Semi-supervised learning in insurance:

Fairness and Active learning

François HU
CREST, ENSAE - Société Générale Insurance

Supervised by :
Caroline HILLAIRET (CREST-ENSAE),
Romuald ELIE (Université Gustave Eiffel)
and **Marc JUILLARD** (Société Générale Insurance)

CIFRE PhD defense

June 15, 2022

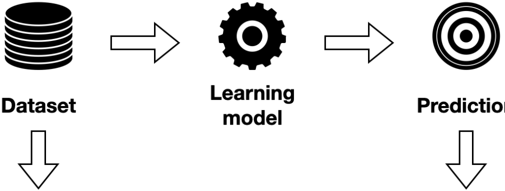
ECOLE DOCTORALE DE MATHEMATIQUES HADAMARD

CREST CENTER FOR RESEARCH IN ECONOMICS AND STATISTICS

SOCIETE GENERALE Insurance

Outline

General context: machine learning process



Given these conditions ...

Big data

+

Good-quality data

Labeled data

Unbiased data

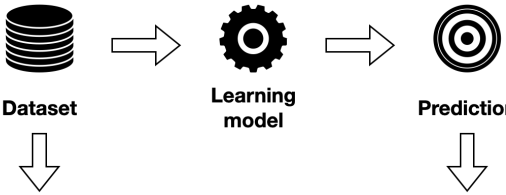
... we (may) have:

Accurate prediction

Robust prediction

Unbiased prediction

General context: machine learning process



Given these conditions ...

Big data

\+

Good-quality data

Labeled data

Unbiased data

... we (may) have:

Accurate prediction

Robust prediction

Unbiased prediction

General context: examples in insurance

Example 1: Car insurance

- Categorising photos of damaged cars
- Vehicle Telematics
- **Aim:** ML-based actuarial pricing

	Age	Sex	Claim
Erwan	31	M	0
Alice	22	F	1
Frank	67	M	0

Example 2: Regulations

- Detection of **GDPR** compliance in (text) documents

(images source: `https://www.123rf.com`)

General context: examples in insurance

Example 1: Car insurance

- Categorising photos of damaged cars
  - Label issue: need expert insurers to label
- Vehicle Telematics
  - Privacy issue: can infer some sensible features
- **Aim:** ML-based actuarial pricing
  - Fairness issue: can reflect social discriminations/prejudices

	Age	Sex	Claim
Erwan	31	M	0
Alice	22	F	1
Frank	67	M	0

Example 2: Regulations

- Detection of **GDPR** compliance in (text) documents
  - Label issue and Fairness issue:
    need legal professionals to label + imbalanced datasets

(images source: https://www.123rf.com)

- **Challenge 1**: learning with limited labeling budget
- **Challenge 2**: ensuring algorithmic fairness

Challenge 1: learning with limited labeling budget

Some **ideas** and their **limits**.

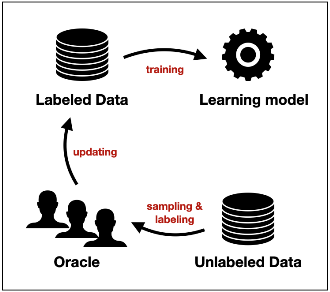


Figure 1: **Parallel labeling**

Costly and time-consuming

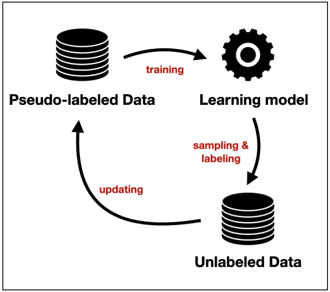


Figure 2: **Semi-supervised Learning**

Produces pseudo-labels

Active learning

- $\mathcal{H} = \{h : \underbrace{\mathcal{X}}_{\textbf{instance}} \to \underbrace{\mathcal{Y}}_{\textbf{label}}\}$ **hypothesis** space
- $\mathcal{D}^{(train)}$ **training set** and $\mathcal{D}_{\mathcal{X}}^{(pool)}$ **pool set**.

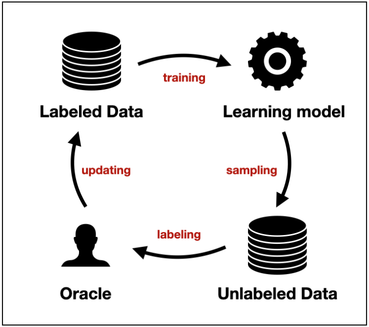


Figure 3: **Active learning in an offline scenario**

Algorithm 1 Outline of active learning (AL) process

Input: $h \in \mathcal{H}$ a base estimator, $\mathcal{D}^{(train)}$ and $\mathcal{D}_{\mathcal{X}}^{(pool)}$

Step 1. Fit h on the training set $\mathcal{D}^{(train)}$

Step 2. Given a **score** $I(x, h)$, we sample:

$$\boxed{x^* = \underset{x \in \mathcal{D}_{\mathcal{X}}^{(pool)}}{\operatorname{argmax}} \{I(x, h)\}}$$

Example: Entropy-based [Sha48]

$$I(x, h) = -\sum_{k=1}^{K} \mathbb{P}(h(x) = k|x) \log \mathbb{P}(h(x) = k|x)$$

Step 3. If y^* is its **label** then we update:

$$\mathcal{D}^{(train)} = \mathcal{D}^{(train)} \cup \{(x^*, y^*)\}$$
$$\mathcal{D}_{\mathcal{X}}^{(pool)} = \mathcal{D}_{\mathcal{X}}^{(pool)} - \{x^*\}$$

Step 4. Return to **step 1** until convergence.

Active learning: experiments on real datasets

Net Promoter Score (NPS) of Société Générale Insurance

- **About the data**: Net Promoter Score (NPS)
  - **Score**: client's score of an insurance product (score between 0 and 10);
  - **Verbatim**: explanation of the score by the client (encoded by doc2vec [LM14]);
  - **Sentiment analysis**: $\mathcal{Y} = \{\text{score} \leq T, > T\}$, parameter T to be **determined**.
- **Learning model**: XGBoost [CG16].
- **Batch-mode AL** (BMAL): at each AL iteration, sample the top b instances.

