Today: Multicalibration

"Calibration for the (Computationally-Identifiable) Masses", Hebert-Johnson, Kim, Reingold, Rothblum

"Preventing fairness gerrymandering"
Kearns, Neel, Roth, Wu

"Multi-accuracy: Black Box Postprocessing for Fairness in Classif."
Kim, Gorbani, Zu

Multisensitive Attributes

- So far we have only considered a single sensitive group/attribute S
- Many situations involve several sensitive groups
- How to handle?
  How to know what groups to consider?

Multisensitive Attributes

- So far we have only considered a single sensitive group/attribute A
- Many situations involve several sensitive groups
- How to handle?
  How to know what groups to consider?
  Ex. Simpson's paradox / Probublica
  Was the classifier fair?

Achieving Multi-Sensitivity

$\boxed{\text{Idea:}}$ Fix some predefined collection of subsets. Each subset $S \subseteq X$ should be Large and simple

Large: $|S| = \gamma \cdot |X|$

Simple: S is easy to compute.

Let $\mathcal{C} \subseteq 2^X$ be a set of concept classes. Each S is computed by some $c \in \mathcal{C}$.

$\mathcal{C}$ simple: low VC-dimension, or small ht decision trees so subsets easy to identify

Preliminaries

$\mathcal{X}$: universe of N individuals

Want to make prediction about some outcome $o \in \{0,1\}^N$

o_i : 1 with prob. p_i^* (p^* is baseline predictor)

Predictor $x : \mathcal{X} \to [0,1]$, x_i is prediction of p_i^*

Note: Typically $\mathcal{X} = \{0,1\}^d$, so $N = 2^d$

Preliminaries

$\mathcal{X}$: universe of N individuals

Want to make prediction about some outcome $o \in \{0,1\}^N$

o_i : 1 with prob. p_i^* (p^* is baseline predictor)

Predictor $x : \mathcal{X} \to [0,1]$, x_i is prediction of p_i^*

Note: Typically $\mathcal{X} = \{0,1\}^d$, so $N = 2^d$

$\mathcal{C} \subseteq 2^{\mathcal{X}}$. Each group S is the 1's of some $c : \mathcal{X} \to \{0,1\}$

$|\mathcal{C}| = N^{O(1)}$

Multi-Sensitive Fairness Definitions

(1) Accuracy with respect to all subgroups

(2) Well-calibrated for all subgroups

(3) Equalized-odds " " "

(4) Demographic parity " " "

Accuracy in Expectation (AE)

Definition 2.1 (Accurate in expectation)**.** *For any* $\alpha > 0$ *and* $S \subseteq \mathcal{X}$*, a predictor* x *is* α-accurate in expectation (α-AE) *with respect to* S *if*

$$\left| \mathop{\mathbb{E}}_{i \sim S} [x_i - p_i^*] \right| \leq \alpha.$$

Multi-Accuracy in Expectation (AE)

Definition 2.7 (α-multi-AE). *Let $\mathcal{C} \subseteq 2^{\mathcal{X}}$ be a collection of subsets of $\mathcal{X}$ and $\alpha \in [0,1]$. A predictor x is α-multi-AE on $\mathcal{C}$ if for all $S \in \mathcal{C}$, x is α-AE with respect to S.*

We think of $\mathcal{C}$ as a simple concept class, and $S \subseteq \mathcal{X}$ as the set of 1's of a function $c : \mathcal{X} \to [0,1]$, $c \in \mathcal{C}$

For example $\mathcal{C}$ could be all low depth decision trees or small/shallow neural nets, (so typically $|\mathcal{C}| = \text{poly}(N)$).

largness: $\forall c \in \mathcal{C}$, $|c^{-1}(1)| \geq \gamma \cdot N$, $N = |\mathcal{X}|$

