



BITS Pilani
Pilani Campus

Fair, Accountable, Transparent Machine Learning (FAccT ML) ZG517

Dr. Sugata Ghosal
Sugata.ghosal@pilani.bits-pilani.ac.in



**Session 1
Date – 21th May 2023
Time – 8:45 AM to 10:45 PM**

These slides are prepared by the instructor, with grateful acknowledgement of many others who made their course materials freely available online.

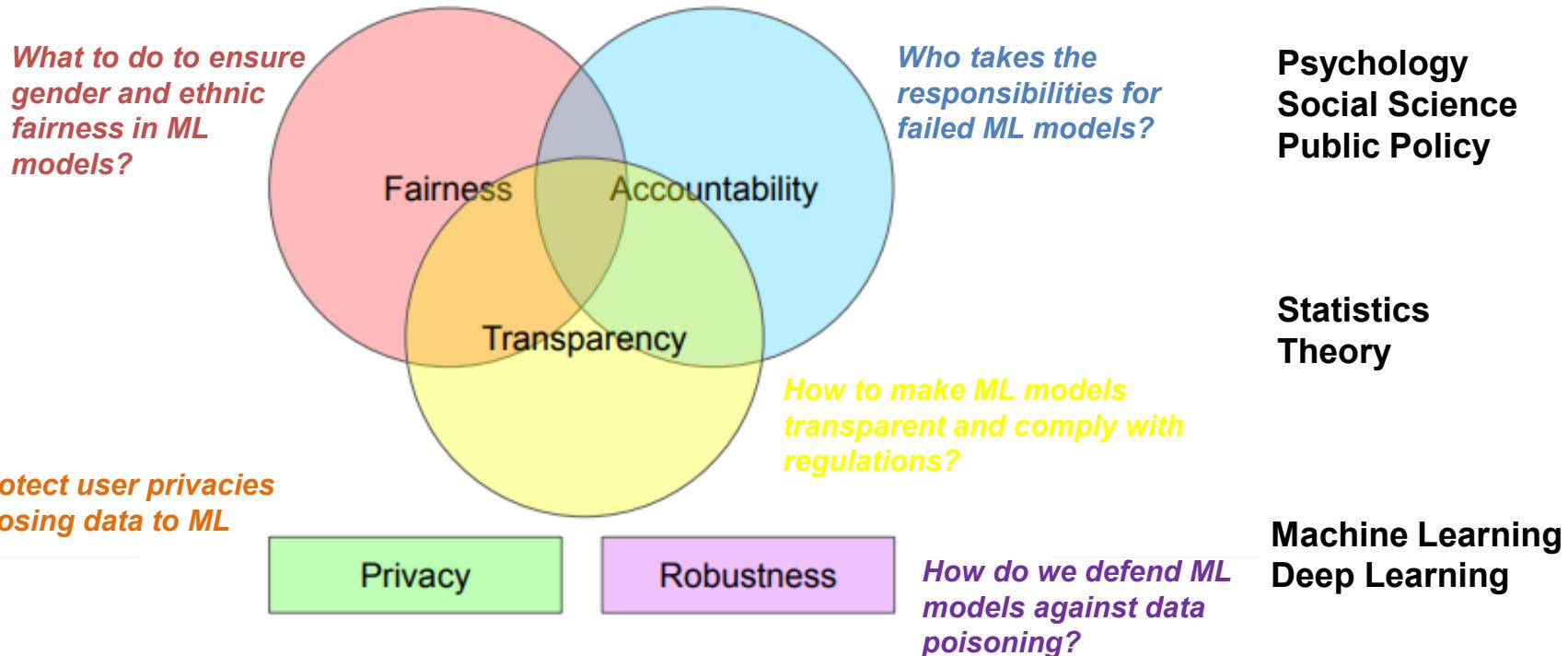
Session Content

- Objective of course
- Evaluation Plan
- Course Overview
- Bias and Fairness

Objective of course

1. Introduce you to the concepts of bias and fairness and techniques for incorporating these in ML
2. Introduce you to the concepts of interpretability and transparency and techniques for incorporating these in ML
3. Introduce you to the concepts of robustness and techniques for robust ML
4. Introduce you to the concepts of privacy in ML

What We'll Cover in this Course



How to protect user privacies when exposing data to ML models?

Privacy

Robustness

Fairness and Bias – 6 Lectures (TBD)

1. Fairness and Bias

- ✓ Sources of Bias
- ✓ Real world examples
- ✓ Sensitive Features
- ✓ Fairness through unawareness

2. Learning Fair Representations

- ✓ Major Fairness criteria
- ✓ Prejudice Removing Regularizer
- ✓ Case Studies: FICO, adult income

3. Fairness thru input manipulation

- ✓ Basic Data Manipulation Techniques
- ✓ Individual Fairness
- ✓ Optimized Pre-processing
- ✓ Learning to Defer

4. Fair Casual Reasoning

- ✓ Causal Fairness and Inherent Bias
- ✓ Counterfactual Fairness
- ✓ Equalized Counterfactual Odds
- ✓ Multiple Causal Worlds

5. Fairness in NLP, Computer Vision

- ✓ Biases in NLP Models
- ✓ Mitigation Strategies
- ✓ Counterfactual Face Attribution
- ✓ Gender Equalized Image Captioning
- ✓ Adversarial Removal of Gender Features

Interpretability – 5 Lectures (TBD)

- **Interpretability and Transparency**
 - ✓ ML Interpretability
 - ✓ Intrinsically Interpretable Models
 - ✓ Interpretability Concepts
 - ✓ Interpretability and performance trade-offs Instance-based Learning
- **Feature interaction for interpretability**
 - ✓ Feature Interaction
 - ✓ Layerwise Relevance Propagation
 - ✓ DeepLift
 - ✓ Shapley Additive Explanations (SHAP)
- **Example and Visualization Based Methods for Interpretability**
 - ✓ Example Based Methods
 - ✓ Counterfactual Explanations
 - ✓ Contrastive Examples
 - ✓ Concept Based Methods
- **Post Hoc Interpretability**
 - ✓ Proxy Models
 - ✓ Local Surrogate Methods, e.g., LIME
 - ✓ Rule Based Learner
- **Interpreting Deep Networks**
 - ✓ Visualization Based Methods
 - ✓ Activation Visualization
 - ✓ Gradient Based Feature Attribution

Robustness and Privacy – 3 Lectures (TBD)

- Robustness and Adversarial Attacks & Defense
 - ✓ Adversarial Attack
 - ✓ White-box Evasion Attack
 - ✓ Transferability of Attack
 - ✓ Black-box Evasion Attack
 - ✓ Adversarial Defense
 - ✓ Defense Strategies
 - ✓ Robust Optimization
 - ✓ Certified Defense
- ML Auditing and Privacy
 - ✓ ML Auditing
 - ✓ Distill-and-Compare
 - ✓ Privacy in ML
 - ✓ Differential Privacy
 - ✓ Model Inversion Attack
 - ✓ Local Differential Privacy
 - ✓ Federated Learning

Textbooks

T1	Barocas, Solon, Moritz Hardt, and Arvind Narayanan. <u>Fairness and Machine Learning</u> , 2018.
T2	Molnar, Christoph. <u>Interpretable machine learning</u> , 2019.

R1	Christopher M. Bishop, <u>Pattern Recognition & Machine Learning</u> , Springer
R2	Aston Zhang, Zach Lipton, Mu Li, Alex Smola, <u>Dive into Deep Learning</u> , 2021

Evaluation Plan (TBC)

Name	Type	Weight
3 Quiz, best 2 scores will be taken	Online	10%
Assignment-I	Take Home	10%
Assignment-II	Take Home	10%
Mid-Semester Test	Closed Book	30%
Comprehensive Exam	Open Book	40%

Please note there will be no change in submission dates for quiz and assignment

Lab Plan

Lab No.	Lab Objective
1	Detecting bias in ML model using different fairness metrics
2	Bias mitigation in ML model using Disparate impact remover
3	Improving the Fairness of a Classifier using Prejudice Removal Regularizer
4	Implementing LIME and SHAP algorithms to interpret different machine learning classifiers
5	Visualizing and interpreting a CNN based Image classifier
6	Implementing Adversarial attacks on CNN based Image classifier

- *Labs not graded*
- *Webinars will be conducted for lab sessions*
- *Labs will be conducted in Python*

Traditional Evaluation of ML Algorithms

- Accuracy
- Precision and recall
- Squared error
- Likelihood
- Posterior probability
- Cost / Utility
- Margin
- Entropy
- etc.

Evaluating Performance

- If y is continuous:
 - Sum-of-Squared-Differences (SSD)
error between predicted and true y :

$$E = \sum_{i=1}^n (f(x_i) - y_i)^2$$

Evaluation Metrics: Confusion Matrix

- If output y is discrete (e.g., binary) :

		PREDICTED CLASS	
		Class=Yes	Class>No
ACTUAL CLASS	Class=Yes	a	b
	Class>No	c	d

a: TP (true positive) b: FN (false negative)

c: FP (false positive) d: TN (true negative)

Evaluation Metrics: Accuracy

		PREDICTED CLASS	
		Class=Yes	Class>No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class>No	c (FP)	d (TN)

- Most widely-used metric:

$$\text{Accuracy} = \frac{a + d}{a + b + c + d} = \frac{TP + TN}{TP + TN + FP + FN}$$

Class Imbalance Problem

- Lots of classification problems where the classes are skewed (more records from one class than another)
 - Credit card fraud
 - Intrusion detection
 - Defective products in manufacturing assembly line
 - COVID-19 test results on a random sample
- Key Challenge:
 - Evaluation measures such as accuracy are not well-suited for imbalanced class

Measures of Classification Performance

	PREDICTED CLASS		
ACTUAL CLASS		Yes	No
	Yes	TP	FN
	No	FP	TN

α is the probability that we reject the null hypothesis when it is true.

This is a Type I error or a false positive (FP).

β is the probability that we accept the null hypothesis when it is false. This is a Type II error or a false negative (FN).

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN}$$

$$ErrorRate = 1 - accuracy$$

$$Precision = Positive\ Predictive\ Value = \frac{TP}{TP + FP}$$

$$Recall = Sensitivity = TP\ Rate = \frac{TP}{TP + FN}$$

$$Specificity = TN\ Rate = \frac{TN}{TN + FP}$$

$$FP\ Rate = \alpha = \frac{FP}{TN + FP} = 1 - specificity$$

$$FN\ Rate = \beta = \frac{FN}{FN + TP} = 1 - sensitivity$$

$$Power = sensitivity = 1 - \beta$$

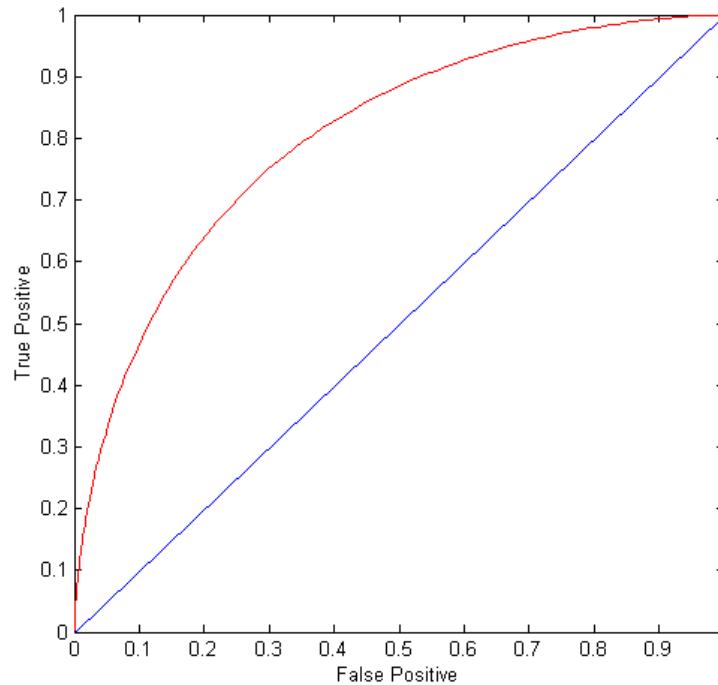
ROC (Receiver Operating Characteristic)

- A graphical approach for displaying trade-off between detection rate and false alarm rate
- Developed in 1950s for signal detection theory to analyze noisy signals
- ROC curve plots TPR against FPR
 - Performance of a model represented as a point in an ROC curve

ROC Curve

(TPR,FPR):

- (0,0): declare everything to be negative class
 - (1,1): declare everything to be positive class
 - (0,1): ideal
 - Diagonal line:
 - Random guessing
 - Below diagonal line:
 - prediction is opposite of the true class



How to Construct an ROC curve

Instance	Score	True Class
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

- Use a classifier that produces a continuous-valued score for each instance
 - The more likely it is for the instance to be in the + class, the higher the score
- Sort the instances in decreasing order according to the score
- Apply a threshold at each unique value of the score
- Count the number of TP, FP, TN, FN at each threshold
 - $TPR = TP/(TP+FN)$
 - $FPR = FP/(FP + TN)$

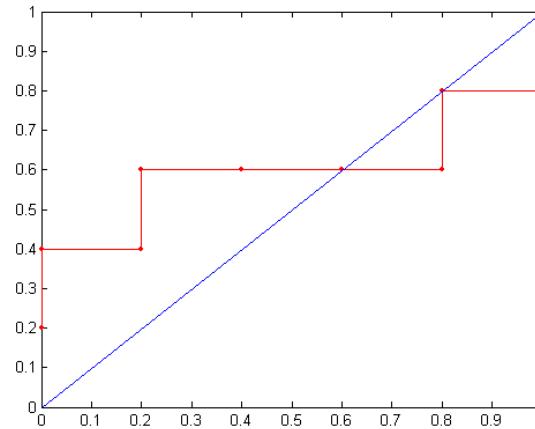
How to construct an ROC curve

Threshold >=

Class	+	-	+	-	-	-	+	-	+	+	
Threshold	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4	3	3	3	3	2	2	1	0
FP	5	5	4	4	3	2	1	1	0	0	0
TN	0	0	1	1	2	3	4	4	5	5	5
FN	0	1	1	2	2	2	2	3	3	4	5
TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0

