Review: On the Within-Group Fairness of screening Classifier

February 22, 2024

Seoul National University

Introduction

- Any threshold decision rule that uses calibrated screening classifiers may be **biased against qualified candidates within demographic groups** of interest
- More specifically, it may shortlist one or more candidates from a group who are less likely to be qualified than one or more rejected candidates from the same group.
- They have developed **a polynomial time algorithm based on dynamic programming** to minimally modify any given calibrated classifier so that it satisfies **within-group monotonicity**, a natural monotonicity property that prevents the occurrence of within-group unfairness.

Preliminaries

- Notation

$$a \text{ candidate with a feature vector } x \in \mathcal{X}$$

$$\textit{demographic group } z \in Z \text{ , can be qualified } (y = 1) \text{ or unqualified } (y = 0)$$

$$f : \mathcal{X} \rightarrow \text{ Range } (f) \subseteq [0, 1] : \textit{ calibrated screening classifier}$$

$$f \textit{ is calibrated iff } \forall a \in Range(f), \; P(Y = 1 \mid f(X) = a) = a$$

$$\text{a screening policy } \pi : [0, 1]^m \rightarrow \mathcal{P}\left(\{0, 1\}^m\right)$$

$$\text{threshold decision rule : } \; s_i = \begin{cases} 1 & \text{if } f\left(x_i\right) > t_f \\ \text{Bernoulli}\left(\theta_f\right) & \text{if } f\left(x_i\right) = t_f \\ 0 & \text{otherwise} \end{cases}$$

$$\textit{with candidate is shortlisted } (s_i = 1) \textit{ or is not shortlisted } (s_i = 0)$$

Unfairness

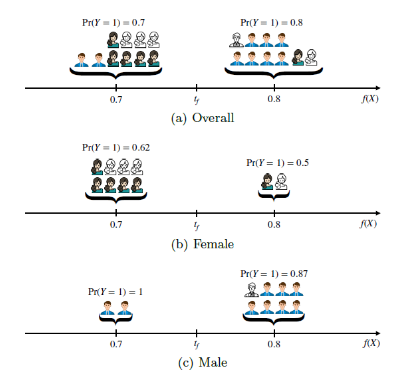
- The following proposition shows that any threshold decision rule may be biased against qualified members within demographic groups

- **Proposition 2.1** Let π be a screening policy given by a threshold decision rule using a calibrated classifier f with threshold t. Assume there exist $a, b \in$ Range (f), with $a < t < b$, and $z \in \mathcal{Z}$ such that $P(Y = 1 \mid f(X) = a, Z = z) > P(Y = 1 \mid f(X) = b, Z = z)$. Then, it holds that

  $$\mathbb{E}_{Y \sim P_{Y|X,Z}, S \sim \pi}[Y(1-S) \mid f(X) = a, Z = z] > \mathbb{E}_{Y \sim P_{Y|X,Z}, S \sim \pi}[YS \mid f(X) = b, Z = z]$$

- The above result implies that there exist pools of applicants for which an optimal policy using a calibrated classifier may shortlist a candidate from a group who is less likely to be qualified than a rejected candidate from the same group.

Unfairness

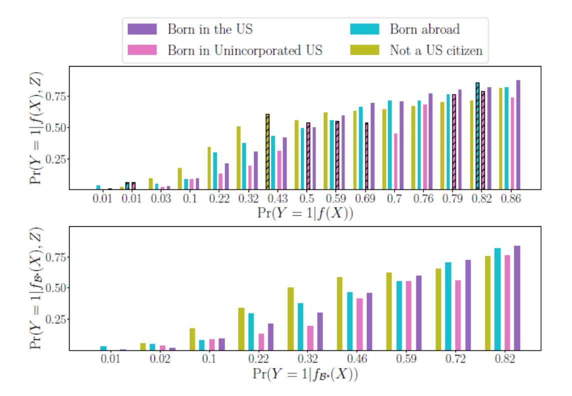


(a): candidates who are shortlisted ($f(X) > t$) are more likely to be qualified (Y$=1$) than those who are rejected ($f(X) < t$)

