Equal Opportunity in Online Classification with Partial Feedback

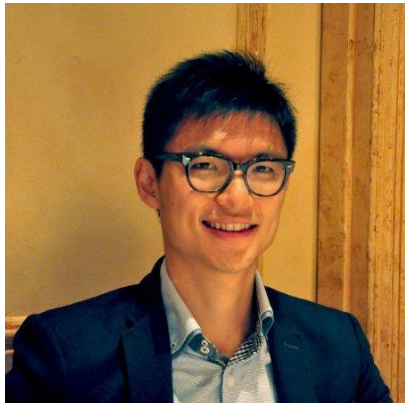
Yahav Bechavod, Hebrew University

Simons Institute, July 2019

Katrina Ligett
Hebrew University

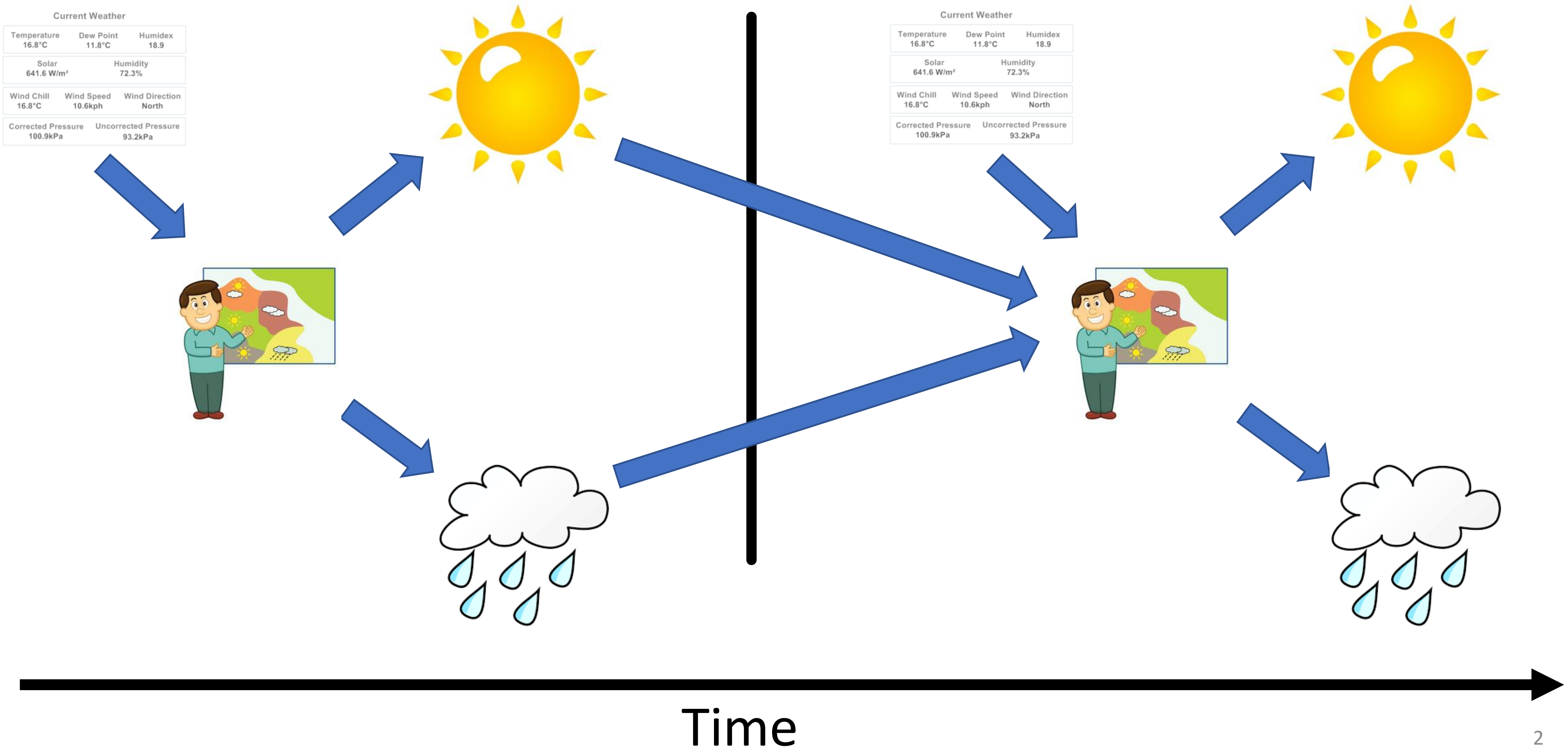
Aaron Roth
University of Pennsylvania

Bo Waggoner
University of Colorado, Boulder



Steven Wu
University of Minnesota

Online Classification with Full Feedback



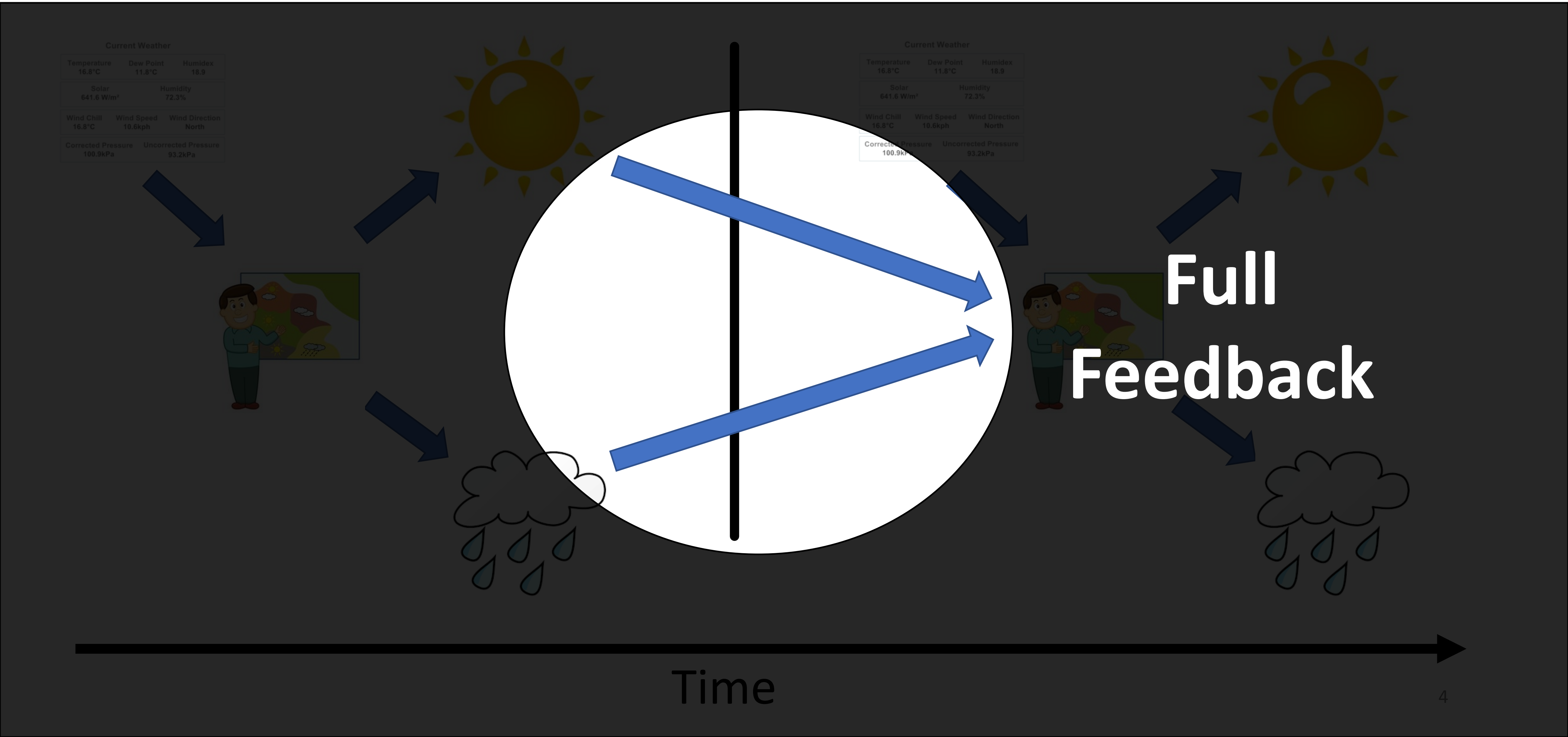
Kim Jong Un warns weather men for incorrect forecasts