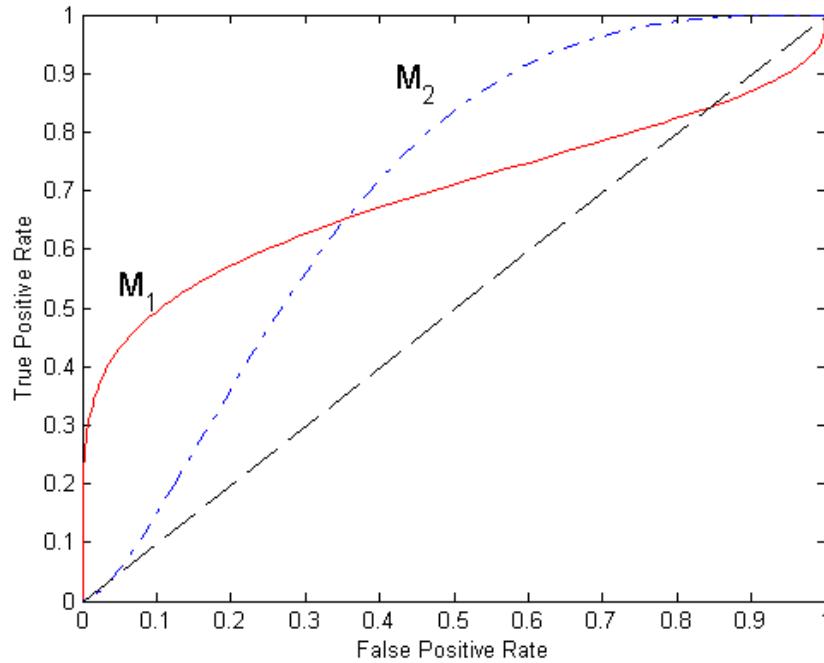
→
→

ROC Curve:



Instance	Score	True Class
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

Using ROC for Model Comparison



- No model consistently outperforms the other
 - M_1 is better for small FPR
 - M_2 is better for large FPR
- Area Under the ROC curve (AUC)
 - Ideal:
 - Area = 1
 - Random guess:
 - Area = 0.5

Bias and Fairness

- Fairness
 - Sources of Biases
 - Real World Examples
 - Sensitive Features
- Major Fairness Criteria
 - Fairness Through Unawareness
 - Group Fairness Metrics

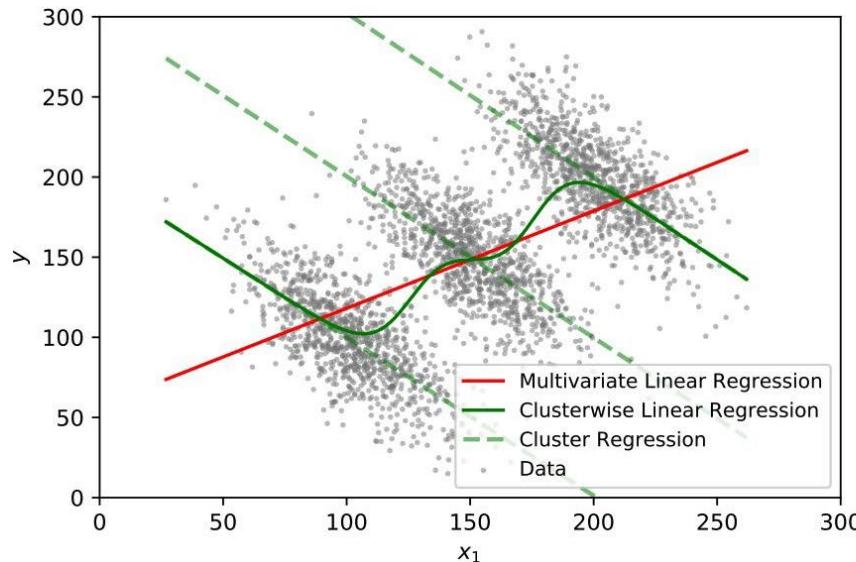
ML Fairness

- What is Fairness?
 - The absence of bias towards an individual or a group ([Mehrabi et al, 2019](#))
- Can ML Models Discriminate?
 - Aren't machines just follow human's instructions?
 - ML models approximate patterns in the data
 - Learns/Amplifies bises at the same time

Sources of Bias

- **Data adequacy.** Infrequent and specific patterns may be down-weighted by the model and so minority records can be unfairly neglected.
 - data collection methodology can exclude or disadvantage certain groups
 - Sometimes records are removed if they contain missing values and these may be more prevalent in some groups than others.
- **Data bias or Negative Legacy.** Even if the amount of data is sufficient to represent each group, training data may reflect existing prejudices / historical unfairness
 - e.g., that female workers are paid less
 - Non-uniform data collection impacting different groups differently
 - choice of input attributes to the model
- **Model adequacy.** The model architecture may describe some groups better than others.
 - a linear model may be suitable for one group but not for another.

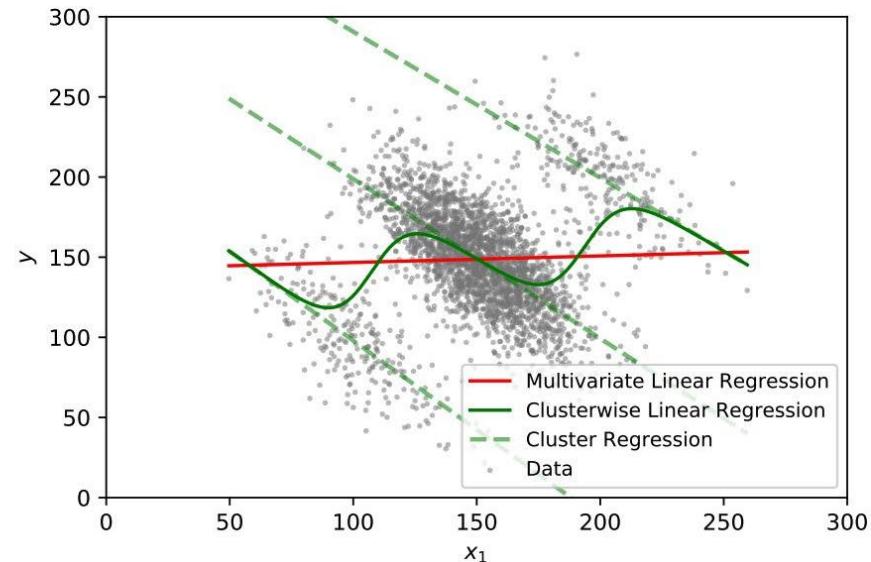
Sources of Bias



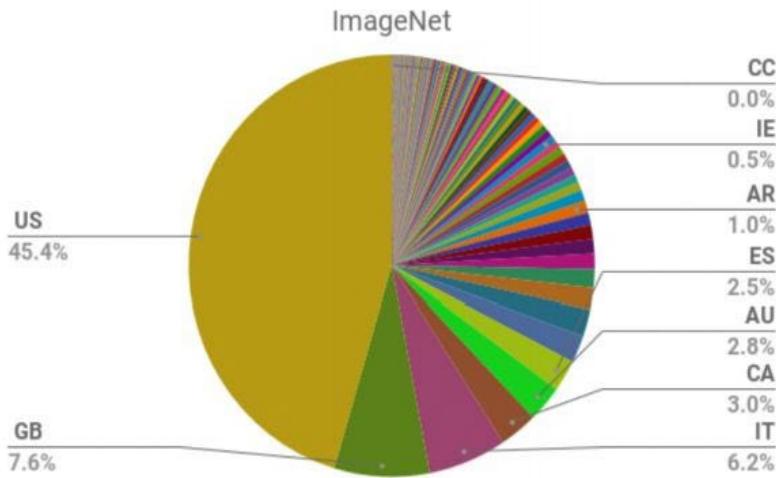
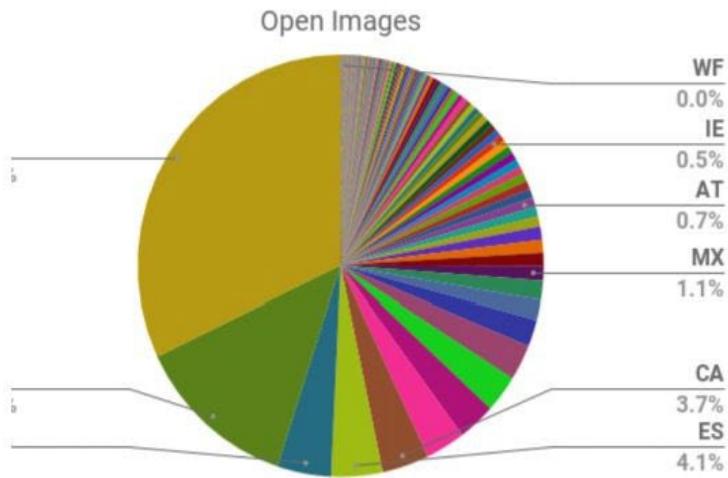
red - biased regression

dashed green - regression for each subgroup

solid green - unbiased regression



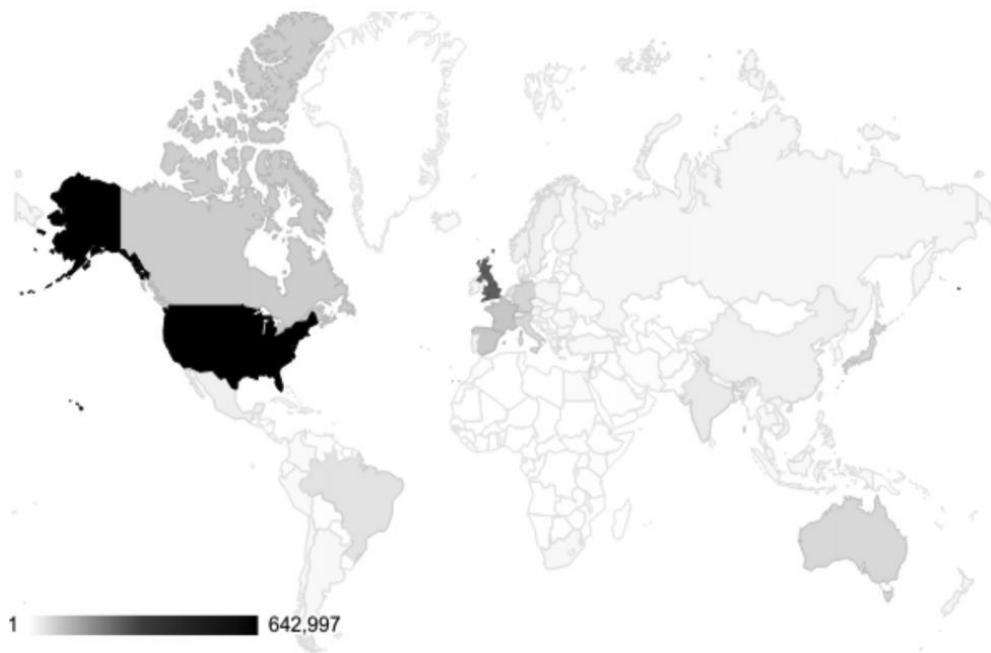
Geographical Representation of ImageNet and Open Images



[Mehrabi et al, 2019](#)

Geographical Representation of Open Images

- One third of the data was collected in US
- 60% of the data was from the six most represented countries.



[Mehrabi et al, 2019](#)

Graduate School Admissions to UC Berkeley, 1973

	Men		Women	
	Applicants	Admitted	Applicants	Admitted
Total	8442	44%	4321	35%

https://en.wikipedia.org/wiki/Simpson%27s_paradox

Graduate School Admissions to UC Berkeley, 1973

Department	Men		Women	
	Applicants	Admitted	Applicants	Admitted
A	825	62%	108	82%
B	560	63%	25	68%
C	325	37%	593	34%
D	417	33%	375	35%
E	191	28%	393	24%
F	373	6%	341	7%

https://en.wikipedia.org/wiki/Simpson%27s_paradox

Real world example of fairness

New Zealand passport robot thinks this Asian man's eyes are closed



By James Griffiths, CNN

Updated 1:46 AM ET, Fri December 9, 2016

X The photo you want to upload does not meet our criteria because:

- Subject eyes are closed

Please refer to the technical requirements. You have 9 attempts left.

Check the photo [requirements](#).

Read more about [common photo problems](#) and [how to resolve them](#).

After your tenth attempt you will need to start again and re-enter the CAPTCHA security check.

Reference number: 20161206-81

Filename: Untitled.jpg

If you wish to [contact us](#) about the photo, you must provide us with the reference number given above.



INSTAGRAM.COM/RICHFRANCY

More from CNN



Two workers at the same Walmart store die of coronavirus



Trump fires intelligence community watchdog who told Congress...

YOU
EARNED IT.
YOU KEEP IT.
START FOR FREE
TaxAct

New Zealand's online passport application system couldn't recognize Richard Lee's open eyes.

HP looking into claim webcams can't see black people

By Mallory Simon, CNN

December 23, 2009 7:25 p.m. EST



A YouTube video shows co-workers trying out an HP webcam with motion-tracking and facial recognition software.

STORY HIGHLIGHTS

- **NEW:** Video was meant to be humorous showing of software glitch, co-workers say
- Co-workers: Motion-tracking webcam moves with white woman, not black man
- "I think my blackness is

(CNN) -- Can Hewlett-Packard's motion-tracking webcams see black people? It's a question posed on a now-viral YouTube video and the company says it's looking into it.

In the video, two co-workers take turns in front of the camera -- the webcam appears to follow Wanda Zamen as she sways in front of the screen and stays still as Desi Cryer moves about.

HP acknowledged in a statement e-mailed to CNN that the cameras may have issues with contrast recognition in certain lighting situations. The webcams, built into HP's new computers, are supposed to keep people's faces and bodies in proportion and centered on the screen as they move.

The video went viral over the weekend, garnering more than 400,000 YouTube page views and a slew of comments on Twitter.

