MS&E 233
Game Theory, Data Science and AI
Lecture 4

Vasilis Syrgkanis

Assistant Professor

Management Science and Engineering

(by courtesy) Computer Science and Electrical Engineering

Institute for Computational and Mathematical Engineering

Class Music Auction!

We will be experimenting with putting music for the first three minutes of the class as people arrive!

You have the chance to choose the song of the day!

Each of you has a total budget of 100 fake dollars for the whole class! You can choose to spend them however you want on each lecture.

For each lecture you can choose to bid anywhere from 0 to 20 dollars.

We will then choose uniformly at random among the highest bidders. The winner of the auction will get to choose the song of the day and they have to pay their bid, i.e. the amount they bid will be subtracted from their 100$ budget.

If you submit an illegal bid (i.e. a bid that goes beyond your total budget, your bid will be disqualified and ignored).

Please be appropriate in your choice of songs; I might need to censor and ask you to choose something else. I'll be emailing the winner on the morning of the lecture to email me the spotify link for the song.

Submit your bid by 11:59pm the day before the lecture. You should submit your bid using the corresponding canvas quiz that will be setup for each lecture

Class Music Auction: Game Theory, Data Science and AI (stanford.edu)

Go to canvas and check the quizzes section.

If there is no participation in the auction, I'll just choose the music myself. But that's not much fun...

Spotify playlist that will be populated with the songs we play each day: https://open.spotify.com/playlist/03yGb6URnCzG4pVV6RhK4C?si=wpINDMSGRJOho_6daaSLsA&pt=ff706933952e0f64d8f8b797368a83ed

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

3
- General games and equilibria (T)
- Online learning in general games, multi-agent RL (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)
- Learning to bid in auctions via online learning (T)
- *HW5: implement bandit algorithms to bid in ad auctions*

5
- Optimal auctions and mechanisms (T)
- Simple vs optimal mechanisms (T)
- *HW6: calculate equilibria in simple auctions, implement simple and optimal auctions, analyze revenue empirically*

6
- Optimizing mechanisms from samples (T)
- Online optimization of auctions and mechanisms (T)
- *HW7: implement procedures to learn approximately optimal auctions from historical samples and in an online manner*

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

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

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

Boosting

Image credits: chat.openai.com

The Boosting Problem

- Given n samples $\mathrm{S} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ from distribution D
- $y_i \in \{0,1\}$: a binary classification label
- x_i: covariates associated with y_i that can be used to predict label

- Suppose that I give you a weak classification "Oracle" algorithm
- Oracle looks at samples S and produces hypothesis h_S from a simple class of base hypotheses H that classifies more than half of the samples correctly

$$\frac{1}{n}\sum_{i=1}^{n} 1\{h(x_i) = y_i\} \geq \frac{1}{2} + \delta$$

The Boosting Problem

- Given n samples $\mathrm{S} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ from distribution D
- $y_i \in \{0,1\}$: a binary classification label
- x_i: covariates associated with y_i that can be used to predict label

- Suppose that I give you a weak classification "Oracle" algorithm
- Oracle looks at weighted samples S and produces hypothesis $h_S \in H$ that classifies more than half of the weight correctly

$$\frac{1}{\sum_i w_i} \sum_{i=1}^{n} w_i 1\{h(x_i) = y_i\} \geq \frac{1}{2} + \delta$$

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*

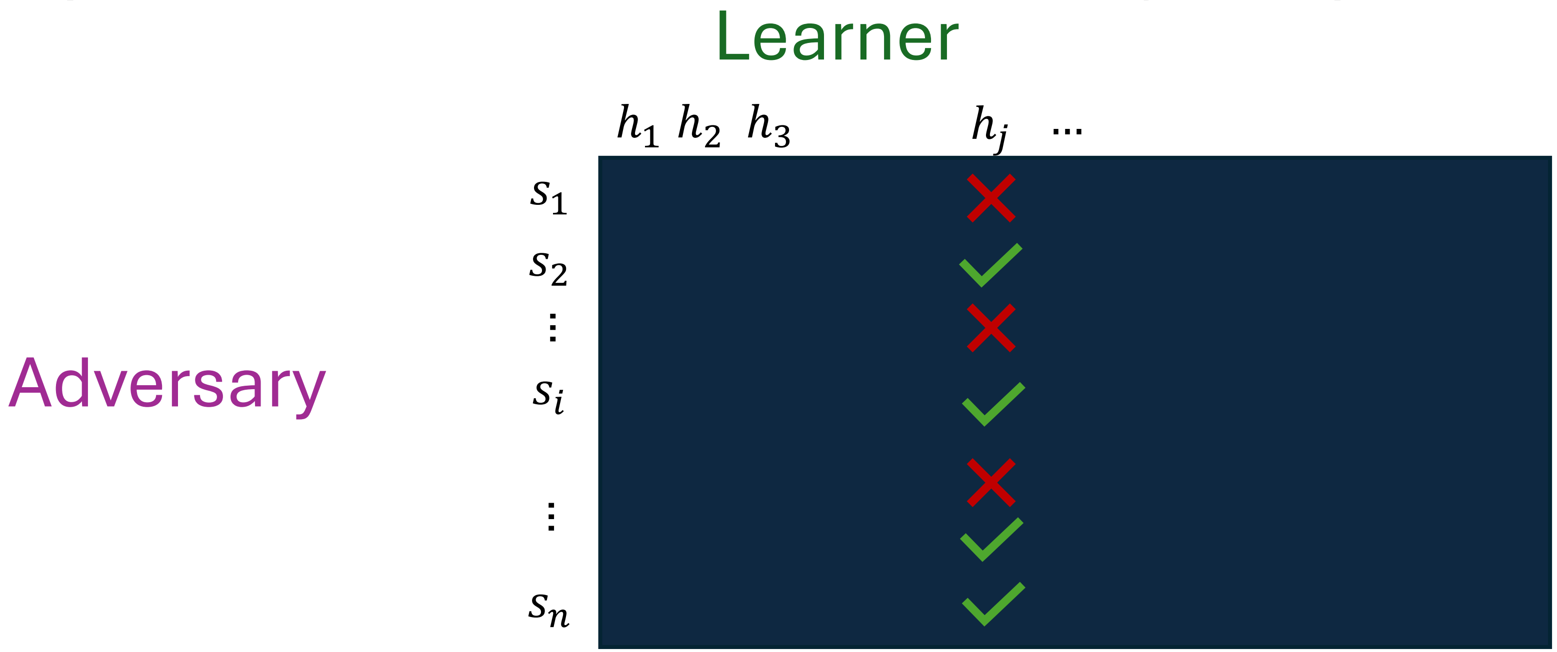
Resolved by Robert Schapire and further developed by Freund-Schapire

Finite Sample Variant. Given “weak” classification oracle, can we construct in a computationally efficient manner a “strong” classifier that classifies all samples correctly and is not much more complex than functions in H

Can then be combined with generalization bound arguments (VC dimension), which we will learn later in class, to fully solve the boosting problem.

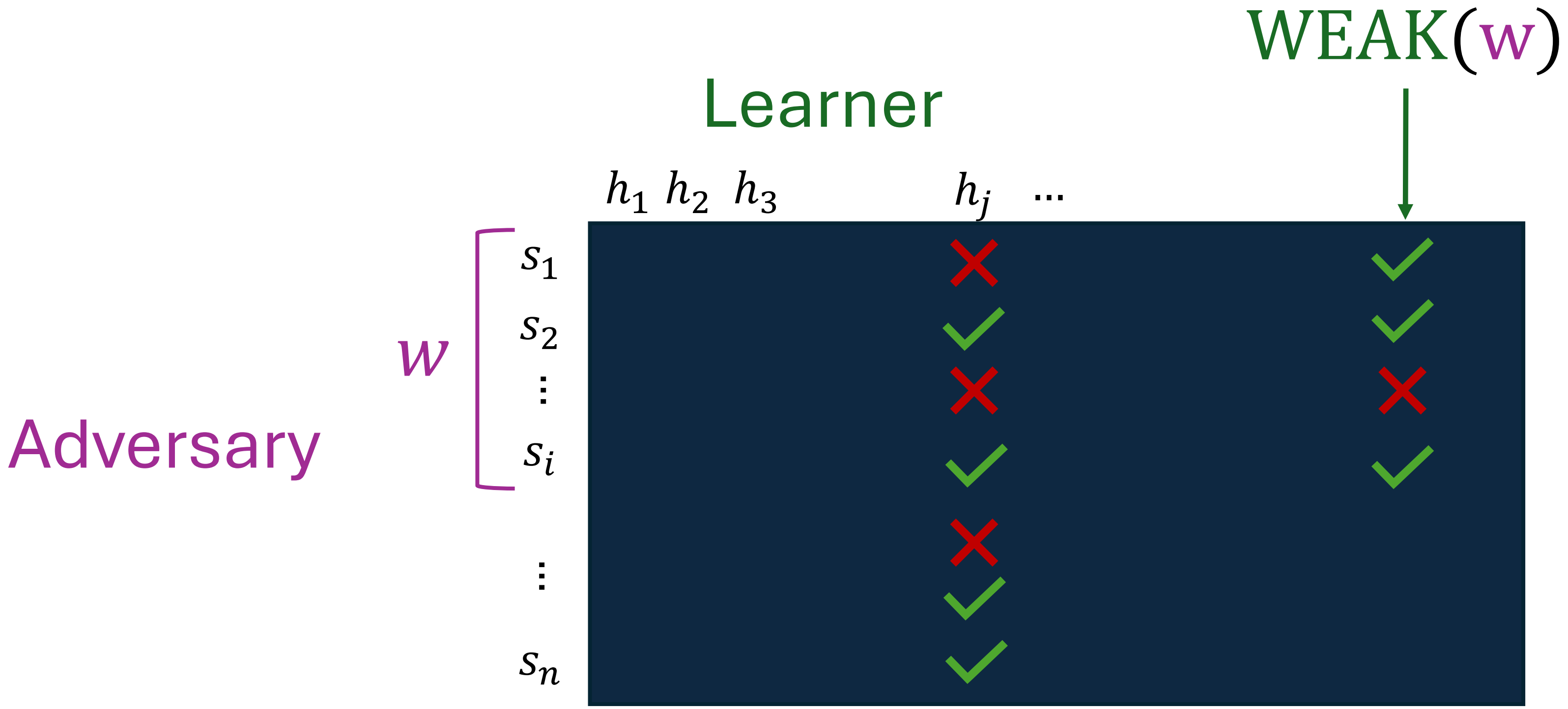
Skyline View of the Solution as a Game

- View the problem as a game between a learner and an adversary
- Adversary chooses distributions w over samples on which the learner makes many mistakes
- Learner chooses distribution P over hypotheses with small expected number of mistakes for any sample distribution



