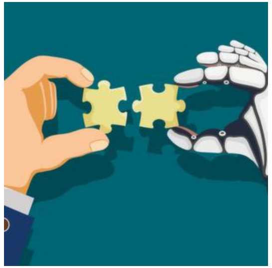


Human-Centric Machine Learning

Feedback loops, Human-AI Collaboration and Strategic Behavior

Manuel Gomez Rodriguez

Max Planck Institute for Software Systems

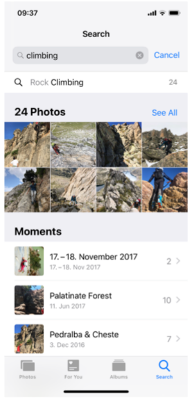
HUMAINT

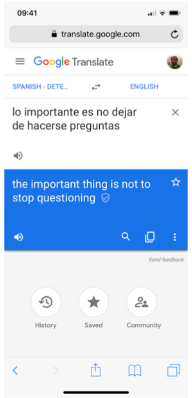
WINTER SCHOOL 22-24 January 2020, Seville (Spain)

Machine learning for automation

Machine learning (ML) has *taught* machines to...

recognize images | **translate between languages** | **play complex games**





...to name a few

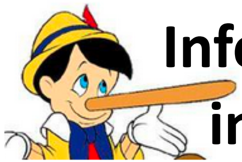
Machines achieve, or surpass, human performance at tasks for which intelligence is required

Machine learning for decision making

ML has the potential to support & enhance high-stakes decision making in a wide range of applications:

Justice

Hiring



Information integrity

Health

Education

Security

Finance

Increasing number of missteps

Machine learning has been **blamed to be one of the root causes** of an **increasing number of missteps**

misleading people in social media

increasing polarization

causing car accidents

discriminating minorities

What went wrong in these cases?

Machine learning has been mostly

developed for automation

Take decisions autonomously
on the basis of
$\underbrace{\text{passively collected data}}_{\text{passive setting}}$

What went wrong in these cases?

Algorithmic and human decisions **feed** and **influence** each other

⇩

$\underbrace{\text{Sequential decision making process}}_{\text{reactive setting}}$

Shortcomings of (traditional) ML models & algorithms

Ignore feedback loop between algorithmic and human decisions

Shortcomings of (traditional) ML models & algorithms

Ignore feedback loop between algorithmic and human decisions

↳ **Fail to anticipate how individuals will react to algorithmic decisions**

Shortcomings of (traditional) ML models & algorithms

Ignore feedback loop between algorithmic and human decisions

Do not account for strategic human behavior

Shortcomings of (traditional) ML models & algorithms

Ignore feedback loop between algorithmic and human decisions

Do not account for strategic human behavior

↳ **Unexpected & undesirable personal, social and economic consequences**

Shortcomings of (traditional) ML models & algorithms

Ignore feedback loop between algorithmic and human decisions

Do not account for strategic human behavior

Fail to balance decisions between machines and humans

Shortcomings of (traditional) ML models & algorithms

Ignore feedback loop between algorithmic and human decisions

Do not account for strategic human behavior

Fail to balance decisions between machines and humans

↳ They are unable to collaborate with humans

Shortcomings of (traditional) ML models & algorithms

- Ignore feedback loop between algorithmic and human decisions
- Do not account for strategic human behavior
- Fail to balance decisions between machines and humans
- Do not provide actionable insights

Shortcomings of (traditional) ML models & algorithms

Ignore feedback loop between algorithmic and human decisions

Do not account for strategic human behavior

Fail to balance decisions between machines and humans

Do not provide actionable insights

↳ **Interpretability is necessary to use ML in critical domains with consequential decisions.**

Outline of the lecture

A glimpse on recent advances on human-centric ML models and algorithms. We will focus on:

- Accounting for the feedback loop between algorithmic and human decisions
- Balancing decisions between human and algorithmic decisions
- Accounting for strategic human decisions

Disclaimer. These are emerging topics. The goal of this lecture is to introduce you to a new set of problems and, for each problem, show you one solution, not *the* solution.

A general problem setting

Decisions **Decision policy**

$$d(\boldsymbol{x}) \in \{0,1\} \sim \pi(d \,|\, \boldsymbol{x})$$

Features

$$\boldsymbol{x} \sim P(\boldsymbol{x})$$

Informed by

Label predictions **Predictive model**

$$y \in \mathcal{Y} \sim P_\theta(y \,|\, \boldsymbol{x})$$

Aim to predict a ground truth label

$$y \sim P(y \,|\, \boldsymbol{x})$$

Example 1: loan decisions

Decisions

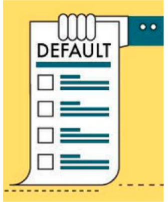
Decision policy

$$d(\boldsymbol{x}) \in \{0, 1\} \sim \pi(d \,|\, \boldsymbol{x})$$

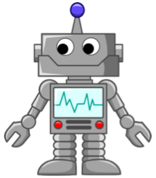
Individual is rejected

Individual receives loan

Informed by



Label predictions



Predictive model

$$y \in \mathcal{Y} \sim P_\theta(y \,|\, \boldsymbol{x})$$

$$\mathcal{Y} = \{0, 1\}$$

Individual defaults

Individual pays back

Example 2: bail decisions

Decisions

Decision policy

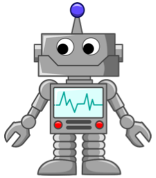
$$d(\boldsymbol{x}) \in \{0, 1\} \sim \pi(d \,|\, \boldsymbol{x})$$

Individual **remains jailed** (0)

Individual **is released** (1)

Informed by →

Label predictions



Predictive model

$$y \in \mathcal{Y} \sim P_\theta(y \,|\, \boldsymbol{x})$$

$$\mathcal{Y} = \{0, 1\}$$

Individual **reoffends** (0)

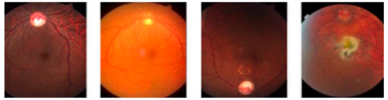
Individual **does not reoffend** (1)

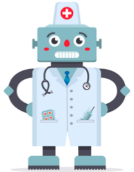
Example 3: medical diagnosis

Decisions

Decision policy

Informed by





Label predictions

Predictive model

$$d(\boldsymbol{x}) \in \{0, 1\} \sim \pi(d \mid \boldsymbol{x})$$

$$y \in \mathcal{Y} \sim P_\theta(y \mid \boldsymbol{x})$$

$$\mathcal{Y} = \{0, 1\}$$

Individual doesn't need further tests (0)

Individual needs further tests (1)

Low severity of Drusen's disease (0)

High severity of Drusen's disease (1)

Utility of a decision policy

The **decision maker** aims to deploy a **decision policy** that **maximizes** a very general definition of **utility:**

$$u(\pi, c) = \mathbb{E}_{\boldsymbol{x} \sim P(\boldsymbol{x}),\, y \sim P(y \,|\, \boldsymbol{x}),\, d \sim \pi(d \,|\, \boldsymbol{x})} [y \, d(\boldsymbol{x}) - c \, d(\boldsymbol{x})]$$

$$= \mathbb{E}_{\boldsymbol{x} \sim P(\boldsymbol{x}),\, d \sim \pi(d \,|\, \boldsymbol{x})} [P(y = 1 \,|\, \boldsymbol{x}) \, d(\boldsymbol{x}) - c \, d(\boldsymbol{x})]$$

Utility of a decision policy

The **decision maker** aims to deploy a **decision policy** that **maximizes** a very general definition of **utility:**

$$u(\pi, c) = \mathbb{E}_{\boldsymbol{x} \sim P(\boldsymbol{x}),\, y \sim P(y \,|\, \boldsymbol{x}),\, d \sim \pi(d \,|\, \boldsymbol{x})} [y\, d(\boldsymbol{x}) - c\, d(\boldsymbol{x})]$$
$$= \mathbb{E}_{\boldsymbol{x} \sim P(\boldsymbol{x}),\, d \sim \pi(d \,|\, \boldsymbol{x})} [P(y = 1 \,|\, \boldsymbol{x})\, d(\boldsymbol{x}) - c\, d(\boldsymbol{x})]$$

Example (loan decisions)

If a loan is granted and individual… repays: $1 - c$

… defaults: $-c$

If a loan is not granted: 0

The parameter c **measures the cost of offering a loan in units of repaid loans**

Benefits of a decision policy

To ensure **fairness,** the **decision maker** may constrain the **benefits** individuals obtain:

$$b(\boldsymbol{x}, c) = \mathbb{E}_{d \sim \pi(d \,|\, \boldsymbol{x})} \underbrace{[f(d(\boldsymbol{x}), c)]}_{\textbf{Problem dependent}}$$

Benefits of a decision policy

To ensure **fairness,** the **decision maker** may constrain the **benefits** individuals obtain:

$$b(\boldsymbol{x}, c) = \mathbb{E}_{d \sim \pi(d \,|\, \boldsymbol{x})} [\underbrace{f(d(\boldsymbol{x}), c)}_{\textbf{Problem dependent}}]$$

Example (loan decisions)

Statistical parity:
Ensure the men and women have the same probability of receiving a loan

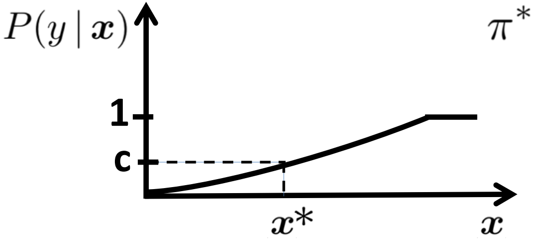
$$\sum_{\boldsymbol{x} \in} b(\boldsymbol{x}, c) \approx \sum_{\boldsymbol{x} \in} b(\boldsymbol{x}, c)$$

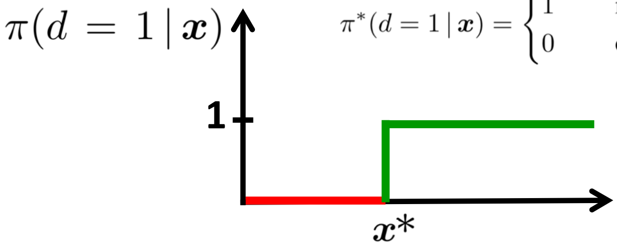
$$f(d(\boldsymbol{x}), c) = d(\boldsymbol{x})$$

Deterministic threshold rules

Under some ***technical conditions***, **deterministic threshold rules** are **optimal decision policies:**