Amazon's Secret AI Hiring Tool Reportedly 'Penalized' Resumes With the Word 'Women's'



Rhett Jones

10/10/18 10:32AM • Filed to: ALGORITHMS ▾

79

3



Photo: Getty

Start building powerful data integrations in minutes, not months

TRY BOOMI FREE

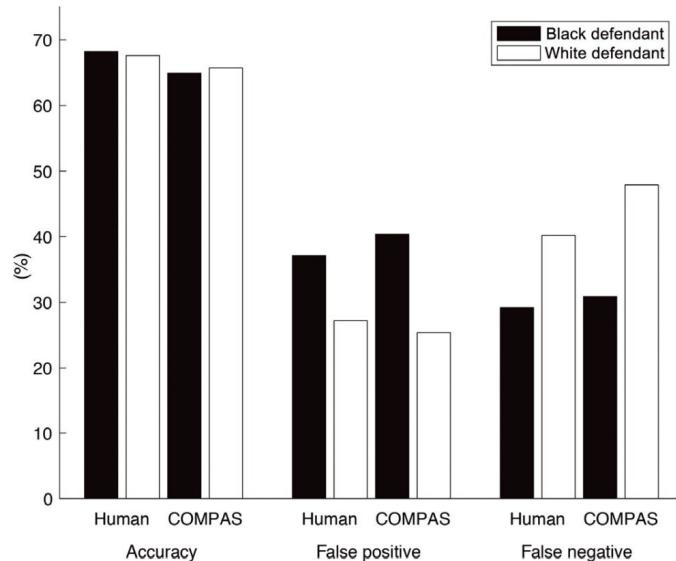


Recent Video



Criminal Justice ([Dressel et al, 2018](#))

- Commercial risk assessment software known as COMPAS
 - assess more than 1 million offenders since 2000
 - predicts a defendant's risk of committing a misdemeanor or felony 137 features



Biases in Word Embedding ([Gard et al, 2018](#))

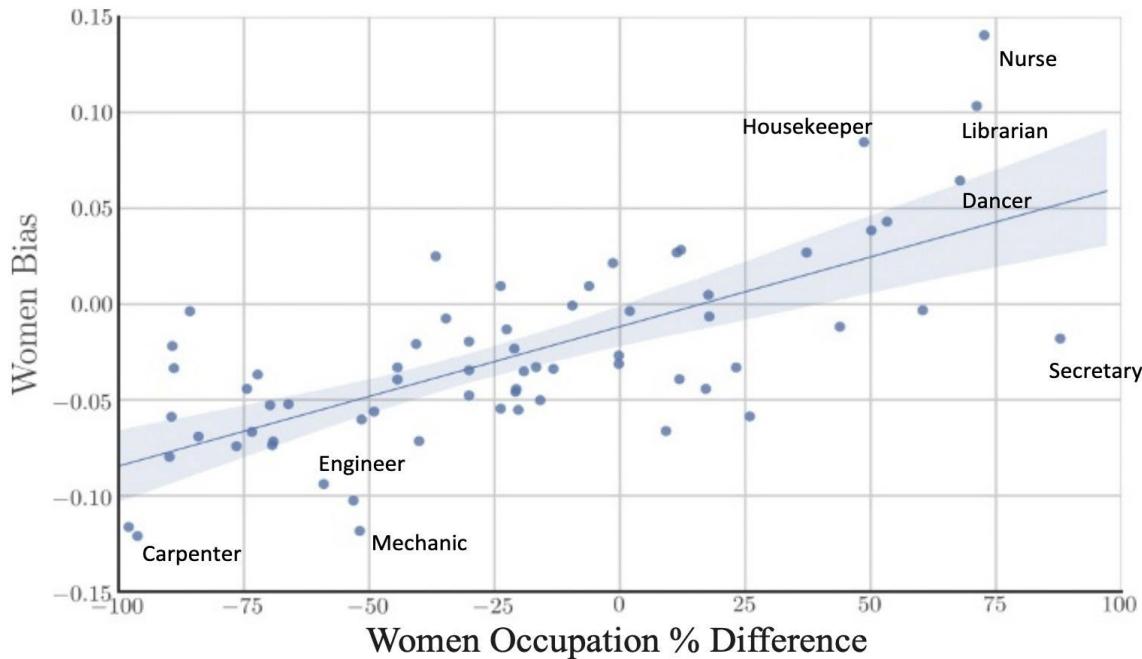
He is...



She is...

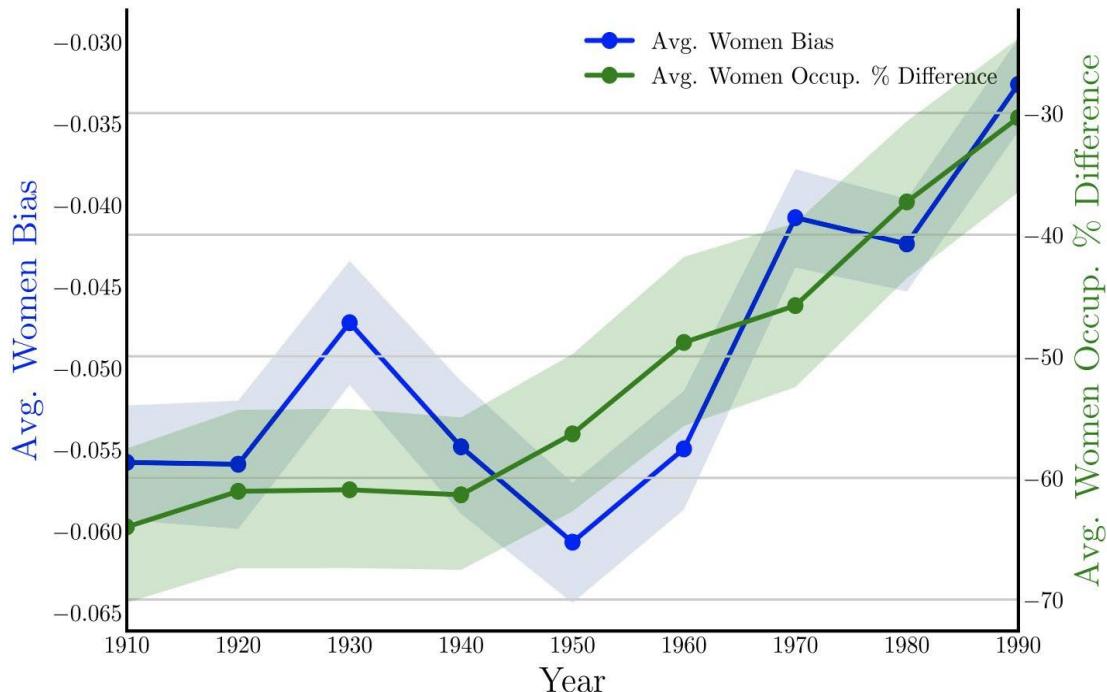


Gender Biases and Occupation



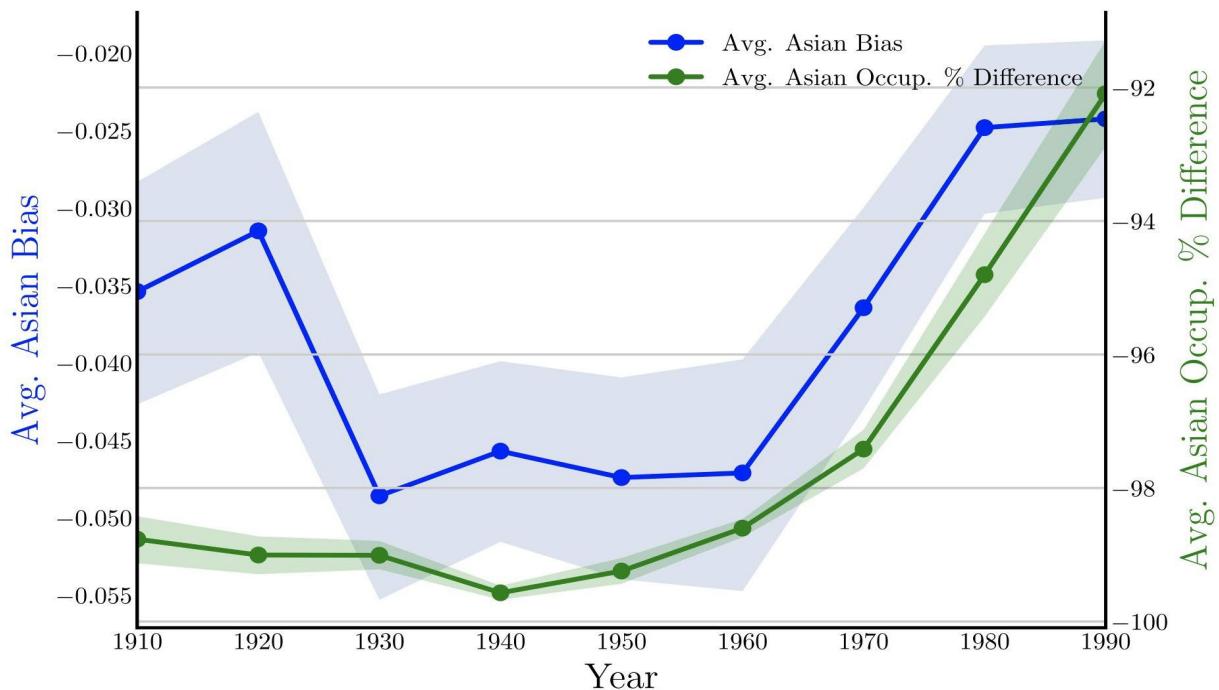
Women's occupation relative percentage vs. embedding bias in Google News vectors.
More positive indicates more associated with women on both axes. The shaded region is the 95%bootstrapped confidence interval

Average Biases Over Time for Woman



More positive means a stronger association with women. In blue is relative bias toward women in the embeddings, and in green is the average percentage of difference of women in the same occupations.

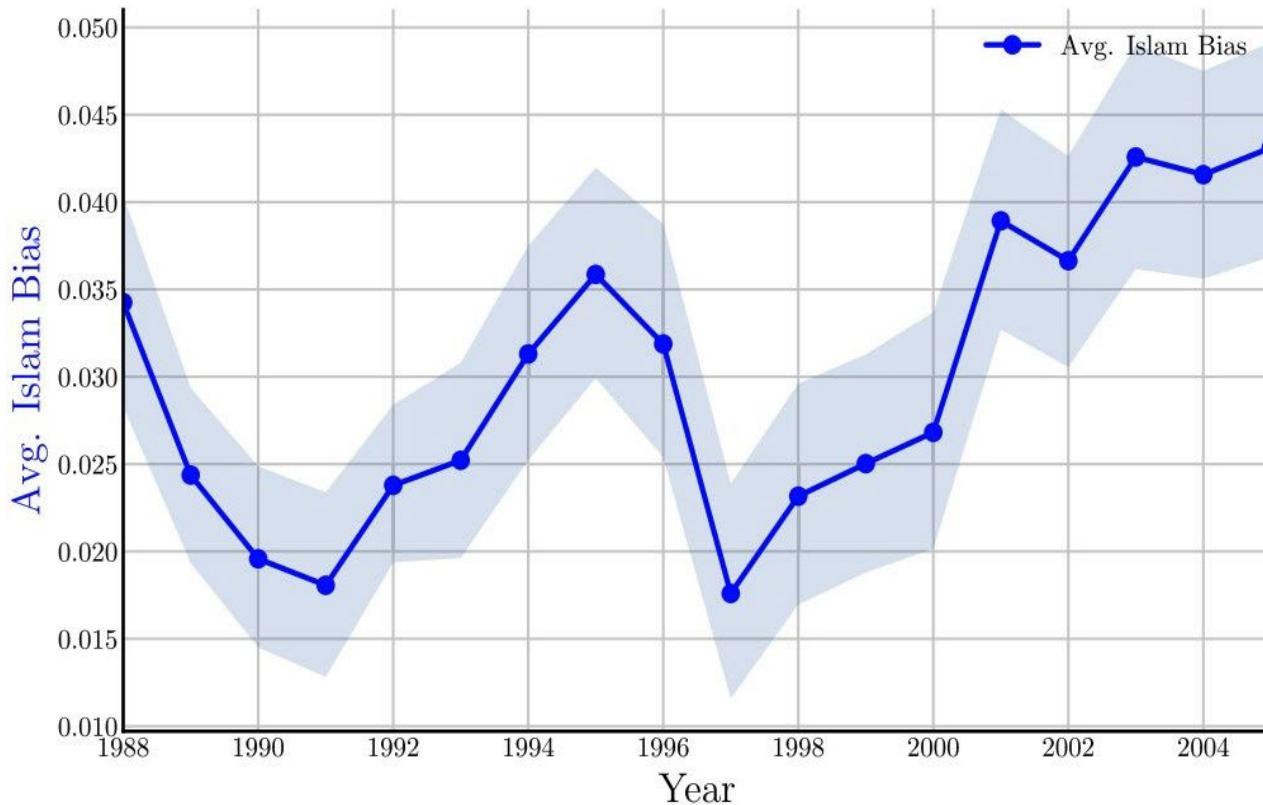
Average Biases Over Time for Asian



Top 10 Occupations and Ethnic Groups

Hispanic	Asian	White
Housekeeper	Professor	Smith
Mason	Official	Blacksmith
Artist	Secretary	Surveyor
Janitor	Conductor	Sheriff
Dancer	Physicist	Weaver
Mechanic	Scientist	Administrator
Photographer	Chemist	Mason
Baker	Tailor	Statistician
Cashier	Accountant	Clergy
Driver	Engineer	Photographer

Religious Bias Related to Terrorism



Words Projected into Gender Axes ([Bolukbasi et al, 2016](#))



Coreference Resolution ([Zhao et al, 2018](#))

	Mention	Coref
1	President is more vulnerable than most.	
2	His unorthodox and controversial style of politics creates more political incentives for Republicans to take a stand against his presidency	Coref
	Coref	

his \Rightarrow her

❖ Stereotypical dataset

The physician hired the secretary because he was overwhelmed with clients.

The physician hired the secretary because she was highly recommended.

❖ Anti-stereotypical dataset

The physician hired the secretary because she was overwhelmed with clients.

The physician hired the secretary because he was highly recommended.

