

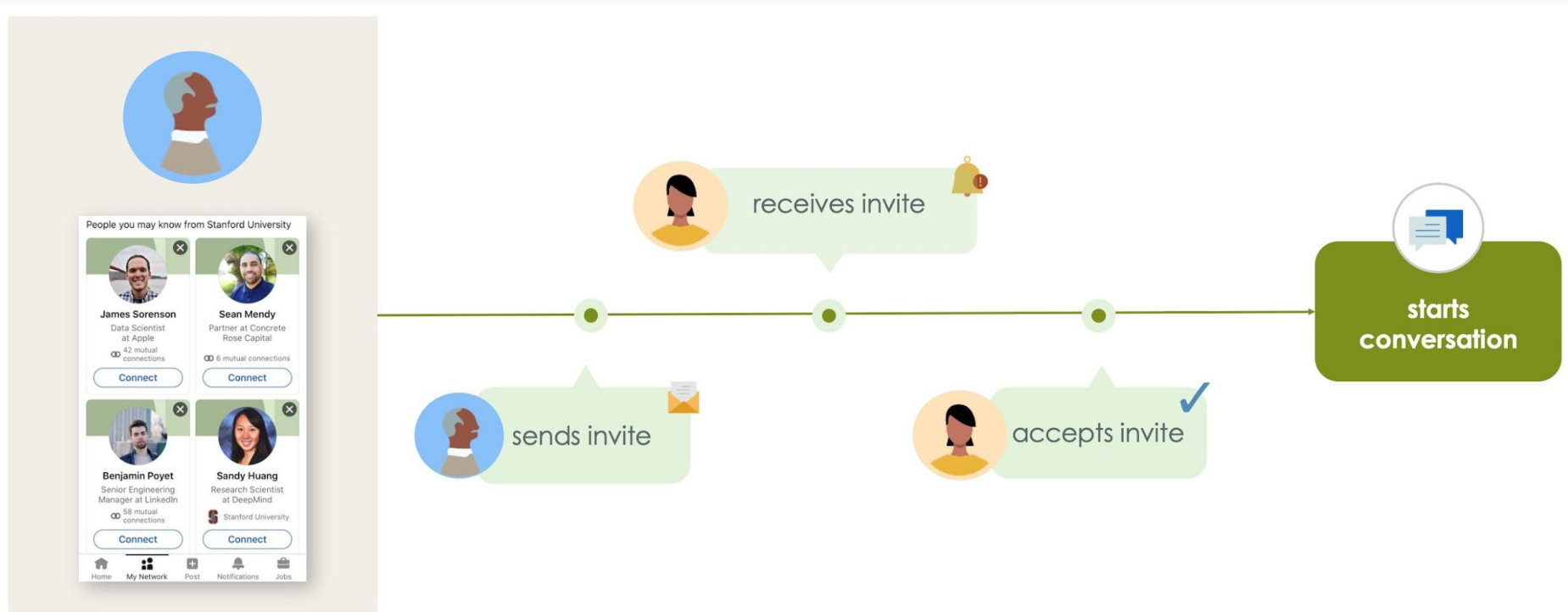
Achieving Fairness via post-processing in Web-Scale Recommender Systems

Kinjal Basu

CMStatistics 2021

Joint work with Preetam Nandy, Cyrus Diccio, Heloise Logan, Divya Venugopalan and Nouredine El Karoui

Connection Recommendation at LinkedIn



Fairness Criteria for score-based Ranking

Notation: Binary response Y , predictors X , prediction Score $s(X)$, characteristic C (e.g. gender).

1. **Unawareness:** X does not contain C
2. **Demographic Parity:** $s(X)$ is independent of C
3. **Equality of Opportunity:** $s(X)$ is independent of C conditional on $Y = 1$
4. **Equalized Odds:** $s(X)$ is independent of C conditional on Y
5. **Fair Offline Performance:** equal (partial) ROC-AUC of $(s(X), Y)$ given $C = c$ for all c

This is by no means an exhaustive list.

Fairness Criteria for score-based Ranking

Notation: Binary response Y , predictors X , prediction Score $s(X)$, characteristic C (e.g. gender).

