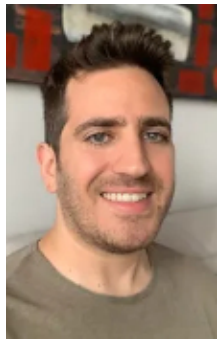
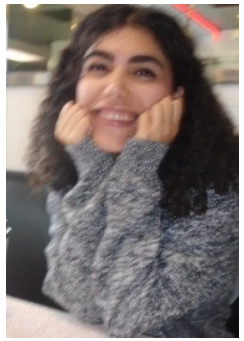
PAC Learning with Improvements

Dravy Sharma (dravy@ttic.edu)

Joint with Idan Attias, Avrim Blum, Keziah Naggita, Donya Saless, Matt Walter



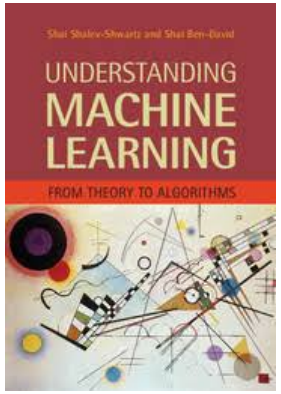

The Setting

Imagine you have a test that is an excellent predictor of whether someone will be successful at some task.

- Be a good employee

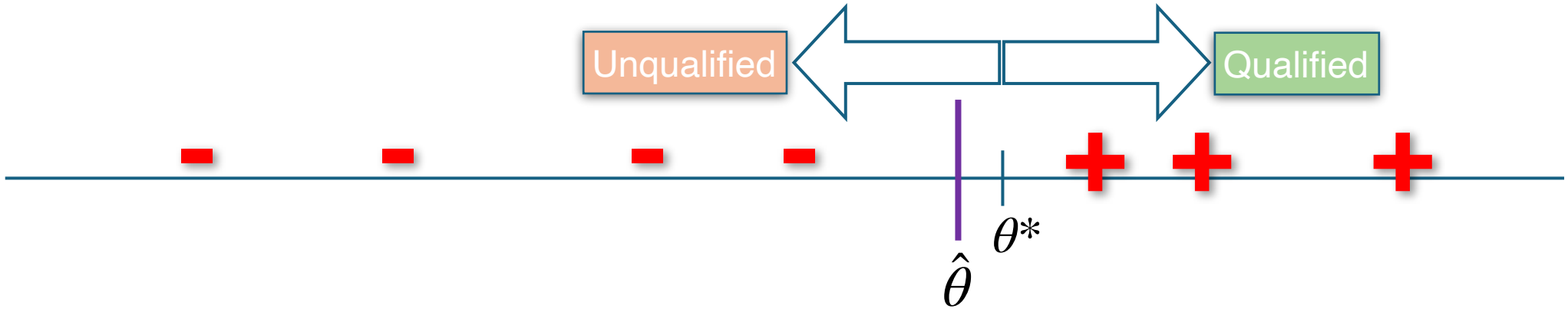
- Safely drive a truck

- Do well in a Machine Learning course



The Setting

Imagine you have a test that is an excellent predictor of whether someone will be successful at some task.