Removing Gender Information ([Wang et al, 2019](#))

COCO Results



imSitu Results



Sensitive (Protected) Features

- Sensitive Features
 - Identify a group
 - e.g., gender, ethnics
- Discrimination Occurs
 - When Sensitive Features Are Used Improperly
 - May lead to ML Discrimination

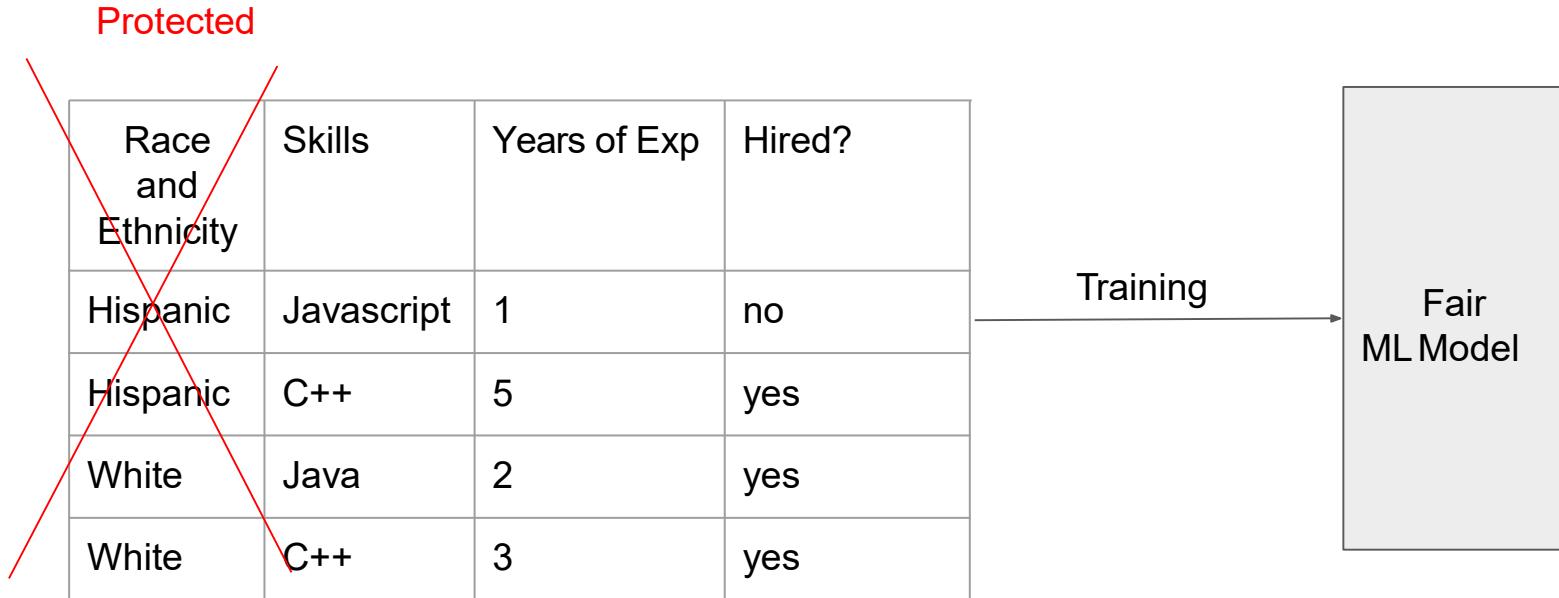
Protected Attributes Specified in US Fair Lending Laws

- Fair Housing Acts (FHA)
- Equal Credit Opportunity ACTs (ECOA)

Attribute	FHA	ECOA
Race	✓	✓
Color	✓	✓
National origin	✓	✓
Religion	✓	✓
Sex	✓	✓
Familial status	✓	
Disability	✓	
Exercised rights under CCPA		✓
Marital status		✓
Recipient of public assistance		✓
Age		✓

Fairness Through Unawareness

- A ML Algorithm Achieves Fair Through Unawareness If
 - None of the sensitive features are directly used in the model



Issues With Fairness Through Unawareness

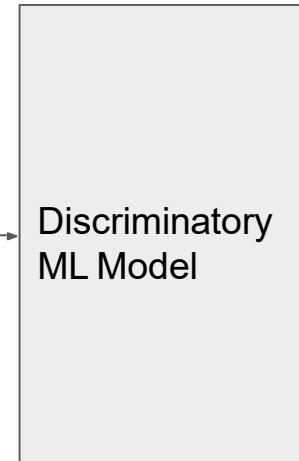
Sensitive Features May Still Be Used

- Inferred from indirect evidence

Inferred

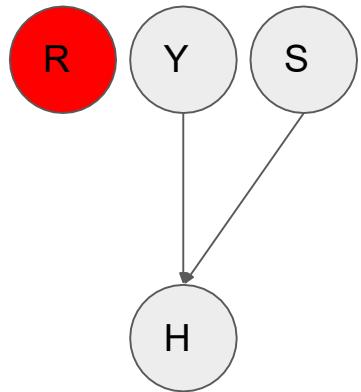
Protected Race and Ethnicity	Skills	Years of Exp	Often Goes to Mexican Markets	Hiring Decision
Hispanic	Javascript	1	yes	no
Hispanic	C++	5	yes	yes
White	Java	2	no	yes
White	C++	3	no	yes

Training

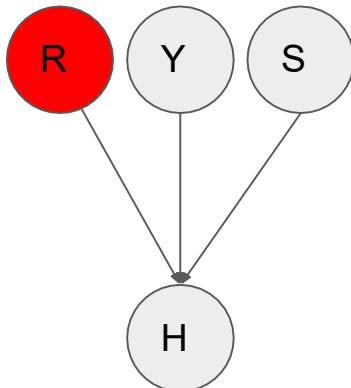


Types of Discriminations

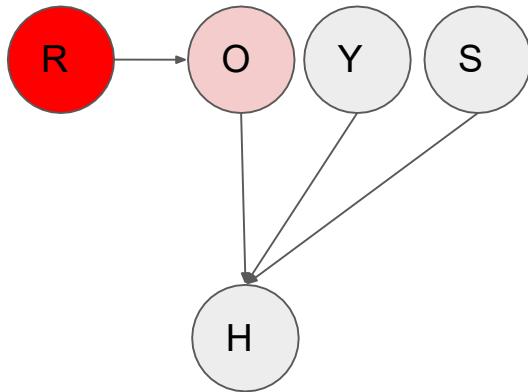
Fair ML Model



Direct Discrimination



Indirect Discrimination



R - Race

Y - Years of Exp

S = Skills

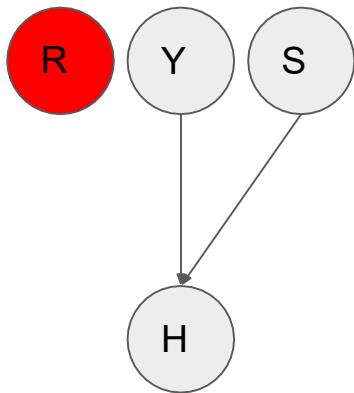
O = Often Goes to Mexico Market

Conditions for Direct Discrimination

- A - set of protected features
- X - set of features other than protected features
- A predictor \hat{Y} is direct discrimination if
 - $P(\hat{Y} | X, A) \neq P(\hat{Y} | X)$
 - i.e., $\hat{Y} \not\perp\!\!\!\perp A | X$

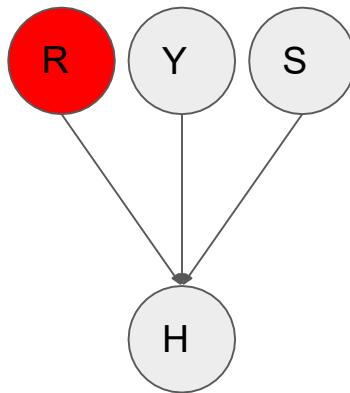
Types of Discriminations

Fair ML Model



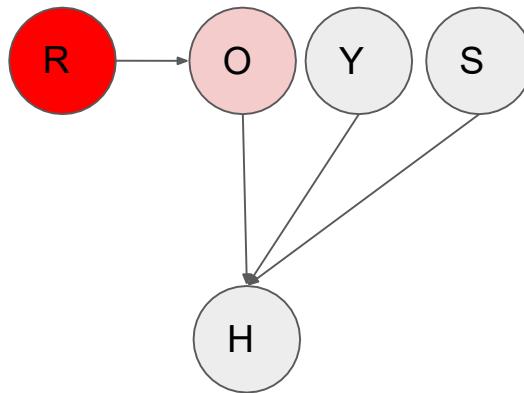
$H \perp\!\!\!\perp R | \{Y, S\}$
not connected

Direct Discrimination



$H \not\perp\!\!\!\perp R | \{Y, S\}$
(head to head)

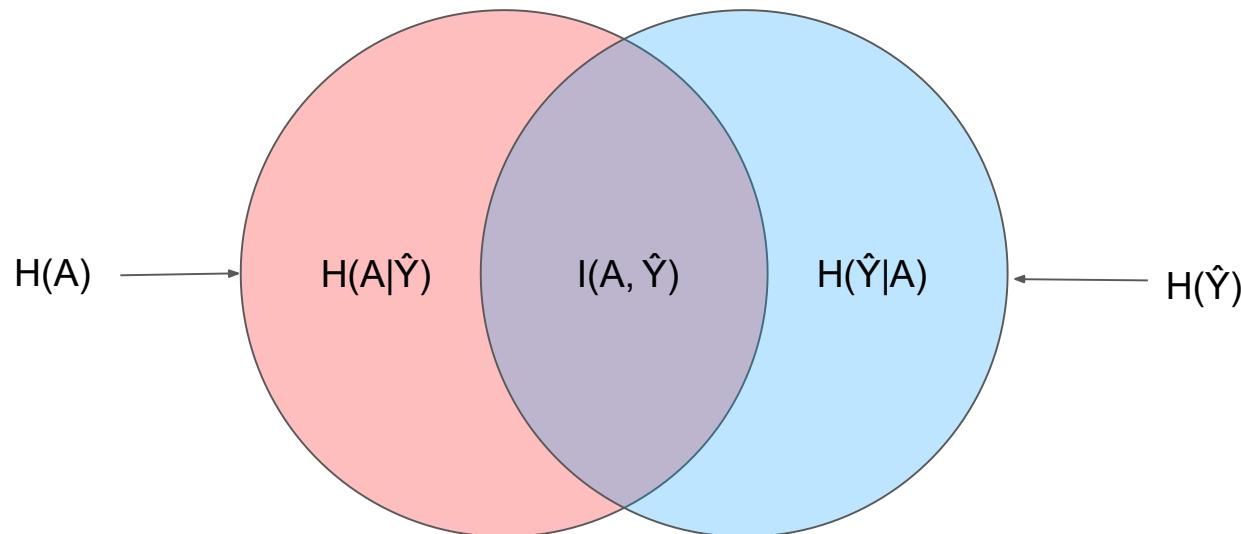
Indirect Discrimination



$H \perp\!\!\!\perp R | \{Y, S, O\}$
(head to tail)

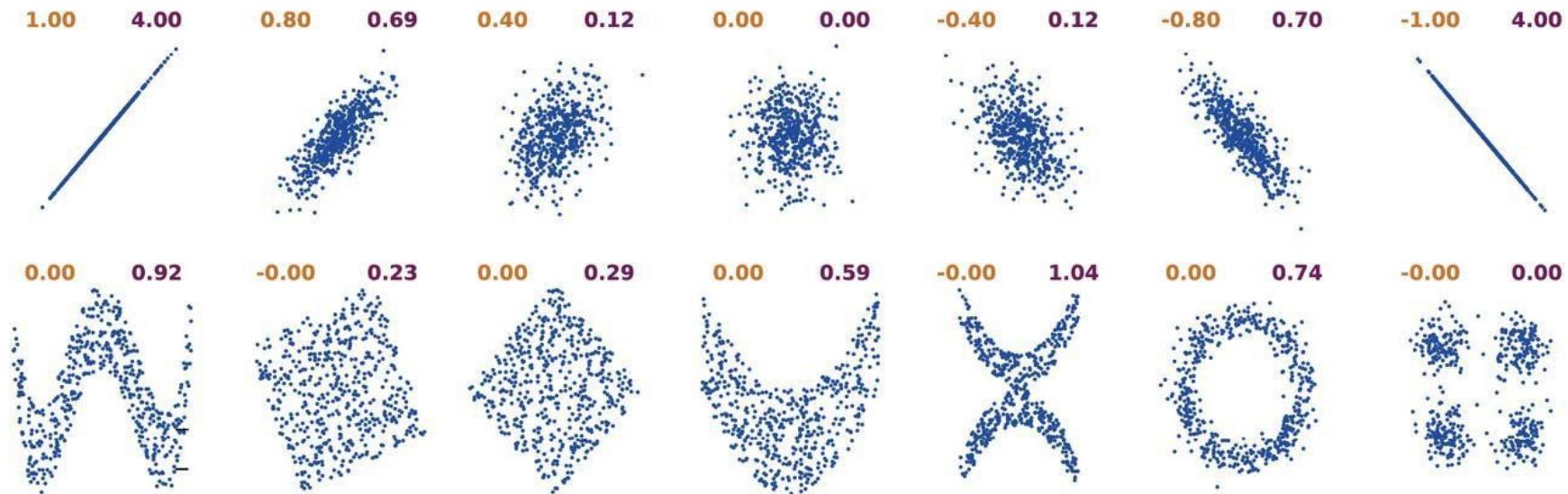
Conditions for Indirect Discrimination

- Mutual Information
 - A measure of the mutual dependence between A and \hat{Y}
 - $I(A, \hat{Y}) = H(A) - H(A | \hat{Y}) = H(\hat{Y}) - H(\hat{Y} | A)$
 - $I(A, \hat{Y}) = 0$ if $P(\hat{Y} | A) = P(\hat{Y})$, or $A \perp\!\!\!\perp \hat{Y}$



Correlation Coefficient and Mutual Information

Correlation Coefficient (left) and Mutual Information (right)



Ince et al, 2016

Limitations

- Processing Sensitive Features
 - Fairness through unawareness requires sensitive features to be masked out
 - Not easy to do in real life
 - Referred to as individual fairness criteria



❖ Stereotypical dataset

The physician hired the secretary because he was overwhelmed with clients.

The physician hired the secretary because she was highly recommended.

❖ Anti-stereotypical dataset

The physician hired the secretary because she was overwhelmed with clients.

The physician hired the secretary because he was highly recommended.

