- We have successfully turned imperfect information extensive form zero-sum games into a familiar object

$$\max_{\tilde{x} \in X} \min_{\tilde{y} \in Y} \tilde{x}^\top A \tilde{y}$$

- X, Y are convex sets, i.e., sequence-form strategies

- We can invoke minimax theorem to prove existence of equilibria
- We can calculate equilibria via LP duality
- We can calculate equilibria via no-regret learning!

Lecture 6

Sum: Nash via FTRL with Dilated Entropy

Each player chooses $\tilde{x}_t, \tilde{y}_t$ based on FTRL with dilated entropy

- For x-player $u_t = A\tilde{y}_t$ and $U_t = U_{t-1} + u_t$ and initialize $Q = U_t$
- Traverse the tree bottom-up; for each infoset $j \in \mathcal{J}_1$
  $x_{t+1}^j \propto \exp\left(\eta_j Q^j\right), \qquad V^j = \mathrm{softmax}_{\eta_j}\left(Q^j\right), \qquad Q_{p_j} \leftarrow Q_{p_j} + V^j$
- Define sequence-form strategies top-down: $\tilde{x}_{t+1}^j = \tilde{x}_{p_j} \cdot x_{t+1}^j$

Similarly, for y player

Return average of sequence-form strategies as equilibrium

Interpreting utility vector

$$u_{t,a} = A\tilde{y}_t = \sum_{a' \in A_{P2}} A_{a,a'} \tilde{y}_{t,a'}$$

$A_{a,a'}$ is zero if the combination of a, a' does not lead to a leaf node

$$u_{t,a} = \sum_{\substack{\text{Leafs } z:\ a \text{ was last P1 action} \\ a' \text{ was last P2 action}}} u(z) \Pr\left(\begin{array}{c}\text{Chance chooses} \\ \text{sequence on} \\ \text{path to } z\end{array}\right) \Pr\left(\begin{array}{c}\text{P2 plays} \\ \text{sequence} \\ \text{leading to } a'\end{array}\right)$$

Interpretation. If I play with the intend to arrive at action a (i.e. $\tilde{x}_a = 1$) *and then don't make any other moves*, what is the expected reward that I will collect, in expectation over the choices of my opponent and nature

Counterfactual Regret Minimization (CRM)

Local Node Utilities

Interpretation of u_a. If I play with the intend to arrive at action a (i.e. $\tilde{x}_a = 1$) *and then don’t make any other moves*, what is the expected reward that I will collect, in expectation over the choices of my opponent and nature

What if we now want to express: If I play with the intend to arrive at action a (i.e. $\tilde{x}_a = 1$) *and then continue playing based on some behavioral policy x*, what is the expected reward that I will collect, in expectation over the choices of my opponent and nature

- Let C_a be all infosets of the player that are reachable **as next infosets** after playing a

$$\tilde{u}_a(x) = \boxed{u_a} + \sum_{k \in C_a} \boxed{V^k(x)}$$

“Instantaneous E[utility]”, if this is the last action I play

Continuation E[utility] from paths that pass through infoset k, if I continue playing based on behavioral strategy x

- Continuation utility $V^j(x)$ from paths that pass through infoset j recursively defined:

$$V^j(x) = \sum_{a \in A^j} x_a \, \tilde{u}_a(x) = \boxed{\sum_{a \in A^j} x_a u_a} + \boxed{\sum_{a \in A^j} x_a \left(\sum_{k \in C_a} V^k(x) \right)}$$

“Instantaneous utility”, if this is the last move I make

“Continuation utility”, if I continue playing based on x

Regret over Time

Same inequalities can be followed for the average regret over time

$$R = \max_{\tilde{x}' \in X} \frac{1}{T} \sum_{t} \langle \tilde{x}', u_t \rangle - \langle \tilde{x}_t, u_t \rangle$$

$$LR^j = \max_{x^j} \frac{1}{T} \sum_{t} \left\langle x^j, \tilde{u}_t(x_t) \right\rangle - \left\langle x_t^j, \tilde{u}_t(x_t) \right\rangle$$

Main CFR Theorem. Regret is upper bounded by local regrets

$$R \leq \sum_{j \in \mathcal{L}_1} LR^j$$

Counterfactual Regret Minimization

- Device local regret algorithms for local regret

$$\mathrm{LR}^j(x) = \max_{x^j} \frac{1}{T} \sum_t \left\langle x^j, \tilde{u}_t(x_t) \right\rangle - \left\langle x_t^j, \tilde{u}_t(x_t) \right\rangle$$

- Standard n-action no-regret problem: reward vector at period t is $\tilde{u}^j(x_t)$ and reward for choice x^j is $\left\langle x^j, \tilde{u}^j(x_t) \right\rangle$

- At period t run bottom-up recursion to calculate $\tilde{u}^j(x_t)$ for $j \in \mathcal{J}_1$

- Update probabilities x_{t+1}^j using reward vectors $\tilde{u}^j(x_t)$ for $j \in \mathcal{J}_1$

The Typical CRM Algorithm Implementation

```
CValue(ActionHistory $h$, AccOtherProb $\pi_{-1}$, AccProb $\pi_1$)
    Let $I$ be infoset corresponding to $h$
    If $I$ is terminal node $z$ return $u(z)$
    If $\text{Player}(I) = \text{chance}$
        Return $\sum_{a \in A_I} \pi_a^C \cdot \text{CValue}(ha, \pi_{-1}\pi_a^C, \pi_1)$
    If $\text{Player}(I) = 2$
        Return $\sum_{a \in A_I} y_a \cdot \text{CValue}(ha, \pi_{-1} y_a, \pi_1)$
    If $\text{Player}(I) = 1$
        For $a \in A_I$: $\tilde{u}_a \mathrel{+}= \pi_{-1} \cdot \text{CValue}(ha, \pi_{-1}, \pi_1 x_a)$
        Set $q(I) = \pi_1$
        Return $\sum_{a \in A_I} x_a \cdot \text{CValue}(ha, \pi_{-1}, \pi_1 x_a)$


CValue($\emptyset$, 1)
```

The Overall Equilibrium Algorithm with CRM

After each period $t \in \{1, \dots, T\}$:

- With last period behavior strategies x_t, y_t call $\mathrm{CValue}(\emptyset, 1, 1)$
- Store $\tilde{u}_{t,a}$ and $q_t(I)$ for each action a and infoset I of P1
- Symmetrically, do so for player P2
- Update strategies at all information sets

$$\forall j \in \mathcal{J}_1\colon x_{t+1}^j \leftarrow \mathrm{Update}\left(\tilde{u}_t^j\right), \qquad \forall j \in \mathcal{J}_2\colon y_{t+1}^j \leftarrow \mathrm{Update}\left(\tilde{u}_t^j\right)$$

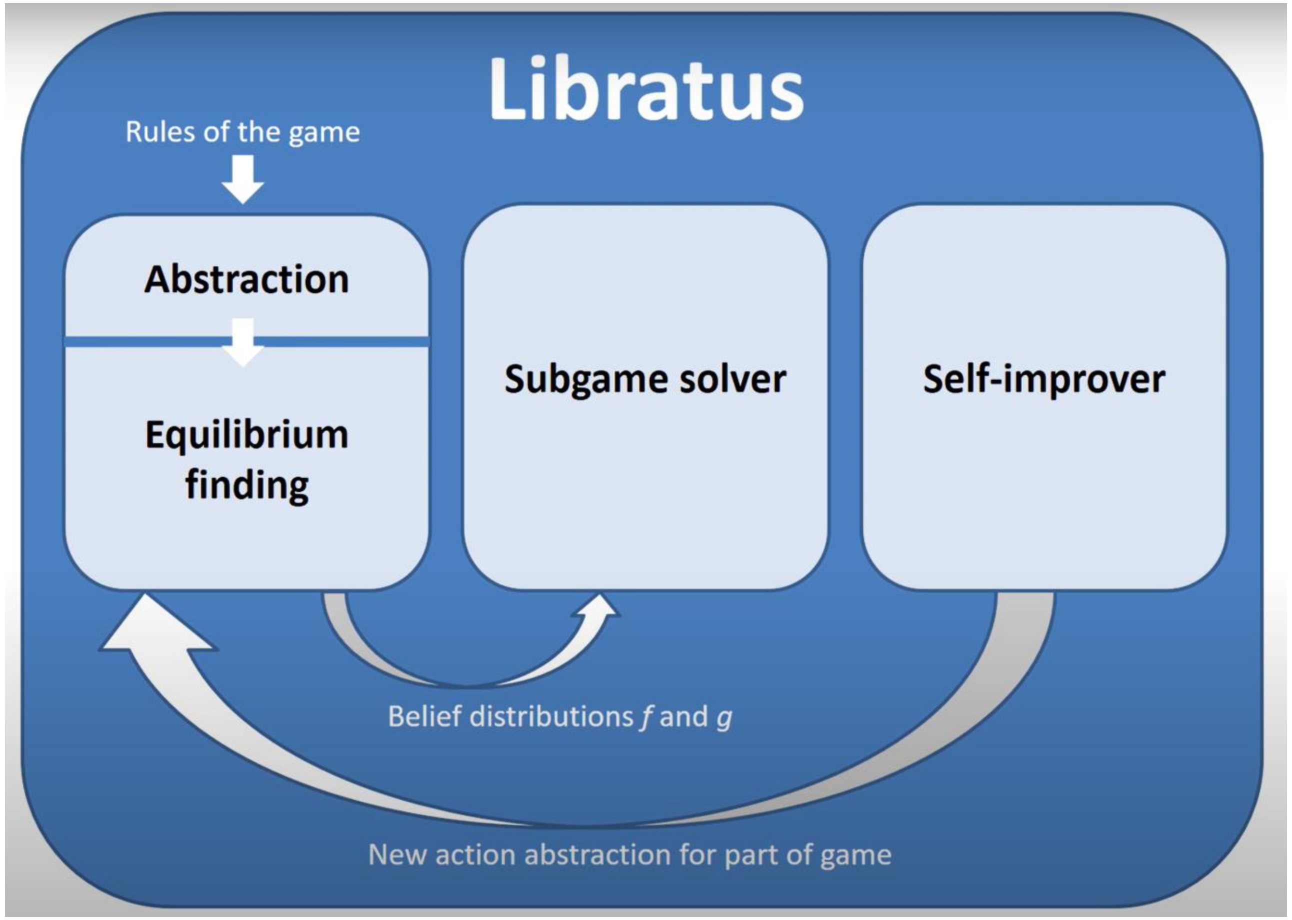
At the end:

$$\forall I \in \mathcal{J}_1 \forall a \in A_I\colon x_a^* = \frac{\sum_t q_t(I) x_{t,a}}{\sum_t q_t(I)}$$

$$\forall I \in \mathcal{J}_2 \forall a \in A_I\colon y_a^* = \frac{\sum_t q_t(I) y_{t,a}}{\sum_t q_t(I)}$$

Approximate Equilibrium in Behavioral Strategies

Elements of Libratus AI



Credits: Superhuman AI for heads-up no-limit poker: Libratus beats top professionals (youtube.com)

Computational Game Theory for Complex Games

1

- Basics of game theory and zero-sum games (T)
- Basics of online learning theory (T)
- Solving zero-sum games via online learning (T)
- *HW1: implement simple algorithms to solve zero-sum games*
- Applications to ML and AI (T+A)
- *HW2: implement boosting as solving a zero-sum game*

2

- Basics of extensive-form games
- Solving extensive-form games via online learning (T)
- *HW3: implement agents to solve very simple variants of poker*

3

- General games, equilibria and online learning (T)
- Online learning in general games (T+A)
- *HW4: implement no-regret algorithms that converge to correlated equilibria in general games*

Data Science for Auctions and Mechanisms

4

- Basics and applications of auction theory (T+A)
- Basic Auctions and Learning to bid in auctions (T)
- *HW5: implement bandit algorithms to bid in ad auctions*

5

- Optimal auctions and mechanisms (T)
- Simple vs optimal mechanisms (T)
- *HW6: implement simple and optimal auctions, analyze revenue empirically*

6

- Basics of Statistical Learning Theory (T)
- Optimizing Mechanisms from Samples (T)
- *HW7: implement procedures to learn approximately optimal auctions from historical samples*

Further Topics

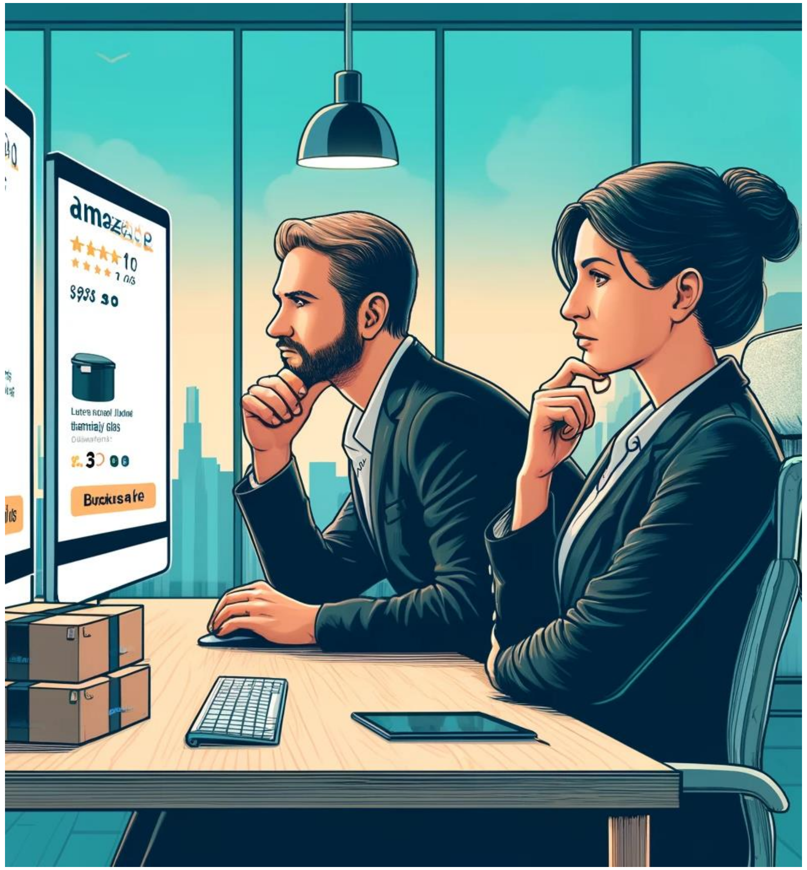
7

- Econometrics in games and auctions (T+A)
- A/B testing in markets (T+A)
- *HW8: implement procedure to estimate values from bids in an auction, empirically analyze inaccuracy of A/B tests in markets*