June 12, 2014

By Associated Press

SEOUL, SOUTH KOREA

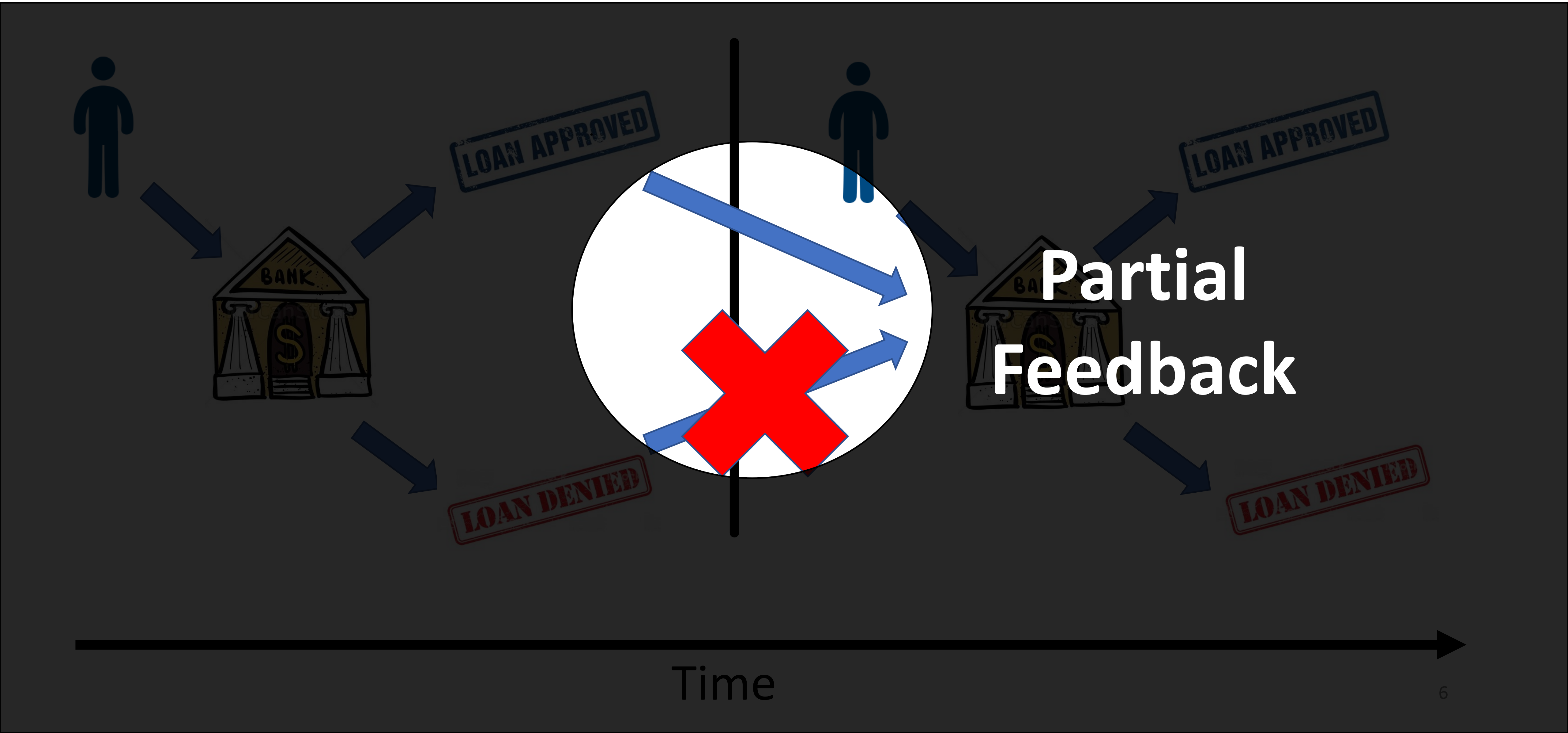
Online Classification with Full Feedback



The Majority of settings where fairness is a primary concern are **Partial Feedback**.

- Lending
- Hiring
- College Admissions
- Recidivism prediction
- Online advertising
- Predictive policing
- Medical treatments

Online Classification with Partial Feedback



Decisions not only affect how accurate we are, but also the **amount and type of data we collect**.

Standard techniques on gathered data may lead to **feedback loops**. Risk being highly unfair.

MOTHERBOARD | By **Caroline Haskins** | Feb 14 2019, 9:57am

TECH BY VICE

Academics Confirm Major Predictive Policing Algorithm is Fundamentally Flawed

PredPol uses an algorithm based on earthquake prediction to “predict crime.” Academics say it’s simplistic and harmful.