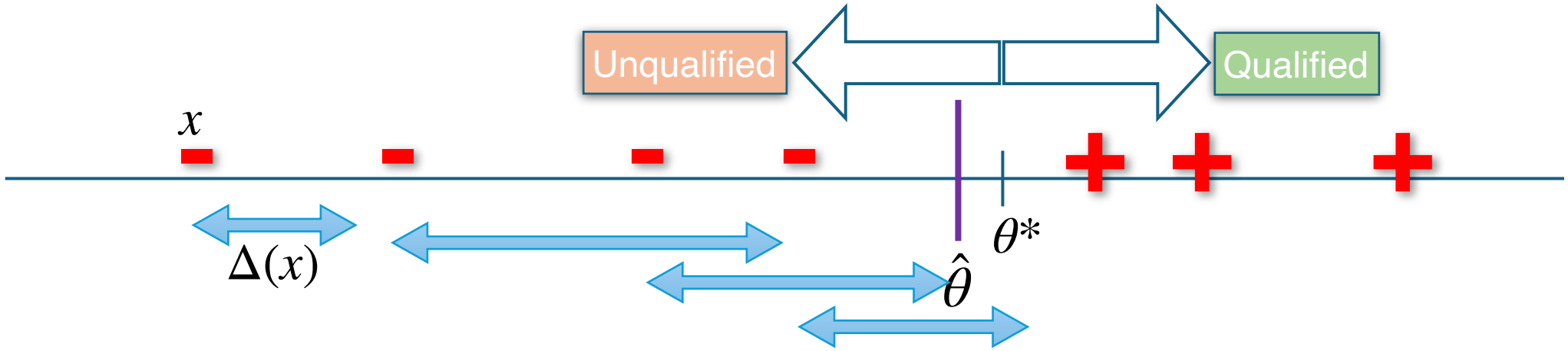


You don't know θ^* but you have past data.

So you publish some test cut-off $\hat{\theta}$.

The Setting

Now suppose people can put in effort to improve (study, practice).

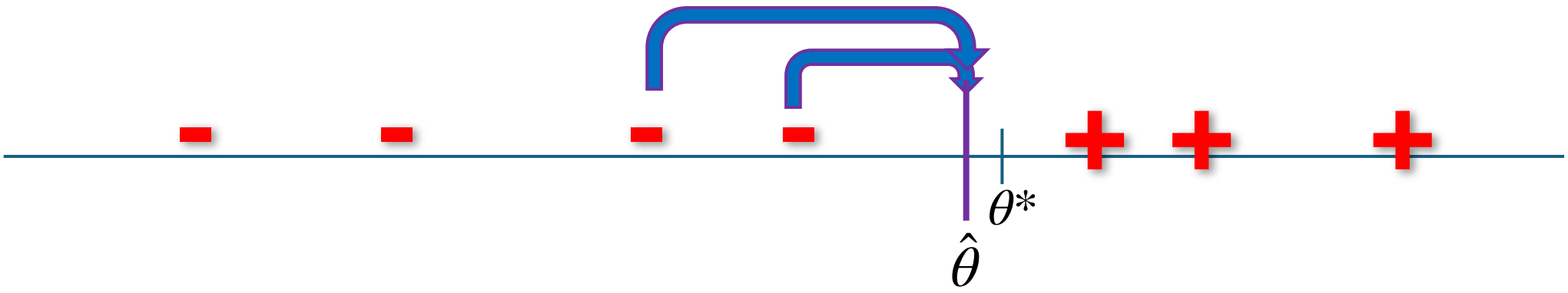


Each person x can improve their score by $\Delta(x)$.

How should we set $\hat{\theta}$?

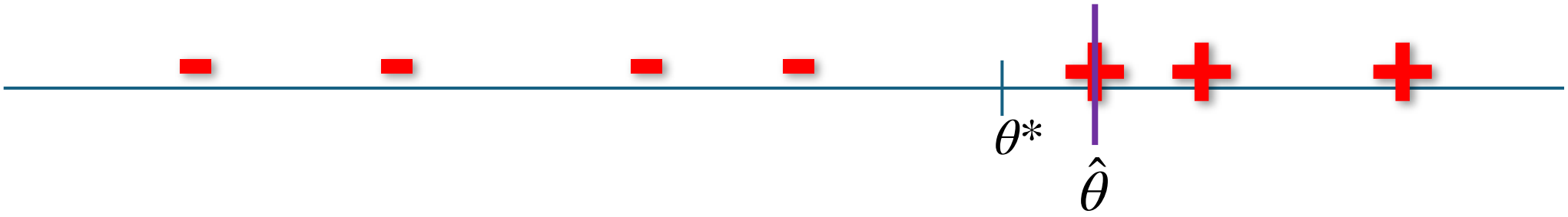
The Setting

- There's a big danger if you put your cutoff too low:
  - people may improve to your cutoff and still not be qualified!



