Algorithms for multi-group learning

Daniel Hsu
Columbia University

Based on joint work with (and slides of) **Christopher Tosh (MSKCC)**

Workshop on "Multigroup Fairness and the Validity of Statistical Judgment"

April 25, 2023

Motivation: statistical learning

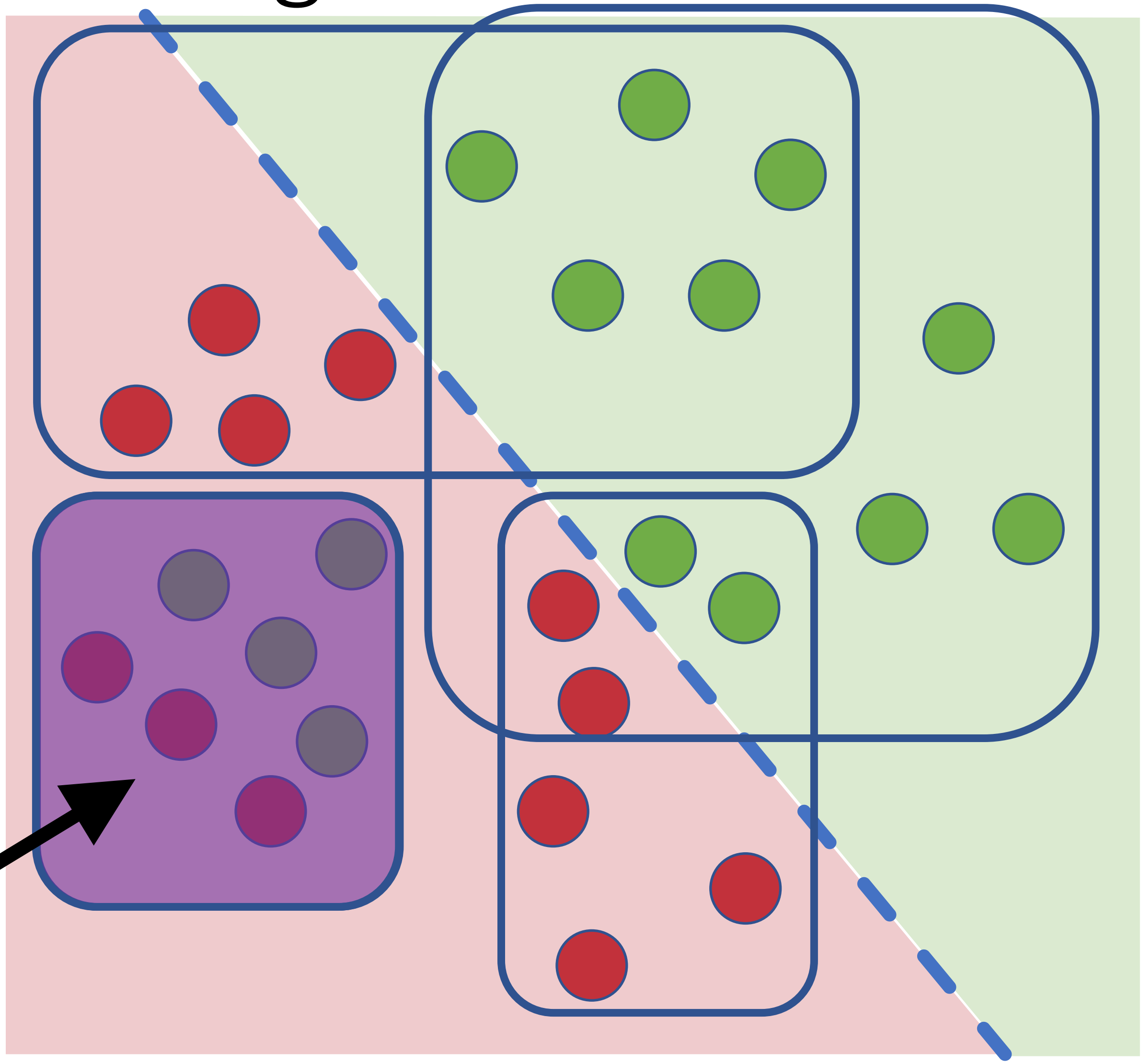
- Aggregate performance over a population P