Common Fairness Criteria

Demographic Parity

Equality of Odds

Equality of Opportunity

Demographic Parity Applied to a Group

Demographic Parity Is Applied to a Group of Samples

- Does not require features to be masked out

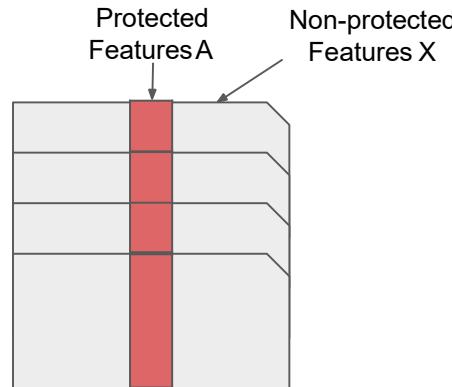
A Predictor \hat{Y} Satisfies Demographic Parity If

- The probabilities of positive predictions are the same regardless of whether the group is protected
- Protected groups are identified as $A = 1$

$$P(\hat{Y} = 1|A = 1) = P(\hat{Y} = 1|A = 0)$$

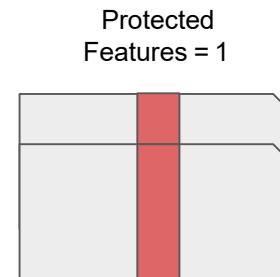
Comparisons

Individual Treatment



Fairness Through Unawareness
 $P(\hat{Y} | X)$

Group Treatment



Demographic Parity
 $P(\hat{Y}=1 | A=1)$



Demographic Parity
 $P(\hat{Y}=1 | A=0)$

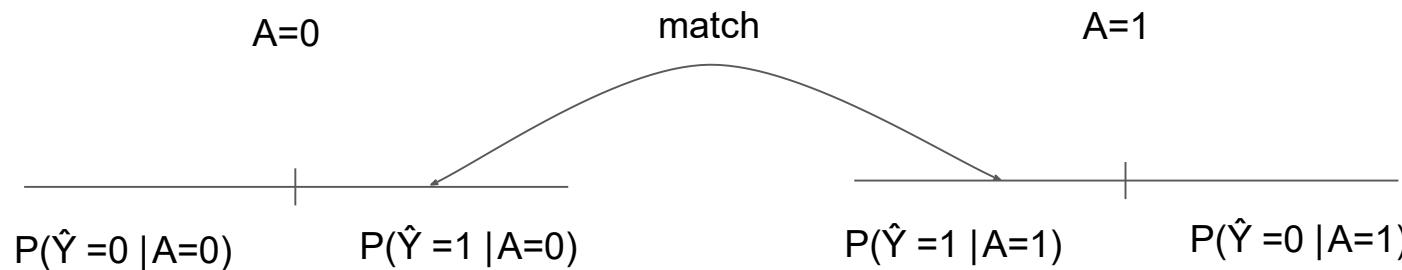
Issues With Demographic Parity

Correlates Too Much With the Performance of the Predictor

$$P(\hat{Y} = 1|A = 1) = P(\hat{Y} = 1|A = 0)$$

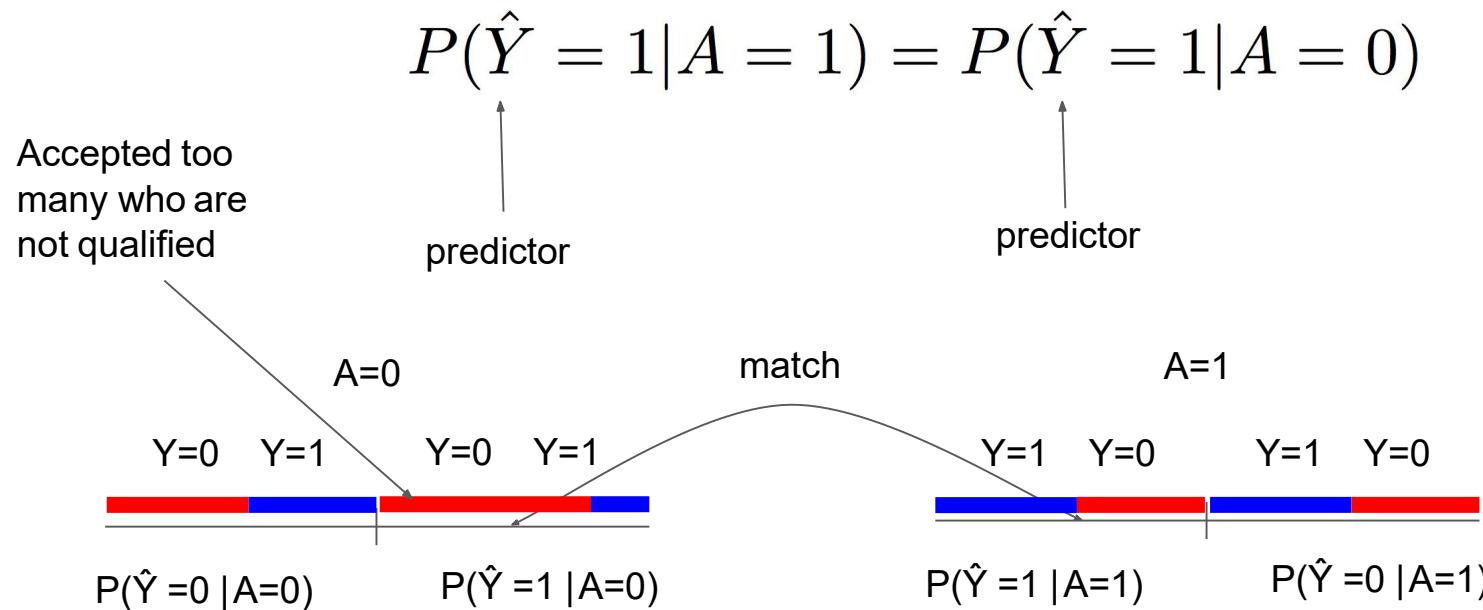
↑
predictor

↑
predictor



Issues With Demographic Parity

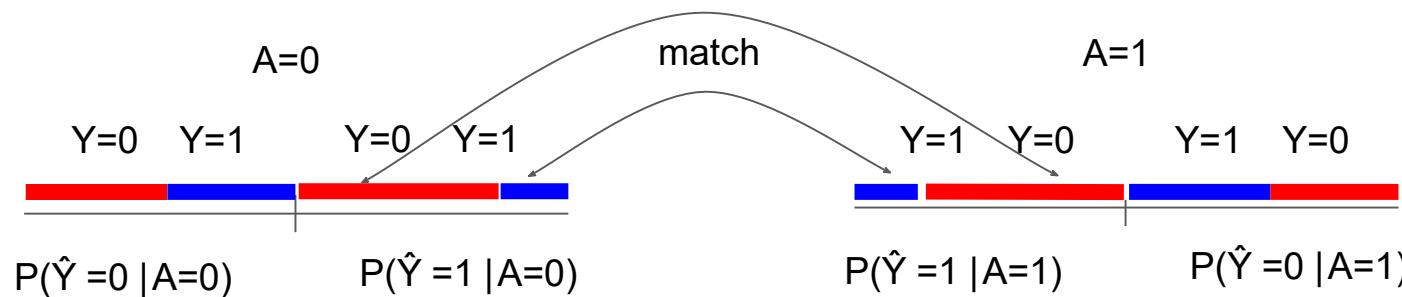
Correlates Too Much With the Performance of the Predictor



Equality of Odds

Equal Probabilities for Both Qualified/Unqualified People Across Protected Groups

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

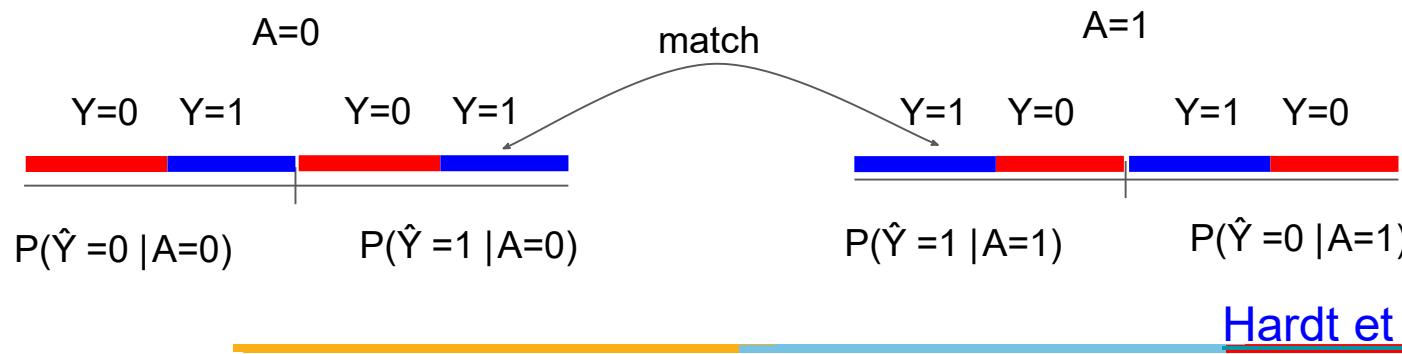


Hardt et al, 2016

Equality of Opportunity

Equal Probabilities for Qualified People Across Protected Groups

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



Practice Question

Find out the Fairness Criteria that \hat{Y}_1 , and \hat{Y}_2 Satisfy

- $A = \{\text{race}\}$, $Y = \{\text{Hiring Decision}\}$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Demographic Parity for Predictor \hat{Y}_1

$$P(\hat{Y}_1 = 1 | R = H) = 2/3$$

$$P(\hat{Y}_1 = 1 | R = W) =$$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Demographic Parity for Predictor \hat{Y}_1

$$P(\hat{Y}_1 = 1 | R = H) = 2/3$$

$$P(\hat{Y}_1 = 1 | R = W) = 2/3$$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Demographic Parity for Predictor \hat{Y}_1

- $P(\hat{Y}_1 = 1 | R = H) = 2/3$
- $P(\hat{Y}_1 = 1 | R = W) = 2/3$



Demographics

$$P(\hat{Y} = \text{Parity} | A = 1) = P(\hat{Y} = 1 | A = 0)$$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_1

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = 1$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) =$
- $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) =$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) =$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_1

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = 1$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) = 0.5$
- $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) =$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) =$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_1

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = 1$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) = 0.5$
- $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) = 0$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) =$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_1

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = 1$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) = 0.5$
- $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) = 0$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) = 1$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_1

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = 1$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) = 0.5$
- $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) = 0$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) = 1$



\times Equality of Opportunity

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

\times Equality of Odds

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

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Demographic Parity for Predictor \hat{Y}_2

- $P(\hat{Y}_1 = 1 | R = H) = 2/3$
- $P(\hat{Y}_1 = 1 | R = W) =$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Demographic Parity for Predictor \hat{Y}_2

- $P(\hat{Y}_1 = 1 | R = H) = 2/3$
- $P(\hat{Y}_1 = 1 | R = W) = 1/3$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Demographic Parity for Predictor \hat{Y}_2

- $P(\hat{Y}_1 = 1 | R = H) = 2/3$
- $P(\hat{Y}_1 = 1 | R = W) = 1/3$

X Demographics Parity

$$P(\hat{Y} = 1 | A = 1) = P(\hat{Y} = 1 | A = 0)$$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_2

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = 1/2$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) =$
- $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) =$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) =$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Demographic Parity for Predictor \hat{Y}_2

- $P(\hat{Y}_1 = 1 | R = H) = 2/3$
- $P(\hat{Y}_1 = 1 | R = W) = 1/3$

X Demographics Parity

$$P(\hat{Y} = 1 | A = 1) = P(\hat{Y} = 1 | A = 0)$$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_2

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = 1/2$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) = 1/2$
- $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) =$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) =$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_2

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = 1/2$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) = 1/2$
- $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) = 1$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) =$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_2

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = \frac{1}{2}$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) = \frac{1}{2}$
- $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) = 1$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) = 0$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_2

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = 1/2$
 - $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) = 1/2$
 - $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) = 1$
 - $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) = 0$
- ✓ ✓ Equality of Opportunity $P(\hat{Y} = 1 | A = 1, Y = 1) = P(\hat{Y} = 1 | A = 0, Y)$
✗ ✗ Equality of Odds $P(\hat{Y} = 1 | A = 0, Y) = P(\hat{Y} = 1 | A = 1, Y)$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Summary of Fairness Criteria

Fairness Criteria	Criteria	Group	Individual
Unawareness	Excludes A in Predictions		✓
Demographic Parity	$P(\hat{Y} = 1 A = 0) = P(\hat{Y} = 1 A = 1)$	✓	
Equalized Odds	$P(\hat{Y} = 1 A = 0, Y) = P(\hat{Y} = 1 A = 1, Y)$	✓	
Equalized Opportunity	$P(\hat{Y} = 1 A = 0, Y = 1) = P(\hat{Y} = 1 A = 1, Y = 1)$	✓	

Required Reading

- Barocas: Ch 2
- Bishop: Ch 8.2
- <https://www.borealisai.com/research-blogs/tutorial1-bias-and-fairness-ai/>

Recommended

- Papers mentioned on the slides

Bishop, Christopher M. Pattern recognition and machine learning. Springer, 2006.

Barocas, S., Hardt, M., & Narayanan, A. Fairness and Machine Learning, 2018.

Additional Reading

- Gajane, Pratik, and Mykola Pechenizkiy. On formalizing fairness in prediction with machine learning. arXiv 2017
- Luong, Binh Thanh, Salvatore Ruggieri, and Franco Turini. k-NN as an implementation of situation testing for discrimination discovery and prevention. SIGKDD 2011
- Hardt, M., Price, E., & Srebro, N. Equality of opportunity in supervised learning. NeurIPS 2016
- Pedreshi, Dino, Salvatore Ruggieri, and Franco Turini. Discrimination-aware data mining. SIGKDD 2008
- Zafar, Muhammad Bilal, et al. Fairness Constraints: Mechanisms for Fair Classification. AIStats 2017



BITS Pilani
Pilani Campus

Fair, Accountable, Transparent Machine Learning (FAccT ML) ZG517

Dr. Sugata Ghosal
Sugata.ghosal@pilani.bits-pilani.ac.in



**Session 2
Date – 28th May 2023
Time – 8:45 AM to 10:45 PM**

These slides are prepared by the instructor, with grateful acknowledgement of many others who made their course materials freely available online.

Session Content

- Real world examples of data bias
 - Gender, occupation
- Fairness Measures
 - Individual based
 - Through unawareness
 - Group based
 - Demographic parity
 - Equality of Odds
 - Equality of Opportunity
- Problem Solving

Biases in Word Embedding ([Gard et al, 2018](#))

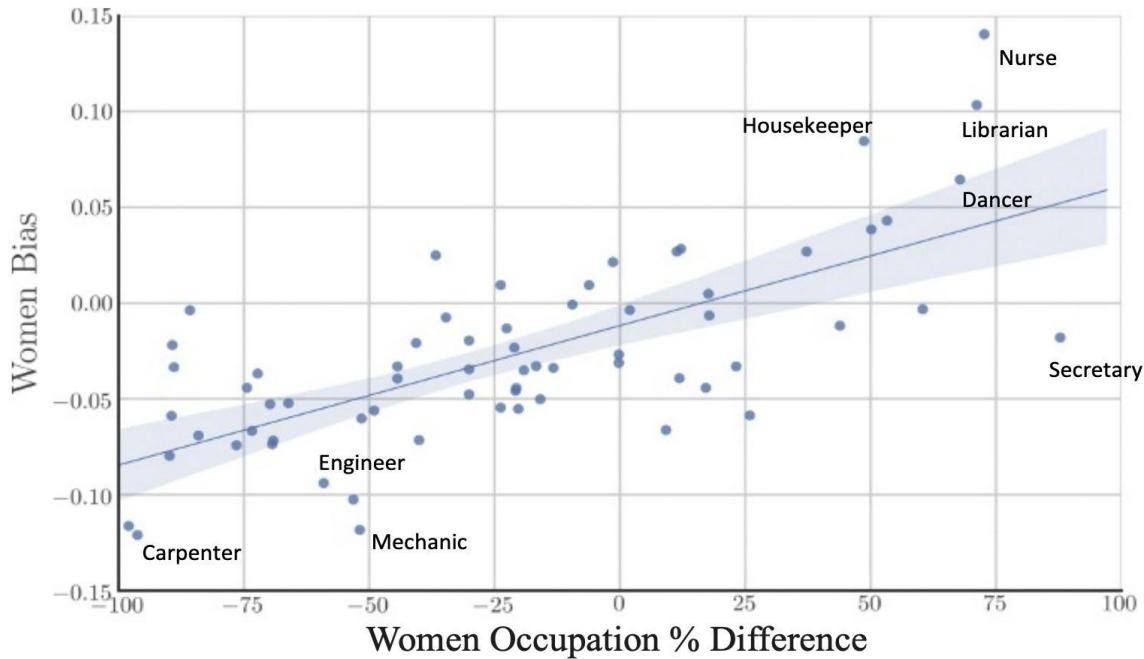
He is...