1. **Unawareness:** X does not contain C
2. **Demographic Parity:** $s(X)$ is independent of C
3. **Equality of Opportunity:** $s(X)$ is independent of C conditional on $Y = 1$
4. **Equalized Odds:** $s(X)$ is independent of C conditional on Y
5. **Fair Offline Performance:** equal (partial) ROC-AUC of $(s(X), Y)$ given $C = c$ for all c

This is by no means an exhaustive list.

Mitigation Strategies

Preprocessing

Removing bias from training and validation data

Inprocessing

Changes in the model training to achieve fairness goals

Regularizers or changes to the loss functions

Post Processing

Changing model-scores after training

Model-agnostic

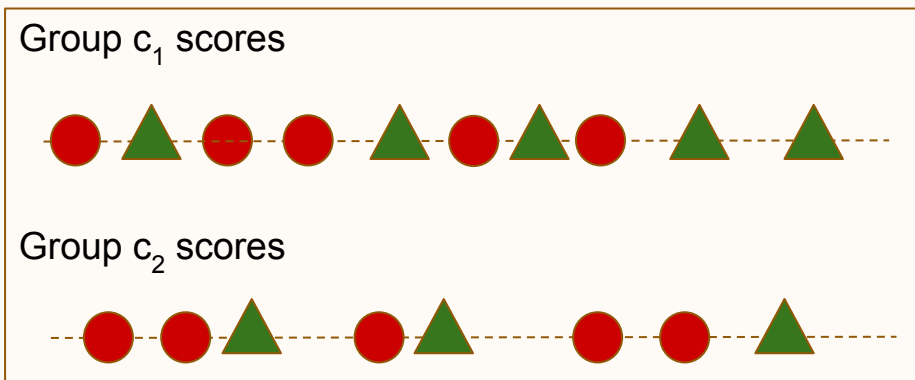
Wide-applications in large-organization

Equality of Opportunity

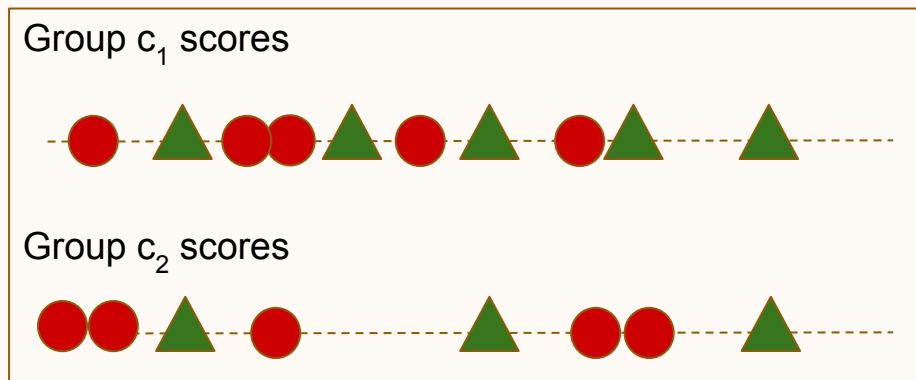
CDF Transformation


- **EOpp Definition:** $P(s \leq t \mid Y = 1, C = c_1) = P(s \leq t \mid Y = 1, C = c_2)$
Equivalently: S is independent of C given Y = 1
- **EOpp Transformation:** Apply $F_{c,1}(s_c)$ to all scores in group $C=c$, where $F_{c,1}$ is the CDF of scores in group c with $Y=1$.

Before EOpp transformation



After EOpp transformation



 Scores with $Y=1$  Scores with $Y=0$

Fairness-Performance Trade-off

- EOpp transformation: $s^* = \sum F_{c,1}(s) 1_{\{C=c\}}$
- G is the CDF of transformed scores $\Rightarrow G(s^*)$ has a $Uniform[0, 1]$ distribution
- F is CDF of original scores $\Rightarrow F^{-1}(G(s^*))$ brings scores back to original scale
- To relax the strict Equality of Opportunity, we may consider the following modification:

$$(1 - \alpha) s + \alpha F^{-1}(G(s^*))$$

where $0 \leq \alpha \leq 1$ can be tuned to achieve a fairness-performance trade-off.