$$\mathbb{E}_{(x,y)\sim P}[\ell(f(x),y)]$$

- No assurance about any particular instance

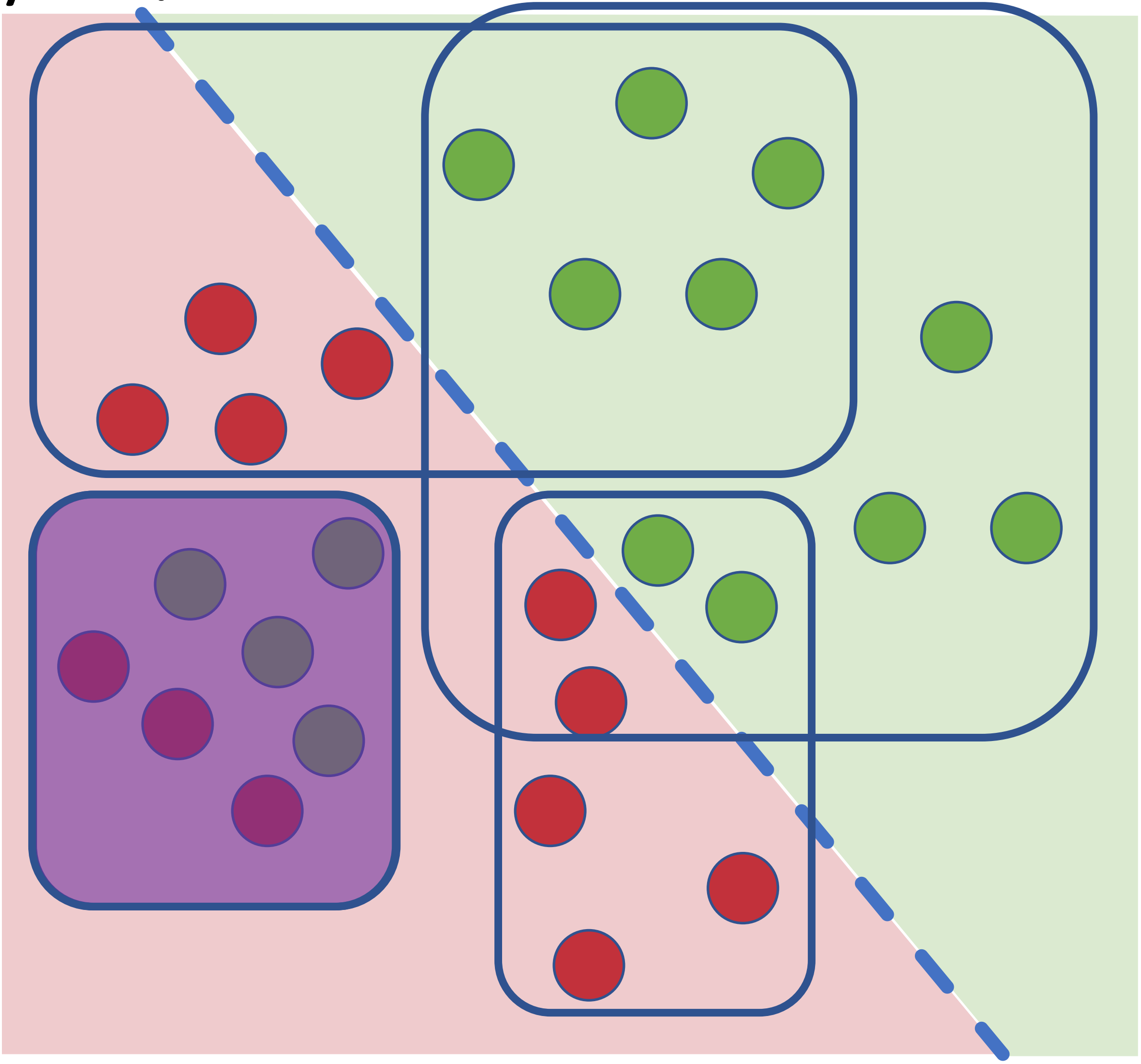
$$\ell(f(x),y)$$

- No assurances even for subpopulations/subgroups



Motivation: trustworthy AI/ML

- Many highlighted failures of AI/ML happen on individual instances & subgroups
- Standard ML objectives fail to address prerequisites for trustworthy AI/ML



Multi-group learning: history

- Formalized by Rothblum and Yona (2021); related to a multi-group extension of "online learning" of Blum and Lykouris (2020)
  - Largely motivated by fairness in ML & trustworthy AI/ML
  - For simplicity, we'll focus on binary classification + error rate objective, but [RY'21] and [BL'20] also consider other objectives (e.g., calibration)
- Our motivation came from "hidden stratification" (Oakden-Rayner, Dummon, Carneiro, and Ré, 2020)
  - Training data is often a data set of convenience, typically stratified
  - Downstream application requires good accuracy on specific strata