Calibration

For $v \in [0,1]$, $S_v = \{i \mid x_i = v\}$

Definition 2.2 (Calibration). *For any $v \in [0,1]$, $S \subseteq \mathcal{X}$, and predictor x, let $S_v = \{i : x_i = v\}$. For $\alpha \in [0,1]$, x is* α-calibrated *with respect to S if there exists some $S' \subseteq S$ with $|S'| \geq (1-\alpha)|S|$ such that for all $v \in [0,1]$,*

$$\left| \mathop{\mathbb{E}}_{i \sim S_v \cap S'} [x_i - p_i^*] \right| \leq \alpha.$$

all but an α-fraction of S, expected value of true probabilities of S_v is α-close to v

Multi Calibration

Definition 2.6 (α-multicalibration). *Let $\mathcal{C} \subseteq 2^{\mathcal{X}}$ be a collection of subsets of $\mathcal{X}$ and $\alpha \in [0,1]$. A predictor x is α-multicalibrated on $\mathcal{C}$ if for all $S \in \mathcal{C}$, x is α-calibrated with respect to S.*

Multi-Calibration Improves accuracy

For $v \in [0,1]$, $S_v = \{i \mid x_i = v\}$

Definition 2.2 (Calibration). *For any $v \in [0,1]$, $S \subseteq \mathcal{X}$, and predictor x, let $S_v = \{i : x_i = v\}$. For $\alpha \in [0,1]$, x is* α-calibrated *with respect to S if there exists some $S' \subseteq S$ with $|S'| \geq (1-\alpha)|S|$ such that for all $v \in [0,1]$,*

$$\left| \mathop{\mathbb{E}}_{i \sim S_v \cap S'} [x_i - p_i^*] \right| \leq \alpha.$$

If x is α-calibrated for S,

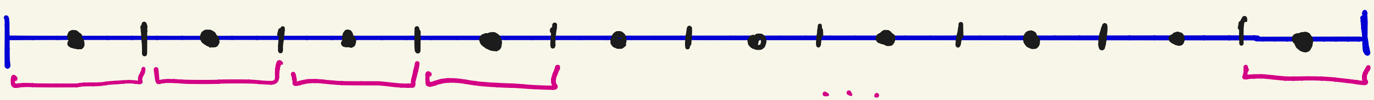
then x is 2α-AE calibrated for S

Calibration with Binning

Definition 2.8 (λ-discretization). *Let $\lambda > 0$. The* λ-discretization *of $[0,1]$, denoted by $\Lambda[0,1] = \{\frac{\lambda}{2}, \frac{3\lambda}{2}, \ldots, 1 - \frac{\lambda}{2}\}$, is the set of $1/\lambda$ evenly spaced real values over $[0,1]$. For $v \in \Lambda[0,1]$, let*

$$\lambda(v) = [v - \lambda/2, v + \lambda/2)$$

be the λ-interval centered around v (except for the final interval, which will be $[1 - \lambda, 1]$).



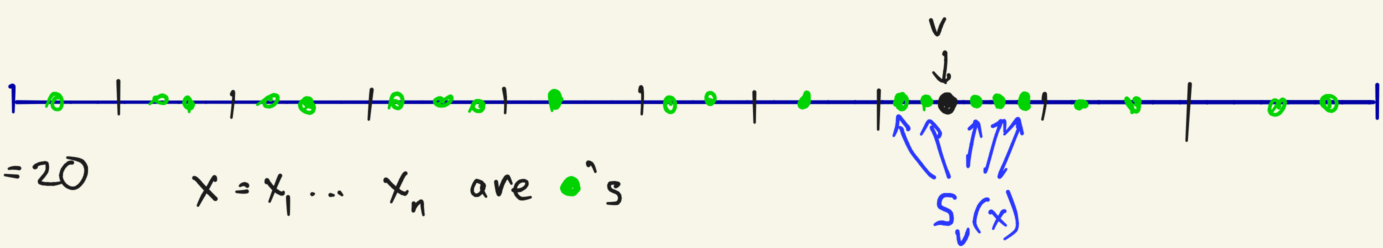
$\lambda = \frac{1}{10}$

Multi-Calibration with Binning

$S_v(x) = \{i : x_i \in \lambda(v)\} \cap S$ for all $S \in \mathcal{C}$ and $v \in \Lambda[0,1]$.

