Safe and Fair Machine Learning

Philip S. Thomas

College of Information and Computer Sciences, UMass Amherst

MSR Reinforcement Learning Day, October 3, 2019

Andy Barto

Yuriy Brun

Emma Brunskill

Stephen Giguere

Blossom Metevier

Bruno Castro da Silva

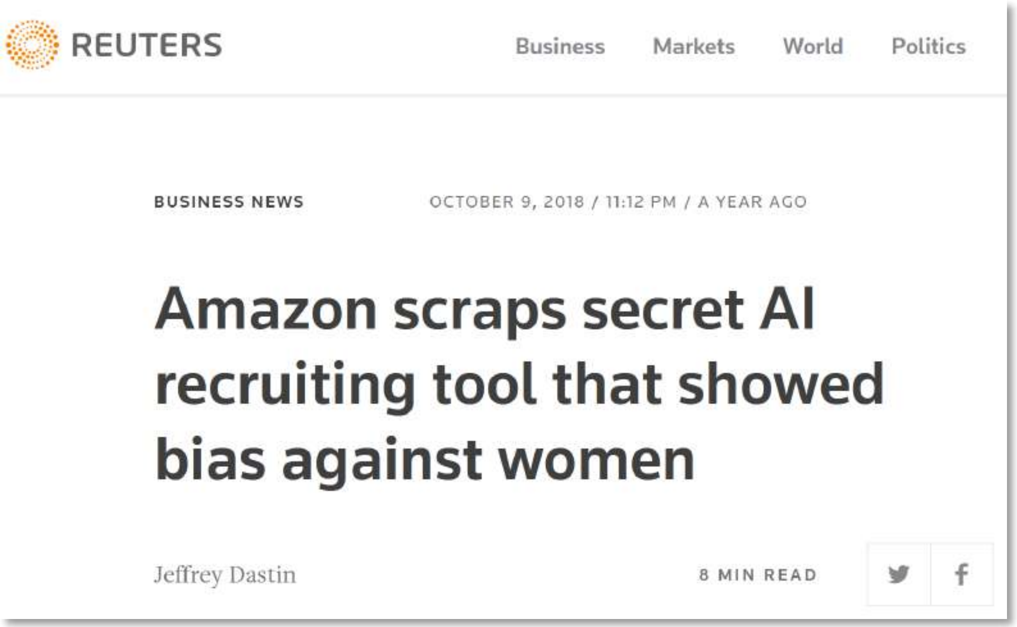
Ari Kobren

Georgios Theocharous

Mohammad Ghavamzadeh

Sarah Brockman

Machine learning algorithms should avoid *undesirable behaviors*.



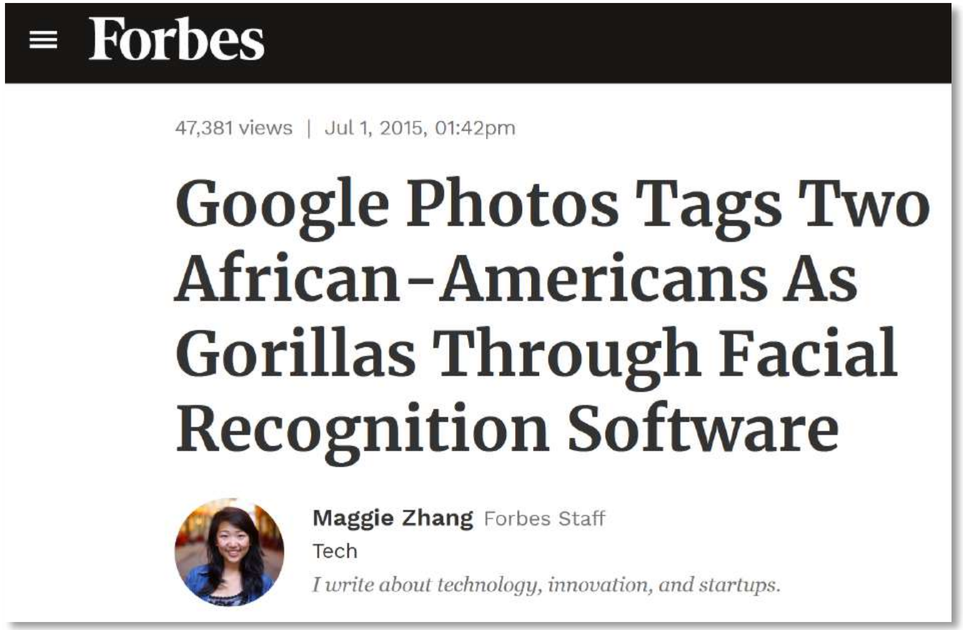
REUTERS

BUSINESS NEWS OCTOBER 9, 2018 / 11:12 PM / A YEAR AGO

Amazon scraps secret AI recruiting tool that showed bias against women

Jeffrey Dastin

8 MIN READ



Forbes

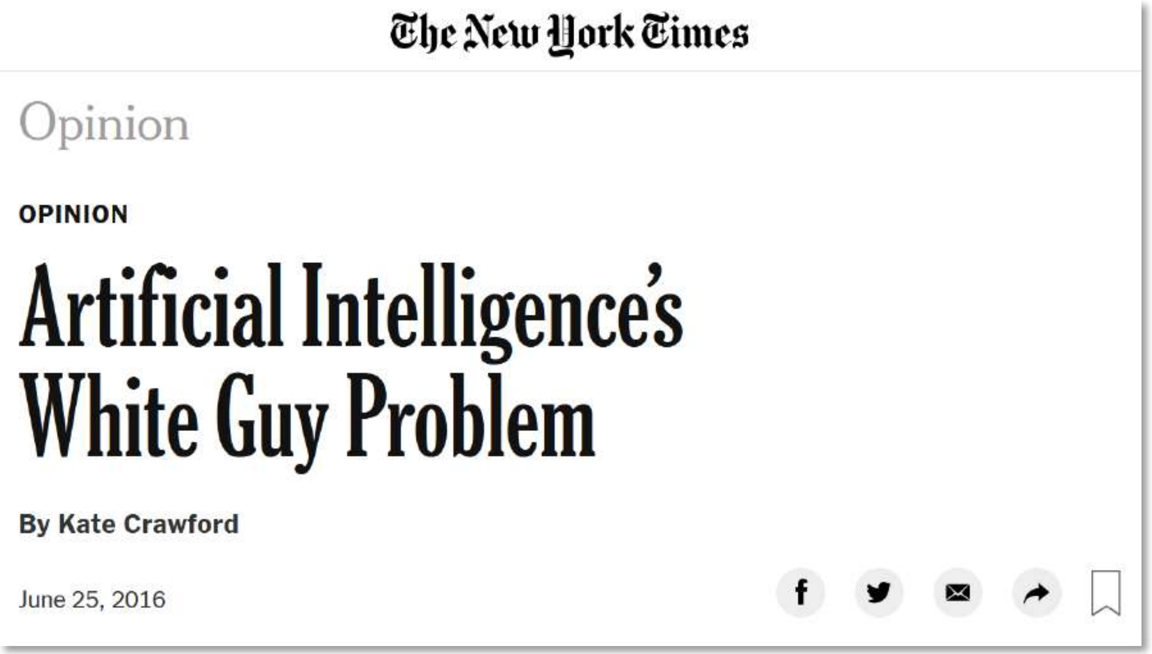
47,381 views | Jul 1, 2015, 01:42pm

Google Photos Tags Two African-Americans As Gorillas Through Facial Recognition Software

Maggie Zhang Forbes Staff

Tech

I write about technology, innovation, and startups.



The New York Times

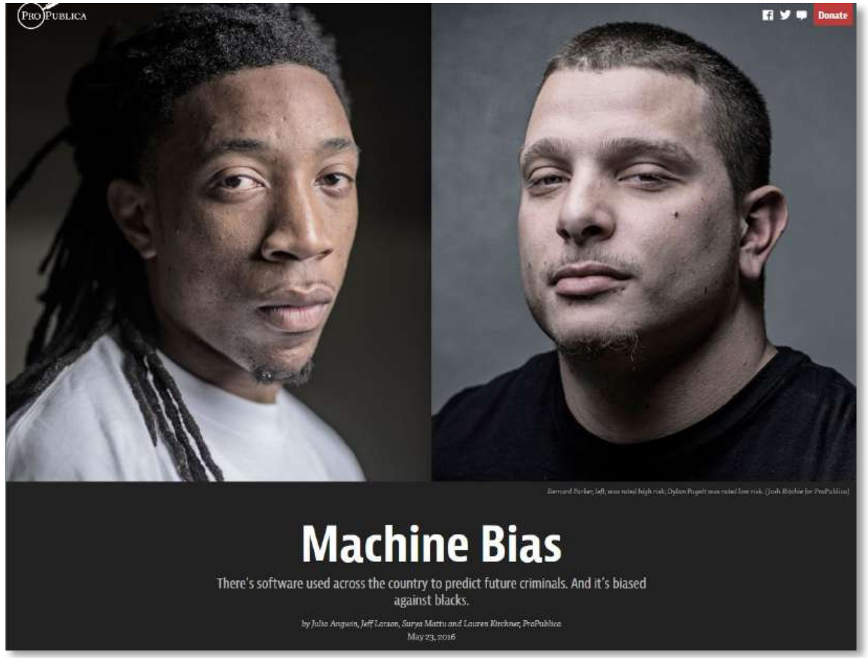
Opinion

OPINION

Artificial Intelligence's White Guy Problem

By Kate Crawford

June 25, 2016



Bernard Parker, left, was rated high risk; Dylan Fugett was rated low risk. (Josh Ritchie for ProPublica)

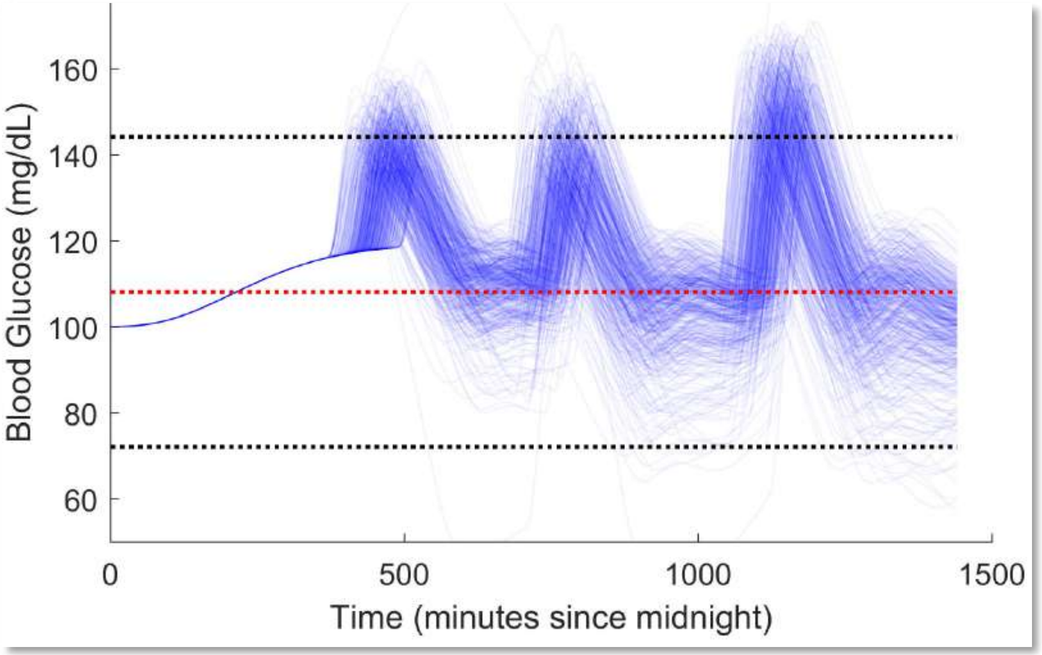
Machine Bias

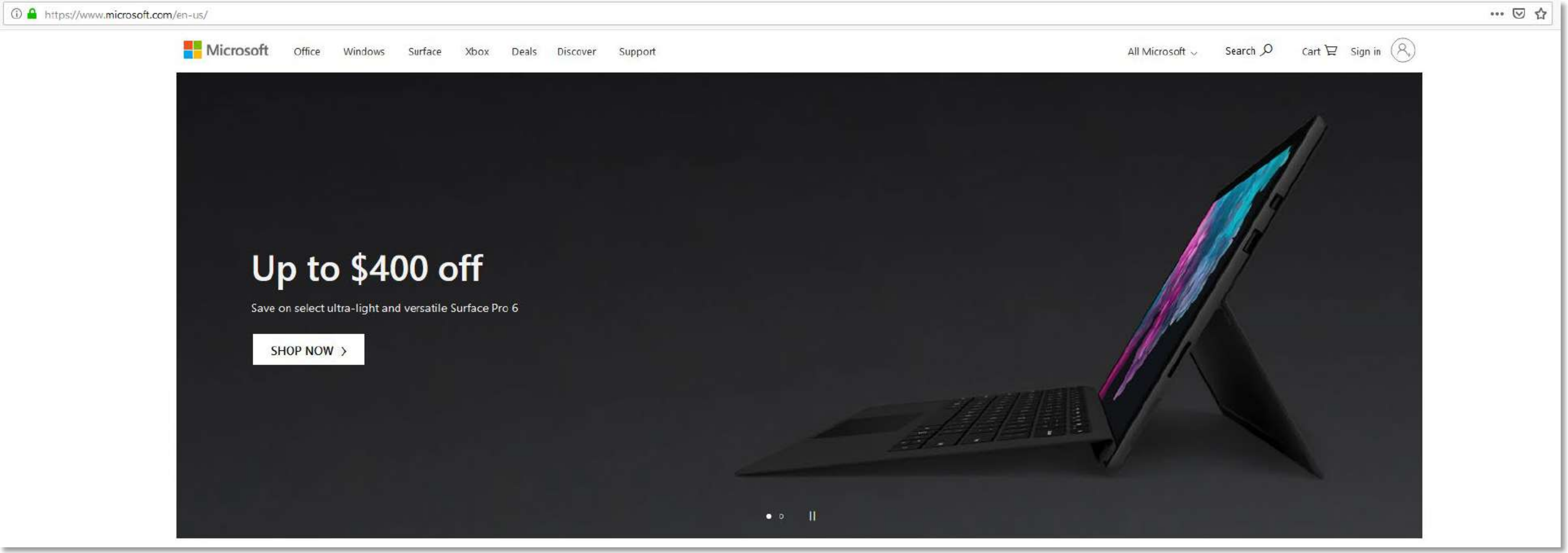
There's software used across the country to predict future criminals. And it's biased against blacks.