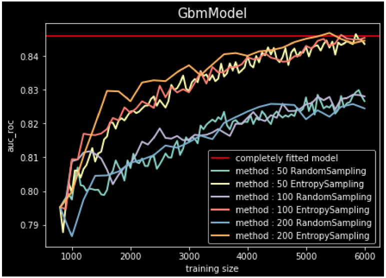


Figure 4: BMAL performance with $b = 50, 100, 200$

- Why use BMAL ? [CSC$^+$17, GSS19, WZS19]
  - i) useful for **parallel labeling**
  - ii) avoid cost of **retraining delays**.
- Calibrate b ?
  - Optimal dynamic batch-size
  - Stochastic control & dynamic programming principle

Active learning: experiments on real datasets

Net Promoter Score (NPS) of Société Générale Insurance

- **About the data**: Net Promoter Score (NPS)
  - **Score**: client's score of an insurance product (score between 0 and 10);
  - **Verbatim**: explanation of the score by the client (encoded by doc2vec [LM14]);
  - **Sentiment analysis**: $\mathcal{Y} = \{\text{score} \leq T, > T\}$, parameter T to be **determined**.
- **Learning model**: XGBoost [CG16].
- **Batch-mode AL** (BMAL): at each AL iteration, sample the top b instances.

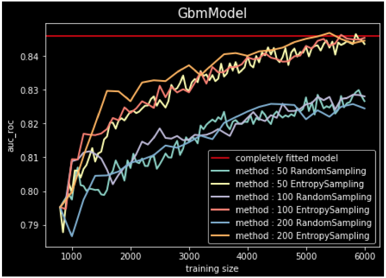
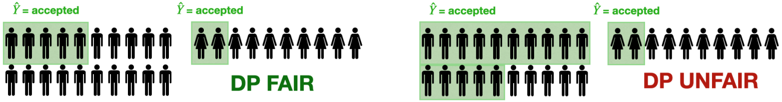


Figure 4: BMAL performance with $b = 50, 100, 200$

- Why use BMAL ? [CSC+17, GSS19, WZS19]
  i) useful for **parallel labeling**
  ii) avoid cost of **retraining delays**.
- Calibrate b ?
  - Optimal dynamic batch-size
  - Stochastic control
    & dynamic programming principle

Challenge 2: ensuring algorithmic fairness (in **group** fairness)

Data: $(\underbrace{\textit{feature}}_{X}, \underbrace{\text{sensitive attribute}}_{S}, \underbrace{\textit{label}}_{Y}) \sim \mathbb{P}$ on $\mathcal{X} \times \mathcal{S} \times [K]$. Consider $\mathcal{S} = \{-1, +1\}$.



Demographic parity [CKP09, BHN17]

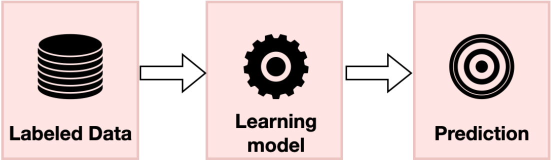
Classifier $h \in \mathcal{H}$,

$$\mathbb{P}\left(h(X,S) = k | S = 1\right) = \mathbb{P}\left(h(X,S) = k | S = -1\right) \quad \forall k \in [K] \ .$$

This **thesis** is about **demographic parity** (DP).

Challenge 2: some ideas and their limits

Where to **reduce** algorithmic bias?



Pre-processing | In-processing | Post-processing

$\boxed{\text{Modify the dataset}}$ $\boxed{\text{Time complexity}}$

Methodologies

- **Pre-processing**: reduce bias in the data before applying ML models [CKS+18, DIKL18]
- **In-processing**: reduce bias during training ML models [GCGF16, ABD+18, Nar18]
- **Post-processing**: reduce bias after fitting ML models [DHP+12, HPS16, KGZ19].

In this thesis: (semi-supervised) **post-processing** strategy.

Algorithmic fairness in multi-class classification

- Most of the work in algorithmic fairness: **binary** or **regression** tasks
- However up to our knowledge, **few works** on multi-class classification framework
- Most (modern) applications are **multi-class** tasks (e.g. risk segmentation)

Our contribution for challenge 2: Algorithmic fairness in multi-class tasks [DEHH21][1]

- Optimal fair classifiers under (exact and approximate) DP constraints
- Theoretical fairness guarantees
- Numerical efficiency of the proposed method

[1] Fairness guarantee in multi-class classification. Christophe Denis, Romuald Elie, François Hu, Mohamed Hebiri (submitted in 2022, in review).

Outline

Context: SS-BMAL and DS-BMAL procedures

Objective

Find sequence of **AL batch sizes** $(b_t)_t$ with a good trade-off between

- **maximizing** the model performance
- **reducing** the number of AL iterations

Static-size BMAL (SS-BMAL) if $b_t = b_0$ for all t ; **dynamic-size BMAL** (DS-BMAL) otherwise

Context: SS-BMAL and DS-BMAL procedures

Objective

Find sequence of **AL batch sizes** $(b_t)_t$ with a good trade-off between

- **maximizing** the model performance
- **reducing** the number of AL iterations

Static-size BMAL (SS-BMAL) if $b_t = b_0$ for all t ; **dynamic-size BMAL** (DS-BMAL) otherwise

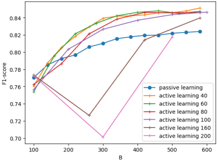


Figure 5: PL and SS-BMAL.

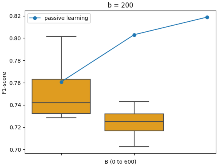


Figure 6: PL and BMAL (boxplot).

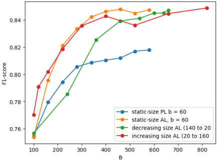


Figure 7: PL and naïve DS-BMAL.

BMAL experiments over 15 **simulations**

- **Data**: Internet Movie Db (IMDb) [MDP$^+$11] containing collection of movie reviews and binary ratings
- **Learning model**: logistic regression
- **State-of-the-art BMAL method**: Given an AL batch-size b_t, [WZS19] sample the top b_t instances in terms of **representativeness** and the **certainty** score.