Definition 2.9 ((α, λ)-multicalibration)**.** *Let $\mathcal{C} \subseteq 2^{\mathcal{X}}$ be a collection of subsets of $\mathcal{X}$. For any $\alpha, \lambda > 0$, a predictor x is* (α, λ)-multicalibrated *on $\mathcal{C}$ if for all $S \in \mathcal{C}$, $v \in \Lambda[0,1]$, and all categories $S_v(x)$ such that $|S_v(x)| \geq \alpha\lambda|S|$, we have*

$$\left| \sum_{i \in S_v(x)} x_i - p_i^* \right| \leq \alpha |S_v(x)| .$$



$n = 20$ $\quad x = x_1 \dots x_n$ are ●'s

Claim 2.10. *For $\alpha, \lambda > 0$, suppose $\mathcal{C} \subseteq 2^{\mathcal{X}}$ is a collection of subsets of $\mathcal{X}$. If x is (α, λ)-multicalibrated on $\mathcal{C}$, then x^{λ} is $(\alpha + \lambda)$-multicalibrated on $\mathcal{C}$.*

Multi-accuracy Learning Algorithm for $\mathcal{C}$

- warmup to multi-calibrated Learning algorithm for $\mathcal{C}$
- First we will give a simple iterative algorithm that learns — sample efficient (but not runtime efficient)
- Then show runtime can be upper bounded by runtime of weak agnostic learner for $\mathcal{C}$

Multi-accuracy Learning Algorithm for $\mathcal{C}$

- warmup to multi-calibrated Learning algorithm for $\mathcal{C}$
- First we will give a simple iterative algorithm that learns – sample efficient (but not runtime efficient)

algorithm is a statistical query learning algorithm (only accesses training data via statistical queries)

- Then show runtime can be upper bounded by runtime of weak agnostic learner for $\mathcal{C}$

PAC Learning (p^* unknown, ξ, ε fixed)

Let $\mathcal{D}$ be a distribution over $\mathcal{X}$ (think of $\mathcal{D}$ as uniform distribution).

Learning algorithm $\mathcal{A}$ get n labelled samples

$$\{(i, o_i), \; i = 1 \ldots n\}$$

where i drawn uniformly from $\mathcal{X}$

and $o_i \in \{0,1\}$ drawn from Bernoulli distrib where $p_i^* =$ prob. of 1

$\mathcal{A}$ outputs a hypothesis h such that with prob $> 1-\xi$

$$\|h - p^*\|_2 \leq \varepsilon$$

Statistical Query Learning Algorithms

- PAC learning where access to training data (labelled samples) is restricted

Definition 2.4 (Statistical Query [Kea98]). *For a subset of the universe $S \subseteq \mathcal{X}$, let $p_S^* = \sum_{i \in S} p_i^*$. For $\tau \in [0,1]$,* a statistical query with tolerance τ *returns some $\tilde{p}(S)$ satisfying*

$$p_S^* - \tau N \leq \tilde{p}(S) \leq p_S^* + \tau N.$$

Learning a Multi-Accurate Predictor

1. Start with estimate $x = \frac{1}{2}, \frac{1}{2} \ldots \frac{1}{2}$

2. Iteratively:

   Find some S such that $p^*(S)$ is far from x_S

   Update x_S accordingly

   When no such S is found, output x

Potential Argument (Main idea)

1. Initially $\|p^* - x\|_2^2$ is at most N

2. Everytime we find an S where accuracy on S is bad, since S is Large, the updated $\|p^* - x\|_2^2$ will drop by αN

$\therefore$ algorithm iterates for $\frac{1}{\alpha}$ steps

Potential Argument (Main idea)

1. Initially $\|p^* - x\|_2^2$ is at most N

2. Everytime we find an S where accuracy on S is bad, since S is Large, the updated $\|p^* - x\|_2^2$ will drop by αN

$\therefore$ algorithm iterates for $\frac{1}{\alpha}$ steps

Since we only get an estimate $\tilde{p}(S)$, analysis slightly more complicated, and number of iterations is polynomial in $\frac{1}{\alpha}$, $\frac{1}{\tau}$

Algorithm 3.1 – Learning an α-multi-AE predictor on $\mathcal{C}$

Let $\alpha, \gamma > 0$ and let $\mathcal{C} \subseteq 2^{\mathcal{X}}$ be such that for all $S \in \mathcal{C}$, $|S| \geq \gamma N$.
For $S \subseteq \mathcal{X}$, let $\tilde{p}(S)$ be the output of a statistical query with tolerance $\tau < \alpha\gamma/4$.

