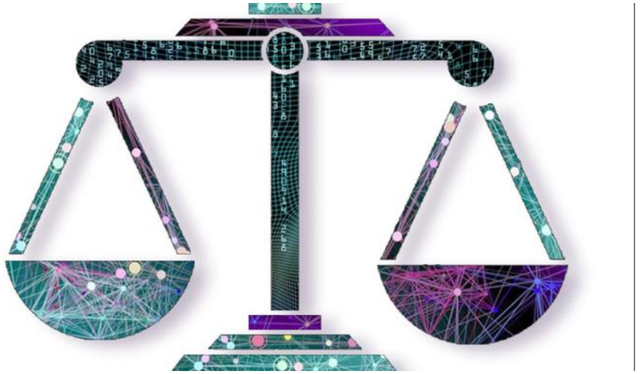
Fairness in Rankings and Recommenders

Konstantinos Stefanidis (TAU), Evaggelia Pitoura (UOI), Georgia Koutrika (ATHENA RC)

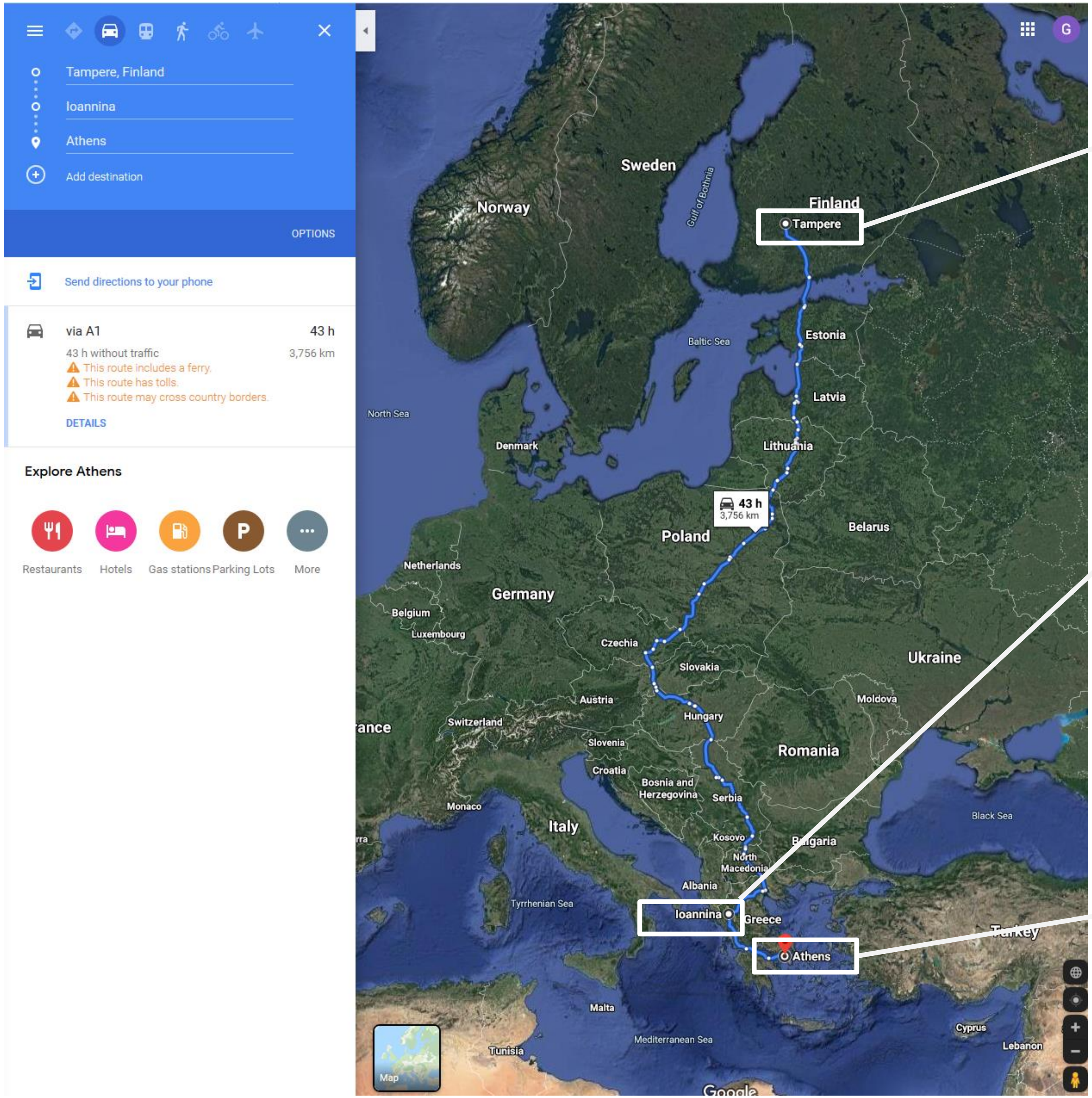


EDBT ICDT COPENHAGEN 2020

EDBT/ICDT 2020 Joint Conference

30th March-2nd April, 2020

Copenhagen, Denmark



Konstantinos Stefanidis
University of Tampere, Finland

Evaggelia Pitoura
University of Ioannina, Greece

Georgia Koutrika
ATHENA RC, Greece

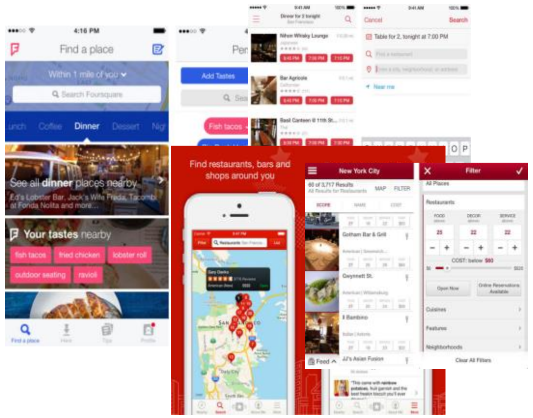
Introduction

Algorithmic fairness: why?

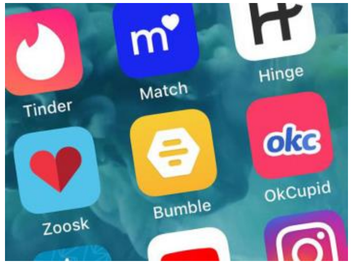
We live in a world where decisions are assisted or even taken by algorithmic systems driven by large amounts of data.

From simple, or not that simple, personal ones

Where to eat?

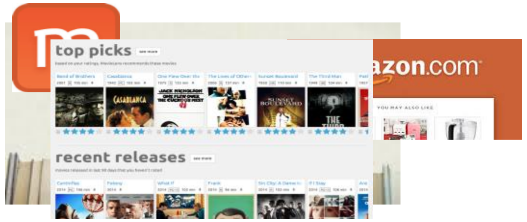


Who to date?



What are the news?

What to read, watch, buy..?



Get informed .. What does this mean?

WIKIPEDIA
The Free Encyclopedia

Google

What job to take? What school to attend? Who to follow? …? ..?

Algorithmic fairness: why?

We live in a world where decisions are assisted or even taken by algorithmic systems driven by large amounts of data.

And not just at a personal level

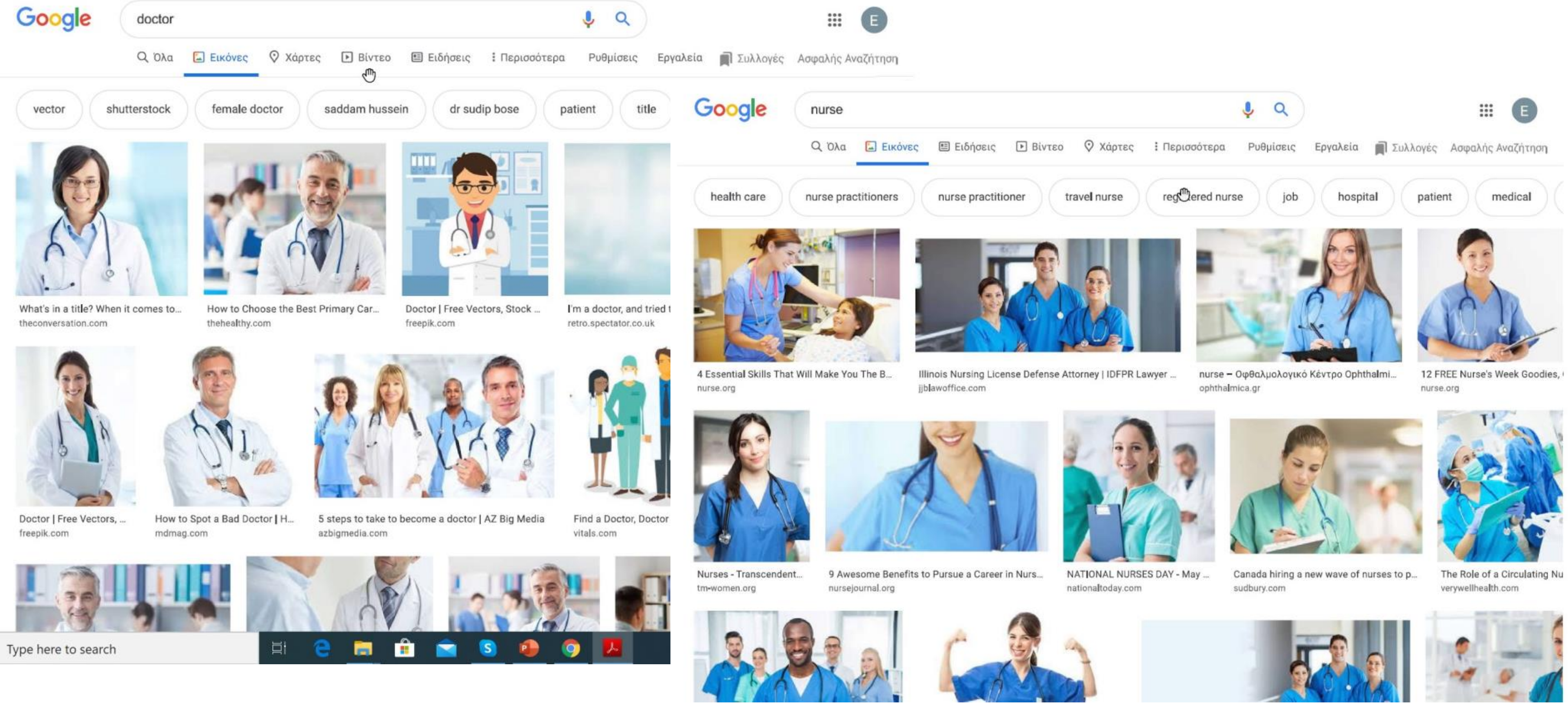
- Insurance, Credit
- Housing
- Pricing of goods and services
- Education, school admission
- Law enforcement, sentencing decisions
- Job recruitment
- …

Raise concerns regarding how much can/should we trust such systems?

Case Studies: Image Search

What images do people choose to represent careers?

E.g., percentage of images portraying women in image search for professions



Case Studies: Image Search

In search results [KMM15]:

- evidence for *stereotype exaggeration*
- systematic *underrepresentation of women* *(compared with the actual percentage as estimated by the US bureau of labor and statistics)*

- People rate search results *higher* when they are *consistent* with stereotypes for a career
- Shifting the representation of gender in image search results can *shift people's perceptions* about real-world distributions. (after search slight increase in their believes)

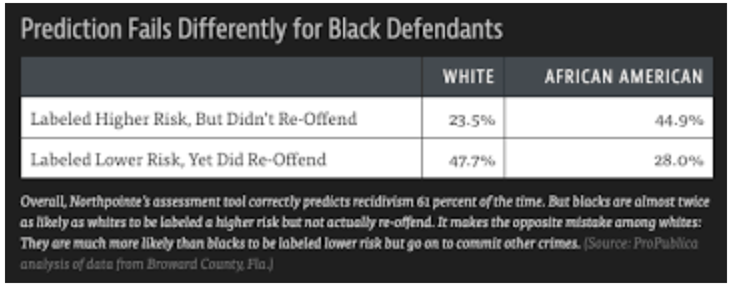
Case Studies: COMPAS

COMPAS (Correctional Offender Management Profiling for Alternative Sanctions): Commercial tool that uses a *risk assessment algorithm* to predict some categories of future crime

Used in courts in the US for bail and sentencing decisions

ProPublica found that

- the false positive rate (i.e., people labeled "high-risk" who did not re-offend) for African American defendants nearly twice as high as for White defendants
- Opposite for false negative rate



Prediction Fails Differently for Black Defendants

	WHITE	AFRICAN AMERICAN
Labeled Higher Risk, But Didn't Re-Offend	23.5%	44.9%
Labeled Lower Risk, Yet Did Re-Offend	47.7%	28.0%

Overall, Northpointe's assessment tool correctly predicts recidivism 61 percent of the time. But blacks are almost twice as likely as whites to be labeled a higher risk but not actually re-offend. It makes the opposite mistake among whites: They are much more likely than blacks to be labeled lower risk but go on to commit other crimes. (Source: ProPublica analysis of data from Broward County, Fla.)