$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, u(\pi, c)$$

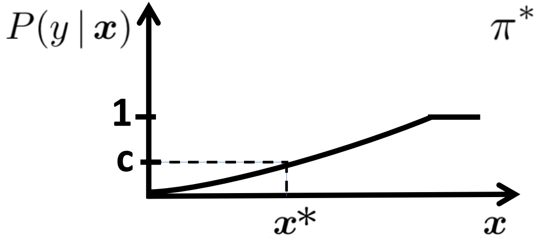


$$\pi^*(d = 1 \mid \boldsymbol{x}) = \begin{cases} 1 & \text{if } P(y = 1 \mid \boldsymbol{x}) \geq c \\ 0 & \text{otherwise.} \end{cases}$$

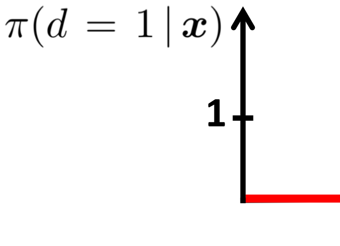
[Corbett-Davies et al., KDD 2017]

Deterministic threshold rules

Under some ***technical conditions***, **deterministic threshold rules** are **optimal decision policies:**

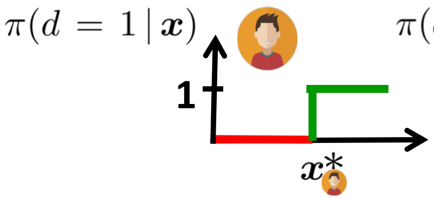


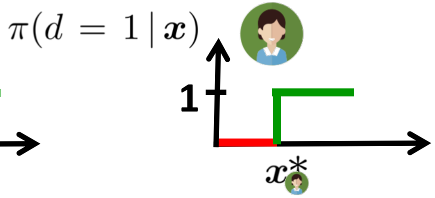
$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, u(\pi, c)$$



$$\pi^*(d = 1 \,|\, \boldsymbol{x}) = \begin{cases} 1 & \text{if } P(y = 1 \,|\, \boldsymbol{x}) \geq c \\ 0 & \text{otherwise.} \end{cases}$$

Under **fairness constraints,** we just need **two thresholds:**





[Corbett-Davies et al., KDD 2017]

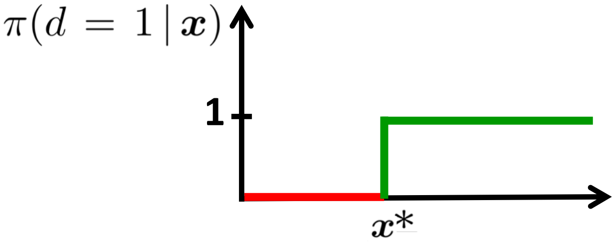
Why are deterministic threshold rules optimal?

To realize **why deterministic threshold rules** are **optimal**, rewrite the utility of the policy as follows:

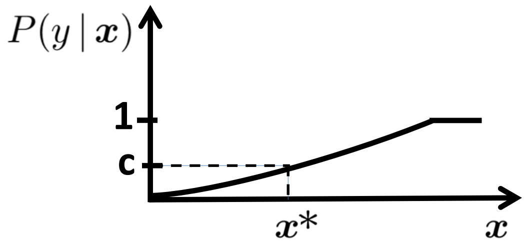
$$u(\pi, c) = \mathbb{E}_{\boldsymbol{x} \sim P(\boldsymbol{x}),\, d \sim \pi(d \,|\, \boldsymbol{x})}\left[d(\boldsymbol{x})\underbrace{\left(P(y=1 \,|\, \boldsymbol{x}) - c\right)}\right]$$

It is only positive if

$$P(y=1 \,|\, \boldsymbol{x}) \geq c$$



If $P(y=1 \,|\, \boldsymbol{x}) \geq c$,
make $d = 1$,
otherwise, make $d = 0$



[Corbett-Davies et al., KDD 2017]

So, are we done?

Let's look into the ***technical conditions***

1. The **predictive model** is ***perfect***

$$\longrightarrow \; P_\theta(y \,|\, \boldsymbol{x}) \;=\; P(y \,|\, \boldsymbol{x})$$

2. The **label/feature distributions** and the **policy** are **independent**

$$\longrightarrow \; P(y \,|\, \boldsymbol{x}, \pi) = P(y \,|\, \boldsymbol{x}) \qquad P(\boldsymbol{x} \,|\, \pi) = P(\boldsymbol{x})$$

3. Individuals are **not strategic**

$\longrightarrow$ **Individuals** do **not seek** to **maximize** their **benefit** $b(\boldsymbol{x}, c)$

So, are we done?

Let's look into the ***technical conditions***

1. The **predictive model** is ***perfect***

In practice, these **technical conditions**
are (often) **violated**.

3. Individuals are **not strategic**

→ **Individuals** do **not seek** to **maximize** their **benefit** $b(x, c)$

Dealing with imperfect predictions

Assume the **predictive model** $P_\theta(y \mid \boldsymbol{x}) = Q(y \mid \boldsymbol{x})$ trained using **historical data** is **imperfect**, i.e.,

$$Q(y \mid \boldsymbol{x}) = P(y \mid \boldsymbol{x}) + \varepsilon(y \mid \boldsymbol{x})$$

We will distinguish two different cases:

(a) Historical data suffers from ***selective labels*** [Lakkaraju et al., KDD 2017]

(b) Historical data is sampled from the **ground truth data distribution**

More common

Less common

Historical data suffers from selective labels

Historical data is not **sampled** from the **ground truth distribution** but **a distribution induced** by a **previously deployed policy**

$$\underbrace{P_{\pi_0}(\boldsymbol{x}, y)}_{\text{Data distribution induced by historical policy}} \propto P(y \,|\, \boldsymbol{x}) \underbrace{\pi_0(d = 1 \,|\, \boldsymbol{x})}_{\text{Deployed historical policy}} P(\boldsymbol{x})$$

Example

Loan decisions:

Historical data only contains individuals who received a loan in the past

The **(induced) label/feature distribution & policy** are **dependent!**

[Kilbertus et al., AISTATS 2020]

Historical data suffers from selective labels

Historical data is not **sampled** from the **ground truth distribution** but **a distribution induced** by a **previously deployed policy**

$$P_{\pi_0}(\boldsymbol{x}, y) \propto P(y \mid \boldsymbol{x})\pi_0(d = 1 \mid \boldsymbol{x})P(\boldsymbol{x})$$

D

This creates a feedback loop between human decisions and algorithmic decisions

Loan decisions:

Historical data only contains individuals who received a loan in the past

dependent!

[Kilbertus et al., AISTATS 2020]

Are threshold rules provably suboptimal?

Take the **optimal policy** under the **original data distribution** and the **data distribution induced by the historical policy**:

$$Q^* \in \underset{Q \in \mathcal{Q}}{\operatorname{argmax}} \, \mathbb{E}_{\boldsymbol{x}, y \sim P} [\mathbf{1}[Q(y = 1 \mid \boldsymbol{x}) \geq c](y - 1)]$$

$$Q_0^* \in \underset{Q \in \mathcal{Q}}{\operatorname{argmax}} \, \mathbb{E}_{\boldsymbol{x}, y \sim P_{\pi_0}} [\mathbf{1}[Q(y = 1 \mid \boldsymbol{x}) \geq c](y - 1)]$$

[Kilbertus et al., AISTATS 2020]

Are threshold rules provably suboptimal?

Take the **optimal policy** under the **original data distribution** and the **data distribution induced by the historical policy**:

$$Q^* \in \operatorname*{argmax}_{Q \in \mathcal{Q}} \mathbb{E}_{\boldsymbol{x},y \sim P}[\mathbf{1}[Q(y=1 \mid \boldsymbol{x}) \geq c](y-1)]$$

$$Q_0^* \in \operatorname*{argmax}_{Q \in \mathcal{Q}} \mathbb{E}_{\boldsymbol{x},y \sim P_{\pi_0}}[\mathbf{1}[Q(y=1 \mid \boldsymbol{x}) \geq c](y-1)]$$

Proposition (negative result!). If $\pi_0 \neq \pi^*$ then

$$u(\pi_{Q_0^*}, c) < u(\pi_{Q^*}, c)$$

[Kilbertus et al., AISTATS 2020]

In which class of policies lies the optimal decision policy?

$\pi_0(d = 1 \mid \boldsymbol{x}, s) > 0$ on any measurable set with positive probability under P

It lies in the set of ***exploring policies*.**

A **policy** π **is** ***exploring*** iff the **true distribution** P is **absolutely continuous** with respect to P_π

[Kilbertus et al., AISTATS 2020]

In which class of policies lies the optimal decision policy?

$\pi_0(d=1\,|\,\boldsymbol{x}, s) > 0$ on any measurable set with positive probability under P

It lies in the set of ***exploring policies*.**

A **policy** π **is** ***exploring*** iff the **true distribution** P is **absolutely continuous** with respect to P_π

Proposition (positive result!). If π_0 is a exploring policy,

$$u(\pi^*, c) = \sup_{\pi \in \Pi} \mathbb{E}_{\boldsymbol{x}, y \sim P_{\pi_0}} \left[\frac{\pi(d=1\,|\,\boldsymbol{x})}{\pi_0(d=1\,|\,\boldsymbol{x})} (y-c) \right]$$

Set of exploring policies

Induced distribution!

[Kilbertus et al., AISTATS 2020]

Not all exploring policies are (equally) acceptable

Consider a **lending scenario**

The following **decision policies** are **exploring**:

Give **loans to everyone,** $\pi_0(d \mid \boldsymbol{x}) = 1$ **for all** $\boldsymbol{x}$

Gives **loans to every individual** at **random,** $\pi_0(d = 1 \mid \boldsymbol{x}) = 0.5$ **for all** $\boldsymbol{x}$

Who thinks a bank will do well under these policies? ☺

[Kilbertus et al., AISTATS 2020]

Learning exploring policies

1. **Deploy** an **initial exploring policy** π_0, which may be far from optimal for not too long.

2. Use **data gathered with this initial exploring policy** to fit a new **parameterized exploring policy** π_θ using **SGA**, i.e.,

$$\theta_{i+1} = \theta_i + \alpha_i \nabla_\theta u(\pi_\theta, c)|_{\theta=\theta_i}$$

Log-derivative and reweighting tricks

Distribution induced by initial policy

$$\mathbb{E}_{\boldsymbol{x},y \sim P_{\pi_0}, d \sim \pi_\theta}\left[\frac{d(y-1)}{\pi_(d = 1 \mid \boldsymbol{x})} \nabla_\theta \log \pi_\theta(d \mid \boldsymbol{x})\right]$$

New policy

3. **Deploy & gather data** with π_θ and fit a better exploring policy. Repeat.