Equalized Odds

Equalized Odds for Ranking

Definition: The ranker satisfies Equalized Odds with respect to characteristic (attribute) C and label Y if

$$\begin{aligned}P(s \geq t \mid C = c_1, Y=1) &= P(s \geq t \mid C = c_2, Y=1), \text{ and} \\P(s \geq t \mid C = c_1, Y=0) &= P(s \geq t \mid C = c_2, Y=0) \\&\text{for all } t,\end{aligned}$$

Equivalently:

- S is independent of C given Y

Properties:

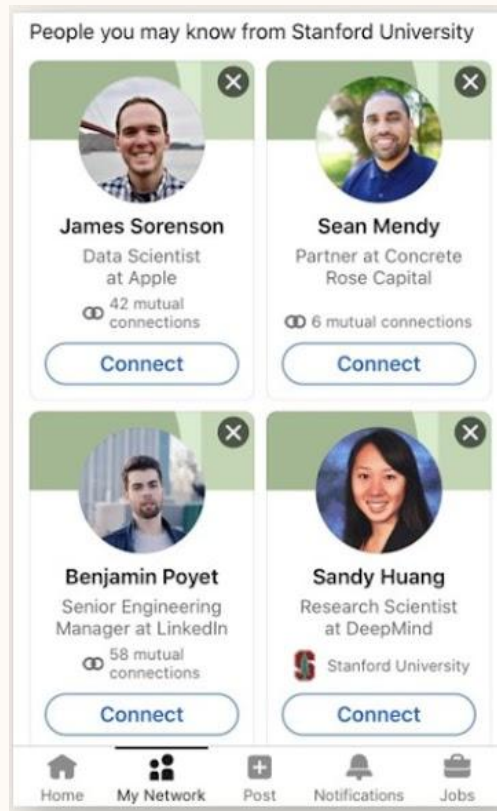
- The ROC curves are identical between groups

Connection Recommendation Example

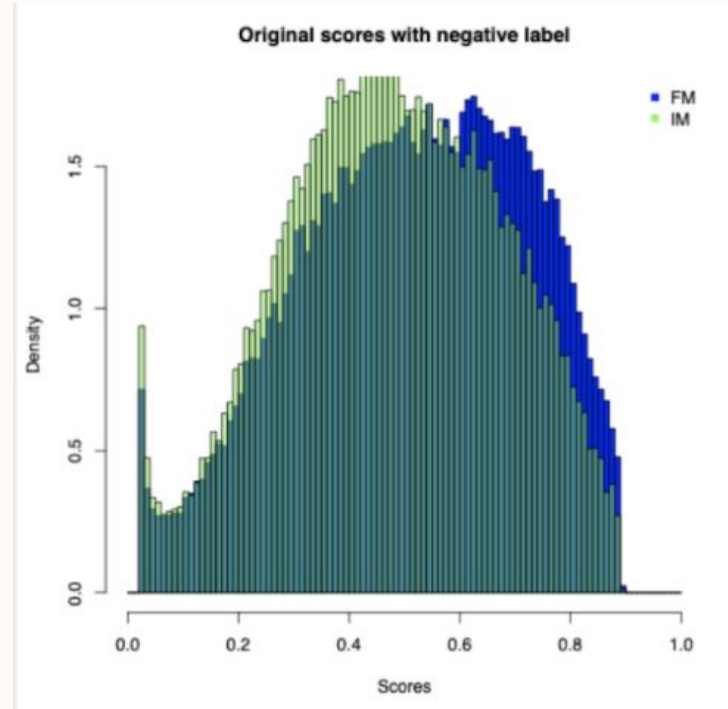
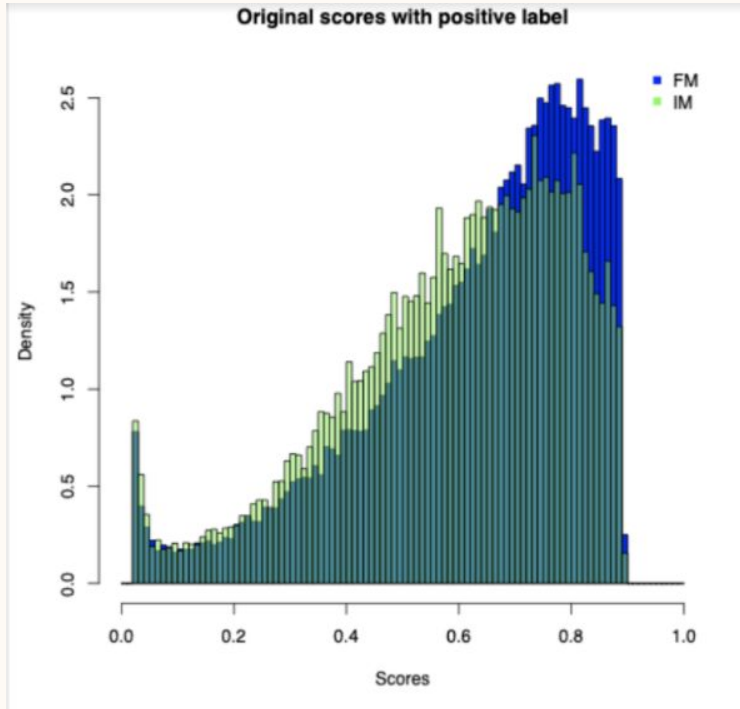
Fairness to members being recommended:

- “Positive” outcome $Y = 1$: Invite sent
- “Negative” outcome $Y = 0$: Invite not sent
- Group: Frequent and Infrequent members

Equalized odds ensures fair exposure of candidate that will (or will not) be sent invitations



Exposure of Frequent and Infrequent Members



How to Re-Rank for Equalized Odds

- Unlike equality of opportunity, there is **not** a (strictly) monotonic transformation that gives equalized odds.
 - Re-ranking may result in some loss of model performance
- **Simple idea:** For a classifier, we could get equalized odds by randomly changing classification with suitable probabilities
 - For an individual/item with characteristic $C=c$, replace $Y = y$ with a draw from $\text{Bernoulli}(p_{y,c})$
 - Probabilities chosen so that the equalized odds constraints hold

Equalized odds can be ensured through a probabilistic allocation

Methodology

- Bin the scores into disjoint intervals b_1, \dots, b_T
- Randomize model scores between the intervals so that the probability that the score falls in each of these intervals is independent of C given Y .
- $t(S)$ = new score after probabilistic allocation

$$\begin{aligned} &|P(t(S) \text{ in } b_t \mid C=k, Y=1) - P(t(S) \text{ in } b_t \mid C=l, Y=1)| < \varepsilon, \text{ and} \\ &|P(t(S) \text{ in } b_t \mid C=k, Y=0) - P(t(S) \text{ in } b_t \mid C=l, Y=0)| < \varepsilon \\ &\text{for all } t, k, l \end{aligned}$$

- ε allows for a tradeoff between fairness and model performance
- How do we choose the probability of assigning the scores to bins?

LP formulation

Objective: Maximize Model Performance

s.t. Equalized Odds constraint hold

What is the objective that “Maximize Model Performance”?

