Human-Centered Machine Learning: Interventions

Dong Nguyen

2024

Utrecht University

Today

Last time: Measuring fairness

Today

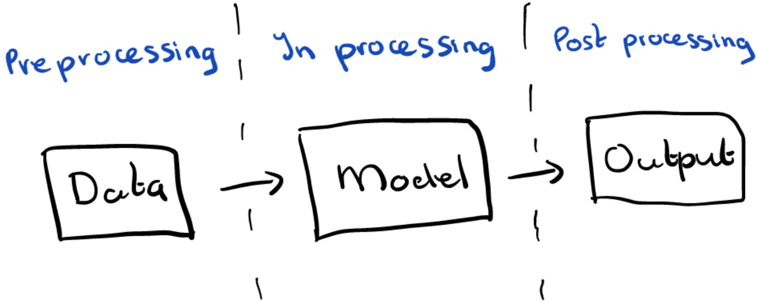
Last time: Measuring fairness

Today:
Making ML systems more fair

- Pre-processing
- Post-processing
- In-processing

Taking stock, outlook

Interventions



Pre-processing

Pre-processing

- Pre-processing
- In-processing
- Post-processing

Pre-process the data **before** training a classifier.

- Early in the pipeline.
- You can then apply a range of black box classifiers.
- More control when releasing a dataset.

But: no direct control on final outcome and *we're changing the data*.

Pre-processing

- Pre-processing
- In-processing
- Post-processing

Pre-process the data **before** training a classifier.

- Early in the pipeline.
- You can then apply a range of black box classifiers.
- More control when releasing a dataset.

But: no direct control on final outcome and *we're changing the data*.

Today: data augmentation, reweighing

Pre-processing: data augmentation

Task: **co-reference resolution (NLP)**

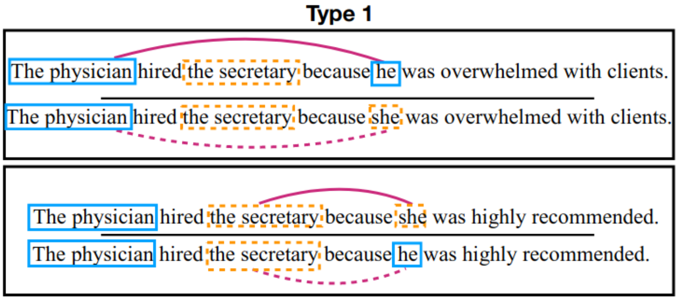


Figure: From Fig 1 from Zhao et al., 2018

Gender Bias in Coreference Resolution: Evaluation and Debiasing Methods, Zhao et al. NAACL 2018 [pdf]

Pre-processing: data augmentation

Task: **co-reference resolution (NLP)**

Zhao et al. found that pronouns are linked to occupations more accurately in pro-stereotypical conditions than in anti-stereotypical conditions.

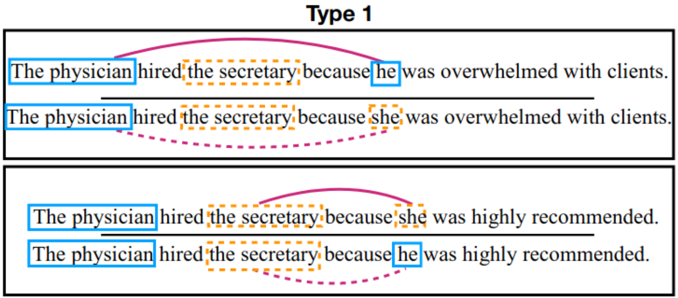


Figure: From Fig 1 from Zhao et al., 2018

Gender Bias in Coreference Resolution: Evaluation and Debiasing Methods, Zhao et al. NAACL 2018 [pdf]

Pre-processing: data augmentation

Data augmentation: Increase the amount of training data using synthetic data, e.g. using slightly changed instances of your existing data.

Here: Create additional data using manually specified rules by replacing male entities with female entities (and vice versa).
For example: $she \rightarrow he$, $Mr \rightarrow Mrs$.

Gender Bias in Coreference Resolution: Evaluation and Debiasing Methods, Zhao et al. NAACL 2018 [pdf]

Pre-processing: Reweighing

Recall: Let A and B be two random variables. If they are independent, then their joint probability is $P(A, B) = P(A)P(B)$.

Suppose we have:

- A sensitive attribute A: *a* and *b*.
- A binary outcome Y: – and +.

If A and Y are independent, then:

$$P(A = a \wedge Y = +) = P(A = a)P(Y = +)$$

Data preprocessing techniques for classification without discrimination,
Kamiran and Calders, Knowl Inf Syst 2012 [link]

Pre-processing: Reweighing

Recall: Let A and B be two random variables. If they are independent, then their joint probability is $P(A, B) = P(A)P(B)$.

Suppose we have:

- A sensitive attribute A: *a* and *b*.
- A binary outcome Y: – and +.

If A and Y are independent, then:

Which fairness criterion does this correspond to?

$$P(A = a \wedge Y = +) = P(A = a)P(Y = +)$$

Data preprocessing techniques for classification without discrimination, Kamiran and Calders, Knowl Inf Syst 2012 [link]

Pre-processing: Reweighing

Reweight instances with $A = b$ and $Y = +$ as follows:
(same holds for other cases.)

$$W(X) = \frac{P_{exp}(A = b \wedge Y = +)}{P_{obs}(A = b \wedge Y = +)}$$

where P_{exp} is the expected probability if A and Y are independent.