The Setting

- There's a big danger if you put your cutoff too low:
  - people may improve to your cutoff and still not be qualified!
- On the other hand, there's a nice benefit of putting your cutoff safely a little too high (say at the smallest positive example):

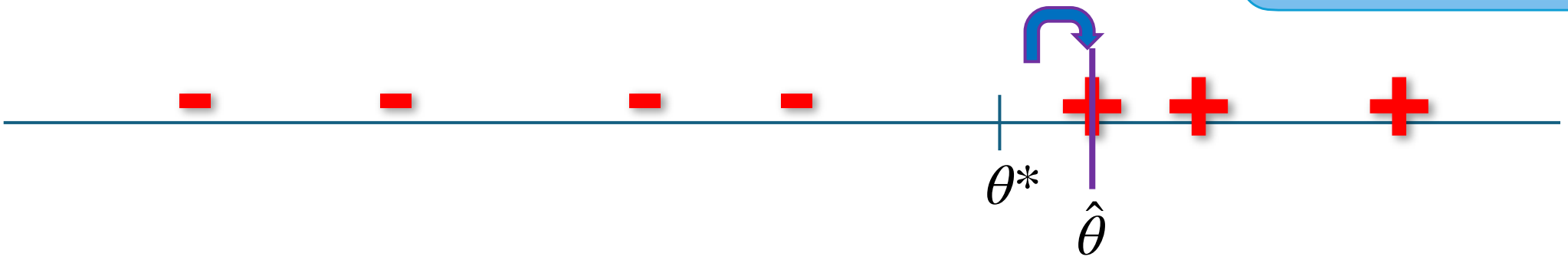


The Setting

- There's a big danger if you put your cutoff too low:
  - people may improve to your cutoff and still not be qualified!
- On the other hand, there's a nice benefit of putting your cutoff safely a little too high (say at the smallest positive example):

➢ If gap to true θ^* is smaller than $\Delta = \min_x \Delta(x)$, then we can actually get ***zero*** error!

Note: Usually you can't get zero error from finite data in PAC learning!



(a) you never have a false positive (everyone you admit is qualified)

(b) any actual positive can put in effort and pass your test, so no false negatives.

Formal Model: PAC Learning with Improvements

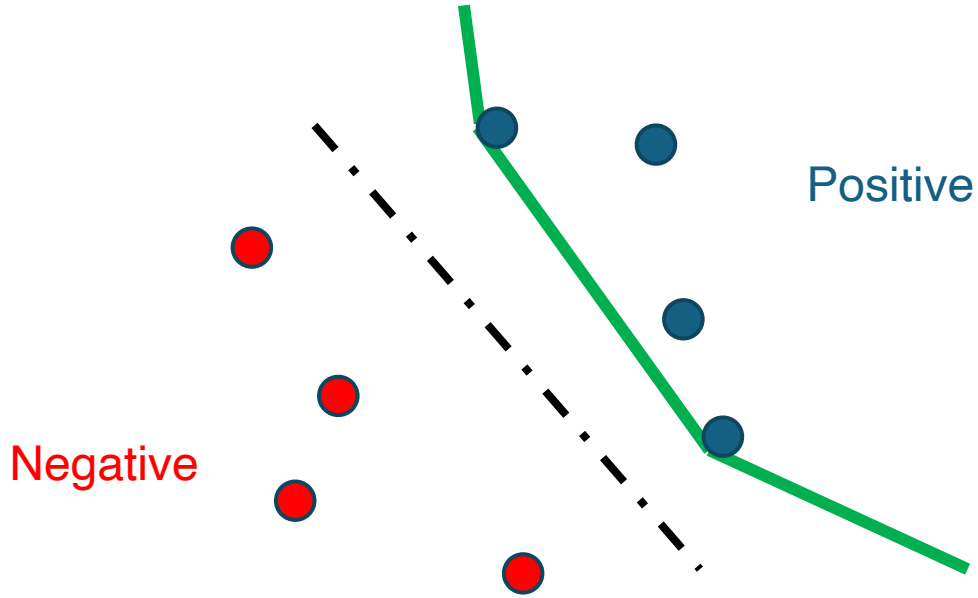
- Assume target function f^* from some family $\mathscr{H}$ (say linear classifiers).
- Agent x has region $\Delta(x)$ that it can (or would be willing to) improve to.

- Assume classifier h is made public.
  - ➢ If $h(x) = 0$ but there is some $x' \in \Delta(x)$ such that $h(x') = 1$, then x will move to some such point (breaking ties adversarially).

- Contrast this with *strategic classification* [Hardt et al. 2016], and adversarial robustness, where agents move to manipulate their features and deceive the classifier.

Some Theoretical Results

- Can have classes of infinite VC-dimension that become easy to learn.
- Can have classes of small VC-dimension that become hard to learn, especially with adversarial tie-breaking.
- In general, the theory favors classifiers that only predict positive when they're sure.
- We get nice sample bounds for achieving zero error whp in several interesting cases.



