FA*IR: A Fair Top-k Ranking Algorithm

Zehlike, Meike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates.

Presented By: Jiajia Liang, Linyang Du, Jie Fan

UVA ENGINEERING

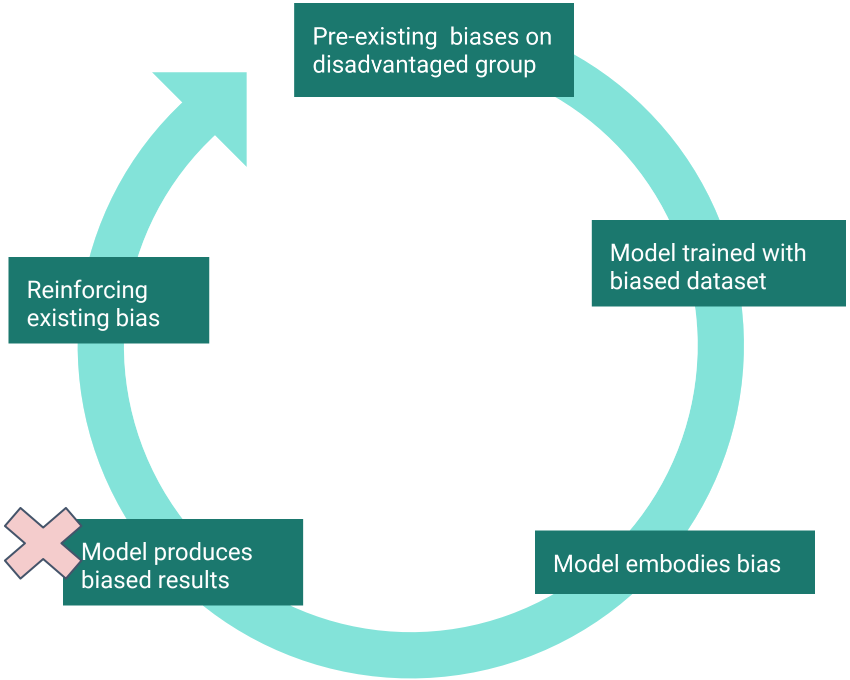
Outline

- Part 1: Motivation
- Part 2: Fair Top-k Ranking Problem
- Part 3: Algorithm
- Part 4: Experiments & Results
- Part 5: Contributions & Limitations

1: Motivation

If you are a hiring manager, you need to select a smaller group that will be interviewed for a position from a large pool of candidates.

1: Motivation



1: Motivation & Why it Matters

Desired properties:

- Fairness: representation of the protected group does not fall below a minimum proportion at any point in the ranking.
- Maximize utility: interview the most qualified candidates

Part 2: The Fair Top-k Ranking Problem

2. The Fair Top-k Ranking Problem

Fairness top-k ranking criteria:

A ranking selection should include candidates with following characteristics.

1. Ranked group fairness: represent protected group.
2. Selection utility: contain most qualified candidates.
3. Ordering utility: ordered by decreasing qualification.

2.2 Group fairness for ranking

Definition 1: Set of candidates must have protected candidates **fairly represent the protected group with minimal proportion p and significance a**.

Definition 2: Every candidate within top-k ranking needs to satisfy the **fairness representation condition with proportion p and adjusted significance a_c**.

2.3 Criteria for Utility

Definition 3: Ranked utility. Maximum ranked individual utility must be at the top of the ranking list.

Definition 4: Selection utility. Prefer rankings in which the more qualified candidates are included and the less qualified excluded.

Definition 5: Ordering utility. Prefer top-k lists in which more qualified candidates are ranked above less qualified ones.

Definition 6: In group monotonicity. Both protected and non-protected candidates must be sorted by decreasing qualifications.