DS-BMAL as a Markov Decision Process (MDP)

- **State processes**: we set the dynamics of the state processes (W_t brownian motion) as

$$\begin{cases} dQ_t = \mu(B_t, b_t) \cdot Q_t(1 - Q_t) \cdot dt + \sigma(B_t, b_t) \cdot Q_t(1 - Q_t) \cdot dW_t & \text{(performance process in [0, 1])} \\ dB_t = b_t \cdot dt & \text{(number of labeled data in } [0, B_{\text{MAX}}] \text{)} \end{cases}$$

- **Optimization problem**:

$$\boxed{V_0 = \sup_b \mathbb{E}\left[U(Q_\tau) - \int_0^\tau \mathcal{C}(b_s)ds\right]} \quad \text{(value function)}$$

  - U **utility function** models user risk-aversion concerning the model labeling-performance
  - τ **stopping time**

$$\tau = \inf\{t \geq 0 \quad | \quad B_t = B_{MAX} \text{ or } Q_t = 0 \text{ or } Q_t = 1\}.$$

  - $\mathcal{C}$ **cost** assumed to be a convex function of the batch size b.

- **Parameters:**
  - μ and σ are inferred by **numerical analysis**

$$\mu(B, b) \propto \frac{b}{B} \text{ and } \sigma(B, b) \propto \frac{b}{B}$$

  - $\mathcal{C}$ and U are **standard power functions**: $C(b) \propto b^2$ and $U(Q) = Q^p$, $p \in (0, 1)$.

DS-BMAL as a Markov Decision Process (MDP)

- **State processes**: we set the dynamics of the state processes (W_t brownian motion) as

$$\begin{cases} dQ_t = \mu(B_t, b_t) \cdot Q_t(1 - Q_t) \cdot dt + \sigma(B_t, b_t) \cdot Q_t(1 - Q_t) \cdot dW_t & \text{(performance process in [0, 1])} \\ dB_t = b_t \cdot dt & \text{(number of labeled data in } [0, B_{\text{MAX}}] \text{)} \end{cases}$$

- **Optimization problem**:

$$\boxed{V_0 = \sup_b \mathbb{E}\left[U(Q_\tau) - \int_0^\tau \mathcal{C}(b_s)ds\right]} \quad \text{(value function)}$$

  - U **utility function** models user risk-aversion concerning the model labeling-performance
  - τ **stopping time**

$$\tau = \inf\{t \geq 0 \quad | \quad B_t = B_{MAX} \text{ or } Q_t = 0 \text{ or } Q_t = 1\}.$$

  - $\mathcal{C}$ **cost** assumed to be a convex function of the batch size b.

- **Parameters:**
  - μ and σ are inferred by **numerical analysis**

$$\mu(B, b) \propto \frac{b}{B} \text{ and } \sigma(B, b) \propto \frac{b}{B}$$

  - $\mathcal{C}$ and U are **standard power functions**: $C(b) \propto b^2$ and $U(Q) = Q^p, p \in (0, 1)$.

DS-BMAL as a Markov Decision Process (MDP)

- **State processes**: we set the dynamics of the state processes (W_t brownian motion) as

$$\begin{cases} dQ_t = \mu(B_t, b_t) \cdot Q_t(1 - Q_t) \cdot dt + \sigma(B_t, b_t) \cdot Q_t(1 - Q_t) \cdot dW_t & \text{(performance process in [0, 1])} \\ dB_t = b_t \cdot dt & \text{(number of labeled data in } [0, B_{\text{MAX}}] \text{)} \end{cases}$$

- **Optimization problem**:

$$\boxed{V_0 = \sup_b \mathbb{E}\left[U(Q_\tau) - \int_0^\tau \mathcal{C}(b_s)ds\right]} \quad \text{(value function)}$$

  - U **utility function** models user risk-aversion concerning the model labeling-performance
  - τ **stopping time**

$$\tau = \inf\{t \geq 0 \quad | \quad B_t = B_{MAX} \text{ or } Q_t = 0 \text{ or } Q_t = 1\}.$$

  - $\mathcal{C}$ **cost** assumed to be a convex function of the batch size b.

- **Parameters:**
  - μ and σ are inferred by **numerical analysis**

$$\mu(B, b) \propto \frac{b}{B} \text{ and } \sigma(B, b) \propto \frac{b}{B}$$

  - $\mathcal{C}$ and U are **standard power functions**: $C(b) \propto b^2$ and $U(Q) = Q^p$, $p \in (0, 1)$.

Optimal feedback control

Dynamic programming principle

Dynamic programming principle (DPP) [Bel58] leads to

$$\text{(value function)} \quad V_t := v(Q_t, B_t) = \sup_{b_s,\ s\in[t,\tau]} \mathbb{E}\left[v(Q_{t+h}, B_{t+h}) - \int_t^{t+h} \mathcal{C}(b_s)ds \middle| \mathcal{F}_t^W \right]$$

Hamilton Jacobi Bellman (HJB) equation

HJB equation in the interior of the domain $[0, 1] \times [0, B_{MAX}]$

$$\sup_{\substack{b\geq 0\\ b\leq B_{MAX}-B}} \underbrace{\left\{ \mu(B,b)Q(1-Q)\frac{\partial v}{\partial Q}(Q,B) + b\frac{\partial v}{\partial B}(Q,B) + \frac{1}{2}\sigma(B,b)^2 Q^2(1-Q)^2 \frac{\partial^2 v}{\partial Q^2}(Q,B) - \mathcal{C}(b) \right\}}_{=\ A(Q,B,b,v)} = 0$$

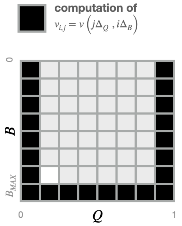
Boundary conditions:

$$\begin{aligned} v(0^+, B) &= U(0) \\ v(1^-, B) &= U(1) \\ v(Q, B_{\text{MAX}}) &= U(Q) \quad \text{for} \quad Q \in (0,1) \end{aligned}$$

Numerical resolution

Discretisation

- **Discretisation** of $[0, 1] \times [0, B_{\text{MAX}}]$ into grid of $n_Q \times n_B$ nodes and
$$v_{i,j} = v\left(j\Delta_Q, i\Delta_B\right)$$
with $\Delta_Q = \dfrac{1}{n_Q}$ and $\Delta_B = \dfrac{B_{MAX}}{n_B}$
- **Discretisation of** A: $\boxed{\widehat{A_{i,j}}(b, v_{i-1,j})}$ by finite difference