High-level summary

- Multi-group learning is a natural generalization of the "classical" setup for supervised learning from statistical learning theory
- Basic sample complexity results from "classical" setup can be extended to multi-group setup…
- …But requires new algorithms
  - In "classical" setup: ERM suffices
  - In multi-group setup: resulting predictors necessarily more complicated

Cast of characters

- $(X,Y)\sim P$ for data distribution P over $\mathcal{X}\times\{0,1\}$
- $\mathcal{H}$ is reference class of functions $\mathcal{X} \to \{0,1\}$ ("hypotheses")
- $\mathcal{G}$ is family of subsets of $\mathcal{X}$ ("groups")
- Eventually assume both $\mathcal{H}$, $\mathcal{G}$ have finite VC dimensions $d_{\mathcal{H}}$, $d_{\mathcal{G}}$

Background: agnostic learning

- **Agnostic learning** (with no groups involved):
  For any $\epsilon \in (0,1)$, given $n = n\left(\frac{1}{\epsilon}, d_{\mathcal{H}}\right)$ iid copies of (X, Y), find classifier $f: \mathcal{X} \to \{0,1\}$ such that, with high probability,

$$\underbrace{P(f(X) \neq Y)}_{\mathrm{err}(f)} \leq \inf_{h \in \mathcal{H}} \underbrace{P(h(X) \neq Y)}_{\mathrm{err}(h)} + \epsilon$$

- Suffices to let f = empirical risk minimizer (ERM) over $\mathcal{H}$
- Optimal sample complexity: $d_{\mathcal{H}} / \epsilon^2$

Multi-group agnostic learning

- **Multi-group agnostic learning** (Rothblum and Yona, 2021):
  For any $\epsilon \in (0,1), \gamma \in (0,1)$, given $n = n\left(\frac{1}{\epsilon}, \frac{1}{\gamma}, d_{\mathcal{H}}, d_{\mathcal{G}}\right)$ iid copies of (X, Y), find classifier $f: \mathcal{X} \to \{0,1\}$ such that, with high probability,
  for all $g \in \mathcal{G}_\gamma \coloneqq \{ g \in \mathcal{G} \mid P(X \in g) \geq \gamma \}$,

$$\underbrace{P(\, f(X) \neq Y \mid X \in g\,)}_{\mathrm{err}(f \mid g)} \leq \inf_{h \in \mathcal{H}} \underbrace{P(\, h(X) \neq Y \mid X \in g\,)}_{\mathrm{err}(h \mid g)} + \epsilon$$

- Possible that no $h \in \mathcal{H}$ can satisfy this requirement on f

Application: hidden stratification

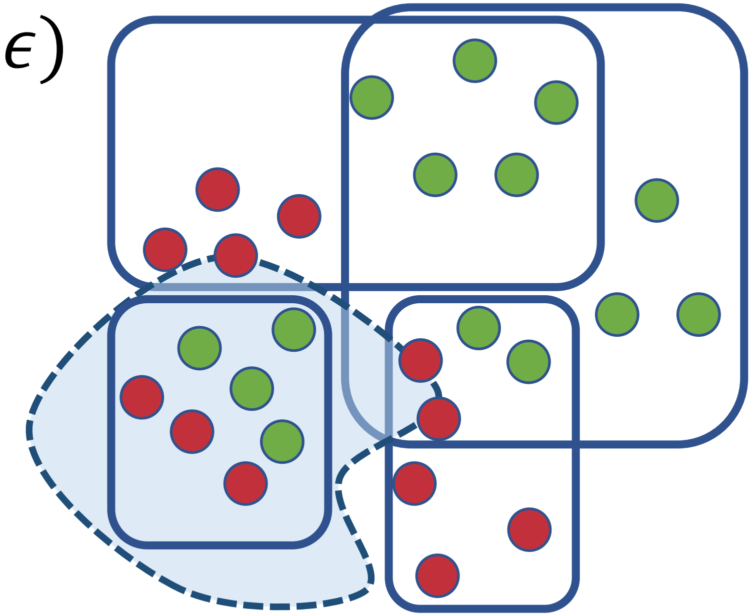
- **Multi-group agnostic learning ⇒ hidden stratification guarantee**

For every $S \subset \mathcal{X}$ that is ϵ-multiplicatively-approx.* by some $g \in \mathcal{G}_\gamma$,

$$\mathrm{err}(f \mid S) \leq \inf_{h \in \mathcal{H}} \mathrm{err}(h \mid S) + O(\epsilon)$$