Some Experimental Results

- We choose some features as modifiable up to some distance $\Delta = r$.
- Train some network h on training data. Then, for each test point, use PGD to simulate agent behavior (like in adversarial ML).
- Different from adversarial ML: we assume the changes are real.
  - We need to know if changing the features actually would have changed the label.
  - Simulate this by training a separate classifier of a different type (a decision tree) to zero error on the entire dataset (training & test) and then using that ***as if*** it were the ground truth.

Evaluation Setup

We evaluate improvement-aware algorithms on three real-world datasets (Adult, OULAD, Law School) and one fully-synthetic dataset.

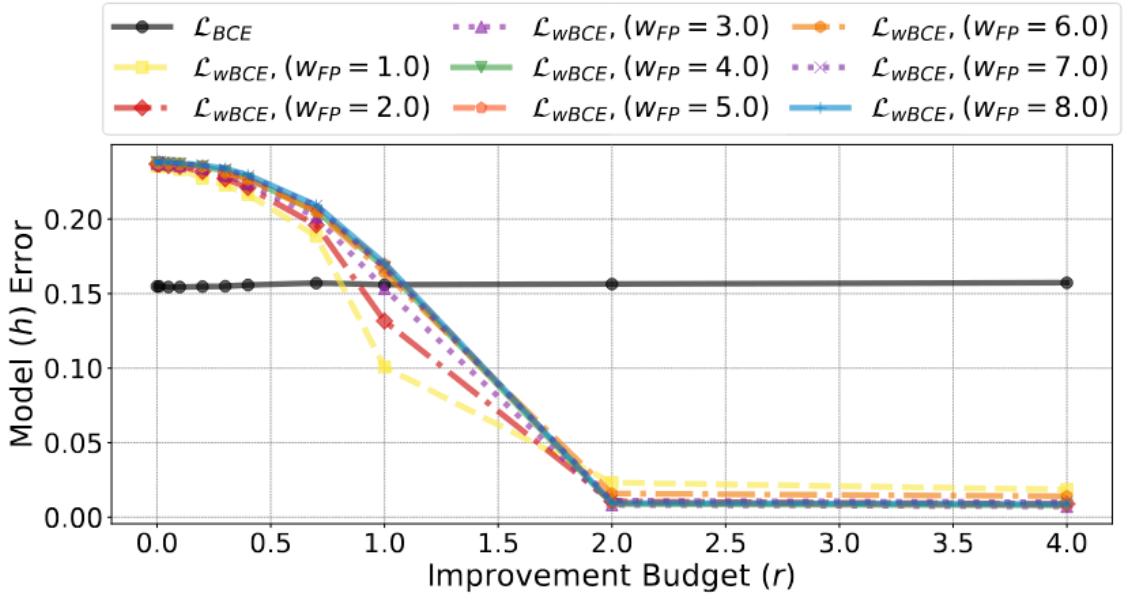
- The true labeling function f^* is a zero-error model.
- The decision-maker classifiers are (modified) neural networks:
  - With weighted (vs. standard) binary cross entropy loss functions.
  - With risk-averse (vs. standard) thresholding.

We compare classifiers trained with standard loss function with those trained with a more risk-averse loss function (higher penalty on false-positives than false negatives)