Guest Lectures

- Mechanism Design for LLMs, Renato Paes Leme, Google Research
- Auto-bidding in Sponsored Search Auctions, Kshipra Bhawalkar, Google Research

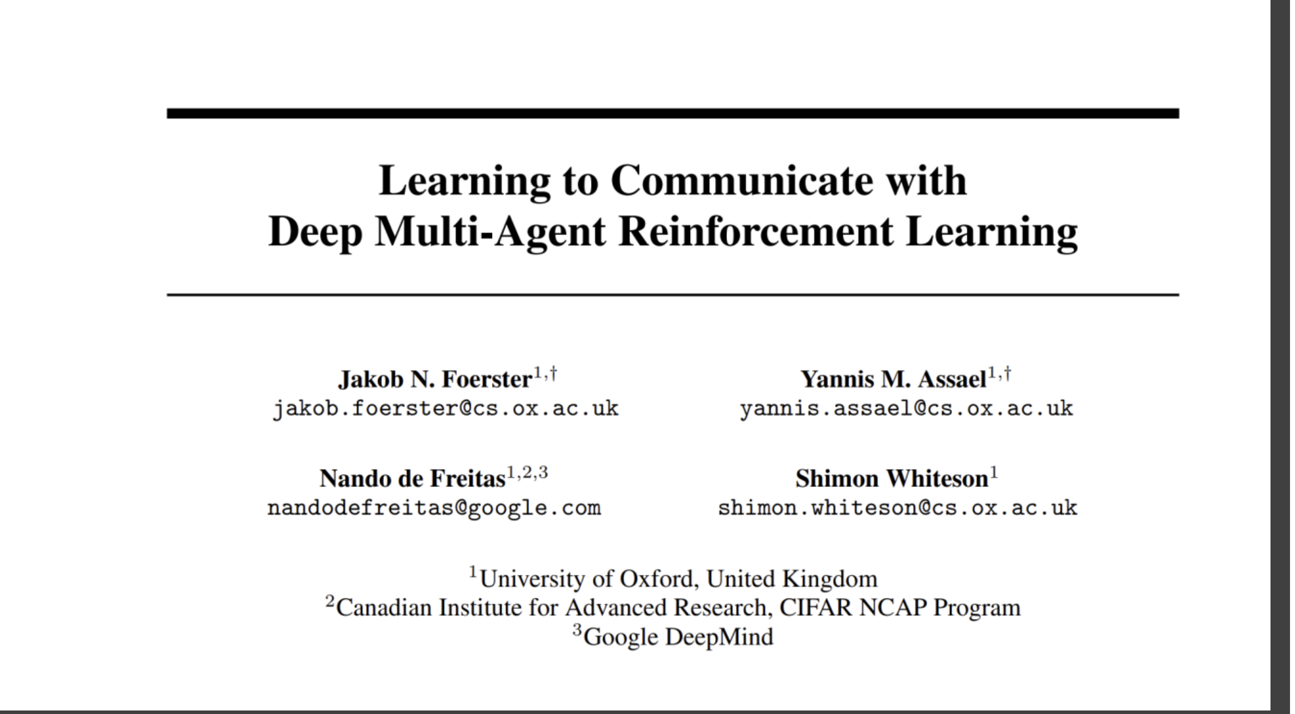
Lecture 7



Many real-world games are not zero-sum

Image credits: chat.openai.com

Recent Successes



Learning to Communicate with Deep Multi-Agent Reinforcement Learning

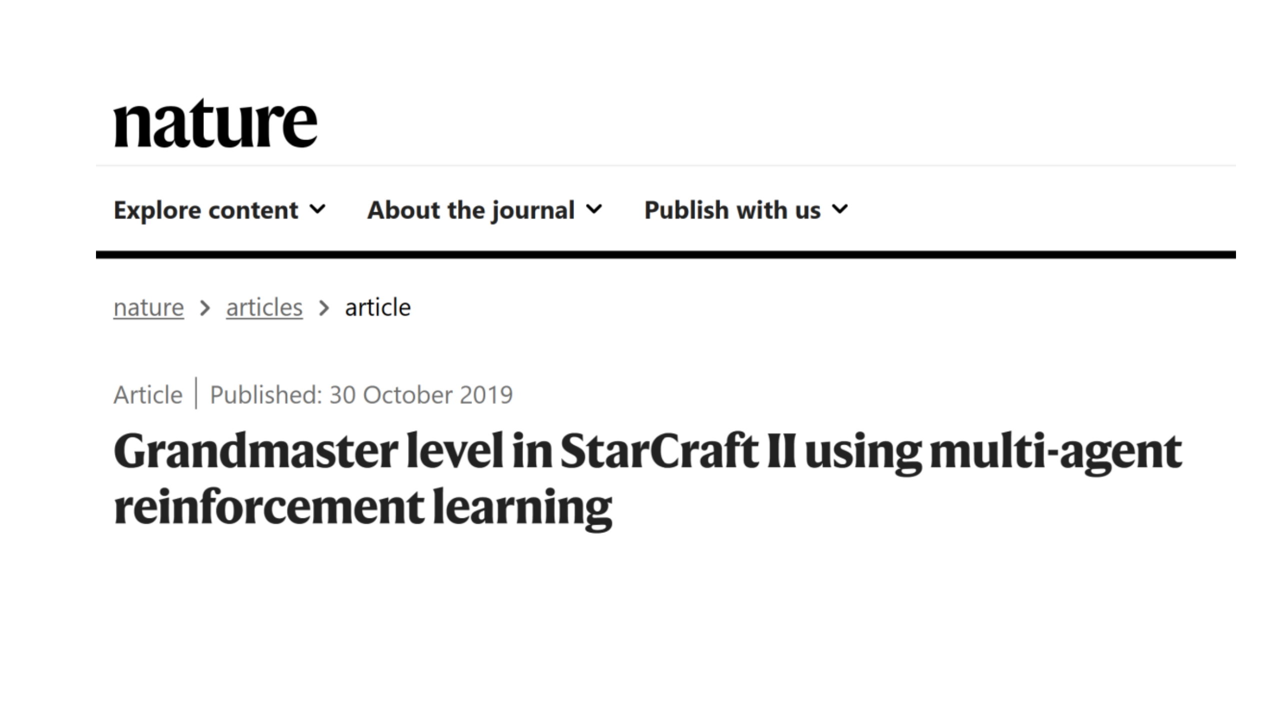
Jakob N. Foerster[1,†]
jakob.foerster@cs.ox.ac.uk

Yannis M. Assael[1,†]
yannis.assael@cs.ox.ac.uk

Nando de Freitas[1,2,3]
nandodefreitas@google.com

Shimon Whiteson[1]
shimon.whiteson@cs.ox.ac.uk

[1]University of Oxford, United Kingdom
[2]Canadian Institute for Advanced Research, CIFAR NCAP Program
[3]Google DeepMind



nature

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature > articles > article

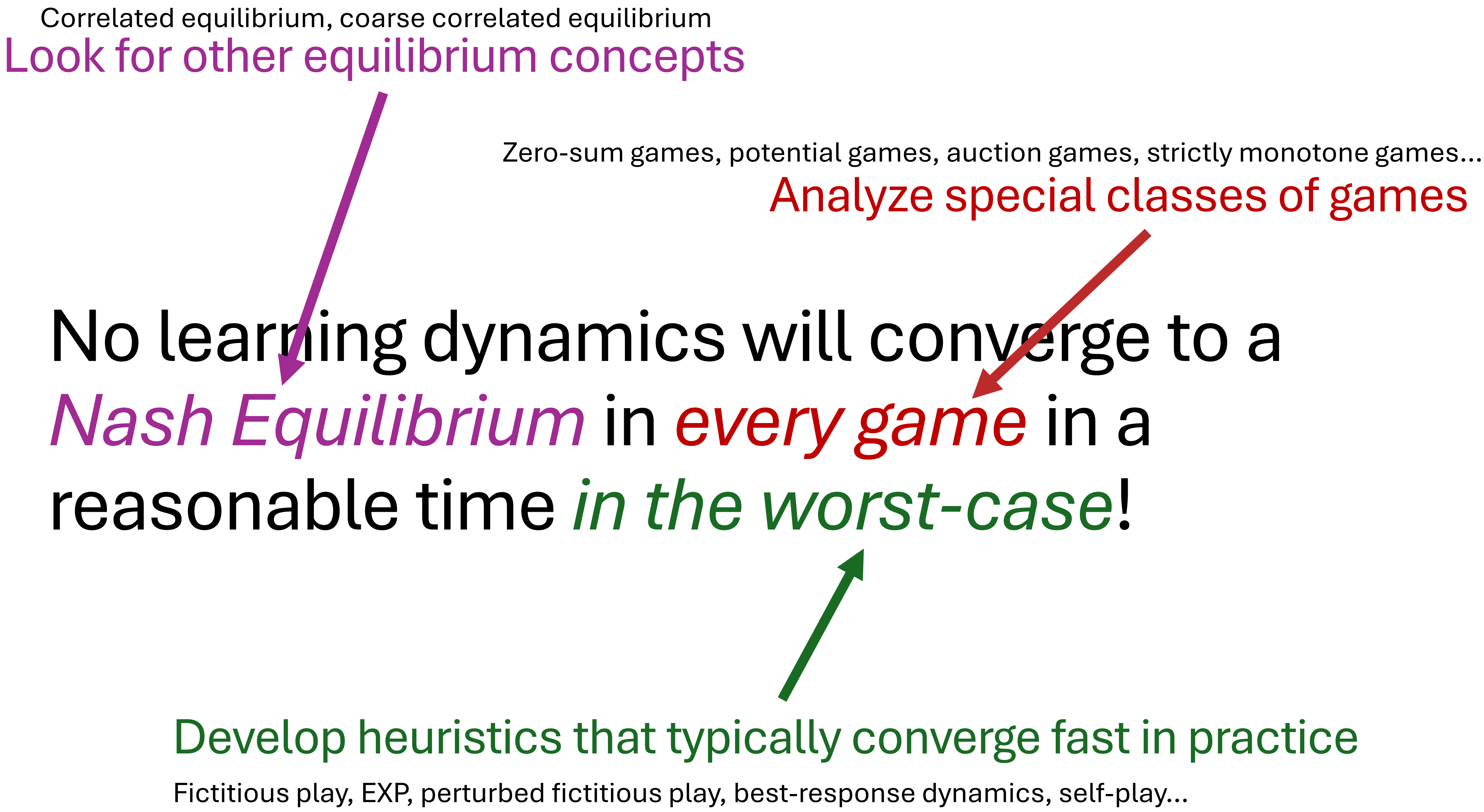
Article | Published: 30 October 2019

Grandmaster level in StarCraft II using multi-agent reinforcement learning

OpenAI Five

Our team of five neural networks, OpenAI Five, has started to defeat amateur human teams at Dota 2.

Much harder to compute equilibria;
theory typically considers relaxed solution concepts that are computationally easy
practice typically uses similar algorithms as in zero-sum games as good heuristics



Correlated equilibrium, coarse correlated equilibrium

Look for other equilibrium concepts

Zero-sum games, potential games, auction games, strictly monotone games…

Analyze special classes of games

No learning dynamics will converge to a *Nash Equilibrium* in *every game* in a reasonable time *in the worst-case*!

Develop heuristics that typically converge fast in practice

Fictitious play, EXP, perturbed fictitious play, best-response dynamics, self-play…

In Search for Other Equilibrium Concepts

Correlated Equilibrium

- A trusted third party draws strategy profiles $s = (s_1, \dots, s_n)$ of the game from some distribution D
- Communicates to each participant their part of the profile, i.e., the recommended strategy s_i

- The distribution D is a correlated equilibrium if participants don’t have incentive to deviate from their recommended strategy

$$\underline{\forall s_i, s_i' \in S_i:} \quad \underline{E_{s\sim D}[\, u(s) \mid s_i \,]} \quad \geq \quad \underline{E_{s\sim D}\left[\, u\left(s_i', s_{-i}\right) \mid s_i \,\right]}$$

For any recommendation s_i and possible deviation s_i'

Expected utility of choosing s_i when recommended s_i

$\geq$

Expected utility of deviating to s_i' when recommended s_i

Correlated Equilibria are Tractable

- Define a variable $\pi(s)$ for every strategy profile $s \in S_1 \times \cdots \times S_n$
- The variables encode a distribution

$$\sum_{s} \pi(s) = 1$$

- The distribution π is a correlated equilibrium if participants don’t have incentive to deviate from their recommended strategy

$$\forall s_i, s_i' \in S_i: \quad \sum_{s_{-i}} \pi(s)\, \Delta_i\left(s, s_i'\right) \geq 0$$

- A Linear Program with variables $\pi(s)$

Learning Dynamics and Correlated Equilibria

Regret vs Correlated Equilibrium

- No-regret property, implies

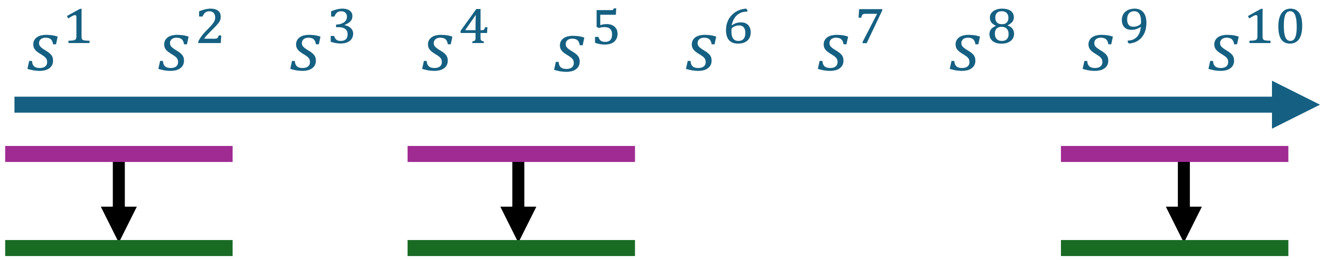
Distributions that satisfy this are called **Coarse Correlated Equilibria**

$$\forall s_i' : \sum_{s} \pi^T(s)\left(u_i(s) - u_i(s_i', s_{-i})\right) \geq -\tilde{\epsilon}(T, \delta) \to 0$$

- Correlated equilibrium requires conditioning on recommendation

$$\forall s_i^*, s_i' : \sum_{s : s_i = s_i^*} \pi^T(s)\left(u_i(s) - u_i(s_i', s_{-i})\right) \geq 0$$

At subset of periods when **played s_i^***



You don’t regret **switching to s_i'**

No-Swap Regret!