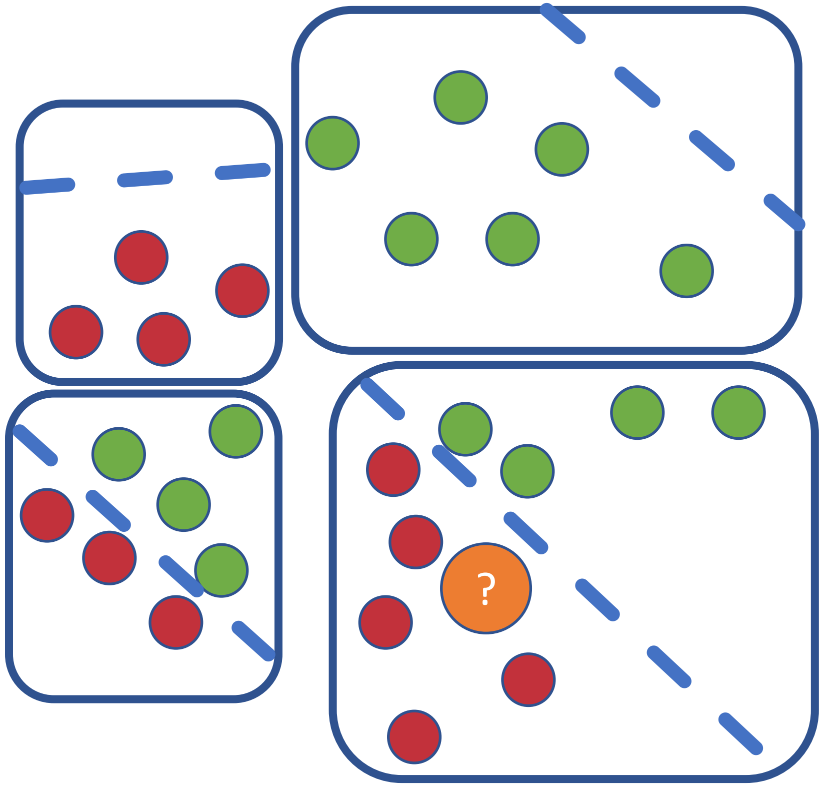
(So we'd like $\mathcal{G}$ as "rich" as possible)

*$P(g \Delta S) \leq \epsilon \min\{P(g), P(S)\}$



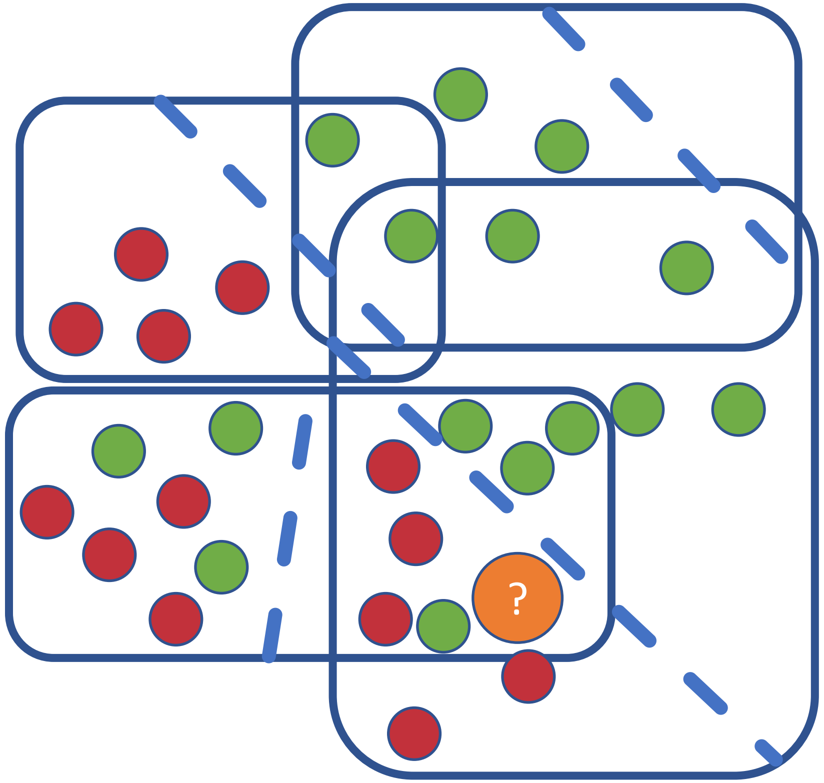
Challenges for multi-group agnostic learning

Easy case



Fit a predictor to each group

Harder case



How do we resolve disagreements among predictors?