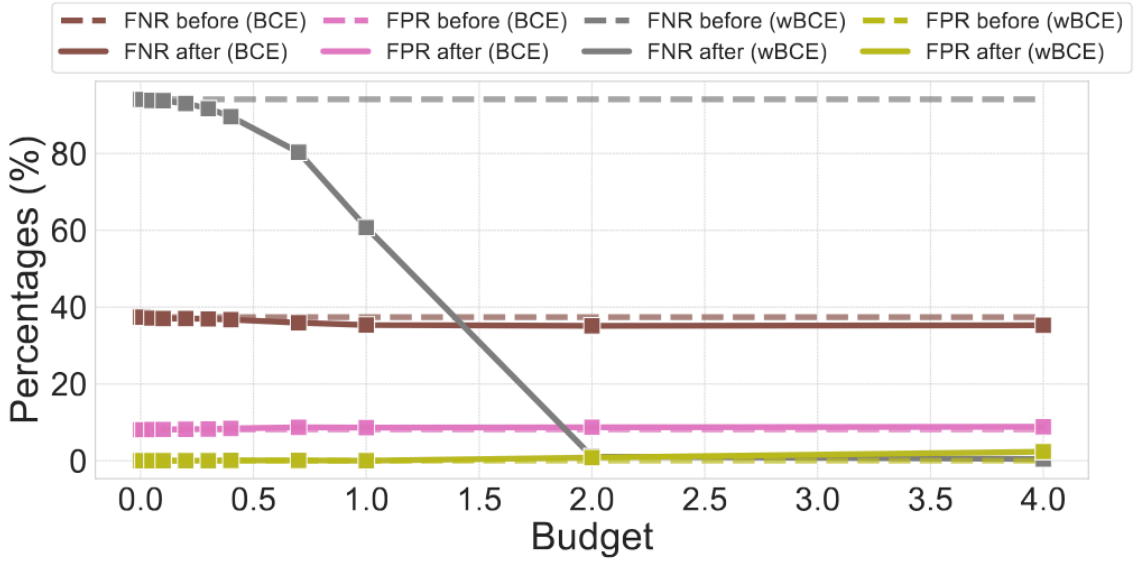
Evaluation Results

Improvement-aware algorithms (risk-averse models) perform better as r increases.

- False negatives decrease, while false positives stay zero.



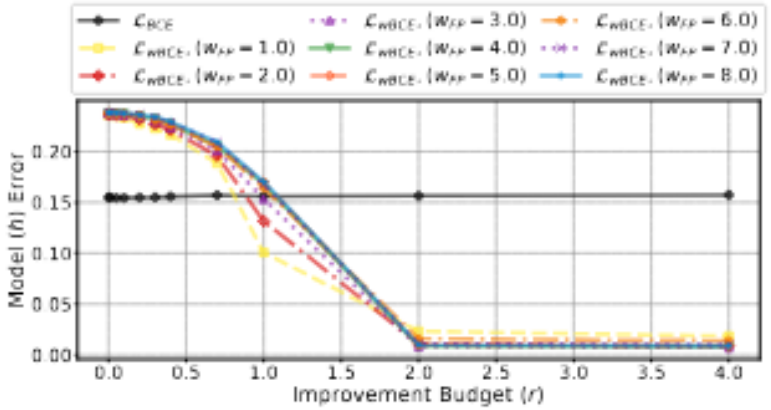
(a) Adult $\left(\mathcal{L}_{\text{wBCE}} \text{ where } w_{\text{FN}} = 0.001\right)$



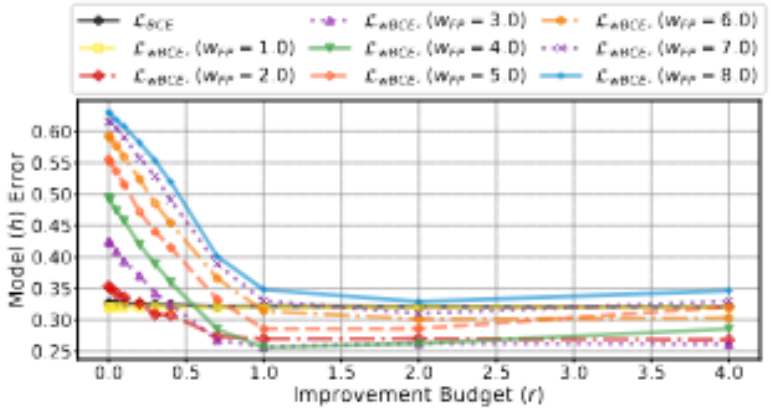
(b) Adult (FNR/FPR before and after agents' improvement)