[Kilbertus et al., AISTATS 2020]

Learning exploring policies

1. **Deploy** an **initial exploring policy** π_0, which may be far from optimal for not too long.

2. U

new

θ_{i+}

Learning to decide rather than **learning to predict!**

Log-derivative and reweighting tricks

by initial policy

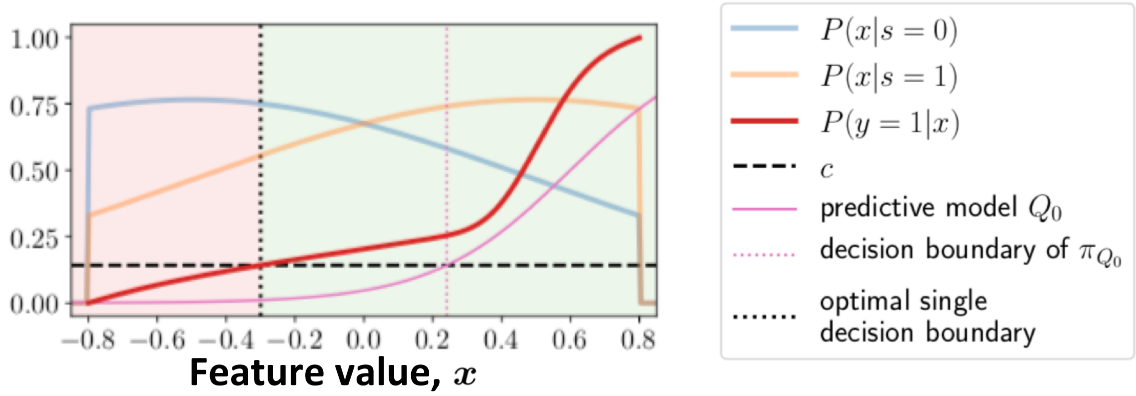
$$\mathbb{E}_{\boldsymbol{x},y\sim P_{\pi_0},d\sim\pi_\theta}\left[\frac{d(y-1)}{\pi_(d=1\,|\,\boldsymbol{x})}\nabla_\theta \log \pi_\theta(d\,|\,\boldsymbol{x})\right]$$

New policy

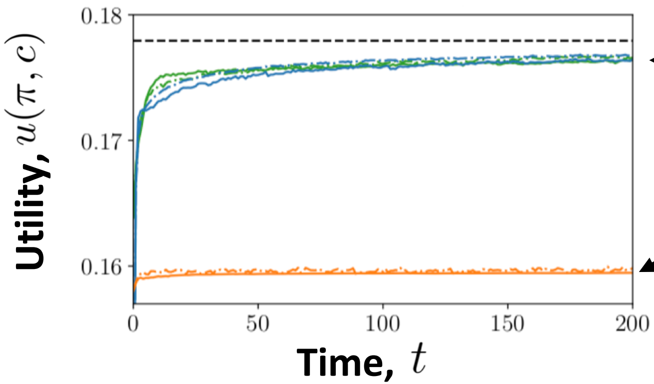
3. **Deploy & gather data** with π_θ and fit a better exploring policy. Repeat.

[Kilbertus et al., AISTATS 2020]

Example 1: strictly monotonic label distribution



Higher is better

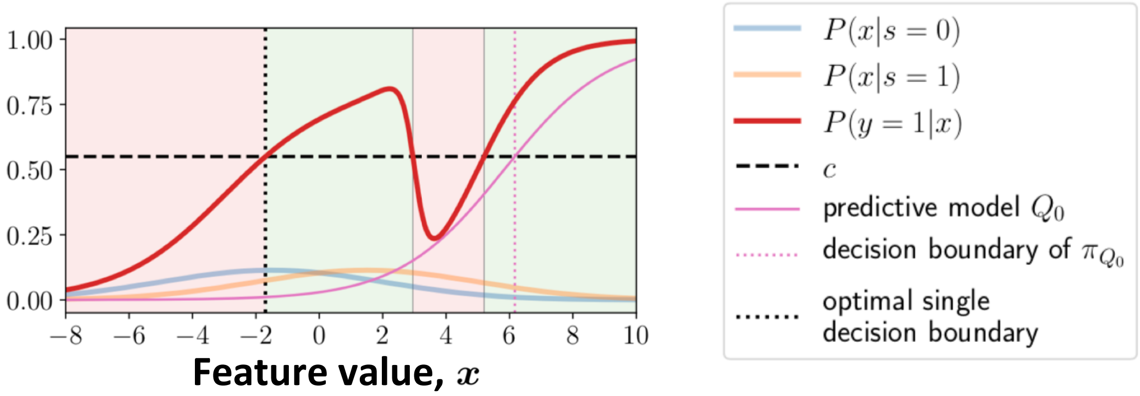


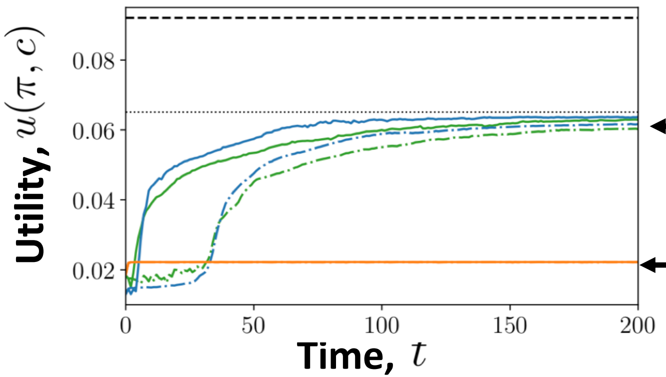
Parameterized exploring policies

Deterministic threshold rules

[Kilbertus et al., AISTATS 2020]

Example 2: nonmonotonic label distribution





Higher is better

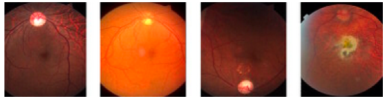
Parameterized exploring policies

Deterministic threshold rules

[Kilbertus et al., AISTATS 2020]

Historical data is sampled from the ground truth distribution

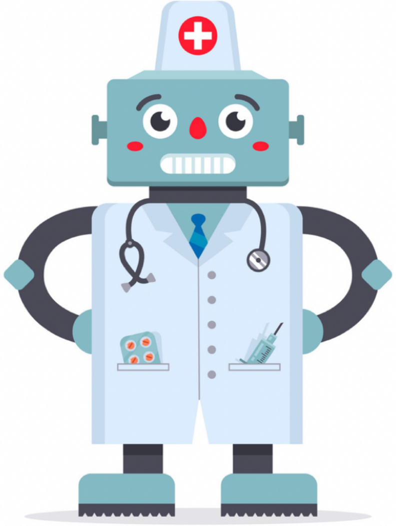
There are **situations** where the **historical data** is sampled from the **ground truth distribution.**



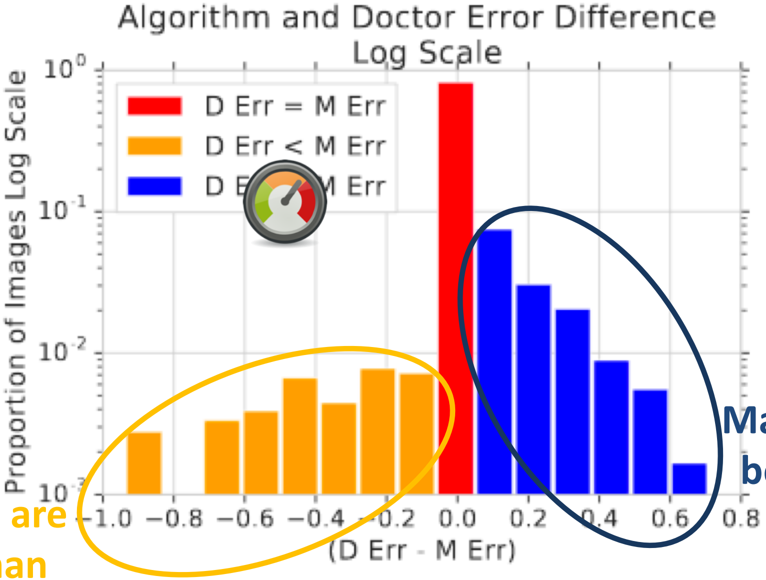
If a person has (or has not) a disease, this fact does not change after a medical diagnosis by a doctor

Then, given the latest deep ML model, can we just gather enough data to train a *perfect model*?

Machines learning is sometimes worse than humans

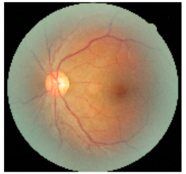


On **some instances, machine predictions** are still **worse** than **predictions** made by **human experts**



Machines are worse than humans

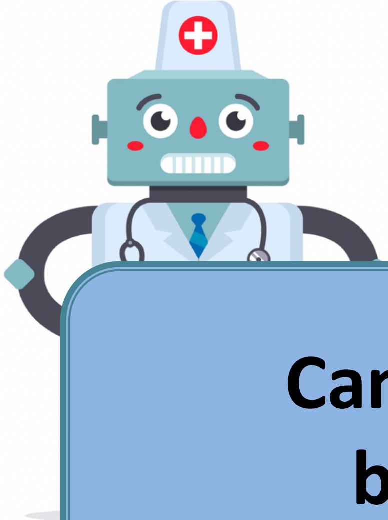
Machines are better than humans



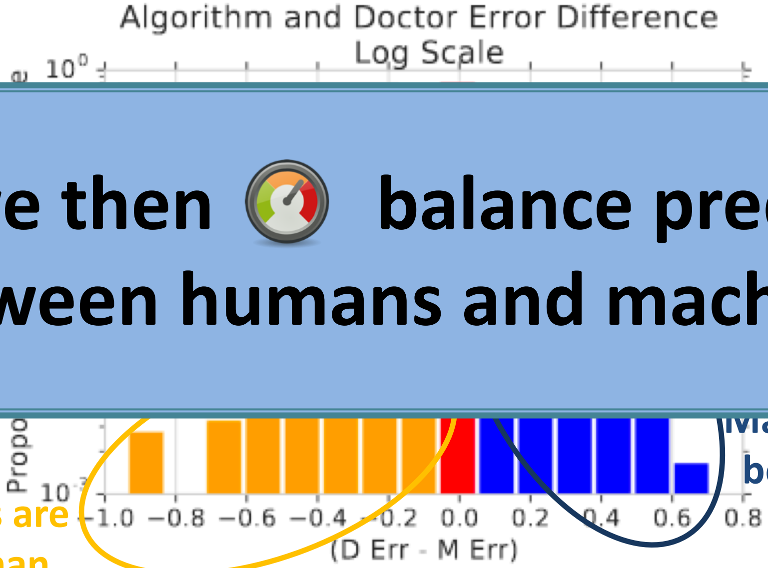
Task: estimating severity of diabetic retinopathy

[Raghu et al., Arxiv 2019]

Machines learning is sometimes worse than humans



On <u>**some instances**</u>, **machine predictions** are still **worse** than **predictions** made by **human experts**



of

Can we then balance predictions between humans and machines?