Easy case: finitely-many disjoint groups

- **Easy case**: assume groups are disjoint
$$g \cap g' = \emptyset \quad \text{for all distinct} \quad g, g' \in \mathcal{G}$$
- **Solution**:
  - Find ERM h_g for each $g \in \mathcal{G}$
  - Return f defined by:
    On input x, find unique $g \in \mathcal{G}$ that contains x, and return $h_g(x)$
  - Sample complexity:
$$\frac{d_{\mathcal{H}} + d_{\mathcal{G}}}{\epsilon^2 \gamma}$$
- (**Also easy**: $\mathcal{G}$ is laminar family of subsets of $\mathcal{X}$)

General case: prior work

- **Rothblum and Yona (2021)**: algorithm requires sample size

$$\frac{1}{\epsilon^8 \gamma} \operatorname{polylog}\left(\frac{|\mathcal{H}| \times |\mathcal{G}|}{\epsilon}\right)$$

  - Final predictor f is functional combination of hypotheses from $\mathcal{H}$ and indicator functions of groups from $\mathcal{G}$
  - But works for other objectives beyond expected loss (e.g., calibration)
  - Based on **Outcome Indistinguishability** [Dwork, Kim, Reingold, Rothblum, Yona, 2021]

General case: our results (Tosh and H, 2022)

1. **Simple and practical algorithm**: PREPEND
   - Sample complexity: $\frac{1}{\epsilon^3 \gamma^2} \left(d_{\mathcal{H}} + d_{\mathcal{G}} \right) \log \frac{1}{\epsilon}$
2. **Near-optimal (but complicated) algorithm**: via online learning
   - Sample complexity: $\frac{1}{\epsilon^2 \gamma} \left(d_{\mathcal{H}} \log \frac{1}{\epsilon} + \log |\mathcal{G}| \right)$

1. Simple and practical algorithm

- "PREPEND" algorithm
  - Learns a decision list (of length $\leq 2/(\epsilon\gamma)$):

    "if $x \in g_1$ then return $h_1(x)$ else if $x \in g_2$ then return $h_2(x)$ else if …"

  - Sample size requirement:

$$\frac{d_{\mathcal{H}} + d_{\mathcal{G}}}{\epsilon^3 \gamma^2} \log \frac{1}{\epsilon}$$

    (somewhat worse dependence on ϵ and γ than we might've hoped for)

  - Algorithm independently found by Globus-Harris, Kearns, Roth (2022)!

PREPEND algorithm

Pick any $h \in \mathcal{H}$; define decision list f that, on input x, returns $h(x)$

While there is a group $g \in \mathcal{G}_\gamma$ and $h \in \mathcal{H}$ such that

$$\widehat{\mathrm{err}}(\, f \mid g\,) > \widehat{\mathrm{err}}(\, h \mid g\,) + \epsilon$$

Prepend "if $x \in g$ then return $h(x)$ else" to decision list f

- Decision list determines an ordering of (some subset of) $\mathcal{G}_\gamma$
- (Algorithm may select same group g in multiple loop iterations)

Analysis of PREPEND

- In iteration t, update current f_t to new f_{t+1} by prepending "if $x \in g_t$ then return $h_t(x)$ else"
- Therefore

$$\begin{aligned} \mathrm{err}(f_{t+1}) &= P(g_t)\mathrm{err}(h_t \mid g_t) + P(g_t^c)\mathrm{err}(f_t \mid g_t^c) \\ &\leq P(g_t)(\mathrm{err}(f_t \mid g_t) - \epsilon/2) + P(g_t^c)\mathrm{err}(f_t \mid g_t^c) \\ &\leq \mathrm{err}(f_t) - \gamma\epsilon/2 \end{aligned}$$