by Julia Angwin, Jeff Larson, Surya Mattu and Lauren Kirchner, ProPublica

May 23, 2016

ACCU-CHEK Spirit
0.7 U/h


Blood Glucose (mg/dL)
160
140
120
100
80
60
0
500
1000
1500
Time (minutes since midnight)

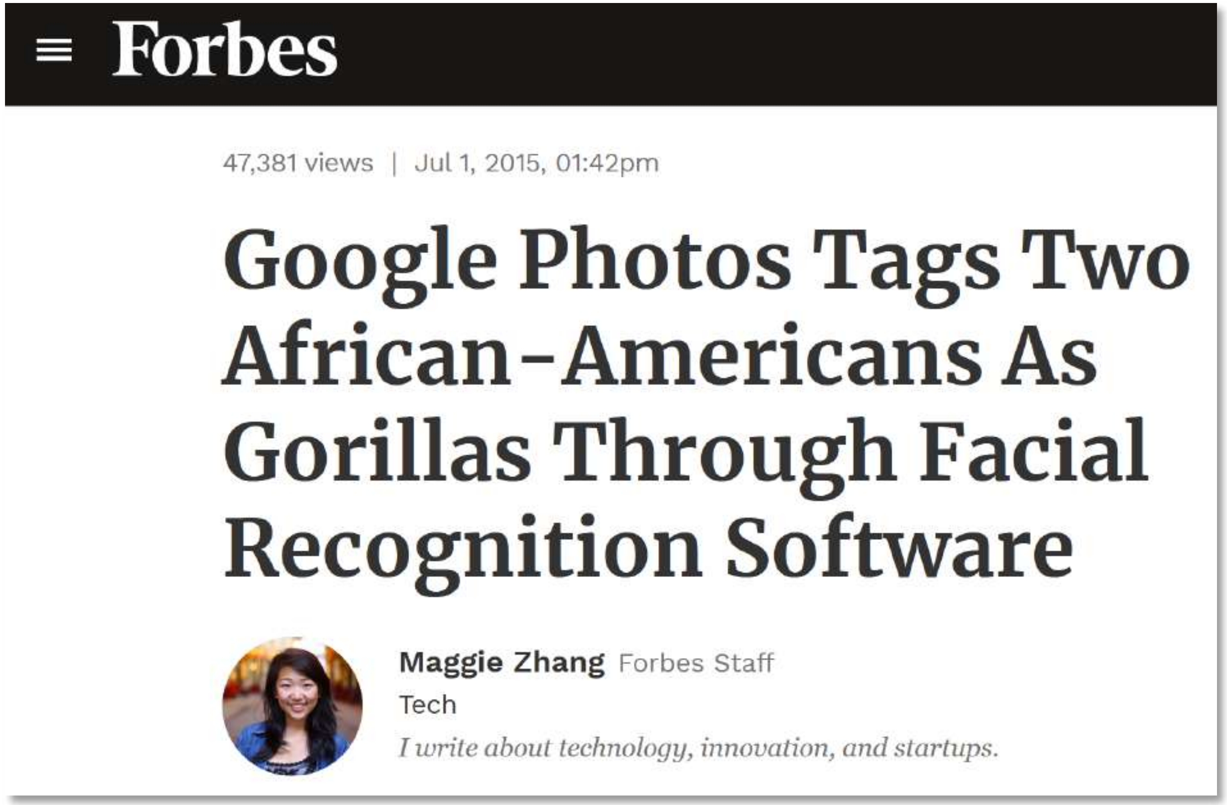


Up to $400 off

Save on select ultra-light and versatile Surface Pro 6

Undesirable behavior of ML algorithms is causing harm.

Supervised Learning
(Classification and Regression)



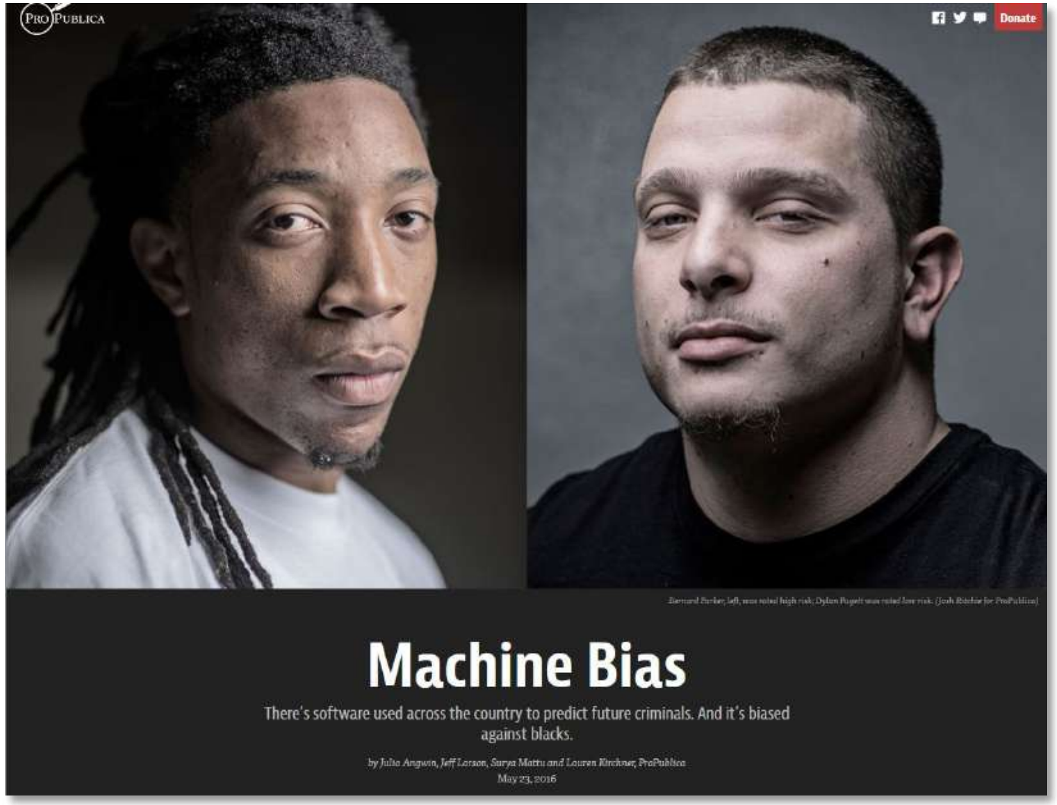
47,381 views | Jul 1, 2015, 01:42pm

Google Photos Tags Two African-Americans As Gorillas Through Facial Recognition Software

Maggie Zhang Forbes Staff

Tech

I write about technology, innovation, and startups.



Bernard Parker, left, was rated high risk; Dylan Fugett was rated low risk. (Josh Ritchie for ProPublica)

Machine Bias

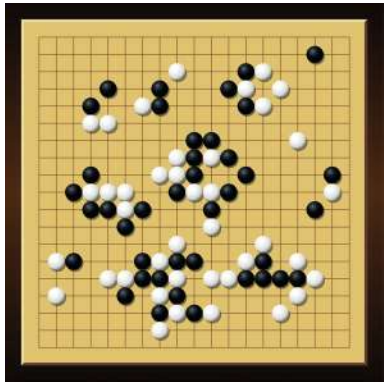
There's software used across the country to predict future criminals. And it's biased against blacks.

by Julia Angwin, Jeff Larson, Surya Mattu and Lauren Kirchner, ProPublica

May 23, 2016

Reinforcement Learning

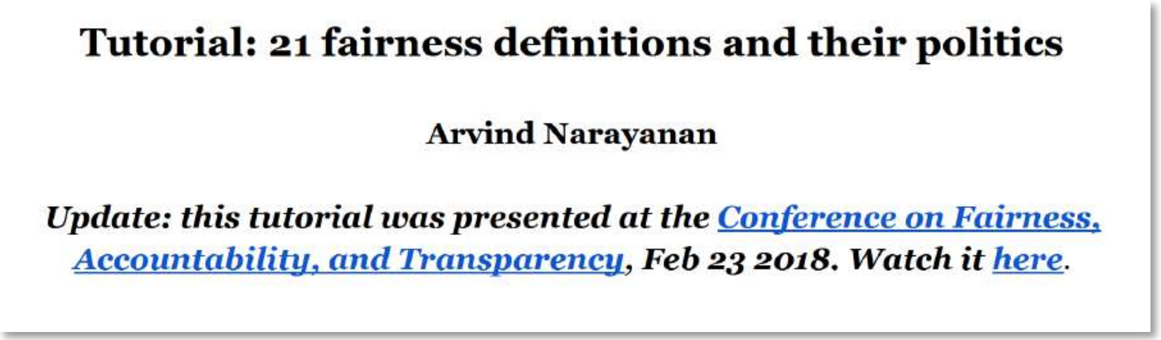
Up to $400 off



Can we create algorithms that allow their users to more easily control their behavior?

Desiderata

- Easy for users to define *undesirable behavior*.



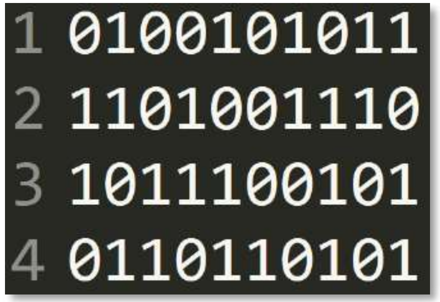
Tutorial: 21 fairness definitions and their politics

Arvind Narayanan

Update: this tutorial was presented at the Conference on Fairness, Accountability, and Transparency, Feb 23 2018. Watch it here.

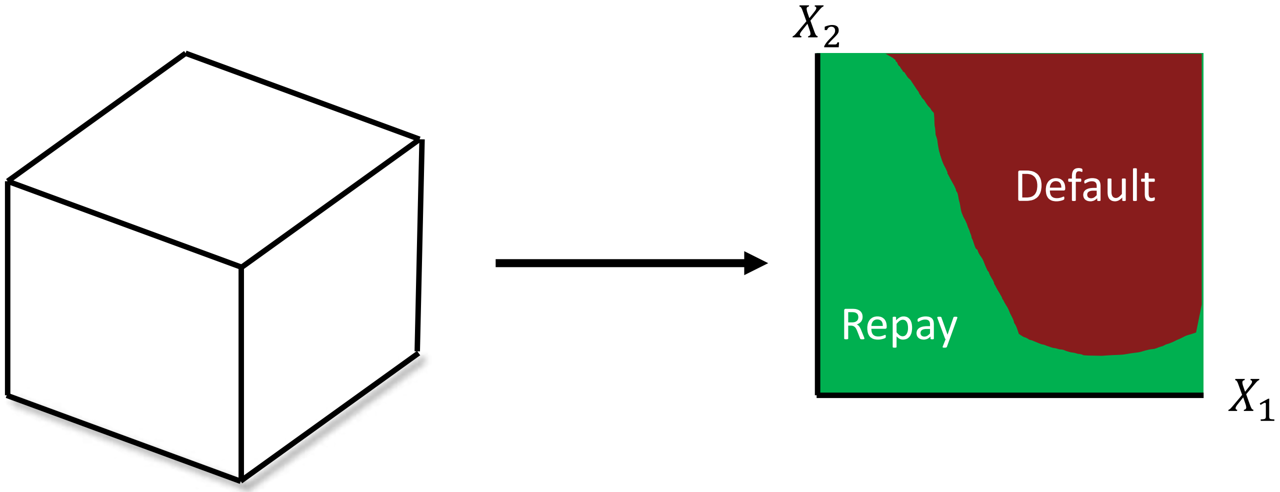
Mean Time Hypoglycemic
vs
Weighted Mean Time Hypoglycemic

- Guarantee that the algorithm will not produce this undesirable behavior.



```
0100101011
1101001110
1011100101
0110110101
```

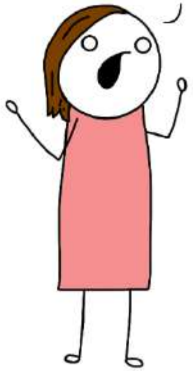
Data, D

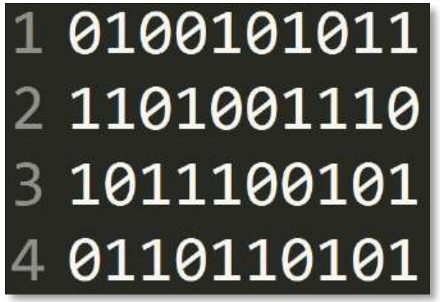


Algorithm, a

Solution, θ

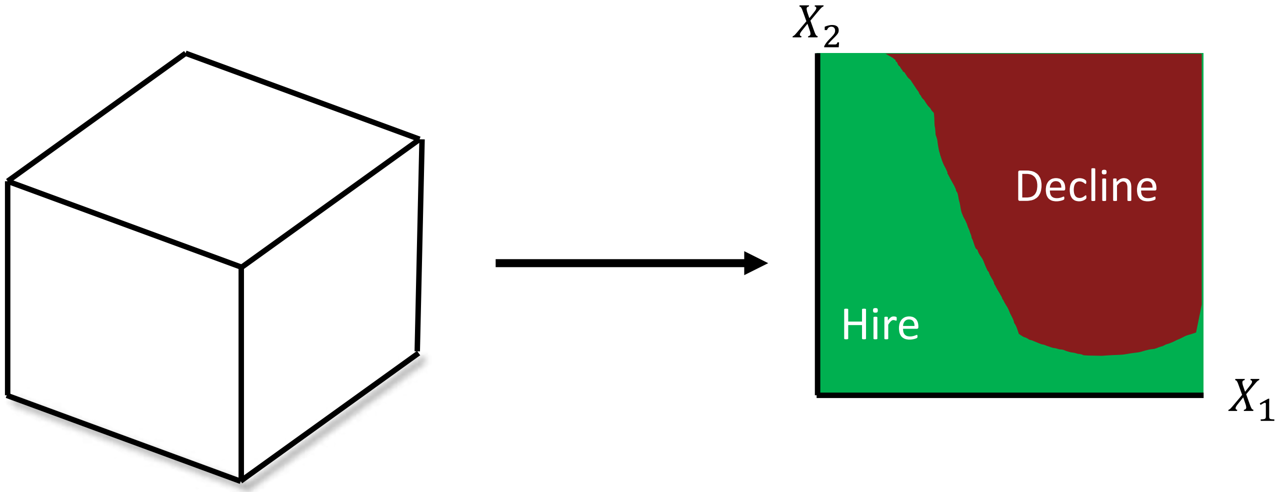
Learn to predict loan repayment, and don't discriminate.





```
0100101011
1101001110
1011100101
0110110101
```

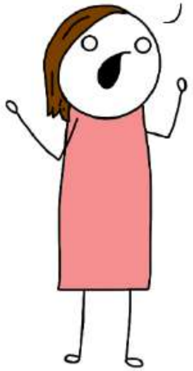
Data, D

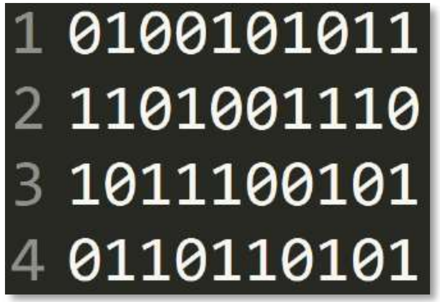


Algorithm, a

Solution, θ

Learn to predict job aptitude, and don't discriminate.