- Minimize score changes due to calibration

Minimize $E|t(S) - S|$ where $t(s)$ is the transformed score

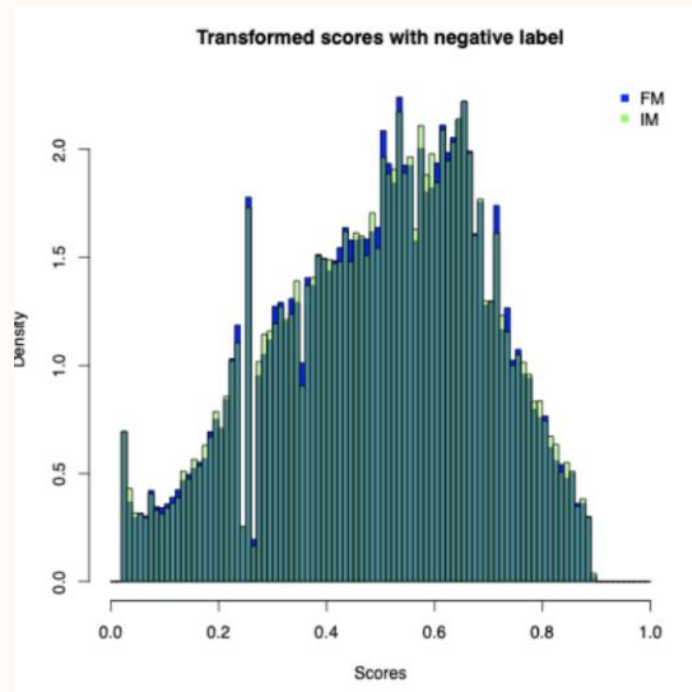
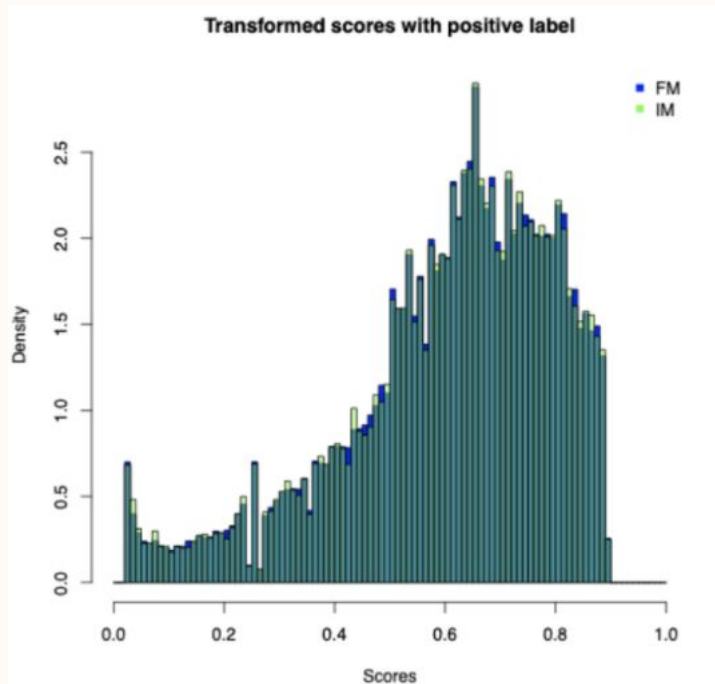
- An approximation to AUC is a quadratic function of the probabilities
- In general, any loss function is okay, but may make solving the optimization problem challenging

Minimize $E|t(S) - S|$

s.t. $|P(t(S) \text{ in } b_t \mid C=k, Y=1) - P(t(S) \text{ in } b_t \mid C=l, Y=1)| < \varepsilon$

$|P(t(S) \text{ in } b_t \mid C=k, Y=0) - P(t(S) \text{ in } b_t \mid C=l, Y=0)| < \varepsilon$ for all t, k, l