w_FP = 0.001, w_FN= 4.4

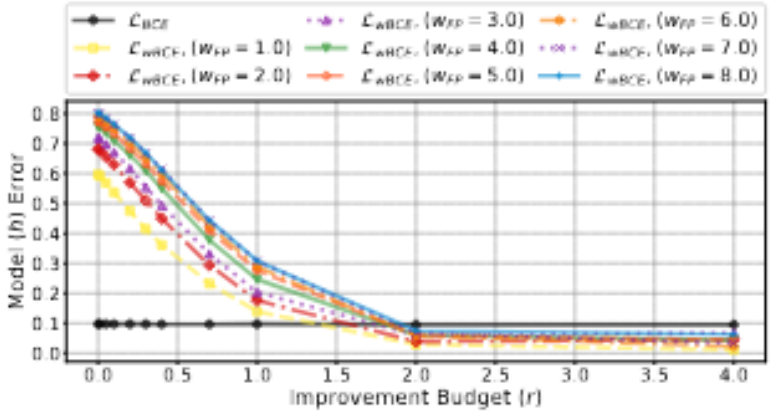
Evaluation Results



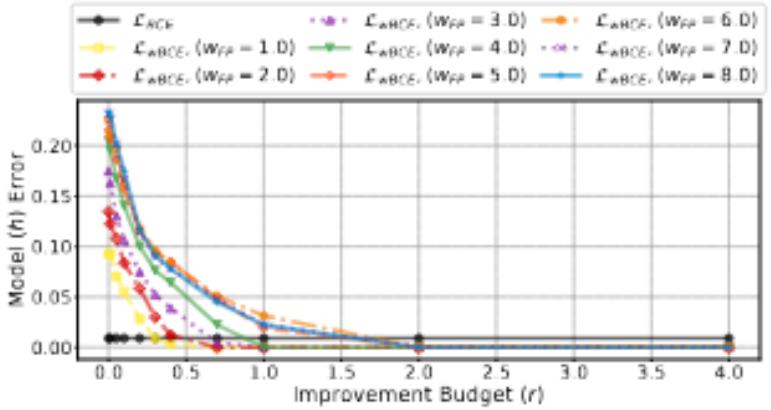
(a) Adult $(w_{\mathbf{FN}} = 0.001)$



(b) OULAD $(w_{\mathbf{FN}} = 1.33)$



(c) Law school $(\mathcal{L}_{\mathbf{wBCE}}$ where $w_{\mathbf{FN}} = 0.01)$



(d) Synthetic $(w_{\mathbf{FN}} = 0.01)$

Figure 2: We compare the performance gains when agents improve to the risk-averse $\left(\mathcal{L}_{\mathbf{wBCE}}, \frac{w_{\mathbf{FP}}}{w_{\mathbf{FN}}} > 1, w_{\mathbf{FP}} = \{i\}_{i=1}^{8}\right)$ and the standard $(\mathcal{L}_{\mathbf{BCE}}, w_{\mathbf{FP}} = w_{\mathbf{FN}} = 1)$ models across four datasets (Adult, OULAD, Law school, and Synthetic) using a fixed classification threshold of 0.5. Higher improvement budgets (r) and greater risk-aversion (high $\frac{w_{\mathbf{FP}}}{w_{\mathbf{FN}}}$) accelerate error reduction. See Figure 10 (Appendix) for a side-by-side comparison with threshold (0.9).

Thank you!

PAC Learning with Improvements

Idan Attias[1,2] **Avrim Blum**[2] **Keziah Naggita**[2]
Donya Saless[2] **Dravyansh Sharma**[2,3] **Matthew Walter**[2*]

Abstract

One of the most basic lower bounds in machine learning is that in nearly any nontrivial setting, it takes *at least* $1/\epsilon$ samples to learn to error ϵ (and more, if the classifier being learned is complex). However, suppose that data points are agents who have the ability to improve by a small amount if doing so will allow them to receive a (desired) positive classification. In that case, we may actually be able to achieve *zero* error by just being "close enough". For example, imagine a hiring test used to measure an agent's skill at some job such that for some threshold θ, agents who score above θ will be successful and those who score below θ will not (i.e., learning a threshold on the line). Suppose also that by putting in effort, agents can improve their skill level by some small amount r. In that case, if we learn an approximation $\hat{\theta}$ of θ such that $\theta \leq \hat{\theta} \leq \theta + r$ and use it for hiring, we can actually achieve error zero, in the sense that (a) any agent classified as positive is truly qualified, and (b) any agent who truly is qualified can be classified as positive by putting in effort. Thus, the ability for agents to improve has the potential to allow for a goal one could not hope to achieve in standard models, namely zero error.

In this paper, we explore this phenomenon more broadly, giving general results and examining under what conditions the ability of agents to improve can allow for a reduction in the sample complexity of learning, or alternatively, can make learning harder. We also examine both theoretically and empirically what kinds of improvement-aware algorithms can take into account agents who have the ability to improve to a limited extent when it is in their interest to do so.