Data preprocessing techniques for classification without discrimination,
Kamiran and Calders, Knowl Inf Syst 2012 [link]

Pre-processing: Reweighing

Sex	Ethnicity	Highest degree	Job type	Class
M	Native	H. school	Board	+
M	Native	Univ.	Board	+
M	Native	H. school	Board	+
M	Non-nat.	H. school	Healthcare	+
M	Non-nat.	Univ.	Healthcare	−
F	Non-nat.	Univ.	Education	−
F	Native	H. school	Education	−
F	Native	None	Healthcare	+
F	Non-nat.	Univ.	Education	−
F	Native	H. school	Board	+

Figure: Table 1 from Kamiran and Calders, 2012

$P_{exp}(A = f \wedge Y = +) =$
$0.5 \times 0.6 = 0.3$

But we have:
$P_{obs}(A = f \wedge Y = +) = 0.2,$

So females with a + outcome will be weighted with: $0.3/0.2 = 1.5$

Data preprocessing techniques for classification without discrimination, Kamiran and Calders, Knowl Inf Syst 2012 [link]

Pre-processing: Reweighing

Sex	Ethnicity	Highest degree	Job type	Class
M	Native	H. school	Board	+
M	Native	Univ.	Board	+
M	Native	H. school	Board	+
M	Non-nat.	H. school	Healthcare	+
M	Non-nat.	Univ.	Healthcare	–
F	Non-nat.	Univ.	Education	–
F	Native	H. school	Education	–
F	Native	None	Healthcare	+
F	Non-nat.	Univ.	Education	–
F	Native	H. school	Board	+

Figure: Table 1 from Kamiran and Calders, 2012

$P_{exp}(A = f \wedge Y = +) =$
$0.5 \times 0.6 = 0.3$

But we have:
$P_{obs}(A = f \wedge Y = +) = 0.2,$

So females with a + outcome will be weighted with: $0.3/0.2 = 1.5$

What weight will males with a + outcome receive?

Pre-processing: Reweighing

Sex	Ethnicity	Highest degree	Job type	Class
M	Native	H. school	Board	+
M	Native	Univ.	Board	+
M	Native	H. school	Board	+
M	Non-nat.	H. school	Healthcare	+
M	Non-nat.	Univ.	Healthcare	−
F	Non-nat.	Univ.	Education	−
F	Native	H. school	Education	−
F	Native	None	Healthcare	+
F	Non-nat.	Univ.	Education	−
F	Native	H. school	Board	+

Figure: Table 1 from Kamiran and Calders, 2012

$P_{exp}(A = f \wedge Y = +) =$
$0.5 \times 0.6 = 0.3$

But we have:
$P_{obs}(A = f \wedge Y = +) = 0.2$,

So females with a + outcome will be weighted with: $0.3/0.2 = 1.5$

What weight will males with a + outcome receive?

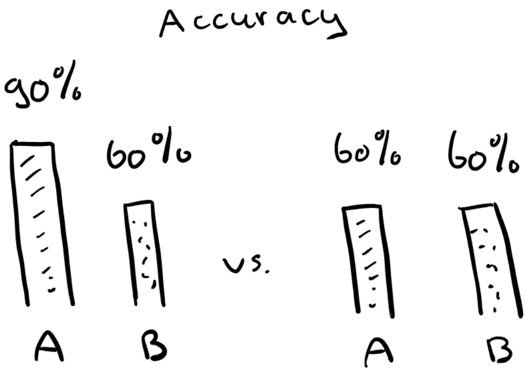
0.75

Pre-processing: Reweighing

- We can apply this idea directly when the classifier can work with weights.
- Alternative: resample the data to mimic weights.

Post-processing

Post-processing



We have two groups (A and B) and two classifiers (left and right)

Which classifier would you prefer: Left or right?

Post-processing

- Pre-processing
- In-processing
- Post-processing

Post-process the predictions **after** training a classifier.

- We may not be able to change the data and/or the model itself. We only have the final output (e.g., intellectual property, black box).
- We can directly control outcome distribution.
- We need access to the protected group attributes.

recap!

Equal decision measures

$A \in \{a, b\}$ sensitive attribute; D is the decision

$$A \perp D$$

In a binary classification scenario (e.g., $D = 1$ means hire this person):

$$P[D = 1 | A = a] = P[D = 1 | A = b]$$

Equal decision measures

We have two groups (A1 and A2). When we have a **score** function R we can make a decision by setting a threshold t: $D = \mathbb{1}\{R > t\}$.

A	R
A	0.9
A	0.8
A	0.7
A	0.2
B	0.6
B	0.5
B	0.4
B	0.5

Equal decision measures

We have two groups (A1 and A2). When we have a **score** function R we can make a decision by setting a threshold t: $D = \mathbb{1}\{R > t\}$.

A	R
A	0.9
A	0.8
A	0.7
A	0.2
B	0.6
B	0.5
B	0.4
B	0.5

Suppose we want to have equal acceptance rates, e.g. 25%. We could use group-specific thresholds! (But yes, we then need to treat groups differently.)

recap!

Conditional on outcome

True positive rates/recall **(equal opportunity)**:

$$P[D = 1|Y = 1, A = a] = P[D = 1|Y = 1, A = b]$$

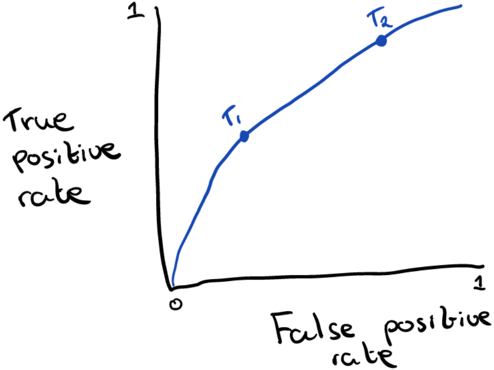
False positive rates:

$$P[D = 1|Y = 0, A = a] = P[D = 1|Y = 0, A = b]$$

Both constraints: **equalized odds**

A=sensitive attribute; D=decision; Y=target variable/outcome

ROC curves

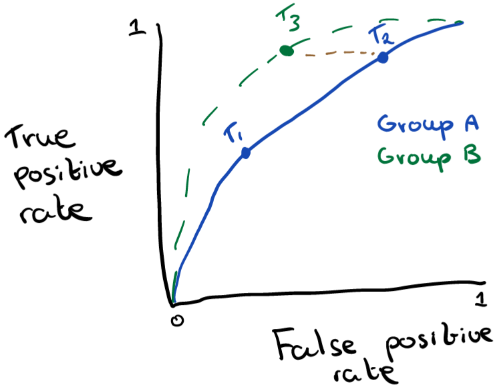


When we have a **score** function R we can make a decision by setting a threshold (t): $D = \mathbb{1}\{R > t\}$

One threshold: Select one best threshold.

D=decision; Y=target variable/outcome

ROC curves



One threshold: Select one best threshold.