Initialisation

Howard algorithm [How60]

Compute $v_{i,j}$ by backward induction. At iteration k:

- **Step 1.** given v^k, find b^{k+1} maximizing
$$\sup_{0 \le b \le B_{MAX} - B} \left\{ \widehat{A_{i,j}}(b, v^k) \right\}$$
- **Step 2.** given b^{k+1}, compute the solution v^{k+1} s.t. $\widehat{A_{i,j}}(b^{k+1}, v^k) = 0$

Numerical results (1/2)

b^* corresponds to the optimal control

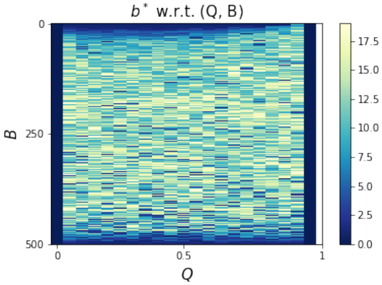


Figure 8: Heatmap of b^* w.r.t. the state process (B, Q).

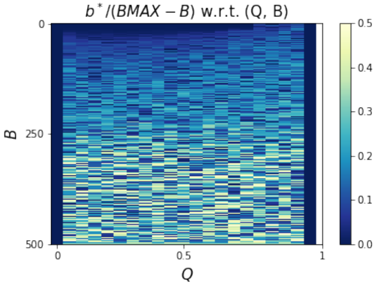


Figure 9: Heatmap of rate $b^* / (B_{MAX} - B)$ w.r.t. (B, Q).

Numerical results (2/2)

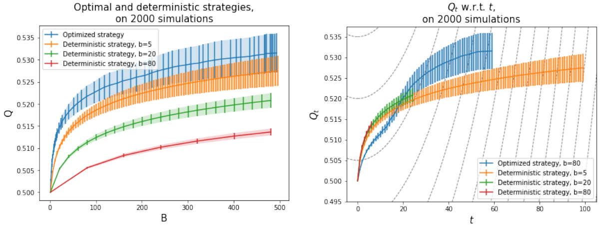


Figure 10: **Comparison between optimal and deterministic strategies** with static batch size $b \in \{5, 20, 80\}$. Initially, we set $(B_0, Q_0) = (0, 0.5)$.

Strategies | Initial performance	$Q_0 = 0.3$	$Q_0 = 0.5$
Optimized	0.577 ± 0.01	0.756 ± 0.007
Deterministic with $b = 5$	0.574 ± 0.001	0.753 ± 0.011
Deterministic with $b = 20$	0.574 ± 0.0	0.728 ± 0.006
Deterministic with $b = 80$	0.574 ± 0.0	0.727 ± 0.0

Table 1: **Value functions as a function of the control strategy.** We report the means and standard deviations over 2000 simulations. Coloured values highlight best strategy.

Results

Optimised strategy **outperforms** deterministic strategies in terms of:

- **model quality** by labelling less at the beginning of the process
- **reduction of (retraining) delays**. Indeed this dynamic strategy considerably reduces the number of iterations.

Outline

Context: **argmax-fairness**

Notations

- **Data:** $(\underbrace{\textit{feature}}_{X}, \underbrace{\text{sensitive attribute}}_{S}, \underbrace{\textit{label}}_{Y}) \sim \mathbb{P}$ on $\mathcal{X} \times \mathcal{S} \times [K]$
- **Distribution** of S: $(\pi_s)_{s \in \mathcal{S}}$
- **Misclassification risk** for a classifier $g : \mathcal{X} \times \{-1, 1\} \to [K]$:

$$\mathcal{R}(g) := \mathbb{P}\left(g(X, S) \neq Y\right)$$

- **Scores**: for $k \in [K]$, we denote $p_k(X, S) := \mathbb{P}\left(Y = k | X, S\right)$
- **Bayes classifier** minimizes the misclassification risk:

$$g^*(x, s) \in \arg\max_k p_k(x, s) \ , \quad \text{for all } (x, s) \in \mathcal{X} \times \mathcal{S}$$

Objective:

- **Minimizing the risk**: $g^* \in \underset{g}{\operatorname{argmin}}\ \mathcal{R}(g)$
- **Enforcing fairness**: $\underbrace{g^*(X, S) \perp\!\!\!\perp S}_{\text{Demographic Parity (DP)}}$ (denoted $g^* \in \mathcal{G}_{\text{fair}}$)

Exact fairness in multi-class classification

Definition (Exact Demographic Parity)

Classifier g is **exactly fair** if for each $k \in [K]$,

$$\mathbb{P}\left(g(X,S) = k | S = 1\right) = \mathbb{P}\left(g(X,S) = k | S = -1\right)$$

Equivalently, if we define the following **unfairness measure**

$$\mathcal{U}(g) := \max_{k \in [K]} \left|\mathbb{P}\left(g(X,S) = k | S = 1\right) - \mathbb{P}\left(g(X,S) = k | S = -1\right)\right|$$

Classifier g is **exactly fair** i.i.f. $\mathcal{U}(g) = 0$.

Optimal exactly fair classifier g^*_{fair} solves

$$\min_{g \in \mathcal{G}_{\text{fair}}} \mathcal{R}(g) \qquad \text{(equality constraint !)}$$

Let us consider its **Lagrangian** and introduce for $\lambda = (\lambda_1, \ldots, \lambda_K) \in \mathbb{R}^K$,

$$\mathcal{R}_\lambda(g) := \mathcal{R}(g) + \sum_{k=1}^{K} \lambda_k \left[\mathbb{P}\left(g(X,S) = k | S = 1\right) - \mathbb{P}\left(g(X,S) = k | S = -1\right)\right]$$

We call this measure **fair-risk**.

Optimal prediction under exact-DP

Continuity assumption

$t \mapsto \mathbb{P}(p_k(X,S) - p_j(X,S) \leq t | S = s)$ considered **continuous**, for any $k, j \in [K]$ and $s \in \mathcal{S}$.

