Unleashing Linear Optimizers for Group-Fair Learning and Optimization

Daniel Alabi[1], Nicole Immorlica[2], Adam Tauman Kalai[2]

Harvard University[1], Microsoft Research[2]

Group Fairness

Intuition: No group of individuals should be treated unfairly

X : set of individuals

Y : set of outcomes

Split X into K groups $X_1, \dots, X_K$ and define "fair function"

$f(X_1, \dots, X_K)$ to minimize unfairness between outcomes of groups.

Two Example "Group-Fair" Definitions

1. Demographic Parity:

Select (randomized) hypothesis h such that $\forall k \in [K], \hat{y} \in \{0,1\}$

$\mathbb{P}[h(x) = \hat{y} | x \in X_k] \approx \mathbb{P}[h(x) = \hat{y}]$

2. Equalized Odds:

Select (randomized) hypothesis h such that $\forall k \in [K], \hat{y}, y \in \{0,1\}$

$\mathbb{P}[h(x) = \hat{y} | x \in X_k, Y = y] \approx \mathbb{P}[h(x) = \hat{y} \,|Y = y]$

The goal is to minimize "group-fair" functions using "black-box" approximate linear optimizers.

Previous Work on Group-Fair Reductions

ABDLW (FATML '17, ICML '18):

Users can specify convex constraints that can be folded into the objective

(e.g. demographic parity, equalized odds can be specified as linear constraints)

NRSA (ICML '15):

Can handle convex functions and ratio-of-linear functions (e.g. F1, G-mean).

DIKL (FAT* '18):

Can only handle non-decreasing functions and needs access to protected attribute during classification stage.

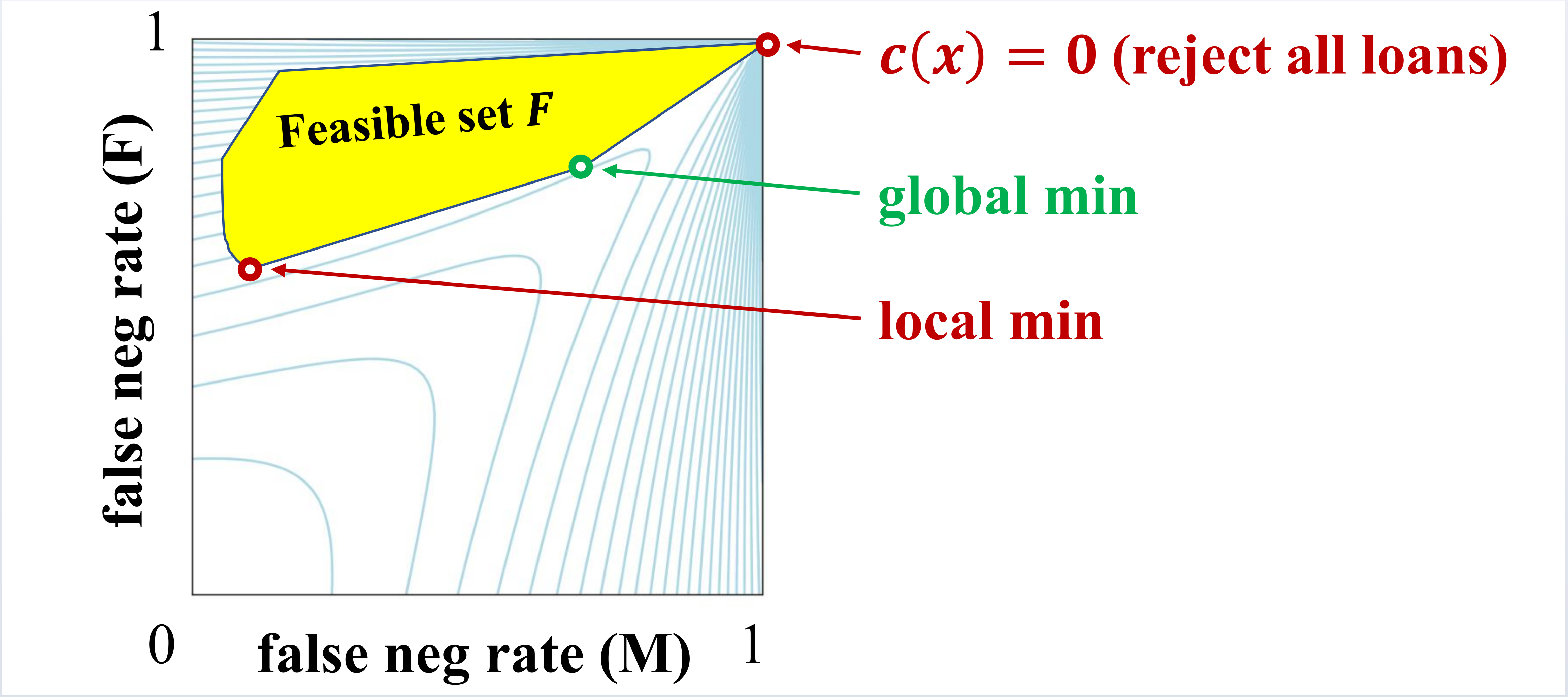
Consider the following non-convex objective that combines misclassification rate with a penalty for M-F disparity amongst loan approvals.