Problem Setting: Online Classification with One-Sided Feedback

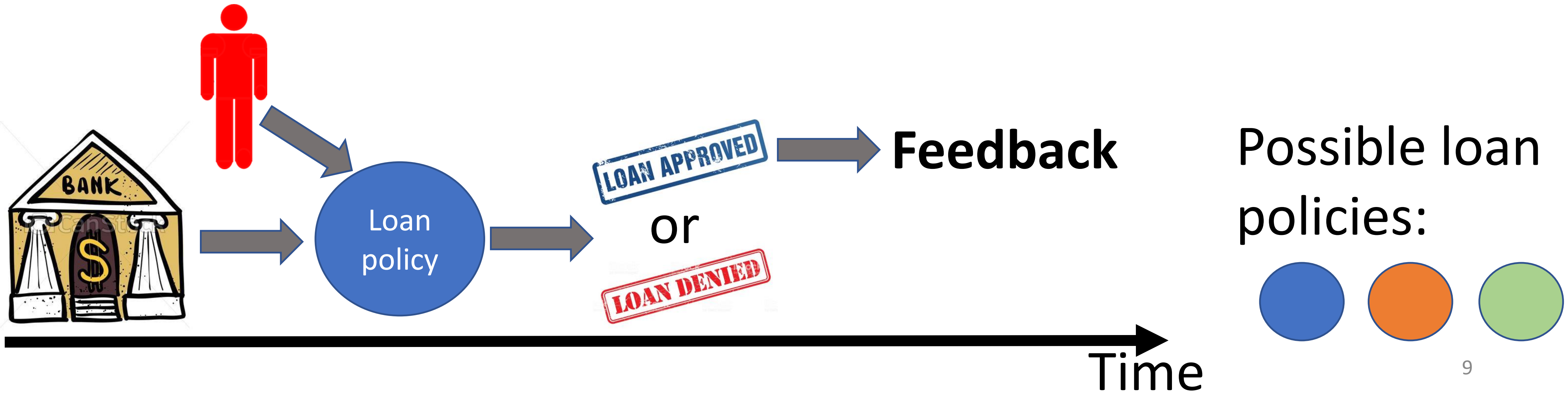
For $t = 1, ..., T$:

Learner selects policy $h_t \in \mathcal{H}$.

Environment draws $(x_t, a_t, y_t) \sim \mathcal{D}$; learner observes x_t, a_t.

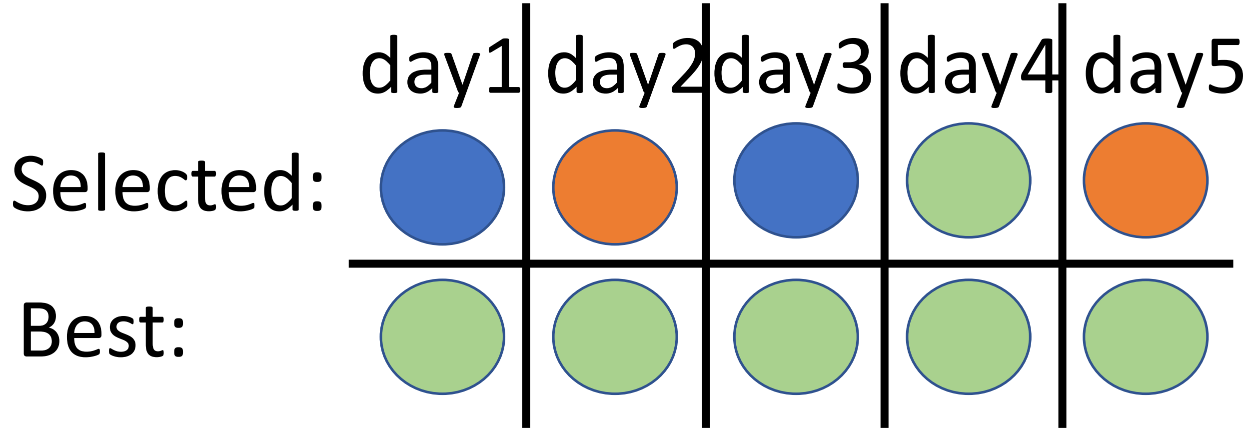
Learner predicts $\hat{y}_t = h_t(x_t)$.

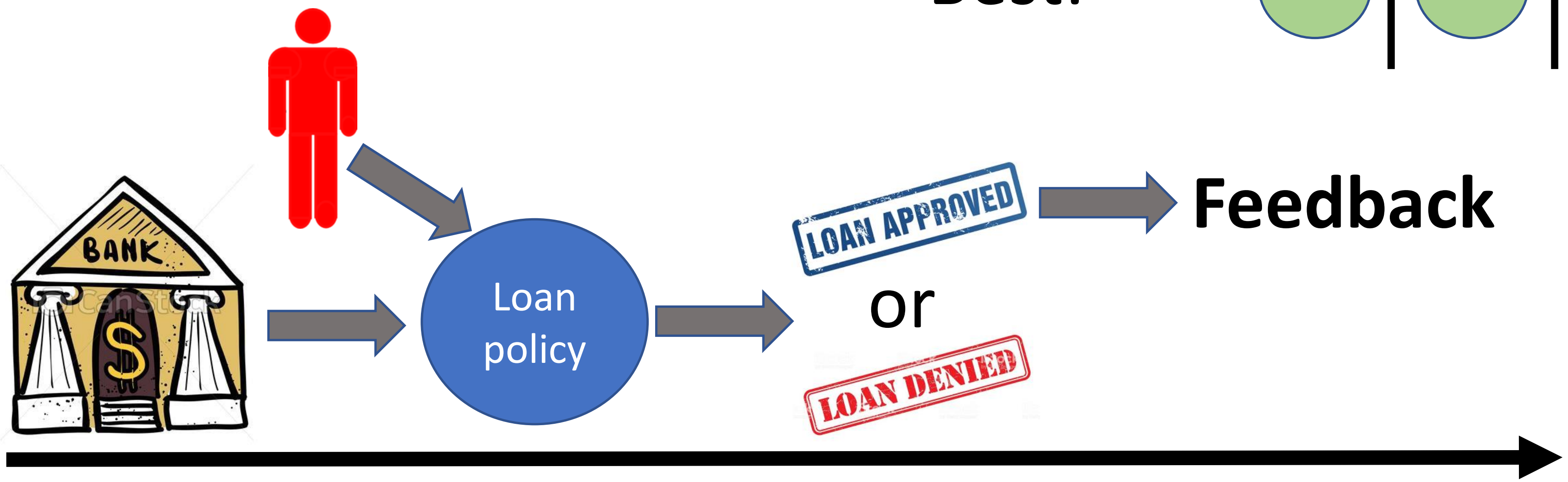
If $\hat{y}_t = +1$, learner observes y_t.



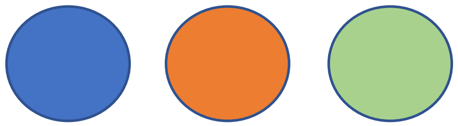
Problem Setting: Online Classification with One-Sided Feedback

"How well did I do compared to **best available policy**?"