Proposition

Under continuity assumption, we define

$$\lambda^* \in \arg\min_{\lambda \in \mathbb{R}^K} \sum_{s \in \mathcal{S}} \mathbb{E}_{X|S=s}\left[\max_k \left(\pi_s p_k(X,s) - s\lambda_k\right)\right].$$

Then, $g^*_{\text{fair}} \in \arg\min_{g \in \mathcal{G}_{\text{fair}}} \mathcal{R}(g)$ i.f.f $g^*_{\text{fair}} \in \arg\min_{g \in \mathcal{G}} \mathcal{R}_{\lambda^*}(g)$.

Corollary

Under continuity assumption, an optimal **exactly fair** classifier is characterized by

$$\boxed{g^*_{\text{fair}}(x,s) \in \arg\max_{k \in [K]} \left(\pi_s p_k(x,s) - s\lambda^*_k\right)}, \quad (x,s) \in \mathcal{X} \times \mathcal{S}.$$

Semi-supervised post-processing estimator

Theoretical fair solution:

$$\lambda^* \in \arg\min_{\lambda \in \mathbb{R}^K} \sum_{s \in \mathcal{S}} \mathbb{E}_{X|S=s}\left[\max_k \left(\pi_s p_k(X,s) - s\lambda_k\right)\right] \ .$$

$$g^*_{\text{fair}}(x,s) \in \arg\max_k \left(\pi_s p_k(x,s) - s\lambda^*_k\right) , \quad (x,s) \in \mathcal{X} \times \mathcal{S} \ .$$

Empirical fair solution

- **Plug-in**: Estimation based on independent samples
  - **Labeled data:** $\mathcal{D}_n = (X_i, S_i, Y_i)_{i=1,\dots,n}$. ζ_k uniform perturbation on $[0,u]$ train estimators $(\hat{p}_k)_k$. Continuity assumption satisfied if $\bar{p}_k(X,S,\zeta_k) := \hat{p}_k(X,S) + \zeta_k$
  - **Unlabeled data:** for all $s \in S$, $X_1^s, \dots, X_{N_s}^s \overset{\text{iid}}{\sim} \mathbb{P}_{X|S=s}$ empirical frequencies $(\hat{\pi}_s)_{s\in\mathcal{S}}$ as estimates of $(\pi_s)_{s\in\mathcal{S}}$ (recall that $\pi_s = \mathbb{P}(S=s)$)
- **Fair estimator:**

$$\hat{\lambda} \in \arg\min_{\lambda} \sum_{s\in\mathcal{S}} \frac{1}{N_s} \sum_{i=1}^{N_s} \left[\max_{k\in[K]} \left(\hat{\pi}_s \bar{p}_k(X_i^s, s, \zeta_{k,i}^s) - s\lambda_k\right)\right]$$

$$\boxed{\hat{g}(x,s) = \arg\max_{k\in[K]} \left(\hat{\pi}_s \bar{p}_k(x,s,\zeta_k) - s\hat{\lambda}_k\right)}$$

Statistical guarantees

Theorem

- **Universal exact fairness guarantee.** There exists constant $C > 0$ depending only on K and $\min_{s\in\mathcal{S}} \pi_s$, s.t., for any estimators $\hat{p}_k$,

$$\mathbb{E}\left[\mathcal{U}(\hat{g})\right] \leq \frac{C}{\sqrt{N}}$$

- **Consistency.** Under continuity assumption

$$\mathbb{E}[\mathcal{R}_{\lambda^*}(\hat{g})] - \mathcal{R}_{\lambda^*}(g^*_{\text{fair}}) \leq C\left(\mathbb{E}\left[\|\hat{p} - p\|_1\right] + \sum_{s\in\mathcal{S}} \mathbb{E}\left[|\hat{\pi}_s - \pi_s|\right] + \mathbb{E}\left[\mathcal{U}(\hat{g})\right] + u\right)$$

with $\|\hat{p} - p\|_1 = \sum_{k\in[K]} |\hat{p}_k(X, S) - p_k(X, S)|$ (L_1-norm b/w estimator and cond.prob.)

Corollary

If $\mathbb{E}\left[\|\hat{p} - p\|_1\right] \to 0$ and $u = u_n \to 0$ when $n \to \infty$, we have

$$|\mathbb{E}[\mathcal{R}(\hat{g})] - \mathcal{R}(g^*_{\text{fair}})| \to 0, \qquad \text{as} \quad n, N \to \infty$$

Under suitable conditions, we have $\mathbb{E}[\mathcal{R}(\hat{g})] \to \mathcal{R}(g^*_{\text{fair}})$ and $\mathbb{E}[\mathcal{U}(\hat{g})] \to 0$ as $n, N \to \infty$.

Approximate fair multi-class classification

Definition (ε-Demographic Parity)

Given $\varepsilon > 0$, classifier g is **ε-fair** if for each $k \in [K]$,

$$|\mathbb{P}(g(X,S)=k|S=1) - \mathbb{P}(g(X,S)=k|S=-1)| \leq \varepsilon \ .$$

Equivalently, (using **unfairness measure**) classifier g is **ε-fair** i.f.f. $\mathcal{U}(g) \leq \varepsilon$.

Optimal exactly fair classifier $g^*_{\varepsilon-fair}$ solves

$$\min_{g \in \mathcal{G}_{\varepsilon-\text{fair}}} \mathcal{R}(g) \qquad \text{(inequality constraint !)}$$

Let us consider its **Lagrangian** and introduce for $\lambda^{(1)} = (\lambda^{(1)}_1, \ldots, \lambda^{(1)}_K) \in \mathbb{R}^K_+$ and $\lambda^{(2)} = (\lambda^{(2)}_1, \ldots, \lambda^{(2)}_K) \in \mathbb{R}^K_+$,

$$\begin{aligned}\mathcal{R}_{\lambda^{(1)},\lambda^{(2)}}(g) := \mathcal{R}(g) &+ \sum_{k=1}^{K} \lambda^{(1)}_k [\mathbb{P}(g(X,S)=k|S=1) - \mathbb{P}(g(X,S)=k|S=-1) - \varepsilon] \\ &+ \sum_{k=1}^{K} \lambda^{(2)}_k [\mathbb{P}(g(X,S)=k|S=-1) - \mathbb{P}(g(X,S)=k|S=1) - \varepsilon] \ .\end{aligned}$$

We call this measure **ε-fair-risk**.