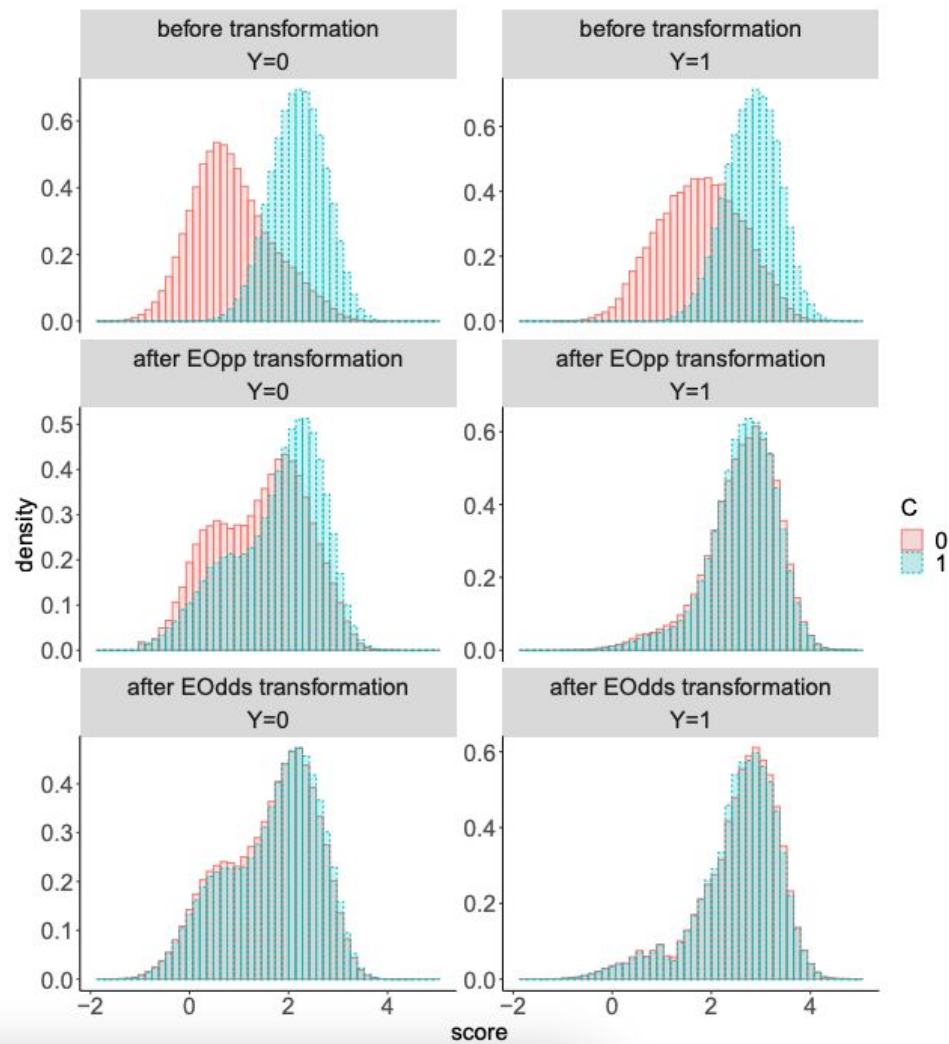
Exposure after re-ranked for equalized odds