The Wisconsin Supreme Court defended the use of COMPAS to inform criminal sentencing decisions

https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

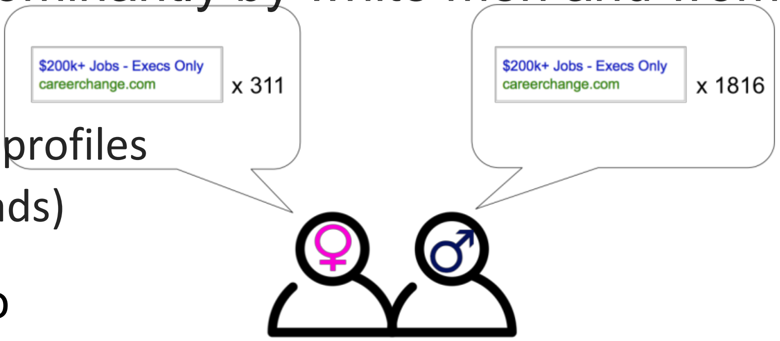
And many more


Ad related to latanya sweeney
Latanya Sweeney Truth
www.instantcheckmate.com/
Looking for Latanya Sweeney? Check Latanya Sweeney's Arrests.

Ads by Google
Latanya Sween
1) Enter Name an
Checks Instantly.
www.instantchecl
Latanya Sween
Public Records F
www.publicrecord
La Tanya
Search for La Tar
www.ask.com/La+Tanya

The importance of being Latanya [S13]

Names used predominantly by *men and women of color* are much more likely to generate *ads* related *to arrest records*, than names used predominantly by white men and women.



Adfisher: tool that automate the creation of *demographic* and *behavioral* profiles

- setting gender = female results in less ads for high-paying jobs (google ads)

In word embeddings: Man is to Computer Programmer as Woman is to Homemaker [BCZ+16]

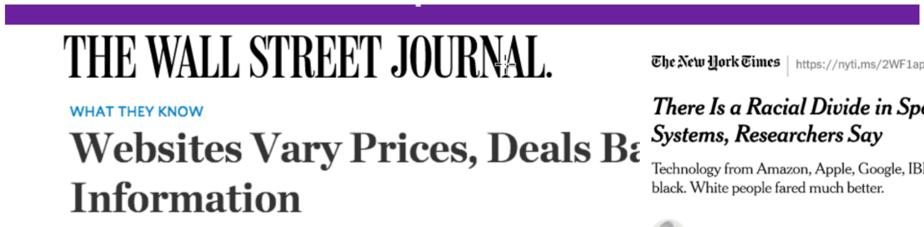

PMLR Proceedings of Machine Learning Research

Volume 81 All Volumes JMLR MLOSS FAQ Submission Format RSS

A case study of algorithm-assisted decision making in child maltreatment hotline screening decisions

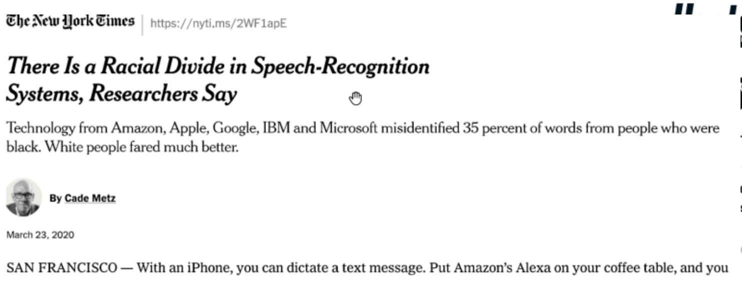
[edit]

Alexandra Chouldechova, Diana Benavides-Prado, Oleksandr Fialko, Rhema Vaithianathan ; Proceedings of the 1st Conference on Fairness, Accountability and Transparency, PMLR 81:134-148, 2018.


THE WALL STREET JOURNAL.

WHAT THEY KNOW

Websites Vary Prices, Deals Ba Information


The New York Times https://nyti.ms/2WF1apE

There Is a Racial Divide in Speech-Recognition Systems, Researchers Say

Technology from Amazon, Apple, Google, IBM and Microsoft misidentified 35 percent of words from people who were black. White people fared much better.

By Cade Metz

March 23, 2020

SAN FRANCISCO — With an iPhone, you can dictate a text message. Put Amazon's Alexa on your coffee table, and you


Can Airbnb Train
" 's Not to Be
:ts?

field experiment found that
e 16 percent less likely to be
s guests than whites.

rhardt Updated Jun. 12, 2019 12:47PM ET
Published Apr. 08, 2019 4:41AM ET

What is the cause: Data

- Correctness and completeness Garbage in, garbage out (GIGO)
  - Poorly selected
  - Incomplete
  - Incorrect
  - Outdated
  - Selected with bias
- Data as a social mirror: perpetuating and promoting historical biases
- Sample size disparity
  - learn on majority (Errors concentrated in the minority class)

What is the cause: Algorithms

- Algorithms as black boxes
- Output models that are hard to understand
- Unrealistic assumptions
- Algorithms that do not compensate for input data problems
- Decision making systems that assume correlation implies causation
- BIAS REINFORCEMENT CYCLE

Tutorial outline

PART 1 (this talk) (~10 min)
Motivation
Introduction to Fairness

PART 2 (~20 min)
Fairness in Ranking

PART 3 (~20 min)
Fairness in Recommenders

PART 4 (~10 min)
Fairness in Other Systems and Conclusions

Fairness

Definition

Fairness: lack of **discrimination**

Protected attributes: the output should not depend on the values of these attributes, differences *should* be explained by other attributes (features)

Two general approaches [DSV+12]

- Individual fairness
- Group fairness

Individual fairness

Similar people should be treated *similarly*

Similarity of ***individuals***

Let V be a set of individuals. Define a *task-specific distance metric* $d: V \times V \rightarrow R$ [DSV+12]

- Task-specific
- Expresses *ground truth* (or, best available approximation)
- Public, open to discussion and refinement
- Externally imposed, e.g., by a regulatory body, or externally proposed, e.g., by a civil rights organization

Individual fairness

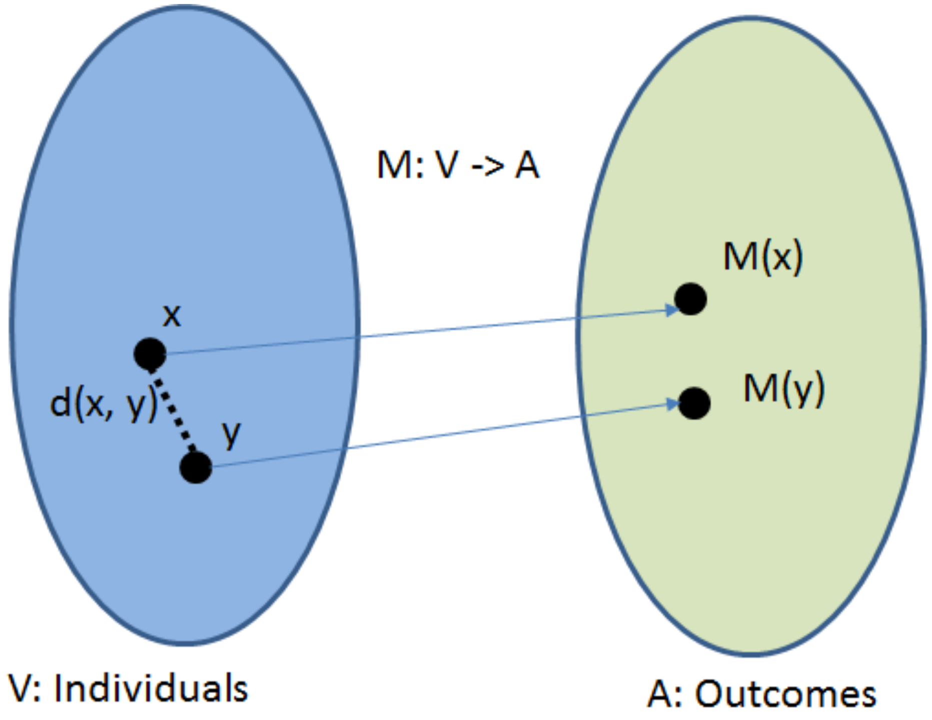
Similarity of ***treatment***

Depends on the algorithm

Assume a classifier *M* that maps individuals V to outcomes A

Randomized mapping from individuals to *probability distributions over outcomes*

- To classify $x \in V$, we choose an outcome $a \in A$ according to distribution $M(x)$



Lipschitz Mapping: a mapping M: V -> Δ(A) satisfies the (*D*, *d*)-Lipschitz property, if for every x, y ∈ V, $D\big(M(x) - M(y)\big) \leq d(x, y)$ where D is a distance measure between probability distributions

Group Fairness

Individuals divided into *groups* based on the value of one or more protected attribute

Assume one binary *protected attribute S* with 1 being the privileged value, two groups:

- Non protected (privileged) group, $S = 1$
- Protected (minority) group, $S \neq 1$

All groups should be treated *similarly*

Group Fairness in classification

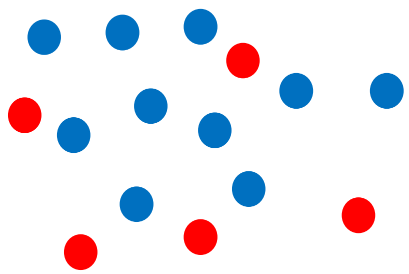
Similarity of ***treatment***

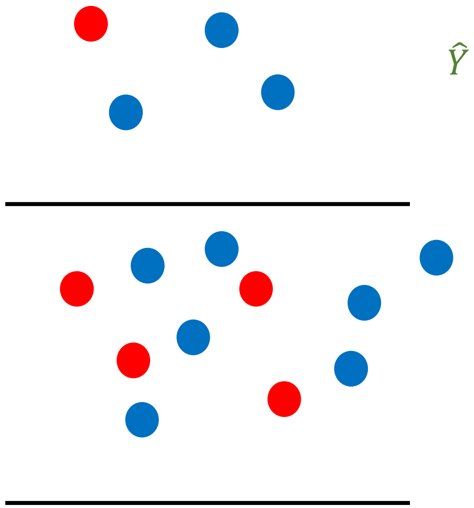
Depends on the algorithm

Binary outcome Y,
Predicted binary outcome $\hat{Y}$
yes the favorable outcome

Predicted Outcome

Dataset D





$\hat{Y}$ = Yes

$\hat{Y}$ = No

Color the protected attribute
Red the protected group

Is this output fair?

Blindness is not enough

Blindness (hiding the value of the protected attribute) does not work

Redundant encoding, (or, proxies) the protected attribute may be correlated with other attributes

Disparate treatment vs disparate impact

Disparate treatment

- Illegal practice of treating an entity differently based on a protected characteristic such as race, gender, age, religion, sexual orientation

Disparate impact

- Outcome depends on the protected attribute even if people are treated the same way

  Disparate impact doctrine solidified in the US after [Griggs v. Duke Power Co. 1971] where a high school diploma was required for unskilled work, excluding black applicants (non-job related training)

Discrimination Based on Redundant Encoding

Redlining: the practice of arbitrarily denying or limiting financial services to specific *neighborhoods*, generally because its residents are people of color or are poor.“, well-known form of discrimination based on redundant encoding. Illegal in the US

Non-discrimination and equality of opportunity

View on fairness

- Non-discrimination seeks to allocate resources in a way that does not consider irrelevant attributes

- Equality of opportunity seeks to correct a historical or present disadvantage for a group.

Group Fairness in classification

Basic types of group fairness, based on [FSV+19]

- Base rates
- Group-conditioned accuracy
- Group-conditions calibration

Group fairness: base rates

Compare the probability of a *favorable outcome for the non-protected group*

$$P\left[\hat{Y} = yes \,\middle|\, S = 1\right]$$

with the probability of a *favorable outcome for the protected group*

$$P\left[\hat{Y} = yes \,\middle|\, S \neq 1\right]$$

Both conditional probabilities evaluated on D.