(b) and (c) show that, after conditioning on their gender, candidates who are rejected ($f(X) < t$) are more likely to be qualified than those who are short listed ($f(X) > t$)

within-group Monotonicity

- **Definition 2.2**
  Given a set of groups $\mathcal{Z}$, a classifier f is within-group monotone if, for any $z \in \mathcal{Z}$ and $a, b \in \text{Range}\,(f)$ such that $a < b, \Pr(Z = z \mid f(X) = a) > 0$ and $\Pr(Z = z \mid f(X) = b) > 0$, it holds that
  $$\Pr(Y = 1 \mid f(X) = a, Z = z) \leq \Pr(Y = 1 \mid f(X) = b, Z = z).$$



A Set Partitioning Post-Processing Framework

A Set Partitioning Post-Processing Framework

f: calibrated classifier with Range(f) $= \{a_1, ..., a_n\},\ Pr(f(X) = a_i) = \rho_i$

$$|Range(f)| = n < \infty$$

WLOG assume that $a_i < a_j$ *for any* $i < j$

$$Pr(Y = 1 \mid f(X) = a_i, Z = z) = a_{i,z} \text{ and}$$

$$Pr(Z = z \mid f(X) = a_i) = \rho_{z|i},\ a_i = \sum_{z \in \mathcal{Z}} \rho_{z|i} a_{i,z}$$

Then, our goal is to modify f minimally so that it is within-group monotone.

A Set Partitioning Post-Processing Framework

- Idea

  classifier f induces a partition of $\mathcal{X}$ into n disjoint bins $\{\mathcal{X}_1, \ldots, \mathcal{X}_n\}$

  where each bin $\mathcal{X}_i$ is characterized by a_i and ρ_i

  Then seek to **merge** a small number of these induced bins to achieve within-group monotonicity.

A Set Partitioning Post-Processing Framework

- Notation

$$\mathcal{P}: \ set\ of\ all\ partitions\ of\ the\ bin\ indices\ \{1, ..., n\}$$

$\mathcal{B} \in \mathcal{P}$: a partition of the bin indices into a collection of disjoint equivalence classes

$$, \{A_1, ..., A_{|B|}\}, \ which\ we\ call\ cells$$

$$i(x) = \{i | f(x) = a_i\}: \ for\ x \in \mathcal{X}, \ index\ of\ the\ bin\ it\ belongs$$

$$\text{represent a cell in } \mathcal{B}\ containg\ index\ i(x)\ by\ [i(x)]_B$$

$$f_{\mathcal{B}} : \mathcal{X} \to \text{Range}\left(f_{\mathcal{B}}\right) = \{a_{\mathcal{A}}\}_{\mathcal{A} \in \mathcal{B}}, \text{ where } a_{\mathcal{A}} = \frac{\sum_{j \in \mathcal{A}} a_j \rho_j}{\sum_{j \in \mathcal{A}} \rho_j} \text{ and } f_{\mathcal{B}}(x) = a_{[i(x)]}.$$

$$\Pr\left(Y = 1 \mid f_{\mathcal{B}}(X) = a_{\mathcal{A}}\right) = \frac{\sum_{j \in \mathcal{A}} a_j \rho_j}{\sum_{j \in \mathcal{A}} \rho_j} = a_{\mathcal{A}}$$

$$\Pr\left(Y = 1 \mid f_{\mathcal{B}}(X) = a_{\mathcal{A}}, Z = z\right) = \frac{\sum_{j \in \mathcal{A}} \rho_j \rho_{z|j} a_{j,z}}{\sum_{j \in \mathcal{A}} \rho_j \rho_{z|j}} := a_{\mathcal{A},z}$$

- Goal

$$\underset{\mathcal{B} \in P}{\text{maximize}} |\mathcal{B}| \quad \text{subject to} \quad a_{\mathcal{A}_i,z} \leq a_{\mathcal{A}_j,z} \forall \mathcal{A}_i, \mathcal{A}_j \in \mathcal{B} \text{ such that } a_{\mathcal{A}_i} < a_{\mathcal{A}_j}, \forall z \in \mathcal{Z}$$

Optimal Set Partitioning via Dynamic Programming

Optimal Set Partitioning via Dynamic Programming

Algorithm 2 It returns the optimal partition $\mathcal{B}^*$ such that $f_{\mathcal{B}^*}$ is within-group monotone.