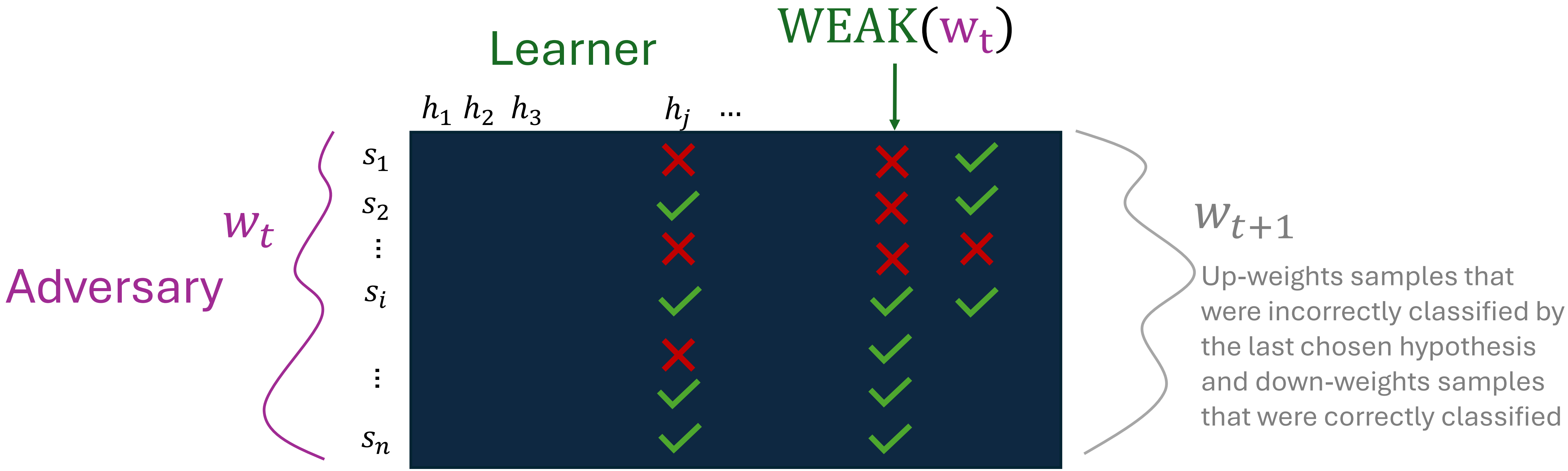
Skyline View of the Solution as a Game

- View the problem as a game between a learner and an adversary
- Weak Oracle assumption. No matter what distribution w over samples you give me, there exists a hypothesis $\mathrm{WEAK}(\mathrm{w})$ that will classify majority correctly



Skyline View of the Solution as a Game

- View the problem as a game between a learner and an adversary
- We solve the game via no-regret dynamics
- At each period adversary chooses a distribution w_t based on EXP
- Learner "best-responds" by applying the weak oracle on w_t



Skyline View of the Solution as a Game

- The average w_t and the uniform mixture over $\{h_1, \ldots, h_T\}$ are an approximate solution to the min-max problem
- Together with weak oracle assumption this means for any sample the majority of $\{h_1, \ldots, h_T\}$ classify it correctly
- The majority vote ensemble model classifies all samples correctly!

- This is roughly AdaBoost…

`sklearn.ensemble`.AdaBoostClassifier

class `sklearn.ensemble.`**`AdaBoostClassifier`**(*estimator=None*, ***, *n_estimators=50*, *learning_rate=1.0*, *algorithm='SAMME.R'*, *random_state=None*) [source]

An AdaBoost classifier.

An AdaBoost [1] classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.

This class implements the algorithm based on [2].

Read more in the User Guide.

Let’s make it formal!

The Boosting Problem as a Zero-Sum Game

- Produce a hypothesis h_* such that no matter which sample we look, we accurately classify

$$\max_{h \in H} \min_{i \in [n]} 1\{h(x_i) = y_i\}$$

- Max player: learner that chooses hypotheses
- Min player: adversary that chooses bad samples

The Boosting Problem as a Zero-Sum Game

- Produce a hypothesis h_* such that no matter which distribution over samples we look, we accurately classify

$$\max_{h \in H} \min_{w \in \Delta[n]} \sum_{i=1}^{n} w_i 1\{h(x_i) = y_i\}$$

- Max player: learner that chooses hypotheses
- Min player: adversary that chooses bad distributions over samples

Convexifying the Learner

- Produce mixture P_* over base hypotheses, such that no matter which distribution over samples we look, we accurately classify

$$\max_{P \in \Delta(H)} \min_{w \in \Delta[n]} \sum_{i=1}^{n} w_i E_{h \sim P}[1\{h(x_i) = y_i\}]$$

- Max player: learner that chooses distributions over hypotheses
- Min player: adversary that chooses bad distributions over samples

Solution as an Equilibrium

- Produce mixture P_* over base hypotheses, such that no matter which distribution over samples we look, we accurately classify

$$\max_{P\in\Delta(H)} \min_{w\in\Delta[n]} \sum_{i=1}^{n} w_i E_{h\sim P}[1\{h(x_i) = y_i\}] =: \ell(w, P)$$

- Max player: learner that chooses distributions over hypotheses
- Min player: adversary that chooses bad distributions over samples

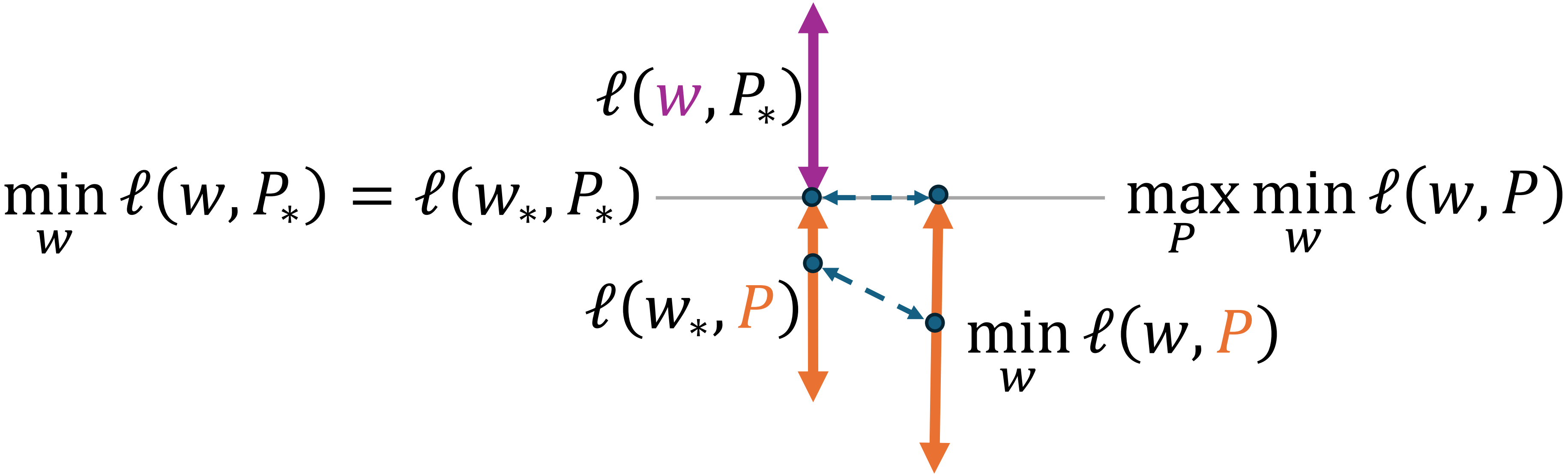
Equilibria vs. Min-Max Solutions

Equilibria and Min-Max Solutions

Theorem. Any equilibrium (w_*, P_*) is a solution to min-max

$$\min_w \ell(w, P_*) \geq \ell(w_*, P_*) \geq \max_P \ell(w_*, P) \geq \max_P \min_w \ell(w, P)$$

Best-Response of w-player   Best-Response of P-player



Equilibria and Min-Max Solutions

Theorem. Any equilibrium (w_*, P_*) is a solution to min-max

$$\min_w \ell(w, P_*) \geq \ell(w_*, P_*) \geq \max_P \ell(w_*, P) \geq \max_P \min_w \ell(w, P)$$

Best-Response of w-player

Best-Response of P-player

Theorem. Any ϵ-equilibrium (w_*, P_*) is a 2ϵ-solution to min-max

ϵ-Best-Response of P-player

$$\min_w \ell(w, P_*) \geq \ell(w_*, P_*) - \epsilon \geq \max_P \ell(w_*, P) - 2\epsilon$$

ϵ-Best-Response of w-player

$$\geq \max_P \min_w \ell(w, P) - 2\epsilon$$

Solving for Equilibrium via No-Regret

We let that two players play the game repeatedly

- At each period, adversary chooses a distribution w_t over samples
- At each period, learner chooses a distribution P_t over hypotheses

We want both players' choice process to have vanishing regret

- Adversary online learning problem easy: n action problem
- Learner's online learning problem not clear…

Solving large games with oracles!