Group-specific thresholds:

- Can be used for equal opportunity (equal TPR)
- May require additional randomization for equalized odds (equal TPR and equal FPR)

D=decision; Y=target variable/outcome

ROC curves

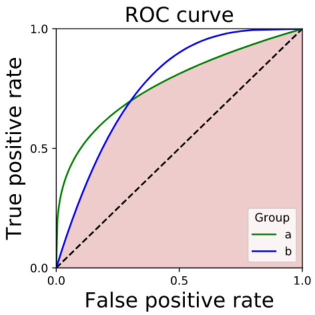


Figure: Fig. 6 from the fairml book, chapter 2

One threshold: Select one best threshold.

Group-specific thresholds:

- Can be used for equal opportunity (equal TPR)
- May require additional randomization for equalized odds (equal TPR and equal FPR)

D=decision; Y=target variable/outcome

Post processing: Equalized odds

We create a “derived” predictor $\tilde{Y}$.
The derived predictor $\tilde{Y}$ is a (possibly randomized) function that only depends on $(\hat{Y}, A)$.
In the binary setting, the derived predictor $\tilde{Y}$ is fully described by four parameters in [0,1]:

- $p_{0,0} = P(\tilde{Y} = 1 | \hat{Y} = 0, A = 0)$
- $p_{0,1} = P(\tilde{Y} = 1 | \hat{Y} = 0, A = 1)$
- $p_{1,0} = P(\tilde{Y} = 1 | \hat{Y} = 1, A = 0)$
- $p_{1,1} = P(\tilde{Y} = 1 | \hat{Y} = 1, A = 1)$

Note slight change of notation!
A=sensitive attribute;
$\hat{Y}$=decision by the original predictor;
$\tilde{Y}$=decision by the derived predictor;
Y=target variable/outcome

Proposed by

Equality of Opportunity in Supervised Learning, Hardt et al., NIPS 2016 [link]

Post processing: Equalized odds

We create a "derived" predictor $\tilde{Y}$.
The derived predictor $\tilde{Y}$ is a (possibly randomized) function that only depends on $(\hat{Y}, A)$.
In the binary setting, the derived predictor $\tilde{Y}$ is fully described by four parameters in [0,1]:

- $p_{0,0} = P(\tilde{Y} = 1 | \hat{Y} = 0, A = 0)$
- $p_{0,1} = P(\tilde{Y} = 1 | \hat{Y} = 0, A = 1)$
- $p_{1,0} = P(\tilde{Y} = 1 | \hat{Y} = 1, A = 0)$
- $p_{1,1} = P(\tilde{Y} = 1 | \hat{Y} = 1, A = 1)$

Note slight change of notation!
A=sensitive attribute;
$\hat{Y}$=decision by the original predictor;
$\tilde{Y}$=decision by the derived predictor;
Y=target variable/outcome

Randomization: For example, for all cases where $\hat{Y} = 0, A = 0$, we first randomize and then assign $p_{0,0}$ of the instances the positive label ($\tilde{Y} = 1$).

Post processing: Equalized odds

We create a "derived" predictor $\tilde{Y}$.
The derived predictor $\tilde{Y}$ is a (possibly randomized) function that only depends on $(\hat{Y}, A)$.
In the binary setting, the derived predictor $\tilde{Y}$ is fully described by four parameters in [0,1]:

- $p_{0,0} = P(\tilde{Y} = 1 | \hat{Y} = 0, A = 0)$
- $p_{0,1} = P(\tilde{Y} = 1 | \hat{Y} = 0, A = 1)$
- $p_{1,0} = P(\tilde{Y} = 1 | \hat{Y} = 1, A = 0)$
- $p_{1,1} = P(\tilde{Y} = 1 | \hat{Y} = 1, A = 1)$

Note slight change of notation!
A=sensitive attribute;
$\hat{Y}$=decision by the original predictor;
$\tilde{Y}$=decision by the derived predictor;
Y=target variable/outcome

Randomization:

Let's say we have $p_{0,0} = 0.5$.

A	$\hat{Y}$	$\tilde{Y}$
0	0	0
0	0	0
0	0	1
0	0	1

Post processing: Equalized odds

We create a "derived" predictor $\tilde{Y}$.
The derived predictor $\tilde{Y}$ is a (possibly randomized) function that only depends on $(\hat{Y}, A)$.
In the binary setting, the derived predictor $\tilde{Y}$ is fully described by four parameters in [0,1]:

- $p_{0,0} = P(\tilde{Y} = 1 | \hat{Y} = 0, A = 0)$
- $p_{0,1} = P(\tilde{Y} = 1 | \hat{Y} = 0, A = 1)$
- $p_{1,0} = P(\tilde{Y} = 1 | \hat{Y} = 1, A = 0)$
- $p_{1,1} = P(\tilde{Y} = 1 | \hat{Y} = 1, A = 1)$

Note slight change of notation!
A=sensitive attribute;
$\hat{Y}$=decision by the original predictor;
$\tilde{Y}$=decision by the derived predictor;
Y=target variable/outcome

Optimization: Find best parameters $(p_{0,0}, p_{0,1}, p_{1,0}, p_{1,1})$ that minimizes loss $l(\tilde{Y}, Y)$ subject to constraints using a linear program.

Note: For finding the parameters we need to also have Y. During prediction, we only need $\hat{Y}$ and A.

A	Y	$\hat{Y}$	$\tilde{Y}$
1	1	1	0
1	1	1	1
1	0	1	1
1	0	1	0
1	0	0	0
1	1	1	0
1	1	1	0
1	0	1	0
1	0	1	0
1	0	0	0
0	1	0	0
0	1	0	0
0	0	0	0
0	0	1	1
0	0	0	0
0	1	0	0
0	1	1	1
0	0	0	0
0	0	0	0
0	0	0	0

Example

True positive rates **(equal opportunity)**:

$$P[\hat{Y} = 1 | Y = 1, A = a] = P[\hat{Y} = 1 | Y = 1, A = b]$$

TPR (recall) of $\hat{Y}$ for A=1: ?
TPR (recall) of $\hat{Y}$ for A=0: ?