Input: $\{a_{1,z}, \dots, a_{n,z}\}_{z \in \mathcal{Z}}$
Initialize: $\mathcal{B}_{l,r} = \{\}\ \forall l, r \in \{2, \dots, n\}$, $\mathcal{B}_{1,r} = \{1, \dots, r\}\ \forall r \in \{1, \dots, n\}$
for $l \in \{2, \dots, n\}$ **do**
  for $r \in \{l, \dots, n\}$ **do**
    $\mathcal{S}_{l,r} = \left\{k | k < l, a_{\{k,\dots,l-1\},z} \leq a_{\{l,\dots,r\},z}\ \forall z \in \mathcal{Z}\right\}$ {Refer to Lemma. 4.3}
    if $\mathcal{S}_{l,r} = \emptyset$ **then**
      Continue {*In this case* $\mathscr{B}_{l,r} = \emptyset$}
    end if
    $k^* = \text{argmax}_{k \in \mathcal{S}_{l,r}} |\mathcal{B}_{k,l-1}|$
    $\mathcal{B}_{l,r} = \mathcal{B}_{k^*,l-1} \cup \{\{l, \dots, r\}\}$
  end for
end for
$l^* = \text{argmax}_{i \in \{1,\dots,n\}} |\mathcal{B}_{i,n}|$
return $\mathcal{B}_{l^*,n}$

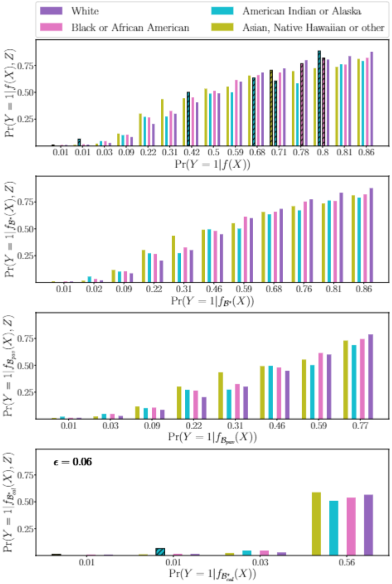
Optimal Set Partitioning via Dynamic Programming

Let B_r be the set of partitions of the *bin indices*$\{1, \ldots, r\}$, *with* $r \leq n$, and $B_{l,r} \subseteq B_r$ be the subset of those partitions such that, for any $\mathcal{B} = \left\{\mathcal{A}_1, \ldots, \mathcal{A}_{|\mathcal{B}|}\right\} \in B_{l,r}$, it holds that $\mathcal{A}_{|\mathcal{B}|} = \{l, \ldots, r\}$ and $f_{\mathcal{B} \cup \mathcal{B}'}$ is within-group monotone on the region of the feature space defined by $\cup_{i \leq r} \mathcal{X}_i$, where $\mathcal{B}'$ is any partition of the bin indices $\{r+1, \ldots, n\}$.

Then, it clearly holds that the optimal partition $\mathcal{B}^* \in \cup_{l=1}^{n} B_{l,n}$ and thus we can break the problem of finding $\mathcal{B}^*$ into n subproblems, i.e., finding the optimal partition $\mathcal{B}^*_{l,n} = \text{argmax}_{\mathcal{B} \in B_{l,n}} |\mathcal{B}|$ within in each subset $B_{l,n}$.

Consequently, we can efficiently find all the partitions in the subsets $B_{l,r}$ iterating through l using the partitions in the subsets $B_{k,l-1}$ with $k < l$. Finally, by construction, it clearly holds that, if $\mathcal{B}^*_{l,r} = \mathcal{B}' \cup \{\{l, \ldots, r\}\}$, with $\mathcal{B}' \in B_{k,l-1}$, is the optimal partition in $B_{l,r}$ then $\mathcal{B}' = \mathcal{B}^*_{k,l-1}$ is the optimal partition in $B_{k,l-1}$. As a result, at each step of the recursion, we only need to store the optimal partition $\mathcal{B}^*_{l,r}$, not all partitions in $B_{l,r}$.

Experiments



(b) Race code (Z)