- Initialize:
  - Let $x = (1/2, \dots, 1/2) \in [0,1]^N$
- Repeat:
  - For each $S \in \mathcal{C}$:
    - Let $\Delta_S = \tilde{p}(S) - \sum_{i \in S} x_i$
    - If $|\Delta_S| > \alpha |S| - \tau N$:
      update $x_i \leftarrow x_i + \frac{\Delta_S}{|S|}$ for all $i \in S$ (projecting x_i onto $[0,1]$ if necessary)
  - If no $S \in \mathcal{C}$ updated: exit and output x

Parameters: Take $\gamma = \alpha$, $\tau = \alpha^2/4$

Lemma 3.2. *Suppose $\alpha, \gamma > 0$ and $\mathcal{C} \subseteq 2^{\mathcal{X}}$ such that for all $S \in \mathcal{C}$, $|S| \geq \gamma N$. Let $\tau = \alpha\gamma/4$. Then Algorithm 3.1 makes $O(1/\alpha^2\gamma)$ updates to x before terminating.*

Proof. We use a potential argument, tracking the progress the algorithm makes on each update in terms of the ℓ_2^2 distance between our learned predictor x and the true predictions p^*. Let x' be the predictor after updating x on set S and let $\pi : \mathbb{R} \to [0,1]$ denote projection onto $[0,1]$. We use the fact that the ℓ_2^2 can only decrease under this projection. For notational convenience, let $\delta_S = \dfrac{\Delta_S}{|S|} = \dfrac{1}{|S|}(\tilde{p}(S) - \sum_{i \in S} x_i)$. We have

$$\begin{aligned}
\|p^* - x\|^2 - \|p^* - x'\|^2 &= \sum_{i \in S}(p_i^* - x_i)^2 - \sum_{i \in S}(p_i^* - \pi(x_i + \delta_S))^2 \\
&\geq \sum_{i \in S}((p_i^* - x_i)^2 - (p_i^* - (x_i + \delta_S))^2) \\
&= \sum_{i \in S}(2(p_i^* - x_i)\delta_S - \delta_S^2) \\
&= \left(2\delta_S \sum_{i \in S}(p_i^* - x_i)\right) - \delta_S^2 |S| \\
&\geq 2\delta_S \left(\delta_S |S| - \operatorname{sgn}(\delta_S)\tau N\right) - \delta_S^2 |S| \\
&\geq \delta_S^2 |S| - 2|\delta_S| \tau N.
\end{aligned}$$

By setting $\tau = \alpha\gamma/4$ and by the bound $|\Delta_S| \geq \alpha |S| - \tau N \geq 3\alpha |S|/4$, the final quantity is at least $\Omega(\alpha^2 |S|)$.

$$\delta_S^2 |S| - 2 |\delta_S| \tau N \geq \left(\frac{3\alpha}{4}\right)^2 |S| - 2\left(\frac{3\alpha}{4}\right)\left(\frac{\alpha\gamma}{4}\right) N$$
$$= \frac{3\alpha^2}{16} |S| .$$

The ℓ_2^2 distance between p^* and any other predictor (in particular, our initial choice for x) is upper-bounded by N. Thus, given that all $S \in \mathcal{C}$ have $|S| \geq \gamma N$, we make at least $\Omega(\alpha^2\gamma N)$ progress in potential at each update, so the lemma follows. $\square$