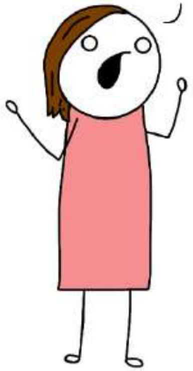


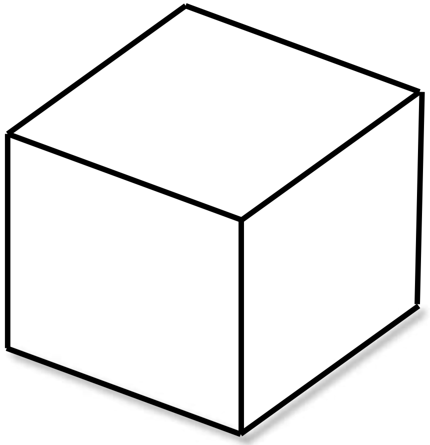


```
0100101011
1101001110
1011100101
0110110101
```

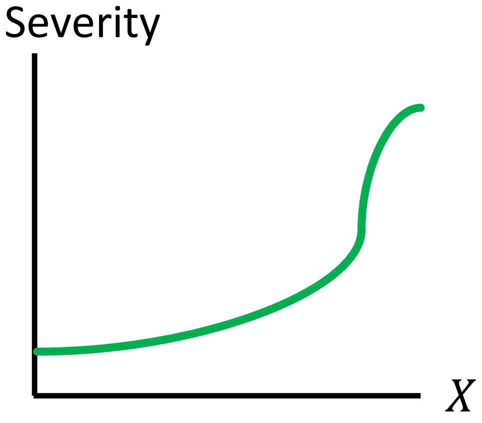
Data, D

Learn to predict landslide severity, but don't under-estimate.

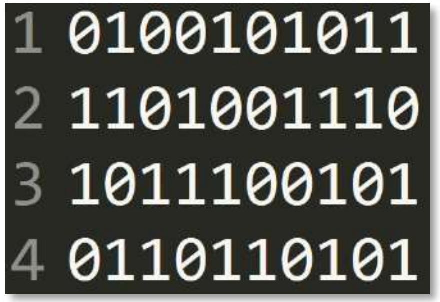




Algorithm, a



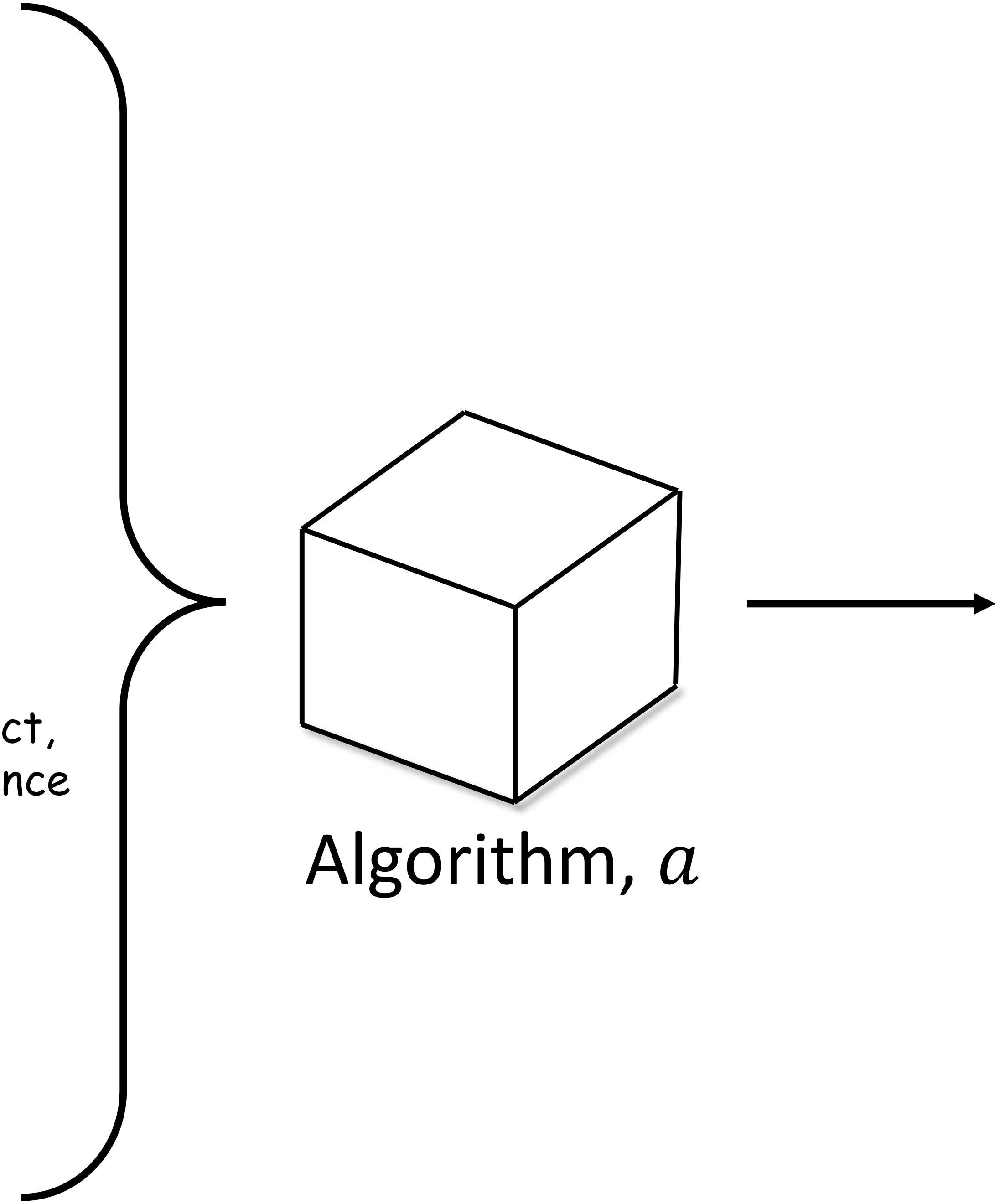
Solution, θ



```
0100101011
1101001110
1011100101
0110110101
```

Data, D

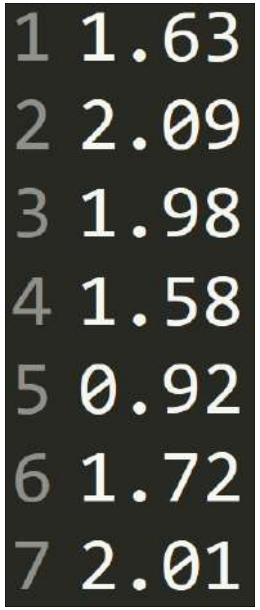
Learn how much insulin to inject, but don't increase the prevalence of hypoglycemia.



Algorithm, a

$$\boxed{\text{injection} = \frac{\text{current} \; - \; \text{target}}{\theta_1} + \frac{\text{meal size}}{\theta_2}}$$

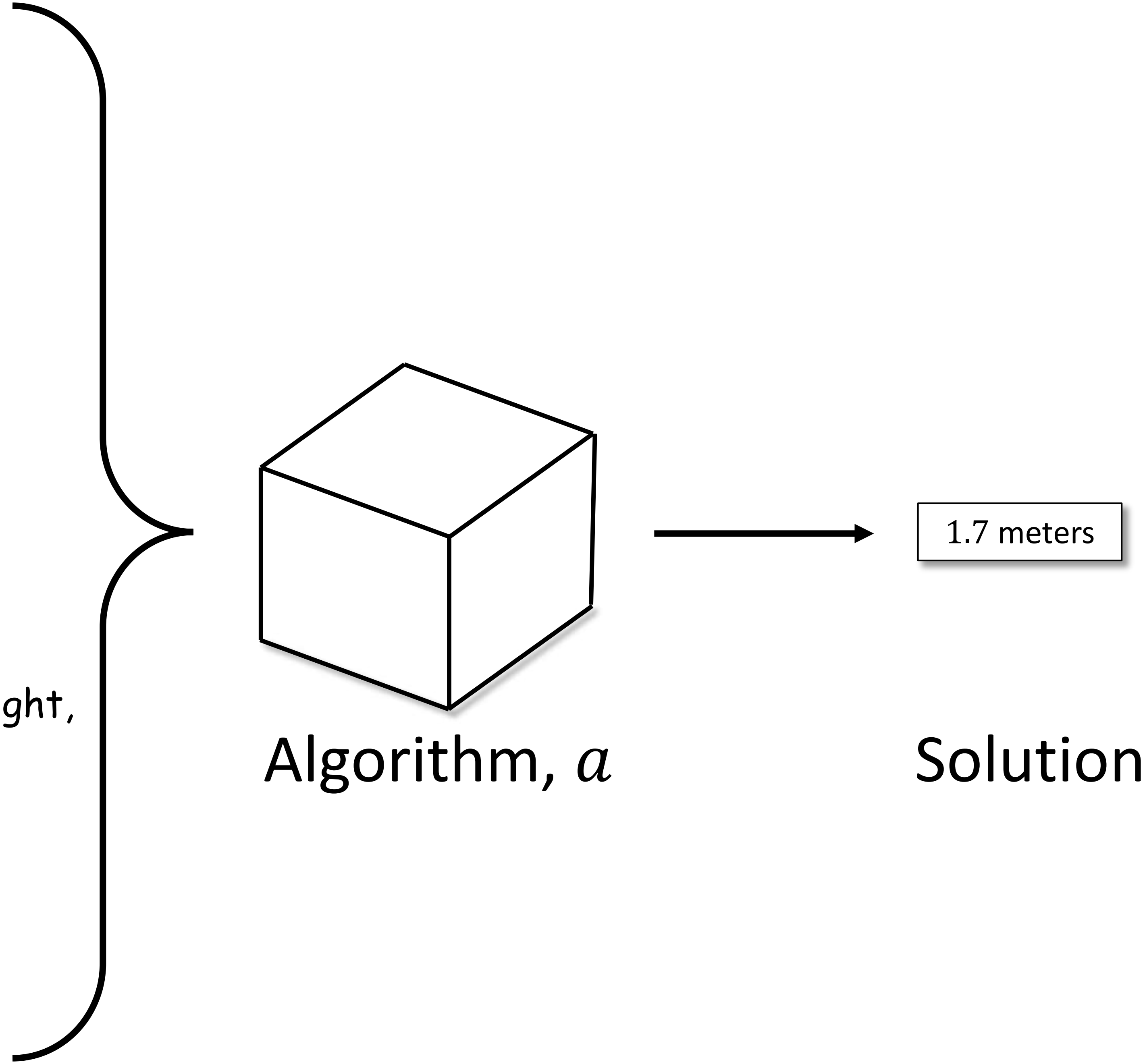
Solution, θ



```
1.63
2.09
1.98
1.58
0.92
1.72
2.01
```

Data, D

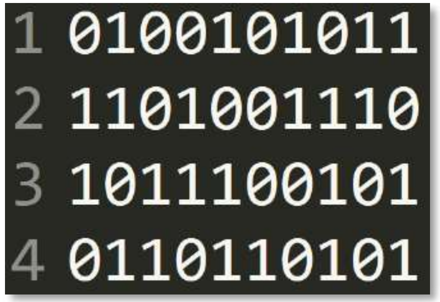
Predict the average human height,
but do not over-estimate.



1.7 meters

Algorithm, a

Solution, θ

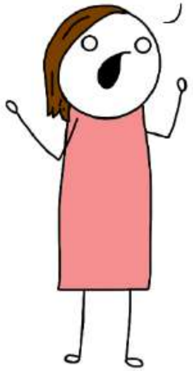


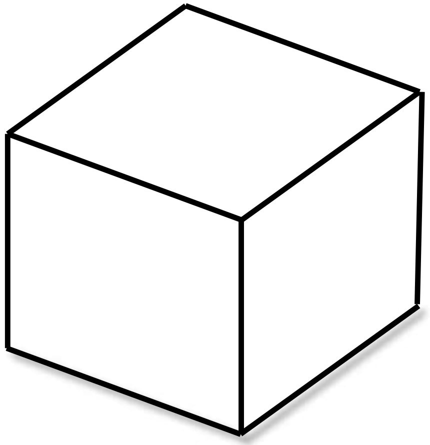
```
1 0100101011
2 1101001110
3 1011100101
4 0110110101
```

Data, D

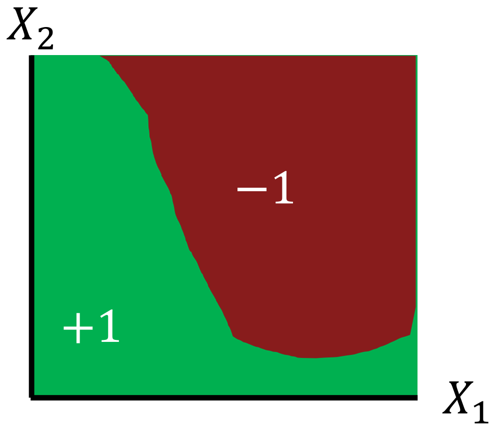
Probability, δ

Achieve [main goal] but do not produce [undesirable behavior].