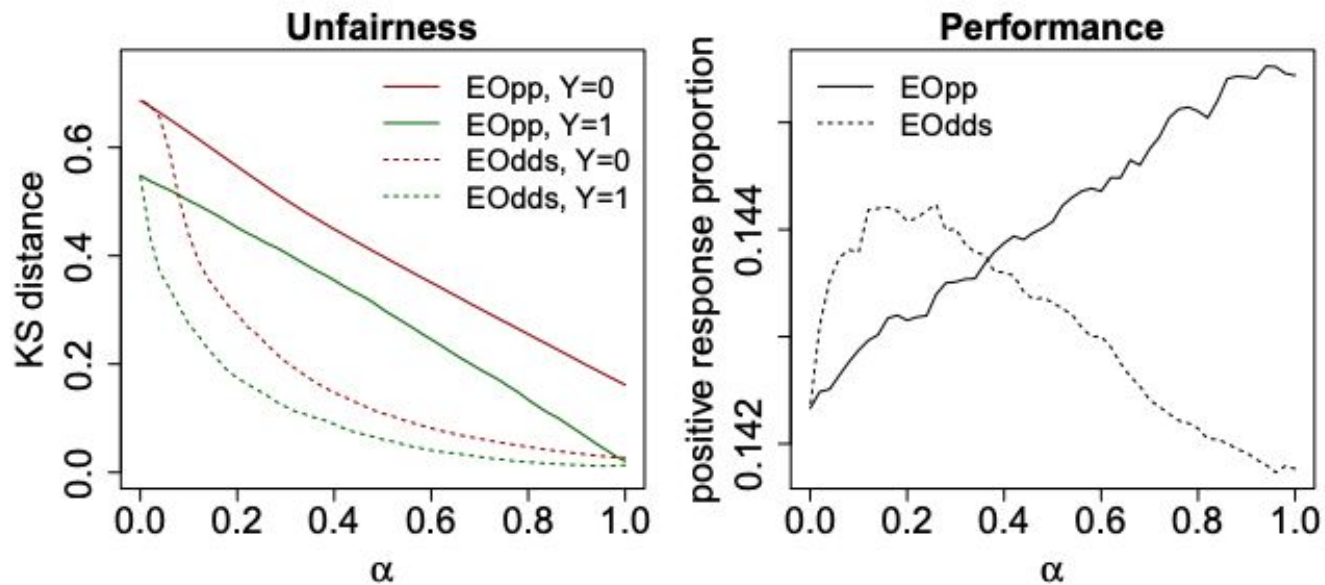
Experiments

Simulation Setup

- We create a population of 50k items with **item_id**, **relevance_score**, **true_label**, **characteristic**
- **Training Data:** 100k sessions with randomly selecting 100 items from the population, where for all 100 items in each session, we obtain
 - $\text{observed_score} = \text{relevance_score} + \text{noise}$;
 - $\text{observed_position} = \text{rank based on observed_scores}$;
 - $\text{observed_label} = \text{true_label} * \text{Bernoulli}(1/\log(1+j))$
- **Validation Data (EOpp/EOdds):** 50k sessions with randomly selecting 100 items from the population, where for all 100 items in each session, we obtain
 - $\text{fair_score} = \text{fairness_transformation}(\text{relevance_score} + \text{noise})$;
 - $\text{observed_position} = \text{rank based on fair_score}$;
 - $\text{observed_label} = \text{true_label} * \text{Bernoulli}(1/\log(1+j))$



Fairness-performance trade-off



Real World Experiments

Invitation Metrics	<i>EOpp</i>		<i>EOdds</i>	
	IM	FM	IM	FM
Sent	+5.72%	Neutral	+2.77%	Neutral
Accepted	+ 4.85%	Neutral	+ 2.26%	Neutral

Table 1. A/B Experimentation results for the two fairness re-rankers. In both setups, we observed improved metrics of invitations sent and accepted by IMs without any statistically significant impact to the same metrics corresponding to FM.

Key Takeaways

- Mechanism of mitigation unfairness through a post-processing system.
 - Agnostic to how a model is getting trained.
 - Wide applicability
 - Scalability
- Fairness Performance tradeoff can vary across applications
- Extensibility
 - multiple outcomes
 - multiple fairness constraints
 - Position bias ([full paper](#))

Thank you!
Questions?

Appendix

Position Bias Adjustment

EOpp/EOdds in the presence of position bias

- Let $Y(j)$ denote the counterfactual response when an item is placed at position j .
- **Position Bias (positive response decay):** $w_j = P(Y(j) = 1 \mid Y(1) = 1) < 1$.
- Let γ denote the observed position and let $Y(\gamma)$ denote the observed response.

- **EOpp and EOdds:**

$$P(s \leq t \mid Y(\gamma) = y, C = c_1) = P(s \leq t \mid Y(\gamma) = y, C = c_2) \text{ for all } c_1, c_2$$

where $y = 1$ for EOpp and $y = 0, 1$ for EOdds.

- Without position bias adjustment we will have

$$P(s^* \leq t \mid Y^*(\gamma) = y, C = c_1) = P(s^* \leq t \mid Y^*(\gamma) = y, C = c_2) \text{ for all } c_1, c_2$$

But we want to have

$$P(s^* \leq t \mid Y^*(\gamma^*) = y, C = c_1) = P(s^* \leq t \mid Y^*(\gamma^*) = y, C = c_2) \text{ for all } c_1, c_2$$

where γ^* is the position based on the ranking given by s^* .