Part 3: Algorithm

```
P0, P1 ← empty priority queues with bounded capacity k
for i ← 1 to n do
    insert i with value q_i in priority queue P_{g_i}
end
for i ← 1 to k do
    m[i] ← F^{-1}(α_c; i, p)
end
(t_p, t_n) ← (0, 0)
while t_p + t_n < k do
    if t_p < m[t_p + t_n + 1] then
        // add a protected candidate
        t_p ← t_p + 1
        τ[t_p + t_n] ← dequeue(P1)
    else
        // add the best candidate available
        if q(peek(P1)) ≥ q(peek(P0)) then
            t_p ← t_p + 1
            τ[t_p + t_n] ← dequeue(P1)
        else
            t_n ← t_n + 1
            τ[t_p + t_n] ← dequeue(P0)
        end
    end
end
return τ
```

1. Create two lists $P_{protected}$ and $P_{notProtected}$ contain top k candidates from *protected* group and *not protected* group

2. Compute ranked group fairness table m with p, k, α

p \ k	1	2	3	4	5	6	7	8	9	10	11	12
0.1	0	0	0	0	0	0	0	0	0	0	0	0
0.2	0	0	0	0	0	0	0	0	0	0	1	1
0.3	0	0	0	0	0	0	1	1	1	1	1	2
0.4	0	0	0	0	1	1	1	1	2	2	2	3
0.5	0	0	0	1	1	1	2	2	3	3	3	4
0.6	0	0	1	1	2	2	3	3	4	4	5	5
0.7	0	1	1	2	2	3	3	4	5	5	6	6

$m_{p,k}$ = the minimum number of candidates in the protected group that must appear

while number of picked < k:

If m demands a protected candidate at the current position: add the best candidate from $P_{protected}$

Otherwise, add the best from $P_{protected}$ U $P_{notProtected}$

FAIR algorithm satisfies all following:

(i) Satisfies in-group monotonicity
(ii) Satisfies ranked group fairness
(iii) Achieve optimal selection utility
(iv) Maximizes ordering utility

Runtime: O(n + k log k)

1. Create two lists $P_{protected}$ and $P_{notProtected}$ contain top k candidates from *protected* group and *not protected* group

2. Compute ranked group fairness table m with p, k, α

p \ k	1	2	3	4	5	6	7	8	9	10	11	12
0.1	0	0	0	0	0	0	0	0	0	0	0	0
0.2	0	0	0	0	0	0	0	0	0	0	1	1
0.3	0	0	0	0	0	0	1	1	1	1	1	2
0.4	0	0	0	0	1	1	1	1	2	2	2	3
0.5	0	0	0	1	1	1	2	2	3	3	3	4
0.6	0	0	1	1	2	2	3	3	4	4	5	5
0.7	0	1	1	2	2	3	3	4	5	5	6	6

$m_{p,k}$ = the minimum number of candidates in the protected group that must appear

while number of picked < k:

If m demands a protected candidate at the current position: add the best candidate from $P_{protected}$

Otherwise, add the best from $P_{protected}$ U $P_{notProtected}$

3: Runtime

O(n) to build P

1. Create two priority queues with k candidates each: $P_{protected}$ and $P_{notProtected}$

O(n) to compute table

2. Compute ranked group fairness table m

p \ k	1	2	3	4	5	6	7	8	9	10	11	12
0.1	0	0	0	0	0	0	0	0	0	0	0	0
0.2	0	0	0	0	0	0	0	0	0	0	1	1
0.3	0	0	0	0	0	0	1	1	1	1	1	2
0.4	0	0	0	0	1	1	1	1	2	2	2	3
0.5	0	0	0	1	1	1	2	2	3	3	3	4
0.6	0	0	1	1	2	2	3	3	4	4	5	5
0.7	0	1	1	2	2	3	3	4	5	5	6	6

$m_{p,k}$ = the minimum number of candidates in the protected group that must appear

Runtime: O(n + k log k)

k iteration

O(logk) to dequeue

while number of picked < k:

If m demands a protected candidate at the current position: add the best candidate from $P_{protected}$

Otherwise, add the best from $P_{protected}$ U $P_{notProtected}$

Part 4: Experiments & Results

4: Experiments & Results

Datasets

1. COMPAS
2. German Credits
3. SAT
4. XING

	Dataset	n	k	Quality criterion	Protected group	Protected %
D1	COMPAS [1]	18K	1K	¬recidivism	Afr.-Am.	51.2%
D2	"	"	"	"	male	80.7%
D3	"	"	"	"	female	19.3%
D4	Ger. credit [27]	1K	100	credit rating	female	69.0%
D5	"	"	"	"	< 25 yr.	14.9%
D6	"	"	"	"	< 35 yr.	54.8%
D7	SAT [34]	1.6 M	1.5K	test score	female	53.1%
D8	XING [ours]	40	40	ad-hoc score	f/m/f	27/43/27%