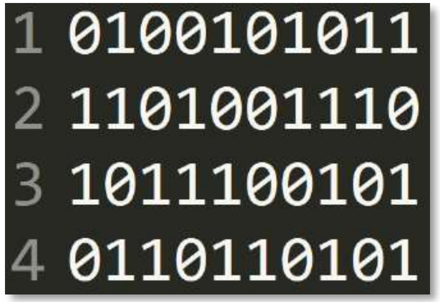




Algorithm, a



Solution, θ

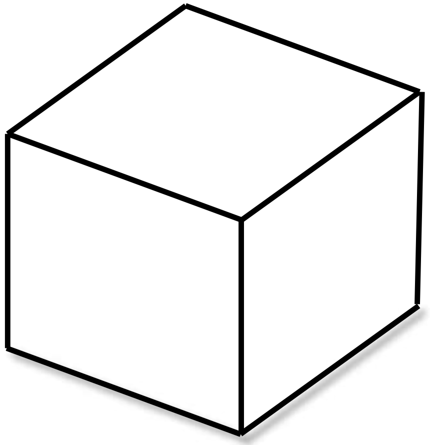


```
0100101011
1101001110
1011100101
0110110101
```

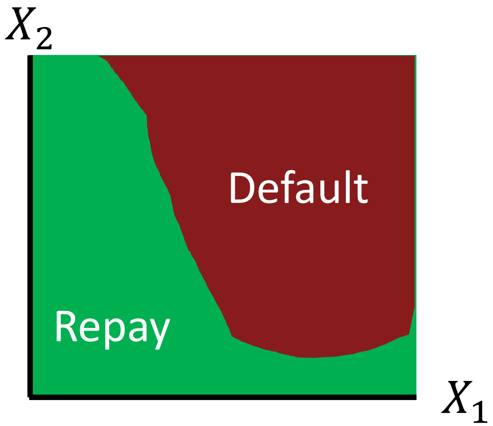
Data, D

(From four people)

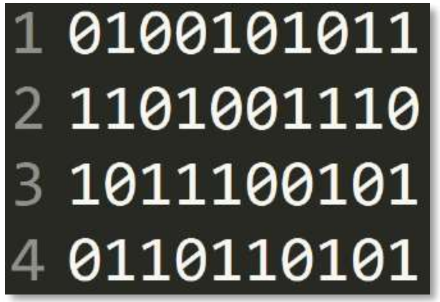
Probability, $\delta = 0.01$



Algorithm, a



Solution, θ

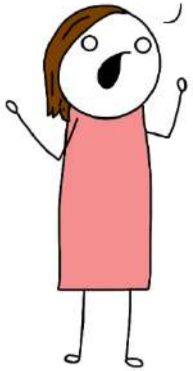


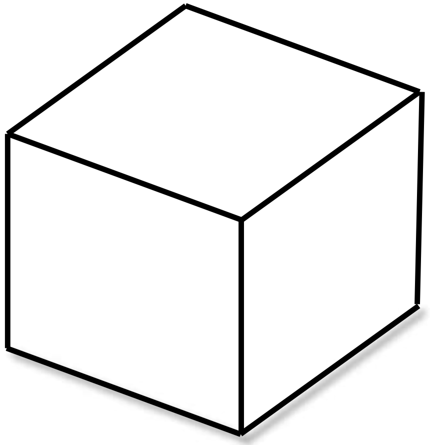
```
0100101011
1101001110
1011100101
0110110101
```

Data, D

Probability, $\delta = 0.05$

Learn how much insulin to inject, but don't ever allow blood glucose to deviate from optimal by more than $1.2\,\frac{\mathrm{mg}}{\mathrm{dL}}$.

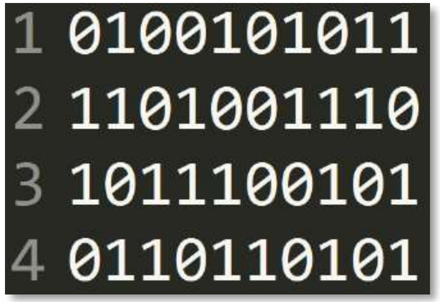




Algorithm, a

$$\text{injection} = \frac{\text{current} \; - \; \text{target}}{\theta_1} + \frac{\text{meal size}}{\theta_2}$$

Solution, θ

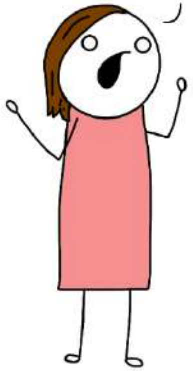


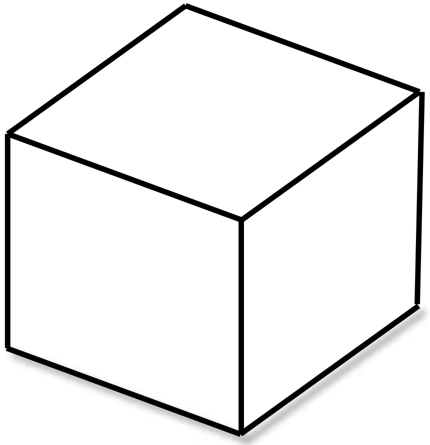
```
0100101011
1101001110
1011100101
0110110101
```

Data, D

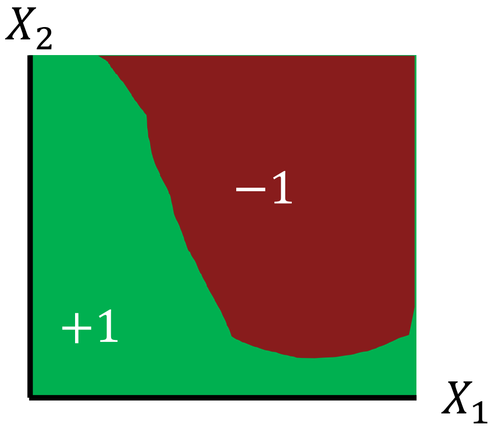
Probability, δ

Achieve [main goal] but do not produce [undesirable behavior].





Algorithm, a



Solution, θ

OR

No Solution Found

Desiderata

- Interface for defining undesirable behavior.
- User-specified probability, δ.
- Guarantee that the probability of a solution that produces undesirable behavior is at most δ.

Notation

- Let D be *all* of the training data.
  - D is a random variable
- Let Θ be the set of all possible solutions the algorithm can return.
- Let $f\colon \Theta \to \mathbb{R}$ be the primary objective function.
- Let a be a machine learning algorithm.
  - $a\colon \mathcal{D} \to \Theta$
  - $a(D)$ is the solution returned by the algorithm when run on data D
- Let $g\colon \Theta \to \mathbb{R}$ be a function that measures undesirable behavior
  - $g(\theta) \leq 0$ if and only if θ does not produce undesirable behavior
  - $g(\theta) > 0$ if and only if θ produces undesirable behavior
- Let $\mathrm{NSF} \in \Theta$ and $g(\mathrm{NSF}) = 0$.

Desiderata

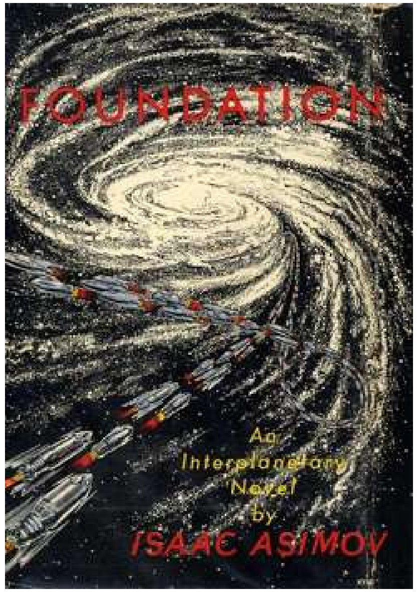
- Provide the user with an interface for defining *undesirable behavior* (i.e., defining g).
- Attempt to optimize a (possibly user-provided) objective f.
- Guarantee that

$$\Pr\big(g\big(a(D)\big) \leq 0\big) \geq 1 - \delta.$$

  - The probability that the algorithm returns a solution that does not produce undesirable behavior is at least $1 - \delta$.
  - The probability that the algorithm returns a solution that produces undesirable behavior is at most δ.
  - We need a name for algorithms that provide this guarantee.

$$\Pr\big(g\big(a(D)\big) \le 0\big) \ge 1 - \delta$$

- An algorithm that provides this guarantee is *safe*.
- An algorithm that provides this guarantee is *Seldonian*.
- Quasi-Seldonian: Reasonable false assumptions
  - Appeals to central limit theorem

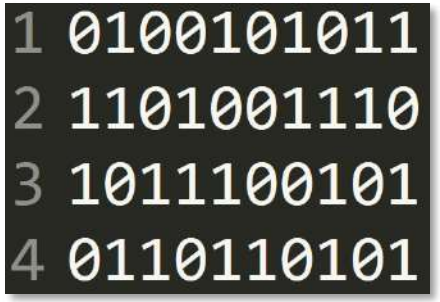


Seldonian Framework

- Framework for designing machine learning algorithms.
  - Provide the user with an interface for defining *undesirable behavior* (i.e., defining g).
  - Attempt to optimize a (possibly user-provided) objective f.
  - Guarantee that
$$\Pr\Big(g\big(a(D)\big) \leq 0\Big) \geq 1 - \delta.$$
    - This guarantee does *not* depend on any hyperparameter settings.
- I am *not* promoting a specific algorithm.
  - The algorithms I am going to discuss are extremely simple examples.
  - These examples show *feasibility.*
- I am promoting the *framework*.

Example Usage

- X is a vector of features describing a person convicted of a crime.
- Y is 1 if the person committed a subsequent violent crime and 0 otherwise.
- Find a solution, θ, such that $\hat{y}(X, \theta)$ is a good estimator of Y.
- $g(\theta) = |\Pr(\hat{y}(X, \theta) = 1|\text{White}) - \Pr(\hat{y}(X, \theta) = 1|\text{Not White})| - \epsilon$
  - "Demographic Parity"
  - $g(\theta) \leq 0$ iff $\Pr(\hat{y}(X, \theta) = 1|\text{White}) \approx \Pr(\hat{y}(X, \theta) = 1|\text{Not White})$
- $g(\theta) = |\Pr(\text{FP}|\text{White}) - \Pr(\text{FP}|\text{Not White})| - \epsilon$
  - "Predictive Equality"
  - $g(\theta) \leq 0$ iff $\Pr(\text{FP}|\text{White}) \approx \Pr(\text{FP}|\text{Not White})$



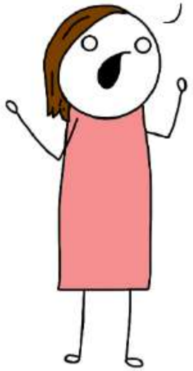
```
0100101011
1101001110
1011100101
0110110101
```

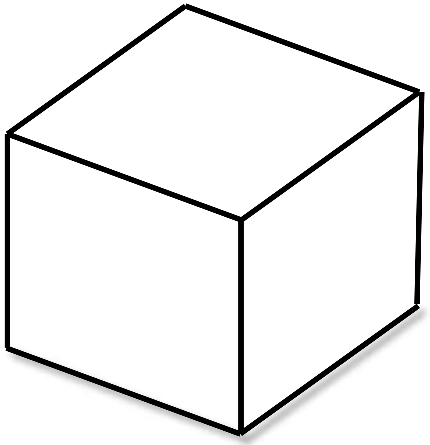
Data, D

Probability, δ

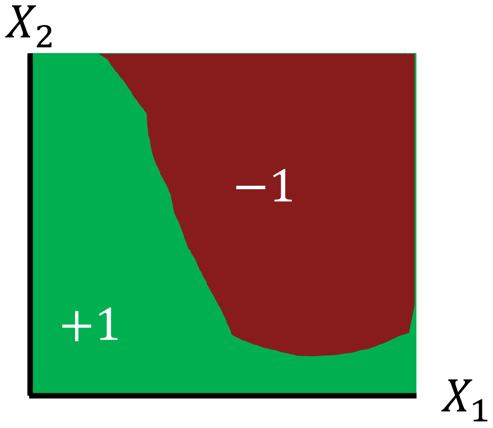
Minimize classification loss, use

$g(\theta) = |\Pr(\text{FP}|\text{White}) - \Pr(\text{FP}|\text{Not White})| - \epsilon$





Algorithm, a



Solution, θ

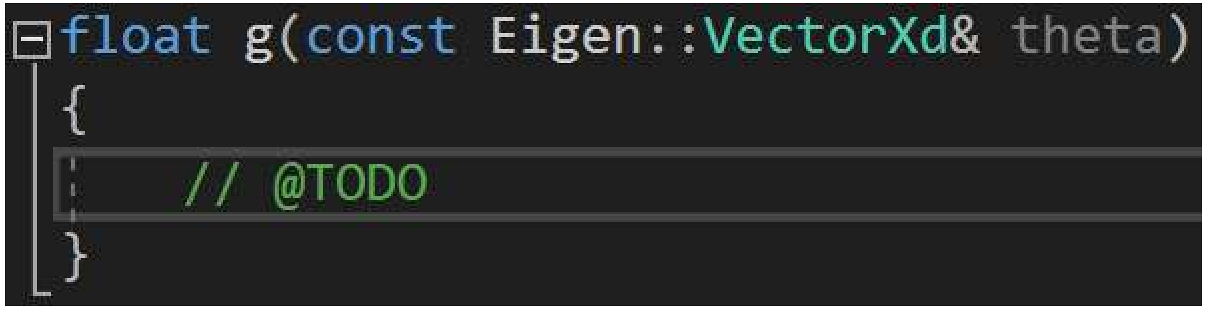
OR

No Solution Found

Minimize classification loss, use

$$g(\theta) = |\Pr(\text{FP}|\text{White}) - \Pr(\text{FP}|\text{Not White})| - \epsilon$$