Theorem 3.3. *For* $\alpha, \gamma > 0$ *and for any* $\mathcal{C} \subseteq 2^{\mathcal{X}}$ *satisfying* $|S| \geq \gamma N$ *for all* $S \in \mathcal{C}$*, there is a statistical query algorithm with tolerance* $\tau = \alpha\gamma/4$ *that learns a* α*-multi-AE predictor on* $\mathcal{C}$ *in* $O(|\mathcal{C}|/\alpha^2\gamma)$ *queries.*

Sample Complexity

Corollary 3.4. *Suppose* $\alpha, \gamma, \xi > 0$ *and* $\mathcal{C} \subseteq 2^{\mathcal{X}}$ *is such that for all* $S \in \mathcal{C}$, $|S| \geq \gamma N$. *Then there is an algorithm that learns an* α*-multi-AE predictor on* $\mathcal{C}$ *with probability at least* $1 - \xi$ *from* $n = \tilde{O}\left(\frac{\log(|\mathcal{C}|/\xi)}{\alpha^2 \gamma}\right)$ *samples.*

Sample Complexity

Corollary 3.4. *Suppose $\alpha, \gamma, \xi > 0$ and $\mathcal{C} \subseteq 2^{\mathcal{X}}$ is such that for all $S \in \mathcal{C}$, $|S| \geq \gamma N$. Then there is an algorithm that learns an α-multi-AE predictor on $\mathcal{C}$ with probability at least $1 - \xi$ from $n = \tilde{O}\left(\frac{\log(|\mathcal{C}|/\xi)}{\alpha^2 \gamma}\right)$ samples.*

Proof (sketch)

1. Use Chernoff bounds to show for any $S \in \mathcal{C}$ with n samples, the probability that empirical estimate $\hat{p}(S)$ of $p^*(S) = \frac{1}{|S|} \sum_{i \in S} p_i^*$ is $> \gamma N$ far away is $\Delta \ll \frac{1}{|\mathcal{C}|}$

2. By union bound, overall bad probability is $\leq \Delta |\mathcal{C}| \ll 1$

Multi-Calibrated Learning Algorithm

Theorem 2. *Suppose $\mathcal{C} \subseteq 2^{\mathcal{X}}$ is collection of sets such that for all $S \in \mathcal{C}$, $|S| \geq \gamma|X|$, and suppose set membership can be evaluated in time t. Then there is an algorithm that learns a predictor of $p^* : \mathcal{X} \to [0,1]$ that is α-multicalibrated on $\mathcal{C}$ from $O(\log(|\mathcal{C}|)/\alpha^{11/2}\gamma^{3/2})$ samples in time $O(|\mathcal{C}| \cdot t \cdot \mathrm{poly}(1/\alpha, 1/\gamma))$.*

Multi-Calibrated Learning Algorithm

- Divide $[0,1]$ into $1/\lambda$ bins
- Run previous algorithm but now run over all pairs $(S, \lambda(v))$, $v \in \Lambda[0,1]$ such that $|S_v|$ is large
  $S_v = \{ i \mid x_i \in \text{interval } \lambda(v) \text{ and } i \in S \}$

  If estimate of S_v is bad, fix it
- Same analysis but now union bound over $|\mathcal{C}| \cdot \frac{1}{\lambda}$ pairs

Algorithm: Learning a (α,λ)-multi-calibrated predictor for $\mathcal{C}$

Let $\alpha, \gamma > 0$ and let $\mathcal{C} \subseteq 2^{\mathcal{X}}$ be such that for all $S \in \mathcal{C}$, $|S| \geq \gamma N$.
For $S \subseteq \mathcal{X}$, let $\tilde{p}(S)$ be the output of a statistical query with tolerance $\tau < \alpha\gamma/4$.