A	Y	$\hat{Y}$	$\tilde{Y}$
1	1	1	0
1	1	1	1
1	0	1	1
1	0	1	0
1	0	0	0
1	1	1	0
1	1	1	0
1	0	1	0
1	0	1	0
1	0	0	0
0	1	0	0
0	1	0	0
0	0	0	0
0	0	1	1
0	0	0	0
0	1	0	0
0	1	1	1
0	0	0	0
0	0	0	0
0	0	0	0

Example

True positive rates **(equal opportunity)**:

$$P[\hat{Y} = 1 | Y = 1, A = a] = P[\hat{Y} = 1 | Y = 1, A = b]$$

TPR (recall) of $\hat{Y}$ for A=1: 1
TPR (recall) of $\hat{Y}$ for A=0: 0.25

A	Y	$\hat{Y}$	$\tilde{Y}$
1	1	1	0
1	1	1	1
1	0	1	1
1	0	1	0
1	0	0	0
1	1	1	0
1	1	1	0
1	0	1	0
1	0	1	0
1	0	0	0
0	1	0	0
0	1	0	0
0	0	0	0
0	0	1	1
0	0	0	0
0	1	0	0
0	1	1	1
0	0	0	0
0	0	0	0
0	0	0	0

Example

Parameters of the derived predictor:

- $p_{0,0} = P(\tilde{Y} = 1|\hat{Y} = 0, A = 0) = 0$
- $p_{0,1} = P(\tilde{Y} = 1|\hat{Y} = 0, A = 1) = 0$
- $p_{1,0} = P(\tilde{Y} = 1|\hat{Y} = 1, A = 0) = 1$
- $p_{1,1} = P(\tilde{Y} = 1|\hat{Y} = 1, A = 1) = 0.25$

A	Y	$\hat{Y}$	$\tilde{Y}$
1	1	1	0
1	1	1	1
1	0	1	1
1	0	1	0
1	0	0	0
1	1	1	0
1	1	1	0
1	0	1	0
1	0	1	0
1	0	0	0
0	1	0	0
0	1	0	0
0	0	0	0
0	0	1	1
0	0	0	0
0	1	0	0
0	1	1	1
0	0	0	0
0	0	0	0
0	0	0	0

Example

Parameters of the derived predictor:

- $p_{0,0}$ = $P(\tilde{Y} = 1|\hat{Y} = 0, A = 0)$ = 0
- $p_{0,1}$ = $P(\tilde{Y} = 1|\hat{Y} = 0, A = 1)$ = 0
- $p_{1,0}$ = $P(\tilde{Y} = 1|\hat{Y} = 1, A = 0)$ = 1
- $p_{1,1}$ = $P(\tilde{Y} = 1|\hat{Y} = 1, A = 1)$ = 0.25

After post processing:

True positive rate (recall) for A=1: 0.25
True positive rate (recall) for A=0: 0.25

A	Y	$\hat{Y}$	$\tilde{Y}$
1	1	1	0
1	1	1	1
1	0	1	1
1	0	1	0
1	0	0	0
1	1	1	0
1	1	1	0
1	0	1	0
1	0	1	0
1	0	0	0
0	1	0	0
0	1	0	0
0	0	0	0
0	0	1	1
0	0	0	0
0	1	0	0
0	1	1	1
0	0	0	0
0	0	0	0
0	0	0	0

Example

Parameters of the derived predictor:

- $p_{0,0} = P(\tilde{Y} = 1 | \hat{Y} = 0, A = 0) = 0$
- $p_{0,1} = P(\tilde{Y} = 1 | \hat{Y} = 0, A = 1) = 0$
- $p_{1,0} = P(\tilde{Y} = 1 | \hat{Y} = 1, A = 0) = 1$
- $p_{1,1} = P(\tilde{Y} = 1 | \hat{Y} = 1, A = 1) = 0.25$

After post processing:

True positive rate (recall) for A=1: 0.25
True positive rate (recall) for A=0: 0.25

But, the TPR when A=1 has decreased... :(

Post processing: Reranking

Suppose you do an image search for “*CEO*” …

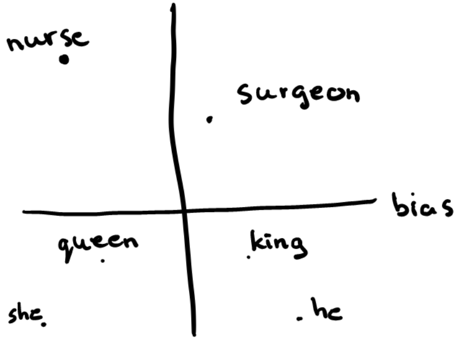
Ranking of individuals: image search (“*CEO*”, “*nurse*”), to find candidates for hiring (“I’m looking for a web developer…”).

Geyik et al. propose re-ranking methods to achieve a desired distribution of top results over protected attributes.

Fairness-Aware Ranking in Search & Recommendation Systems with Application to LinkedIn Talent Search, Geyik et al., KDD 2019 [link]

recap!

Word embeddings can contain biases

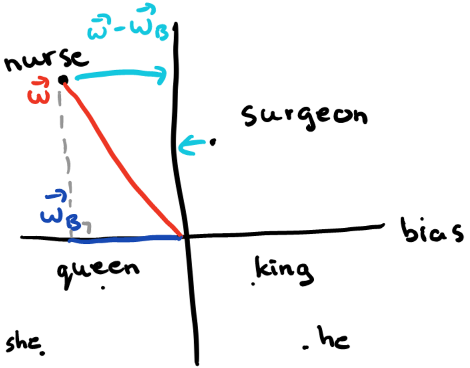


Man is to computer programmer as woman is to homemaker? Debiasing word embeddings, Bolukbasi et al. NIPS 2016 [link]

Semantics derived automatically from language corpora contain human-like biases, Caliskan et al., Science 2017 [link]

nurse [-0.1, 0.3, 0.5, -0.8 ...]