- Done within $2/(\gamma\epsilon)$ iterations

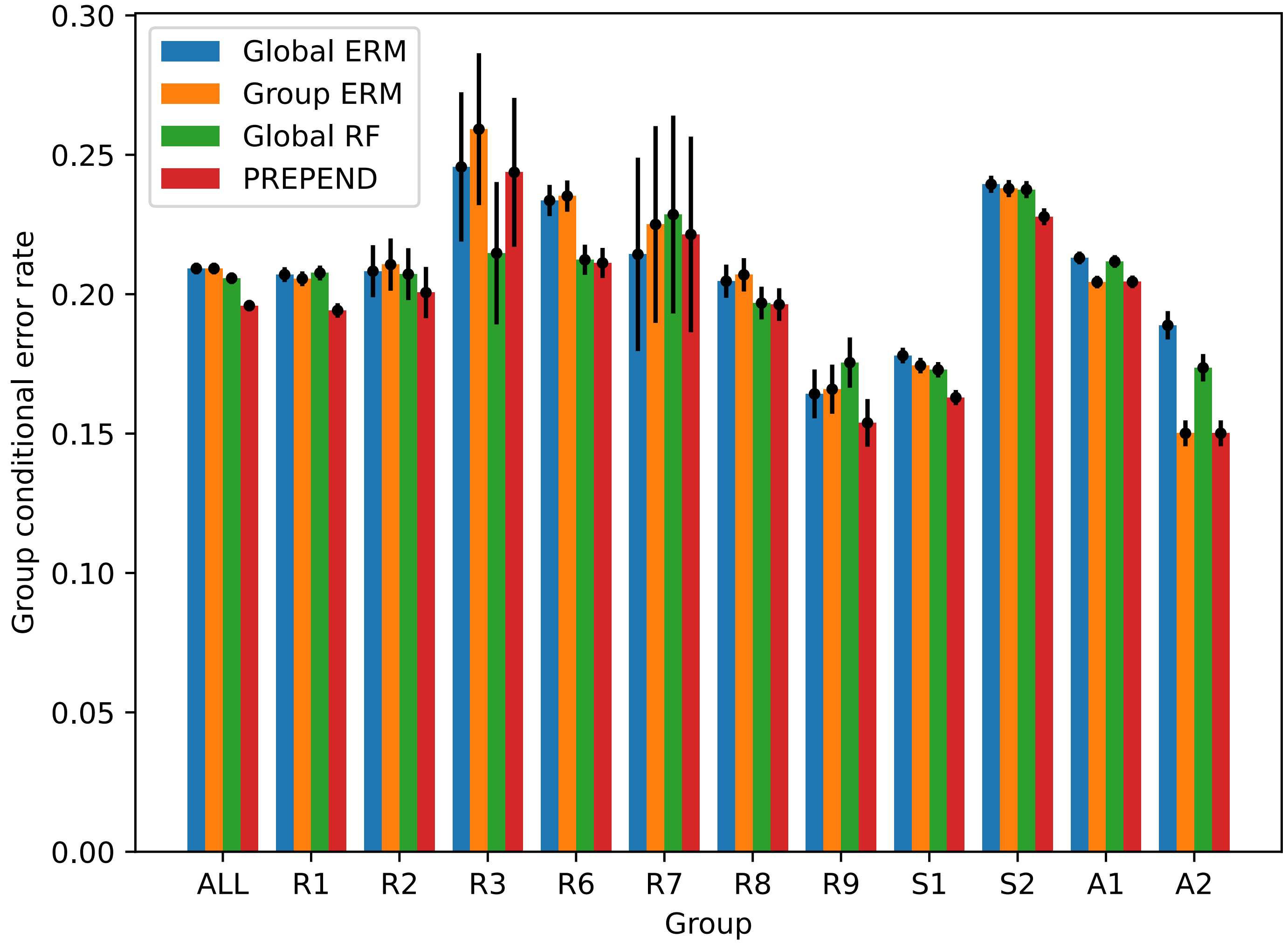
Non-iteratively learn a decision list?

- **Q: Learn a decision list with better sample complexity?**
- Cannot determine decision list just from "first-order statistics"

$$P(X \in g), \qquad \mathrm{err}(h \mid g)$$

  - Suppose $g \cap g' \neq \emptyset$
    - What should be done for $x \in g \cap g'$?
    - It may depend on $P(X \in g \cap g')$

Employment prediction in California



2016 American Community Survey

Groups:

ALL	overall population
R{1,2,3,6,7,8,9}	group by race
S{1,2}	group by sex
A{1,2}	group by age

Global ERM: logreg on all data
Group ERM: logreg on group
Global RF: random forest on all data

Data is from "Folkstable" package
(Ding, Hardt, Miller, Schmidt, 2021)

2. Near-optimal algorithm

- Algorithm based on **on-line learning**, with sample complexity

$$\frac{1}{\epsilon^2\gamma}\left(d_{\mathcal{H}}\log\frac{1}{\epsilon}+\log|\mathcal{G}|\right)$$