She is...

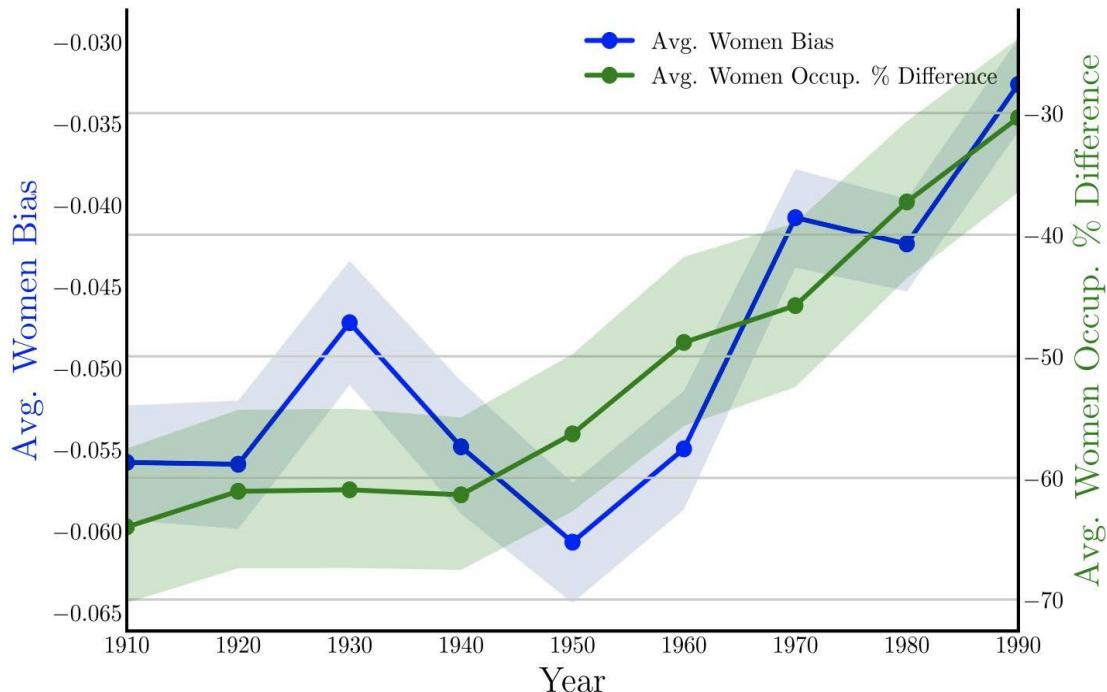


Gender Biases and Occupation



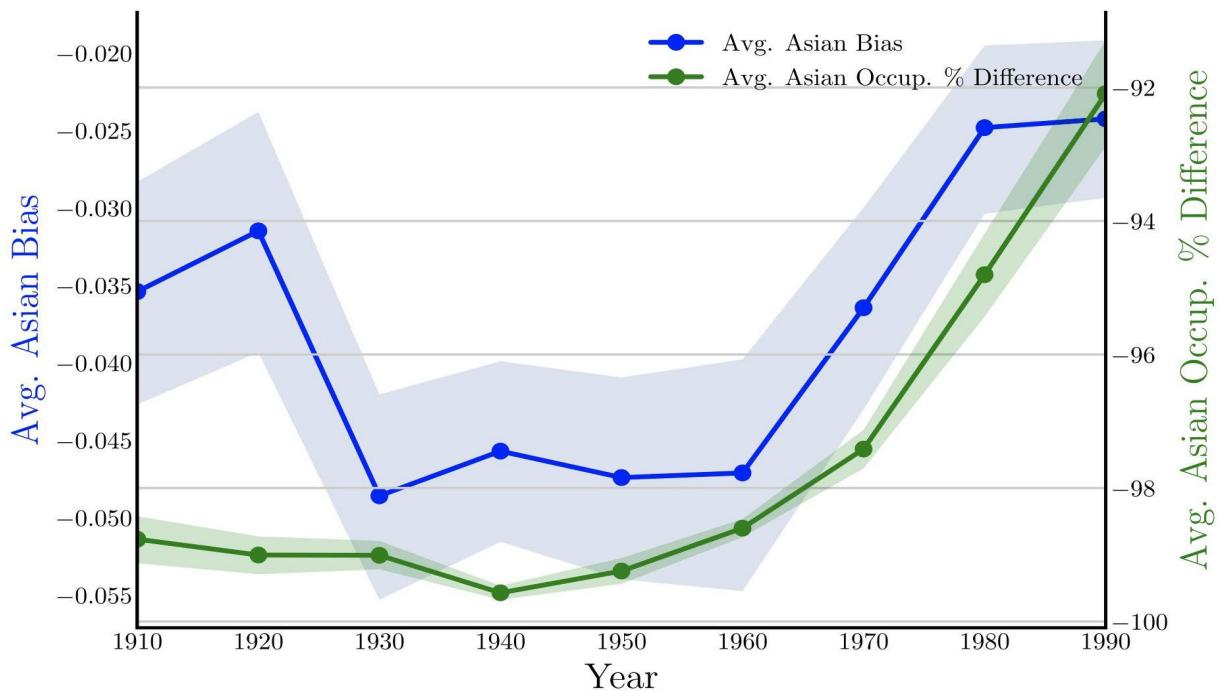
Women's occupation relative percentage vs. embedding bias in Google News vectors.
More positive indicates more associated with women on both axes. The shaded region is the 95%bootstrapped confidence interval

Average Biases Over Time for Woman



More positive means a stronger association with women. In blue is relative bias toward women in the embeddings, and in green is the average percentage of difference of women in the same occupations.

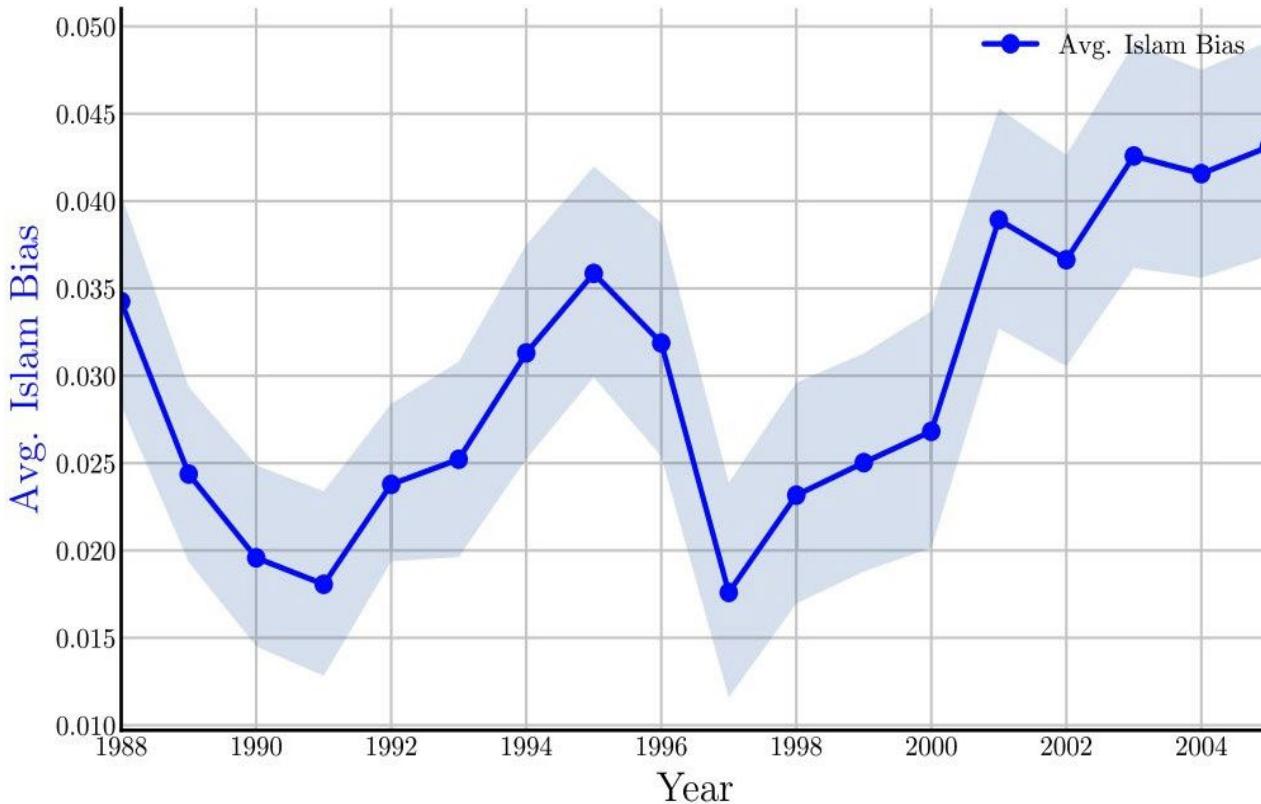
Average Biases Over Time for Asian



Top 10 Occupations and Ethnic Groups

Hispanic	Asian	White
Housekeeper	Professor	Smith
Mason	Official	Blacksmith
Artist	Secretary	Surveyor
Janitor	Conductor	Sheriff
Dancer	Physicist	Weaver
Mechanic	Scientist	Administrator
Photographer	Chemist	Mason
Baker	Tailor	Statistician
Cashier	Accountant	Clergy
Driver	Engineer	Photographer

Religious Bias Related to Terrorism



Words Projected into Gender Axes ([Bolukbasi et al, 2016](#))



Coreference Resolution ([Zhao et al, 2018](#))

	Mention	Coref
1	President is more vulnerable than most.	
2	His unorthodox and controversial style of politics creates more political incentives for Republicans to take a stand against his presidency	Coref
	Coref	

his \Rightarrow her

❖ Stereotypical dataset

The physician hired the secretary because he was overwhelmed with clients.

The physician hired the secretary because she was highly recommended.

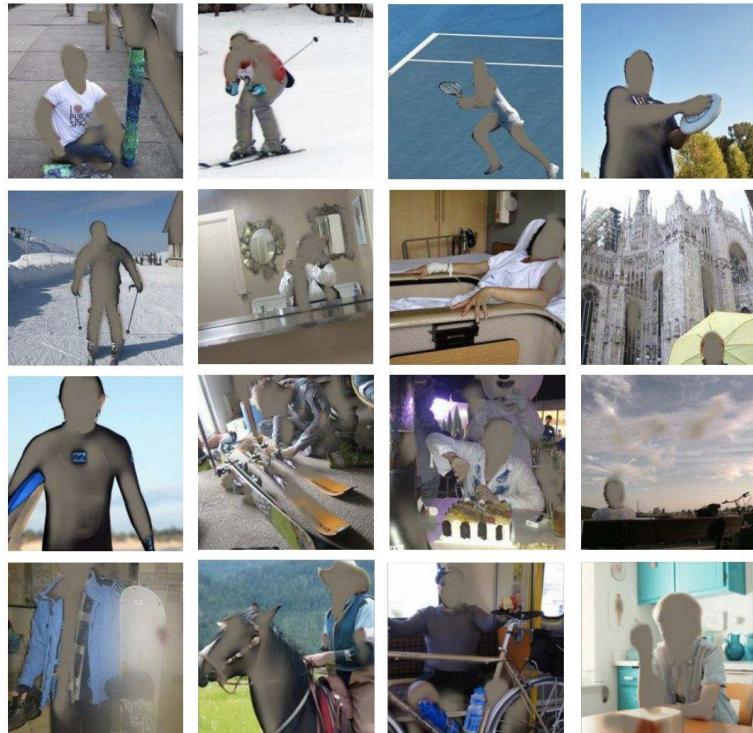
❖ Anti-stereotypical dataset

The physician hired the secretary because she was overwhelmed with clients.

The physician hired the secretary because he was highly recommended.