Optimal prediction under ε-DP

Proposition

Let $H : \mathbb{R}_+^{2K} \to \mathbb{R}$ be the function

$$H(\lambda^{(1)}, \lambda^{(2)}) \;=\; \sum_{s\in\mathcal{S}} \mathbb{E}_{X|S=s}\left[\max_k \left(\pi_s p_k(X,s) - s(\lambda_k^{(1)} - \lambda_k^{(2)})\right)\right] + \varepsilon \sum_{k=1}^{K} (\lambda_k^{(1)} + \lambda_k^{(2)}) \;.$$

- Under continuity assumption, we define $\lambda^{*(1)}, \lambda^{*(2)} \in \mathbb{R}_+^{2K}$ by

$$(\lambda^{*(1)}, \lambda^{*(2)}) \in \arg \min_{(\lambda^{(1)},\lambda^{(2)})\in\mathbb{R}_+^{2K}} H(\lambda^{(1)}, \lambda^{(2)}) \;.$$

  Then, $g^*_{\varepsilon-\text{fair}} \in \arg\min_{g\in\mathcal{G}_{\varepsilon-\text{fair}}} \mathcal{R}(g)$ iff $g^*_{\varepsilon-\text{fair}} \in \arg\min_{g\in\mathcal{G}} \mathcal{R}_{\lambda^{*(1)},\lambda^{*(2)}}(g)$.

- In addition, $\boxed{g^*_{\varepsilon-\text{fair}}(x,s) = \arg\max_{k\in[K]} \left(\pi_s p_k(x,s) - s(\lambda_k^{*(1)} - \lambda_k^{*(2)})\right)} \quad \forall (x,s) \in \mathcal{X}\times\mathcal{S} \;.$

Same **methodology** and **extension** of exact-fairness:

i) **Plug-in** ε **fair classifier** in a **semi-supervised** manner with **randomization** trick.

ii) **Distribution-free** ε**-fairness.** Estimator $\hat{g}_\varepsilon$: right fairness level $|\mathbb{E}[\mathcal{U}(\hat{g}_\varepsilon)] - \varepsilon| \leq \frac{C}{\sqrt{N}}$.

iii) **Consistency.** If estimator L_1-norm consistent then $\mathbb{E}\left[\mathcal{R}(\hat{g}_\varepsilon)\right] \to \mathcal{R}(g^*_{\varepsilon-\text{fair}})$ as $n, N \to \infty$.

Numerical evaluation: real data

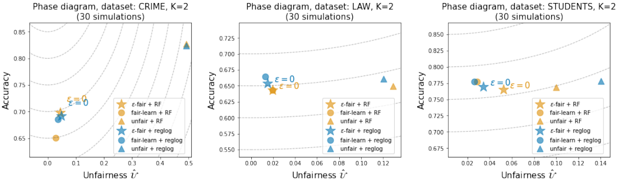


Figure 11: (Accuracy, Unfairness) phase diagrams in **binary** case

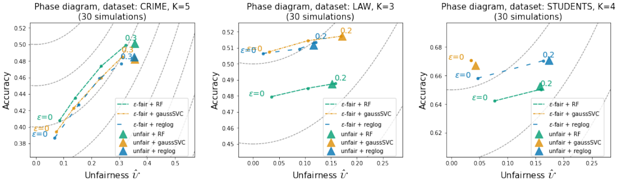


Figure 12: (Accuracy, Unfairness) phase diagrams in **multi-class** case.

Models & Datasets

- **Datasets**: CRIME, LAW and STUDENTS.
- **models**: logistic regression (reglog) and RF

Results

1. **Competitive unfairness.** Exactly-fair algorithm achieves similar performance as the state-of-the-art **fair-learn**[a]
2. **Competitive accuracy.** Achieve a better accuracy on CRIME when we consider RF (.70 vs .65)
3. **Time complexity.** Baseline running time more higher than with our method.

[a] https://fairlearn.org/

Outline

Conclusion

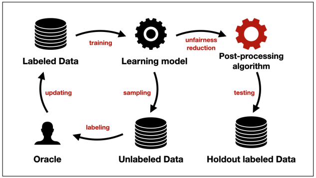
This thesis propose **some methods** for both challenges

1. learning with limited labeling budget
2. ensuring algorithmic fairness

motivated by **insurance applications**.

Challenge 1-2: fair active learning [EHHJ21]

- Accuracy analysis
- Robustness analysis
- Fairness analysis



Perspective: further works on these challenges

- **Challenge 1: Generalizing** optimal AL batch-size
- **Challenge 2:** Multi-label classification with **score-fair** (instead of **exactly-fair**).

$$R_2(g) = \mathbb{E}\left[\sum_{k=1}^{K}\left(\mathbb{1}_{Y=k} - g_k(X,S)\right)^2\right] \text{ VS } R(g) = \mathbb{P}(g(X,S) \neq Y)$$

References I

[ABD+18] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *International Conference on Machine Learning*, pages 60–69. PMLR, 2018.

[Bel58] Richard Bellman. Dynamic programming and stochastic control processes. *Information and control*, 1(3):228–239, 1958.

[BHN17] Solon Barocas, Moritz Hardt, and Arvind Narayanan. Fairness in machine learning. *Nips tutorial*, 1:2, 2017.

[CDH+20] E. Chzhen, C. Denis, M. Hebiri, L. Oneto, and M. Pontil. Fair regression with wasserstein barycenters. In *Advances in Neural Information Processing Systems*, 2020.

[CG16] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[CKP09] T. Calders, F. Kamiran, and M. Pechenizkiy. Building classifiers with independency constraints. In *IEEE international conference on Data mining*, 2009.

References II

[CKS+18] Elisa Celis, Vijay Keswani, Damian Straszak, Amit Deshpande, Tarun Kathuria, and Nisheeth Vishnoi. Fair and diverse dpp-based data summarization. In *International Conference on Machine Learning*, pages 716–725. PMLR, 2018.

[CSC+17] Thiago NC Cardoso, Rodrigo M Silva, Sérgio Canuto, Mirella M Moro, and Marcos A Gonçalves. Ranked batch-mode active learning. *Information Sciences*, 379:313–337, 2017.

[DEHH21] C. Denis, R. Elie, M. Hebiri, and F. Hu. Fairness guarantee in multi-class classification, 2021.

[DHP+12] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.