Image credits: chat.openai.com

Zero-Sum with Too Many Actions!

- Min player has a set I of n actions
- Max player has a potentially infinite set of actions J
- Exists well-defined loss $\mathrm{A}(i, j)$ for every choice $i \in I, j \in J$
- If min player chooses a distribution w over I and max player chooses distribution P over J, expected loss is

$$\ell(w, P) = E_{i \sim w, j \sim P}[A(i, j)]$$

- $\ell_j \coloneqq \big(A(1, j), \dots, A(n, j)\big)$ is loss vector for action j of max player

$$\ell(w, P) = w^\top E_{j \sim P}\left[\ell_j\right]$$

Solving with No-Regret Learning

We let players play for T periods

- Suppose we could device choice algorithms that satisfy the vanishing regret property for both players
- We already know that average of solutions of each player, are an equilibrium; hence also a solution to min-max problem

- For the finite action min player, it's easy (e.g. EXP)

- For the infinite action player?

Online Learning with Oracles

- **IDEA.** We just need the sequence to be no-regret; don't need to fully respect the online learning protocol

- **Solution.** Tell max player the choice of the min player before playing and have them choose best-response to it at each period

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

(Best-Response) $$j_t = \operatorname*{argmax}_{P \in \Delta(J)} w_t^\top E_{j \sim P}\left[\ell_j\right] = \operatorname*{argmax}_{j \in J} w_t^\top \ell_j = \mathrm{BR}(w_t)$$

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 $\Rightarrow$ P_* is $2\sqrt{\frac{2\log(n)}{T}}$-solution to max-min.

Back to Boosting

Image credits: chat.openai.com

Boosting as Oracle Based No-Regret in Games

- Weak classification oracle can be viewed as Best-Response oracle
- For every distribution w over the samples, it returns

$$h = \mathrm{WEAK}(w), \qquad \sum_{i=1}^{n} w_i 1\{h(x_i) = y_i\} \geq \frac{1}{2} + \delta$$

- We can use the oracle approach to solving large games
- $\ell_h = (1\{h(x_1) = y_1\}, \dots, 1\{h(x_n) = y_n\})$: accuracy on each sample

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

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

Boosting as Oracle Based No-Regret in Games

- Even if weak oracle not Best-Response oracle, its good enough
- If P_* the uniform distribution over $\{h_1, \ldots, h_T\}$

$$\min_{i \in [n]} \ell(i, P_*) + \epsilon = \min_{w} \frac{1}{T} \sum_{t=1}^{T} \ell(w, h_t) + \epsilon \underset{\text{EXP regret guarantee}}{\geq} \frac{1}{T} \sum_{t=1}^{T} \ell(w_t, h_t) \overset{\text{Weak oracle guarantee}}{\geq} \frac{1}{2} + \delta$$

Boosting as Oracle Based No-Regret in Games

- Even if weak oracle not Best-Response oracle, its good enough
- If P_* the uniform distribution over $\{h_1, \ldots, h_T\}$

Weak oracle guarantee

$$\min_{i \in [n]} \ell(i, P_*) + \epsilon = \min_{w} \frac{1}{T} \sum_{t=1}^{T} \ell(w, h_t) + \epsilon \geq \frac{1}{T} \sum_{t=1}^{T} \ell(w_t, h_t) \geq \frac{1}{2} + \delta$$

EXP regret guarantee

- For T large enough, such that $\epsilon < \delta$, for every sample $i \in [n]$

$$\frac{1}{T} \sum_{t=1}^{T} 1\{h_t(x_i) = y_i\} \geq \frac{1}{2} + \delta - \epsilon > \frac{1}{2}$$

- **Alternatively:** majority of $\{h_1, \ldots, h_T\}$ classify sample correctly

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

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

Then the following majority classifier classifies all samples correctly

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

Distributional Robustness

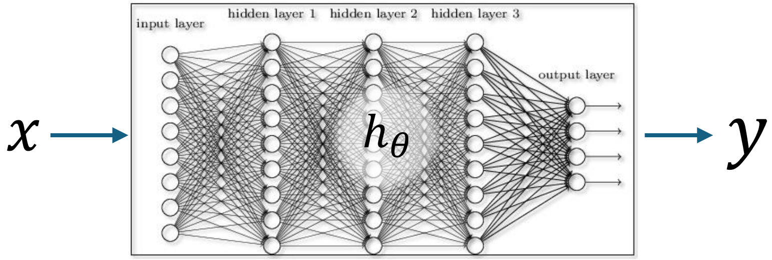
Image credits: chat.openai.com

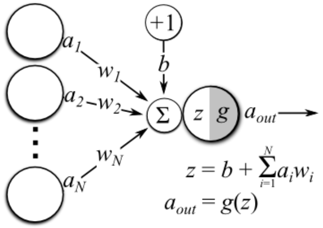
Robust Models to Distribution Shifts

Supervised learning. Given samples $(x, y) \sim D$ learn model h_θ

$$\min_{\theta \in \Theta} E_{(x,y)\sim D}\left[\ell\left(y, h_\theta(x)\right)\right]$$

Example. θ are weights of neural network with output $h_\theta(x)$





Loss is mean squared prediction error

$$\min_{\theta \in \Theta} E_{(x,y)\sim D}\left[\left(y - h_\theta(x)\right)^2\right]$$

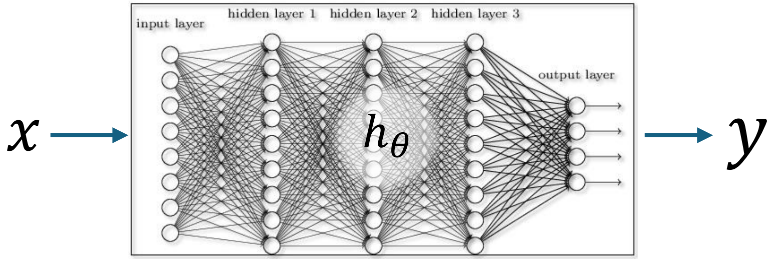
Image credits: chat.openai.com

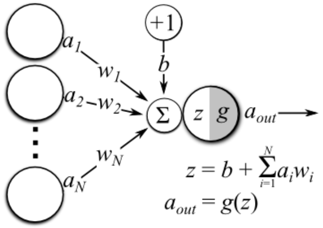
Robust Models to Distribution Shifts

Supervised learning. Given samples $(x, y) \sim D$ learn model h_θ

$$\min_{\theta \in \Theta} E_{(x,y)\sim D}\left[\ell(y, h_\theta(x))\right]$$

Example. θ are weights of neural network with output $h_\theta(x)$





Loss is mean squared prediction error

$$\min_{\theta \in \Theta} E_{(x,y)\sim D}\left[\left(y - h_\theta(x)\right)^2\right]$$

Problem

- Data are typically biased and have under-represented groups
- Model trained to optimize average performance on training data
- Can perform poorly on sub-groups!

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

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

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…

Group DRO as a Zero-Sum Game

[1707.01047] Robust Optimization for Non-Convex Objectives (arxiv.org)

- The learner player chooses $\theta \in \Theta$
- The adversary player chooses a distribution w_t over G
- **Alternative.** We have oracle (*Stochastic Gradient Descent*) that finds neural network h_θ with small loss on distribution Q when trained on samples from Q

Group DRO as a Zero-Sum Game

[1707.01047] Robust Optimization for Non-Convex Objectives (arxiv.org)

- The learner player chooses $\theta \in \Theta$
- The adversary player chooses a distribution w_t over G
- **Alternative.** We have oracle (*Stochastic Gradient Descent*) that finds neural network h_θ with small loss on distribution Q when trained on samples from Q
- Solve via no-regret vs. best-response

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

(Best-Response) $$h_t = \mathrm{Oracle}\left(\sum_g w_t^g D_g\right)$$

- Return mixture (ensemble) of neural networks $\{h_1, \dots, h_T\}$
- Provably a solution to min-max, assuming loss ℓ convex in $h_{t-1}(x)$, not θ!