- Initialize:
  - Let $x = (1/2, \ldots, 1/2) \in [0,1]^N$
- Repeat:
  - For each $S \in \mathcal{C}$, $v \in \Lambda[0,1]$, for each $S_v = S \cap \{i \mid x_i \in \lambda(v)\}$ such that $|S_v| > \alpha\lambda|S|$
    - Let $\Delta_{S_v} = \tilde{p}(S_v) - \sum_{i \in S_v} x_i$
    - If $|\Delta_{S_v}| > \alpha\,|S_v| - \tau N$:
      update $x_i \leftarrow x_i + \frac{\Delta_{S_v}}{|S_v|}$ for all $i \in S_v$ (projecting x_i onto $[0,1]$ if necessary)
  - If no S_v updated: exit and output x

Parameters: Take $\gamma = \alpha$, $\tau = \alpha^2/4$

Multi-Calibrated Learning Algorithm

- Divide $[0,1]$ into $1/\lambda$ bins
- Run previous algorithm but now run over all pairs $(S, \lambda(v))$, $v \in \Lambda[0,1]$ such that $|S_v|$ is large

  $S_v = \{ i \mid x_i \in \text{interval } \lambda(v) \text{ and } i \in S \}$

  If estimate of S_v is bad, fix it
- Same analysis but now union bound over $|\mathcal{C}| \cdot \frac{1}{\lambda}$ pairs

Naive analysis gives $\frac{1}{\alpha^4 \gamma^4}$ iterations,

and $n \simeq \frac{1}{\alpha^6 \gamma^6}$ samples

Multi-Calibrated Learning Algorithm

- Divide $[0,1]$ into $1/\lambda$ bins
- Run previous algorithm but now run over all pairs $(S, \lambda(v))$, $v \in \Lambda[0,1]$ such that $|S_v|$ is <u>large</u>

  $S_v = \{ i \mid x_i \in \text{interval } \lambda(v) \text{ and } i \in S \}$

  If estimate of S_v is bad, fix it
- Same analysis but now union bound over $|\mathcal{C}| \cdot \frac{1}{\lambda}$ pairs

More complicated analysis using differential privacy gives $\frac{1}{\alpha^4 \gamma}$ queries, $\frac{1}{\alpha^{5/2} \gamma^{3/2}}$ samples

Bad News

Algorithm is sample-efficient but terrible runtime -- $\Omega(|\mathcal{C}|)$, and $|\mathcal{C}|$ is typically $> N$, where N is universe size

Bad News

Algorithm is sample-efficient but terrible runtime -- $\Omega(|\mathcal{C}|)$, and

$|\mathcal{C}|$ is typically $> N$, where N is universe size

good News

Achieving (α, λ) multi-calibrated learning for $\mathcal{C}$ is no harder than learning $\mathcal{C}$

(in fact it is polynomially equivalent to weak agnostic learning $\mathcal{C}$)

Efficient agnostic learning algorithm for $\mathcal{C}$

$\Rightarrow$ efficient multi-calibrated learner for $\mathcal{C}$

Theorem 3 (Informal)**.** *If there is a weak agnostic learner for* $\mathcal{C}$ *that runs in time* T*, then there is an algorithm for learning an* α*-multicalibrated predictor on* $\mathcal{C}' = \{S \in \mathcal{C} : |S| \gamma |X|\}$ *that runs in time* $O(T \cdot \mathrm{poly}(1/\alpha, 1/\gamma))$.

Efficient multi-calibrated learner for $\mathcal{C}$

$\Rightarrow$ efficient agnostic learning algorithm for $\mathcal{C}$

Theorem 4 (Informal). *If there is an algorithm for learning an α-multicalibrated predictor on a collection of sets $\mathcal{C}' = \{S \in \mathcal{C} : |S| \geq \gamma N\}$ that runs in time T, then there is an algorithm that implements a (ρ, τ)-weak agnostic learner in time $O(T \cdot \mathrm{poly}(1/\tau))$ for any $\rho > 0$ where $\tau = \mathrm{poly}(\rho, \gamma, \alpha)$.*

Agnostic Learning $\mathcal{C}$

Let $\mathcal{D}$ be a distribution over $\mathcal{X}$

A (ρ, τ) - weak agnostic learner $\mathcal{L}$ for $\mathcal{C}$ over $\mathcal{D}$ solves the following problem:

given samples $\{(i, y_i)\}$, where $i \sim \mathcal{D}$, $y_i \in [-1, 1]$ such that some concept $c \in \mathcal{C}$ has high correlation with the samples: $\langle c, y \rangle_{\mathcal{D}} > \rho$,