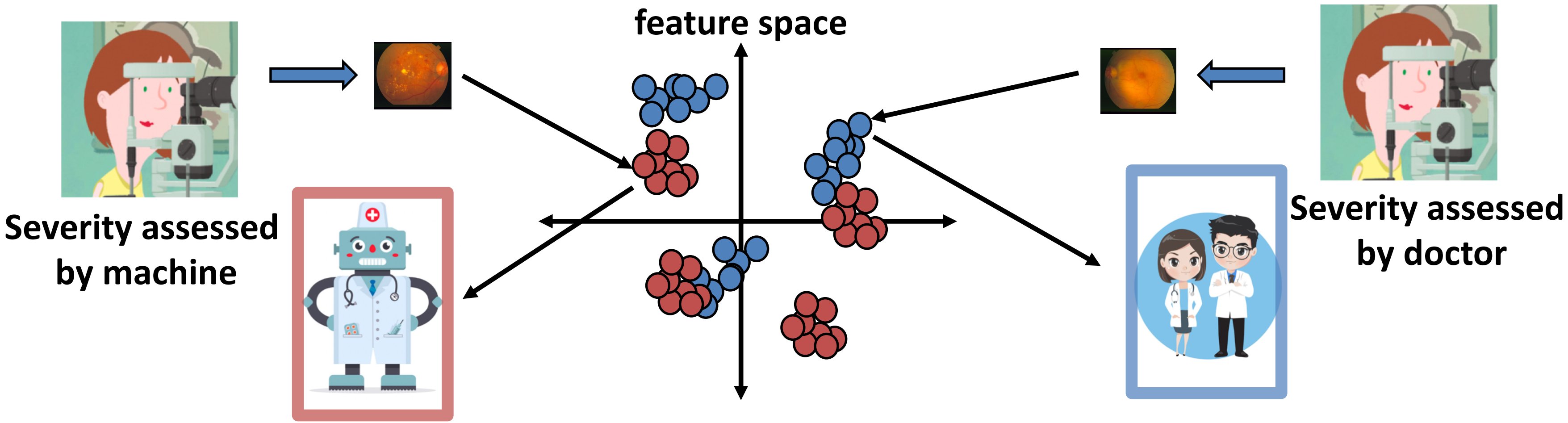
Machines are worse than humans

Machines are better than humans

[Raghu et al., Arxiv 2019]

Machine learning for different automation levels

Key idea: develop **machine learning models** that are **optimized** to operate under **different automation levels**



They take **decisions** for a given fraction of the instances and **leave the remaining ones to humans**

[De et al., AAAI 2020]

Optimizing the machine during training and test time

Key idea

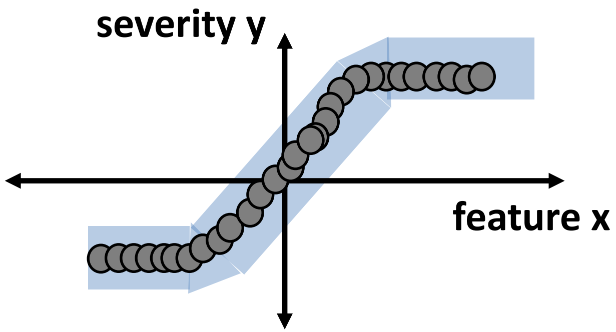
optimize the **design** of the **machine during training**

[De et al., AAAI 2020]

Optimizing the machine during training and test time

Key idea

optimize the **design** of the **machine during training**



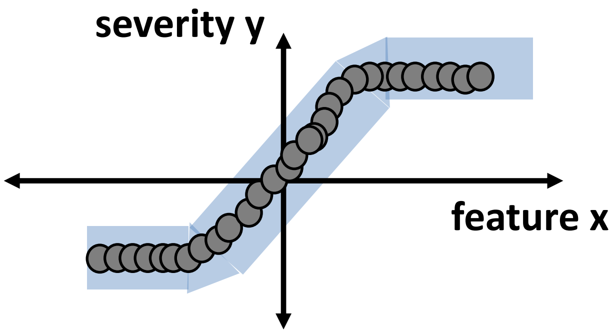
1. The machine model is a linear function
2. We can defer some samples to humans

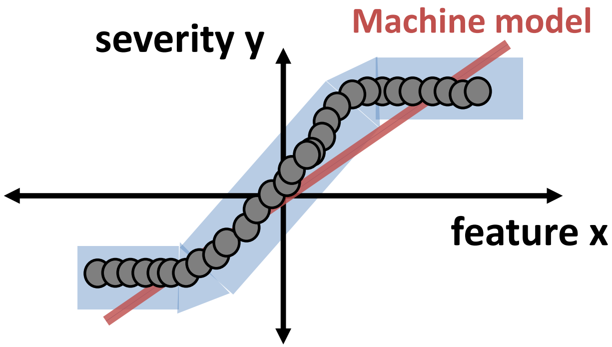
[De et al., AAAI 2020]

Optimizing the machine during training and test time

Key idea

optimize the **design** of the **machine during training**





1. The machine model is a linear function
2. We can defer some samples to humans

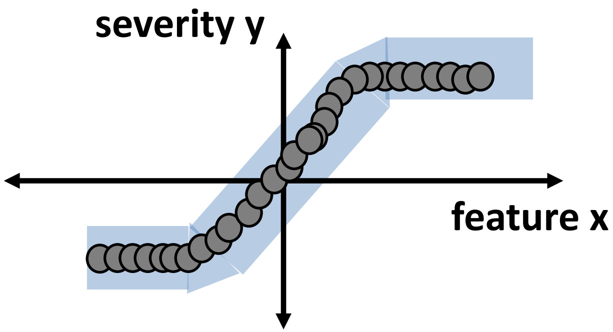
Machine model is not optimized during training

[De et al., AAAI 2020]

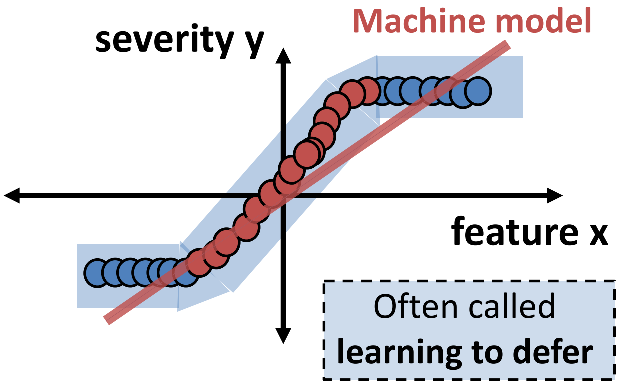
Optimizing the machine during training and test time

Key idea

optimize the **design** of the **machine during training**



1. The machine model is a linear function
2. We can defer some samples to humans



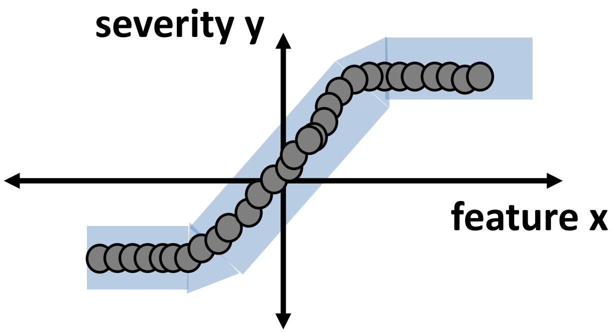
Machine model is not optimized during training

[De et al., AAAI 2020]

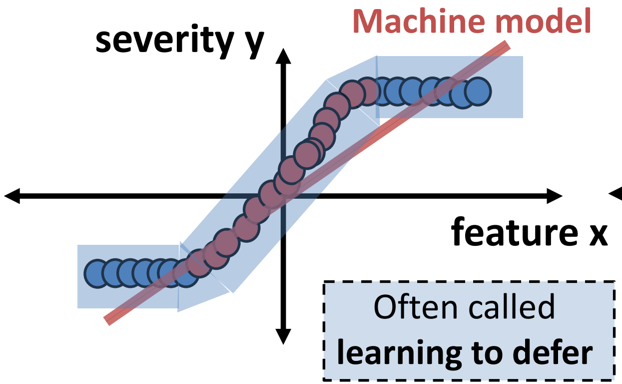
Optimizing the machine during training

Key idea

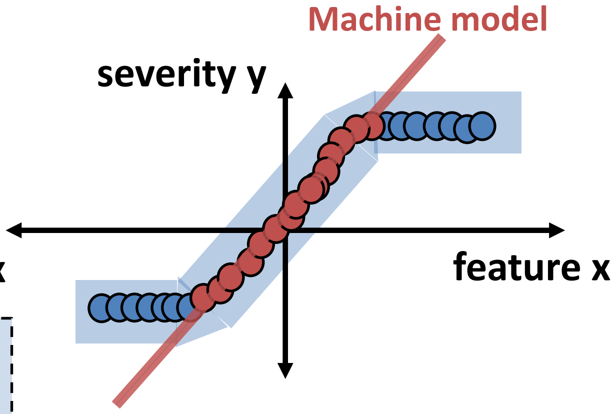
optimize the **design** of the **machine during training**



1. The machine model is a linear function
2. We can defer some samples to humans



Machine model is not optimized during training



Machine model is optimized during training

[De et al., AAAI 2020]

Optimizing the machine during training

Key idea

optimize the **design** of the **machine during training**

odel

Next, we will show how to **design** a **ridge regression model optimized** to **operate** under **different automation levels**

ture x

1. The machine model is a linear function
2. We can defer some samples to humans

Machine model is not optimized during training

Machine model is optimized during training

[De et al., AAAI 2020]