Robust Models to Adversarial Attacks

[1707.01047] Robust Optimization for Non-Convex Objectives (arxiv.org)

- **Supervised learning.** given samples $(x, y) \sim D$ learn model θ

$$\min_{\theta \in \Theta} E_{(x,y)\sim D}[\ell(x, y; \theta)]$$

- Adversaries can try to locally corrupt the samples to fool model
- We have in mind a set of adversarial attacks A, which alter the distribution D, i.e. D_a formed by sampling $(x, y) \sim D$ and applying corruption $a(x, y) = (\tilde{x}, \tilde{y})$

- Mathematically same problem as group DRO:

$$\min_{\theta \in \Theta} \max_{a \in A} E_{(x,y)\sim D_a}[\ell(x, y; \theta)]$$

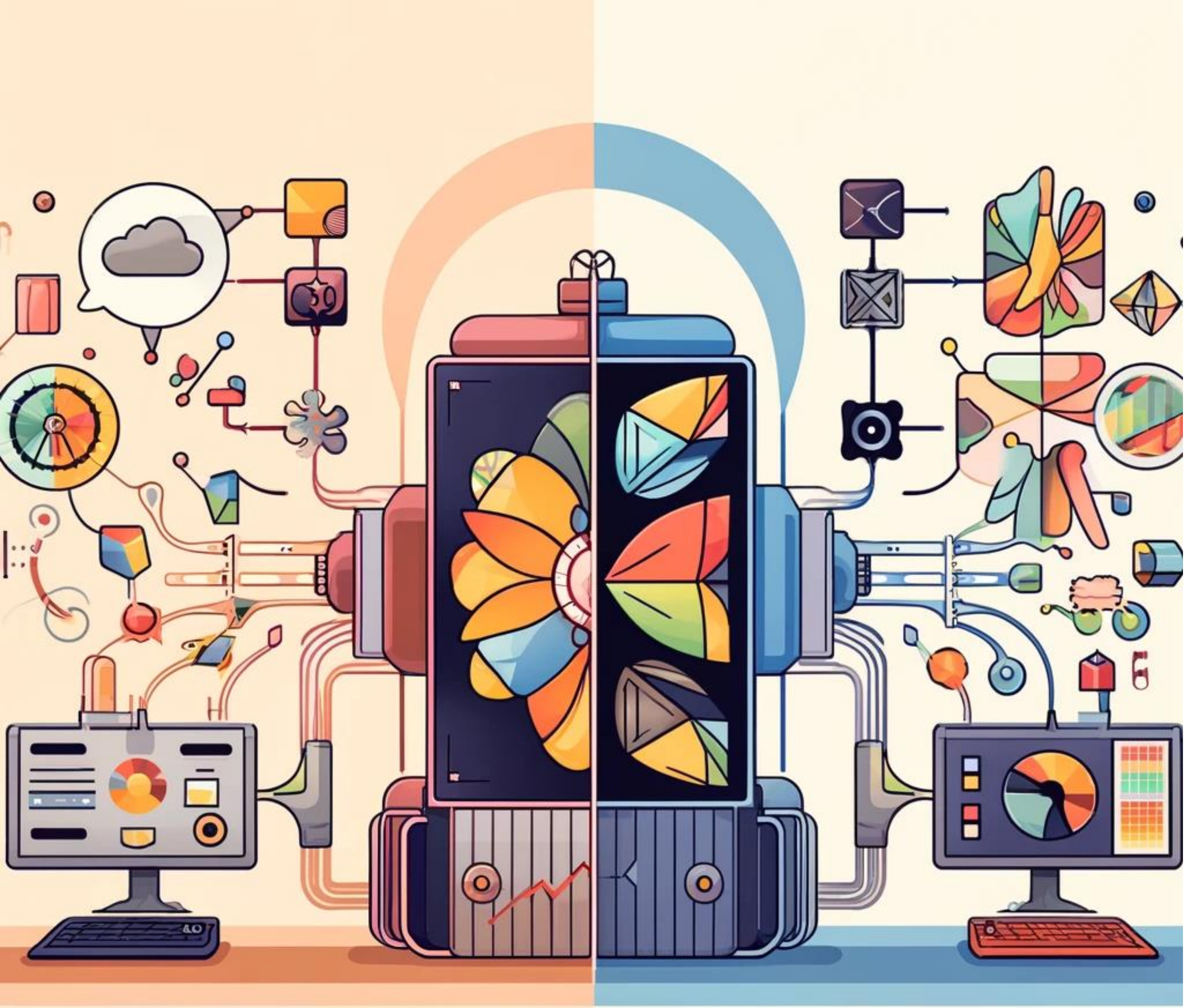
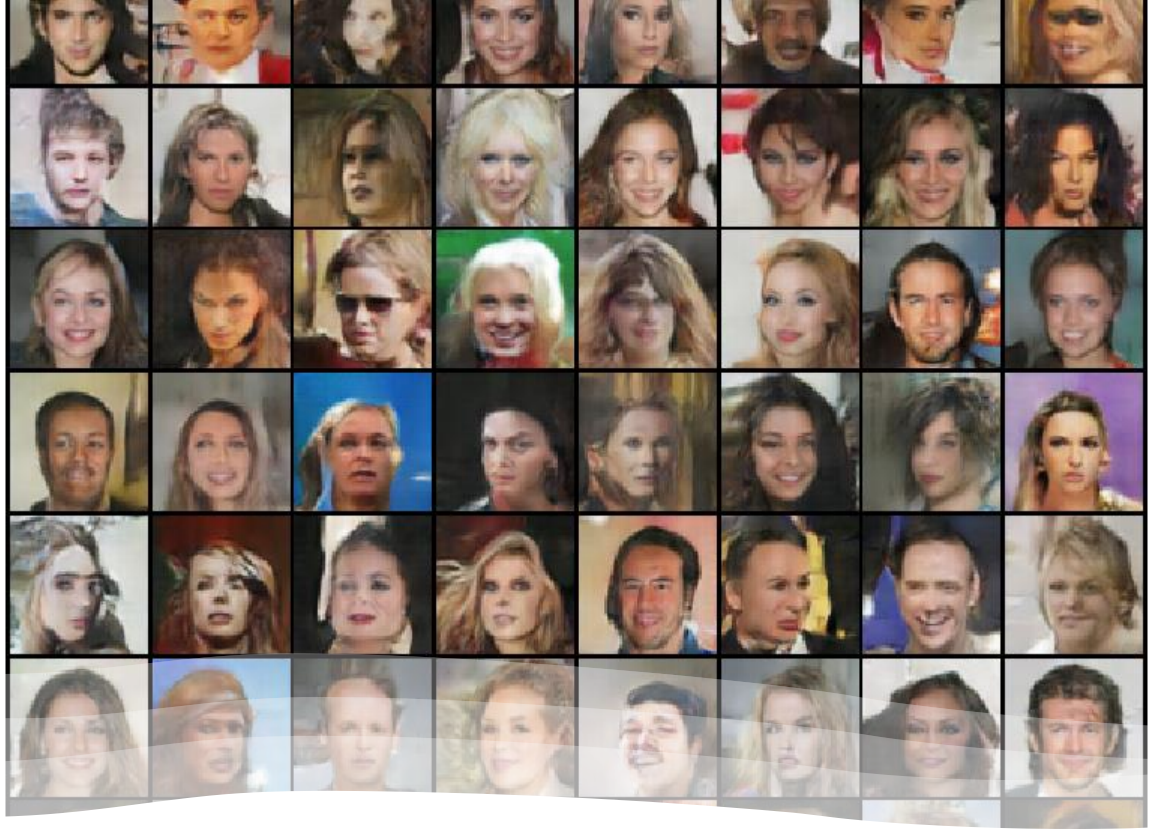


Image credits: chat.openai.com

Generative Adversarial Networks

Given samples $z_1, \ldots, z_n$ from a distribution D



Be able to generate new samples from D

Generative Modelling

Image credits: https://medium.datadriveninvestor.com/artificial-intelligence-gans-can-create-fake-celebrity-faces-44fe80d419f7

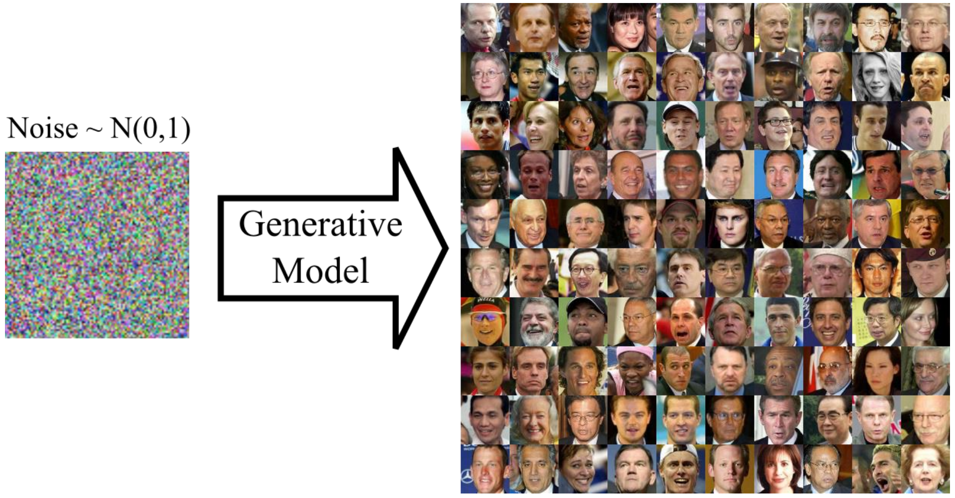
Generative Modelling

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

Given samples $z_1, \ldots, z_n$ from some distribution D

Goal. Be able to generate new samples from D

- Learn a neural sample generator



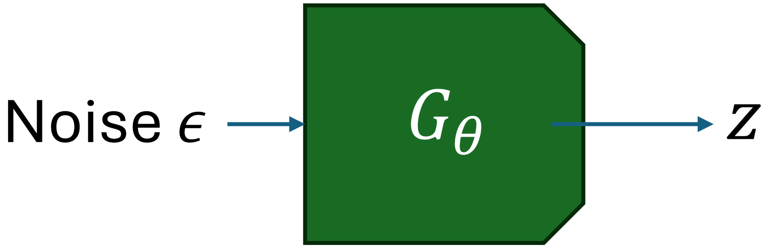
Generative Modelling

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

Given samples $z_1, \ldots, z_n$ from some distribution D

Goal. Be able to generate new samples from D

- Learn a neural sample generator

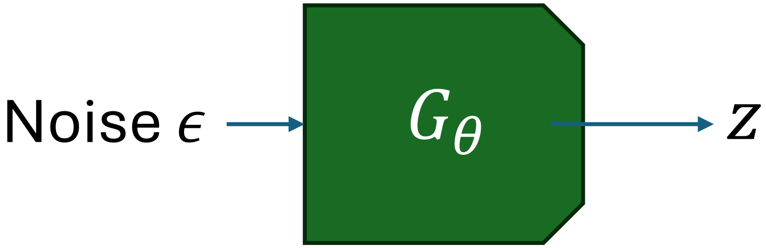


- How do we train parameter θ, so that $G_\theta(\epsilon) \approx$ distributed as D?