Removing Gender Information ([Wang et al, 2019](#))

COCO Results



imSitu Results



Sensitive (Protected) Features

- Sensitive Features
 - Identify a group
 - e.g., gender, ethnics
- Discrimination Occurs
 - When Sensitive Features Are Used Improperly
 - May lead to ML Discrimination

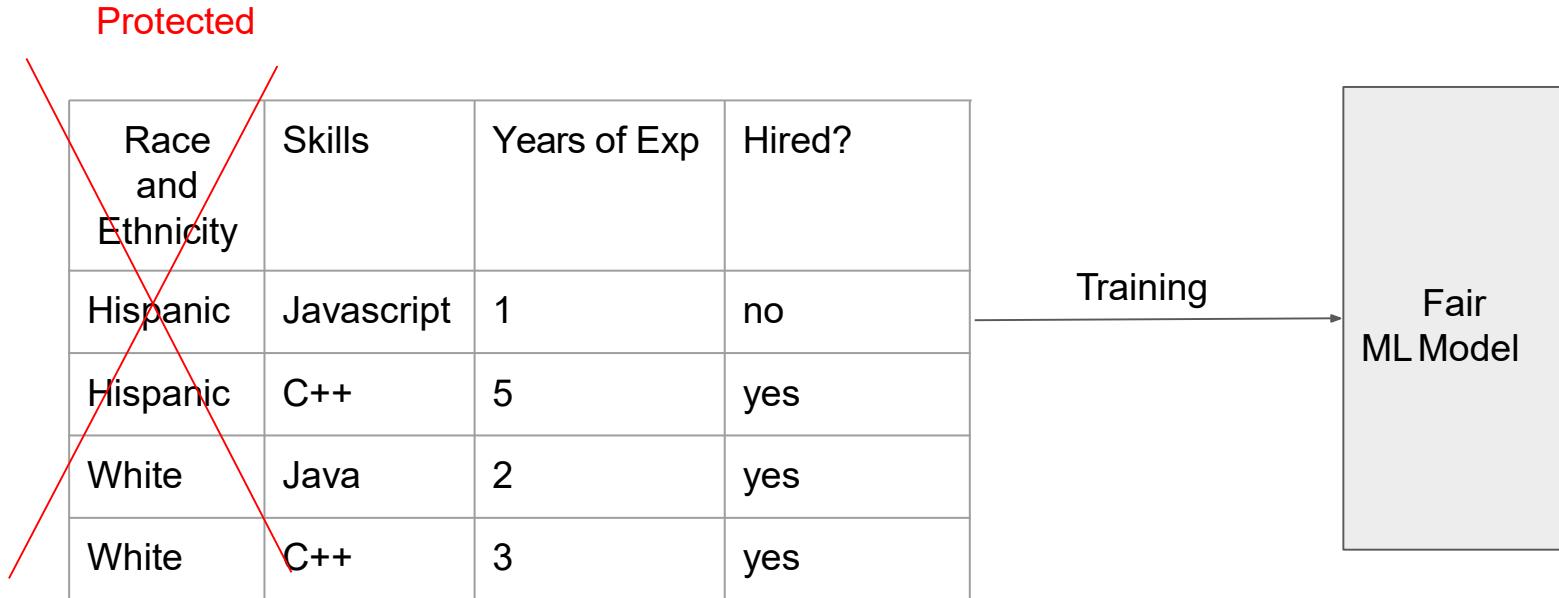
Protected Attributes Specified in US Fair Lending Laws

- Fair Housing Acts (FHA)
- Equal Credit Opportunity ACTs (ECOA)

Attribute	FHA	ECOA
Race	✓	✓
Color	✓	✓
National origin	✓	✓
Religion	✓	✓
Sex	✓	✓
Familial status	✓	
Disability	✓	
Exercised rights under CCPA		✓
Marital status		✓
Recipient of public assistance		✓
Age		✓

Fairness Through Unawareness

- A ML Algorithm Achieves Fair Through Unawareness If
 - None of the sensitive features are directly used in the model



Issues With Fairness Through Unawareness

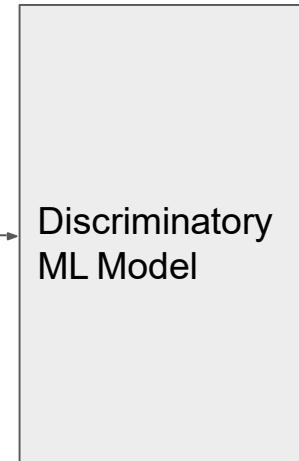
Sensitive Features May Still Be Used

- Inferred from indirect evidence

Inferred

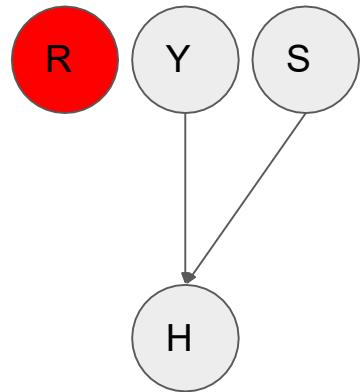
Protected Race and Ethnicity	Skills	Years of Exp	Often Goes to Mexican Markets	Hiring Decision
Hispanic	Javascript	1	yes	no
Hispanic	C++	5	yes	yes
White	Java	2	no	yes
White	C++	3	no	yes

Training

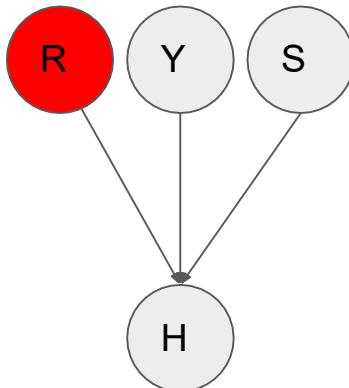


Types of Discriminations

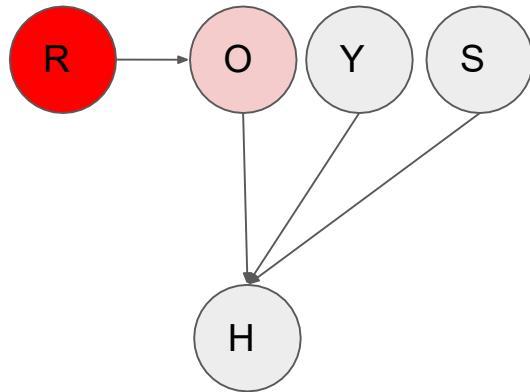
Fair ML Model



Direct Discrimination



Indirect Discrimination



R - Race

Y - Years of Exp

S = Skills

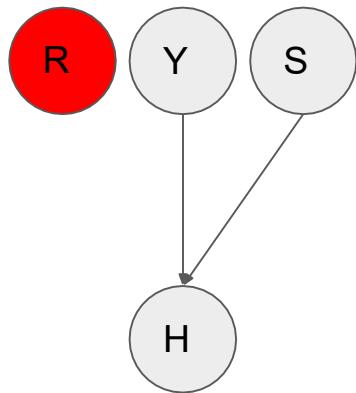
O = Often Goes to Mexico Market

Conditions for Direct Discrimination

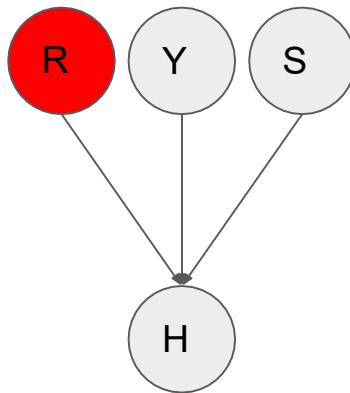
- A - set of protected features
- X - set of features other than protected features
- A predictor \hat{Y} is direct discrimination if
 - $P(\hat{Y} | X, A) \neq P(\hat{Y} | X)$
 - i.e., $\hat{Y} \not\perp\!\!\!\perp A | X$

Types of Discriminations

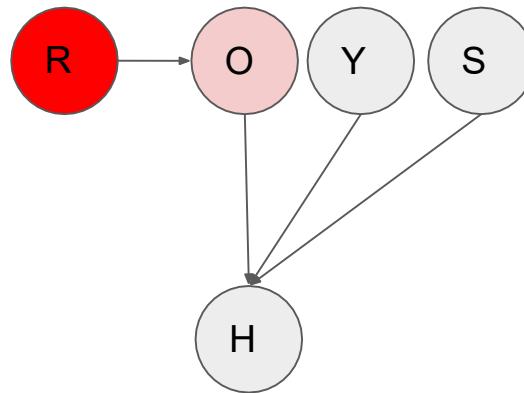
Fair ML Model



Direct Discrimination



Indirect Discrimination



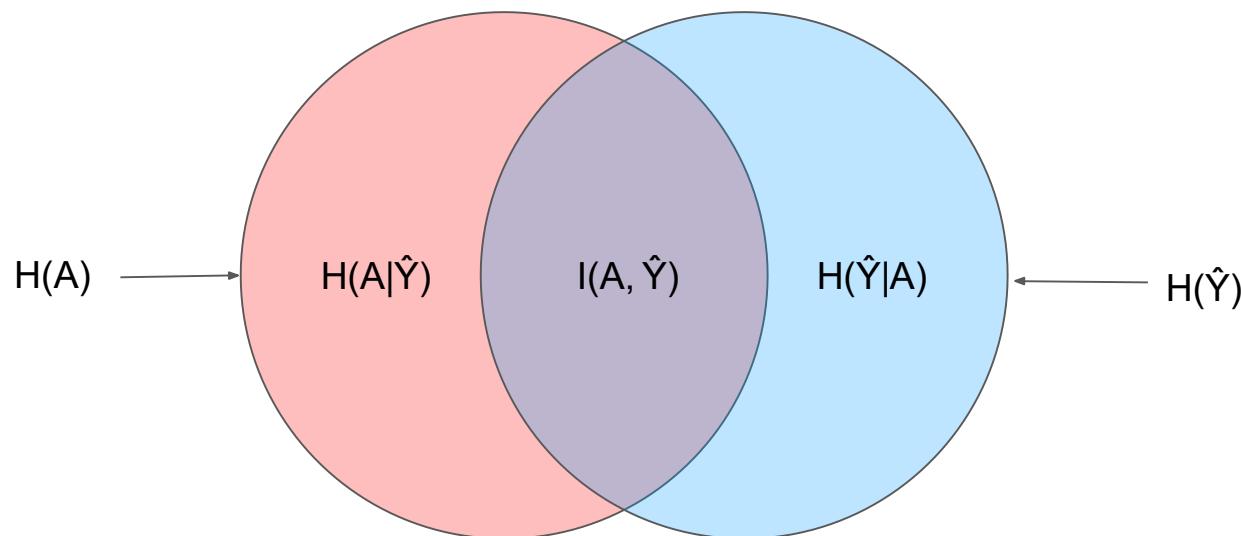
$$H \perp\!\!\!\perp R \mid \{Y, S\}$$

$$H \not\perp\!\!\!\perp R \mid \{Y, S\}$$

$$H \perp\!\!\!\perp R \mid \{Y, S, O\}$$

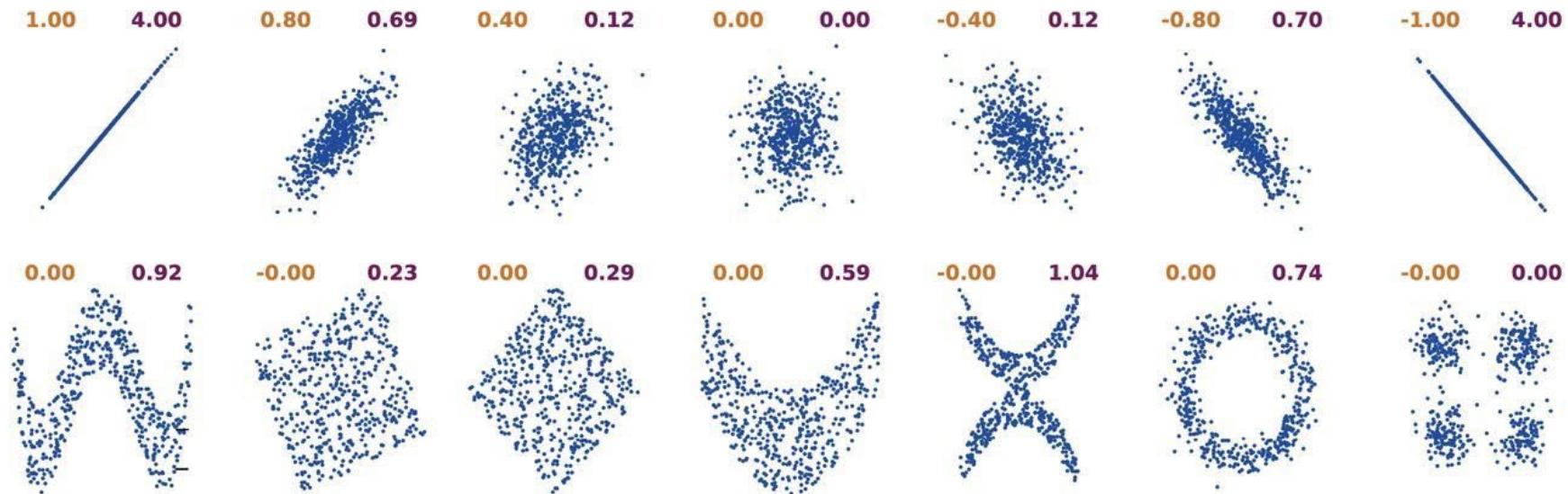
Conditions for Indirect Discrimination

- Mutual Information
 - A measure of the mutual dependence between A and \hat{Y}
 - $I(A, \hat{Y}) = H(A) - H(A | \hat{Y}) = H(\hat{Y}) - H(\hat{Y} | A)$
 - $I(A, \hat{Y}) = 0$ if $P(\hat{Y} | A) = P(\hat{Y})$, or $A \perp\!\!\!\perp \hat{Y}$



Correlation Coefficient and Mutual Information

Correlation Coefficient (left) and Mutual Information (right)



Ince et al, 2016

Limitations

- Processing Sensitive Features
 - Fairness through unawareness requires sensitive features to be masked out
 - Not easy to do in real life
 - Referred to as individual fairness criteria



❖ Stereotypical dataset

The physician hired the secretary because he was overwhelmed with clients.

The physician hired the secretary because she was highly recommended.

❖ Anti-stereotypical dataset

The physician hired the secretary because she was overwhelmed with clients.

The physician hired the secretary because he was highly recommended.