Position bias estimation

Under mild assumptions, $w_j = P(Y(j) = 1) / P(Y(1) = 1)$

With randomization

$$\hat{w}_j = \frac{(\sum_i 1_{\{Y_i(\gamma)=1, \gamma=j\}}) / (\sum_i 1_{\{\gamma=j\}})}{(\sum_i 1_{\{Y_i(\gamma)=1, \gamma=1\}}) / (\sum_i 1_{\{\gamma=1\}})}.$$

Without randomization

Step 1: Importance weighting to adjust for score distribution discrepancies

$$\eta_j = \frac{\mathbf{E} \left(Y(\gamma) \frac{f_{j-1}(s(X))}{f_j(s(X))} \mid \gamma = j \right)}{\mathbf{E} (Y(\gamma) \mid \gamma = j - 1)}.$$

Step 2: Truncated product to control the variance of the estimator

$$\hat{w}_j = \prod_{r=2}^{\min(j, T)} \hat{\eta}_r$$

Algorithm 1 Position Bias Adjusted Equality of Opportunity

1: Reranker Training

2: Input: Score, position, label and characteristic data $(s_i, \gamma_i, y_i, c_i), i = 1, \dots, n$

3: Compute the weighted empirical CDF $\hat{F}_{c,1}^*$ of the conditional scores $s(X)$ given $Y(\gamma) = 1$ and $C = c$ with weights $1/\hat{w}_\gamma$ computed as in Section 4.3

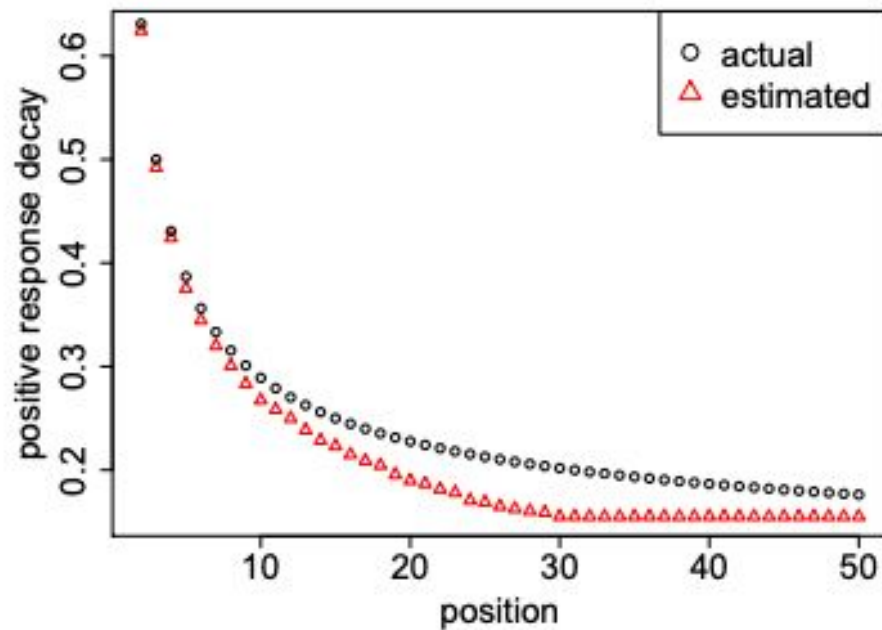
4: Output: The empirical distribution functions $\hat{F}_{c,1}^*$

5: Reranker Scoring

6: Input: Score S , characteristic C

7: Output: Fair score, $\tilde{S} = \sum_c \hat{F}_{c,1}^*(S) 1_{\{C=c\}}$

Position bias estimation



Extensions to multiple outcomes

This framework can be extended to categorical outcomes with (arbitrary) possible values as well as arbitrary number of (attribute) groups

PYMK example: Fairness to members being recommended. We can consider more granular outcomes:

- Invite not sent
- Invite sent but not accepted
- Invite sent and accepted

Generally, we can use the equalized odds framework to balance exposure according to outcomes along funnel metrics

LP formulation: Simply add constraints for additional outcomes!

Objective: Maximize Model Performance

s.t. Bin probabilities for each outcome are within ϵ for each pair of groups

What is the objective that “Maximize Model Performance”?

- Minimize score changes due to calibration

$$\text{Minimize } E|t(S) - S|$$

where $t(s)$ is the transformed score

- For multiple outcomes, auc may no longer make sense