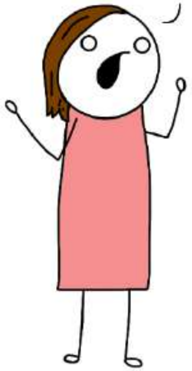
- Provide code for g:



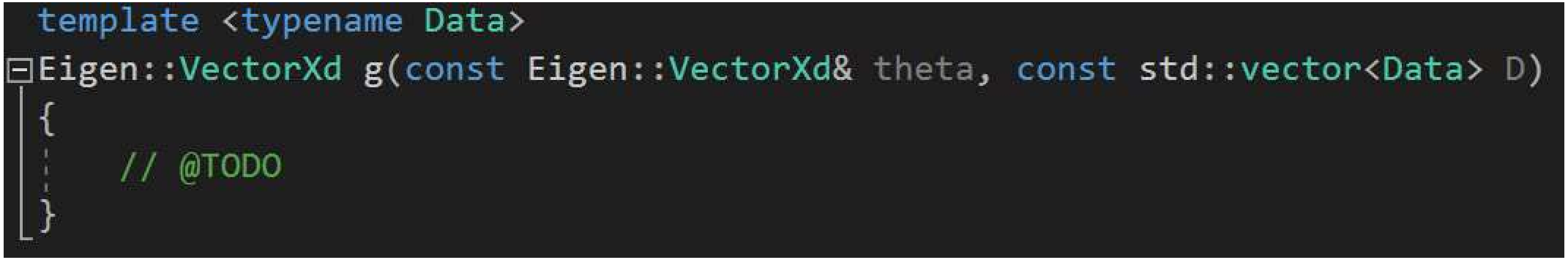
```
float g(const Eigen::VectorXd& theta)
{
    // @TODO
}
```

Minimize classification loss, use

$$g(\theta) = |\Pr(\text{FP}|\text{White}) - \Pr(\text{FP}|\text{Not White})| - \epsilon$$



- Provide code for unbiased estimates of g:



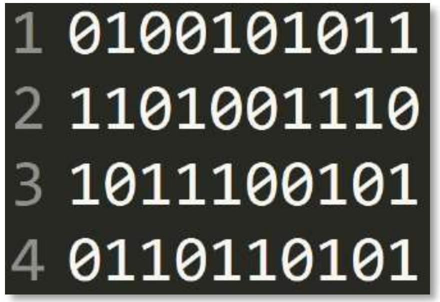
```
template <typename Data>
Eigen::VectorXd g(const Eigen::VectorXd& theta, const std::vector<Data> D)
{
    // @TODO
}
```

Minimize classification loss, use
$g(\theta) = |\Pr(\text{FP}|\text{White}) - \Pr(\text{FP}|\text{Not White})| - \epsilon$

- Write an equation for g:

$$g(\theta) = |\Pr(\text{FP}|\text{White}) - \Pr(\text{FP}|\text{Not White})| - \epsilon$$

  - Can use any of the common variables already provided.
    - Classification: FP, FN, TP, TN, conditional FP, accuracy, probability positive, etc.
    - Regression: MSE, ME, conditional MSE, conditional ME, mean prediction, etc.
    - Reinforcement learning: Expected return, conditional expected return
  - Can use any other variable for which they can provide unbiased estimates from data.
  - Can use any supported operators $+, -, \%, \text{abs}, \min, \max$, etc.



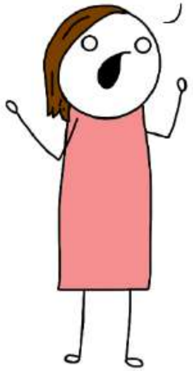
```
0100101011
1101001110
1011100101
0110110101
```

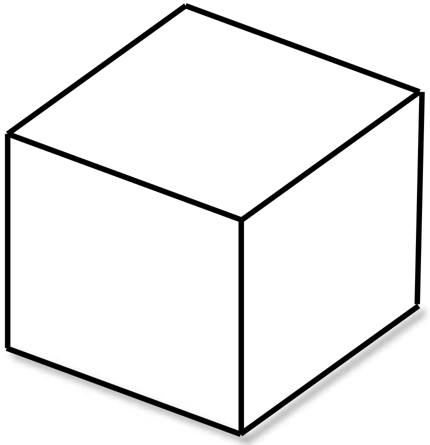
Data, D

Probability, δ

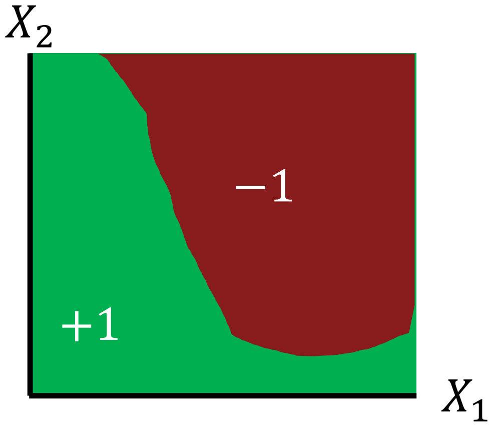
Minimize classification loss, use

$g(\theta) = |\Pr(\text{FP}|\text{White}) - \Pr(\text{FP}|\text{Not White})| - \epsilon$





Algorithm, a

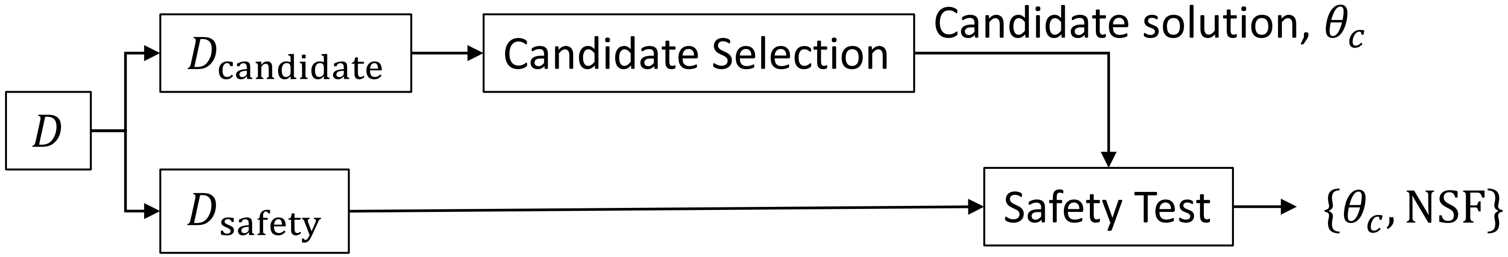


Solution, θ

OR

No Solution Found

Algorithm 11

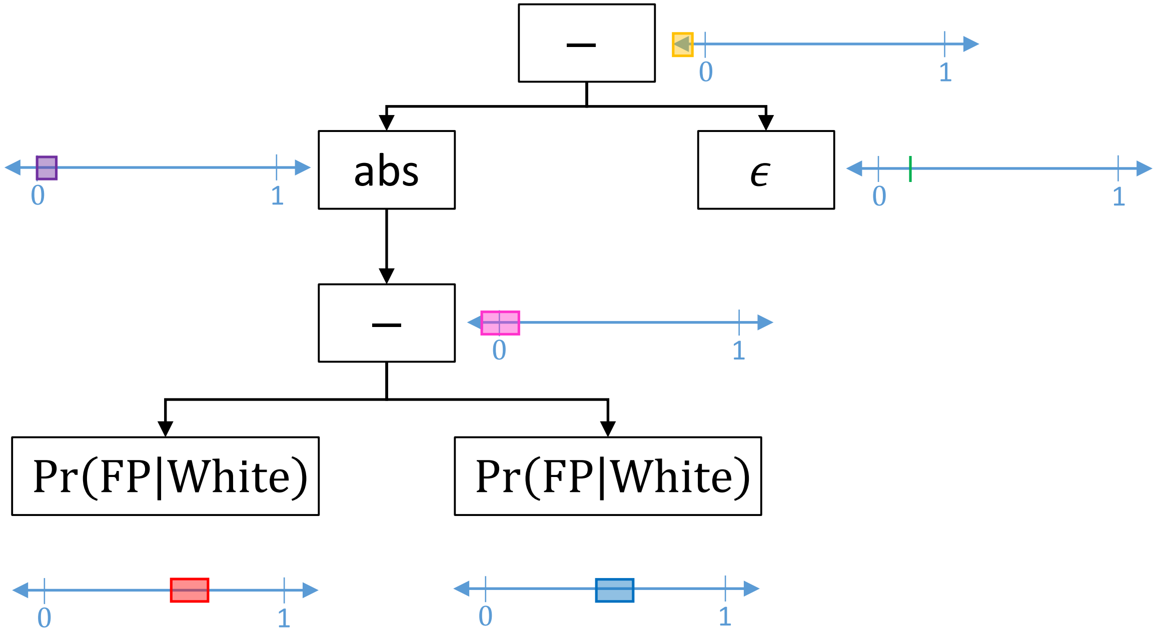


Safety Test

- Consider the earlier example:

$$g(\theta) = |\Pr(\mathrm{FP}|\mathrm{White}) - \Pr(\mathrm{FP}|\mathrm{Not\ White})| - \epsilon$$

- Given θ_c and D_{safety}, output either θ_c or NSF



Candidate Selection

- Use $D_{\text{candidate}}$ to pick the solution, θ_c, predicted to:
  - Optimize the primary objective, f
  - Pass the subsequent safety test

Reinforcement Learning

- Historical data, D, is data collected from running some current policy, π_{cur}.
- A solution, θ, is a policy or policy parameters.
- User can define multiple objectives (reward functions), and can require improvement (or limit degradation) with respect to all.
- $g(\theta) = \mathbf{E}[\sum_{t=0}^{\infty} \gamma^t R_t \,|\pi_{\text{cur}}] - \mathbf{E}[\sum_{t=0}^{\infty} \gamma^t R_t \,|\theta]$

$$\Pr(\mathbf{E}[\sum_{t=0}^{\infty} \gamma^t R_t \,|\pi_{\text{cur}}] \leq \mathbf{E}[\sum_{t=0}^{\infty} \gamma^t R_t \,|a(D)]) \geq 1 - \delta$$

- Monte Carlo returns are unbiased estimates of $\mathbf{E}[\sum_{t=0}^{\infty} \gamma^t R_t \,|\pi_{\text{cur}}]$.
- Use importance sampling to obtain unbiased estimates of $\mathbf{E}[\sum_{t=0}^{\infty} \gamma^t R_t \,|\theta_c]$.

Reinforcement Learning

- The ability to require improvement w.r.t. multiple objectives makes objective specification easier.
- Try to change the current policy to one that reaches the goal quicker in expectation.
- Do not increase the probability that the agent steps in the water.

Start	🟦	Goal
	🟦	

A Powerful Interface for Reinforcement Learning

- Have user label trajectories with a value $L \in \{1,0\}$:
  - Undesirable event: 1
  - No undesirable event: 0
- $\mathbf{E}[L]$ is the probability that the undesirable event will occur.
- Let $g(\theta) = \mathbf{E}[L|\theta] - \mathbf{E}[L|\pi_{\text{cur}}]$

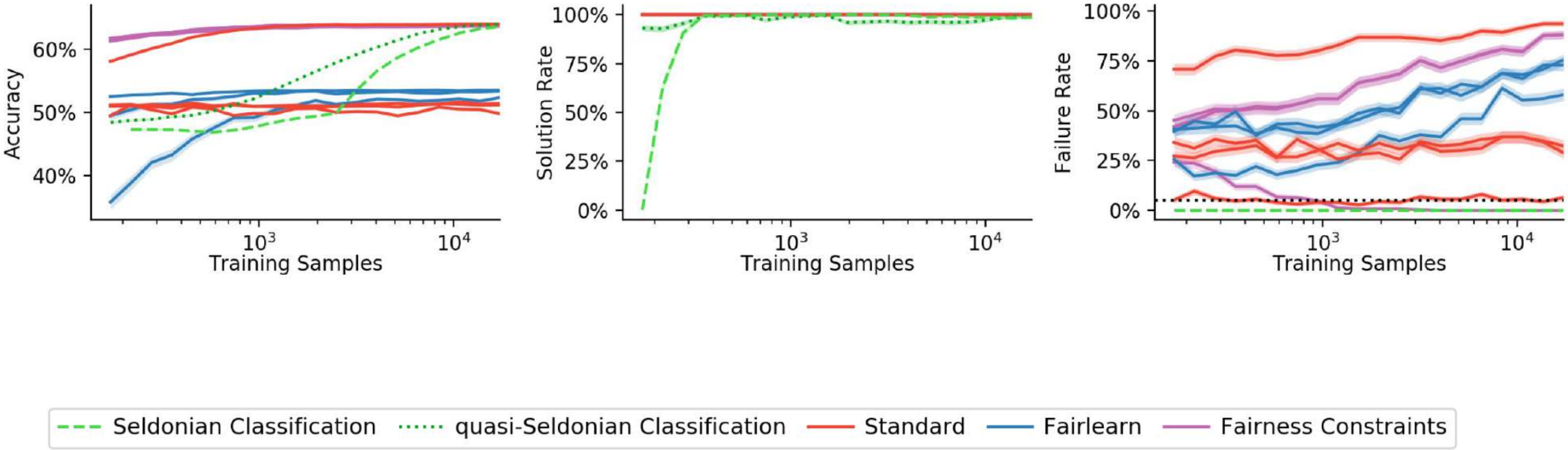
$$\Pr(\mathbf{E}[L|a(D)] \leq \mathbf{E}[L|\pi_{\text{cur}}]\,) \geq 1 - \delta$$

  - The probability that the policy will be changed to one that increases the probability of an undesirable event is at most δ.
- The user need only be able to *identify* undesirable events!