Ridge regression, revisited

Training

$$\underset{\boldsymbol{w},\, \mathcal{S}}{\text{minimize}} \quad \sum_{i\in\mathcal{S}} \overbrace{c(\boldsymbol{x}_i, y_i)}^{\text{Human error per sample}} + \sum_{j\in\mathcal{S}^c} \left[(y_j - \boldsymbol{x}_j^\top \boldsymbol{w})^2 + \lambda||\boldsymbol{w}||_2^2\right] \qquad \text{subject to} \quad |\mathcal{S}| \leq n$$

Regularization parameter

Training samples assigned to humans

Training samples assigned to machines

Machine model

Max. number of samples that can be assigned to humans

[De et al., AAAI 2020]

Ridge regression, revisited

Training

$$\underset{\boldsymbol{w},\, \mathcal{S}}{\text{minimize}} \quad \sum_{i \in \mathcal{S}} \overbrace{c(\boldsymbol{x}_i, y_i)}^{\text{Human error per sample}} + \sum_{j \in \mathcal{S}^c} \left[(y_j - \boldsymbol{x}_j^\top \boldsymbol{w})^2 + \lambda ||\boldsymbol{w}||_2^2 \right]$$

$$\text{subject to} \quad |\mathcal{S}| \leq n$$

Regularization parameter

Machine model

Training samples assigned to humans

Training samples assigned to machines

Max. number of samples that can be assigned to humans

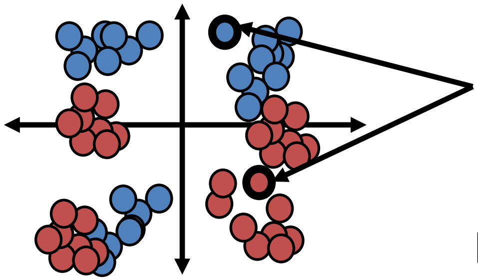
Test

We **assign** a **test sample** with **features** $\boldsymbol{x}$ to **humans** if

$$\text{argmin}_{i \in \mathcal{V}} ||\boldsymbol{x}_i - \boldsymbol{x}|| \in \mathcal{S}^*$$

Training samples assigned to humans

Training samples $\mathcal{V}$



Test samples

[De et al., AAAI 2020]

Ridge regression becomes a combinatorial problem

Given a fixed **set** $\mathcal{S}$**,** the **optimal machine model** is given by

$$\boldsymbol{w}^*(S) = \left(\lambda|\mathcal{S}^c|\mathbb{I} + \boldsymbol{X}_{\mathcal{S}^c}\boldsymbol{X}_{\mathcal{S}^c}^\top\right)^{-1}\boldsymbol{X}_{\mathcal{S}^c}\boldsymbol{y}_{\mathcal{S}^c}$$

[De et al., AAAI 2020]

Ridge regression becomes a combinatorial problem

Given a fixed **set** $\mathcal{S}$, the **optimal machine model** is given by

$$\boldsymbol{w}^*(S) = \left(\lambda|\mathcal{S}^c|\mathbb{I} + \boldsymbol{X}_{\mathcal{S}^c}\boldsymbol{X}_{\mathcal{S}^c}^\top\right)^{-1}\boldsymbol{X}_{\mathcal{S}^c}\boldsymbol{y}_{\mathcal{S}^c}$$

Then, we can **rewrite** the **ridge regression problem** as a purely **combinatorial maximization problem**

$$\underset{\mathcal{S}}{\text{maximize}} \quad -\underbrace{\log \ell(\mathcal{S})}_{\Downarrow} \qquad \text{subject to} \quad |\mathcal{S}| \leq n$$

$$\textstyle\sum_{i\in\mathcal{S}} c(\boldsymbol{x}_i, y_i) + \boldsymbol{y}_{\mathcal{S}^c}^\top\boldsymbol{y}_{\mathcal{S}^c} - \boldsymbol{y}_{\mathcal{S}^c}^\top\boldsymbol{X}_{\mathcal{S}^c}^\top\left(\lambda|\mathcal{S}^c|\mathbb{I} + \boldsymbol{X}_{\mathcal{S}^c}\boldsymbol{X}_{\mathcal{S}^c}^\top\right)^{-1}\boldsymbol{X}_{\mathcal{S}^c}\boldsymbol{y}_{\mathcal{S}^c}$$

[De et al., AAAI 2020]

Ridge regression under human assistance is hard

Finding the solution to

$$\underset{\mathcal{S}}{\text{maximize}} \quad -\log \ell(\mathcal{S}) \qquad \text{subject to} \quad |\mathcal{S}| \leq n$$

is a **NP-hard problem**

[De et al., AAAI 2020]

Ridge regression under human assistance is hard

Finding the solution to

$$\underset{\mathcal{S}}{\text{maximize}} \quad -\log \ell(\mathcal{S}) \quad \text{subject to} \quad |\mathcal{S}| \leq n$$

is a **NP-hard problem**

Proof sketch

k-sparse noise vector

Assume $c(\boldsymbol{x}_i, y_i) = 0$, $\lambda = 0$ and $\boldsymbol{y} = \boldsymbol{X}^\top \boldsymbol{w}^* + \boldsymbol{b}^*$

Then, **the problem** can be viewed as the **robust least square (RLSR) problem,** which **has been shown to be NP-hard:**

$$\underset{\boldsymbol{w}, \mathcal{S}}{\text{minimize}} \sum_{i \in \mathcal{S}} (y_i - \boldsymbol{x}_i^\top \boldsymbol{w})^2 \;\; \text{subject to } |\mathcal{S}| = |\mathcal{V}| - n$$

[De et al., AAAI 2020]

A simple greedy algorithm

The **greedy algorithm** proceeds **iteratively.**

At each iteration, it **assigns to a human** the **sample in the training set** that provides the **largest** marginal gain, i.e.,

$$k^* \leftarrow \operatorname{argmax}_{k \in \underbrace{\mathcal{V} \backslash \mathcal{S}}_{\text{Points not yet assigned to humans}}} \overbrace{- \log \ell(\mathcal{S} \cup k) + \log \ell(\mathcal{S})}$$

$$\mathcal{S} \leftarrow \mathcal{S} \cup \{k^*\}$$

Does this simple greedy algorithm has approximation guarantees? ☺

[De et al., AAAI 2020]

The greedy algorithm has approximation guarantees

The function $-\log \ell(\mathcal{S})$ satisfies an **approximate notion of submodularity**

$$-\log \ell(\mathcal{S} \cup k) + \log \ell(\mathcal{S}) \geq (1-\alpha)\left[-\log \ell(\mathcal{T} \cup k) + \log \ell(\mathcal{T})\right]$$

for all $\mathcal{S} \subseteq \mathcal{T} \subset \mathcal{V}$

where $\alpha \geq \alpha^*$ is the **generalized curvature**

Data dependent constant

We can conclude that the greedy algorithm will find a set $\mathcal{S}$ such that

Optimal value

$$\boxed{-\log \ell(\mathcal{S}) \geq \left(1 + \frac{1}{1-\alpha}\right)^{-1} OPT}$$

[Gatmiry et al., Arxiv 2018]

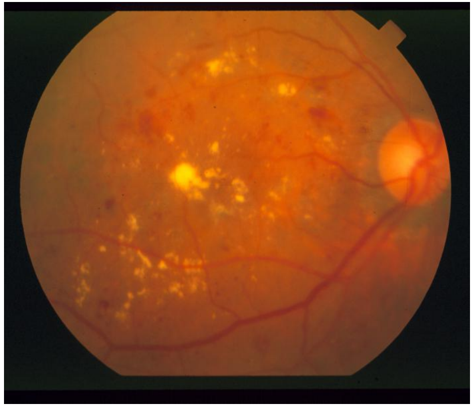
What samples are outsourced?

Drusen disease is characterized by **pathological yellow spots...**

…however, both images are given a **score of severity zero**

Easy sample

It is assigned to the machine



Difficult sample

It is assigned to the human

[De et al., AAAI 2020]

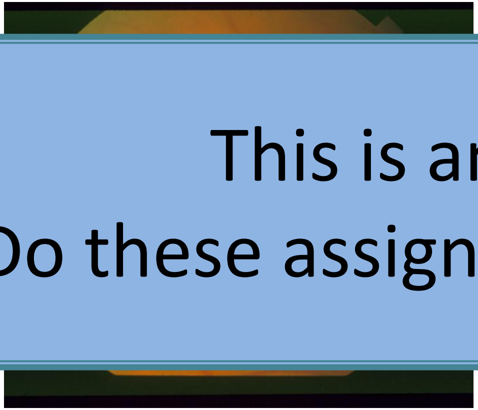
What samples are outsourced?

Drusen disease is characterized by **pathological yellow spots...**

...however, both images are given a **score of severity zero**

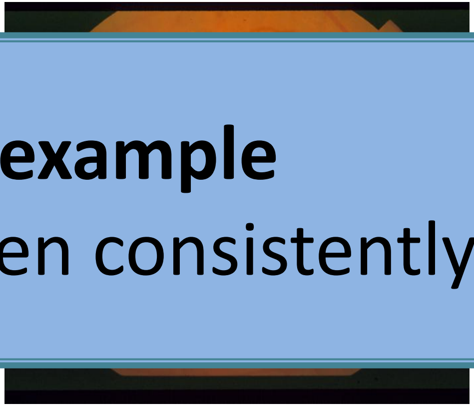
This is an **anecdotal example**

Do these assignments happen consistently?