- No-regret property requires

$$\frac{1}{T}\sum_{t=1}^{T} u_i\left(s^t\right) \geq \max_{s_i' \in S_i} \frac{1}{T}\sum_{t=1}^{T} u_i\left(s_i', s_{-i}^{t}\right) - \tilde{\epsilon}(T, \delta)$$

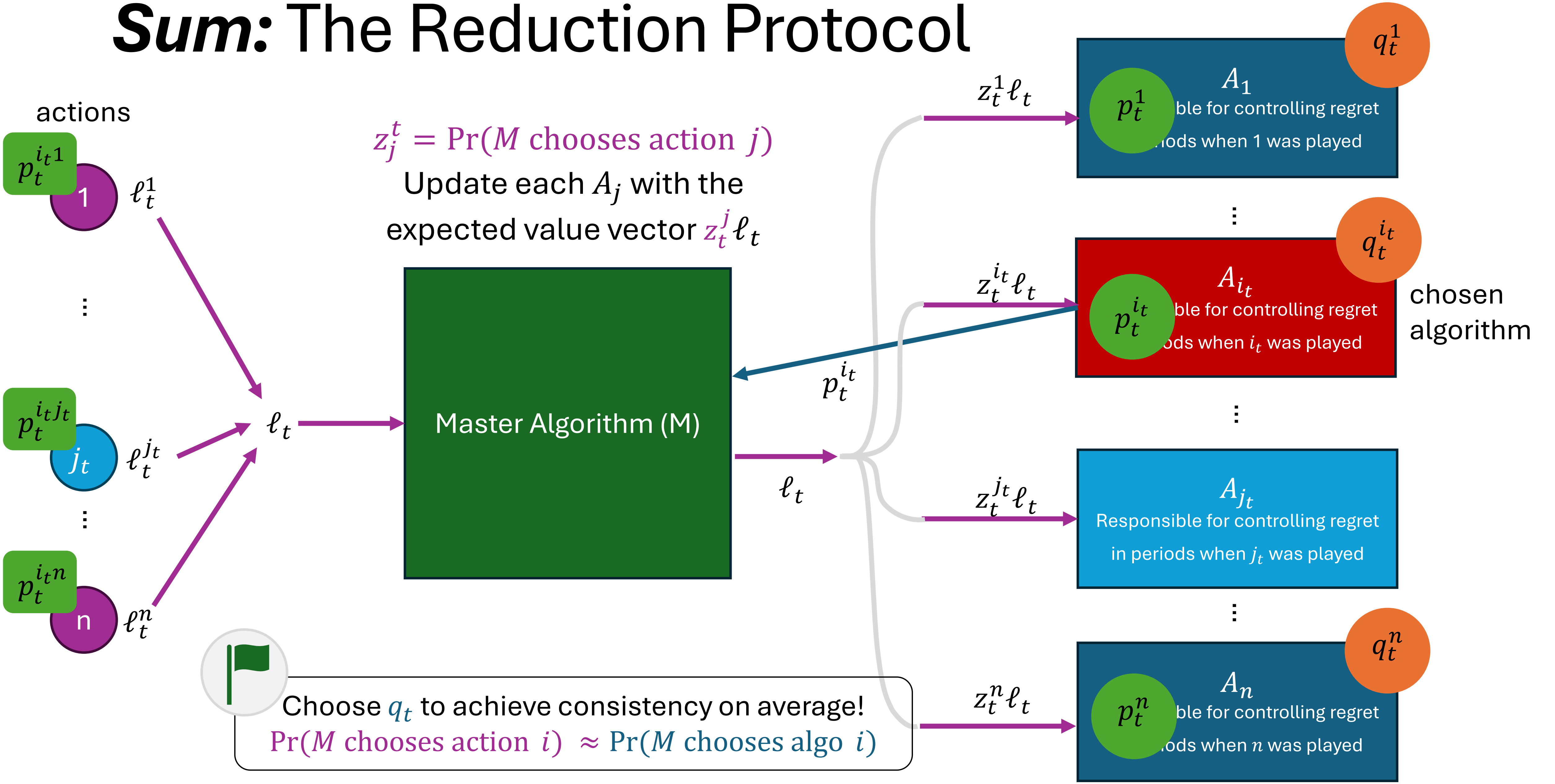
- No-swap regret property requires

$$\forall \phi\colon \frac{1}{T}\sum_{t=1}^{T} u_i\left(s^t\right) \geq \frac{1}{T}\sum_{t=1}^{T} u_i\left(\phi\left(s_i^t\right), s_{-i}^{t}\right) - \tilde{\epsilon}(T, \delta)$$

Theorem. If all players use no-swap regret algorithms, then the empirical joint distribution converges to a Correlated Equilibrium

Lecture 8

Sum: The Reduction Protocol



$z_j^t = \Pr(M \text{ chooses action } j)$

Update each A_j with the expected value vector $z_t^j \ell_t$

Choose q_t to achieve consistency on average!

$\Pr(M \text{ chooses action } i) \approx \Pr(M \text{ chooses algo } i)$

Sum: The reduction protocol

- At each period we choose each action with probability

$$z_t^j = \Pr(M \text{ choose action } j)$$

$$= \sum_i \underbrace{\Pr(M \text{ choose algo } A_i)}_{q_t^i} \cdot \underbrace{\Pr(A_i \text{ choose action } j)}_{p_t^{ij}}$$

- We update each algorithm A_j with loss vector

$$z_t^j \ell_t = \Pr(M \text{ choose action } j) \cdot (\text{loss vector})$$

- The distribution over algorithms q_t is chosen such that

$$\Pr(M \text{ choose action } j) \approx \Pr(M \text{ choose algo } A_j)$$

Recap: Choosing Distribution over Algos

Corollary. If we choose q_t as stationary distribution of the Markov Chain defined by transition probabilities $\mathrm{Pr}(\mathrm{i} \rightarrow \mathrm{j}) = p_t^{ij}$ then

$$\mathrm{Pr}(M \text{ choose action } j) = \mathrm{Pr}(M \text{ choose algo } A_j)$$

Therefore

$$\text{Swap Regret of Master} = \text{Total Fixed Action Regret of Algos} \rightarrow 0$$

Sum: The reduction protocol

- At each period calculate stationary distribution q_t of the Markov Chain defined by the transition probabilities $\Pr(i \to j) = p_t^{ij}$
- Choose each action with probability

$$z_t^j = \Pr(M \text{ choose action } j) = \Pr(M \text{ choose algo } j) = q_t^j$$

- Update each algorithm A_j with loss vector

$$z_t^j \ell_t = \Pr(M \text{ choose action } j) \cdot (\text{loss vector})$$

Overall Algorithm using EXP for each Algo

```
Initialize Pt with each row being the uniform distribution
For t in 1..T
    # Calculate choice probability q of master based on
    # choice probabilities Pt of algos
    Calculate stationary distribution q of matrix Pt
    Draw action jt based on distribution q
    Observe loss vector lt

    # update each algorithms choice probabilities
    For i in 1..n
        Calculate perceived loss plt[i] = q[i] * lt
        Pt[i] = EXP-Update(Pt[i], plt[i])
```

Recap: Final Theorem

Theorem. If we choose q_t as stationary distribution of the Markov Chain defined by transition probabilities $\Pr(\mathrm{i} \to \mathrm{j}) = p_t^{ij}$ and each algorithm updates their choice probabilities using the EXP rule then

$$\text{Average Swap Regret of Master} \leq 2n\sqrt{\frac{2\log(n)}{T}} \to 0$$

Recent example research in multi-agent RL using Correlated Equilibrium Techniques

Multi-Agent Training beyond Zero-Sum with Correlated Equilibrium Meta-Solvers

Luke Marris [1,2] Paul Muller [1,3] Marc Lanctot [1] Karl Tuyls [1] Thore Graepel [1,2]

Abstract

Two-player, constant-sum games are well studied in the literature, but there has been limited progress outside of this setting. We propose Joint Policy-Space Response Oracles (JPSRO), an algorithm for training agents in n-player, general-sum extensive form games, which provably converges to an equilibrium. We further suggest correlated equilibria (CE) as promising meta-solvers, and propose a novel solution concept Maximum Gini Correlated Equilibrium (MGCE), a principled and computationally efficient family of solutions for solving the correlated equilibrium selection problem. We conduct several experiments using CE meta-solvers for JPSRO and demonstrate convergence on n-player, general-sum games.

1. Introduction

Recent success in tackling two-player, constant-sum games (Silver et al., 2016; Vinyals et al., 2019) has outpaced progress in n-player, general-sum games despite a lot of interest (Jaderberg et al., 2019; OpenAI et al., 2019; Brown & Sandholm, 2019; Lockhart et al., 2020; Gray et al., 2020; Anthony et al., 2020). One reason is because Nash equilibrium (NE) (Nash, 1951) is tractable and interchangeable in the two-player, constant-sum setting but becomes intractable (Daskalakis et al., 2009) and potentially non-interchangeable[1] in n-player and general-sum settings. The problem of selecting from multiple solutions is known as the equilibrium selection problem (Goldberg et al., 2013; Avis et al., 2010; Harsanyi & Selten, 1988).[2]

Outside of normal form (NF) games, this problem setting arises in multi-agent training when dealing with empirical games (also called meta-games), where a game payoff tensor is populated with expected outcomes between agents playing an extensive form (EF) game, for example the StarCraft League (Vinyals et al., 2019) and Policy-Space Response Oracles (PSRO) (Lanctot et al., 2017), a recent variant of which reached state-of-the-art results in Stratego Barrage (McAleer et al., 2020).

In this work we propose using correlated equilibrium (CE) (Aumann, 1974) and coarse correlated equilibrium (CCE) as a suitable target equilibrium space for n-player, general-sum games[3]. The (C)CE solution concept has two main benefits over NE; firstly, it provides a mechanism for players to correlate their actions to arrive at mutually higher payoffs and secondly, it is computationally tractable to compute solutions for n-player, general-sum games (Daskalakis et al., 2009). We provide a tractable approach to select from the space of (C)CEs (MG), and a novel training framework that converges to this solution (JPSRO). The result is a set of tools for theoretically solving any complete information[4] multi-agent problem. These tools are amenable to scaling approaches; including utilizing reinforcement learning, function approximation, and online solution solvers, however we leave this to future work.

In Section 2 we provide background on a) correlated equilibrium (CE), an important generalization of NE, b) coarse correlated equilibrium (CCE) (Moulin & Vial, 1978), a similar solution concept, and c) PSRO, a powerful multi-agent training algorithm. In Section 3 we propose novel solution concepts called Maximum Gini (Coarse) Correlated Equilibrium (MG(C)CE) and in Section 4 we thoroughly explore its properties including tractability, scalability, invariance, and

[1] DeepMind [2] University College London [3] Université Gustave Eiffel. Correspondence to: Luke Marris <marris@google.com>.