Possible formulations:

Ratio [ZWS+13, FFM+15]

$$\frac{P\left[\hat{Y} = yes \,\middle|\, S \neq 1\right]}{P\left[\hat{Y} = yes \,\middle|\, S = 1\right]}$$

Difference [CV10]

$$1 - \left(P\left[\hat{Y} = yes \,\middle|\, S = 1\right] - P\left[\hat{Y} = yes \,\middle|\, S \neq 1\right]\right)$$

Group fairness: base rates

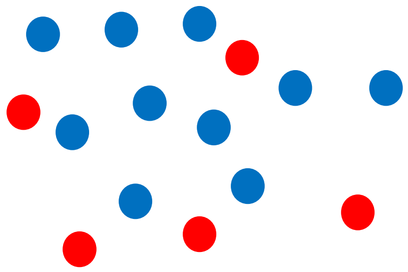
$$\frac{P[\hat{Y} = yes \mid S \neq 1]}{P[\hat{Y} = yes \mid S = 1]} = 1$$

demographic parity (statistical parity)

Preserves the input ratio: the *demographics of the individuals receiving a favorable outcome the same as demographics of the underlying population*

Group fairness: base rates

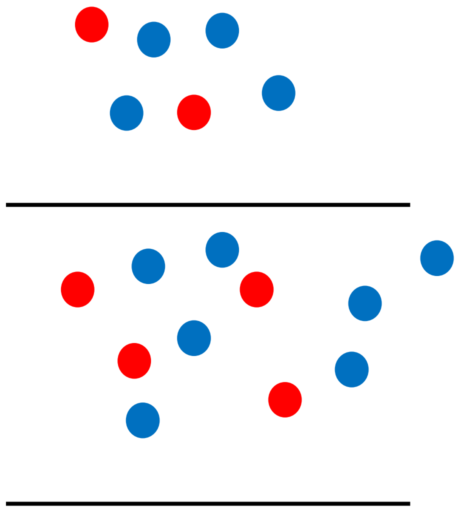
Dataset D



Red nodes $\frac{5}{15}$

Blue nodes $\frac{10}{15}$

Predicted Outcome



$\hat{Y}$ = Yes

$\hat{Y}$ = No

$P[\hat{Y} = yes \mid S = blue] = 4/10$

$P[\hat{Y} = yes \mid S = red] = 2/5$

$$\frac{P[\hat{Y} = yes \mid S = red]}{P[\hat{Y} = yes \mid S = blue]} = 1$$

Demographic parity

Group fairness: base rates

$$\frac{P[\hat{Y} = yes \mid S \neq 1]}{P[\hat{Y} = yes \mid S = 1]} \leq \tau$$

Disparate impact (unintended discrimination) [FFM+15], *τ = 0.8 based* on a generalization of the 80 percent rule advocated by the US Equal Employment Opportunity Commission

Group fairness: criticism

Ignores utility/goodness of the individuals in the group

Self-fulfilling prophecy

Deliberately *choosing the "wrong" members of the protected group* in order *to build a bad "track record"* for the group

Reverse tokenism

Deny access to a qualified member of the privileged group

Goal is to create convincing refutations

Other definitions of fairness

Group based

- *Classification-accuracy based ones*: Consider the performance of the classifier, for example whether the classification errors for each group are similar

- *Calibration-based ones:* Probabilistic classifiers: output the probability that an individual belongs to the positive class, probability estimates should be *well-calibr*ated for both groups (e.g, KMR17])

Counterfactual fairness [KLR+17]:
A decision is *fair towards an individual*, if it is the same in both the actual world and a counterfactual world where the individual belonged to a different demographic group. (using casual inference)

References

General

[KMM15] Matthew Kay, Cynthia Matuszek, and Sean A Munson. *Unequal representation and gender stereotypes in image search results for occupations.* CHI 2015

[BCZ+16] Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, Adam Tauman Kalai: *Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings*. NIPS 2016: 4349-4357

[S13] Latanya Sweeney *Discrimination in Online Ad Delivery*. Communications of the ACM, Vol. 56 No. 5, Pages 44-54.

[DTD15] Amit Datta, Michael Carl Tschantz, and Anupam Datta *Automated Experiments on Ad Privacy Settings*, Proceedings on Privacy Enhancing Technologies 2015; 2015 (1):92–112

[DHP+12] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, Richard S. Zemel: *Fairness through awareness*. ITCS 2012: 214-226

[FFM+15] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, Suresh Venkatasubramanian: *Certifying and Removing Disparate Impact*. KDD 2015

[KMR17] Jon M. Kleinberg, Sendhil Mullainathan, Manish Raghavan: *Inherent Trade-Offs in the Fair Determination of Risk Scores*. ITCS 2017: 43:1-43:23

[ZWS+13] Richard S. Zemel, Yu Wu, Kevin Swersky, Toniann Pitassi, Cynthia Dwork: *Learning Fair Representations*. ICML (3) 2013: 325-333

[FSV+19] Sorelle A. Friedler, Carlos Scheidegger, Suresh Venkatasubramanian, Sonam Choudhary, Evan P. Hamilton, Derek Roth: *A comparative study of fairness-enhancing interventions in machine learning*. FAT 2019: 329-338

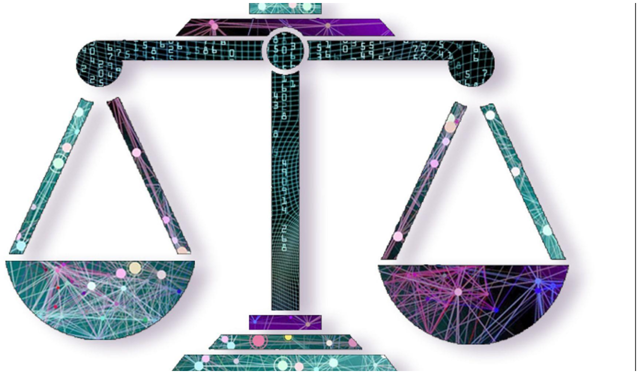
[CV+10] Toon Calders, Sicco Verwer: *Three naive Bayes approaches for discrimination-free classification*. Data Min. Knowl. Discov. 21(2): 277-292 (2010)

[KLR+17] Matt J. Kusner, Joshua R. Loftus, Chris Russell, Ricardo Silva: *Counterfactual Fairness*. NIPS 2017: 4066-4076

Fairness in Rankings and Recommenders

Konstantinos Stefanidis (TAU), Evaggelia Pitoura (UOI), Georgia Koutrika (ATHENA RC)

PART 2



EDBT/ICDT 2020 Joint Conference

30th March-2nd April, 2020
Copenhagen, Denmark

Fairness in Ranking

Fairness in ranking

- In many applications, the output is a *ranked list* where items are ordered in descending order of some *measure of the relative quality* of the items
  - E.g., Search engines, job search applications, News feeds, recommendations, etc

- Most often, the measure of quality, or the *utility* of an item, is the *relevance of the item* to the input query
- Commonly expressed with a relevance score, (or, pairwise preference relation)

Formally, given a set items $\{i_1, i_2, \ldots i_N\}$, a ranking is *an assignment (mapping) of* ***items to ranking positions***

Rank	ID	Group	Score
1	x299	blue	0.56
2	x78	red	0.55
3	x45	blue	0.45
4	x329	blue	0.44
5	x23	red	0.44
6	x981	blue	0.25
7	x665	blue	0.23
8	x724	blue	0.18
9	x87	red	0.16
10	x232	red	0.15

Fairness in ranking

- Position bias: People tend to "see" only few top results

Rank	ID	Group	Score
1	x299		0.56
2	x78		0.55
3	x45		0.45
4	x329		0.44
5	x23		0.44
6	x981		0.25
7	x665		0.23
8	x724		0.18
9	x87		0.16
10	x232		0.15

Fairness in ranking (in a nutshell):

- *Individual*: Items with similar relevance scores should receive similar "visibility"
- Group: All groups should receive similar "visibility"

Let us see how these notions have been formalized

Fairness constraints

Fairness constraints [CSV18]: Given a number of protected attributes, or, properties,
as *an upper bound* U_{lk} and *a lower bound* L_{lk} on the number of items with property l that are allowed to appear in the top k positions of the ranking

$L_{red\ 4}$ = *1: At least 1 item with property red in the top-4 positions*

Rank	ID	Group	Score
1	x299		0.56
2	x78		0.55
3	x45		0.45
4	x329		0.44
5	x23		0.44
6	x981		0.25
7	x665		0.23
8	x724		0.18
9	x87		0.16
10	x232		0.15

Discounted cumulative fairness

- Focus on the representation (i.e., number of items) of the protected group in *the top-p ranking positions* for various values of *p*.
- Set based

Metrics are inspired by Discounted Cumulative Gain (DCG) commonly used to evaluate the quality in information retrieval

DGC: Values are accumulated at discrete points in the ranking *with a logarithmic discount*

$$DCG_p(r) = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{log_2(i+1)}$$

Rank	ID	Group	Score
1	x299		0.56
2	x78		0.55
3	x45		0.45
4	x329		0.44
5	x23		0.44
6	x981		0.25
7	x665		0.23
8	x724		0.18
9	x87		0.16
10	x232		0.15

Normalized DCG (NDGC)

$$NDCG_p(r) = \frac{DCG_p(r)}{opt\_DCG_p}$$

$$DGC_4(\mathrm{r}) = 0.56 + \frac{0.55}{log_2(3)} + \frac{0.45}{log_2(4)} + \frac{0.44}{log_2(5)}$$

Discounted cumulative fairness

Let G^+ be the protected and G^- be the non protected group. Three metrics [YS17]

Normalized discounted difference (rND)

Accumulate the number of items belonging to the protected group at discrete positions in the ranking (e.g., p =10, 20, …) and discount these numbers according (it is better to have many protected items in higher positions)

$$rND(r) = \frac{1}{opt\_rND} \sum_{p=10,20,..}^{N} \frac{1}{log_2(p)} \left| \frac{|G^+_{1,..p}|}{p} - \frac{|G^+|}{N} \right|$$

	Rank	ID	Group	Score
	1	x299		0.56
	2	x78		0.55
	3	x45		0.45
	4	x329		0.44
p = 5	5	x23		0.44
	6	x981		0.25
	7	x665		0.23
	8	x724		0.18
	9	x87		0.16
p = 10	10	x232		0.15

$$\frac{1}{log_2(5)} \left| \frac{2}{5} - \frac{4}{10} \right| +$$

$$\frac{1}{log_2(10)} \left| \frac{4}{10} - \frac{4}{10} \right|$$

Discounted cumulative fairness

Normalized discounted difference (rND)

$$rND(r) = \frac{1}{opt\_rND} \sum_{p=10,20,..}^{N} \frac{1}{log_2(p)} \left| \frac{|G^+_{1,..p}|}{p} - \frac{|G^+|}{N} \right|$$

Normalized discounted ratio (rRD)

Again, we accumulate the number of items belonging to the protected group <u>at discrete positions</u> in the ranking (p =10, 20, …) and discount these accordingly, only difference in the denominator

$$rRD(r) = \frac{1}{opt\_rND} \sum_{p=10,20,..}^{N} \frac{1}{log_2(p)} \left| \frac{|G^+_{1,..p}|}{|G^-_{1,..p}|} - \frac{|G^+|}{|G^-|} \right|$$

Normalized discounted KL divergence (rKL)

use KL-divergence to compute the expectation of the difference between *the membership probability distribution of the protected group at top-p positions* (for p = 10, 20, ..) and in the *over-all population*

Fairness of exposure

Counting items at discrete positions does not fully capture the fact that:

minimal differences in relevance scores *may* translate into *large differences in visibility/exposure* for different groups because of *position bias* that results in a large skew in the distribution of exposure.

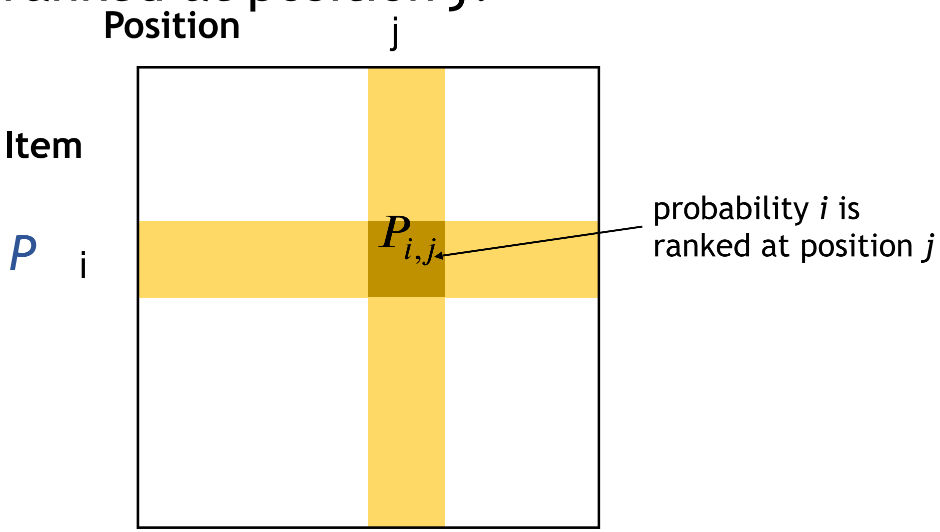
Fairness of exposure

Fairness of exposure [SJ18]

Position discount vector *v* to capture position bias

v_j represents the importance of position j (i.e., the fraction of users that examine an item at position j.)

Probabilistic ranking of N items in N positions modeled as ***a doubly stochastic NxN matrix P***, where $P_{i,j}$ is the probability that item *i* is ranked at position *j*.