Easy sample

our algorithm assigns it to machine



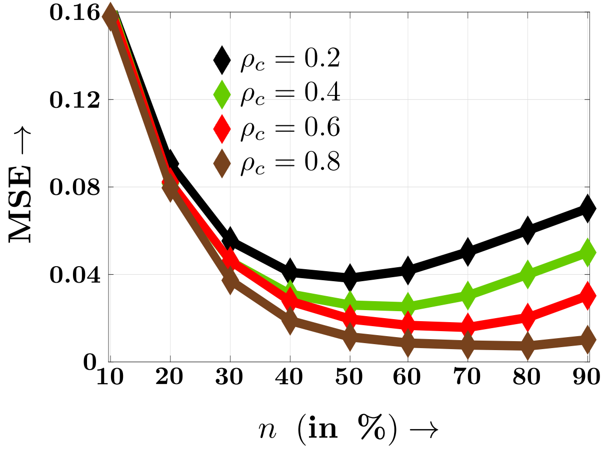
Difficult sample

our algorithm assigns it to human

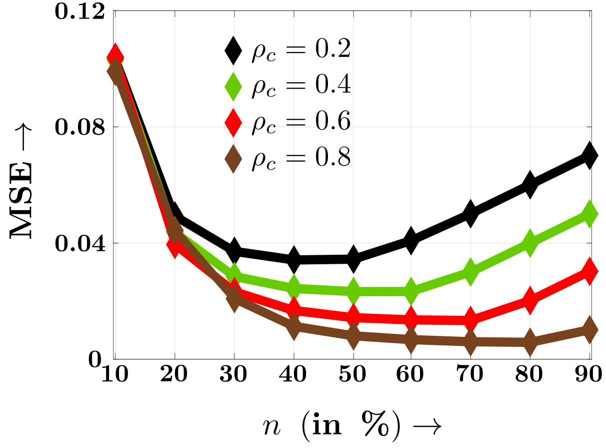
[De et al., AAAI 2020]

The greedy algorithm spots samples where humans are accurate

Stare-H dataset



Stare-D dataset



ρ_c: fraction of samples with low human error

As long as there are **samples that humans can predict with low error**, the **greedy algorithm outsources them to humans** and the **performance improves**

[De et al., AAAI 2020]

Strategic behavior and transparency

Until now, we have **assumed individuals** are **not strategic**:

→ **Individuals** do **not seek** to **maximize** their **benefit**

Ho
th
ab

Can we **design transparent ML** models that **account for strategic behavior?**

→ **Individuals** may use **knowledge,** gained by **transparency,** to **invest effort strategically** to **maximize** their **chances** of **receiving** a **beneficial decision.**

Transparency on predictive models and policies

We can be **transparent** about:

→ **Predictive models**

Goal: develop **accurate predictive models** under strategic behavior. Most work view strategic behavior as **gaming**.

[Brückner et al., JMLR 2012; Hardt et al., NIPS 2016; Dong et al., EC 2018; Hu et al., WWW 2019]

→ **Policies**

Goal: design **policies that maximize utility** under strategic behavior. Most work view strategic behavior as **self-improvement**.

[Kleinberg & Raghavan, EC 2019; Khajehnejad et al., Arxiv 2019]

Transparency on predictive models and policies

We can be **transparent** about:

→ **Predictive models**

Goal: develop **accurate predictive models** under strategic behavior. Most work view strategic behavior as **gaming**.

[Brückner et al., JMLR 2012; Hardt et al., NIPS 2016; Dong et al., EC 2018; Hu et al., WWW 2019]

→ **Policies**

Goal: design **policies that maximize utility** under strategic behavior. Most work view strategic behavior as **self-improvement**.

[Kleinberg & Raghavan, EC 2019; Khajehnejad et al., Arxiv 2019]

Causal vs non causal features

Individual's strategic behavior as **gaming** or **self-improvement?**

Individuals **invest effort** on **changing noncausal features**

$P(\boldsymbol{x})$ changes

$P(y \mid \boldsymbol{x})$ changes

Goodhart's law

"When a measure becomes a target, it ceases to be a good measure"

Individuals **invest effort** on **changing causal features**

$P(\boldsymbol{x})$ changes

$P(y \mid \boldsymbol{x})$ does not change

[Miller et al., Arxiv 2019]

Causal vs non causal features

Individual's strategic behavior as **gaming** or **self-improvement?**

Individuals **invest effort** on **changing noncausal features**

$P(\boldsymbol{x})$ changes

$P(y \mid \boldsymbol{x})$ changes

Goodhart's law

"When a measure becomes a target, it ceases to be a good measure"

Individuals **invest effort** on **changing causal features**

$P(\boldsymbol{x})$ changes

$P(y \mid \boldsymbol{x})$ does not change

[Miller et al., Arxiv 2019]

Example 1: car insurance decisions

1. Insurance company reveals it uses **the number of speeding tickets** to decide the **insurance premium**

2. Drivers may **drive more carefully** to **pay** a **lower price**

3. This will likely **make** them **better drivers**

causal feature
self-improvement

Example 2: loan decisions

1. A **bank reveals** it uses **credit card debt** to decide **loans interest rates**

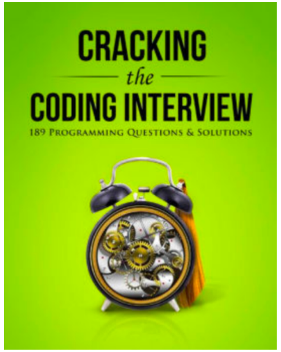
2. Applicants may **avoid credit card debt overall** to **pay less interest**

3. This will **improve** their **financial situation**

causal feature
self-improvement

Example 3: hiring decisions

1. A **software company publishes** the **coding exercises** it uses during **recruiting**



2. **Applicants** just **practice only those coding exercises**

3. This will not **necessarily make them better employees**

noncausal feature gaming

Full transparency: a Stackelberg game

Stackelberg game-theoretic formulation

The **decision maker publishes** the **decision policy** π before **individuals (best-)respond.**

For each **individual** with initial set of **features** $\boldsymbol{x}_i$ **, her best response** is:

$$\boldsymbol{x}_j = \underset{k \in [m]}{\operatorname{argmax}} \, \underbrace{b(\boldsymbol{x}_k, c)} - \underbrace{c(\boldsymbol{x}_i, \boldsymbol{x}_k)}$$

Benefit individual obtains for **having features** $\boldsymbol{x}_k$

Cost individual pays for **changing from** $\boldsymbol{x}_i$ **to** $\boldsymbol{x}_k$

We assume

$$b(\boldsymbol{x}, c) = \mathbb{E}_{d \sim \pi(d \,|\, \boldsymbol{x})}[f(d(\boldsymbol{x}), c)] \\ = \pi(\boldsymbol{x})$$

[Khajehnejad et al., Arxiv 2019]

From individual to population best response

Individual best response

$$\boldsymbol{x}_j = \underset{k \in [m]}{\operatorname{argmax}}\, b(\boldsymbol{x}_k, c) - c(\boldsymbol{x}_i, \boldsymbol{x}_k)$$

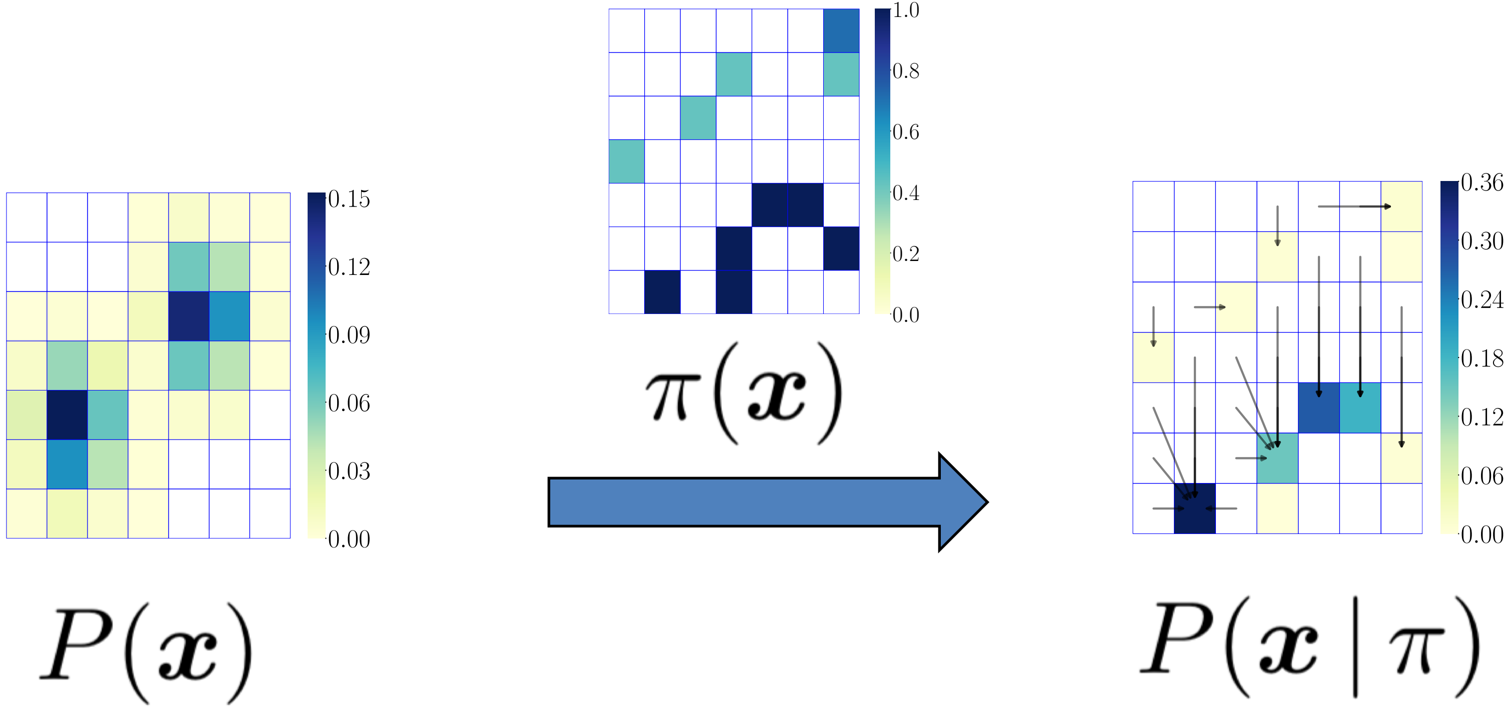
(Optimal) transportation of mass

Induced distribution

$$P(\boldsymbol{x}_j | \pi) = \sum_{i \in [m]} \underbrace{P(\boldsymbol{x}_i)\mathbb{I}(\boldsymbol{x}_j = \underset{k \in [m]}{\operatorname{argmax}}\, b(\boldsymbol{x}_k, c) - c(\boldsymbol{x}_i, \boldsymbol{x}_k))}_{\text{Flow between } P(\boldsymbol{x}_i) \text{ and } P(\boldsymbol{x}_j|\pi)}$$