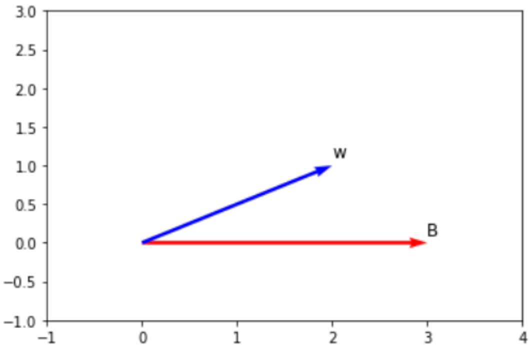
“Debiasing” word embeddings: projections



Man is to computer programmer as woman is to homemaker? Debiasing word embeddings, Bolukbasi et al. NIPS 2016 [link]

Projections: Example

Suppose we have a gender direction $B = [3, 0]$
and a word embedding $\vec{w} = [2, 1]$



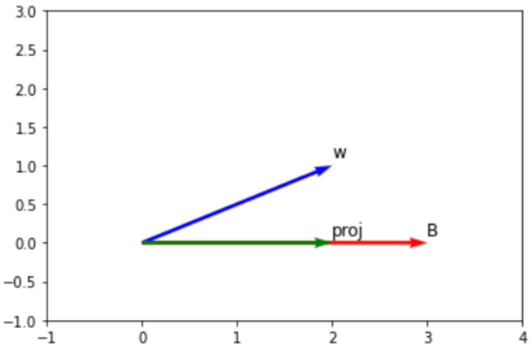
Projections: Example

Suppose we have a gender direction $B = [3, 0]$
and a word embedding
$\vec{w} = [2, 1]$

$\vec{proj}$: the projection of $\vec{w}$ onto the (gender) bias direction B.

$$\vec{proj} = \frac{\vec{w} \cdot B}{B \cdot B} B$$

Here: $\vec{proj} = [2, 0]$

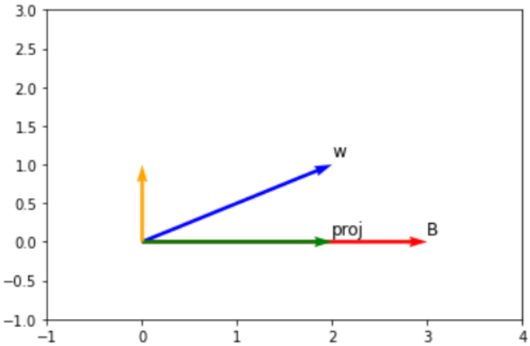


Projections: Example

New embedding:

$$\frac{\vec{w} - \vec{proj}}{||\vec{w} - \vec{proj}||}$$

Here: $[0, 1]$



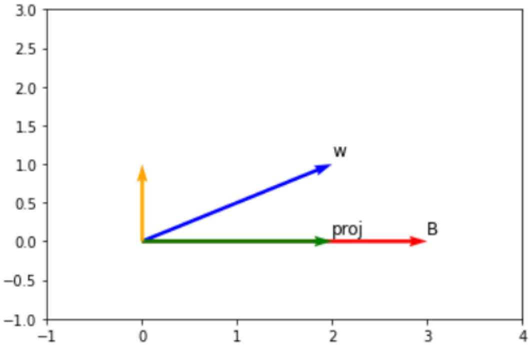
Projections: Example

New embedding:

$$\frac{\vec{w} - \vec{proj}}{||\vec{w} - \vec{proj}||}$$

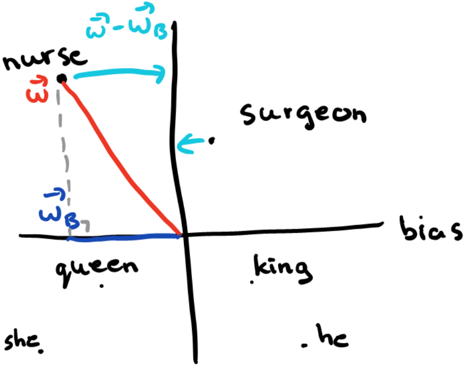
Here: $[0, 1]$

Note: $\vec{w} - \vec{proj}$ is orthogonal to B (dot product is 0!).



Haven't seen projections before? See this Khan academy video on projections: [link]

“Debiasing” word embeddings: projections



New embedding for a word w:

$$\frac{\vec{w} - \vec{w}_B}{||\vec{w} - \vec{w}_B||}$$

with $\vec{w}_B$ the projection of $\vec{w}$ onto the (gender) bias direction.

Man is to computer programmer as woman is to homemaker? Debiasing word embeddings, Bolukbasi et al. NIPS 2016 [link]

In-processing

In-processing

- Pre-processing
- In-processing
- Post-processing

- Requires access to model and data.
- Often model specific.
- Optimize with criteria in mind.

Today: constraints, adversarial debiasing

Logistic Regression

We have N instances in our training set. With logistic regression we want to find the parameters $\boldsymbol{\theta}$ minimize the following loss:

$$\hat{\boldsymbol{\theta}} = \text{arg}\min_{\boldsymbol{\theta}} \frac{1}{N} \sum L(\hat{y}, y; \boldsymbol{\theta}) + \lambda R(\boldsymbol{\theta})$$

With $R(\boldsymbol{\theta})$ a regularization term, for example: $||\boldsymbol{\theta}||_2^2$

Logistic Regression

We have N instances in our training set. With logistic regression we want to find the parameters $\boldsymbol{\theta}$ minimize the following loss:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{N} \sum L(\hat{y}, y; \boldsymbol{\theta}) + \lambda R(\boldsymbol{\theta})$$

With $R(\boldsymbol{\theta})$ a regularization term, for example: $||\boldsymbol{\theta}||_2^2$

Idea by Kamishima et al.: Add a regularization term to make the classifier more fair!

Fairness-Aware Classifier with Prejudice Remover Regularizer, Kamishima et al. ECML PKDD 2012 [link]

Logistic Regression

We have N instances in our training set. With logistic regression we want to find the parameters $\boldsymbol{\theta}$ minimize the following loss:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{N} \sum L(\hat{y}, y; \boldsymbol{\theta}) + \lambda R(\boldsymbol{\theta}) + \eta * ?$$

With $R(\boldsymbol{\theta})$ a regularization term, for example: $\|\boldsymbol{\theta}\|_2^2$

Idea by Kamishima et al.: Add a regularization term to make the classifier more fair!

Fairness-Aware Classifier with Prejudice Remover Regularizer, Kamishima et al. ECML PKDD 2012 [link]

Mutual information

The mutual information between two random variables X and Y:

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} P(x,y) log \frac{P(x,y)}{P(x)P(y)}$$

$I(X;Y) = 0$ if and only if X and Y are independent.