Probabilistic ranking *P*

Position	v
1	0.42
2	0.28
..	..
j	0.08
..	..
N	0.00001

Importance of position *j*

Position discount vector *v*

Fairness of exposure

Item exposure

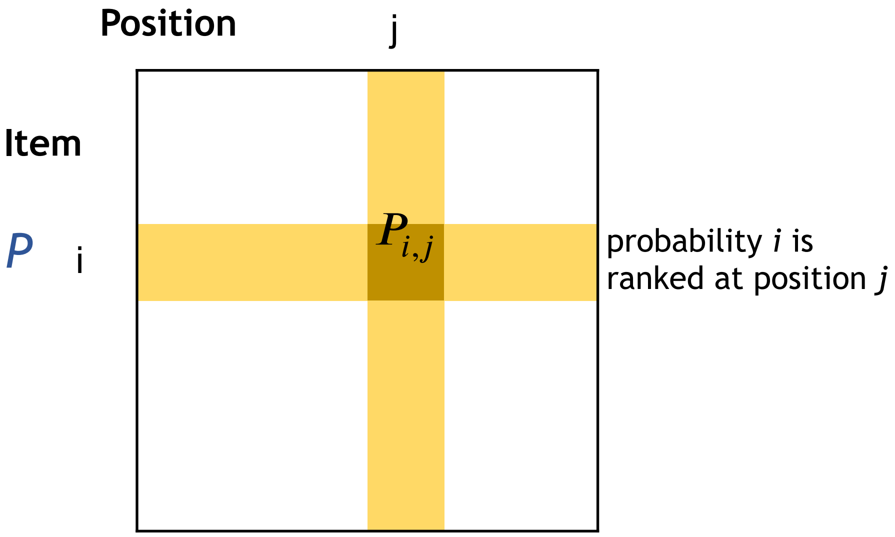
$$Exposure(i \mid P) = \sum_{j=1}^{N} P_{i,j} v_j$$

Position discount vector v

Position	v
1	0.42
2	0.28
..	..
j	0.08
..	..
N	0.00001

Importance of position j

Probabilistic ranking P



Group G_k exposure

$$Exposure\left(G_k \middle| P\right) = \frac{1}{|G_k|} \sum_{i \in G_k} Exposure(i \mid P)$$

Fairness of exposure

Demographic parity

- the two groups get the same average exposure

$$\frac{Exposure(G_o \mid P)}{Exposure(G_1 \mid P)} = 1$$

Disparate treatment

- the exposures (treatments) for the two groups are proportional to their average utility

$$\frac{Exposure(G_o \mid P)}{Utility(G_0 \mid q)} = \frac{Exposure(G_1 \mid P)}{Utility(G_1 \mid q)}$$

Disparate impact

- the impact (clickthrough rate (CTR) which depends on exposure and relevance) for the two groups are proportional to their average utility

$$\frac{CTR(G_o \mid P)}{Utility(G_0 \mid q)} = \frac{CTR(G_1 \mid P)}{Utility(G_1 \mid q)}$$

Equity of attention [BGW18]

Equity of attention: each item ***i*** receives attention ***a*** (i.e., views, clicks) that is proportional to its relevance ***rel*** in a given query

$$\frac{a_1}{rel_1} = \frac{a_2}{rel_2} \; \forall\, i_1, i_2$$

- An idea similar to fairness of exposure but for *individual items*

- Unlikely to be satisfied in *any single ranking*, since relevance scores are determined by the data and the query, while the attention is strongly influenced by position bias.
- If multiple items are *similarly relevant*, yet obviously cannot occupy the *same ranking position*

Idea: Consider a sequence $\rho^1, \rho^2, \ldots \rho^m$ of rankings and asks that an item receives cumulative attention proportional to its cumulative relevance

Equity of amortized attention [BGW18]

Equity of amortized attention: A sequence $\rho^1, \rho^2, \ldots \rho^m$ of rankings offers amortized equity of attention if each item receives cumulative attention proportional to its cumulative relevance, i.e.:

$$\frac{\sum_{l=1}^{m} a_1^l}{\sum_{l=1}^{m} rel_1^l} = \frac{\sum_{l=1}^{m} a_2^l}{\sum_{l=1}^{m} rel_2^l} \forall\ i_1, i_2$$

- allows to permute individual rankings so as to satisfy *fairness requirements over time*.

Unfairness: How much a sequence $\rho^1, \rho^2, \ldots \rho^m$ violates equity?
KL-divergence between the empirical distribution A of attention and the empirical distribution Rel of relevance

A_j Rel_j

$$unfairness(\rho^1, \rho^2, \ldots \rho^m) = \sum_{j=1}^{N} \left| \sum_{l=1}^{m} a_j^l - \sum_{l=1}^{m} rel_j^l \right|)$$

Definition of fairness in ranking (summary)

Set-based

Fairness constraints

Cumulative-based metrics

- normalized discounted difference
- Normalized discounted ratio
- Normalized discounted KL-divergence

All group-based

Exposure based

Group-based

- Demographic parity
- Disparate impact
- Disparate treatment

Individual

- Equity of attention

Amortized (over time)

- Amortized equity of attention

Achieving fairness

Methods for achieving fairness in ranking and in recommenders can be distinguished as:

Pre-processing: Transform the data so that any underlying bias or discrimination is removed

In-processing: modify existing or introduce new algorithms that result in fair rankings and recommendations

Post-processing: treat the algorithms for producing rankings and recommendations as black boxes and modify their output to ensure fairness

Pre-processing

Pre-processing

Generic techniques, we will come back to this in the recommender part of this tutorial

In-processing

In-processing

- Learning to rank
- Linear ranking function

In-processing: Learning to rank algorithms

- Learning to rank obtains a ranking function f that is learned by solving a minimization problem with respect to a *loss function* which most often is a measure of accuracy with respect to the training data

- Training data may be pair of items, item-scores, ranked lists

General approach: Extend the loss function by adding an *extra term* to ensure fairness

In-processing: extending the loss function in learning to rank

The DELTR approach [ZDC20]

Extends the ListNet learning to rank framework

- List-wise
- Training set: A query q and a *list of documents* ordered by their relevance to q
- Learn a ranking function f that minimizes a loss function L_{LN} that measures the extent to which the ordering $\hat{r}$ of documents induced by f for a query differs from the ordering r in which the documents appear in the training set for this query.

unfairness term

$$L_{DELTR}\left(r(q),\hat{r}(q)\right) = L_{LN}\left(r(q),\hat{r}(q)\right) + \gamma\, \mathrm{F}(\hat{r}(q))$$

- γ depends on desired trade-offs between ranking utility and fairness
- As a measurement of fairness democratic parity based on exposure is used

$$F\left(r(q)\right) = \max(0,\ exposure\left(G_0 \middle| P_{\hat{r}(q)}\right) - exposure\left(G_1 \middle| P_{\hat{r}(q)}\right))^2$$

- squared hinge loss: a differentiable loss function that prefers rankings in which the exposure of the protected group is not less than the exposure of the non protected group but not vice versa

In-processing: learning fair representations

Extend learning algorithm for fair classification [ZWS+13]
Basic idea:

- Introduce an intermediate level Z between the input space X that represents individuals and the output space Y that represents classification outcomes

X → Z → Y

Input: individuals — **Fair representation** — Classification output

Z: fair representation of X

- best encodes X and
- obfuscates any information about membership in the protected group

Z is a multinomial random variable of size k where each of the k values represents *a prototype (cluster)* in the space of X.

In-processing: learning fair representations

A learning system that minimizes the loss function

$$L = \underbrace{A_x L_x}_{\text{Quality of the encoding}} + \underbrace{A_z L_z}_{\text{Fairness}} + \underbrace{A_y L_y}_{\text{Accuracy}}$$

Quality of the encoding: Distance from points in X to their representation in Z should be small

Fairness: Statistical parity

Accuracy: Prediction based on the representation should be accurate

$$X \Rightarrow Z \Rightarrow Y$$

X: Input: individuals

Z: **Fair representation**

Y: Classification output

A_x, A_z , A_y hyper-parameters that control the trade-off among the three objectives

Statistical parity

$$P(z = k \mid x \in G^{+}) = P(z = k \mid x \in G^{-}) \ \forall\ k$$

The probability that a random element that belongs to the protected group of X maps to a particular prototype of Z is equal to the probability that a random element that belongs to the non-protected group of X maps to the same prototype

In-processing: learning fair representations

Modify the loss function to work for ranking [YS17]

$$L = \overbrace{A_x L_x}^{\text{Quality of the encoding}} + \overbrace{A_z L_z}^{\text{Fairness}} + \overbrace{A_y L_y}^{\text{Ranking accuracy}}$$

Distance from points in X to their representation in Z should be small

Statistical parity

Distance between the ground truth ranking and the estimated ranking should be small

$X \Rightarrow Z \Rightarrow Y$

Input: individuals

Fair representation

Ranking output

Distance used:

- average per-item score difference between the ground ruth ranking and the estimated ranking

Other:

- position accuracy (per-item rank difference),
- Kendall-τ distance, and
- Spearman and Pearson's correlation coefficients

In-processing: adjusting the weights in ranking functions [AJS19]

For each item *i*, d scoring attributes {i[1], i[2], …, i[d]}

Linear ranking functions that use a weigh vector $w = \{w_1, w_2, \ \ldots, \ w_d\}$ to compute a utility (goodness) score for each item

$$f(i) = \sum_{j=1}^{d} w_j i[j]$$

Given a function f with weights $w = \{w_1, w_2, \ \ldots, \ w_d\}$, find a function f^* with weight vector $w^* = \left\{w_1^*, \ w_2^*, \ \ldots, \ w_d^*\right\}$ s.t.

cos(w, w^*) is minimized, and

f^* is fair

Post-processing

Post-processing

- Generative process
- Constraint optimization

Post-processing: generative process

Input: a ranking and a fairness parameter f, $0 \leq f \leq 1$, that specifies the desired relative fairness of the two groups [YS17]
Output: a new ranking based on f

Start with an empty list
For each position j in the new ranking, perform a Bernoulli trial with probability f
If the trial *succeeds*,
the best available item *from the protected group* is selected;
else,
the best available item from the *non protected group* is selected.

$f = 1$	All items in the protected group precede all items in the non-protected group
$f = 0$	All items in the non-protected group precede all items in the protected group
f	Items in the protected group are preferred over items in the non-protected group
f	All items in the non-protected group are preferred over items in the protected group

Post-processing: generative process

Start with an empty list
For each position j in the new ranking, perform a Bernoulli trial with probability *f*
  If the trial *succeeds*,
    the best available item *from the protected group* is selected;
  else,
    the best available item from the *non-protected group* is selected.

Property: the relative order of two items that belong to the same group is not changed

Rank	ID	Group	Score
1	x299		0.56
2	x78		0.55
3	x45		0.45
4	x329		0.44
5	x23		0.44
6	x981		0.25
7	x665		0.23
8	x724		0.18
9	x87		0.16
10	x232		0.15

Rank	ID	Group	Score
1	x78		0.55
2	x23		0.44
3	x87		0.16
4	x232		0.15
5	x299		0.56
6	x45		0.45
7	x329		0.44
8	x981		0.25
9	x665		0.23
10	x724		0.18

f = 1

Rank	ID	Group	Score
1	x78		0.55
2	x299		0.56
3	x23		0.44
4	x45		0.45
5	x87		0.16
6	x329		0.44
7	x232		0.15
8	x981		0.25
9	x665		0.23
10	x724		0.18

f > 0.5

Post-processing: generative process

Fair* presents a statistical test for this generative model that given a ranking determines the probability that the ranking was generated by the model [ZBC+17]:

Given that at a specific position we have seen a specific number of items from each group, a one-tailed Binomial test is used to compare the null hypotheses that *the ranking was generated using the model with parameter $f_* = f$, or with $f_* < f$,* which would mean that the protected group is represented less than desired.

Post-processing: Constraint optimization problem

Many variants

Given a query *q*, a utility definition $U(r|q)$ of a ranking r and a fair ranking definition, find ranking r that

$$r = argmax_r U(r|q)$$

s.t. r is fair

If unfairness measure instead of condition

Given a query *q*, a utility definition $U(r|q)$ of a ranking r and a fair ranking measure F, produce a ranking $\hat{r}$ such that that:

$$\hat{r} = argmax_{\hat{r}} F\left(\hat{r} \middle| q\right)$$

s.t.

$$\text{distance}\left(U\left(\hat{r} \middle| q\right), U(r, q)\right) \leq \theta$$

Post-processing: LP optimization [SJ18]

Given utility vector *u*, position importance vector *v*, find probabilistic ranking *P*

$$P = argmax_p u^t P v$$

$$\text{s.t.} \quad \left.\begin{aligned} 1^T P &= 1^T \\ P1 &= 1 \\ 0 \le P_{i,j} &\le 1 \end{aligned}\right\} \text{P is a doubly stochastic matrix}$$

P is fair

Post-processing: Constraint optimization (amortized fairness [BGW18])

Amortized individual fairness

Offline version

Given a ranking sequence $\rho^1, \rho^2, \ldots \rho^m$, produce a ranking sequence $\rho^{1*}, \rho^{2*}, \ldots \rho^{m*}$ so as to *minimize unfairness* subject to a *constraint in utility (quality) loss*

$$\text{minimize} \quad \sum_{i=1}^{N} |A_i - Rel_i|$$

$$\text{subject to} \quad \frac{NDCG(\rho^j)}{NDCG(\rho^{j*})} \geq \theta \quad \forall\, j$$

Post-processing: Constraint optimization (amortized fairness [BGW18])

Online version

Given the ranking sequence $\rho^1, \rho^2, \ldots \rho^{l-1}$, seen so far, reorder the current ranking ρ^l so as to *minimize the unfairness seen so far* subject to a *constraint in utility (quality) loss of the current ranking*

$$\text{minimize} \quad \sum_{i=1}^{N} \left| \left(A_i^{l-1} + a_i^l\right) - \left(Rel_i^{l-1} + rel_i^l\right) \right|$$

$$\text{subject to} \quad \frac{NDCG(\rho^l)}{NDCG(\rho^{l*})} \geq \theta$$

Use *Integer Linear Programming (ILP)* to solve the online optimization problem:

Introduce N^2 decision variables $X_{i,j}$ set to 1 if item i is assigned to the ranking position j, and 0 otherwise.