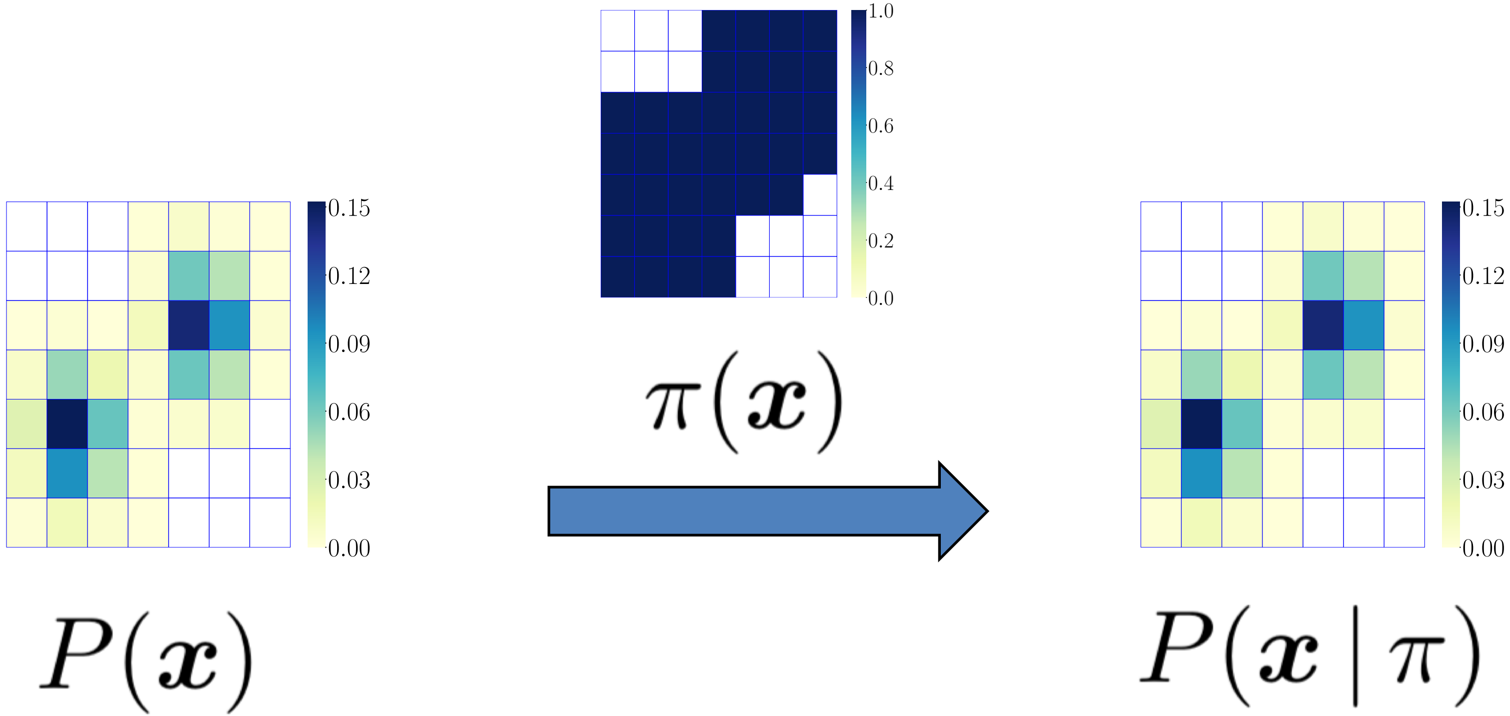
[Khajehnejad et al., Arxiv 2019]

Example 1: original and induced distributions



[Khajehnejad et al., Arxiv 2019]

Example 2: original and induced distributions



[Khajehnejad et al., Arxiv 2019]

Finding optimal decisions is hard

Finding the solution to

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, u(\pi, c)$$
$$= \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}_{\boldsymbol{x} \sim P(\boldsymbol{x} \,|\, \pi),\, d \sim \pi(d \,|\, \boldsymbol{x})} [P(y = 1 \,|\, \boldsymbol{x})\, d(\boldsymbol{x}) - c\, d(\boldsymbol{x})]$$

That makes it hard

is a **NP-hard problem**

Proof idea

Using a **reduction** to the **Boolean satisfiability (SAT) problem** [Karp, 1972]

[Khajehnejad et al., Arxiv 2019]

Optimal decisions may be stochastic

The **NP-hardness** result implies that **threshold rules** are **not always optimal.**

$$\pi^*(d=1\,|\,\boldsymbol{x}) = \begin{cases} 1 & \text{if } P(y=1\,|\,\boldsymbol{x}) \geq c \\ 0 & \text{otherwise.} \end{cases}$$

There are many **scenarios** in which the **optimal decision policies** are **not deterministic.** For example:

$$P(\boldsymbol{x}) = 0.1\,\mathbb{I}(x=1) + 0.4\,\mathbb{I}(x=2) + 0.5\,\mathbb{I}(x=3)$$
$$P(y=1\,|\,\boldsymbol{x}) = 1.0\,\mathbb{I}(x=1) + 0.7\,\mathbb{I}(x=2) + 0.4\,\mathbb{I}(x=3)$$

$$c(\boldsymbol{x}_i, \boldsymbol{x}_j) = \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.3 & 0.0 & 0.0 \\ 1.2 & 0.3 & 0.0 \end{bmatrix}$$

$$c = 0.1$$

Non-strategic: $\pi^*(d\,|\,\boldsymbol{x}) = 1$ for all $\boldsymbol{x}$

Strategic:
$$\pi^*(d=1\,|\,\boldsymbol{x}=1) = 1$$
$$\pi^*(d=1\,|\,\boldsymbol{x}=2) = 0.7$$
$$\pi^*(d=1\,|\,\boldsymbol{x}=3) = 0$$

[Khajehnejad et al., Arxiv 2019]

Highest outcome and negative outcomes

Given any **instance** of the **utility maximization problem under strategic behavior, it easy to realize that:**

$\pi^*(\boldsymbol{x}_1) = 1$ where $\boldsymbol{x}_1$ is the feature value with highest outcome $P(y \,|\, \boldsymbol{x}_1) \geq c$

→ The ***best individuals*** always receive a **beneficial decision**

$\pi^*(\boldsymbol{x}_i) = 0$ for all $\boldsymbol{x}_i$ such that $P(y \,|\, \boldsymbol{x}_i) < c$

→ Always **decide negatively** about **individuals** providing ***negative utility***

[Khajehnejad et al., Arxiv 2019]

Highest outcome and negative outcomes

Given any **instance** of the **utility maximization problem under strategic behavior, it easy to realize that:**

$\pi^*(\boldsymbol{x}_1) = 1$ where $\boldsymbol{x}_1$ is the feature value with highest

n

What about **individuals** in the **middle range** providing **positive utility**?

Always **decide negatively** about **individuals** providing ***negative utility***

[Khajehnejad et al., Arxiv 2019]

Outcome monotonic costs

We can **further characterize** a **family of optimal policies** if the **cost individuals pay to change features** satisfies **natural property**, **outcome monotonicity, i.e.,**

[Improving an individual's outcome requires increasing amount of effort]

$$P(y=1\,|\,\boldsymbol{x}_i) < P(y=1\,|\,\boldsymbol{x}_j) < P(y=1\,|\,\boldsymbol{x}_k) \Leftrightarrow c(\boldsymbol{x}_i, \boldsymbol{x}_j) < c(\boldsymbol{x}_i, \boldsymbol{x}_k)$$

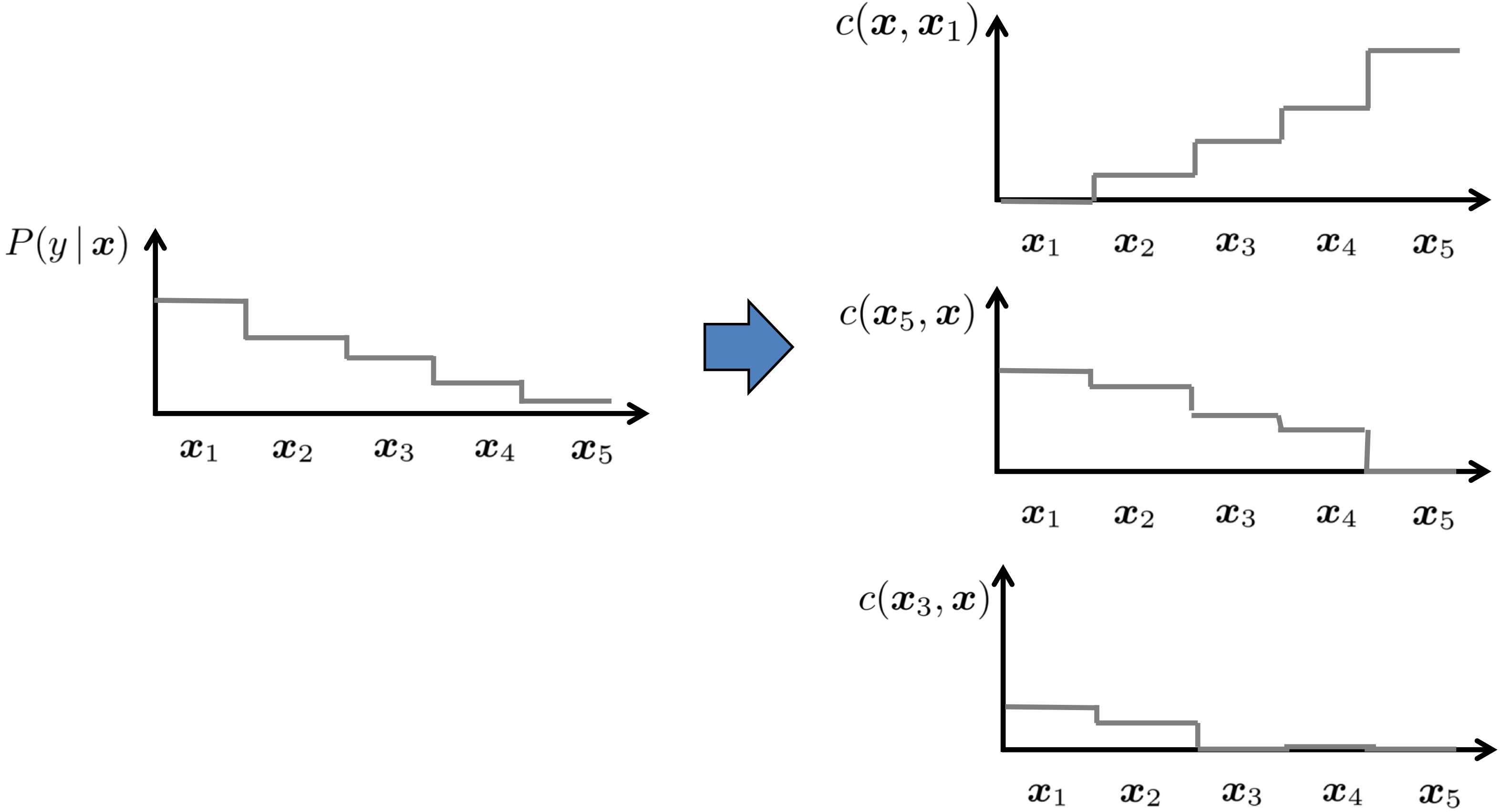
$$P(y=1\,|\,\boldsymbol{x}_i) > P(y=1\,|\,\boldsymbol{x}_j) > P(y=1\,|\,\boldsymbol{x}_k) \Leftrightarrow c(\boldsymbol{x}_j, \boldsymbol{x}_i) < c(\boldsymbol{x}_k, \boldsymbol{x}_i)$$