Possible loan policies:



Learner’s Goal – Minimize Regret

Optimal policy:

$$h^* = \arg\min_{h \in \mathcal{H}} \sum_{t=1}^{T} \mathop{\mathbb{E}}_{(x_t, y_t) \sim \mathcal{D}} [\ell(h^*(x_t), y_t)]$$

Learner’s (pseudo) regret:

$$Regret(T) = \sum_{t=1}^{T} \mathop{\mathbb{E}}_{(x_t, y_t) \sim \mathcal{D}} [\ell(h_t(x_t), y_t)] \;-\; \sum_{t=1}^{T} \mathop{\mathbb{E}}_{(x_t, y_t) \sim \mathcal{D}} [\ell(h^*(x_t), y_t)]$$

Talk Outline

1. Low regret with one-sided feedback.

2. What about fairness?

3. Fairness + one-sided feedback.
    - Algorithm
    - Lower bound

Talk Outline

1. Low regret with one-sided feedback.

2. What about fairness?

3. Fairness + one-sided feedback.

- Algorithm
- Lower bound

Warmup Question: Can we guarantee low regret despite only having one-sided feedback?

From One-Sided Feedback to Contextual Bandits

Add $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ **to second column**

$$L = \begin{array}{c} \\ \text{Approve} \\ \text{Deny} \end{array} \begin{array}{cc} \text{Repays} & \text{Defaults} \\ \left[\begin{array}{c} 0 \\ 1 \end{array}\right. & \left.\begin{array}{c} 1 \\ 0 \end{array}\right] \end{array}$$

$$\tilde{L} = \begin{array}{c} \\ \text{Approve} \\ \text{Deny} \end{array} \begin{array}{cc} \text{Repays} & \text{Defaults} \\ \left[\begin{array}{c} 0 \\ 1 \end{array}\right. & \left.\begin{array}{c} 2 \\ 1 \end{array}\right] \end{array}$$

From One-Sided Feedback to Contextual Bandits

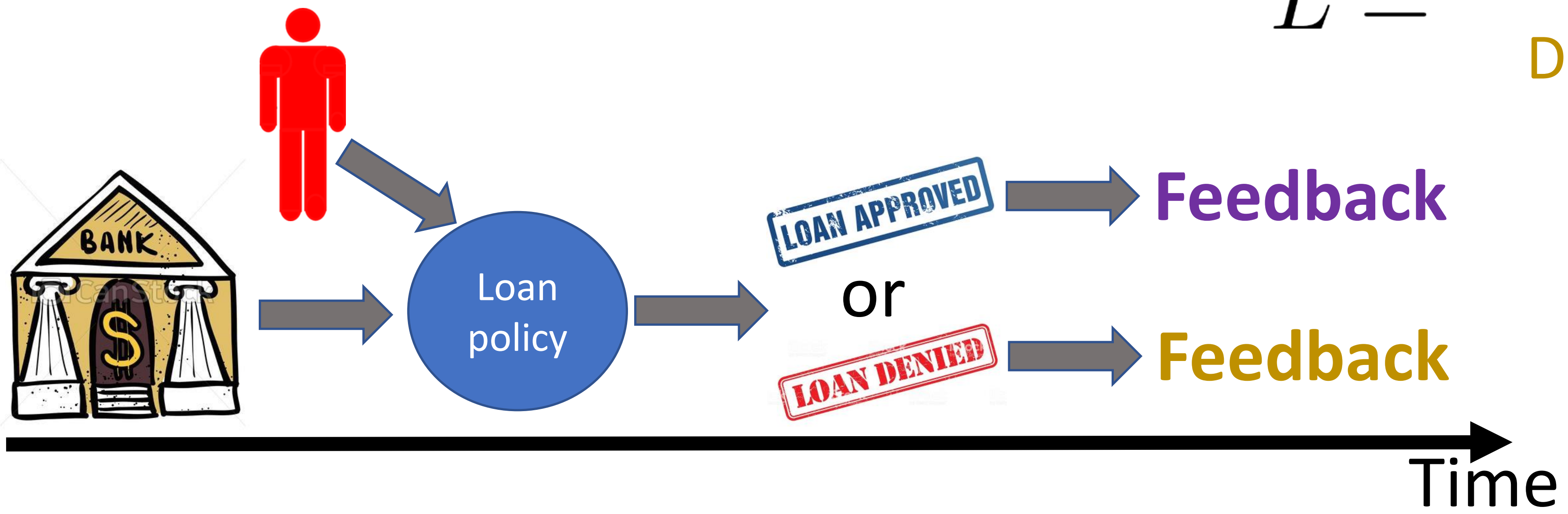
Contextual Bandits

2 actions: Approve, Deny

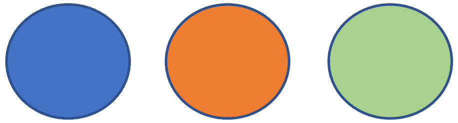
Contexts: Individuals

Policy class: Mappings from contexts to actions

$$\tilde{L} = \begin{array}{c} \\ \text{Approve} \\ \text{Deny} \end{array} \begin{array}{c} \begin{array}{cc} \text{Repays} & \text{Defaults} \end{array} \\ \begin{bmatrix} 0 & 2 \\ 1 & 1 \end{bmatrix} \end{array}$$



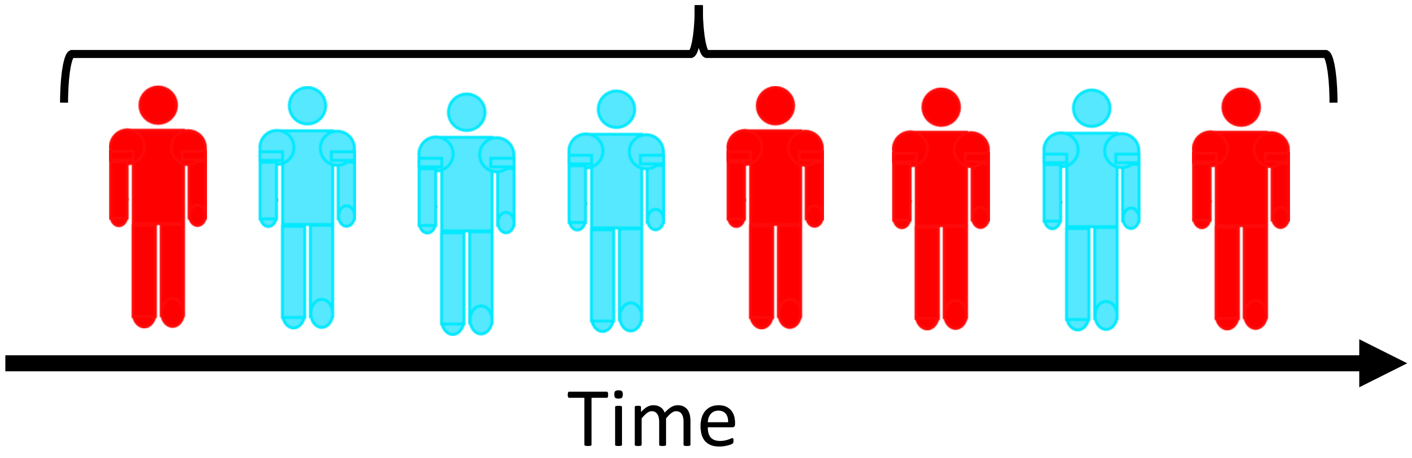
Possible loan policies:



From One-Sided Feedback to Contextual Bandits

Loss matrix transformation is **Regret-Preserving.**

$$S = \{(x_i, y_i)\}_{i=1}^{T}$$



$$\tilde{L} = \begin{array}{r} \\ \text{Approve} \\ \text{Deny} \end{array} \begin{array}{cc} \text{Repays} & \text{Defaults} \\ \hline 0 & 2 \\ 1 & 1 \end{array}$$

$$\forall h : \tilde{L}(h, S) = L(h, S) + \sum_{i=1}^{T} \mathbb{1}_{[y_i = -1]}$$

From One-Sided Feedback to Contextual Bandits

Conclusion: Given a contextual bandit algorithm that guarantees $Regret(T)$ w.h.p., we can translate it to an One-Sided Feedback algorithm that guarantees $2Regret(T)$ w.h.p.

Talk Outline

1. Low regret with one-sided feedback.

2. What about fairness?

3. Fairness + one-sided feedback.
   - Algorithm
   - Lower bound

Online Learning setting with Partial Feedback

For $t = 1, ..., T$:

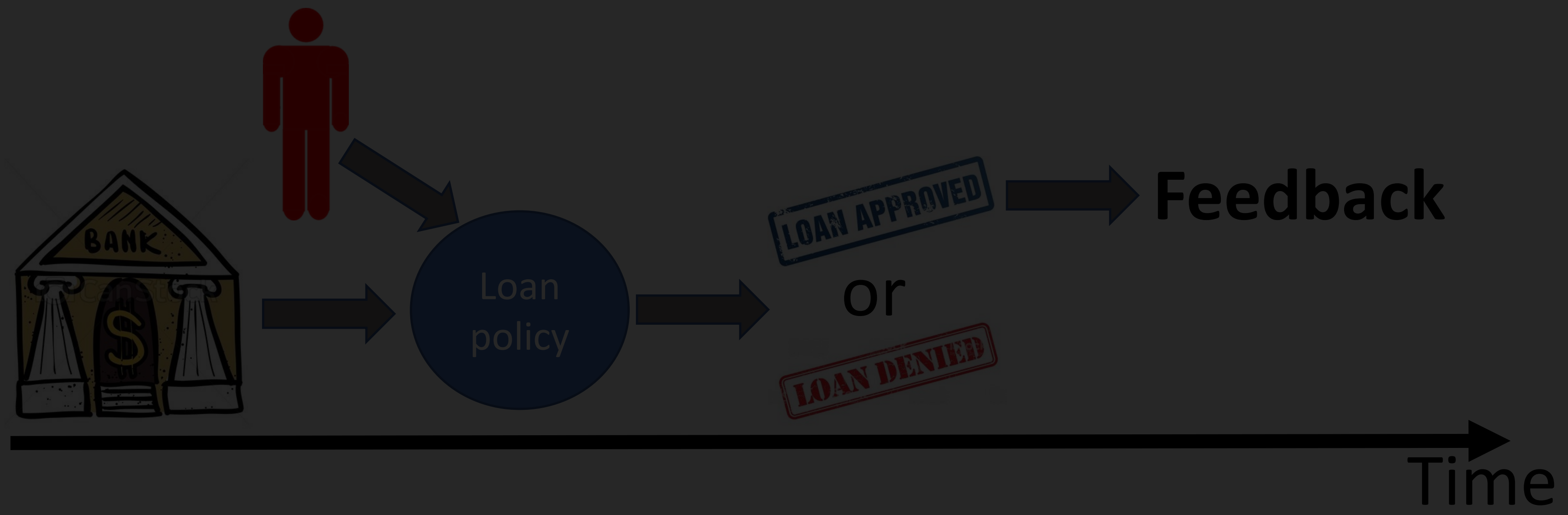
Learner selects policy $h_t \in \mathcal{H}$.

Fairness?

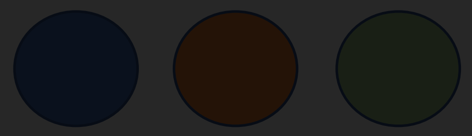
Environment draws $(x_t, a_t, y_t) \sim \mathcal{D}$; learner observes x_t, a_t.

Learner predicts $\hat{y}_t = h_t(x_t)$.

If $\hat{y}_t = +1$, learner observes y_t.



Possible loan policies:



Fairness

Question: Is the policy we deploy at every round fair?

Randomization. We allow deploying $\pi \in \Delta(\mathcal{H})$.

Definition. False positive rate:

$$FPR_j(\pi) = \mathop{\mathbb{E}}_{h \sim \pi}\left[\mathop{\mathbb{P}}_{(x,y)\sim\mathcal{D}}(h(x) = +1 | a = j, y = -1)\right]$$