Mutual information

The mutual information between two random variables X and Y:

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} P(x,y) log \frac{P(x,y)}{P(x)P(y)}$$

$I(X;Y) = 0$ if and only if X and Y are independent.

X	Y
0	1
0	1
0	1
0	0
0	0
1	1
1	1
1	1
1	0
1	0

I(X,Y) = 0

Mutual information

The mutual information between two random variables X and Y:

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} P(x,y) log \frac{P(x,y)}{P(x)P(y)}$$

$I(X;Y) = 0$ if and only if X and Y are independent.

Here: the mutual information between classification results ($\hat{Y}$) and sensitive attributes (A).

X	Y
0	1
0	1
0	1
0	0
0	0
1	1
1	1
1	1
1	0
1	0

I(X,Y) = 0

Constraints

We can add constraints to the optimization process.

Just discussed: Adding it as a regularizer. Flexible strategy, we could add variants of this term to a model based on minimizing loss. We still need to choose η (the weight of the regularization term)

Alternative: hard constraint.

Fairness Constraints: Mechanisms for Fair Classification, Zafar et al, AISTATS, 2017 [link]

recap!

Adversarial debiasing

“Blinding” (removing sensitive features) doesn’t work!
It is likely that there will be many other features that can act as a proxy for the sensitive feature (e.g., zip code for race).

Adversarial debiasing (informally): *Can we still somehow encourage the ML model to not make (implicit) use of sensitive features?*

Adversarial debiasing: GANs

Generative Adversarial Networks (GANs)

Figure: From https://developers.google.com/machine-learning/gan/gan_structure

Adversarial debiasing: GANs

Generative Adversarial Networks (GANs)

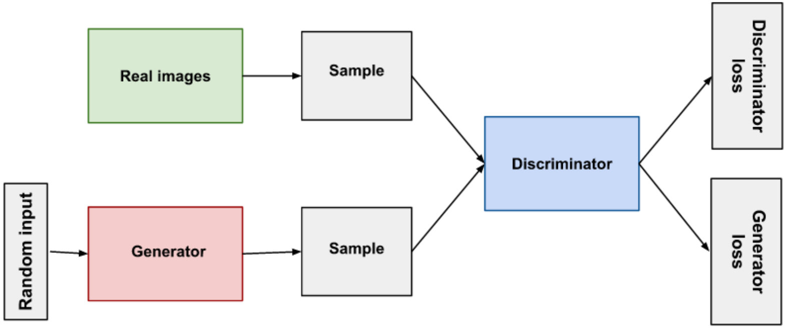


Figure: From https://developers.google.com/machine-learning/gan/gan_structure

Adversarial debiasing: General idea

Two competing goals:

Predictor:

- Try to predict Y from X.
- Loss function: $L_p(\hat{y}, y)$.

Mitigating Unwanted Biases with Adversarial Learning, Zhang et al. AIES '18 [pdf]

Adversarial debiasing: General idea

Two competing goals:

Predictor:

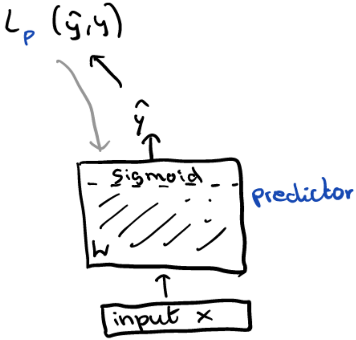
- Try to predict Y from X.
- Loss function: $L_p(\hat{y}, y)$.

Adversary:

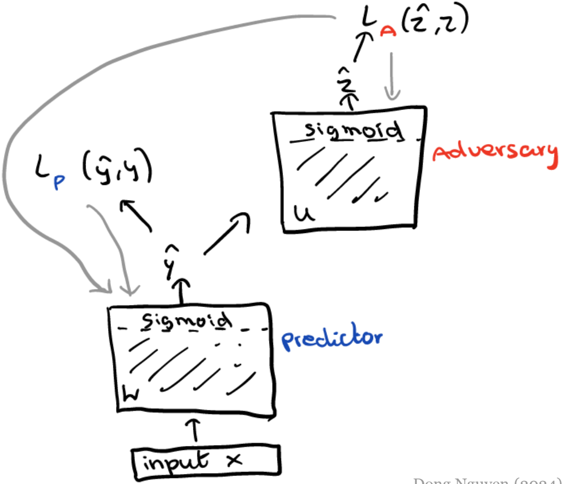
- Try to predict the protected attribute Z from the output layer of the network. *Exact input depends on the fairness criterion.*
- Loss function: $L_A(\hat{z}, z)$
- Similar to the discriminator in a GAN.

Mitigating Unwanted Biases with Adversarial Learning, Zhang et al. AIES '18 [pdf]

Adversarial debiasing: setup



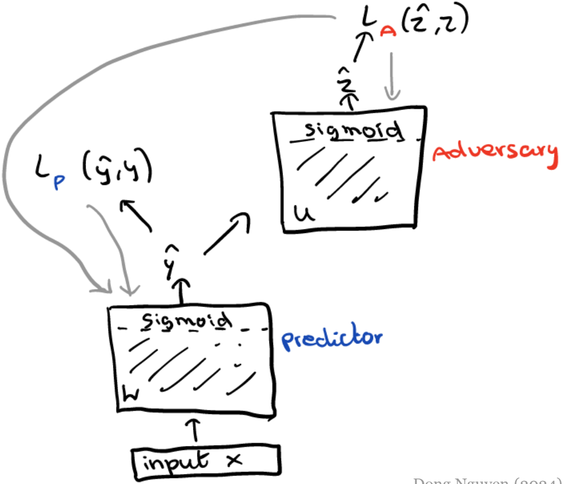
Adversarial debiasing: setup



X: input; Y: target variable;
Z: protected attribute
L_p: predictor loss
L_A: adversary loss

Mitigating Unwanted Biases with Adversarial Learning, Zhang et al. AIES ’18 [pdf]