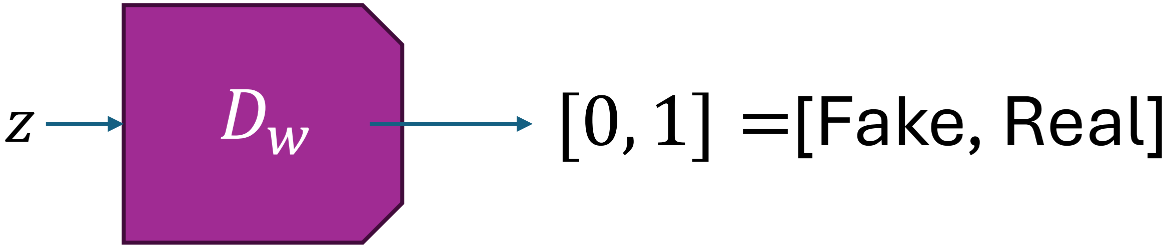
Generative Adversarial Networks (GANs)

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

- Learn a neural sample generator



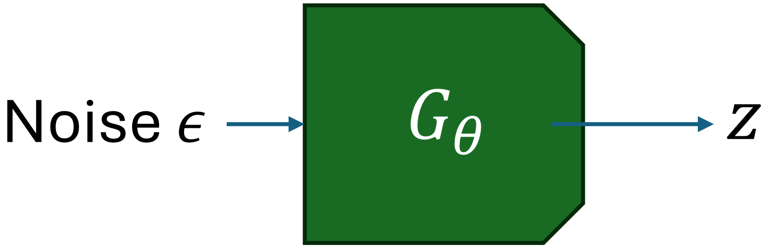
- Learn a discriminator



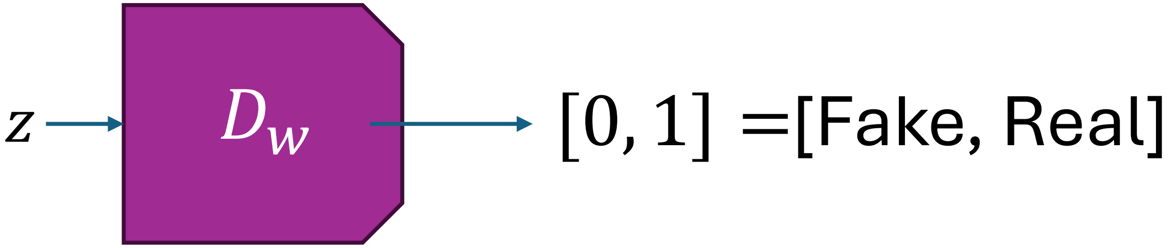
Generative Adversarial Networks (GANs)

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

- Learn a neural sample generator



- Learn a discriminator



- Discriminator wants to minimize classification error

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

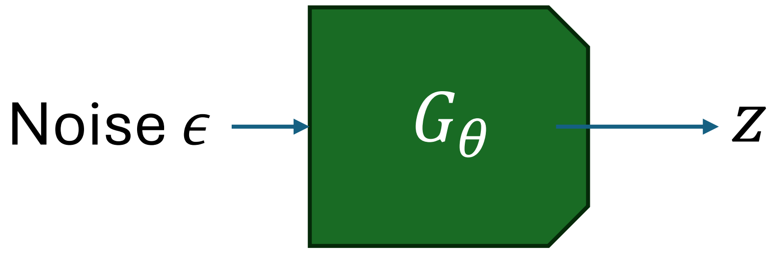
$D_w(z)$ close to 1 when z is real

$D_w(z)$ close to 0 when fake

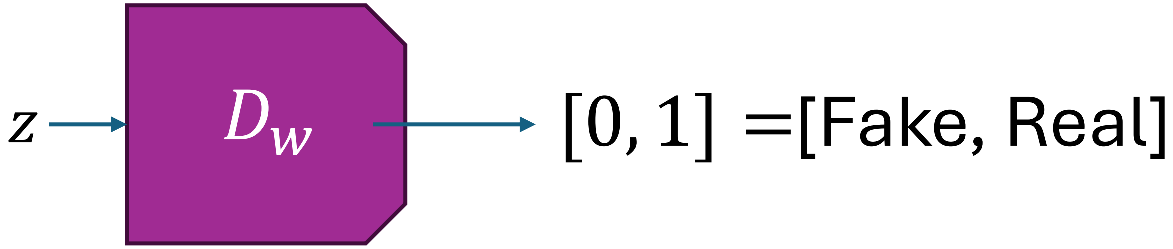
GANs as a Zero-Sum Game

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

- Learn a neural sample generator (max player)



- Learn a discriminator (min player)



- 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

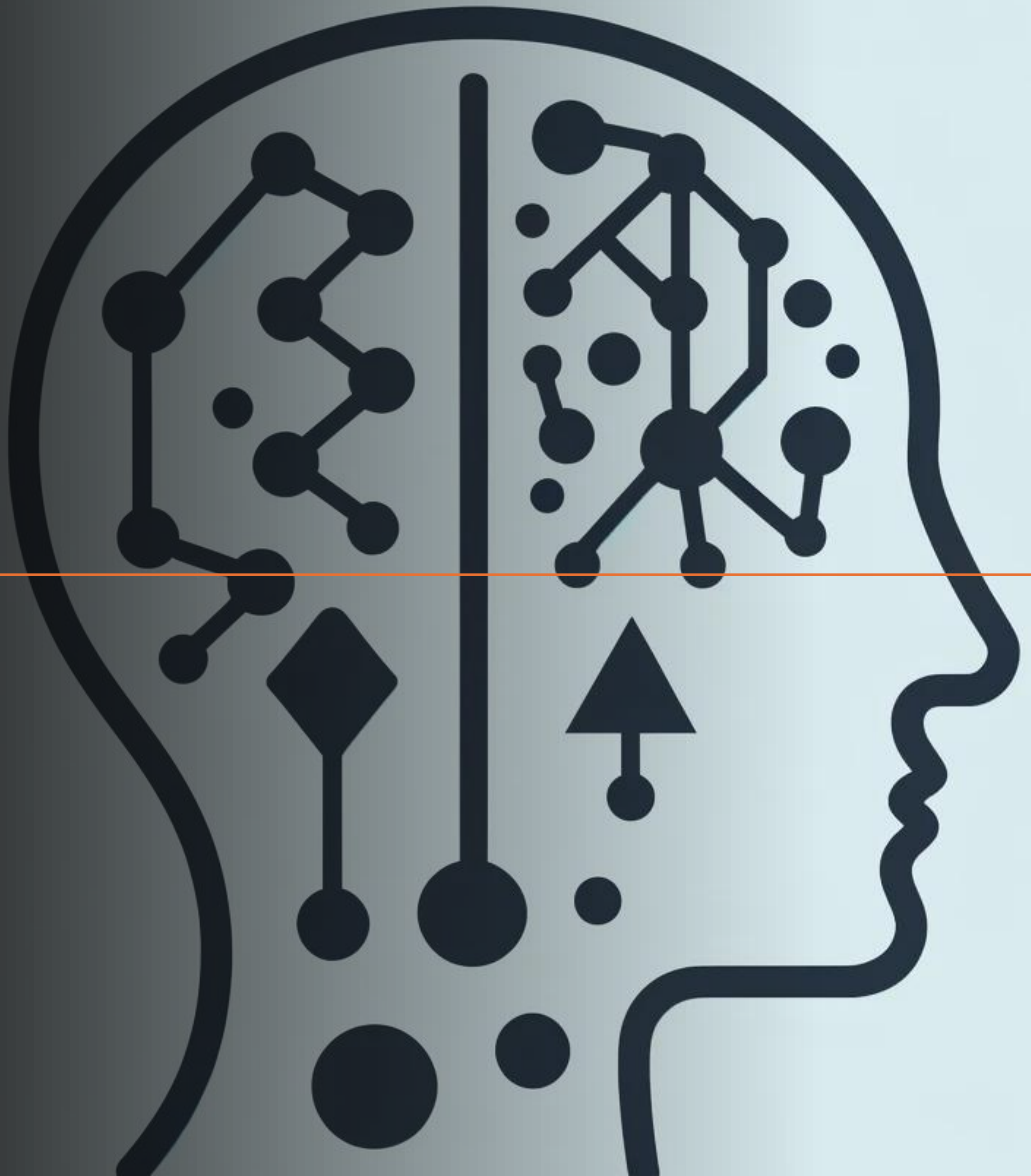


Image credits: chat.openai.com

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 $\pi' > \pi$

Reward Based Approaches

- Most RLHF methods assume preference data are noisy versions of a reward model

- Implies that preference function is inline with a scalar reward function $r: \Pi \rightarrow R$
$$P(\pi, \pi') > 0 \Leftrightarrow r(\pi) > r(\pi')$$

- If population has heterogeneous preferences, preference function P might not be consistent with any reward model!