Proceedings of the 38th International Conference on Machine

[2] The equilibrium selection problem is subtle and can have various interpretations. We describe it fully in Section 4.1 based

Computational Game Theory for Complex Games

1

- Basics of game theory and zero-sum games (T)
- Basics of online learning theory (T)
- Solving zero-sum games via online learning (T)
- *HW1: implement simple algorithms to solve zero-sum games*
- Applications to ML and AI (T+A)
- *HW2: implement boosting as solving a zero-sum game*

2

- Basics of extensive-form games
- Solving extensive-form games via online learning (T)
- *HW3: implement agents to solve very simple variants of poker*

3

- General games, equilibria and online learning (T)
- Online learning in general games (T+A)
- *HW4: implement no-regret algorithms that converge to correlated equilibria in general games*

Data Science for Auctions and Mechanisms

4

- Basics and applications of auction theory (T+A)
- Basic Auctions and Learning to bid in auctions (T)
- *HW5: implement bandit algorithms to bid in ad auctions*

5

- Optimal auctions and mechanisms (T)
- Simple vs optimal mechanisms (T)
- *HW6: implement simple and optimal auctions, analyze revenue empirically*

6

- Basics of Statistical Learning Theory (T)
- Optimizing Mechanisms from Samples (T)
- *HW7: implement procedures to learn approximately optimal auctions from historical samples*

Further Topics

7

- Econometrics in games and auctions (T+A)
- A/B testing in markets (T+A)
- *HW8: implement procedure to estimate values from bids in an auction, empirically analyze inaccuracy of A/B tests in markets*

Guest Lectures

- Mechanism Design for LLMs, Renato Paes Leme, Google Research
- Auto-bidding in Sponsored Search Auctions, Kshipra Bhawalkar, Google Research

Lecture 9

Sotheby's Sells $7.3 Billion in Art, Fueled by Moneyed Millennials

Auction house says millennials drove art market rebound by bidding up everything from luxury goods to NFTs

Credits: https://www.wsj.com/articles/sothebys-sells-7-3-billion-in-art-fueled-by-moneyed-millennials-11639581146

Sum: Auction Applications

- Traditionally, selling of luxury goods, art
- Digital auction markets for goods (eBay)
- Energy markets
- Digital ad markets (sponsored search, display ads, amazon ads)
- Spectrum auctions
- Government procurement auctions
- Web3.0 transaction protocols

Auction Basics

- n bidders are interested in acquiring an item

- Bidder i has value v_i for the item
- Value is known only to them (private information)
- If bidder wins the item ($x_i = 1$) they gain a value v_i

- If at the end they are asked to pay a price p_i they gain

$$u_i(x_i, p_i; v_i) = v_i \cdot x_i - p_i$$

Sum: First Price

- First Price is arguably the simplest auction rule
- It can be hard to strategize in such an auction
- The auction can lead to inefficient allocations

- Though approximately efficient
- Still used in practice in many settings (e.g. online advertising, government procurement)
- Primarily because it has very transparent rules

Sum: Second Price

- Second Price is arguably the simplest truthful auction rule
- It is very easy to strategize in such an auction (be truthful)
- Auction always leads to efficient allocations (highest value wins)
- Auction can be run very quickly (computationally efficient)

- Still not always the auction used in many places
- Primarily because it has not very transparent rules
- Susceptible to collusion and manipulations by the auctioneer

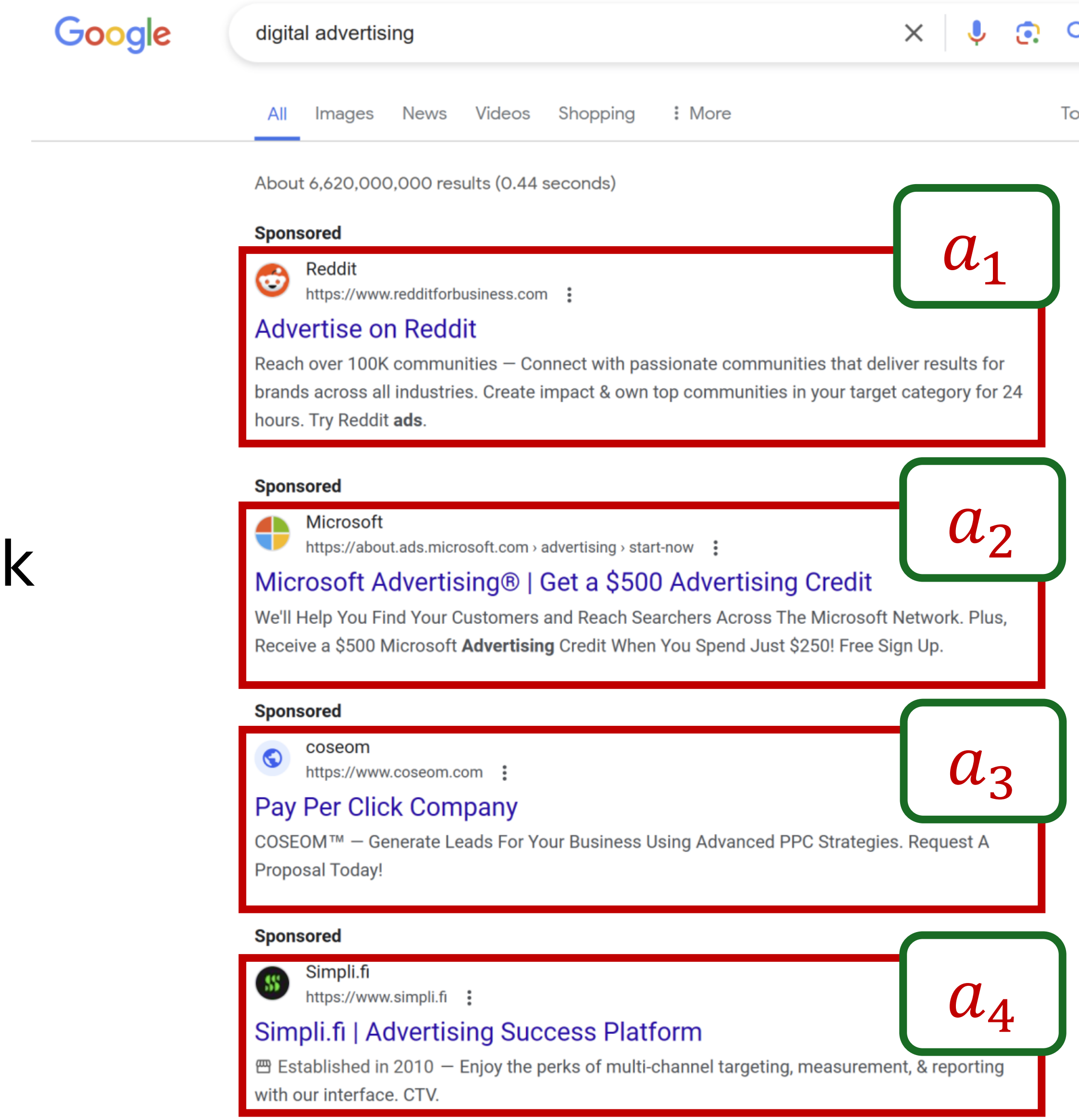
Sponsored Search Auctions

- Now we have many items to sell
- Slots on a web impressions

- Higher slots get more clicks!
- Each slot has some probability of click

$$a_1 > a_2 > \cdots > a_m$$

- Bidders have a value-per-click v_i



About 6,620,000,000 results (0.44 seconds)

Sponsored

Reddit
https://www.redditforbusiness.com

Advertise on Reddit

Reach over 100K communities — Connect with passionate communities that deliver results for brands across all industries. Create impact & own top communities in your target category for 24 hours. Try Reddit **ads**.

Sponsored

Microsoft
https://about.ads.microsoft.com › advertising › start-now

Microsoft Advertising® | Get a $500 Advertising Credit

We'll Help You Find Your Customers and Reach Searchers Across The Microsoft Network. Plus, Receive a $500 Microsoft **Advertising** Credit When You Spend Just $250! Free Sign Up.

Sponsored

coseom
https://www.coseom.com

Pay Per Click Company

COSEOM™ — Generate Leads For Your Business Using Advanced PPC Strategies. Request A Proposal Today!

Sponsored

Simpli.fi
https://www.simpli.fi

Simpli.fi | Advertising Success Platform

Established in 2010 — Enjoy the perks of multi-channel targeting, measurement, & reporting with our interface. CTV.

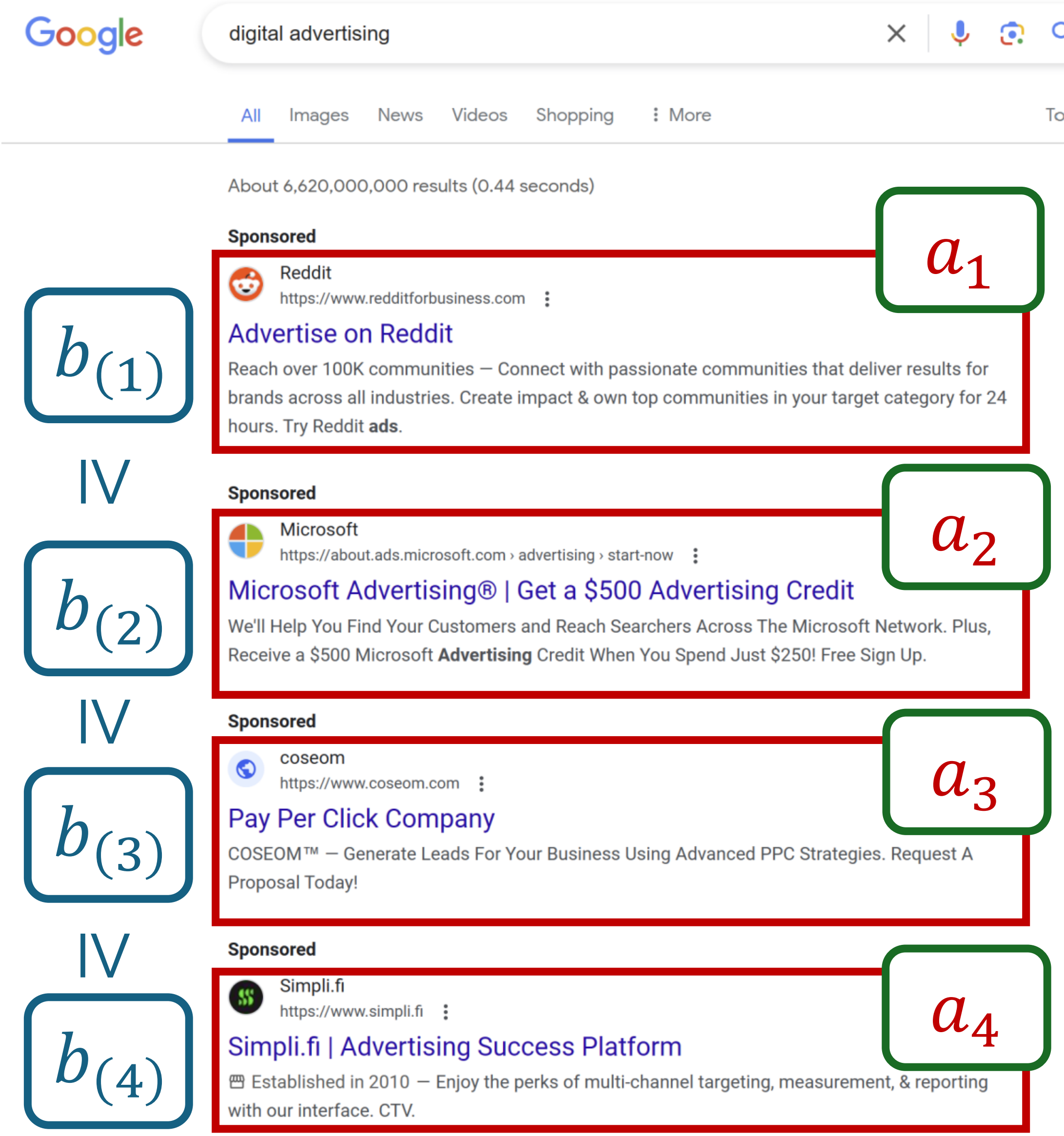
Generalized Second Price (GSP) Auction

- Bidders submit a bid-per-click b_i

- Slots allocated in decreasing order of bids
- Bidder i is allocated slot $j_i(b)$

- Bidder pays the next highest bid when clicked

$$u_i(b; v_i) = a_{j_i(b)} \cdot \left(v_i - b_{(j_i(b)+1)}\right)$$



digital advertising

About 6,620,000,000 results (0.44 seconds)

Sponsored

Reddit
https://www.redditforbusiness.com

Advertise on Reddit

Reach over 100K communities — Connect with passionate communities that deliver results for brands across all industries. Create impact & own top communities in your target category for 24 hours. Try Reddit **ads**.

Sponsored

Microsoft
https://about.ads.microsoft.com › advertising › start-now

Microsoft Advertising® | Get a $500 Advertising Credit

We'll Help You Find Your Customers and Reach Searchers Across The Microsoft Network. Plus, Receive a $500 Microsoft **Advertising** Credit When You Spend Just $250! Free Sign Up.

Sponsored

coseom
https://www.coseom.com

Pay Per Click Company

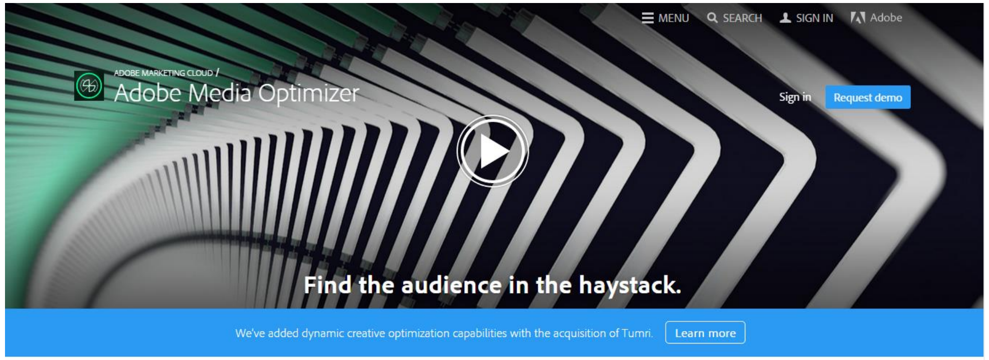
COSEOM™ — Generate Leads For Your Business Using Advanced PPC Strategies. Request A Proposal Today!

Sponsored

Simpli.fi
https://www.simpli.fi

Simpli.fi | Advertising Success Platform

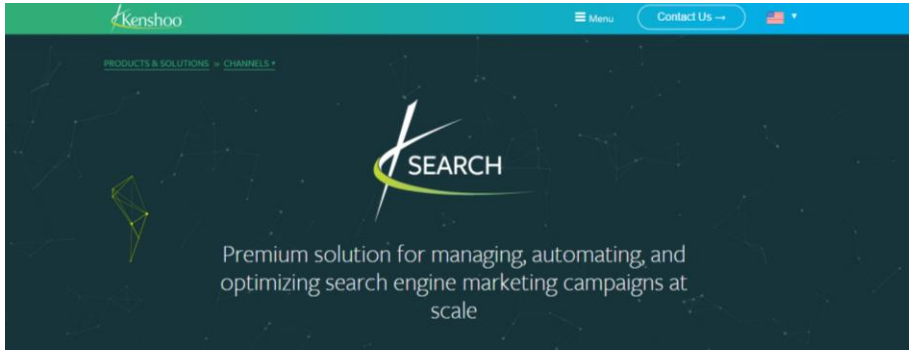
Established in 2010 — Enjoy the perks of multi-channel targeting, measurement, & reporting with our interface. CTV.