Post-processing: Constraint optimization (ranking maximization [CSV18])

Extend the following ranking maximization problem

Given m items, n ranking positions, $n \ll m$, and values W_{ij} of placing item i in ranking position j,
Find an assignment of the items to each of the m position, such that the total value is maximized

Equivalent to maximum weigh matching

Post-processing: Constraint optimization (ranking maximization [CSV18])

Fairness constraints as s *an* *upper bound* U_{lk} and *a* *lower bound* L_{lk} on the number of items with property *l* that are allowed to appear in the top *k* positions of the ranking

Constrained ranking maximization problem: Let the *n x m* assignment matrix *X* with $X_{i,j}$ set to 1 if item i is assigned to the ranking position j, and 0 otherwise.

$$X = argmax_X \sum_{i=1}^{n} \sum_{j=1}^{m} W_{i,j} X_{i,j}$$

s.t. *X* satisfies all fairness constraints

- Hardness
- Approximation algorithms

Ensuring fairness in ranking (summary)

Approaches depend both on the
- Definition of fairness
- Ranking algorithm

In-processing
Learning to rank
- Extent the objective function
- Introduce fair representations

Linear preference functions
- Adjust the weights

Post-processing
Generative process
Constraint optimization problem

Fair vs diverse rankings

Diversity in ranking different objectives [DJP+17, PTF+17]

- Cover different user intents as well address query ambiguity
- Make results more informative, interesting and engaging by avoiding redundancy, support serendipity and novelty

References

Ranking

[YS17] Ke Yang, Julia Stoyanovich: *Measuring Fairness in Ranked Outputs*. SSDBM 2017: 22:1-22:6

[ZWS+13] Richard S. Zemel, Yu Wu, Kevin Swersky, Toniann Pitassi, Cynthia Dwork: *Learning Fair Representations*. ICML (3) 2013: 325-333

[AJS+19] Abolfazl Asudeh, H. V. Jagadish, Julia Stoyanovich, Gautam Das: *Designing Fair Ranking Schemes*. SIGMOD Conference 2019: 1259-1276

[ZDC20] Meike Zehlike, Gina-Theresa Diehn, Carlos Castillo, *Reducing Disparate Exposure in Ranking: A Learning to Rank Approach*, WWW 2020

[ZBC+17] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, Ricardo Baeza-Yates: *FA*IR: A Fair Top-k Ranking Algorithm*. CIKM 2017: 1569-1578

[SJ18] Ashudeep Singh, Thorsten Joachims: *Fairness of Exposure in Rankings*. KDD 2018: 2219-2228

[BGW18] Asia J. Biega, Krishna P. Gummadi, Gerhard Weikum: *Equity of Attention: Amortizing Individual Fairness in Rankings*. SIGIR 2018: 405-414

[C18] Carlos Castillo: *Fairness and Transparency in Ranking*. SIGIR Forum 52(2): 64-71 (2018)

[CSV18] L. Elisa Celis, Damian Straszak, Nisheeth K. Vishnoi: *Ranking with Fairness Constraints*. ICALP 2018: 28:1-28:15

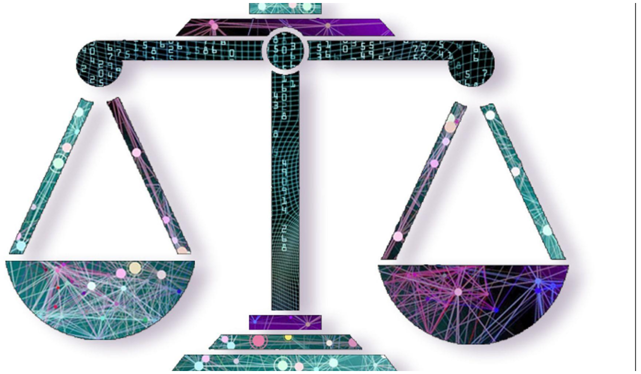
[DJP+17] Marina Drosou, HV Jagadish, Evaggelia Pitoura, Julia Stoyanovich: *Diversity in big data: A review,* Big data 5 (2), 73-84 (2017)

[PTF+17} Evaggelia Pitoura, Panayiotis Tsaparas, Giorgos Flouris, Irini Fundulaki, Panagiotis Papadakos, Serge Abiteboul, Gerhard Weikum: On Measuring Bias in Online Information. SIGMOD Rec. 46(4): 16-21 (2017)

Fairness in Rankings and Recommenders

Konstantinos Stefanidis (TAU), Evaggelia Pitoura (UOI), Georgia Koutrika (ATHENA RC)

PART III



EDBT ICDT COPENHAGEN 2020

EDBT/ICDT 2020 Joint Conference
30th March-2nd April, 2020
Copenhagen, Denmark

Konstantinos Stefanidis (TAU)

Fairness in Recommenders

Konstantinos
Stefanidis (TAU)

Multi-sided Fairness

Recommendations for different stakeholders:

[B17,TP+19]

- Consumers of recommendations
  - Recommenders care only for consumers fairness
    - A credit card company recommending consumer credit offers - No producer-side fairness issues since the products are coming from the same bank
- Providers/producers of data items to be recommended
- *System owners*
- *Regulators/auditors*
  - Decision making for data scientists, ML researchers, policymakers and governmental auditors

Stakeholders have a varying level of familiarity and expertise with the system and the underlying technologies

Multi-sided Fairness in Recommenders

Providers/producers of data items to be recommended

- Fairness needs to be preserved for the providers only

Example:
Interest in ensuring market diversity and avoiding monopoly domination

- Online craft marketplace Etsy: the system wishes to ensure that new entrants to the market get a reasonable share of recommendations even though they have fewer shoppers than established vendors

Etsy

Consumers vs Producers fairness:
Producers fairness is passive - Producers do not seek out recommendation opportunities but rather wait for users to come to the system and request recommendations

Multi-sided Fairness in Recommenders

Can a recommender requires fairness for both consumers and providers?

Consider any domain in which both consumers and providers can belong to protected groups

- A rental property recommender
  - The recommender may treat minority applicants as a protected class and wish to ensure that they are recommended properties similar to white renters
  - The recommender may wish to treat minority landlords as a protected class and ensure that highly-qualified tenants are referred to them at the same rate as to white landlords
- Employment scenario

Ensuring Fairness in Recommenders

Ensuring Fairness in Recommenders

Fairness methods: Methods for achieving fairness in recommendations can be distinguished between:

- *Pre-processing*
  - Target at transforming the data so that any underlying bias or discrimination is removed
- *In-processing*
  - Target at modifying existing or introducing new algorithms that result in fair recommendations, e.g., by removing bias
- *Post-processing*
  - Treat the algorithms for producing recommendations as black boxes
  - To ensure fairness, modify the output of the algorithm

Pre-processing Methods

Pre-processing methods modify the input to the recommender:

- Sampling [CD+16]
- Re-weighting [KC11]
  - Generate weights for the training examples in each (group, label) combination differently, to ensure fairness before classification

Pre-processing Methods

- Representation learning
  - Learn a probabilistic transformation that edits the attributes and labels in the data with group fairness, individual distortion, and data fidelity constraints and objectives [CW+17]
  - Find a latent representation that encodes the data well but makes unclear information about protected attributes [ZW+13]
- Disparate impact remover
  - Edit attribute so that the marginal distributions based on the subsets of an attribute with a given sensitive value are all equal [FF+15]
  - Database repair [SR+19]
- Antidote data
  - Add more data to the input of the recommender to improve fairness with minimum accuracy loss [RG+19]

In-processing Methods

In-processing methods design fairness-aware algorithms, that is, algorithms that produce fair recommendations. E.g.:

- Use matrix factorization [YH17]
- Alter the objective of the algorithm to emphasize fairness, typically by adding regularization [KA+18, KA+18b]
- Incorporate randomness in variational autoencoders recommenders [BS19]

The STEM Example

Recommendation in education in science, technology, engineering, and mathematics topics - STEM

- 2010 - Women accounted for only 18% of the bachelor’s degrees awarded in CS
- The underrepresentation of women causes historical rating data of CS courses to be dominated by men
- The learned model may underestimate women’s preferences and be biased toward men
- If the ratings provided by students accurately reflect their true preferences, the bias in which ratings are reported leads to unfairness

The STEM Example

Two forms of underrepresentation

- Population imbalance: different types of users occur in the dataset with different frequencies
  - Significantly fewer women succeed in STEM than those who do not; however more men succeed in STEM than those who do not

- Observation bias: certain types of users may have different tendencies to rate different types of items
  - Women are rarely recommended to take STEM courses, there may be significantly less training data about women in STEM courses

USE MF & Count Fairness

[YH17]

Value unfairness: Count inconsistency in estimation errors across the user types

- When one class of users is given higher or lower predictions than their true preferences
  - Male students are recommended STEM courses when they are not interested in STEM, while female students not being recommended even if they are interested

Absolute unfairness: Count inconsistency in absolute estimation error across user types

- A single statistic representing the quality of prediction for each user type
  - If female students are given predictions 0.5 points below their true preferences and male students are given predictions 0.5 points above their true preferences, there is no absolute unfairness
    - One type of user has the unfair advantage of good recommendation, while the other user type has poor recommendation

USE MF & Count Fairness

[YH17]

Underestimation unfairness: Count inconsistency in how much the predictions underestimate the true ratings

- Missing recommendations are more critical than extra recommendations
  - A top student is not recommended to explore a topic he/she would excel in

Overestimation unfairness: Count inconsistency in how much the predictions overestimate the true ratings

- Users may be overwhelmed by recommendations, so providing too many recommendations would be especially detrimental → big evaluation time

Non-parity unfairness: Count the absolute difference between the overall average ratings of disadvantaged users and those of advantaged users

USE MF & Count Fairness

[YH17]

Traditionally, the matrix-factorization targets at minimizing a regularized, squared reconstruction error

The above fairness metrics are used to augment the learning objective of MF, by helping reducing discontinuities in the objective, making optimization more efficient

The Regularization Approach

[KA+18, KA+18b]

Random variables X for users, Y for items and R for recommendation outcomes

Standard recommendations

In addition: sensitive feature S, i.e., information to be ignored in the recommendation process (e.g., user's gender, or item's popularity)

<u>Standard Recommendations</u>	→	<u>Independence-enhanced recommendations</u>
Dataset: D = {(xi, yi, ri)}	→	Dataset: D = {(xi, yi, ri, si)}
Prediction function: r(x, y)	→	Prediction function: r(x, y, s)

The goal is to achieve: Recommendation (or statistical) independence

- No information about a sensitive feature influences the outcome
- Recommendations are selected so as to satisfy a recommendation independence constraint

The Regularization Approach

Adopting a regularizer imposing a constraint of independence while training a recommendation model

$\Sigma_D \mathrm{loss}(r_i, r(x_i, y_i, s_i)) - \eta \mathrm{ind}(R, S) + \lambda \mathrm{reg}(\Theta)$

- loss: empirical loss
- η: independence parameter - control the balance between independence and accuracy
- ind: independence term - a regularizer to constrain independence
  - The larger value indicates that recommendation outcomes and sensitive values are more independent
- λ: regularization parameter
- Θ: L2 regularizer

The Regularization Approach

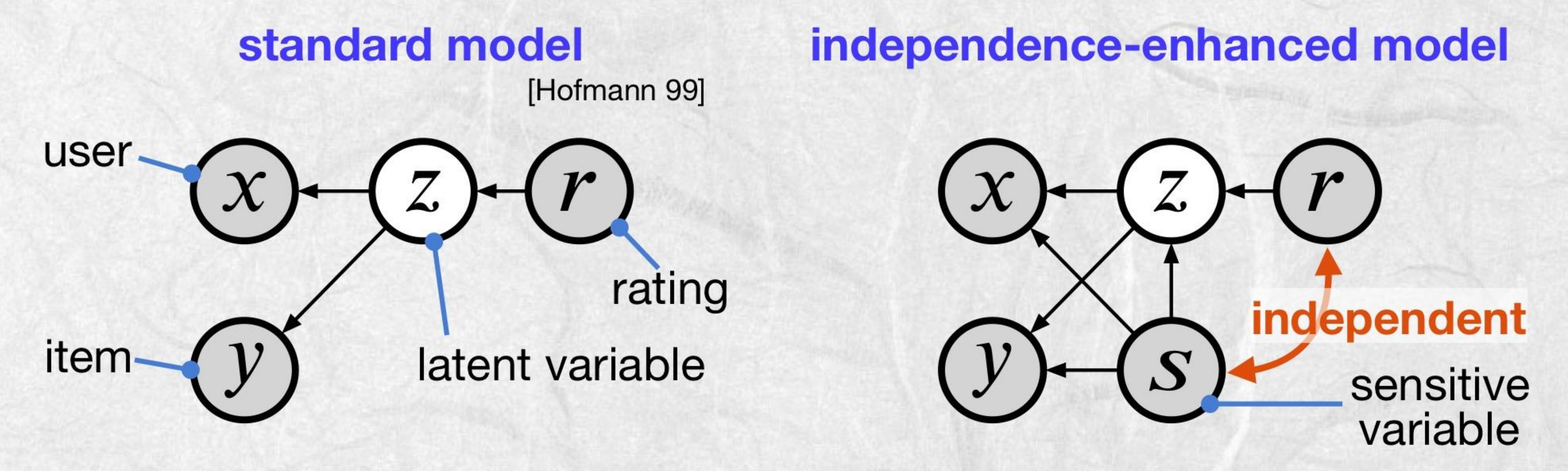
Several alternatives for the independence term