$$\Delta_{FPR}(\pi) := FPR_1(\pi) - FPR_{-1}(\pi)$$

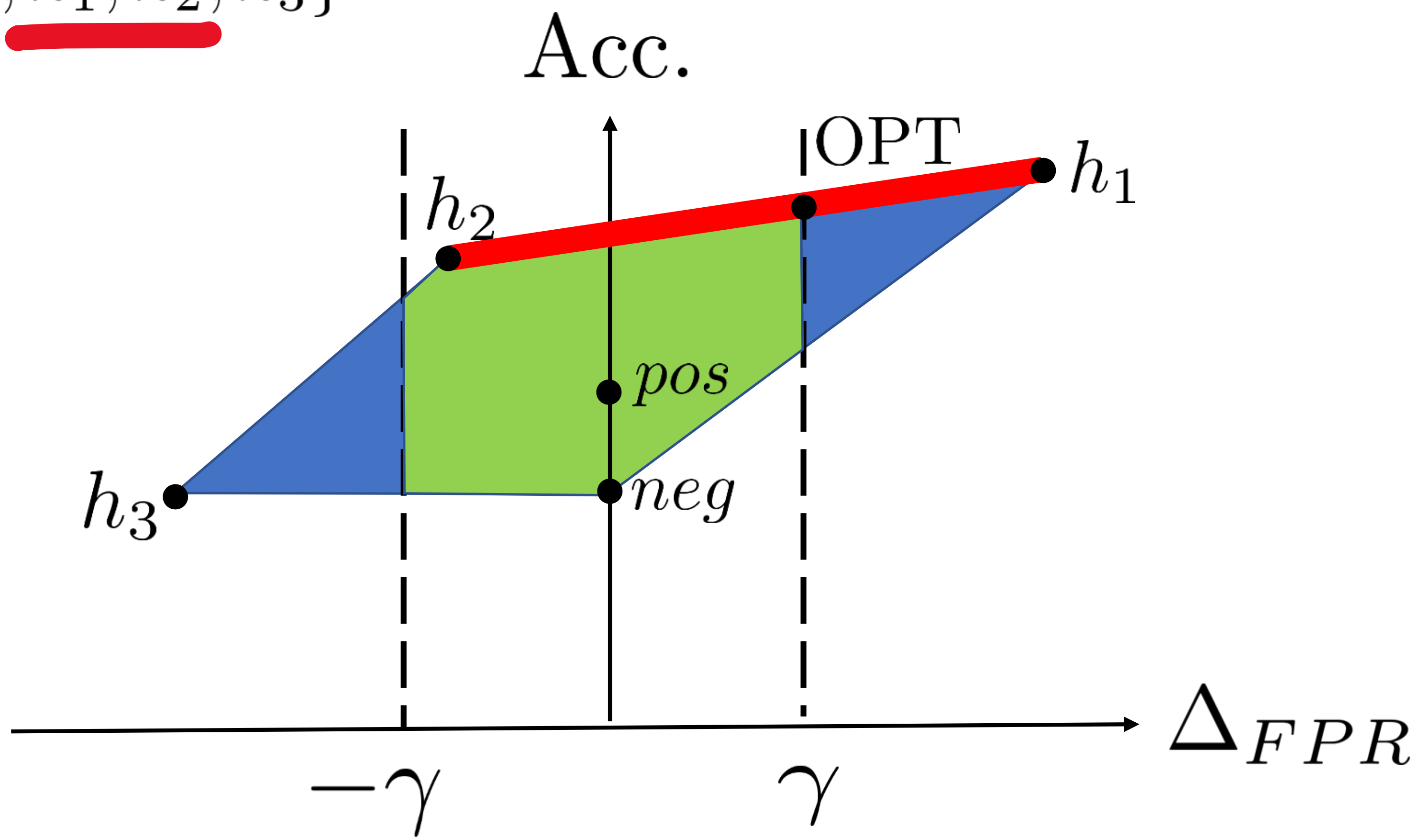
Definition. We say an algorithm is γ**-fair** if:

$$\forall t : |\Delta_{FPR}(\pi^t)| \leq \gamma$$

Example

Optimal γ-fair policy $\pi \in \Delta(\mathcal{H})$ is always of **support size at most 2**.

$\mathcal{H} = \{pos, neg, h_1, h_2, h_3\}$



Partial Feedback + Fairness

Question: What is the tradeoff between fairness and regret in the partial feedback setting?

More specifically: Regret guarantee if algorithm has to be γ-fair on every round?

Talk Outline

1. Low regret with one-sided feedback.

2. What about fairness?

3. Fairness + one-sided feedback.

- **Algorithm**
- Lower bound

Main Result: Oracle-efficient fair online learning algorithm

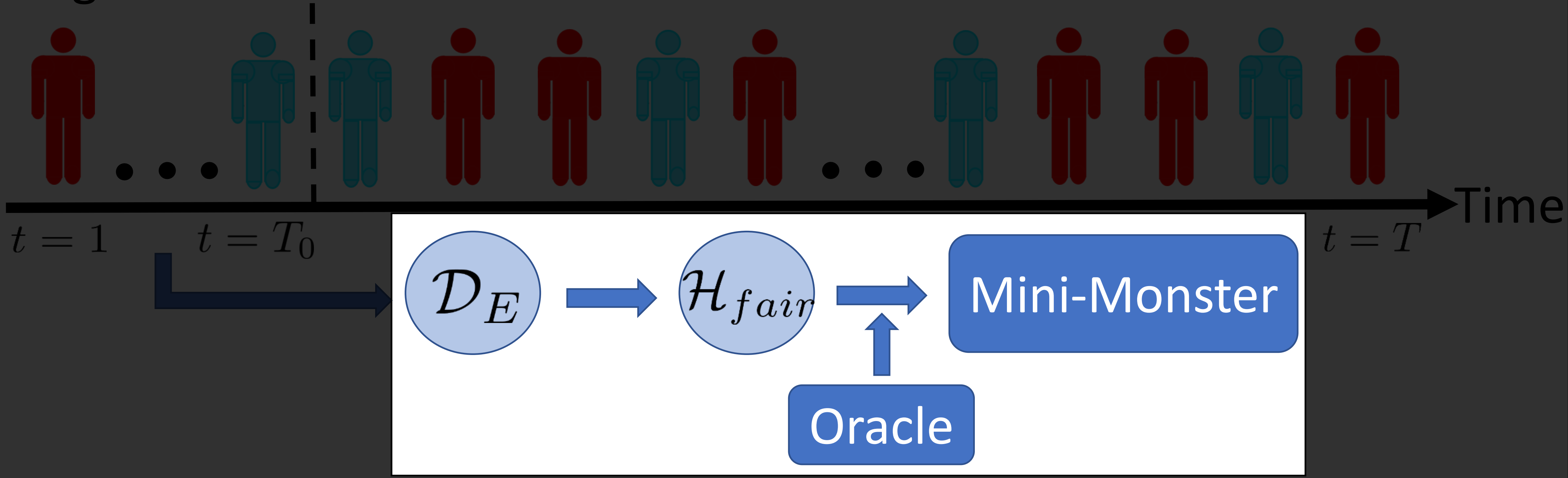
There exists an algorithm that runs in polynomial time given access to optimization oracle over $\mathcal{H}$ and guarantees:

1. Fairness: $\gamma + T^{-\frac{1}{4}}$-fair on every round.
2. Regret: $\tilde{O}(\sqrt{Tln(\mathcal{H})})$ to best γ-fair policy in $\mathcal{H}$.