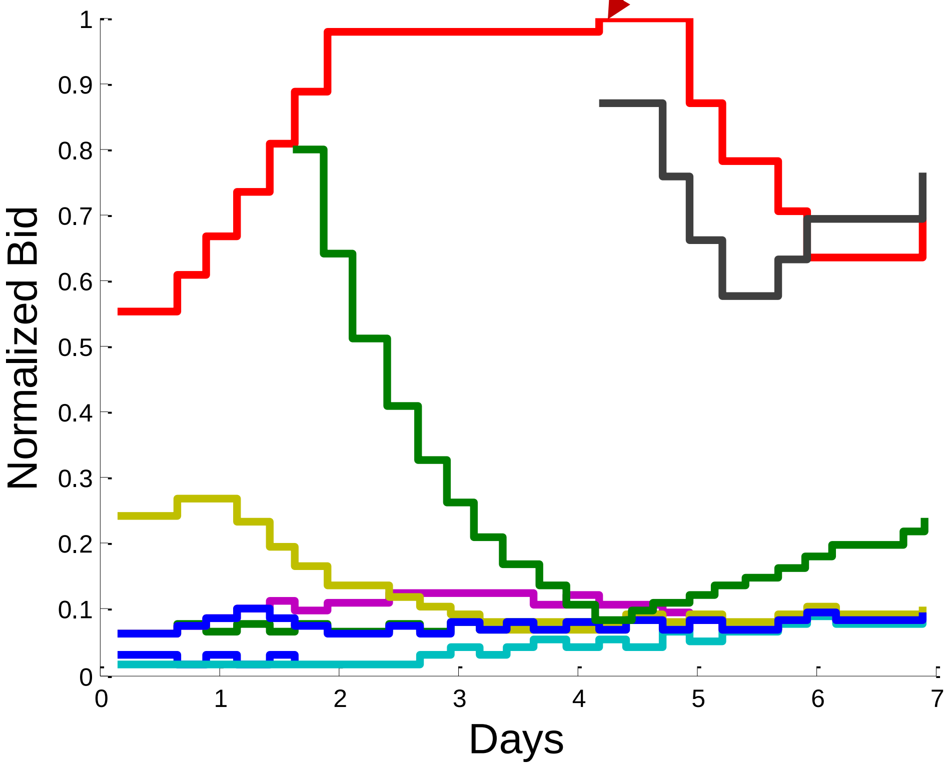
Adobe Media Optimizer

Find the audience in the haystack.

We've added dynamic creative optimization capabilities with the acquisition of Tumri.



Premium solution for managing, automating, and optimizing search engine marketing campaigns at scale



Bid on some keyword

Credits: https://arxiv.org/abs/1505.00720

Lecture 10

Right intuition, why Second-Price is truthful

- Second price is truthful **not because** we charge next highest bid
- Second price is truthful **not because** we charge smallest bid to maintain the same allocation

- Second price is truthful **because** we charged the winner their “externalities to the rest of society”

The Vickrey-Clarke-Groves (VCG) Mechanism

General Auction (Mechanism Design) Setting

- Auctioneer (Designer) wants to choose among set of outcomes O
- Each bidder i has some value for each outcome $v_i(o) \in R$
- The value function v_i is called the **type** of player i
- Designer elicits **types/bids** from players $b = (b_1, \dots, b_n)$
- Designer chooses allocation that maximizes the reported welfare

$$x(b) = \operatorname*{argmax}_{o \in O} RW(o; b) \coloneqq \sum_{i=1}^{n} b_i(o)$$

Total Reported Welfare

General Auction (Mechanism Design) Setting

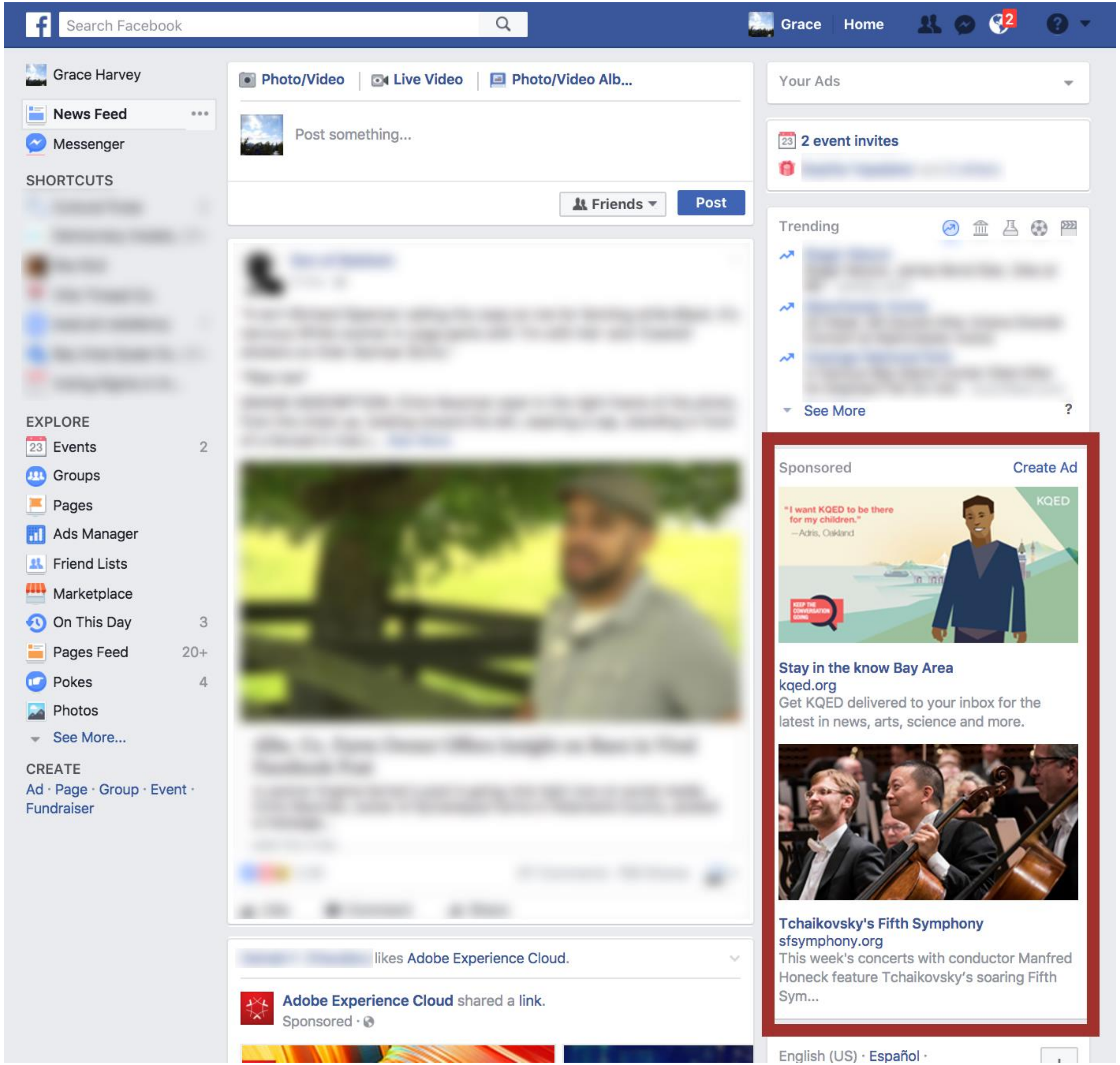
- Designer chooses allocation that maximizes the reported welfare

$$x(b) = \operatorname*{argmax}_{o \in O} RW(o; b) \coloneqq \sum_{i=1}^{n} b_i(o)$$

- Charges to each player their externalities as payment

$$p_i(b) = \underbrace{\max_{o \in O} \sum_{j \neq i} b_j(o)}_{\text{RWelfare of others without me}} - \underbrace{\sum_{j \neq i} b_j\big(x(b)\big)}_{\text{RWelfare of others with me}} \boxed{\geq 0}$$

Why?



Stay in the know Bay Area
kqed.org
Get KQED delivered to your inbox for the latest in news, arts, science and more.

Tchaikovsky's Fifth Symphony
sfsymphony.org
This week's concerts with conductor Manfred Honeck feature Tchaikovsky's soaring Fifth Sym...

likes Adobe Experience Cloud.

Adobe Experience Cloud shared a link.
Sponsored ·

Credits: https://rethinkmedia.org/blog/facebook-instagram-advertising-what-you-need-know-get-started

Learning in Non-Truthful Auctions

Learning how to bid in auctions

- Given the complexity of digital auction markets
- Given the hardness of strategizing in non-truthful auctions
- Many of these auctions are repeated!

- It makes sense to study learning over time, to decide how to bid

- How do we learn over time when we repeatedly participate in an auction? Can we compete with the best fixed bid in hindsight?

No-Regret Learning with Limited Feedback

- Want to choose my bids b_i^t, based on algorithm that guarantees

$$\frac{1}{T}\sum_{t=1}^{T} u_i\left(b^t\right) \geq \max_{b_i \in [N]} \frac{1}{T}\sum_{t=1}^{T} u_i\left(b_i, b^t\right) - \epsilon(T)$$

- Seems like a standard N action no-regret problem

- What's the catch! I don't receive after each period the utility for all my actions. Only the utility for action I took!
- Limited Feedback. I cannot calculate how much I would have gotten with any other bid (e.g. in an FP, solely knowing whether I won or not).

No-Regret Learning with Bandit Feedback

At each period t

- Adversary chooses a loss vector $\ell_t \in [0,1]^N$
- I choose an action i_t (not knowing ℓ_t)
- I observe loss of my chosen action $\ell_t^{i_t}$

- I want to guarantee small expected regret with any fixed action:

$$\max_{i \in N} E\left[\frac{1}{T}\sum_{t=1}^{T} \ell_t^{i_t} - \ell_t^{i}\right] \leq \epsilon(T)$$

The EXP Algorithm with Bandit Feedback

```
Initialize pt to the uniform distribution
For t in 1..T
    Draw action jt based on distribution pt
    Observe loss of chosen action lt[jt]

    Construct un-biased proxy loss vector
      ltproxy[j] = 1(jt=j) * lt[jt] / pt[jt]

    Update probabilities based on EXP update
      pt = pt * exp(-eta * ltproxy)
      pt = pt / sum(pt)
```

Update: Regret of EXP

(EXP) $$p_t = \operatorname*{argmin}_{p\in\Delta} \sum_{\tau<t} \langle p, \tilde{\ell}_\tau \rangle + \boxed{\frac{1}{\eta}\mathcal{R}(p)} \left(\begin{matrix}\text{Negative}\\ \text{Entropy}\end{matrix}\right) \quad \mathcal{R}(p) = \sum_{i=1}^{n} p_i \log(p_i)$$

$$p_t \propto p_{t-1} \exp\left(-\eta\, \tilde{\ell}_{t-1}\right)$$

Theorem. Assuming $\tilde{\ell}_t$ are random proxies that, conditional on history, have expected value equal to true loss vector ℓ_t and $\tilde{\ell}_t \geq 0$, then regret of EXP is bounded as:

$$\text{Regret} - \text{EXP}(T) \leq \frac{\eta}{T} \sum_t E\left[\sum_j p_t^j\, E\left[\left(\tilde{\ell}_t^j\right)^2\right]\right] + \frac{\log(N)}{\eta T}$$

Expected Average
“Variance”?

Update: Regret of EXP

(EXP) $$p_t = \operatorname*{argmin}_{p \in \Delta} \sum_{\tau < t} \langle p, \tilde{\ell}_\tau \rangle + \boxed{\frac{1}{\eta} \mathcal{R}(p)} \begin{pmatrix} \text{Negative} \\ \text{Entropy} \end{pmatrix} \quad \mathcal{R}(p) = \sum_{i=1}^{n} p_i \log(p_i)$$

$$p_t \propto p_{t-1} \exp(-\eta\, \tilde{\ell}_{t-1})$$

Theorem. Assuming $\tilde{\ell}_t$ are random proxies that, conditional on history, have expected value equal to true loss vector ℓ_t and $\tilde{\ell}_t \geq 0$, then regret of EXP is bounded as:

$$\text{Regret} - \text{EXP}(T) \leq \eta N + \frac{\log(N)}{\eta T} \Rightarrow \text{Regret} - \text{EXP}(T) \lesssim \sqrt{\frac{N \log(N)}{T}}$$

For $\eta \sim \sqrt{\frac{\log(N)}{NT}}$

See Lemma 20 and Theorem 2.22 and Corollary 4.2 of mal-018.dvi (huji.ac.il)

Computational Game Theory for Complex Games

1

- Basics of game theory and zero-sum games (T)
- Basics of online learning theory (T)
- Solving zero-sum games via online learning (T)
- *HW1: implement simple algorithms to solve zero-sum games*
- Applications to ML and AI (T+A)
- *HW2: implement boosting as solving a zero-sum game*

2

- Basics of extensive-form games
- Solving extensive-form games via online learning (T)
- *HW3: implement agents to solve very simple variants of poker*

3

- General games, equilibria and online learning (T)
- Online learning in general games (T+A)
- *HW4: implement no-regret algorithms that converge to correlated equilibria in general games*

Data Science for Auctions and Mechanisms

4

- Basics and applications of auction theory (T+A)
- Basic Auctions and Learning to bid in auctions (T)
- *HW5: implement bandit algorithms to bid in ad auctions*

5

- Optimal auctions and mechanisms (T)
- Simple vs optimal mechanisms (T)
- *HW6: implement simple and optimal auctions, analyze revenue empirically*

6

- Basics of Statistical Learning Theory (T)
- Optimizing Mechanisms from Samples (T)
- *HW7: implement procedures to learn approximately optimal auctions from historical samples*

Further Topics

7

- Econometrics in games and auctions (T+A)
- A/B testing in markets (T+A)
- *HW8: implement procedure to estimate values from bids in an auction, empirically analyze inaccuracy of A/B tests in markets*

Guest Lectures

- Mechanism Design for LLMs, Renato Paes Leme, Google Research
- Auto-bidding in Sponsored Search Auctions, Kshipra Bhawalkar, Google Research

Lecture 11

What if we want to maximize revenue?

What if we post a reserve price?

How do we optimize over all possible mechanisms!