The regularizer to constrain independence

- Mutual information with histogram models
- Mean matching
  - Matching means of predicted ratings for distinct sensitive groups
- Mutual information with normal distributions
- Distribution matching with Bhattacharyya distance

The Regularization Approach

[KA+18, KA+18b]

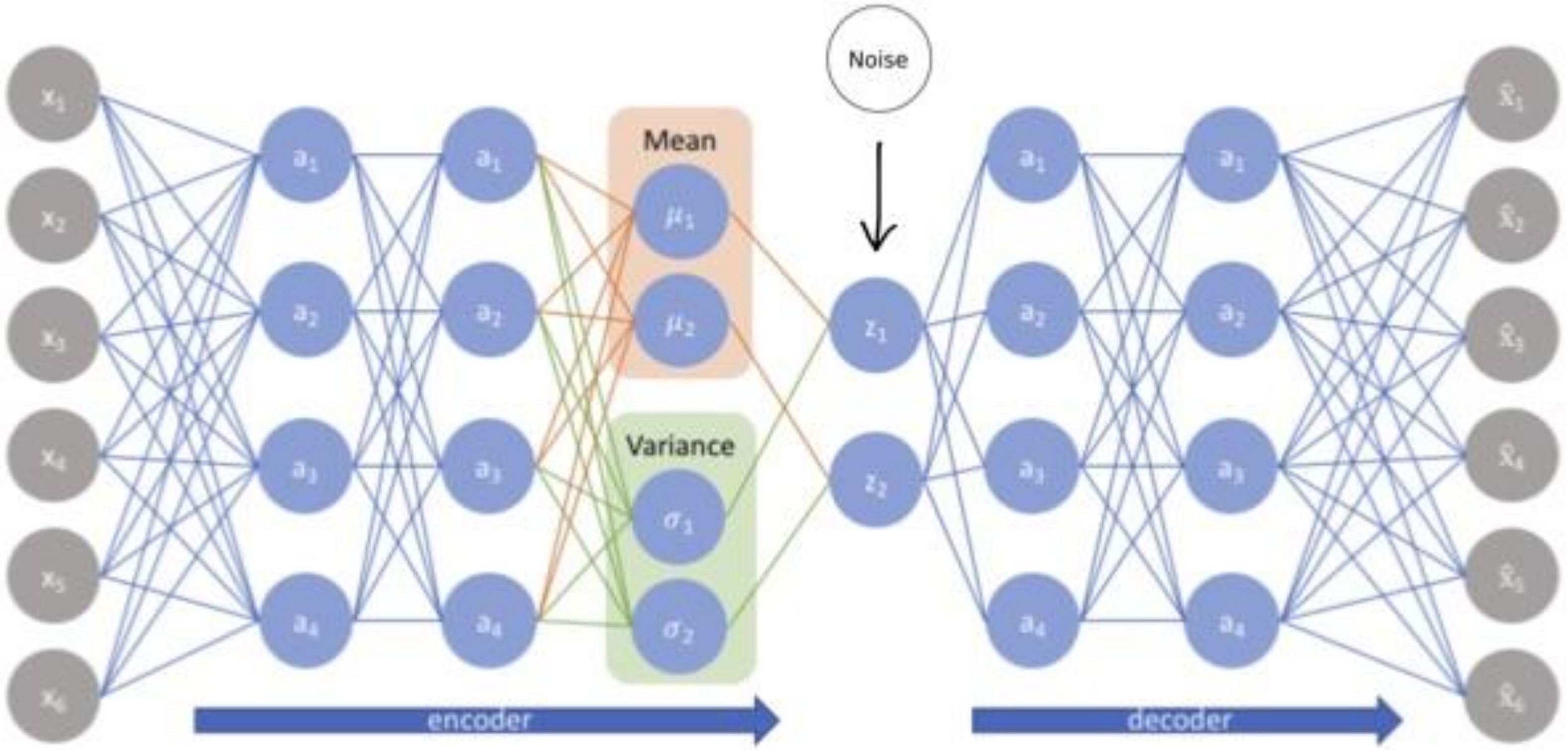


A sensitive variable is added to a recommendation model so that it satisfies an independence constraint

Randomness in VAE Recommenders

Encoder: The input is mapped to a latent space (normal distributions) through hidden layers

Sampling Phase: Samples are drawn from the the distributions propagate to decoder



Decoder: The estimated output is compared with labels and propagates back

Explore the probability distribution learned in the training phase for varying ranking position in a collaborative manner

[BS19]

Post-processing Methods

Post-processing methods modify the output of the recommender algorithms to ensure fairness:

- Calibrated recommendations [S18]

Calibration Method

Results are fair if they achieve fair representation

- Results are evenly balanced, reflect population, user historical data

Re-ranking, aka *post-processing*

$I^* = \text{argmax}_I \ (1-\lambda)s(I) - \lambda C_{KL}(p, q(I))$

- λ determines the trade-off between accuracy and calibration
- $s(I)$: the summation of the predicted relevance recommendation scores
- C_{KL}: Kullback-Leibler divergence, i.e., *how similar are p and q?*

[S18]

Post-Processing Methods:
Fairness in Group Recommenders

Fairness in Group Recommendations

Typically, recommenders provide suggestions adapted to the preferences of a single user

However, many times, the recommended data items are consumed by a group of users

- A travel with friends
- A movie to watch with the family during Christmas holidays
- Music to be played in a car for the passengers

But: users in a group may be heterogeneous

- People with potentially different interests and preferences

Fairness in Group Recommendations

Most works on group recommenders aim to maximize the group's overall satisfaction with the recommended list

This way, there could be one or more users that do not like the items in the list

- By using the average method, the opinion of some users can be lost

Need for fair group recommendations!

Intuitively: fairness attempts to minimize the feeling of dissatisfaction within group members

Individual Utility, Social Welfare & Fairness

Assume a measure of quantifying the satisfaction, or utility, of a user (in a group) given a list of recommendations

- How relevant the K recommended items are to the user

Group utility, or social welfare: ways for averaging user utilities

Fairness: the balance of user utilities inside the group, i.e., fairness can be the minimum user utility

- *Intuitively, a list that minimizes the dissatisfaction of any user in the group can be considered as the most fair*

In this sense, fairness enforces the least misery principle among users utilities

[XM+17]

Individual Utility

Assume a user u in a group g and a set of items I (|I| = K) recommended to g

The individual utility U(u, I) : U×I → [0, 1] of the relevances rel(u, i), where i ∈ I, is defined as:

$$\text{(1) Average: } U(u, I) = \frac{1}{K \times rel_{max}} \sum_{i \in I} rel(u, i)$$

$$\text{(2) Proportionality: } U(u, I) = \frac{\sum_{i \in I} rel(u, i)}{\sum_{i \in I(u, K)} rel(u, i)}$$

I(u,K) denotes the set of items which are among the top-K favourite items of user u

Social Welfare & Fairness

Aggregate individual utilities as social welfare

The Social Welfare SW(g,I), is the overall utility of all users in g given group recommendations I

$$SW(g, I) = \frac{1}{|g|} \sum_{u \in g} U(u, I), \forall g, I$$

Fairness reflects the comparison between the utilities of users in the group

$$Least\ Misery: \ F_{LM}(g, I) = \min\{U(u, I), \forall u \in g\}$$

$$Variance: \ F_{Var}(g, I) = 1 - Var(\{U(u, I), \forall u \in g\})$$

$$Jain's\ Fairness: \ F_J(g, I) = \frac{(\sum_{u \in g} U(u, I))^2}{|U| \cdot \sum_{u \in g} U(u, I)^2}$$

$$Min - Max\ Ratio: \ F_M(g, I) = \frac{\min\{U(u, I), \forall u \in g\}}{\max\{U(u, I), \forall u \in g\}}$$

Ensuring Fairness

Maximize social welfare and fairness

Use the following scheme to assign weights to each objective:

$$\lambda \cdot SW\ (g, I) + (1 - \lambda) \cdot F\ (g, I)$$

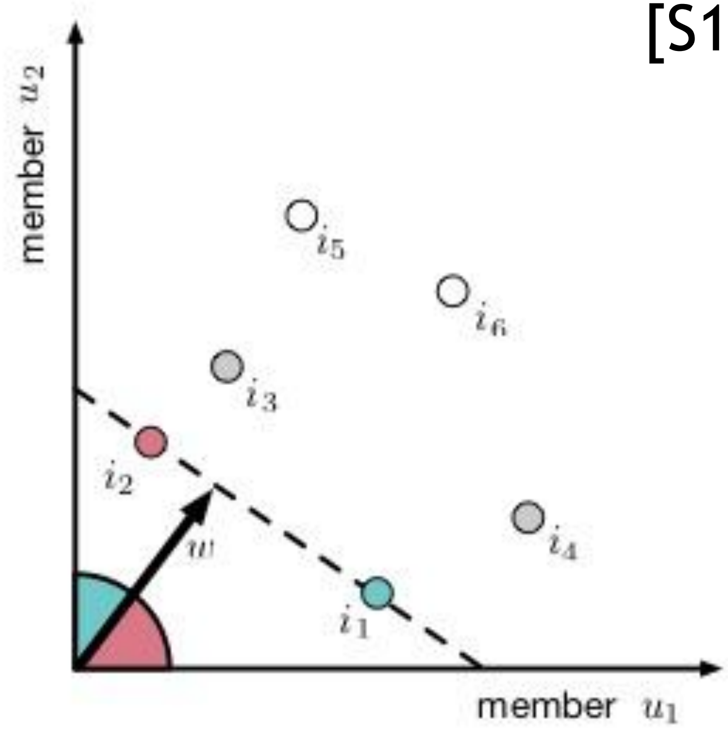
Greedy algorithm: Select an item that achieves the highest fairness (above function) when it is added to the current recommendation list

- Time-efficient, because of one item per round

Alternatives via integer programming techniques

Fairness via Pareto

[S19]



Items in space: each dimension corresponds to a group member u and its coordinate equals the rank rel(u,i) of the item i for u

Top-6 for u1: i2, i3, i5, i1, i6, i4, and for u2: i1, i4, i2, i3, i6, i5

- Item i1 ranks 4th for u2 and 1st for u1, and is thus represented by the point (4,1)
- E.g., i1 is clearly better than another i4

We say that i dominates i′ for a group g, if for each user, item i ranks at least as good as i′, and there exists at least one user for whom i ranks better:

$$\forall u \in g : rel(u,i) \leq rel(u,i'), \text{ and } \exists u' \in g : rel(u',i) < rel(u',i')$$

Fairness via Pareto

The top items not dominated by any other item are called Pareto optimal

- Items i1 and i2 comprise the set of Pareto optimal items in the example

N-level Pareto optimal: contain items dominated by at most $N - 1$ other items

- Thus, the top-N choices are within the N-level Pareto optimal set
    - E.g., i3 is 2-level Pareto optimal as it is dominated by only i2

Ensuring Fairness

Impractical to identify the exact set of N-level Pareto optimal items

- It needs the ranks of each item to each user

Approximation:

- Request top-N′ recommendations for each user in the group, and take their union
  - N′>N is the largest number of items the system can recommend
- Identify the N-level Pareto optimal items among the N′ ones

m-Proportionality

Package-to-group recommendations

For a user u and a package P, P is m-proportional to u, if there exist at least m items in P that u likes

For a group g, the m-proportionality of P for g is defined as:

$$|g_P| \ / \ |g|$$

where g_P is the set of users in g for which P is m-proportional

[SQ17]

m-Envy-Freeness

Package-to-group recommendations

A user u in g is envy-free for an item i in P, if rel(u,i) is in the top-Δ% of the preferences in the set {rel(v,i) : v $\in$ g}

A package P is m-envy-free for u, if u is envy-free for at least m items in P

For a group of users g and a package P, the m-envy-freeness of P for g is defined as:

$$|g_{ef}| \ / \ |g|$$

where g_{ef} is the set of users in g for which P is m-envy-free

[SQ17]