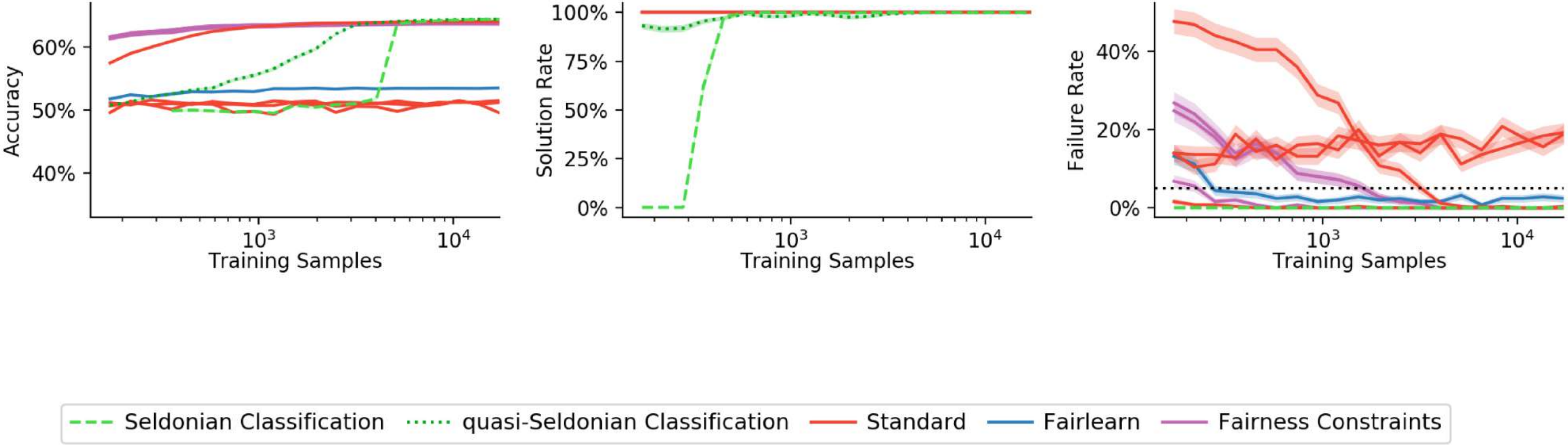
Example: Type 1 Diabetes Management

- Undesirable event: the person experienced hypoglycemia during the day.
- Try to keep blood glucose close to ideal levels (primary reward function), but guarantee with probability 0.95 that the probability of a hypoglycemic event will not increase.

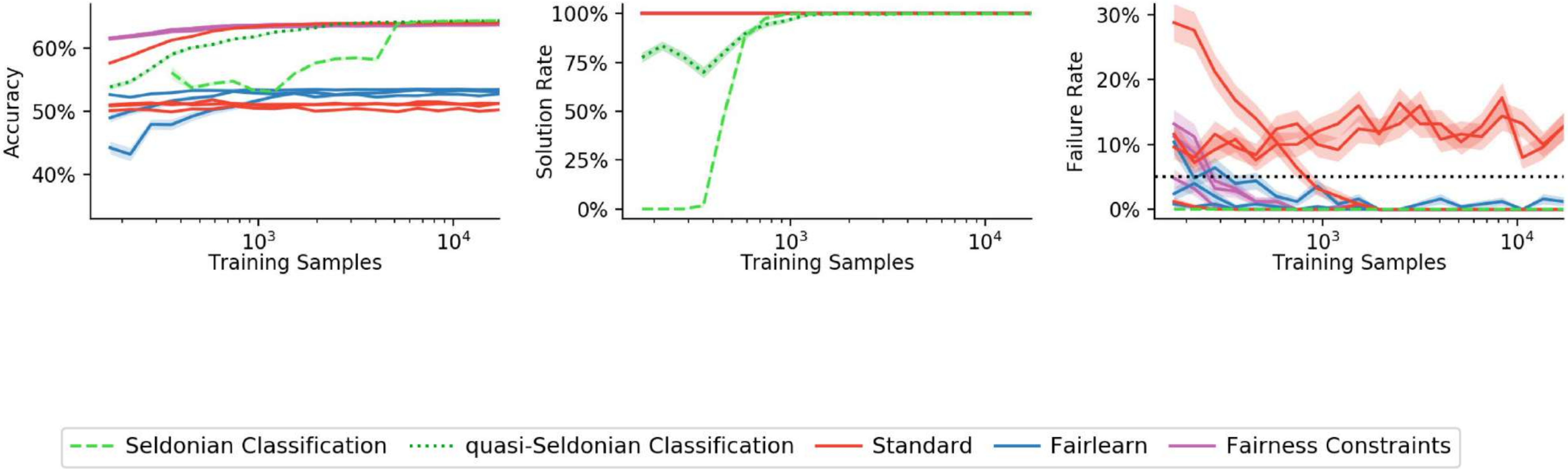
Example: Classification (GPA, Disparate Impact)



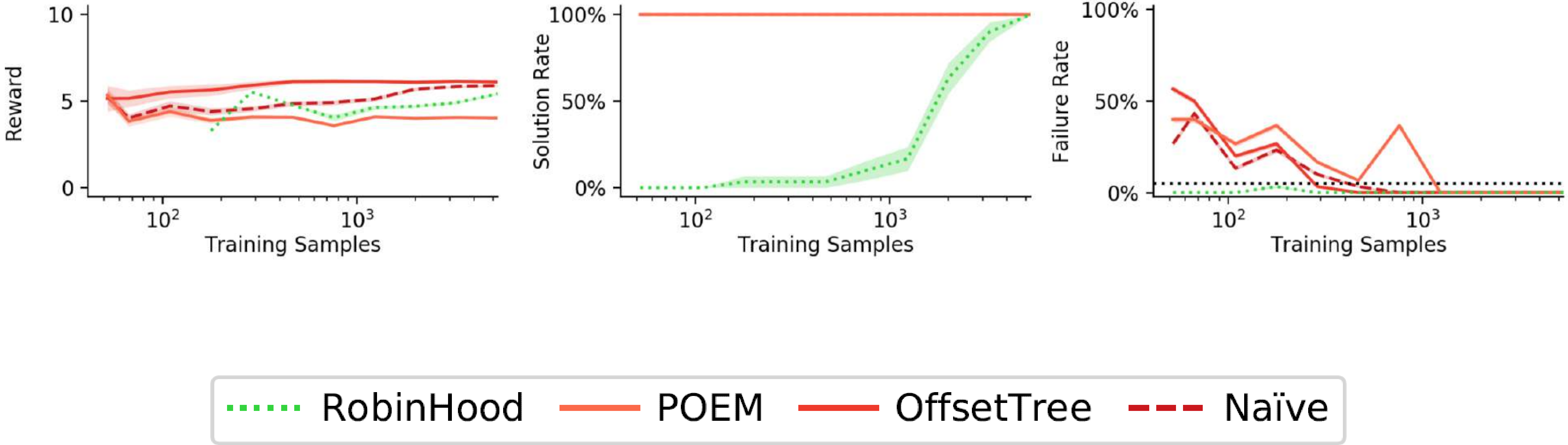
Example: Classification (GPA, Demographic Parity)



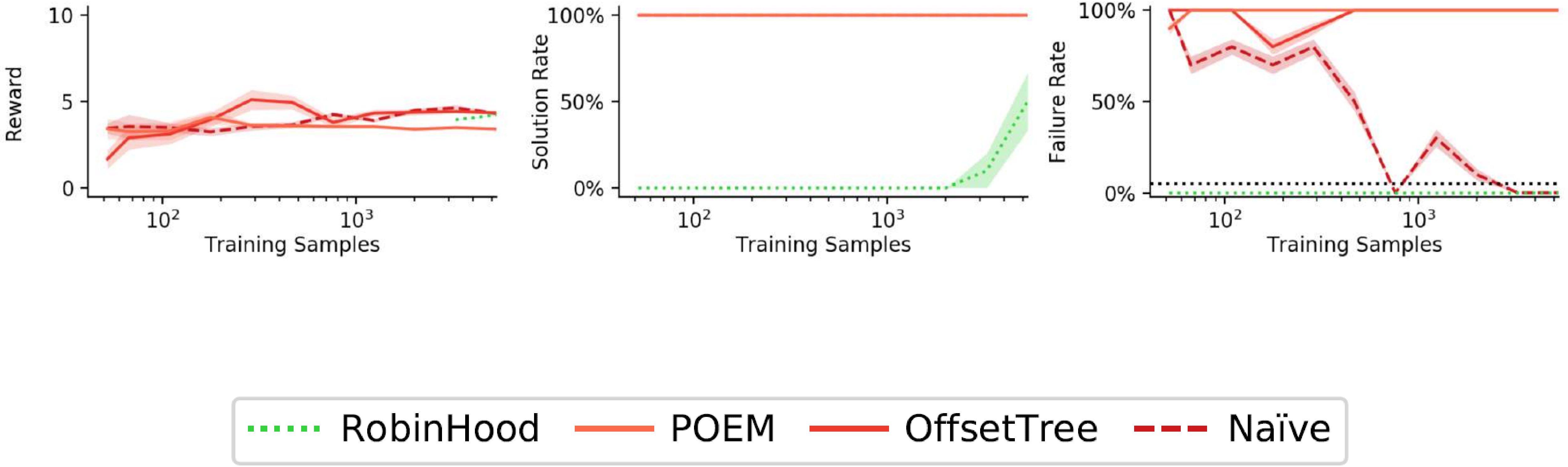
Example: Classification (GPA, Equalized Odds)



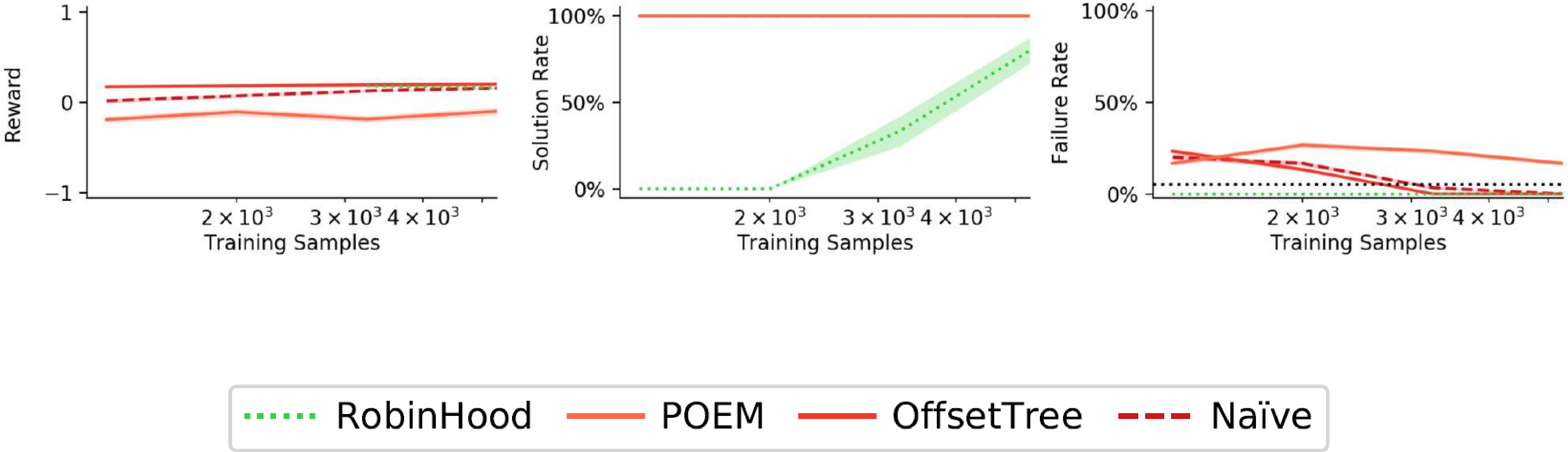
Example: Bandit (Tutoring)



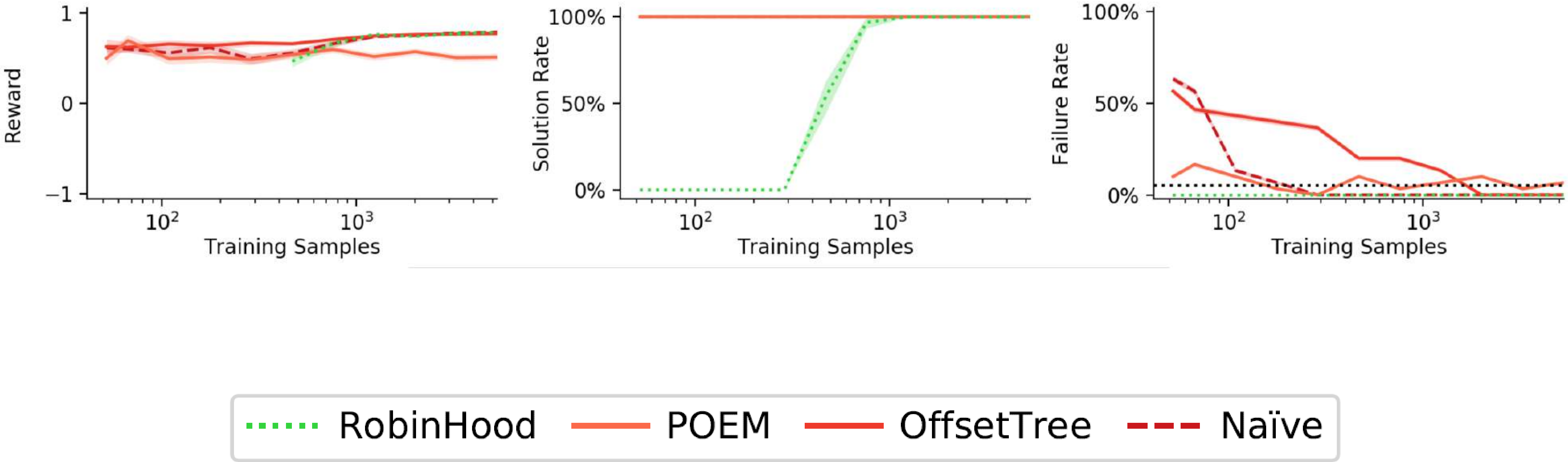
Example: Bandit (Tutoring, Skewed Proportions)