Oracle-efficient algorithm

Agarwal et al. 2014 – “Mini-Monster”
Oracle-efficient algorithm
High probability guarantees for contextual bandits

Algorithm



1. Label first $T_0 = \Theta(\sqrt{T ln|\mathcal{H}|})$ arrivals as $\hat{y}_t = +1$, observe labels.

2. Instantiate mini-monster with policy class:

$$\mathcal{H}_{fair} = \{\pi \in \Delta(\mathcal{H}) : \Delta_{FPR}(\pi, \mathcal{D}_E) \le \gamma\}$$

3. Label remaining arrivals according to the instantiated algorithm. Use loss matrix transformation for feedback.

Cost-Sensitive Classification (CSC) Oracle

Step 1: CSC Oracle

Given: $\{(x_i, c_i^{(-1)}, c_i^{(+1)})\}_{i=1}^n$ **Compute:** $\arg\min_{h\in\mathcal{H}} \sum_{i=1}^{n} c_i^{h(x_i)}$

Equivalent to weighted binary classification

CSC Oracle -> Fair CSC Oracle

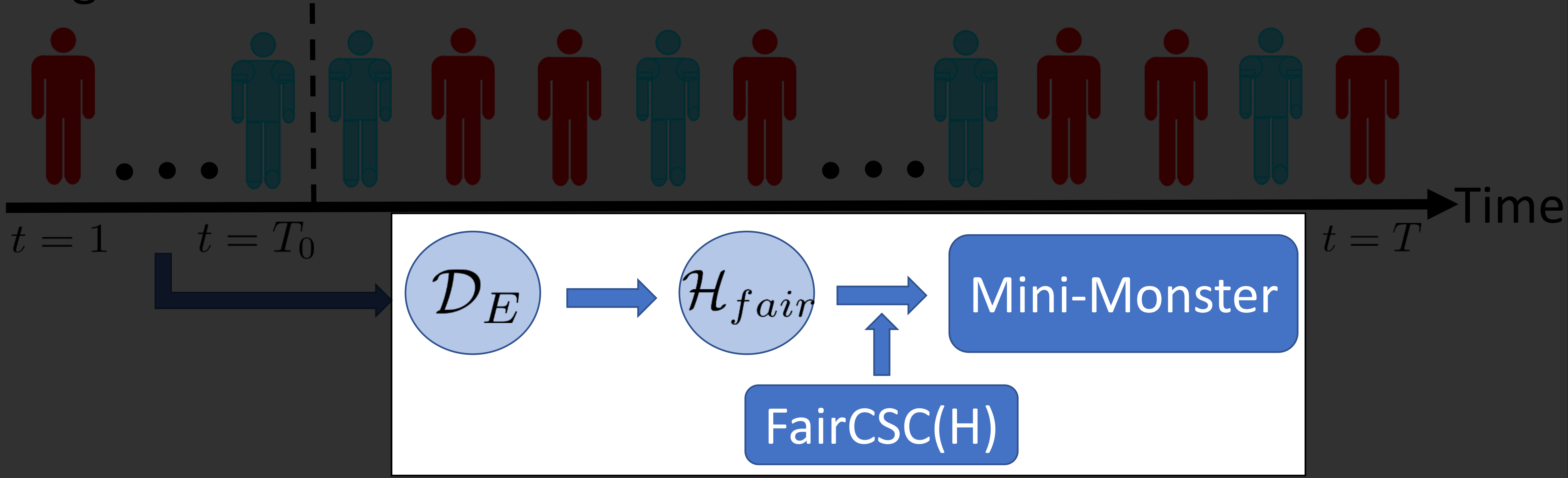
Step 2: Fair CSC Oracle

Theorem: Let $0 < \nu < \gamma/2$. There exists an oracle-efficient algorithm that calls $CSC(\mathcal{H})$ at most $O(1/\nu^2)$ times, and outputs a γ-fair $\pi \in \Delta(\mathcal{H})$ such that:

$$\mathop{\mathbb{E}}_{h \sim \pi}\left[\sum_{i=1}^{n} c_i^{h(x_i)}\right] \leq OPT + \nu$$

Adapted from Agarwal et al. 2018

Algorithm



1. Label first $T_0 = \Theta(\sqrt{T ln|\mathcal{H}|/\delta})$ arrivals as $\hat{y}_t = +1$, observe labels.
2. Instantiate mini-monster with policy class:

$$\mathcal{H}_{fair} = \{\pi \in \Delta(\mathcal{H}) : \Delta_{FPR}(\pi, \mathcal{D}_E) \leq \gamma\}$$

3. Label remaining arrivals according to the instantiated algorithm. Use loss matrix transformation for feedback.

In the paper

1. Adapting the CSC(H)->Fair CSC(H) construction to the case where the fairness constraint is only defined on a subset of the points considered in the cost objective.

2. Regret analysis for a fair version of Mini-Monster, also taking into account additional approximation error induced by fairness constraints.

Talk Outline

1. Low regret with one-sided feedback.

2. What about fairness?

3. Fairness + one-sided feedback.

- Algorithm
- **Lower bound**

Lower bound

Claim (simplified): There exists a hypothesis class $\mathcal{H}$ such that any algorithm that is $T^{-\alpha}$-fair must have expected regret $\Omega(T^{2\alpha})$.

Proof Idea

1. Two similar distributions $\mathcal{D}_1, \mathcal{D}_2$.

2. Until it is able to distinguish $\mathcal{D}_1, \mathcal{D}_2$, algorithm has to act conservatively, otherwise risks being unfair.

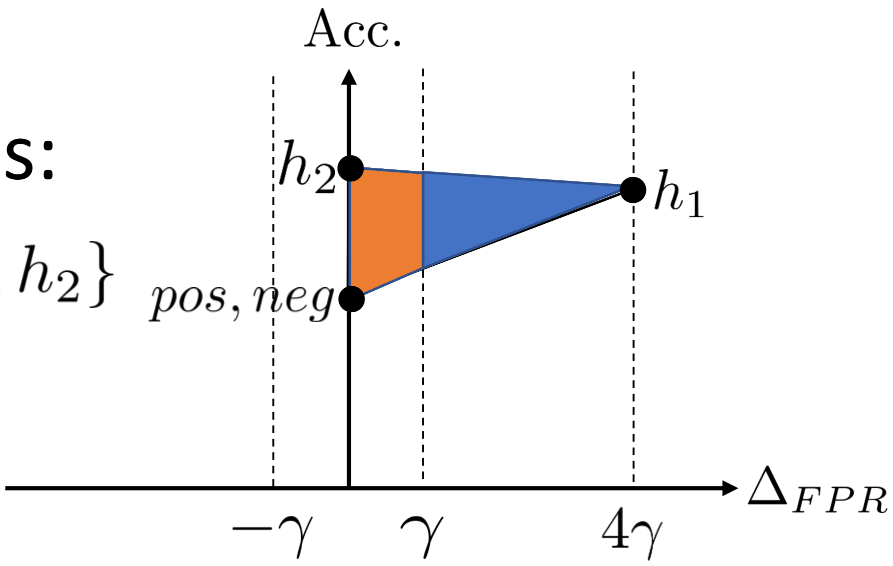
3. Acting conservatively in the first rounds forces high regret on each of these rounds.