Single-Parameter Settings

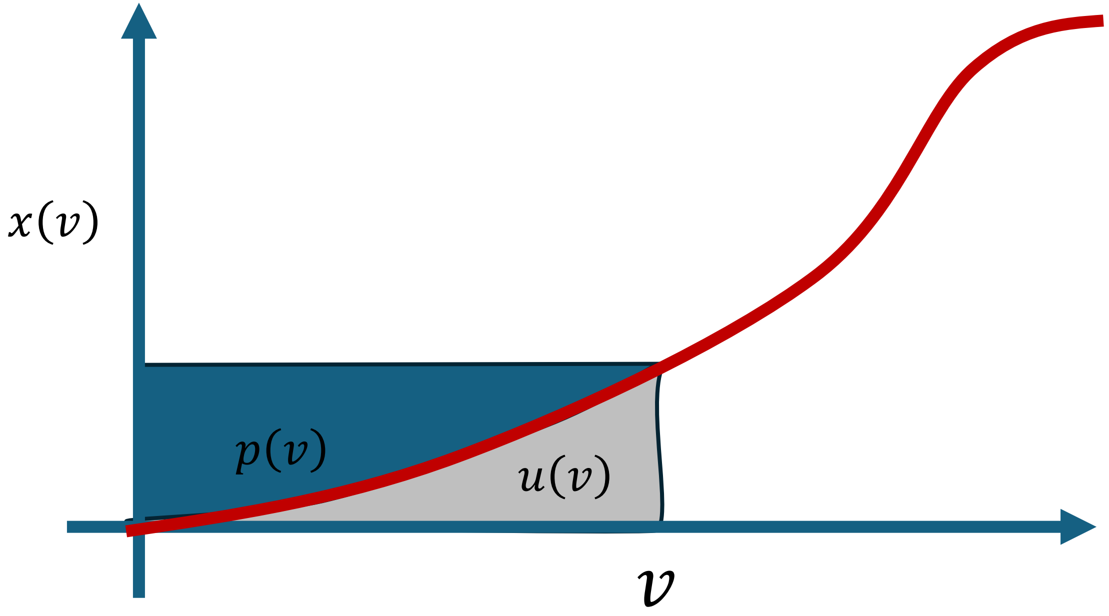
- Each bidder has some value v_i for being allocated
- Bidders submit a reported value b_i (without loss of generality)
- Mechanism decides on an allocation $x \in X \subseteq \{0,1\}^n$
- Mechanism fixes a probabilistic allocation rule:
$$x(b) \in \Delta(X)$$

- **First question.** Given an allocation rule, when can we find a payment rule p so that the overall mechanism is truthful?
- If we can find such a payment, we will say that x is implementable

Any implementable allocation rule must be monotone!

"If not allocated with value v, I should not be allocated if I report a lower value!"

For any dominant-strategy truthful, NNT and IR mechanism, given an allocation rule, utility and payment are uniquely determined!



Myerson's Theorem. When valuations are independently distributed, for any dominant-strategy truthful, NNT and IR mechanism, the payment contribution of each player is their expected virtual value

$$E[p_i(v)] = E[x_i(v) \cdot \phi_i(v_i)], \qquad \phi_i(v_i) = v_i - \frac{1 - F_i(v_i)}{f_i(v_i)}$$

Myerson's Optimal Auction. Assuming that virtual value functions are monotone non-decreasing, the optimal mechanism is the mechanism that maximizes virtual welfare

$$x(v) = \mathrm{argmax}_{x \in X} \sum_i x \cdot \phi_i(v_i), \qquad p_i(v) = v_i x_i(v) - \int_0^{v_i} x_i(z, v_{-i})\, dz$$

$$\mathrm{Rev} = E\left[\max_{x \in X} \sum_i x \cdot \phi_i(v_i)\right]$$

Lecture 12

Can non-truthful mechanisms generate higher revenue at some Bayes-Nash equilibrium?

Bayesian-Incentive Compatible Mechanism

- A **direct mechanism** elicits private values and comprises of an allocation function x and a payment function p
- **BIC.** bidders have no incentive to deviate from truthful reporting

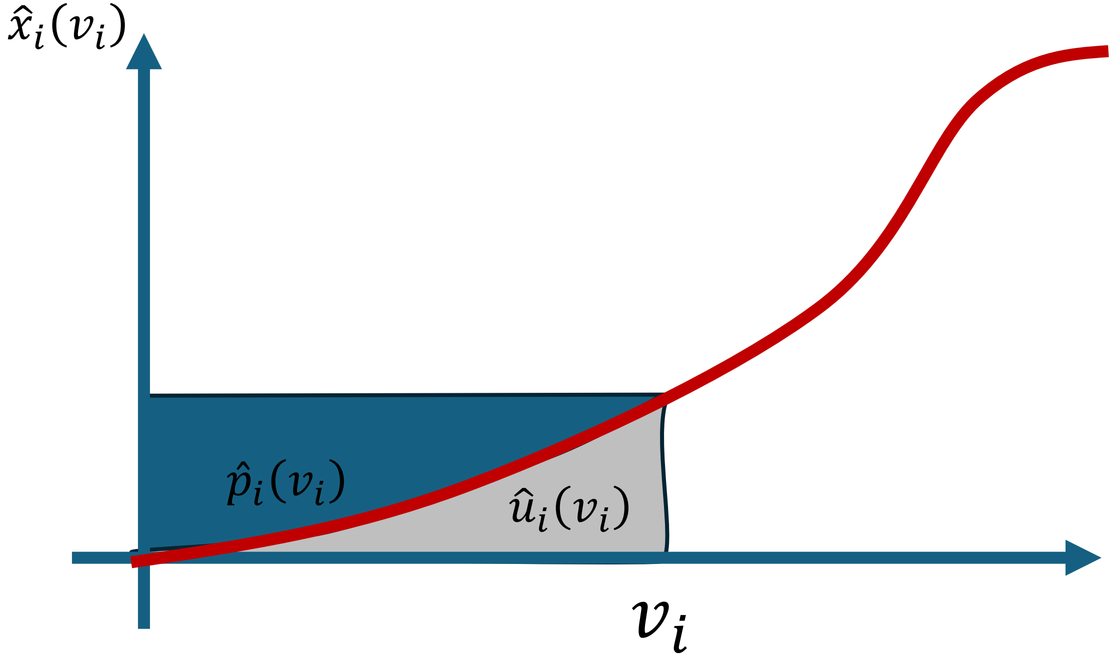
$$E[u_i(v; v_i) \mid v_i] \geq E\left[u_i\left(v_i', v_{-i}; v_i\right) \mid v_i\right]$$

$$E[v_i x_i(v) - p(v) \mid v_i] \geq E\left[v_i x_i\left(v_i', v_{-i}\right) - p_i\left(v_i', v_{-i}\right) \mid v_i\right]$$

- Implies "interim" expected utility, allocation and payment for bidder i

$$\hat{u}_i(v_i) = E_{v_{-i}}[u_i(v)], \qquad \hat{x}_i(v_i) = E_{v_{-i}}[x_i(v)], \qquad \hat{p}_i(v_i) = E_{v_{-i}}[p_i(v)]$$

For any BIC, NNT and BIR mechanism (and any BNE of a non-truthful mechanism), given the interim allocation rule, utility and payment are uniquely determined!



Myerson's Theorem. When valuations are independently distributed, for any BIC, NNT and IR mechanism (and any BNE of a non-truthful mechanism), the payment contribution of each player is their expected virtual value

$$E[\hat{p}_i(v_i)] = E[\hat{x}_i(v_i) \cdot \phi_i(v_i)], \qquad \phi_i(v_i) = v_i - \frac{1 - F_i(v_i)}{f_i(v_i)}$$

Myerson's Optimal Auction. Assuming that virtual value functions are monotone non-decreasing, the mechanism that maximizes virtual welfare, achieves the largest possible revenue among all possible mechanisms and Bayes-Nash

$$x(v) = \text{argmax}_{x\in X} \sum_i x \cdot \phi_i(v_i)\,, \qquad p_i(v) = v_i x_i(v) - \int_0^{v_i} x_i(z, v_{-i})\, dz$$

$$\text{Rev} = E\left[\max_{x\in X} \sum_i x \cdot \phi_i(v_i)\right]$$

Optimal auction is
1) cumbersome, 2) hard to understand, 3) hard to explain, 4) does not always allocate to the highest value player, 5) discriminates a lot, 6) is many times counter-intuitive, 7) can seem unfair!

Are there simpler auctions that always achieve almost as good revenue?

Second-Price with Player-Specific Reserves

- What if we simply run a second price auction but have different reserves for each bidder

- Each bidder i has a reserve price r_i
- Reject all bidders with bid below the reserve
- Among all bidders with value $v_i \geq r_i$, allocate to highest bidder
- Charge winner max of their reserve and the next highest surviving bid

Second-Price with Player-Specific Reserves

Theorem. There exist personalized reserve prices such that the above auction achieves at least ½ of the optimal auction revenue!

- Choose θ such that:

$$\Pr\left(\max_i \phi_i^+(v_i) \geq \theta\right) = 1/2$$

- Then set personalized reserve prices implied by:

$$\phi_i^+(v_i) \geq \theta \Leftrightarrow v_i \geq r_i$$

Computational Game Theory for Complex Games

1

- Basics of game theory and zero-sum games (T)
- Basics of online learning theory (T)
- Solving zero-sum games via online learning (T)
- *HW1: implement simple algorithms to solve zero-sum games*
- Applications to ML and AI (T+A)
- *HW2: implement boosting as solving a zero-sum game*

2

- Basics of extensive-form games
- Solving extensive-form games via online learning (T)
- *HW3: implement agents to solve very simple variants of poker*

3

- General games, equilibria and online learning (T)
- Online learning in general games (T+A)
- *HW4: implement no-regret algorithms that converge to correlated equilibria in general games*

Data Science for Auctions and Mechanisms

4

- Basics and applications of auction theory (T+A)
- Basic Auctions and Learning to bid in auctions (T)
- *HW5: implement bandit algorithms to bid in ad auctions*

5

- Optimal auctions and mechanisms (T)
- Simple vs optimal mechanisms (T)
- *HW6: implement simple and optimal auctions, analyze revenue empirically*

6

- Basics of Statistical Learning Theory (T)
- Optimizing Mechanisms from Samples (T)
- *HW7: implement procedures to learn approximately optimal auctions from historical samples*

Further Topics

7

- Econometrics in games and auctions (T+A)
- A/B testing in markets (T+A)
- *HW8: implement procedure to estimate values from bids in an auction, empirically analyze inaccuracy of A/B tests in markets*

Guest Lectures

- Mechanism Design for LLMs, Renato Paes Leme, Google Research
- Auto-bidding in Sponsored Search Auctions, Kshipra Bhawalkar, Google Research

Lecture 13

All these designs required knowledge of distributions of values F_i!

What can we do if we only have data from F_i?

Basic Elements of Statistical Learning Theory

General Framework

- Given samples $S = \{v_1, \ldots, v_m\}$ that are i.i.d. from distribution F
- Given a hypothesis/function space H
- Given a reward function $r(v; h)$

- Goal is to maximize the expected reward over distribution F

$$R(h) = E_{v\sim F}[r(v; h)]$$

Desiderata

- Without knowledge of distribution F, we want to produce a hypothesis h_S, that achieves good reward on this distribution
- For some $\epsilon(m) \to 0$ as the number of samples grows:

$$R(h_S) \stackrel{\text{def}}{=} E_{v\sim F}[r(v;h)] \geq \max_{h\in H} R(h) - \epsilon(m)$$

- Either in expectation over the draw of the samples, i.e.

$$E_S[R(h_S)] \geq \max_{h\in H} R(h) - \epsilon(m)$$

- Or with high-probability over the draw of the samples, i.e.

$$\text{w.p.}\, 1-\delta: \quad R(h_S) \geq \max_{h\in H} R(h) - \epsilon_\delta(m)$$

Desiderata (Mechanism Design from Samples)

- Without knowledge of Distribution of value profiles F, we want to produce a hypothesis h_S, that achieves good Revenue on this distribution
- For some $\epsilon(m) \to 0$ as the number of samples grows:

$$R(h_S) \overset{\text{def}}{=} E_{v\sim F}\left[\sum_i p_i(v)\right] \geq \max_{h\in H} R(h) - \epsilon(m)$$

- Either in expectation over the draw of the samples, i.e.

$$E_S[R(h_S)] \geq \max_{h\in H} R(h) - \epsilon(m)$$

- Or with high-probability over the draw of the samples, i.e.

$$\text{w.p. } 1-\delta: \quad R(h_S) \geq \max_{h\in H} R(h) - \epsilon_\delta(m)$$

The Obvious Algorithm

- We want to choose r that maximizes

$$\max_{h \in H} R(h) \stackrel{\text{def}}{=} E_{v \sim F}[r(v; h)], \qquad \text{(population objective)}$$

- With m samples, we can optimize average reward on samples!

$$\max_{h \in H} R_S(h) \stackrel{\text{def}}{=} \frac{1}{m} \sum_{j=1}^{m} r(v_j; h), \qquad \text{(empirical objective)}$$

- This approach is called Empirical Reward Maximization (ERM)
- **Intuition.** Since each value is drawn from distribution F the empirical average over i.i.d. draws from F, by law of large numbers, should be very close to expected value