Adversarial debiasing: setup

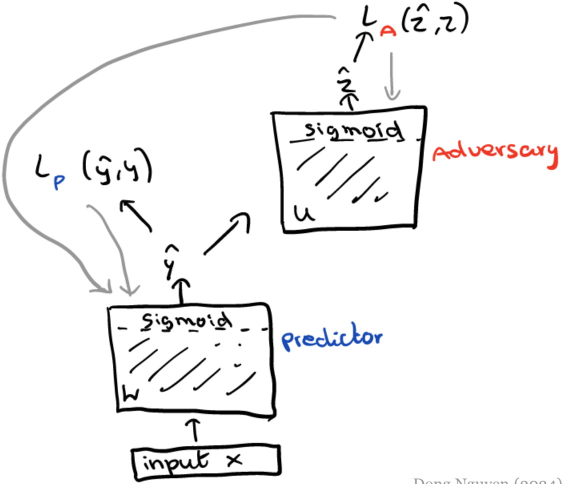


X: input; Y: target variable;
Z: protected attribute
L_p: predictor loss
L_A: adversary loss

Adversary:
Demographic parity: $Z \perp \hat{Y}$
The adversary shouldn't be able to predict Z from $\hat{Y}$!

Therefore: Adversary gets as input the prediction $\hat{Y}$

Adversarial debiasing: setup

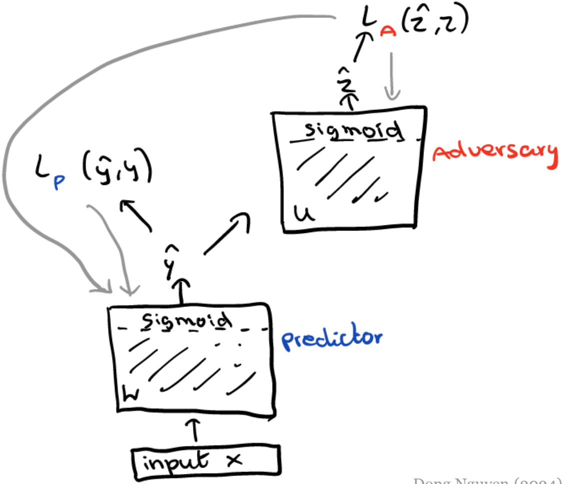


X: input; Y: target variable;
Z: protected attribute
L_p: predictor loss
L_A: adversary loss

Adversary:
Equal of opportunity:
$\hat{Y} \perp Z | Y = 1$

Restrict the training set for the adversary to $Y = 1$.

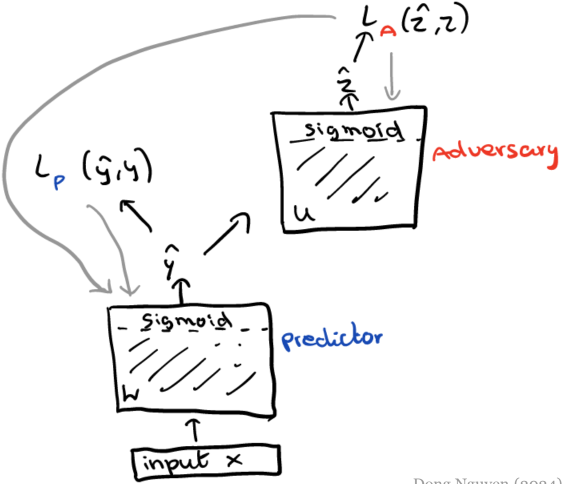
Adversarial debiasing: setup



X: input; Y: target variable;
Z: protected attribute
L_p: predictor loss
L_A: adversary loss

Learning: In a classification setting, we can use the cross entropy loss for both L_p (predictor loss) and L_A (adversary loss).

Adversarial debiasing: setup



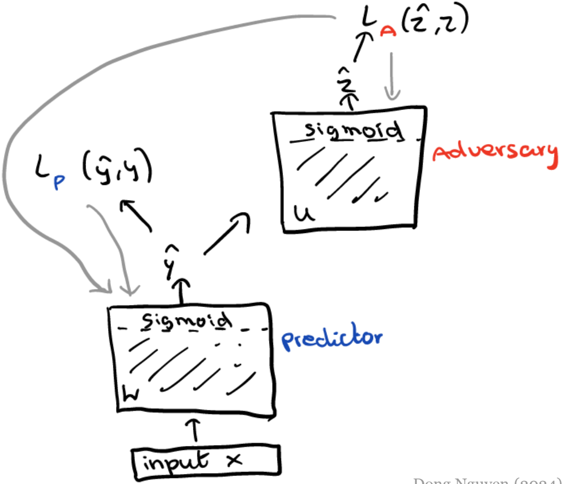
X: input; Y: target variable;
Z: protected attribute
L_p: predictor loss
L_A: adversary loss

Learning:

Update adversary weights (U) using: $\nabla_U L_A$

Update predictor weights (W) using: $\nabla_W L_p$?

Adversarial debiasing: setup



X: input; Y: target variable;
Z: protected attribute
L_p: predictor loss
L_A: adversary loss

Learning:

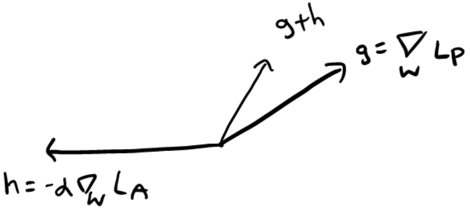
Update adversary weights (U) using: $\nabla_U L_A$

Update predictor weights (W) using: $\nabla_W L_p$ - $\alpha \nabla_W L_A$?

Adversarial debiasing: Learning

Update the weights of the classifier (W) using:

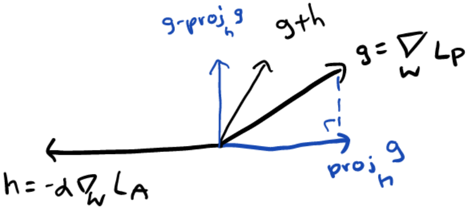
$$\nabla_W L_p - \alpha \nabla_W L_A$$



Adversarial debiasing: Learning

Update the weights of the classifier (W) using:

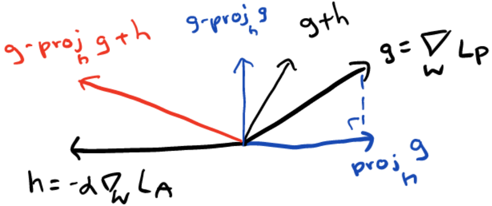
$$\nabla_W L_p - \boxed{proj_{(\nabla_W L_A)} \nabla_W L_p} - \alpha \nabla_W L_A$$



Adversarial debiasing: Learning

Update the weights of the classifier (W) using:

$$\nabla_W L_p - \boxed{proj_{(\nabla_W L_A)} \nabla_W L_p} - \alpha \nabla_W L_A$$



Adversarial debiasing

Can be applied to many neural network architectures, as long as training is using gradients.