Ensuring Fairness

Fairness maximization

Construct P greedily

- In rounds, add to P the item that satisfies the largest number of non-satisfied users
  - Maximize: $f_G(P,i) = |Sat_G(P \cup \{i\}) \setminus Sat_G(P)|$, at each round

    where $Sat_G(P)$ denotes the users satisfied by P

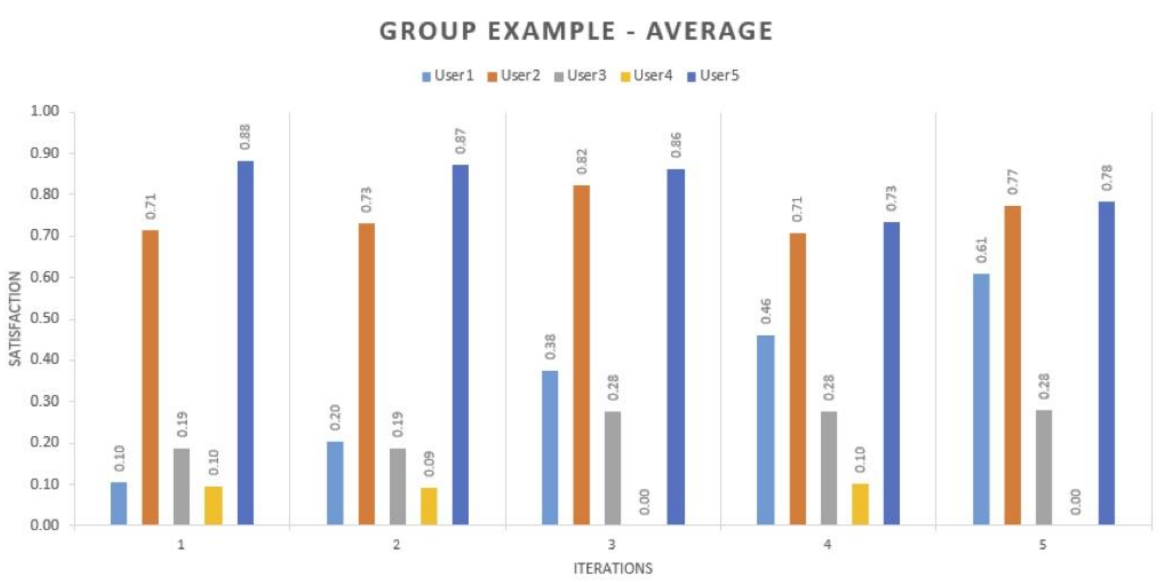
With category constraints:

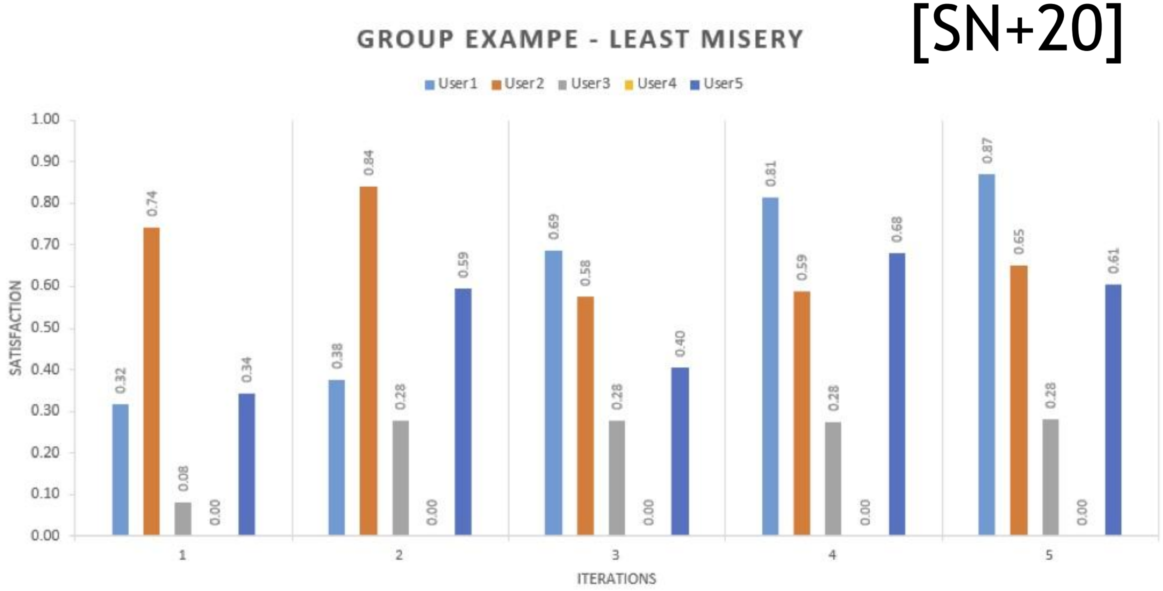
- When selecting an item from a specific category, we remove the items of this category from the candidate set

With distance constraints:

- Consider as candidate items only the items that when added to the existing solution satisfy the distance constraints

(*Un*)Fairness in Sequential Recommendations





[SN+20]

5 friends // watch a movie // top-10 // 5 iterations

Count satisfaction for each member: *How relevant are the group list's items, over the best items for each group member*

- **<u>User 4</u>** has a low satisfaction score: almost no interesting recommendations

The recommender is unfair to him/her - unfairness continues throughout the 5 iterations

Satisfaction & Disagreements

Satisfaction per iteration: directly compare the user's satisfaction from the group recommendations with the ideal case for that user

- $p_j(u_i, d_z)$: preference score of ui for item d_z at iteration j

$$sat(u_i, Gr_j) = \frac{GroupListSat(u_i, Gr_j)}{UserListSat(u_i, A_{u_i,j})}$$

$$GroupListSat(u_i, Gr_j) = \sum_{d_z \in Gr_j} p_j(u_i, d_z)$$

$$UserListSat(u_i, A_{u_i,j}) = \sum_{d_z \in A_{u_i,j}} p_j(u_i, d_z)$$

Average for group satisfaction

Disagreements in the group: difference in the satisfaction scores between the most satisfied and the least satisfied user in the group

Fairness in Sequential Recommendations

Sequential hybrid aggregation method

A weighted combination of the avg and min aggregations

$$score(G, d_z, j) =$$
$$(1 - \alpha_j) * avgScore(G, d_z, j) + \alpha_j * leastScore(G, d_z, j)$$

Dynamic α in each iteration

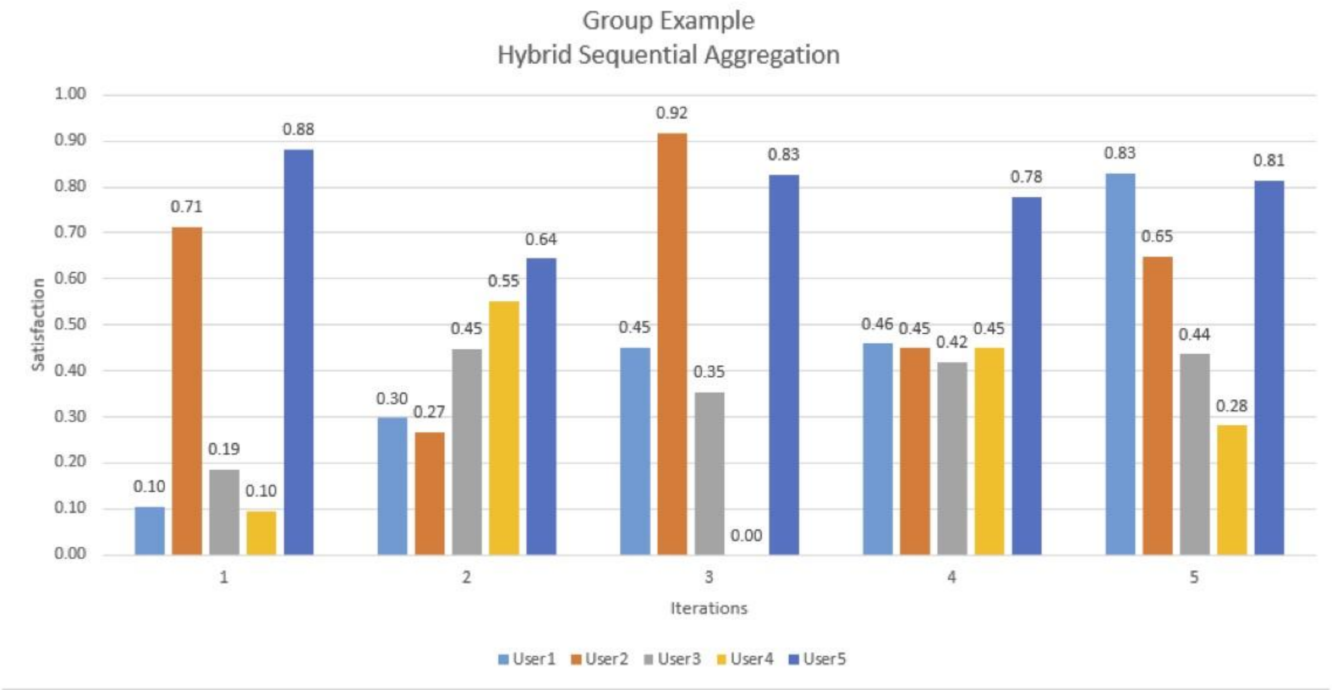
Subtract the min satisfaction score of the group members in the previous iteration from the max score

$$\alpha_j = max_{u \in G} sat(u, Gr_{j-1}) - min_{u \in G} sat(u, Gr_{j-1})$$

- For an extremely unsatisfied user in a previous iteration
  - α takes a high value and promotes that user's preferences
- For equally satisfied users at the last round
  - α takes low values, use a close to the avg aggregation, everyone is treated as an equal

Fairness in Sequential Recommendations

A group member that was not satisfied in the previous iteration, is satisfied in the next



User 4: In the first iteration has a low satisfaction score, and in the second has a higher one

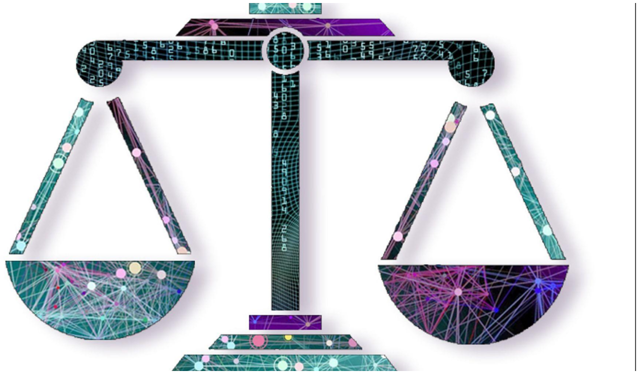
- Improvement over the previous results, where User 4 was always the least satisfied member of the group

References

Fairness in Recommenders

[CD+16] L.Elisa Celis, Amit Deshpande, Tarun Kathuria and Nisheeth K. Vishnoi. 2016. How to be Fair and Diverse? CoRR abs/1610.07183 (2016).
[SR+19] Babak Salimi, Luke Rodriguez, Bill Howe, and Dan Suciu. 2019. Interventional Fairness: Causal Database Repair for Algorithmic Fairness. In SIGMOD.
[S18] Harald Steck. 2018. Calibrated recommendations. In RecSys. 154–162.
[YH17] Sirui Yao and Bert Huang. 2017. Beyond Parity: Fairness Objectives for Collaborative Filtering. In NIPS. 2921–2930.
[BS19] Rodrigo Borges and Kostas Stefanidis. 2019. Enhancing Long Term Fairness in Recommendations with Variational Autoencoders. In MEDES.
[XM+17] Lin Xiao, Zhang Min, Zhang Yongfeng, Gu Zhaoquan, Liu Yiqun and Ma Shaoping. 2017. Fairness-Aware Group Recommendation with Pareto-Efficiency. In RecSys.
[S19] Dimitris Sacharidis. 2019. Top-N Group Recommendations with Fairness. In SAC.
[SQ17] Dimitris Serbos, Shuyao Qi, Nikos Mamoulis, Evaggelia Pitoura and Panayiotis Tsaparas. 2017. Fairness in Package-to-Group Recommendations. In WWW.
[SN+20] Maria Stratigi, Jyrki Nummenmaa, Evaggelia Pitoura and Kostas Stefanidis. 2020. Fair Sequential Group Recommendations. In ACM SAC.
[KA+18b] T. Kamishima, S. Akaho, H. Asoh and J. Sakuma. 2018. Recommendation independence. In FAT.
[B17] Robin Burke. 2017. Multisided Fairness for Recommendation. CoRR abs/1707.00093
[KC11] Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. Knowl. Inf. Syst. 33(1): 1-33 (2011)
[CW+17] F. du Pin Calmon, D. Wei, B. Vinzamuri, K. N. Ramamurthy and K. R. Varshney. 2017. Optimized Pre-Processing for Discrimination Prevention. In NIPS.
[ZW+13] R. S. Zemel, Y. Wu, K. Swersky, T. Pitassi and C. Dwork. 2012. Learning Fair Representations. In ICML.
[RG+19] B. Rastegarpanah, K. P. Gummadi and M. Crovella. 2019. Fighting Fire with Fire: Using Antidote Data to Improve Polarization and Fairness of Recommender Systems. In WSDM.
[FF+15] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger and S. Venkatasubramanian. 2015. Certifying and Removing Disparate Impact. In KDD.
[TP+19] V. Tsintzou, E. Pitoura and P. Tsaparas. 2019. Bias Disparity in Recommendation Systems. In RMSE.

Fairness in Rankings and Recommenders



Konstantinos Stefanidis (TAU), Evaggelia Pitoura (UOI), Georgia Koutrika (ATHENA RC)

PART IV

EDBT/ICDT 2020 Joint Conference

30th March-2nd April, 2020
Copenhagen, Denmark

Fairness as a Program Property

Program Fairness

- Fairness Verification
- Fairness-Aware Programming

Program Fairness Verification

- Is a given program P fair, under some definition of fairness?
- How fair/unfair is P?