[DIKL18] C. Dwork, N. Immorlica, A. T. Kalai, and M. D. M. Leiserson. Decoupled classifiers for group-fair and efficient machine learning. In *Conference on Fairness, Accountability and Transparency*, 2018.

[EHHJ21] Romuald Elie, Caroline Hillairet, François Hu, and Marc Juillard. An overview of active learning methods for insurance with fairness appreciation. *arXiv preprint arXiv:2112.09466*, 2021.

References III

[GCGF16] Gabriel Goh, Andrew Cotter, Maya Gupta, and Michael P Friedlander. Satisfying real-world goals with dataset constraints. *Advances in Neural Information Processing Systems*, 29, 2016.

[GSS19] Daniel Gissin and Shai Shalev-Shwartz. Discriminative active learning. *arXiv preprint arXiv:1907.06347*, 2019.

[How60] Ronald A Howard. Dynamic programming and markov processes. 1960.

[HPS16] M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In *Neural Information Processing Systems*, 2016.

[KGZ19] Michael P Kim, Amirata Ghorbani, and James Zou. Multiaccuracy: Black-box post-processing for fairness in classification. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 247–254, 2019.

[LM14] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR, 2014.

References IV

[MDP+11] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[Nar18] Harikrishna Narasimhan. Learning with complex loss functions and constraints. In *International Conference on Artificial Intelligence and Statistics*, pages 1646–1654. PMLR, 2018.

[Sha48] C.E. Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[WZS19] Hanmo Wang, Runwu Zhou, and Yi-Dong Shen. Bounding uncertainty for active batch selection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 5240–5247, 2019.

Outline

Active learning: comparison with passive learning

Empirical risk minimization (ERM)

(Misclassification) Risk: $R(h) = \mathbb{P}(h(x) \neq y)$;

Empirical risk: $\hat{R}_n(h) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left(h(x_i) \neq y_i\right)$;

ERM: $\hat{h} = \underset{h \in \mathcal{H}}{\operatorname{argmin}}\ \hat{R}_n(h)$.

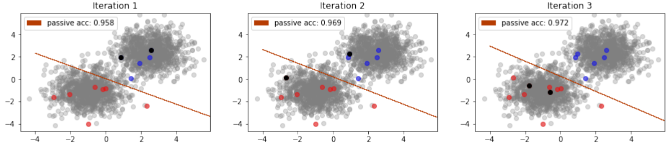


Figure 13: **Passive learning** (random sampling)

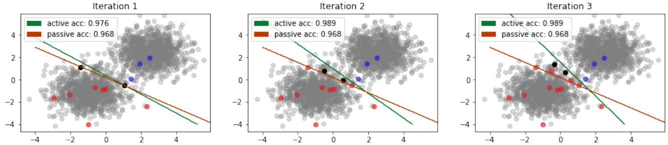


Figure 14: **Active learning** (entropy-based sampling, Shannon entropy [Sha48]).

Outline

Numerical evaluation: synthetic data

Synthetic data: Gaussian mixture model with Bernoulli contamination

Features & Sensitive feature. Consider $c^k \sim \mathcal{U}_d(-1, 1)$, and $\mu_1^k, \ldots, \mu_m^k \sim \mathcal{N}_d(0, I_d)$,

$$
\begin{aligned}
(X|Y = k) \quad &\sim \quad \frac{1}{m}\sum_{i=1}^{m} \mathcal{N}_d(c^k + \mu_i^k, I_d), \quad \text{for } k \in [K], \\
(S|Y = k) \quad &\sim \quad 2 \cdot \mathcal{B}(p) - 1, \quad \text{if} \quad k \leq \lfloor K/2 \rfloor, \\
(S|Y = k) \quad &\sim \quad 2 \cdot \mathcal{B}(1-p) - 1, \quad \text{if} \quad k > \lfloor K/2 \rfloor.
\end{aligned}
$$

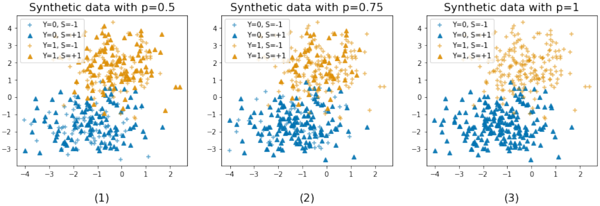


Figure 15: Example of synthetic data in binary case. We set $d = 2$ and $m = 1$. *(1)* $p = 0.5$ (*e.g.* **no unfairness**) *(2)* $p = 0.75$ (*e.g.* **unfair dataset**) *(3)* $p = 1$ (*e.g.* **highly unfair dataset**)

Numerical evaluation: synthetic data

⚠ Fairness of g is measured via the empirical version of the unfairness measure $\mathcal{U}(g)$.

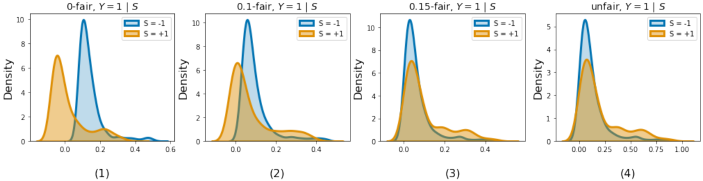


Figure 16: Empirical distribution of **Random Forest** (RF) score functions for the class $Y = 1$, conditional to the sensitive feature $S = \pm 1$. *(1)-(3)* ϵ-fairness with $\epsilon \in \{0, 0.1, 0.15\}$, *(4)* unfair.

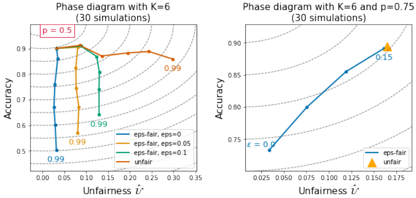


Figure 17: **RF** (Accuracy, Unfairness) phase diagrams for synthetic datasets w.r.t. *Left* the level of bias p; *Right* the accuracy-fairness trade-off parameter ε. Best trade-off at top-left corner.