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

Reward Based Approaches

- Most RLHF methods assume preference data are noisy versions of a reward model
- Implies that preference function is inline with a scalar reward function $r: \Pi \rightarrow R$

$$P(\pi, \pi') > 0 \Leftrightarrow r(\pi) > r(\pi')$$

- If population has heterogeneous preferences, preference function P might not be consistent with any reward model!

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

Intransitive Preferences
$a > c > d > a$

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

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

- Such a p is known as the Minimax Winner

- Does a Minimax Winner (MW) exist?

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

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

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

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

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

Anti-Symmetric Zero-Sum Games

$$\ell(p, p') = -E_{\pi \sim p, \pi' \sim p'}[P(\pi, \pi')]$$

Theorem. (p, q) is equilibrium $\Rightarrow$ (p, p) is equilibrium

- Anti-symmetry of $P \Rightarrow \ell(p, p') = -\ell(p', p)$
- Value of any equilibrium is zero (player can deviate to opponent choice)

$$0 = \ell(p, p) \leq \ell(p, q) \leq \ell(q, q) = 0$$

Anti-Symmetric Zero-Sum Games

$$\ell(p,p') = -E_{\pi\sim p,\pi'\sim p'}[P(\pi,\pi')]$$

Theorem. (p,q) is equilibrium $\Rightarrow$ (p,p) is equilibrium

- Anti-symmetry of $P \Rightarrow \ell(p,p') = -\ell(p',p)$
- Value of any equilibrium is zero (player can deviate to opponent choice)

$$0 = \ell(p,p) \leq \ell(p,q) \leq \ell(q,q) = 0$$

- Best response constraints for q and anti-symmetry imply

$$0 \geq \max_{p'} \ell(p,p') = -\min_{p'} \ell(p',p)$$

- Which directly imply best response constraints for (p,p)

$$\ell(p,p) = \boxed{0 \geq \max_{p'} \ell(p,p')} \qquad \text{and} \qquad \ell(p,p) = 0 \leq \min_{p'} \ell(p',p)$$

Exact definition of MW: $\min_{p'} E_{\pi\sim p,\pi'\sim p'}[P(\pi,\pi')] \geq 0 \Leftrightarrow \min_{p'} -\ell(p,p') \geq 0 \Leftrightarrow \max_{p'} \ell(p,p') \leq 0$

Learning in Anti-Symmetric Games

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

- It suffices to only use one no-regret learning algorithm!
- We use the choice of the algorithm as the choice of the adversary
- By anti-symmetry, this will be no-regret for the adversary too
- For instance, when we have finitely many policies Π

$$p_{t+1}(\pi) \propto p_t(\pi) \exp\big(-\ell_t(\pi)\big), \qquad \ell_t(\pi) = -E_{\pi' \sim p_t}[P(\pi, \pi')]$$

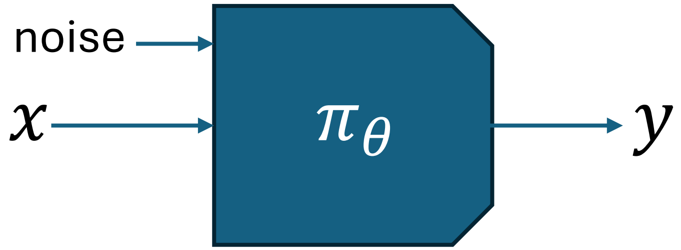
- More generally:

$$p_{t+1} = \mathrm{NoRegret}(\ell_{1:t}), \qquad \ell_t(p) = -E_{\pi \sim p, \pi' \sim p_t}[P(\pi, \pi')]$$

- Return mixture of $p_1, \ldots, p_T$ as final distribution

Self-Play Preference Optimization; SPO

- Practical neural training algorithm that avoids adversarial training
- Represent policy by a neural network



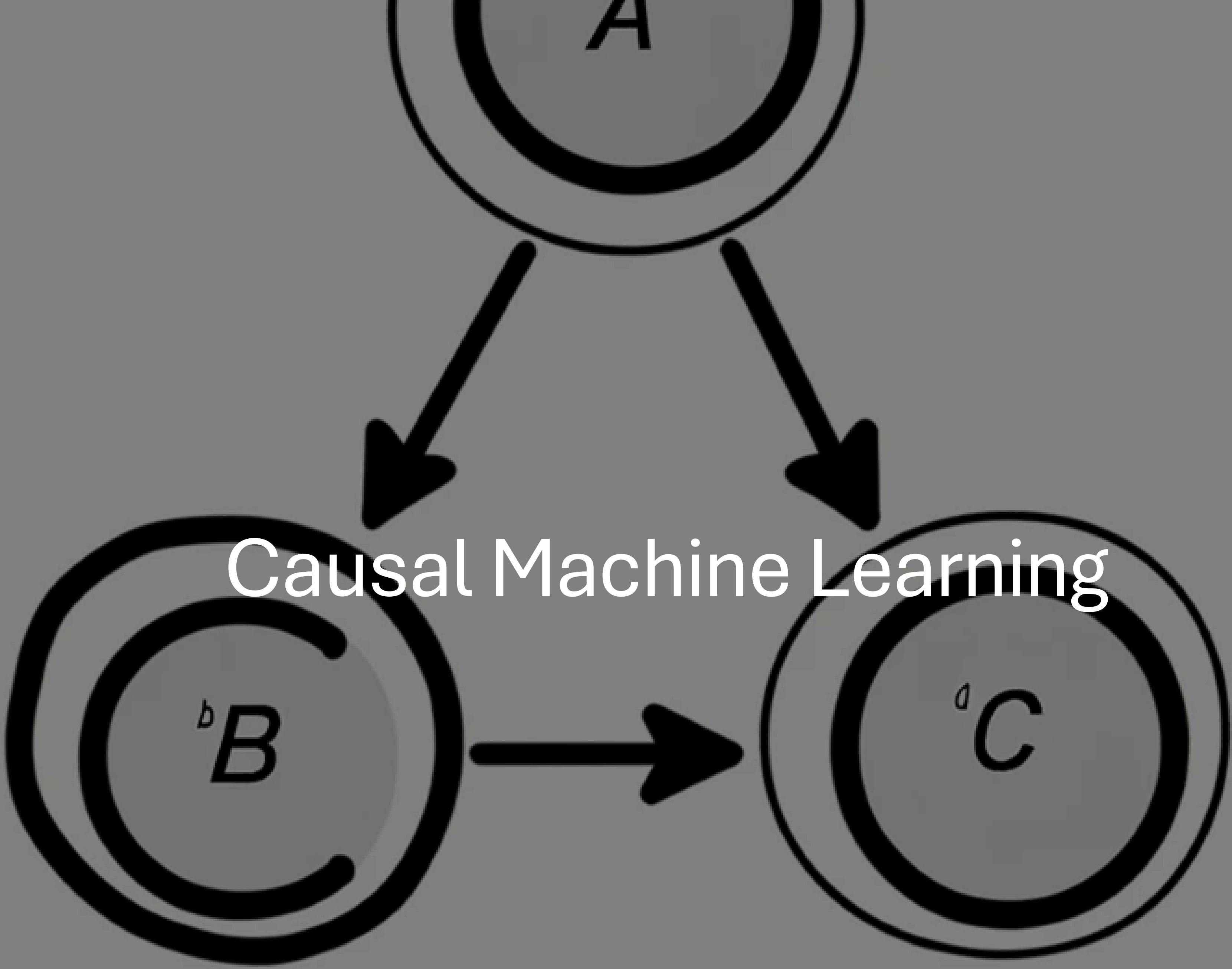
- At each period draw M samples $\tilde{z} = (\tilde{x}, \tilde{y}) \sim D_x \times \pi_\theta = \mathcal{P}_\theta$
- Draw one more sample $z_t = (x_t, y_t) \sim \mathcal{P}_\theta$ and assign reward

$$r_t(\theta) = \frac{1}{M} \sum_{m=1}^{M} P(z_t, \tilde{z}_m) \boxed{\approx E_{z \sim \pi_\theta, \tilde{z} \sim \pi_\theta}[P(z, \tilde{z})]}$$