Many variants on this idea.

Like GANs, can be tricky to get the training "right".

If you're interested in seeing implementations of this:

- https://colab.research.google.com/notebooks/ml_fairness/adversarial_debiasing.ipynb
- https://github.com/Trusted-AI/AIF360/blob/master/aif360/algorithms/inprocessing/adversarial_debiasing.py

Which method should I use

- Do I have access to the data, or the model?
- Which fairness criterion do I prioritize?
  - Do I take my data as the ground truth? Do I want models that reproduce the status quo? Focusing on balancing error rates? (“fairness preserving”)
  - Do I believe my data is a result of existing inequalities? (“fairness transforming”)
- Explore fairness libraries!

Taking stock. Outlook.

Most of our discussion has taken a narrow view on fairness.

Focusing on input, output, features alone is a *very narrow view* on fairness.

Beyond only race or sex

- Intersectionality: e.g., *black woman*. Relatively little attention so far in literature, see previous lecture.
- Other groups: e.g., work by Hutchinson et al. 2020 look at biases towards mentions of disability.
- Binary gender.

Social Biases in NLP Models as Barriers for Persons with Disabilities, Hutchinson et al., ACL 2020 [pdf]

Critiques

Lots of the conversation and research is *US(/Europe) centric*.

Critiques

Lots of the conversation and research is

Think about your *own* context. How do the ideas discussed so far translate to your own context? Do you identify any gaps?

Critiques

Selbst et al. 2019 outline five traps:

- **The Framing Trap**: Failure to model the entire system over which a social criterion, such as fairness, will be enforced.
- **The Portability Trap**: Failure to understand how repurposing algorithmic solutions designed for one social context may be misleading, inaccurate, or otherwise do harm when applied to a different context.
- **The Formalism Trap**: Failure to account for the full meaning of social concepts such as fairness, which can be procedural, contextual, and contestable, and cannot be resolved through mathematical formalisms.
- **The Ripple Effect Trap**: Failure to understand how the insertion of technology into an existing social system changes the behaviors and embedded values of the pre-existing system.
- **The Solutionism Trap**: Failure to recognize the possibility that the best solution to a problem may not involve technology.

Long term effects

We often only focus on immediate effects but what about *long-term* effects? *Because decision making systems can shape the environment they are applied to.*

D'Amour et al. (2020) propose the use of simulations to study long-term dynamics. See also ML-fairness-gym.

Fairness Is Not Static: Deeper Understanding of Long Term Fairness via Simulation Studies, D'Amour et al., FAT* 2020 [pdf]

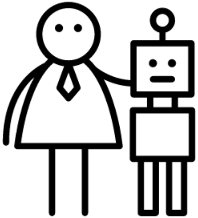
Algorithmic monoculture

- An increasing concern: *algorithmic monoculture*, when many/all decision makers rely on the same systems, or systems that share components (e.g. pre-trained models).
- As an individual: impossible to get a second opinion; You might get locked out of the market.
- Wide-spread failures.

Algorithmic monoculture and social welfare, Kleinberg and Raghavan, PNAS 2021 [pdf]
Picking on the same person: Does algorithmic monoculture lead to outcome homogenization?, Bommasani et al., NeurIPS 2022,[link]

Don’t forget the human!

Decision making is rarely fully automatic! But how are people’s decisions influenced by ML systems?

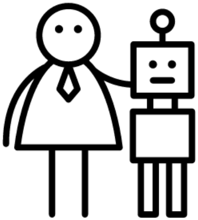


Don’t forget the human!

Decision making is rarely fully automatic! But how are people’s decisions influenced by ML systems?

Automation bias: The tendency to favor output from automated systems (*example: spell checker*).

Algorithm aversion: The reluctance to use imperfect but better automated systems.

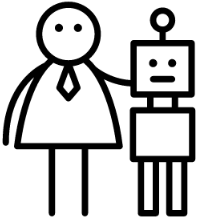


Algorithm Aversion: People Erroneously Avoid Algorithms after Seeing Them Err, Dietvorst et al., Journal of Experimental Psychology 2015 [link]
A systematic review of algorithm aversion in augmented decision making, Burton et al., Journal of Behavioral Decision Making 2020 [link]

Don’t forget the human!

Decision making is rarely fully automatic! But how are people’s decisions influenced by ML systems?

Cummings 2006: “*[…] cause operators to relinquish a sense of responsibility and subsequently accountability because of a perception that the automation is in charge*”



Automation and Accountability in Decision Support System Interface Design, Cummings, Journal of Technology Studies 2006 [link]

Perception of fairness

Wang et al. 2020: "Outcome favorability" bias: People tend to rate ML decision making systems as more fair when they predict in their favor.

Binns et al. 2018: How do explanation styles influence fairness perceptions? In short: it's complicated...

Factors Influencing Perceived Fairness in Algorithmic Decision-Making: Algorithm Outcomes, Development Procedures, and Individual Differences, Wang et al., CHI 2020 [link]
'It's Reducing a Human Being to a Percentage': Perceptions of Justice in Algorithmic Decisions, Binns et al., CHI 2018 [link]

Explainable ML

- What signals is my system using? Is it latching on to features that act as proxies for protected attributes?
- People often find it difficult to use the output of ML systems effectively. Can I *trust* this decision?
- *Why was my loan rejected?* “Right to explanation”. What can I do to get a loan in the future? (Recourse)

A case for explainable ML?

Literature

Required literature

- Fairness and Abstraction in Sociotechnical Systems, Selbst et al., FAT* 2019 [link]
- Mitigating Unwanted Biases with Adversarial Learning, Zhang et al. AIES ’18 [link]