  - Final predictor f is stochastic ensemble of n base classifiers

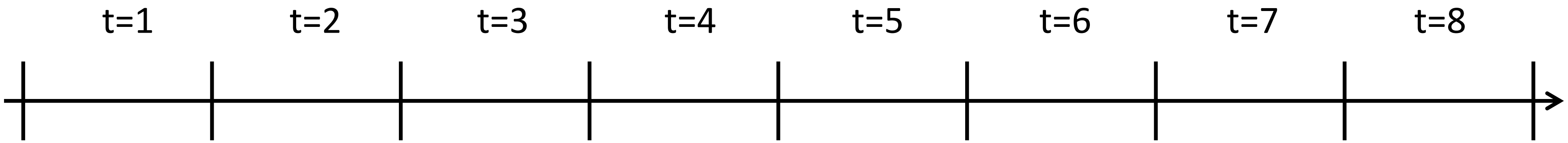
Main idea of near-optimal algorithm

- Reduction to online learning ("learning with expert advice") followed by "online-to-batch conversion"
  - Simulate instance of sequential bit prediction problem using training data
  - Use suitable online learning algorithm to solve it
  - Combine information from algorithm transcript to produce final predictor
- **Complication**: Requires "sleeping experts" variant of online learning (Freund, Schapire, Singer, Warmuth, 1997; Blum and Mansour, 2007)
- Online part is same as Blum and Lykouris (2020)

Online learning with *N* experts

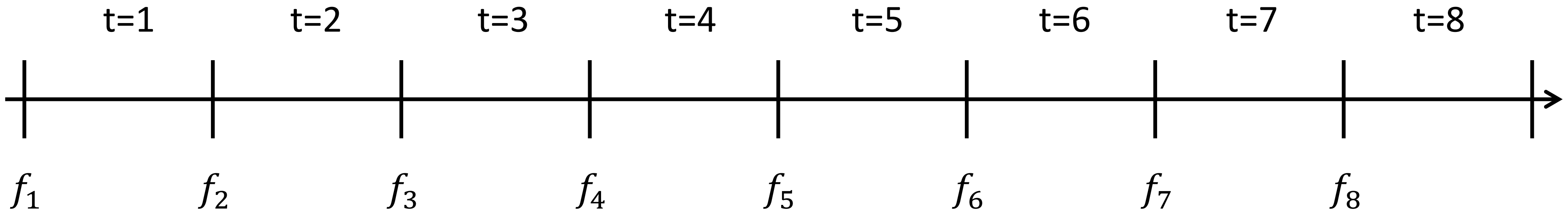
- In round $t = 1, \ldots, T$:
  - Get "context" $x_t \in \mathcal{X}$
  - Learner sees *N* experts' predictions: $\hat{y}_t^i$ for $i = 1, \ldots, N$
  - Learner makes own prediction $\hat{y}_t$, then sees true label y_t
- Regret to Expert i:
  (number of mistakes by learner) − (number of mistakes by Expert i)
- Weighted majority algorithm (Littlestone, Warmuth, 1994):

  Regret to best expert $\leq O\left(\sqrt{T \log N}\right)$



Online-to-batch conversion

Stochastic ensemble over Learner's "memory states" between rounds



Final stochastic ensemble predictor F

On input x:

- Pick t uniformly at random from $\{1, \dots, T\}$
- Return $f_t(x)$

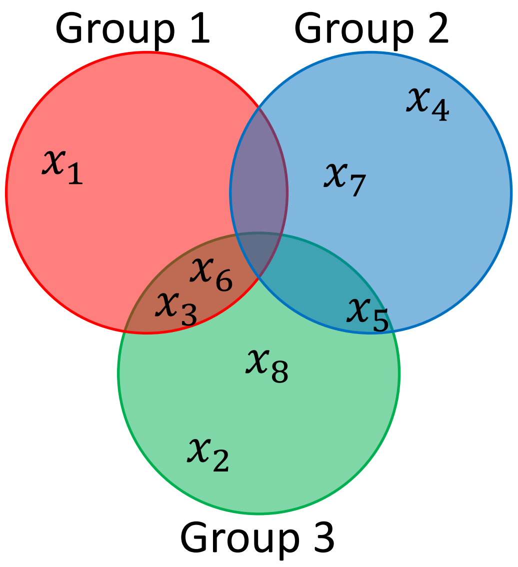
Sleeping experts variant

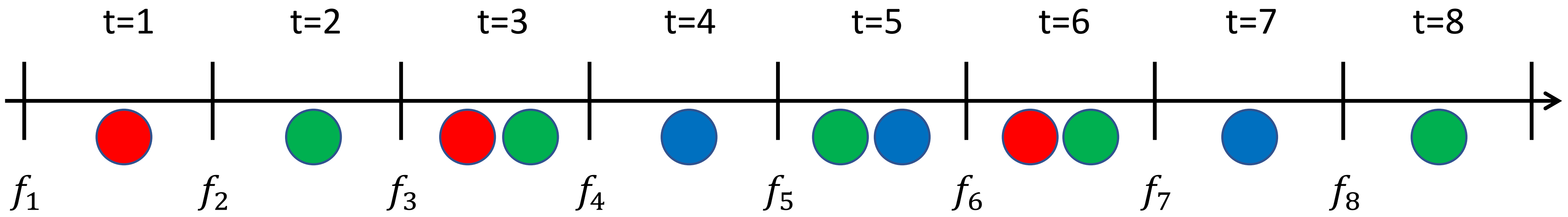
- In round $t = 1, \ldots, T$:
  - Get "context" $x_t \in \mathcal{X}$; determines subset $E_t \subseteq \{1, \ldots, N\}$ of "awake" experts
  - Learner sees "awake" experts' predictions: $\hat{y}_t^i$ for $i \in E_t$
  - Learner makes own prediction $\hat{y}_t$, then sees true label y_t
- Regret to Expert i:
  (number of mistakes by learner) − (number of mistakes by Expert i)
  … but only within the T_i rounds that Expert i is "awake"
- Variant of weighted majority (Blum and Mansour, 2007):