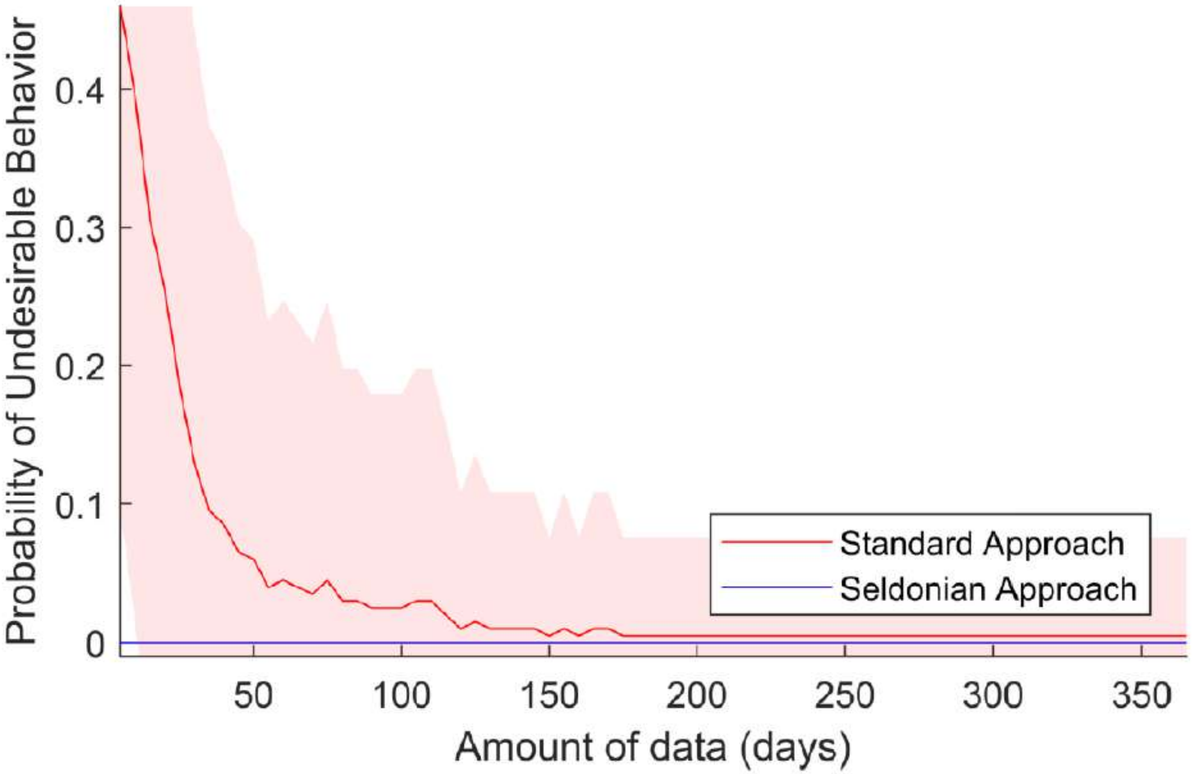
Example: Bandit (Loan Approval, Disparate Impact)

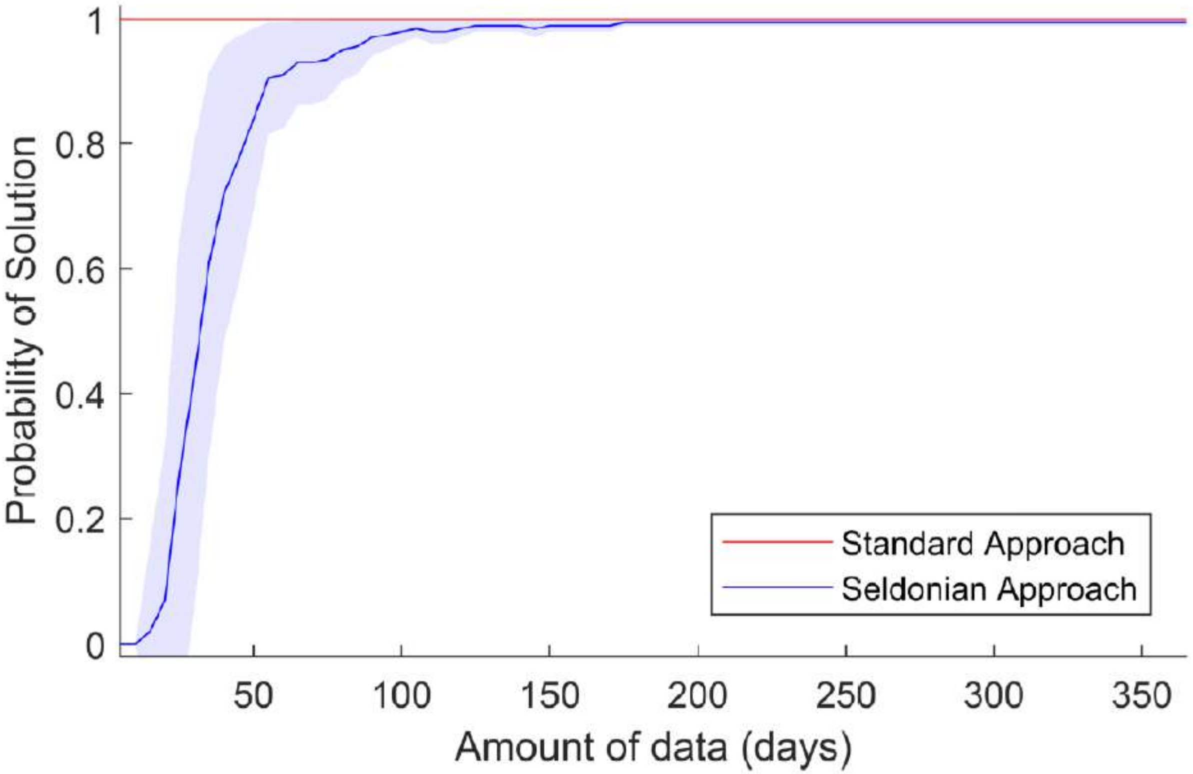


Example: Bandit (Recidivism, Statistical Parity)

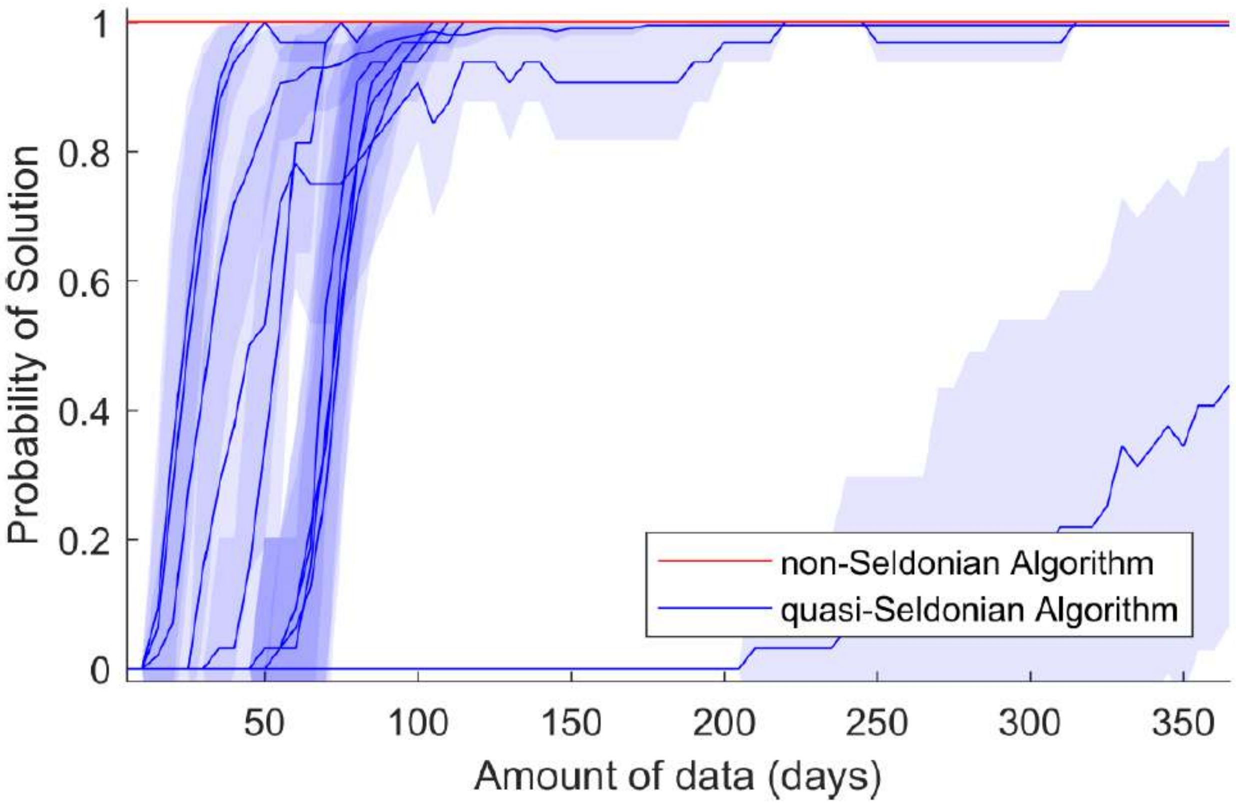


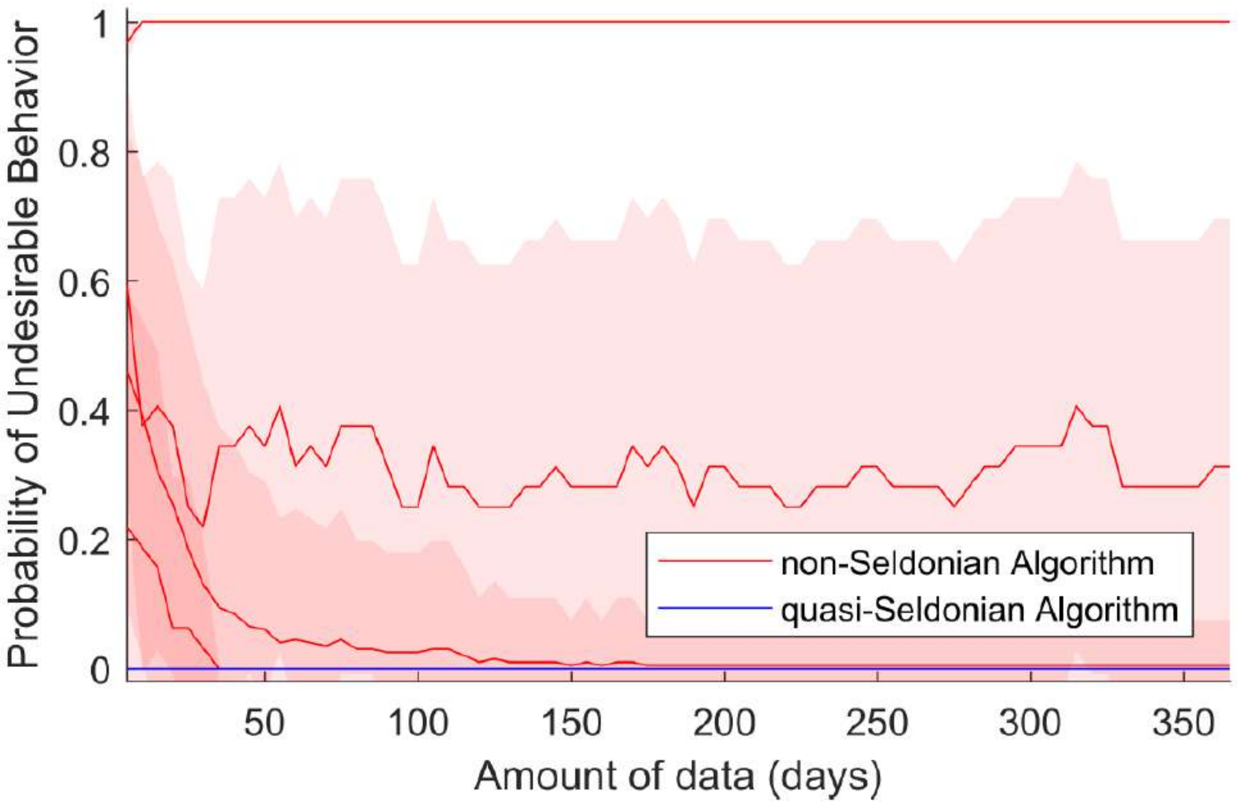
Example: RL (Type 1 Diabetes Management)



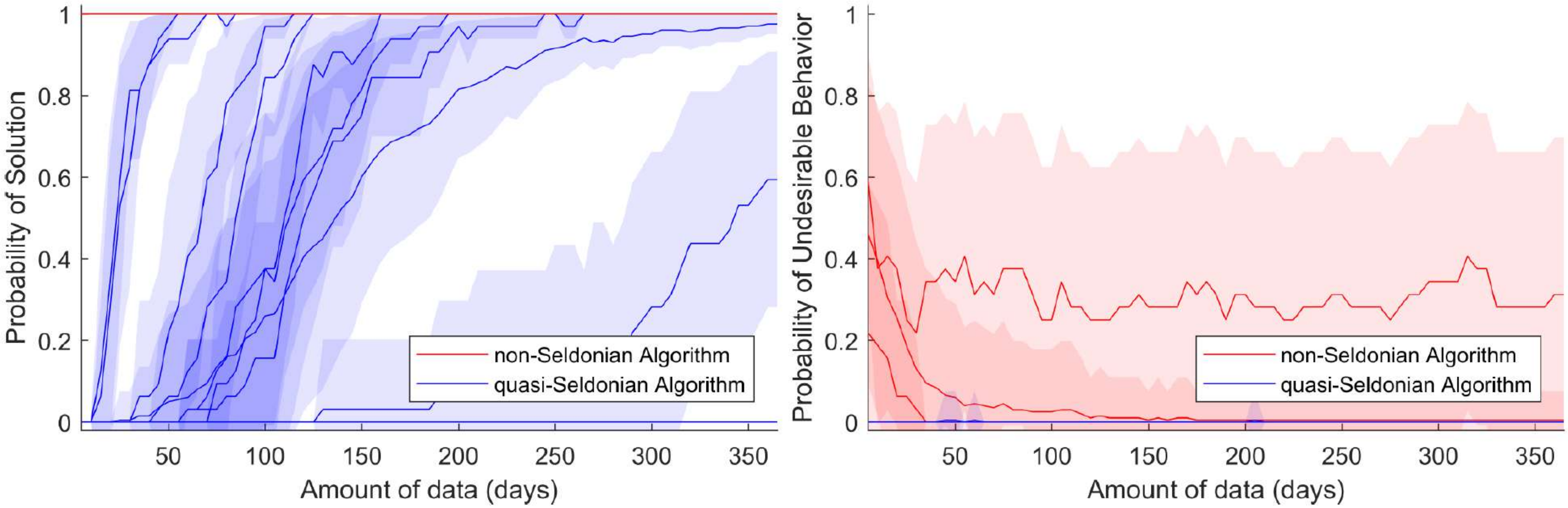


Example: RL (Type 1 Diabetes Management)