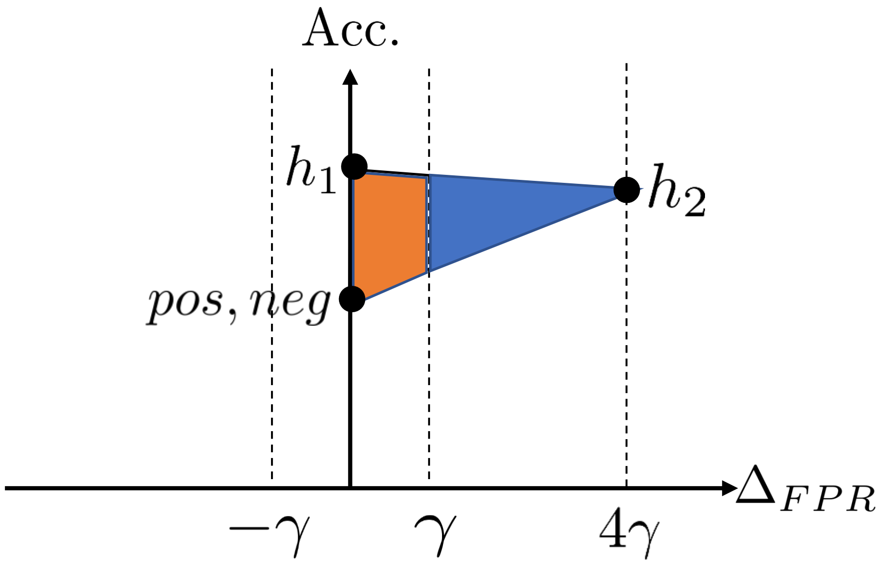
Proof Idea

Two similar distributions: D_1 $KL(\mathcal{D}_1 || \mathcal{D}_2) = O(\gamma^2)$ D_2

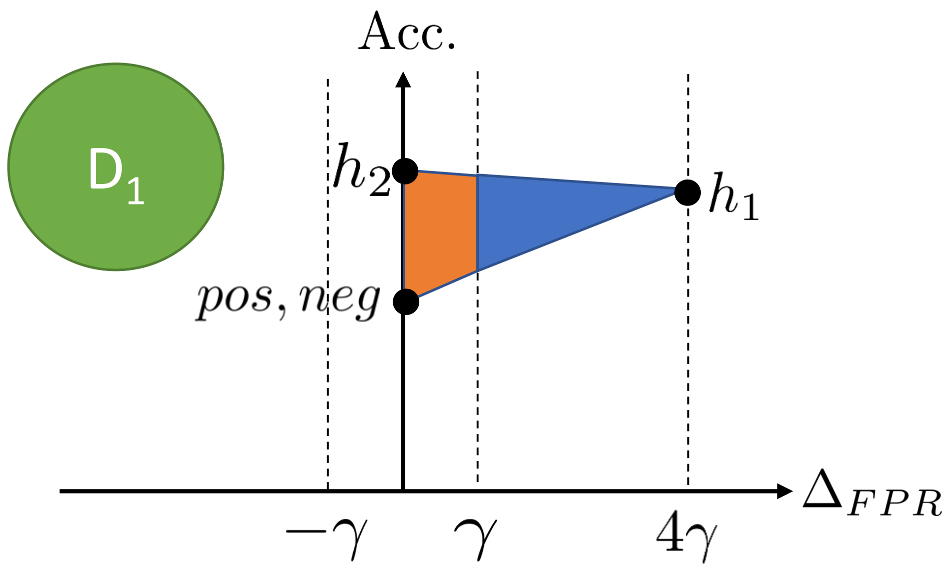
Hypothesis class:

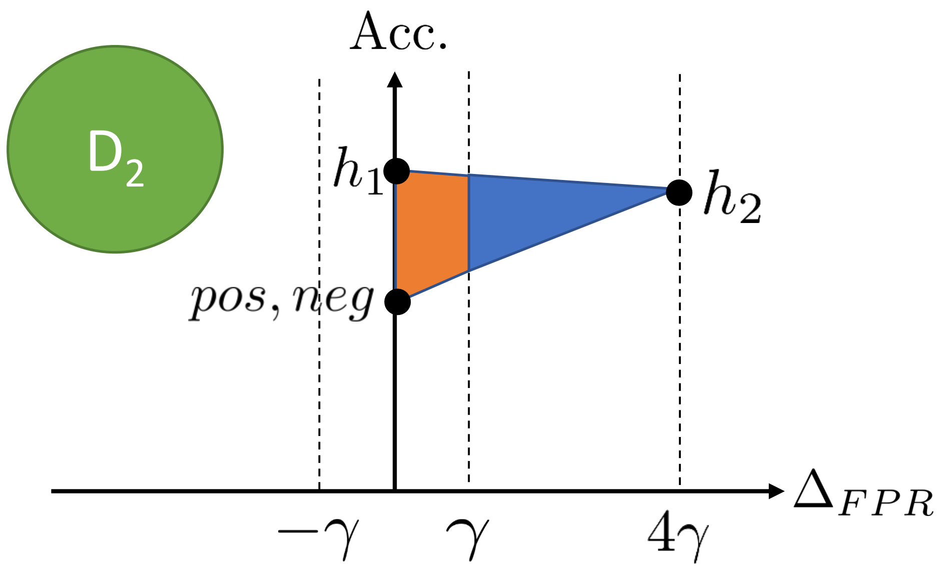
$\mathcal{H} = \{pos, neg, h_1, h_2\}$





Proof Idea





Alg. Is γ-fair:

$\forall t : P_t(h_1, \mathcal{D}_1) \leq \frac{1}{4}$

$\forall t : P_t(h_2, \mathcal{D}_2) \leq \frac{1}{4}$

For $\Omega(\frac{\beta}{\gamma^2})$ rounds:

$P_t(h_2, \mathcal{D}_1) \leq \frac{1}{4} + \beta$

$P_t(h_1, \mathcal{D}_2) \leq \frac{1}{4} + \beta$

Linear regret for $\Omega(\frac{\beta}{\gamma^2})$ rounds.

Open problems

1. Both equal FP, FN constraints.

2. Other definitions of fairness in the one-sided feedback setting.

Thank you!

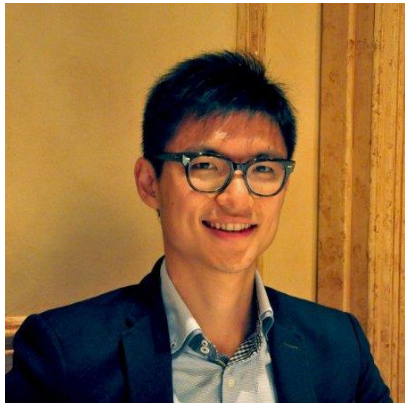
Yahav Bechavod, Hebrew University

yahav.bechavod@cs.huji.ac.il

Katrina Ligett
Hebrew University

Aaron Roth
University of Pennsylvania

Bo Waggoner
University of Colorado, Boulder



Steven Wu
University of Minnesota