$$\text{Regret to expert } i \leq O\left(\sqrt{T_i \log N}\right)$$

How we use sleeping experts

- One expert per $(g, h) \in \mathcal{G} \times \mathcal{H}$ pair, so $N = |\mathcal{G}| \cdot |\mathcal{H}|$
- Consider new training example (x_t, y_t) in round t
- Expert (g, h) is "awake" in round t iff $x_t \in g$





Analysis of the simulation

$T_g = \sum \mathbb{I}\{x_t \in g\}$

- **Regret guarantees**: For all $g \in \mathcal{G}$ and $h \in \mathcal{H}$,

$$\sum \mathbb{I}\{x_t \in g\}\mathbb{I}\{\hat{y}_t \neq y_t\} - \mathbb{I}\{x_t \in g\}\mathbb{I}\{h(x_t) \neq y_t\} \leq O\left(\sqrt{T_g \log N}\right)$$

- **Concentration**: With high probability, for all $g \in \mathcal{G}$ and $h \in \mathcal{H}$,

$$\sum P(g)\mathrm{err}(f_t \mid g) - \mathbb{I}\{x_t \in g\}\mathbb{I}\{\hat{y}_t \neq y_t\} \leq O\left(\sqrt{T_g \log N}\right)$$

$$\sum \mathbb{I}\{x_t \in g\}\mathbb{I}\{h(x_t) \neq y_t\} - P(g)\mathrm{err}(h \mid g) \leq O\left(\sqrt{T_g \log N}\right)$$

Sleeping experts online-to-batch

- Online-to-batch conversion + analysis of simulation ⇒ with high probability, for all $g \in \mathcal{G}$ and $h \in \mathcal{H}$

$$\mathrm{err}(F \mid g) \leq \mathrm{err}(h \mid g) + O\left(\sqrt{\frac{\log N}{P(g)T}}\right)$$

- But:
  - F is stochastic ensemble of T predictors ☹️
  - Each individual predictor is already (roughly like) big decision list
- **Q: Better online-to-batch conversion?**
  Or "batch analogue" of sleeping experts algorithms?

Aside: bound sample size or excess error?

- **Sample complexity**: what sample size ensures excess error $\leq \epsilon$?
- **Excess error bound**: given sample size n, what is the excess error?
  - Agnostic learning (no groups), same as uniform convergence for all $h \in \mathcal{H}$:

$$O\left(\sqrt{\frac{d_{\mathcal{H}}}{n}}\right)$$

  - Uniform convergence for $h \in \mathcal{H}$ and all $g \in \mathcal{G}$ [Balsubramani et al, '19]:

$$\tilde{O}\left(\sqrt{\frac{d_{\mathcal{H}} + d_{\mathcal{G}}}{n_g}}\right)$$

  - "Near-optimal algorithm" (sorta) gets above bound in multi-group setting

Summary

- **Multi-group learning**: extension of statistical learning that is addresses many practical concerns in trustworthy AI/ML
- Tools from statistical learning theory are useful here, but **need to remix the algorithmic ideas**
  - Open problems: Simpler optimal algorithms? Polynomial-time algorithms?

Thank you!

Support: NSF CCF-1740833, IIS-1563785, and JP Morgan Faculty Award

Thanks to Kamalika Chaudhuri for introducing me to the hidden stratification problem

Laminar groups

- **Special case**: $\mathcal{G}$ is laminar (e.g., hierarchical clustering)
  - Every pair g, g' satisfies $g \cap g' = \emptyset$, $g \subset g'$, or $g \supset g'$
  - Very similar to disjoint group case
  - Sample complexity: $(d_{\mathcal{H}} + \log|\mathcal{G}|)/(\epsilon^2 \gamma)$
  - Can structure PREPEND decision list as a tree (following structure of $\mathcal{G}$)