	Method	% Prot. output	NDCG	Ordering utility loss	Rank drop	Selection utility loss
D1 (51.2%)	Color-blind	25%	1.0000	0.0000	0	0.0000
COMPAS,	FA*IR p=0.5	46%	**0.9858**	**0.2026**	**319**	**0.1087**
race=Afr.-Am.	Feldman et al.	51%	0.9779	0.2281	393	0.1301
D2 (80.7%)	Color-blind	73%	1.0000	0.0000	0	0.0000
COMPAS,	FA*IR p=0.8	77%	**1.0000**	**0.1194**	**161**	**0.0320**
gender=male	Feldman et al.	81%	0.9973	0.2090	294	0.0533
D3 (19.3%)	Color-blind	28%	1.0000	0.0000	0	0.0000
COMPAS,	FA*IR p=0.2	28%	**0.9999**	**0.2239**	**1**	**0.0000**
gender=female	Feldman et al.	19%	0.9972	0.3028	278	0.0533
D4 (69.0%)	Color-blind	74%	1.0000	0.0000	0	0.0000
Ger. cred,	FA*IR p=0.7	74%	**1.0000**	**0.0000**	**0**	**0.0000**
gender=female	Feldman et al.	69%	0.9988	0.1197	8	0.0224
D5 (14.9%)	Color-blind	9%	1.0000	0.0000	0	0.0000
Ger. cred,	FA*IR p=0.2	15%	**0.9983**	**0.0436**	**7**	**0.0462**
age < 25	Feldman et al.	15%	0.9952	0.1656	8	**0.0462**
D6 (54.8%)	Color-blind	24%	1.0000	0.0000	0	0.0000
Ger. cred,	FA*IR p=0.6	50%	**0.9913**	**0.1137**	**30**	**0.0593**
age < 35	Feldman et al.	55%	0.9853	0.2123	36	0.0633
D7 (53.1%)	Color-blind	49%	1.0000	0.0000	0	0.0000
SAT,	FA*IR p=0.6	57%	**0.9996**	**0.0167**	365	0.0083
gender=female	Feldman et al.	56%	**0.9996**	**0.0167**	**241**	**0.0042**
D8a (27.5%)	Color-blind	28%	1.0000	0.0000	0	0.0000
Economist,	FA*IR p=0.3	28%	**1.0000**	**0.0000**	**0**	**0.0000**
gender=female	Feldman et al.	28%	0.9935	0.6109	5	**0.0000**
D8b (42.5%)	Color-blind	43%	1.0000	0.0000	0	0.0000
Mkt. Analyst,	FA*IR p=0.4	43%	**1.0000**	**0.0000**	**0**	**0.0000**
gender=male	Feldman et al.	43%	0.9422	1.0000	5	**0.0000**
D8c (29.7%)	Color-blind	30%	1.0000	0.0000	0	0.0000
Copywriter,	FA*IR p=0.3	30%	**1.0000**	**0.0000**	**0**	**0.0000**
gender=female	Feldman et al.	30%	0.9782	0.4468	10	**0.0000**

4: Experiments & Results

Baseline

Color-blind ranking

-Without considering group fairness

Ranking method by Feldman et al

-Align the probability distribution of the protected candidates with the non-protected ones.

-Candidate i in the protected group, qi ← qj by choosing a candidate j in the non-protected group having Fn(j) = Fp(i)

-Fp(·) - quantile of candidates in protected group

-Fn(·) - quantile of candidates in non-protected group

Part 5: Contributions & Limitations

5: Contributions & Limitations

Contributions -Principled definition of ranked group fairness -Effective algorithm consider group fairness (create rankings for different portion of protected group)

Limitation -Lack of considering multiple protected groups or combinations of protected attributes

Zehlike, Meike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. “FA*IR: A Fair Top-k Ranking Algorithm.” In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 1569–78. CIKM ’17. New York, NY, USA: Association for Computing Machinery, 2017. https://doi.org/10.1145/3132847.3132938.

Questions?