$\mathcal{L}$ returns some hypothesis $h : \mathcal{X} \to [-1, 1]$ such that $\langle h, y \rangle_{\mathcal{D}} > \tau$

Efficient agnostic learning algorithm for $\mathcal{C}$

$\Rightarrow$ efficient multi-accurate learner for $\mathcal{C}$

Theorem 3 (Informal). *If there is a weak agnostic learner for $\mathcal{C}$ that runs in time T, then there is an algorithm for learning an α-multi-accurate predictor on $\mathcal{C}' = \{S \in \mathcal{C} : |S| \gamma |X|\}$ that runs in time $O(T \cdot \mathrm{poly}(1/\alpha, 1/\gamma))$.*

IDEA Instead of bruteforce search over all subgroups S to find one where $\|p^*(S) - x_S\|_2$ is large, use agnostic learner A to find some S' that is close to S, and update x accordingly.

Efficient agnostic learning algorithm for $\mathcal{C}$

$\Rightarrow$ efficient multi-accurate learner for $\mathcal{C}$

Theorem 3 (Informal). *If there is a weak agnostic learner for $\mathcal{C}$ that runs in time T, then there is an algorithm for learning an α-multi-accurate predictor on $\mathcal{C}' = \{S \in \mathcal{C} : |S| \gamma |X|\}$ that runs in time $O(T \cdot \mathrm{poly}(1/\alpha, 1/\gamma))$.*

Idea: assume $\mathcal{A}$ is a weak agnostic learner for $\mathcal{C}$

If some $c \in \mathcal{C}$ has $\underbrace{\left\| \sum_{i \in c^{-1}(1)} x_i - p_i^* \right\|}_{\Delta_c}{}_2 > \alpha \, |c^{-1}(1)|$

Then since $c^{-1}(1)$ is large, $\langle c, \Delta_c \rangle > \rho$

(c is correlated with Δ_c)

So run agnostic learner $\mathcal{A}$ on samples

$(i, \; x_i - p_i^*)$

ACHIEVING MULTIACCURACY

Kim, Gorbani, Zou, 2018

Main idea: Auditor iteratively uses a binary classifier to find "sensitive variable(s)" that most violates multiaccuracy, and improves current classifier to satisfy it

Postprocessing procedure (boosting style)

Definition (Multiaccuracy auditing). *Let* $\alpha > 0, m \in \mathbb{N}$, *and let* $\mathcal{A} : \mathcal{X}^m \to [-1,1]^{\mathcal{X}}$ *be a learning algorithm. Suppose* $D \sim \mathcal{D}^m$ *is a set of independent random samples. A hypothesis* $f : \mathcal{X} \to (0,1)$ *passes* $(\mathcal{A}, \alpha)$-multiaccuracy auditing *if for* $h = \mathcal{A}(D)$*:*

$$\mathop{\mathbf{E}}_{x\sim\mathcal{D}}\left[h(x)\cdot(f(x)-y(x))\right] \leq \alpha. \tag{2}$$

Multiaccuracy-Boost algorithm:

1. Starts with black-box classifier f_0
2. Iterative post-processing algorithm like boosting:

- Auditor identifies most sub-optimal predictions
- Classifier uses multiplicative weights to improve those predictions, not harm others

ACHIEVING MULTIACCURACY

Kim, Gorbani, Zou, 2018

Main idea: Auditor iteratively uses a binary classifier to find "sensitive variable(s)" that most violates multiaccuracy, and improves current classifier to satisfy it

Postprocessing procedure (boosting style)

Definition (Multiaccuracy auditing). *Let* $\alpha > 0, m \in \mathbb{N}$, *and let* $\mathcal{A} : \mathcal{X}^m \to [-1,1]^{\mathcal{X}}$ *be a learning algorithm. Suppose* $D \sim \mathcal{D}^m$ *is a set of independent random samples. A hypothesis* $f : \mathcal{X} \to (0,1)$ *passes* $(\mathcal{A}, \alpha)$-multiaccuracy auditing *if for* $h = \mathcal{A}(D)$*:*

$$\mathop{\mathbf{E}}_{x\sim\mathcal{D}} [h(x) \cdot (f(x) - y(x))] \leq \alpha. \quad (2)$$

p_i^*

x is i $\quad x_i$

h is close to some $c \in \mathcal{C}$

No c is correlated with $x - p^*$

Multiaccuracy-Boost algorithm:

1. Starts with black-box classifier f_0
2. Iterative post-processing algorithm like boosting:

- Auditor identifies most sub-optimal predictions
- Classifier uses multiplicative weights to improve those predictions, not harm others

MULTIACCURACY BOOST

Given: initial hypothesis $f_0 : \mathcal{X} \to (0,1)$; auditing algorithm $\mathcal{A}$; accuracy parameter $\alpha > 0$;
validation data $D = D_0, \ldots, D_T \sim \mathcal{D}^m$:

$\mathcal{X}_0 \leftarrow \{x \in \mathcal{X} : f_0(x) \leq 1/2\}$
$\mathcal{X}_1 \leftarrow \{x \in \mathcal{X} : f_0(x) > 1/2\}$ `Partition X according to f0`
$\mathcal{S} \leftarrow \{\mathcal{X}, \mathcal{X}_0, \mathcal{X}_1\}$

Repeat: from $t = 0, 1, \ldots$

- For $S \in \mathcal{S}$:
  $h_{t,S} \leftarrow \mathcal{A}^{f_t}(D_t)$ `// audit current hypothesis ft on X, X0, and X1`
- $S^* \leftarrow \text{argmax}_{S \in \mathcal{S}} \mathbf{E}_{x \sim D_t}[h_{t,S}(x) \cdot (f_t(x) - y(x))]$ `// take largest residual`
- **if** $\mathbf{E}_{x \sim D_t}[h_{t,S^*}(x) \cdot (f_t(x) - y(x))] \leq \alpha$: `// terminate when at most alpha`
  return f_t
- $f_{t+1}(x) \propto e^{-\eta h_{t,S^*}(x)} f_t(x) \;\; \forall x \in S^*$ `// multiplicative weights update`

Intuition – h_t based on gradient of (cross-entropy) loss wrt predictions *f(x)*

Code available online

EXPERIMENTS

1). **Adult**:

- gender and race removed
- train 2-layer network on 27K individuals
- Multiaccuracy boost on 31K validation examples

Stage	All	F	M	B	W	BF	BM	WF	WM
Population Percentage (%)	100.0	32.3	67.7	86.1	9.2	4.6	4.7	26.2	59.9
Initial Model (%)	19.3	9.3	24.2	10.5	20.3	4.8	15.8	9.8	24.9
MULTIACCURACY BOOST (%)	**14.7**	**7.2**	**18.3**	**9.4**	**15.0**	**4.5**	**13.9**	**7.3**	**18.3**
Subgroup-Specific (%)	19.7	9.5	24.6	10.5	19.9	5.5	15.3	10.2	25.3

Table 1: **Test error rates for Adult Income Data Set**

2). **Faces**:

- Train base network on Celeb-A, classify gender, race removed
- Multiaccuracy boost on LFW+a dataset

Stage	All	F	M	B	N	BF	BM	NF	NM
Population Percentage (%)	100	21.0	79.0	4.9	95.1	2.1	18.8	2.7	76.3
Initial Model (%)	5.4	23.1	**0.7**	10.2	5.1	20.4	**2.1**	23.4	**0.6**
MULTIACCURACY BOOST (%)	**4.1**	**11.3**	3.2	**6.0**	4.9	**8.2**	4.3	**11.7**	3.2
Subgroup-Specific (%)	4.5	14.0	2.0	8.1	**4.4**	14.3	3.2	14.0	2.0
Retraining (%)	4.5	13.5	2.1	**6.0**	**4.4**	8.8	3.7	14.0	2.1

Table 2: **Test error rates for LFW+a gender classification data set.**

"Preventing Fairness Gerrymandering"

Instead of multicalibration use other notions of fairness — equalized odds and statistical parity