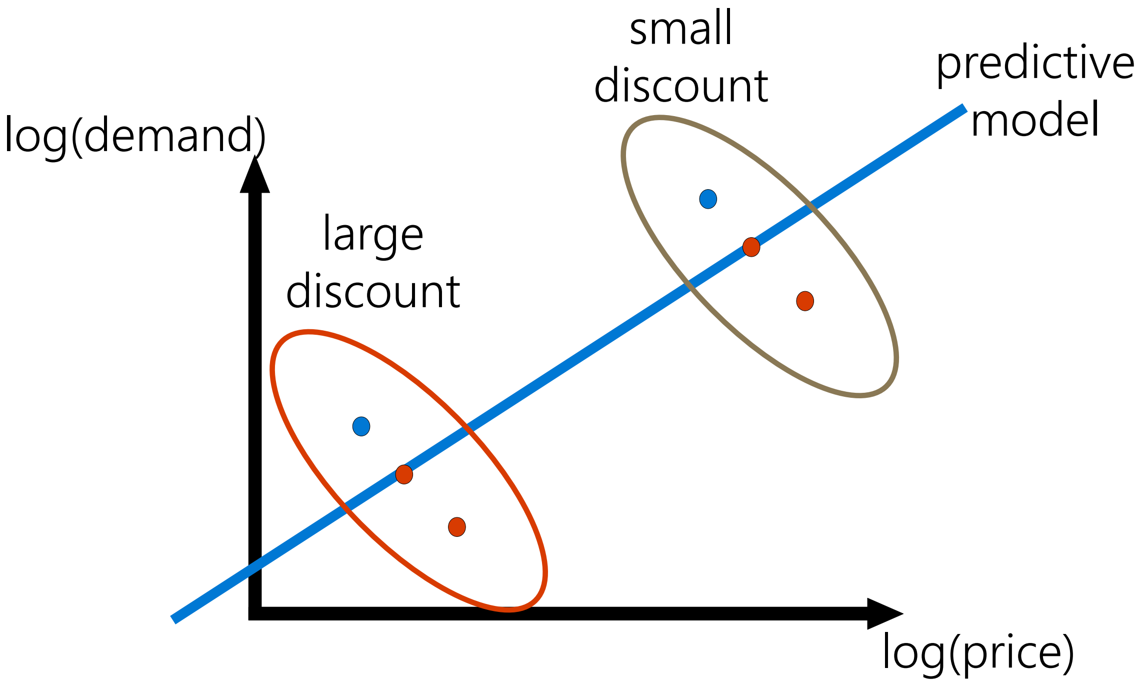
Reward of a sample z_t: fraction of times the sample is preferred to samples from the distribution of the current randomized policy

- Perform online gradient descent on the policy parameters
- Extends to Reinforcement Learning, where y is trajectory of decisions

Causal Machine Learning



Estimating Price Elasticity of Demand



$$\underset{\text{log(demand)}}{Y} = \underset{\text{elasticity}}{\theta_0} \cdot \underset{\text{log(price)}}{T} + \underset{\text{noise}}{\epsilon}$$

Conclusion: Increasing price **increases** demand!

Problem: Salespeople give discount to customers they foresee will have low probability of buying through other signals (confounders) they observe

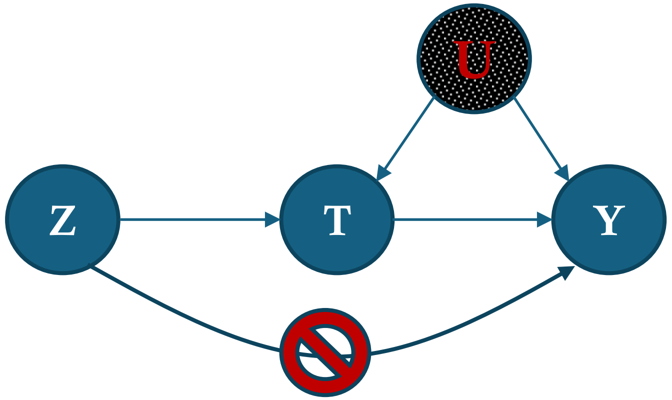
Unobserved Confounding: the major hurdle of causal analysis

- Healthcare:
  - Estimating the effects of cancer immunotherapy (IO) from non-trial data
  - Typically, late-stage patients undergo IO after standard of care treatment
  - Stage of disease not available in many public datasets.
  - Stage is an unobserved confounder

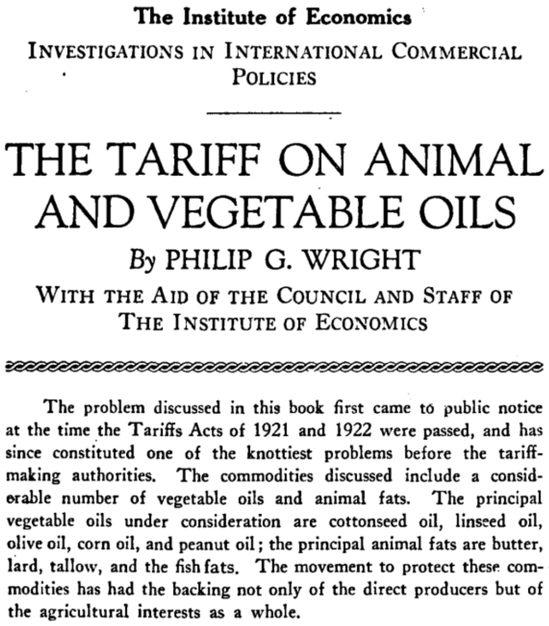
- In the tech sector:
  - Estimating the effects of customer characteristics on engagement
  - People who report more bugs last year, use the product more in the next year
  - Customer need for product leads to more usage last year ⇒ more bug reporting
  - Customer need is an unobserved confounder

Instrumental Variables and 2SLS

Instrumental Variable: any random variable **Z** that affects the treatment (log-price) **T** but does not affect the outcome (log-demand) **Y** other than through the treatment [Wright'28, Bowden-Turkington'90, Angrist-Krueger'91, Imbens-Angrist'94]



Philip Wright



The Institute of Economics

INVESTIGATIONS IN INTERNATIONAL COMMERCIAL POLICIES

THE TARIFF ON ANIMAL AND VEGETABLE OILS

By PHILIP G. WRIGHT

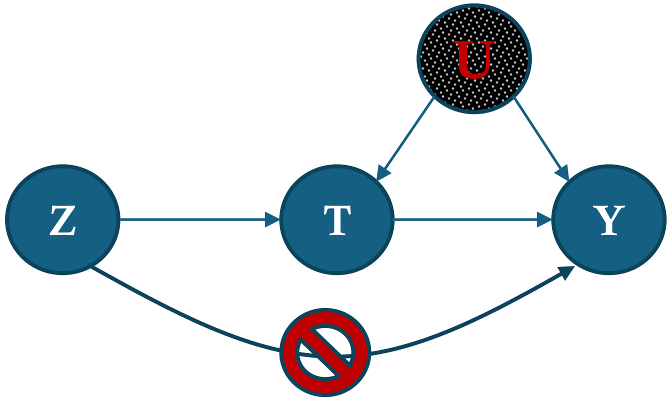
WITH THE AID OF THE COUNCIL AND STAFF OF THE INSTITUTE OF ECONOMICS

The problem discussed in this book first came to public notice at the time the Tariffs Acts of 1921 and 1922 were passed, and has since constituted one of the knottiest problems before the tariff-making authorities. The commodities discussed include a considerable number of vegetable oils and animal fats. The principal vegetable oils under consideration are cottonseed oil, linseed oil, olive oil, corn oil, and peanut oil; the principal animal fats are butter, lard, tallow, and the fish fats. The movement to protect these commodities has had the backing not only of the direct producers but of the agricultural interests as a whole.

Sewall Wright

Instrumental Variables and 2SLS

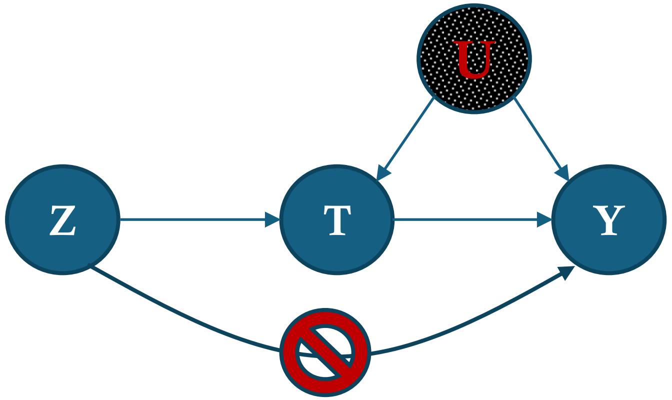
Instrumental Variable: any random variable **Z** that affects the treatment (log-price) **T** but does not affect the outcome (log-demand) **Y** other than through the treatment [Wright'28, Bowden-Turkington'90, Angrist-Krueger'91, Imbens-Angrist'94]