The goal is to

- **analyze** a given decision-making program and
- **construct a proof** of its fairness or unfairness

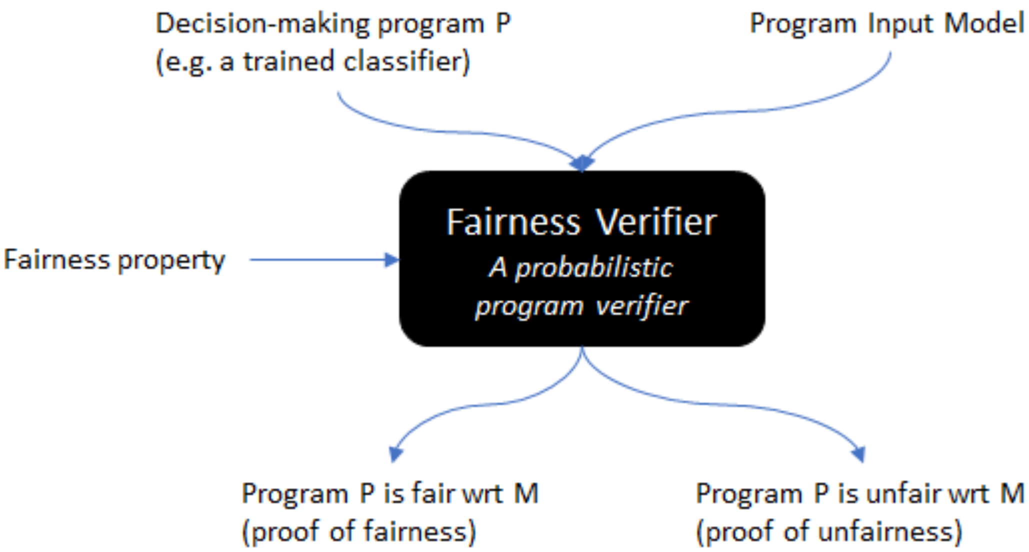
Program Fairness Verification

Challenges

- What **class of decision-making programs** can our program model capture?
- How can we define **the set of possible inputs** to the program in a way that is useful and amenable to verification?
- How can we **describe what a fair program** is?
- How can we **fully automate** the verification process?

[CG20] [KM+20][AV19][GB+17]

Program Fairness Verification



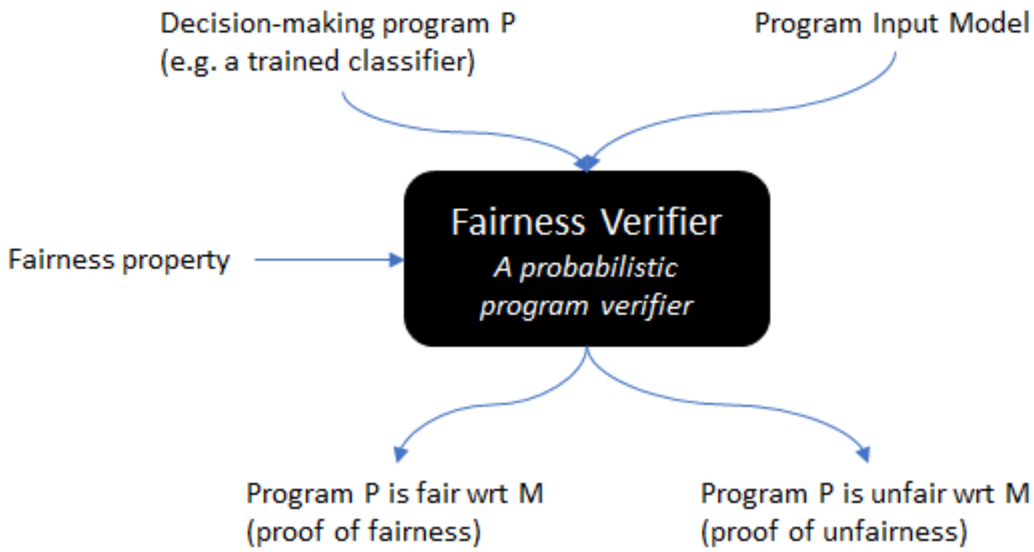
1. **Modeling Input of the Program**
   - Dataset
   - Population Model M (a probabilistic program)

1. **Fairness Properties**
   There are many ways to define when and why a program is fair
   or unfair.
   An example of group fairness:

$$\frac{\Pr[\mathcal{P}(v) = \mathit{true} \mid v_s = m]}{\Pr[\mathcal{P}(v) = \mathit{true} \mid v_s \neq m]} > 1 - \epsilon$$

i.e., *the algorithm is just as likely to hire a minority applicant (m) as it is for other, non-minority applicants*

Program Fairness Verification



3. **Proving Fairness**

- For simple definitions, such as group fairness, the verification problem reduces to computing the probability of a number of events with respect to the program and the population model.

- For more complex definitions, such as individual fairness, proving fairness requires more complex reasoning involving multiple runs of the programs (a notoriously hard problem).

Additionally, producing a human readable proof might be challenging

Fairness-Aware Programming

Make fairness a first-class concern in programming.

- Developers can state **fairness expectations natively in their code**
- A runtime system monitors decision-making and reports violations of fairness.

- This approach is analogous to the notion of assertions.
- However, detecting the violation of fairness assertions cannot be done through a single execution

[AV19]

Fairness-Aware Programming

Example: movie recommendation

Train a recommender that, given a user profile, recommends a single movie, for simplicity.

Goal: Ensuring that male users are not isolated from movies with a strong female lead.

@spec(pr(femaleLead(r)|s = male) > 0.2)

The above specification ensures that for male users, the procedure recommends a movie with a female lead at least 20% of the time.

Fairness-Aware Programming

Runtime analysis

- To determine that a procedure f satisfies a fairness specification φ, we need to **maintain statistics** over the inputs and outputs of f as it is being applied.
- We **compile** the specification φ **into runtime monitoring code** that executes every time f is applied, storing aggregate results of every probability event appearing in φ.

For example:

@spec(pr(femaleLead(r)|s = male) > 0.2)

Here, the monitoring code would maintain the number of times the procedure returned true for a movie with a female lead.

Fairness-Aware Programming

Runtime analysis

- To determine that a procedure f satisfies a fairness specification φ, we need to **maintain statistics** over the inputs and outputs of f as it is being applied.
- We **compile** the specification φ **into runtime monitoring code** that executes every time f is applied, storing aggregate results of every probability event appearing in φ.

Challenge:
In the case of individual fairness, the runtime system has to **remember all decisions made** explicitly, so as to compare new decisions with past ones.

Fairness: Beyond Ranking and Recommenders

Some examples

- Cache allocation in multi-tenant environments (e.g., SPARK) [KF+17]
- Multiple resource allocation [GZ+11]
- Scheduling [GM+09]

Fairness in resource allocation

Desirable properties:

1. **Sharing incentive**: Each user should be better off in the shared allocation setting than she would expect from simply having access to all of the resources with probability 1/N, where N the number of users.

1. **Pareto efficiency**: It should not be possible to increase the allocation of a user without decreasing the allocation of at least another user. This property is important as it leads to maximizing system utilization subject to satisfying the other properties.

1. **Strategy-proofness**: Users should not be able to benefit by lying about their resource demands. This provides incentive compatibility, as a user cannot improve her allocation by lying.

1. **Envy-freeness:** A user should not prefer the allocation of another user. This property embodies the notion of fairness.

Example: ROBUS

ROBUS

Cache allocation that can speed up a multi-tenant workload while guaranteeing fairness in terms of the tenants' performance

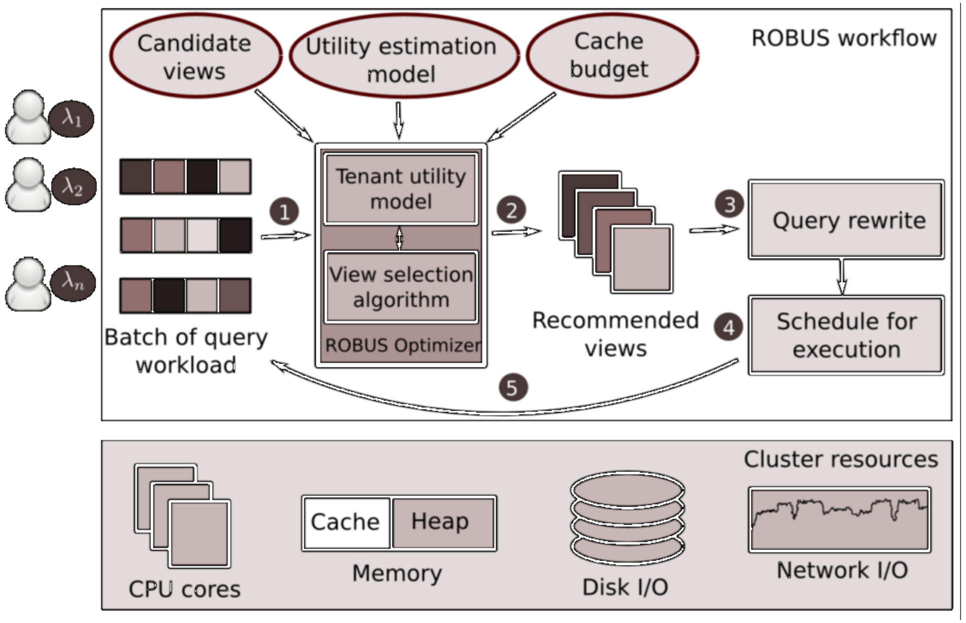
Fairness Model

- **Pareto Efficiency**: An allocation is Pareto-efficient if no other allocation simultaneously improves the expected utility of at least one tenant and does not decrease the expected utility of any tenant.
- **Sharing Incentive**: For N tenants, each tenant should expect higher utility in the shared allocation setting than she would expect from simply having access to all of the resources with probability 1/N

[KF+17]

Example: ROBUS

prototype implemented on Spark



- Queries submitted by tenants to queues are processed in batches of a fixed time interval.
- Queries within a batch are optimized together and are scheduled for execution at the same time.

Conclusions

1. Many different fairness definitions.

- How do fairness definitions fare?
- Which one is suitable for which context?
- How do people perceive fairness in different contexts?

[SH+20][HR+18][PR+17]

Conclusions

2. Different approaches at different stages
(pre-processing, in-processing, post-processing, verification)

- Which one fairs better when?
- What combinations of methods would work best?

Conclusions

3. Applying fairness in practice

- What are the challenges (and hopes) ?
- How to combine fair desiderata with other optimization objectives?
- How to evaluate ?

Case: Online A/B tests in LinkedIn Talent Search of applying a fair framework for achieving representative ranking showed tremendous improvement in the fairness metrics (nearly three fold increase in the number of search queries with representative results) without statistically significant change in the business metrics, which paved the way for deployment to 100% of LinkedIn Recruiter users worldwide.

[GA+19]

“Fair” has many meanings

thank you

References

[AV19] Aws Albarghouthi, Samuel Vinitsky: Fairness-Aware Programming. FAT 2019: 211-219

[AA+16] Aws Albarghouthi, Loris D'Antoni, Samuel Drews, Aditya V. Nori: Fairness as a Program Property. CoRR abs/1610.06067 (2016)

[GB+17] Sainyam Galhotra, Yuriy Brun, Alexandra Meliou: Fairness testing: testing software for discrimination. ESEC/SIGSOFT FSE 2017: 498-510

[CG20] Efrén Cruz Cortés, Debashis Ghosh. An Invitation to System-wide Algorithmic Fairness. AIES '20: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society

[KM+20] Nathan Kallus, Xiaojie Mao, Angela Zhou. Assessing algorithmic fairness with unobserved protected class using data combination. FAT* '20: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency

[AS19] Serge Joseph Abiteboul, Julia Stoyanovich. Transparency, Fairness, Data Protection, Neutrality: Data Management Challenges in the Face of New Regulation. Journal of Data and Information Quality, June 2019 Article No.: 15

[Wi20] Maranke Wieringa. What to account for when accounting for algorithms: a systematic literature review on algorithmic accountability. FAT* '20: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency

[SH+20] Nripsuta Saxena, Karen Huang, Evan DeFilippis, Goran Radanovic, David Parkes, Yang Liu. How Do Fairness Definitions Fare? Examining Public Attitudes Towards Algorithmic Definitions of Fairness. AIES '19: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society

[HR+18] Grgi´c-Hlaca, N.; Redmiles, E. M.; Gummadi, K. P.; and Weller, A. 2018. Human perceptions of fairness in algorithmic decision making: A case study of criminal risk prediction. arXiv preprint arXiv:1802.09548.

[PR+17] Plane, A. C.; Redmiles, E. M.; Mazurek, M. L.; and Tschantz, M. C. 2017. Exploring user perceptions of discrimination in online targeted advertising. In USENIX Security.

[GA+19] Sahin Cem Geyik, Stuart Ambler, Krishnaram Kenthapadi. Fairness-Aware Ranking in Search & Recommendation Systems with Application to LinkedIn Talent Search. KDD 2019

[KF+17] Mayuresh Kunjir, Brandon Fain, Kamesh Munagala, Shivnath Babu: ROBUS: Fair Cache Allocation for Data-parallel Workloads. SIGMOD Conference 2017: 219-234

[GZ+11] Ali Ghodsi, Matei Zaharia, Benjamin Hindman, Andy Konwinski, Scott Shenker, Ion Stoica. Dominant Resource Fairness: Fair Allocation of Multiple Resource Types NDSI 2011

[GM+09] Chetan Gupta, Abhay Mehta, Song Wang, Umesh Dayal. Fair, effective, efficient and differentiated scheduling in an enterprise data warehouse. EDBT 2009