Outline

Alternative method: **score-fairness**

Definition (score-fair in demographic parity)

$f : \mathcal{X} \times \{-1, 1\} \mapsto \mathbb{R}^K$ is **score-fair** in DP if each coordinate f_k of f is DP fair,

$$\mathbb{P}(f_k(X,S) \leq t \mid S=-1) = \mathbb{P}(f_k(X,S) \leq t \mid S=1) \quad \forall k, t \in [K] \times \mathbb{R} \ .$$

Consider minimization task

$$f^*_{score-fair} \in \text{argmin} \ \{R_2(f) : f \text{ is } \textbf{score-fair}\} \ .$$

where $R_2(f) = \mathbb{E}\left[\sum_{k=1}^K (\mathbb{1}_{Y=k} - f_k(X,S))^2\right]$.

Theorem: Optimal prediction under DP score-fairness (L_2-risk based) [CDH+20]

[CDH+20] identifies the distribution of score-fair classifier $f^*_{score-fair}$ as solutions of a Wasserstein barycenter problem. In particular, $f^*_{score-fair} = (f^*_{sf,1}, \ldots, f^*_{sf,K}) \in \mathbb{R}^K$ with

$$f^*_{sf,k}(x,s) = (\pi_{-s} \cdot \underbrace{Q_{f^*_k|-s})}_{\textbf{quantile function}} \circ \underbrace{F_{f^*_k|s}}_{\textbf{CDF}}(f^*_k(x,s))$$

Plug-in estimator by estimating for all $s \in \mathcal{S}$, $\boxed{\pi_s, F_{f^*_k|s} \text{ and } Q_{f^*_k|s}}$.

Numerical evaluation on synthetic data (1/2) : argmax-fair VS score-fair

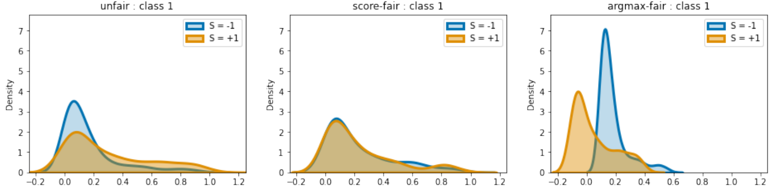


Figure 18: Empirical distribution of the score functions for the class $Y = 1$, conditional to $S = \pm 1$.

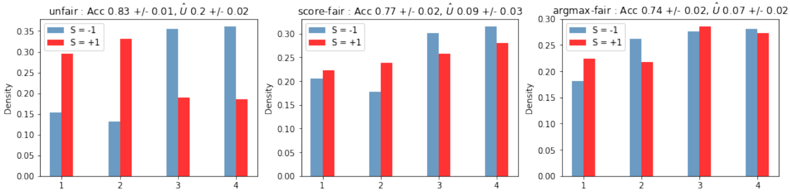


Figure 19: Emp. distribution of **unfair**(left), **score-fair**(middle) and **argmax-fair**(right) classifiers conditional to $S = \pm 1$

Numerical evaluation on synthetic data (2/2) : argmax-fair VS score-fair

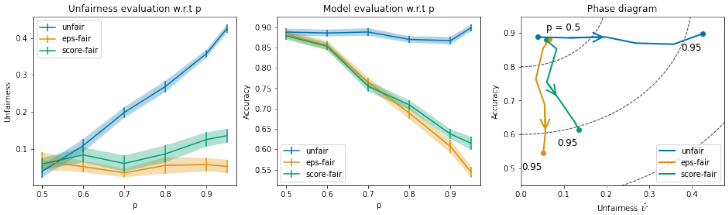


Figure 20: Performance (accuracy, fairness) for **unfair**, **argmax-fair**, and **score-fair** classifiers (30 simulations).

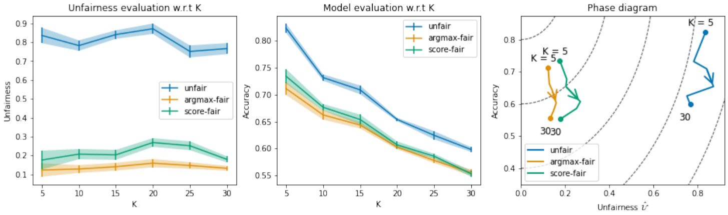


Figure 21: Performance (accuracy, fairness) for **unfair**, **argmax-fair**, and **score-fair** classifiers (30 simulations).

Numerical evaluation: real data

	CRIME, K = 7		LAW, K = 4	
	Accuracy	Unfairness (sum)	Accuracy	Unfairness (sum)
reglog + unfair	0.34 ± 0.02	1.12 ± 0.07	0.43 ± 0.01	0.89 ± 0.05
reglog + score-fair (baseline)	0.33 ± 0.01	0.78 ± 0.09	0.42 ± 0.01	0.09 ± 0.02
reglog + argmax-fair	0.28 ± 0.01	0.26 ± 0.07	0.42 ± 0.01	0.05 ± 0.02
linearSVC + unfair	0.36 ± 0.02	1.12 ± 0.07	0.43 ± 0.01	0.97 ± 0.07
linearSVC + score-fair (baseline)	0.31 ± 0.02	0.88 ± 0.05	0.42 ± 0.01	0.1 ± 0.03
linearSVC + argmax-fair	0.29 ± 0.02	0.25 ± 0.08	0.42 ± 0.01	0.04 ± 0.02
GaussSVC + unfair	0.36 ± 0.02	1.4 ± 0.13	0.43 ± 0.01	1.04 ± 0.04
GaussSVC + score-fair (baseline)	0.35 ± 0.02	1.02 ± 0.07	0.42 ± 0.01	0.16 ± 0.04
GaussSVC + argmax-fair	0.3 ± 0.02	0.22 ± 0.05	0.42 ± 0.01	0.10 ± 0.03
RF + unfair	0.37 ± 0.02	1.02 ± 0.04	0.40 ± 0.01	0.65 ± 0.04
RF + score-fair (baseline)	0.34 ± 0.02	0.67 ± 0.06	0.39 ± 0.01	0.11 ± 0.05
RF + argmax-fair	0.3 ± 0.02	0.33 ± 0.11	0.39 ± 0.01	0.07 ± 0.02

Table 2: (accuracy & unfairness) in **multi-class** tasks over 30 repetitions. Colored values highlight fairness.

Results

- **argmax-fair** procedure outperforms **unfair** and **score-fair** approaches.
- Provide **optimal fair classification** rule under DP constraint.
- Our approach can be applied on top of any probabilistic base estimator.