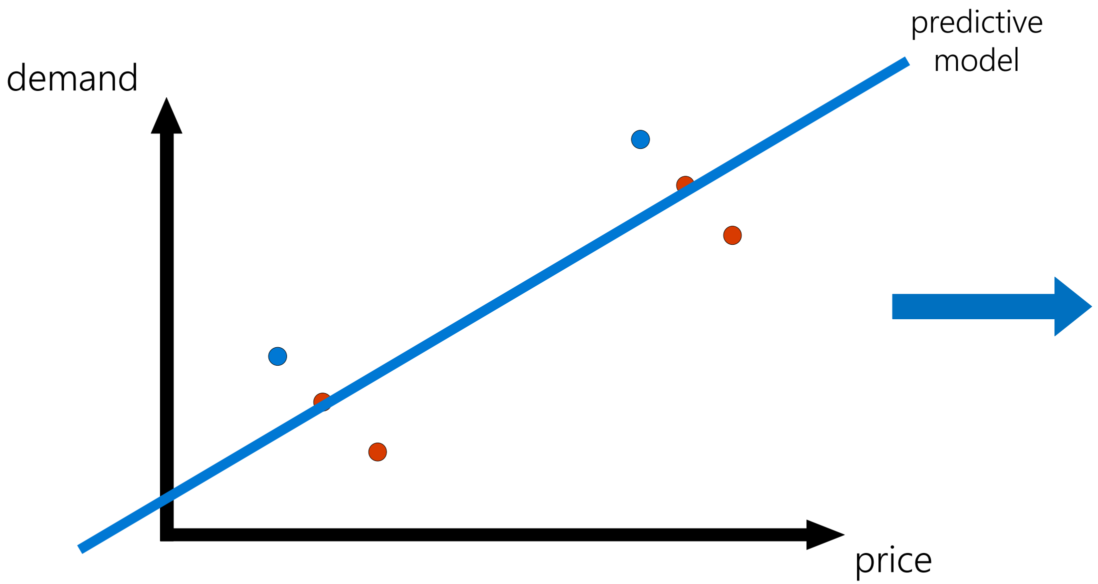


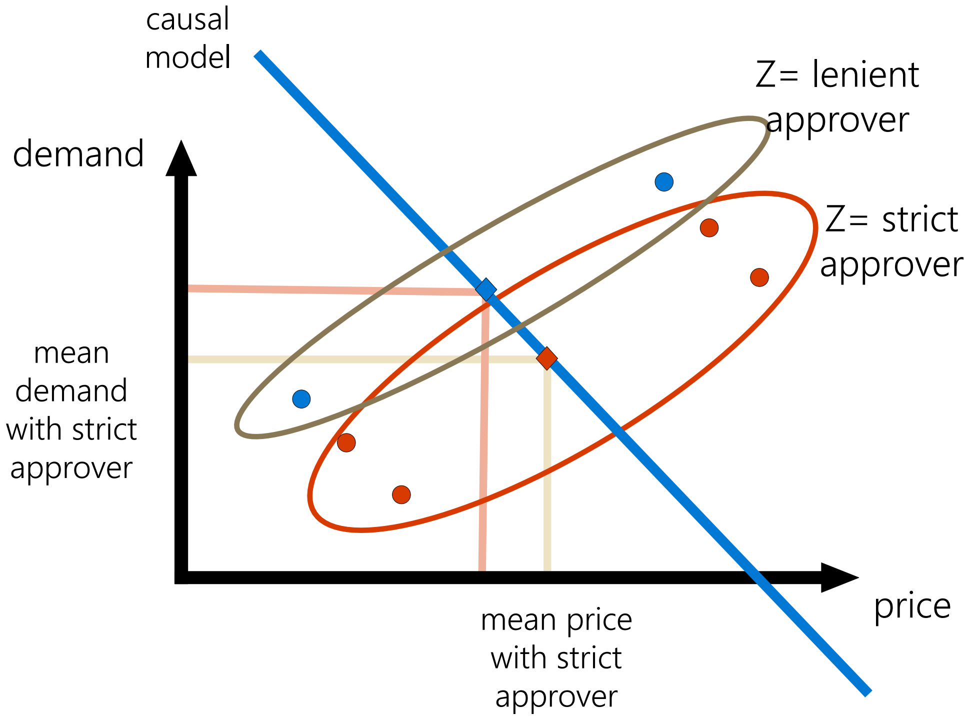
- In pricing (c.f. [Kling AER06, Aizer-Doyle15] for effects of incarceration): Discounts sent to an approver desk; Approver assignment is random; approvers are more/less "lenient"; leniency is instrument
- In healthcare [Doyle et al., JPE15]: Random assignment to ambulance companies of nearby patients is an instrument for measuring hospital quality
- In Tech [Syrgkanis et al, NeurIPS19]: Recommendation A/B tests as instruments for the effects of downstream actions

Instrumental Variables and 2SLS

- **Instrumental Variable:** any random variable **Z** that affects the treatment (log-price) **T** but does not affect the outcome (log-demand) **Y** other than through the treatment







Flexible Causal Models with Instruments

- Given outcome y, treatment T, instrument Z
- Want to estimate the causal effect model $h(T)$:

$$y = h(T) + U$$

- Since U is independent of Z (and without loss mean-zero)

$$\mathbb{E}[y - h(T)|Z] = \mathbb{E}[U|Z] = E[U] = 0$$

- Implies that for any function $f: Z \rightarrow R$

$$\mathbb{E}\left[\left(y - h(T)\right)f(Z)\right] = \mathbb{E}[\mathbb{E}[y - h(T) \mid Z]\, f(Z)] = 0$$

Instrumental Variables as Zero-Sum Game

- We know that the causal model must satisfy: for any function $f: Z \to R$

$$\mathbb{E}\left[\left(y - h(T)\right) f(Z)\right] = 0$$

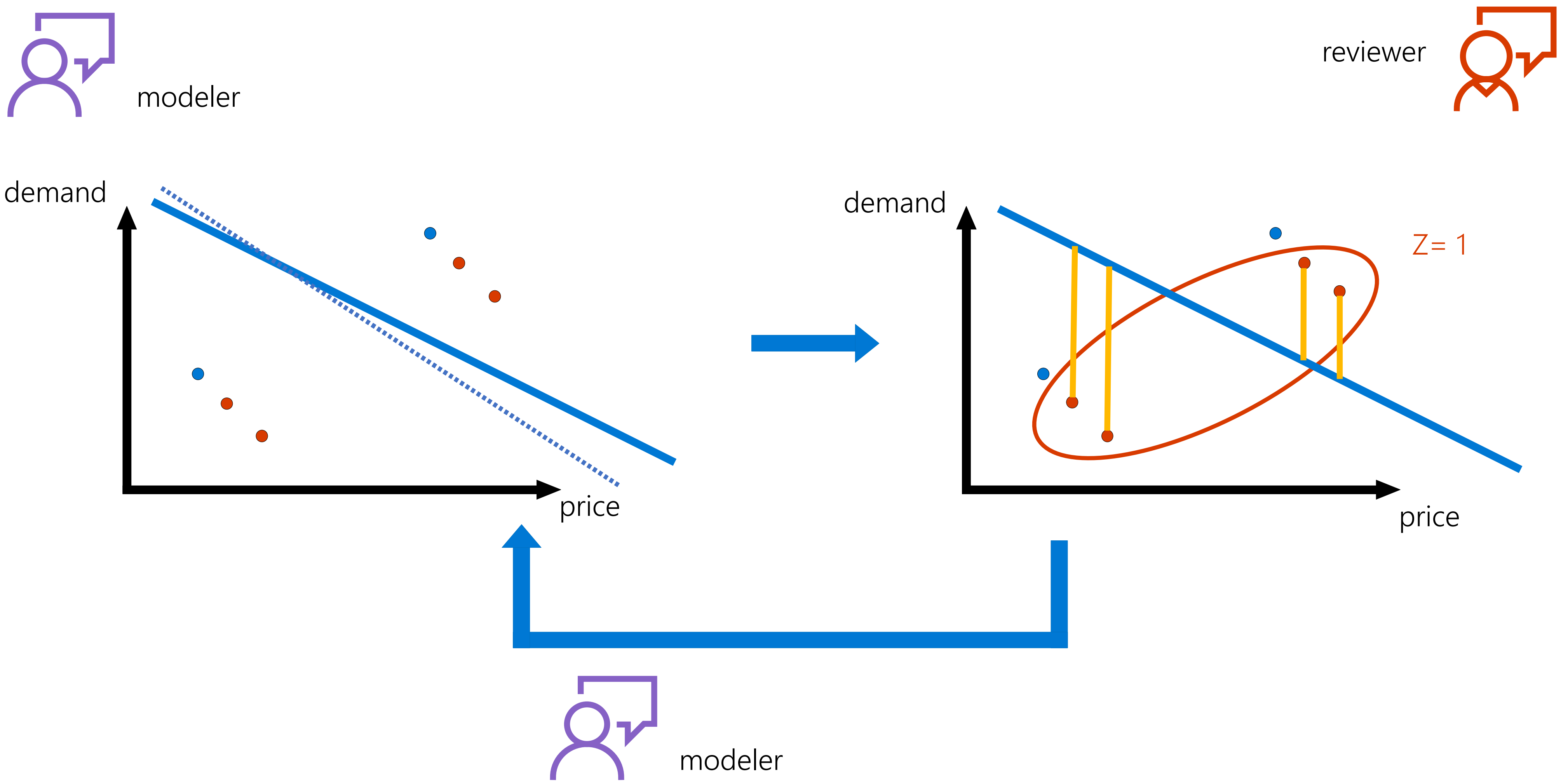
- Learner produces a candidate model h
- Adversary examines whether there exists a violation, i.e.

$$\max_{f \in F} \mathbb{E}\left[\left(y - h(T)\right) f(Z)\right] > 0$$

- Learner wants to find a model that adversary cannot complain about

$$\min_{h} \max_{f} \mathbb{E}\left[\left(y - h(T)\right) f(Z)\right]$$

Adversarial Estimation of Moment Models



Instrumental Variables as Zero-Sum Game

- Consider case when h and f are linear functions

$$\min_{\theta} \max_{w} \mathbb{E}[(y - \theta' T) w' Z]$$

- A convex-concave zero-sum game. Solve via OGD vs OGD

(OGD) $$\theta_t = \theta_{t-1} + \eta \mathbb{E}[T Z' w_{t-1}]$$

(OGD) $$w_t = w_{t-1} + \eta E[(y - \theta_{t-1}' T) Z]$$

- Translates to practical training algorithm for neural networks

$$\min_{\theta} \max_{w} \mathbb{E}\left[\left(y - h_{\theta}(T)\right) f_w(Z)\right]$$

- Variants have been proposed that have slightly better statistical properties

Punchline

Many problems in ML and AI can be cast as finding an equilibrium of a complex zero-sum game

No-regret dynamics are a typical paradigm that is used as solution

Translates to practical neural network training variants

Examples are Boosting, Distributional Robustness, Adversarial Robustness, Generative Learning, Learning from Human Feedback, Causal ML, Fair ML, Imitation Learning

Click on each topic for an example paper