$$\min \mathbb{P}[c(x) \neq y] + (\mathbb{P}[x \in X_F | c(x) = 1] - \mathbb{P}[x \in X_M | c(x) = 1])^2$$

over $c \in \mathcal{C}$

where $X = X_F \cup X_M$

For simplicity, assume that $\mathbb{P}[c(x) = 1 \wedge y = 0] = 0$ (no false positives)



Algorithm 1 GroupOpt: Minimizing group-loss f using linear optimizer

Input: accuracy $\epsilon > 0$, $f : [0,1]^K \to \mathbb{R}$, loss assessor ℓ_τ, (nonnegative) linear optimizer M_τ
Output: $c \in \mathcal{C}$

Let $\beta = \frac{\epsilon}{5}, q = \frac{\beta}{\sqrt{K}}, \tau = \frac{\beta^2}{\sqrt{K}}, T = \frac{K}{\beta^2} \ln \frac{K}{\beta^2}$.

Create grid $G = \{0, q, 2q, 3q, \dots, \lfloor 1/q \rfloor q\}^K \subseteq [0,1]^K$.

Check if f is nondecreasing coordinatewise on G. If so, let $N = 1$ else $N = 0$.

Sort points in G by $f(r)$ in increasing order

```
for r in G do
    c_1 = M_τ(0)    // any initial choice
    for t = 1 to T do
        l̂_t = max((1/t)(ℓ_τ(c_1) + ··· + ℓ_τ(c_t)), N r)
        c_{t+1} = M_τ(l̂_t − r)
    end
    if ‖l̂_T − r‖ ≤ 3β then
        return c = UniformDist({c_1, c_2, ..., c_T})    // uniform probability distribution
    end
end
```

New Work: *Any Continuous* Objective of Group Losses

$\min f(\ell_1(c), \dots, \ell_K(c))$ to within ϵ over $c \in \mathcal{C}$ where $\ell_k(c)$ is the loss incurred by group $k \in [K]$ for classifier $c \in \mathcal{C}$ using:

1. (Approximate) Loss Assessor: $\ell_\tau : \mathcal{C} \to [0,1]^K$ such that
$$\|\ell_\tau(c) - \ell(c)\| \leq \tau$$
2. (Approximate) Linear Optimizer: $M_\tau : \mathbb{R}^K \to \mathcal{C}$ such that for any $w \in \mathbb{R}^K$
$$w \cdot \ell(M_{\tau(w)}) \leq \min_{c \in \mathcal{C}} w \cdot \ell(c) + \tau \|w\|$$
3. Oracle access to $f : [0,1]^K \to \mathbb{R}$

Main Theorem & Learning Corollary

Theorem: For constant (small) $K \geq 1$ and any $\epsilon \in (0,1]$,
$$f\left(\ell(GroupOpt(\epsilon, f, \ell_\tau, M_\tau))\right) \leq \min_{c \in \mathcal{C}} f(\ell(c)) + \epsilon$$
GroupOpt makes poly($1/\epsilon$) calls to ℓ_τ, M_τ, f with $\tau = \frac{\epsilon^2}{25\sqrt{K}}$.

Corollary: Let M be an efficient agnostic learner and $f : [0,1]^{2K} \to \mathbb{R}$ be any L-Lipschitz function. For any $\epsilon, \delta \in (0,1]$, with probability $\geq 1 - \delta$, we output $\hat{c}$ such that
$$f(FPR(\hat{c}), FNR(\hat{c})) \leq \min_{c \in \mathcal{C}} f(FPR(c), FNR(c)) + \epsilon$$
using oracle access to $p_k^i = \mu(X_k \times \{i\})$ and poly$\left(L, \frac{1}{\epsilon}, \frac{1}{\delta}, \frac{1}{\min_{\{k,i\}} p_k^i}\right)$ examples and calls to f, M.

Idea: We simulate ℓ_τ, M_τ using a polynomial number of examples.