Lecture 14

If we can bound representativeness

$$\mathrm{Rep} = E_S\left[\sup_h R_S(h) - R(h)\right] \leq \epsilon(m)$$

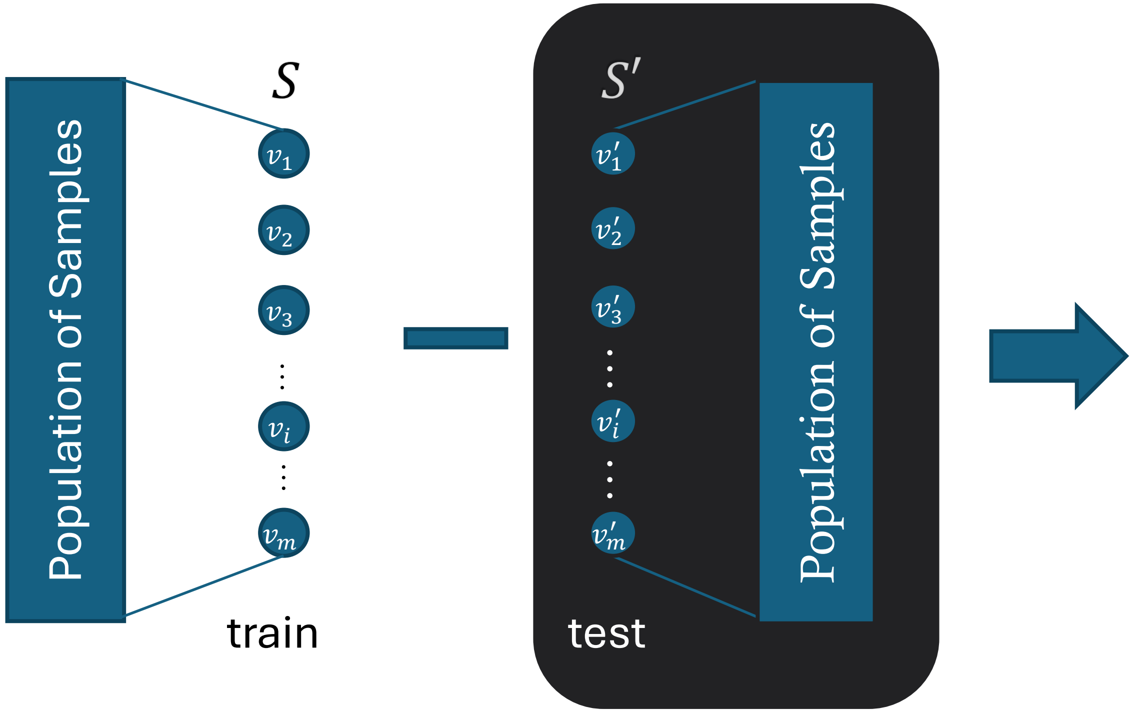
Then we can bound expected performance

$$E[R(h_S)] \geq E[R(h_*)] - \epsilon(m)$$

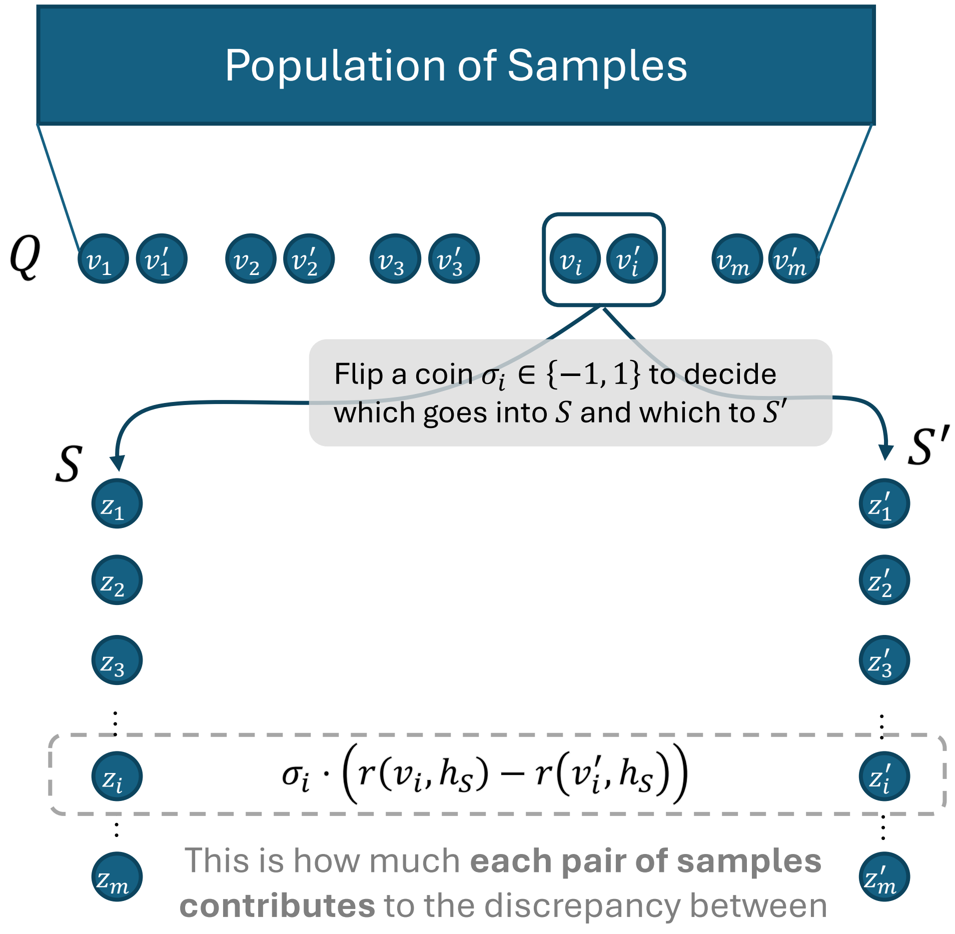
$$\mathrm{Rep}_* = E[R_S(h_S) - R_D(h_S)]$$

$$\mathrm{Rep}_* = E\left[R_S(h_S) - E_{S'}[R_{S'}(h_S)]\right]$$

$$\mathrm{Rep}_* = E_{S,S'}[R_S(h_S) - R_{S'}(h_S)]$$



Equivalent Process


Population of Samples
Q
v_1 v'_1 v_2 v'_2 v_3 v'_3 v_i v'_i v_m v'_m
S
S'
z_1 z_2 z_3 z_i z_m
z'_1 z'_2 z'_3 z'_i z'_m

Flip a coin $\sigma_i \in \{-1, 1\}$ to decide which goes into S and which to S'

$$\sigma_i \cdot \left(r(v_i, h_S) - r(v'_i, h_S)\right)$$

This is how much **each pair of samples contributes** to the discrepancy between train and test reward

$$\mathrm{Rep}_* \le E_{Q,\sigma}\left[\max_{h\in H} \frac{1}{m}\sum_{i=1}^{m} \sigma_i \cdot \left(r(v_i; h) - r(v'_i; h)\right)\right]$$

Choose $h_S \in H$ to maximize average reward on S

Learner

For any Q, σ, choose $h \in H$ to maximize the average discrepancy

Adversary

Empirical Rademacher Complexity

Empirical Rademacher Complexity of hypothesis space H on samples S:

$$\mathrm{Rad}(S,H) \coloneqq 2E_\sigma\left[\max_{h\in H}\frac{1}{m}\sum_{i=1}^{m}\sigma_i\cdot r(v_i;h)\right]$$

Theorem. We have thus proven that:

$$E[R(h_S)] \geq R(h_*) - E_S[\mathrm{Rad}(S,H)]$$

Massart’s lemma. For any finite hypothesis space H:

$$\mathrm{Rad}(S, H) \leq 2\sqrt{\frac{2\log(|H|)}{m}}$$

Growth Rate of Function Space

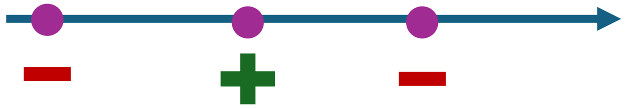
- Suppose we can find a finite subspace $\widetilde{H}_S \subseteq H$ such that every $h \in H$ has a **representative** $\tilde{h} \in \widetilde{H}_S$ that has the exact same behavior on the samples S
$$\forall v_i \in S: r(v_i; h) = r\left(v_i; \tilde{h}\right)$$
- Empirical Rademacher Complexity of H is upper bounded by that of $\widetilde{H}_S$
- Growth Rate $\tau(m, H)$: the size of the smallest $\widetilde{H}_S$ that satisfies the above property, in the worst case over sample dataset of size m
- Example. For threshold classifiers $\tau(m, H) = m + 1$

Theorem. For any hypothesis H

$$\mathrm{Rad}(S, H) \leq \boxed{2\sqrt{\frac{2\log\left(\tau(m, H)\right)}{m}}}$$

SideNote For classification, a seminal notion is the Vapnik-Chervonenkis **(VC) dimension**: size d of largest dataset that the hypothesis can assign labels in all possible manners

Cannot be assigned by threshold classifiers ⇒ $d = 2$



Sauer's Lemma. If has VC-dim $\leq d$ then $\tau(m, H) \lesssim 2^d \Rightarrow \mathrm{Rad}(S, H) \lesssim \sqrt{d/m}$

Discretization on Samples

- Suppose we can find a finite subspace $\widetilde{H}_{S,\epsilon} \subseteq H$ such that every $h \in H$ has a **representative** $\tilde{h} \in \widetilde{H}_{S,\epsilon}$ that has approximately the same behavior on the samples S

$$\forall v_i \in S\colon \left|r(v_i; h) - r\left(v_i; \tilde{h}\right)\right| \leq \epsilon$$

- Empirical Rademacher Complexity of H upper bounded approximately by $\widetilde{H}_{S,\epsilon}$

$$\mathrm{Rad}(S, H) \coloneqq 2E_\sigma\left[\max_{h\in H} \frac{1}{m}\sum_{i=1}^{m} \sigma_i \cdot r(v_i; h)\right]$$

$$\leq 2E_\sigma\left[\max_{h\in \widetilde{H}_{S,\epsilon}} \frac{1}{m}\sum_{i=1}^{m} \sigma_i \cdot r(v_i; h)\right] + 2\epsilon \leq 2\sqrt{\frac{2\log\left(\left|\widetilde{H}_{S,\epsilon}\right|\right)}{m}} + 2\epsilon$$

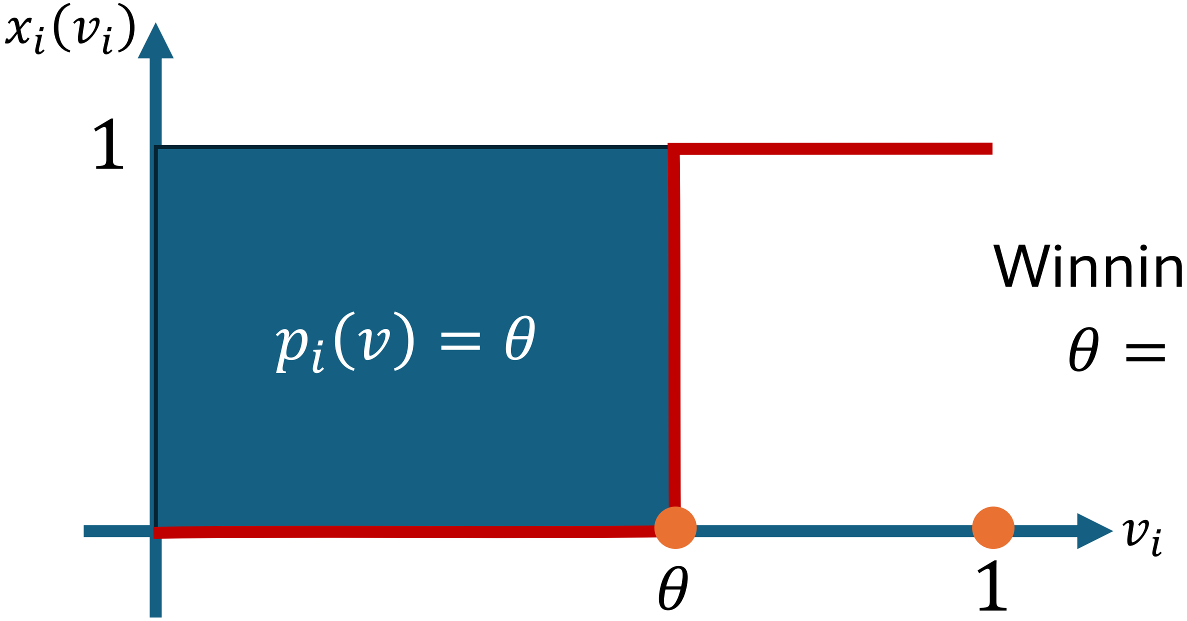
Second Price with Player-Specific Reserves

- Suppose we are given a set of samples S of n bidder value profiles
- Optimize over the space of Second-Price with Player-Specific Reserves
- For every price vector $r = (r_1, \dots, r_n)$ we want to find a vector $\tilde{r}$ that achieves almost the same revenue as r for every value in the samples
$$\forall v_i = (v_{i1}, \dots, v_{in}) \in S\colon\ |\mathrm{rev}(v_i; r) - \mathrm{rev}(v_i; \tilde{r})| \leq \epsilon$$
- For every r_j, pick maximum of {largest multiple of ϵ below r, largest sampled value for bidder j below r}. At most $(m + 1/\epsilon)^n$ prices.

$$\mathrm{Rad}(S, H) \leq 2\sqrt{\frac{2n\log(m + 1/\epsilon)}{m}} + 2\epsilon \underset{\substack{\uparrow \\ \epsilon = 1/m}}{\leq} 4\sqrt{\frac{2n\log(2m)}{m}}$$

Competing with the Myerson Auction

- Want to optimize over virtual welfare maximizing mechanisms
- For each bidder i, we assign a monotone virtual value function ϕ_i
- Allocate to the bidder with highest positive virtual value $\phi_i(v_i)$
- Charge dominant strategy truthful payments