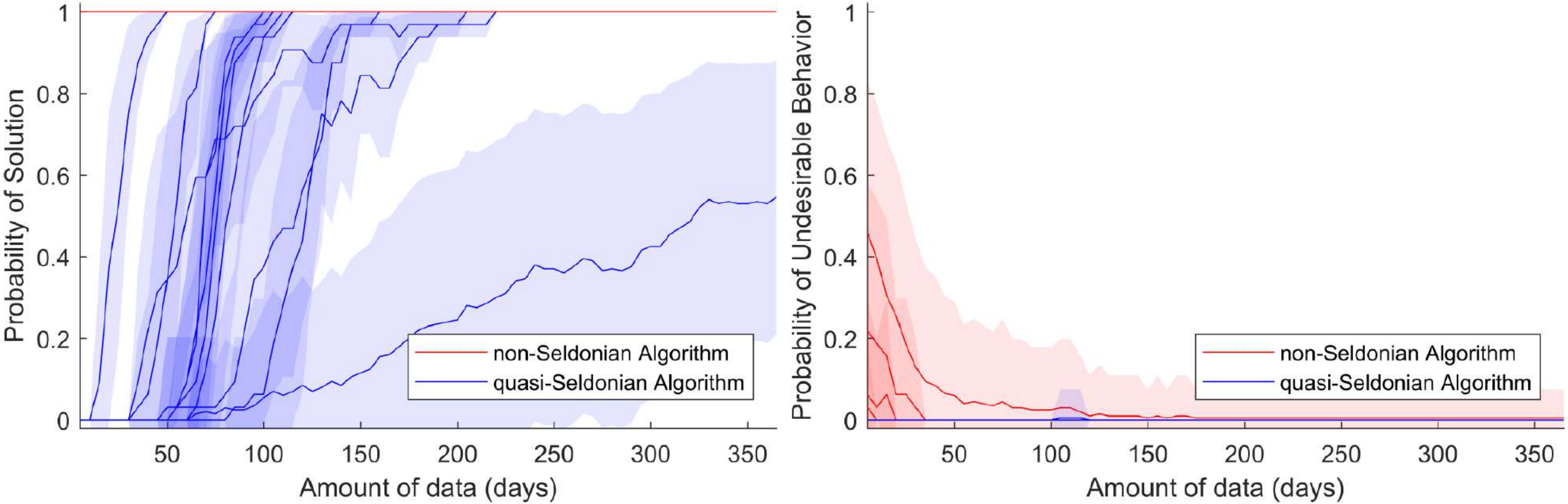




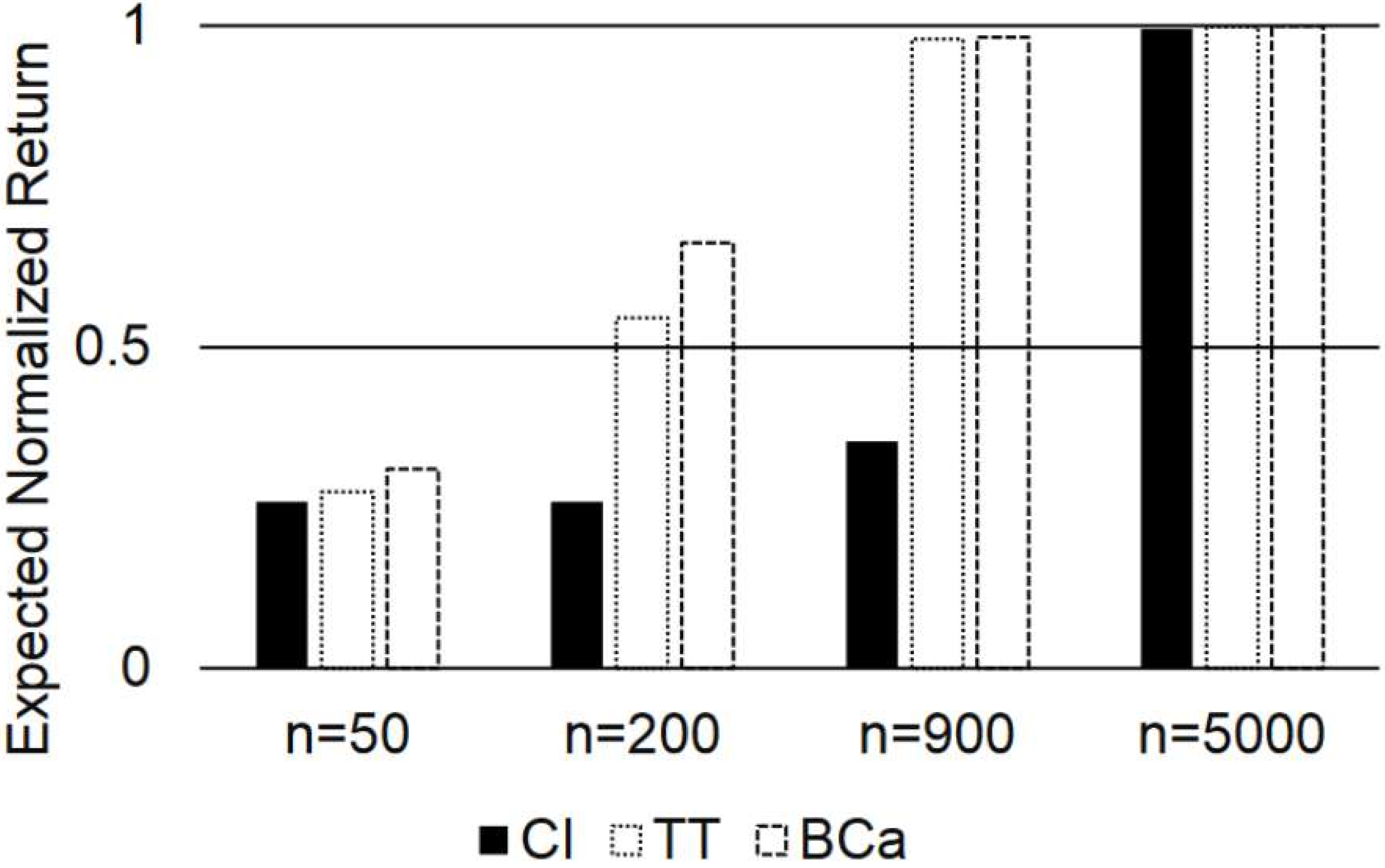
Example: RL (Type 1 Diabetes Management)



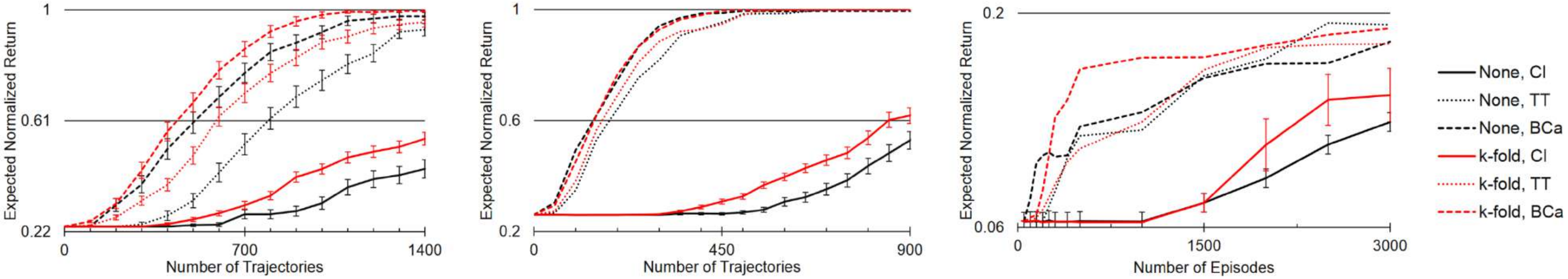
Example: RL (Type 1 Diabetes Management)



Example: RL (HCPI, Mountain Car)



Example: RL (Daedalus)

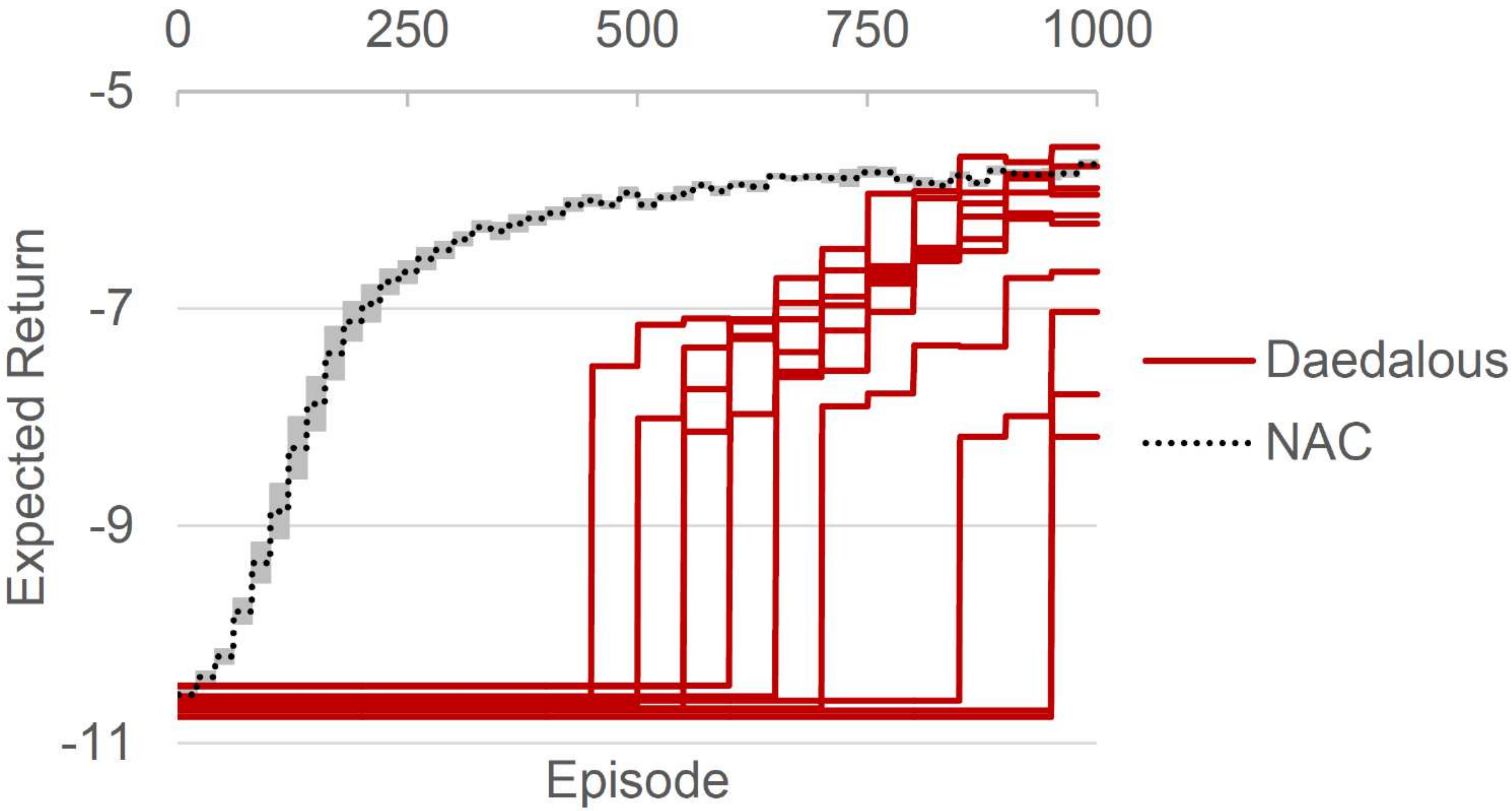


(a) 4 × 4 gridworld results.

(b) Mountain Car results.

(c) Digital marketing results.

Example: RL (Require *significant* improvement)



Future Research Directions

- How to partition data?
- Why NSF (Not enough data? Conflicting constraints? Failed internal prediction? Available δ?)
- How to divide up δ among base variables and solutions / intelligence interval propagation?
- How to trade-off primary objective and predicted-safety-test in candidate selection in a principled way?
- *Secure* Seldonian algorithms
- Combine with reward machines (specification for g, and perhaps f)?
- Multi-Agent RL, (with different constraints on different agents)?
- Extend Fairlearn to be Seldonian / to settings other than classification?
- Improved off-policy estimators for RL safety tests
- Sequential Seldonian algorithms
- Efficient optimization in candidate selection
- Actual HCI interface (natural language?)
- Better concentration inequalities (sequences?)

Watch For:

- High Confidence Policy Improvement (ICML 2015)
- Offline Contextual Bandits with High Probability Fairness Guarantees
    - NeurIPS 2019
- On Ensuring that Intelligent Machines are Well-Behaved
    - 2017 Arxiv paper updated soon!