[Worsening an individual's outcome requires no effort]

$$P(y=1\,|\,\boldsymbol{x}_i) > P(y=1\,|\,\boldsymbol{x}_j) \Leftrightarrow c(\boldsymbol{x}_i, \boldsymbol{x}_j) = 0$$

[Hardt et al., NIPS 2016; Hu et al., FAT* 2019]

Example: outcome monotonic costs



Outcome monotonic policies

Proposition (positive result!). If costs are **outcome monotonic,** there exists an **outcome monotonic policy** that is **optimal in terms of utility.**

An **outcome monotonic policy** satisfies that:

$$P(y=1 \,|\, \boldsymbol{x}_i) < P(y=1 \,|\, \boldsymbol{x}_j) \Leftrightarrow \pi(\boldsymbol{x}_i) < \pi(\boldsymbol{x}_j)$$

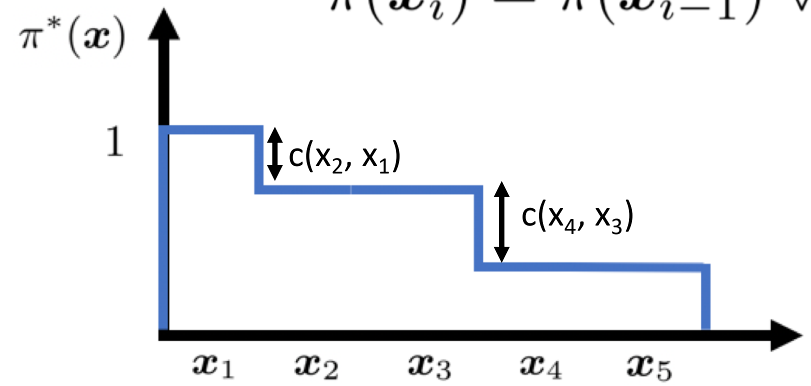
Better individuals are **more likely** to receive a **beneficial decision** ⇒ **Fair property**

[Khajehnejad et al., Arxiv 2019]

Outcome monotonic binary policies (I)

$$c(\boldsymbol{x}_i, \boldsymbol{x}_j) = c(\boldsymbol{x}_i, \boldsymbol{x}_k) + c(\boldsymbol{x}_k, \boldsymbol{x}_j) \longleftrightarrow$$ Unidimensional features

If costs are additive, there exists an optimal outcome monotonic "binary" policy that satisfies that:



$$\pi(\boldsymbol{x}_i) = \pi(\boldsymbol{x}_{i-1}) \vee \pi(\boldsymbol{x}_i) = \pi(\boldsymbol{x}_{i-1}) - c(\boldsymbol{x}_i, \boldsymbol{x}_{i-1})$$

$$i < j \Rightarrow P(y \,|\, \boldsymbol{x}_i) \geq P(y \,|\, \boldsymbol{x}_j)$$

$$P(y \,|\, \boldsymbol{x}_i) \geq 0$$

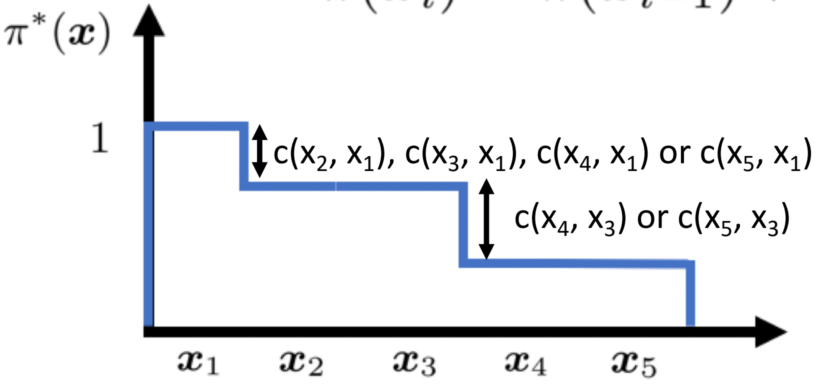
[Khajehnejad et al., Arxiv 2019]

Outcome monotonic binary policies (II)

$c(\boldsymbol{x}_i, \boldsymbol{x}_j) + c(\boldsymbol{x}_j, \boldsymbol{x}_k) \geq c(\boldsymbol{x}_i, \boldsymbol{x}_k)$ $\longleftrightarrow$ Multidimensional features

If costs are subadditive, there exists an optimal outcome monotonic "binary" policy that satisfies that:

$$\pi(\boldsymbol{x}_i) = \pi(\boldsymbol{x}_{i-1}) \vee \bigvee_j \pi(\boldsymbol{x}_i) = \pi(\boldsymbol{x}_{i-1}) - c(\boldsymbol{x}_j, \boldsymbol{x}_{i-1})$$



$$i < j \Rightarrow P(y \mid \boldsymbol{x}_i) \geq P(y \mid \boldsymbol{x}_j)$$

$$P(y \mid \boldsymbol{x}_i) \geq 0$$

[Khajehnejad et al., Arxiv 2019]

An iterative algorithm for general costs

$\pi \leftarrow \textsc{InitializePolicy}$
repeat
    $\pi' \leftarrow \pi$
    for $i = 1, \ldots, m$ **do**
        $\pi(\boldsymbol{x}_i) \leftarrow \mathrm{SOLVE}(i, \pi, \boldsymbol{C}, \boldsymbol{P}, \boldsymbol{Q})$
    end for
until $\pi = \pi'$
Return $\pi', u(\pi', c)$

$[c(\boldsymbol{x}_i, \boldsymbol{x}_j)]$ $[P(y \,|\, \boldsymbol{x}_i)]$

$[P(\boldsymbol{x}_i)]$

Tractable problem

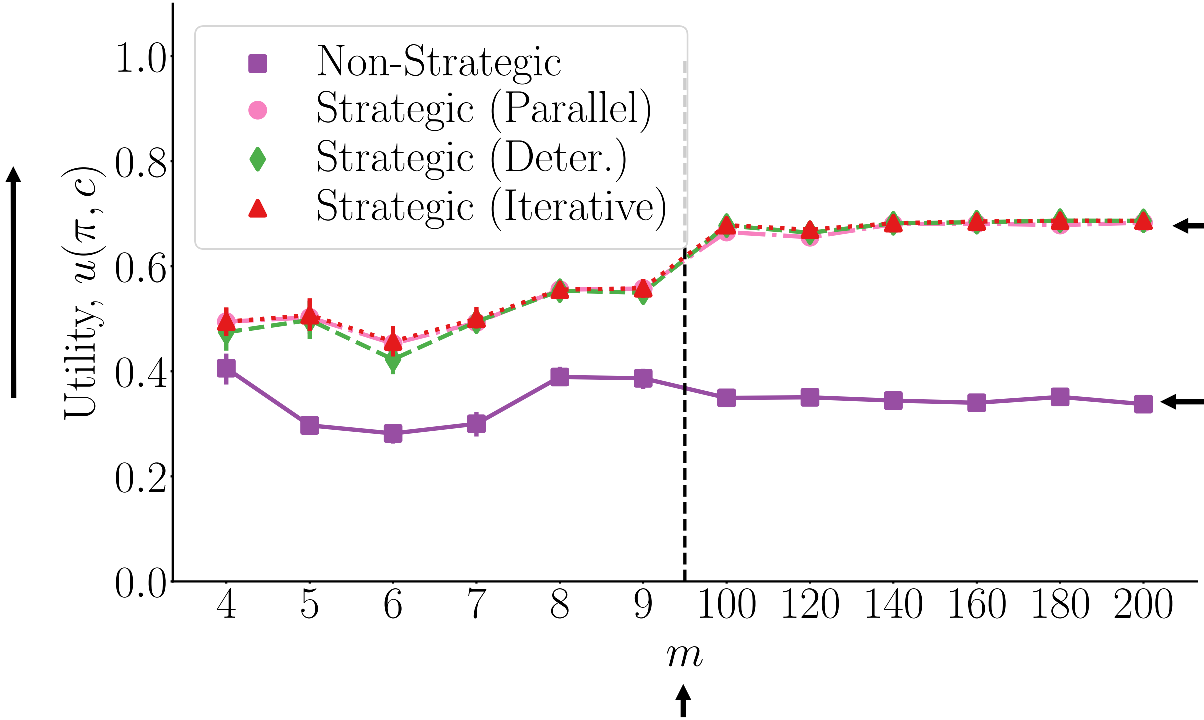
Fix all $\pi(\boldsymbol{x}_k)$ **with** $\boldsymbol{x}_k \neq \boldsymbol{x}_i$ **and find best** $\pi(\boldsymbol{x}_i)$

The **iterative algorithm** is **guaranteed** to **terminate** in **polynomial time** and **find** a is **locally optimal policy**

[Khajehnejad et al., Arxiv 2019]

Example: utility under strategic behavior

Higher is better



Policies accounting for strategic behavior

Optimal policy under nonstrategic behavior

Number of feature values

[Khajehnejad et al., Arxiv 2019]

Beyond full transparency

We have **assumed the decision maker publishes the entire policy**.

In practice, it will reveal only **part of the policy** to **each individual.** For example:

Counterfactual explanations

[Wachter et al., Harvard JL and Tech 2017; Ustun et al., FAT* 2019]

→ **Reveal** one **example** of **feature value "close" to the original feature value**, which would lead to a **beneficial decision**

Beyond full transparency

We have **assumed the decision maker publishes the entire policy**.

In practice, it will reveal only **part of the policy to each** i…

Open problem: find **counterfactual explanations** under strategic behavior

→ **Reveal** one **example** of **feature value "close" to the original feature value**, which would lead to a **beneficial decision**

Summary of the lecture

Under a simple problem setting, we have learned about a few machine learning models and methods to:

- **Account for the feedback loop between algorithmic and human decisions**
- **Balance decisions between human and algorithmic decisions**
- **Account for strategic human decisions**

Disclaimer. This was a ***biased view***. These are emerging topics and there is still a lot of open problems and research directions! Join us ☺

Thanks!

more at learning.mpi-sws.org