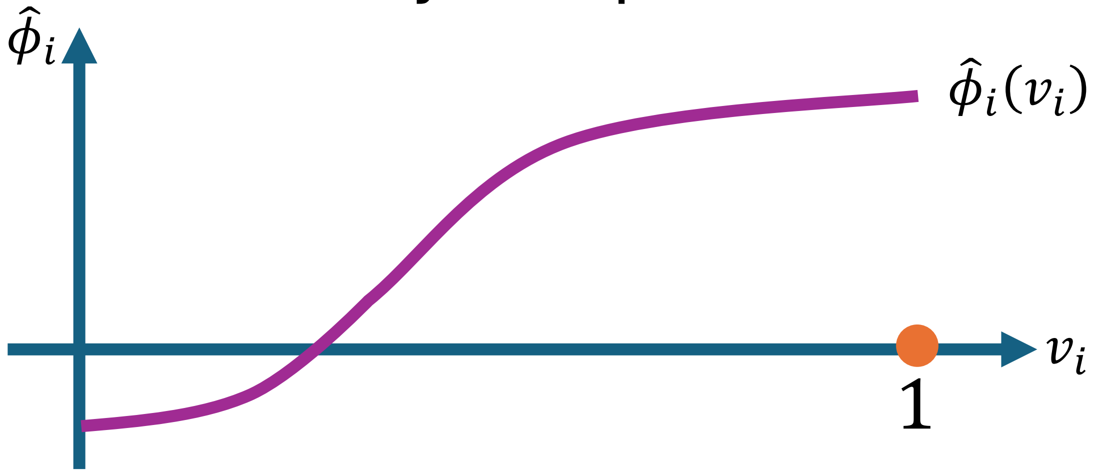


Winning threshold:

$$\theta = \phi_i^{-1}\left(\max_{j \neq i} \phi_j(v_j), \eta_i\right)$$

Optimizing over Virtual Value Functions

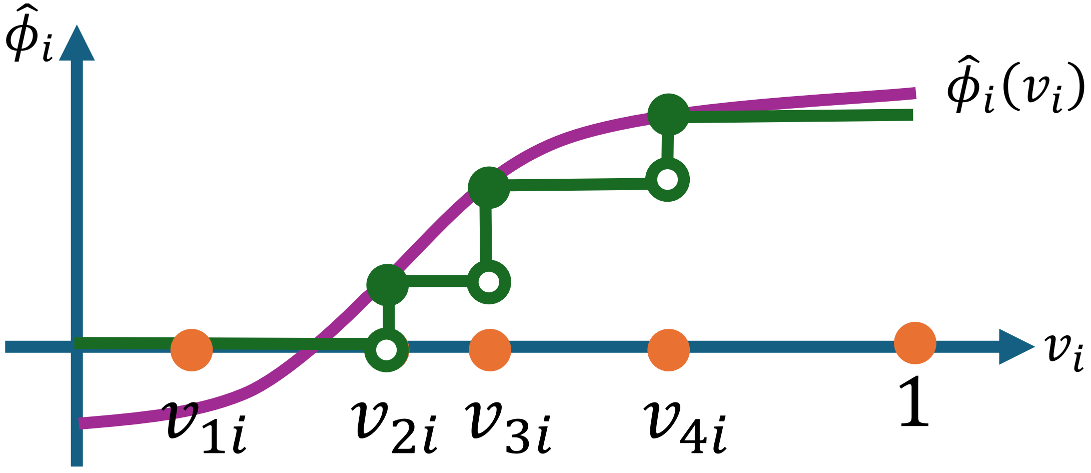
- ERM optimizes over all monotone functions for each bidder
- This space is infinite and a bit harder to discretize
- We will see that monotonicity is important!



- We introduce a variant of Rademacher complexity analysis that will help us in the analysis of ERM over virtual welfare maximizers

Optimizing over Virtual Value Functions

- ERM optimizes over all monotone functions for each bidder
- For any monotone function, we receive strictly larger payment had we used step-function on the samples (threshold to win is higher)!



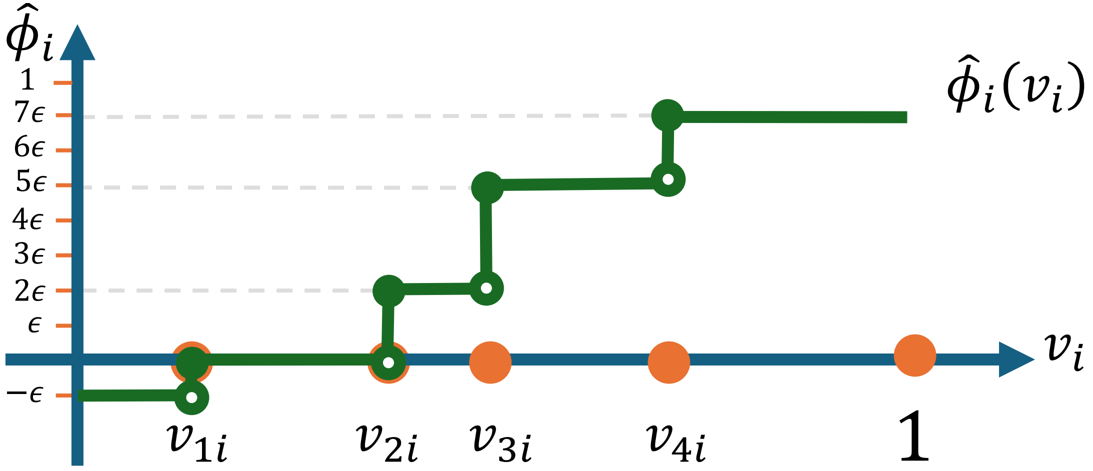
Samples of bidder i values

- H_Q contains only monotone step functions that change on one of the $2m$ samples for each bidder

Coarsen Space of Mechanisms we Optimize

- Consider only virtual value functions that take values on an ϵ-grid

$$\phi_i(v_i) \in \{-\epsilon, 0, \epsilon, \dots, 1\}$$



- These step functions in H_Q can be described by

  "for each value r on the grid, specify the smallest of the $2m$ sampled values for which the rank of the bidder goes above r"

- These are $\approx (2m)^{\frac{1}{\epsilon}}$ combinations for each player

Putting it all together

- If we output the mechanism h_S that optimizes the empirical revenue among all monotone virtual welfare maximizers, with virtual value functions taking values in an ϵ-grid

$$E_S[\text{Rev}(h_\epsilon)] \gtrsim \text{Rev}(h_*) - \sqrt{\frac{2\ \text{nlog}(2m)}{\epsilon \cdot m}} - \epsilon$$

- For $\epsilon = \left(\frac{2n\log(2m)}{m}\right)^{\frac{1}{3}}$

$$E_S[\text{Rev}(h_\epsilon)] \gtrsim \text{Rev}(h_*) - 2\left(\frac{2n\log(2m)}{m}\right)^{\frac{1}{3}}$$

Computational Game Theory for Complex Games

1

- Basics of game theory and zero-sum games (T)
- Basics of online learning theory (T)
- Solving zero-sum games via online learning (T)
- *HW1: implement simple algorithms to solve zero-sum games*
- Applications to ML and AI (T+A)
- *HW2: implement boosting as solving a zero-sum game*

2

- Basics of extensive-form games
- Solving extensive-form games via online learning (T)
- *HW3: implement agents to solve very simple variants of poker*

3

- General games, equilibria and online learning (T)
- Online learning in general games (T+A)
- *HW4: implement no-regret algorithms that converge to correlated equilibria in general games*

Data Science for Auctions and Mechanisms

4

- Basics and applications of auction theory (T+A)
- Basic Auctions and Learning to bid in auctions (T)
- *HW5: implement bandit algorithms to bid in ad auctions*

5

- Optimal auctions and mechanisms (T)
- Simple vs optimal mechanisms (T)
- *HW6: implement simple and optimal auctions, analyze revenue empirically*

6

- Basics of Statistical Learning Theory (T)
- Optimizing Mechanisms from Samples (T)
- *HW7: implement procedures to learn approximately optimal auctions from historical samples*

Further Topics

7

- Econometrics in games and auctions (T+A)
- A/B testing in markets (T+A)
- *HW8: implement procedure to estimate values from bids in an auction, empirically analyze inaccuracy of A/B tests in markets*

Guest Lectures

- Mechanism Design for LLMs, Renato Paes Leme, Google Research
- Auto-bidding in Sponsored Search Auctions, Kshipra Bhawalkar, Google Research

Lecture 15

Where do we get these samples from?

Typically, from historical executions of a truthful mechanism

Example: if we had run second price auctions in the past, we can use the bids of the players, in each of these historical auctions, as samples from their values

What if our auction platform is based on a non-truthful auctions?

Example: If we typically run a First Price Auction, now we have historical samples of bids in an FPA. These are not samples of values; bidders submit bids that are much lower than values in an FPA.

How do we go from bids to values?

Econometrics in Games and Auctions

Econometrics in Games and Auctions

- We are given data from actions of players in a game (and potentially auxiliary contextual information about the game)

- Multiple instances were players played the same type of game

- We don’t know the exact utilities of the players in the game

- We want to use the data to learn the parameters of the utilities of the players in the game or the distribution of these parameters

If I know the equilibrium bid distribution G, then whenever *I see a bid b_i*, I can *reverse engineer* and *uniquely determine the value* that led to such a bid

unobserved value

$$\underbrace{v_i}_{\text{unobserved value}} = \underbrace{b_i}_{\text{observed equilibrium bid}} + \frac{1}{(n-1)\frac{g(b_i)}{G(b_i)}}$$

Reverse hazard ratio of distribution of bids "Probability that opponent bid is immediately below b_i given that it is below b_i"

observed equilibrium bid

More competition ⇒ less "value reduction"

Estimating CDFs and PDFs of Bids from FPA Bid Samples

Given bids $b_1, \dots, b_m$ of players in instances of First Price Auction the CDF and PDF of the *bid distribution* can be approximated by empirical CDF and a Kernel Density Estimate

$$G(z) \overset{\text{def}}{=} \Pr(b < z) \approx \frac{1}{n \cdot m} \sum_{i,j} 1\{b_{ij} < z\} \overset{\text{def}}{=} \hat{G}(z)$$

$$g(z) = \partial_z G(z), \qquad \hat{g}(z) = \frac{1}{n \cdot m} \sum_{i,j} \frac{1}{h_n} K\left(\frac{b_{ij} - z}{h_n}\right)$$

Fraction of samples that ≈lie within h from z, divided by region length

Estimating CDFs and PDFs of Values from FPA Bid Samples

Given bids $b_1, \dots, b_m$ of players in instances of First Price Auction the CDF and PDF of the *value distribution* can be approximated using the plug-in approach, by approximately "inverting the bid" and using the "recovered value as a truthful sample"

$$\hat{v}_{ij} = b_{ij} + \frac{\hat{G}\left(b_{ij}\right)}{(n-1)\,\hat{g}\left(b_{ij}\right)}$$

$$\hat{F}(z) \overset{\text{def}}{=} \frac{1}{n \cdot m} \sum_{i,j} 1\{\hat{v}_{ij} < z\}, \qquad \hat{f}(z) = \frac{1}{n \cdot m} \sum_{i,j} \frac{1}{h_n} K\left(\frac{\hat{v}_{ij} - z}{h_n}\right)$$

Example 2: Econometrics in Entry Games

- Two firms deciding whether to enter a market
- Example: airline firms deciding whether to enter a particular route
- Observe entry decisions $y_i \in \{0, 1\}$ for different markets with characteristics x
- Each firm has profits from entering

$$
\begin{aligned}
\pi_1 &= \boxed{\begin{matrix} x^\top \beta_1 \\ x^\top \beta_2 \end{matrix}} + \boxed{\begin{matrix} y_2 \delta_1 \\ y_1 \delta_2 \end{matrix}} + \boxed{\begin{matrix} \epsilon_1 \\ \epsilon_2 \end{matrix}} \\
\pi_2 &
\end{aligned}
$$

Private costs or payoff shocks $\epsilon_i \sim F_i$ **known only by player i**

effect of market characteristics

effect of competition

- Learn parameters β, δ

Key Idea: Two Stage Estimation

Two-Stage Estimation Approach

[Hotz-Miller’93, Bajari-Benkard-Levin’07, Pakes-Ostrovsky-Berry’07, Aguirregabiria-Mira’07, Bajari-Hong-Chernozhukov-Nekipelov’09]

1. Compute non-parametric estimate $\hat{\sigma}_i(x)$ of function $\sigma_i(x)$ from data

2. Run parametric regressions for each agent individually using that:

$$\sigma_i(x) \propto \exp[x \cdot \beta_i + \hat{\sigma}_{-i}(x)\ \delta_i]$$

3. The latter is a simple logistic regression for each player to estimate β_i, δ_i

Lecture 16

What if all we want is to compare between auctions A and B in terms of revenue?

What I could potentially do is:
For each auction flip a coin;
If heads, then run auction A else run auction B

After many auctions compare average revenue from A auctions, vs., average revenue from B auctions

We will see that it can be problematic and needs thought of how to analyze such data or structure such A/B tests!

Interference

- Social Network interference
- Equilibrium effects
- Stateful systems and time effects

A/B Testing over Position Auction Formats

We observe a bid distribution, described by the quantile function $b(q)$, from a randomized k-unit auction (which chooses each k with positive probability)

For any other randomized k-unit (with probabilities w_k) first-price auction among symmetric bidders, we have:

$$\mathrm{Rev} = n \sum_{k \leq N} w_k \, E[b(q) \cdot f(q)]$$

for a function $f(q)$ known in closed form

With access to bidding data from a single randomized k-unit auction (which chooses each k with positive probability), we can estimate Rev of any other k-unit auction.

Estimate CDF of bids using the empirical CDF $\hat{G}$.
Then use $\hat{b} = \hat{G}^{-1}$ and

$$\widehat{\mathrm{Rev}} = n \sum_{k \leq N} w_k \int_0^1 \hat{b}(q) \cdot f(q) dq$$

By convergence rates of empirical CDF, we can show:

$$\left|\widehat{\mathrm{Rev}} - \mathrm{Rev}\right| \lesssim 1/\sqrt{m}$$

What we did not learn!

- Monte-Carlo tree search
- Neural network approximation of values
- Multi-agent RL
- Budgets in auctions
- Correlated values in auctions
- A/B testing for pricing and equilibrium effects
- Econometrics in complete info games and partial identification

Course Learning Objectives

- Learn the fundamentals of game theory
- Learn how game theory can be applied in many real-world settings (e.g. ad auctions, complex games)
- Learn the fundamentals of tools from data science and ML that are useful in game theoretic contexts (online learning theory, statistical learning theory, econometrics)
- Learn how these topics can be combined to
  - provide computational solutions to the design of agents that perform well in competitive environments
  - optimize and analyze markets, mechanisms and platforms from data
- Be able to implement and code up these solutions in Python

Course Evaluations

http://course-evaluations.stanford.edu/