Common Fairness Criteria

Demographic Parity

Equality of Odds

Equality of Opportunity

Demographic Parity Applied to a Group

Demographic Parity Is Applied to a Group of Samples

- Does not require features to be masked out

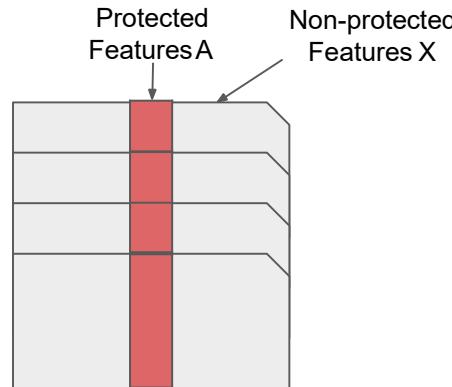
A Predictor \hat{Y} Satisfies Demographic Parity If

- The probabilities of positive predictions are the same regardless of whether the group is protected
- Protected groups are identified as $A = 1$

$$P(\hat{Y} = 1|A = 1) = P(\hat{Y} = 1|A = 0)$$

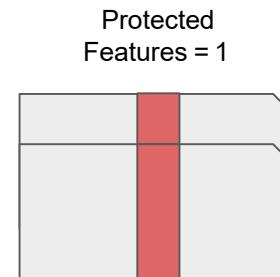
Comparisons

Individual Treatment

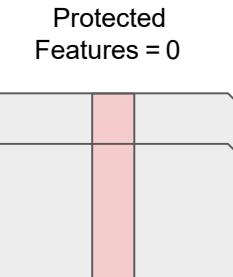


Fairness Through Unawareness
 $P(\hat{Y} | X)$

Group Treatment



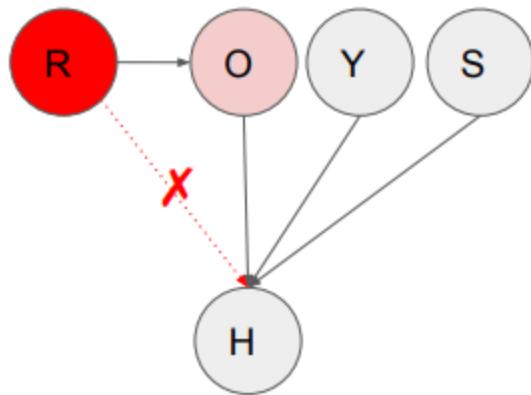
Demographic Parity
 $P(\hat{Y}=1 | A=1)$



Demographic Parity
 $P(\hat{Y}=1 | A=0)$

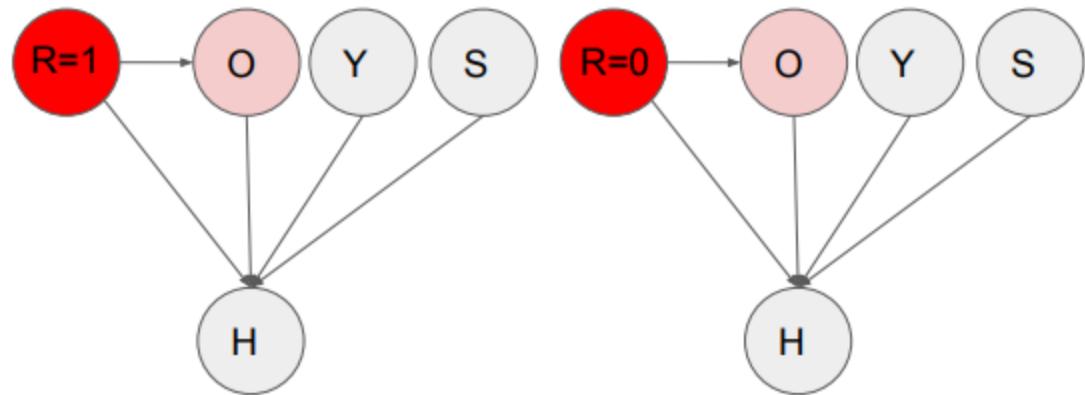
Graphical Model Explanations

Individual Treatment



$$P(H | O, Y, S)$$

Group Treatment

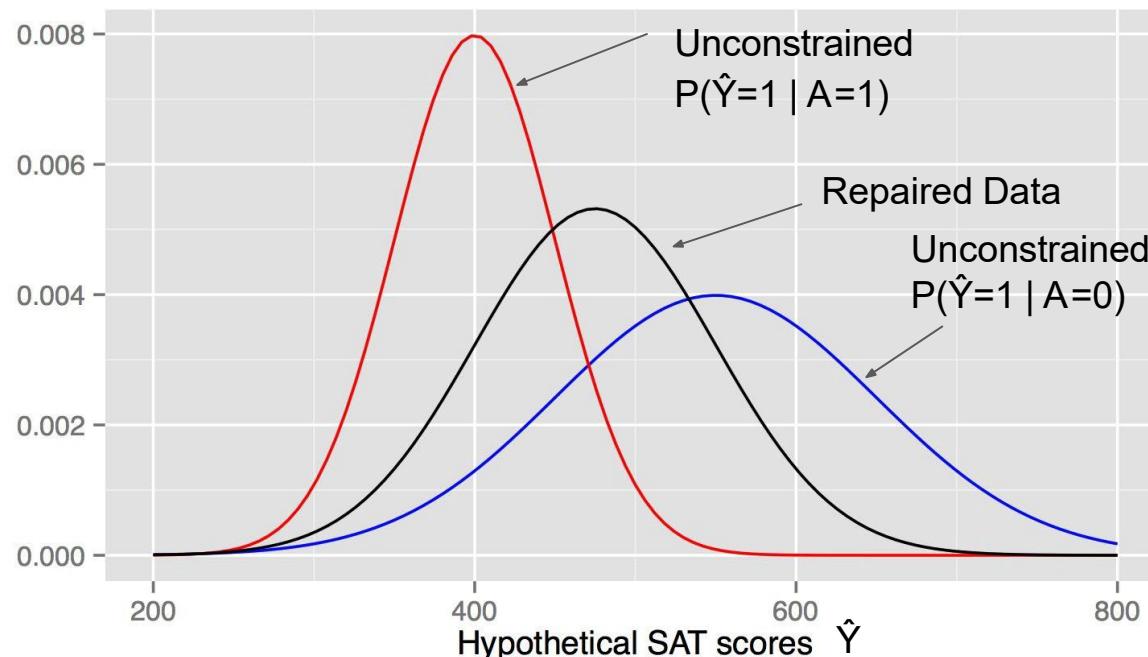


$$P(H = 1 | R=1)$$

=

$$P(H = 1 | R=0)$$

SAT Score Prediction



[Feldman et al, 2015](#)

The blue curve shows the distribution of SAT scores (Y) for $X = \text{female}$, with $\mu = 550$, $\sigma = 100$, while the red curve shows the distribution of SAT scores for $X = \text{male}$, with $\mu = 400$, $\sigma = 50$. The resulting fully repaired data is the distribution in black, with $\mu = 475$, $\sigma = 75$. Male students who originally had scores in the 95th percentile, i.e., had scores of 500, are given scores of 625 in the 95th percentile of the new distribution in Y^* , while women with scores of 625 in Y^* originally had scores of 750.

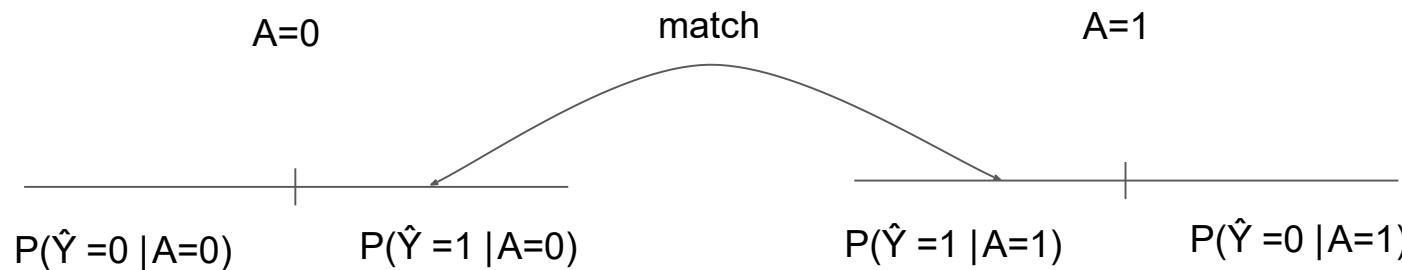
Issues With Demographic Parity

Correlates Too Much With the Performance of the Predictor

$$P(\hat{Y} = 1|A = 1) = P(\hat{Y} = 1|A = 0)$$

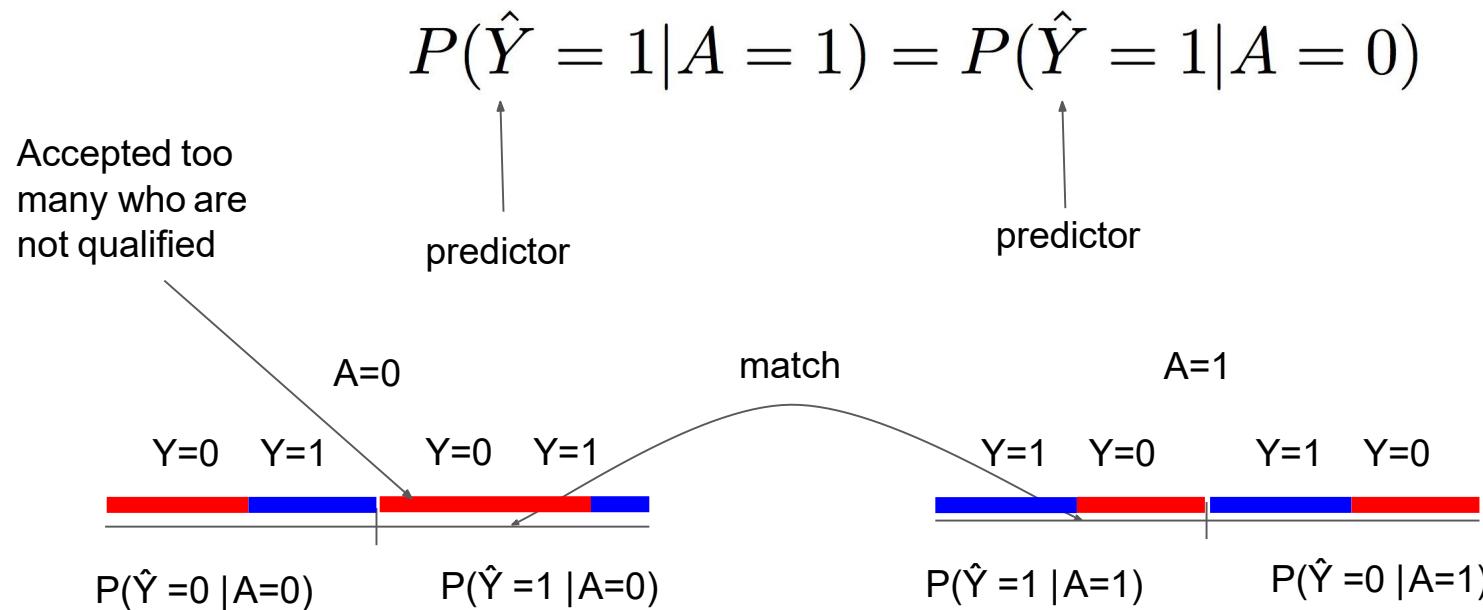
↑
predictor

↑
predictor



Issues With Demographic Parity

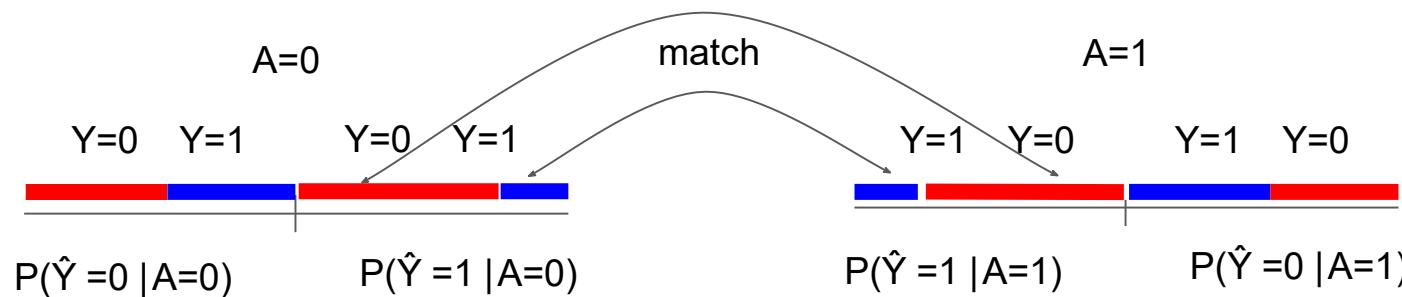
Correlates Too Much With the Performance of the Predictor



Equality of Odds

Equal Probabilities for Both Qualified/Unqualified People Across Protected Groups

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

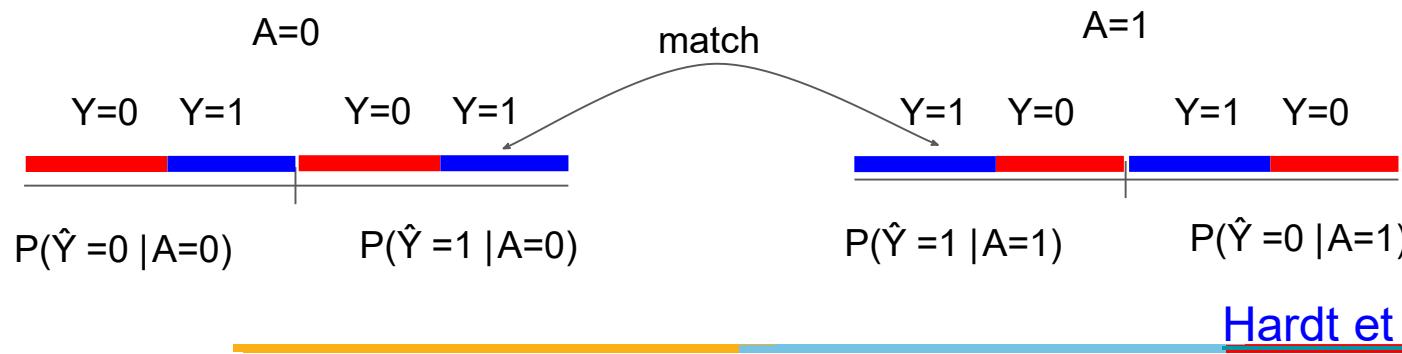


Hardt et al, 2016

Equality of Opportunity

Equal Probabilities for Qualified People Across Protected Groups

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



Practice Question

Find out the Fairness Criteria that \hat{Y}_1 , and \hat{Y}_2 Satisfy

- $A = \{\text{race}\}$, $Y = \{\text{Hiring Decision}\}$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Demographic Parity for Predictor \hat{Y}_1

$$P(\hat{Y}_1 = 1 | R = H) = 2/3$$

$$P(\hat{Y}_1 = 1 | R = W) =$$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Demographic Parity for Predictor \hat{Y}_1

$$P(\hat{Y}_1 = 1 | R = H) = 2/3$$

$$P(\hat{Y}_1 = 1 | R = W) = 2/3$$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Demographic Parity for Predictor \hat{Y}_1

- $P(\hat{Y}_1 = 1 | R = H) = 2/3$
- $P(\hat{Y}_1 = 1 | R = W) = 2/3$



Demographics

$$P(\hat{Y} = \text{Parity} | A = 1) = P(\hat{Y} = 1 | A = 0)$$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_1

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = 1$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) =$
- $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) =$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) =$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_1

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = 1$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) = 0.5$
- $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) =$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) =$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_1

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = 1$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) = 0.5$
- $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) = 0$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) =$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_1

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = 1$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) = 0.5$
- $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) = 0$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) = 1$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_1

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = 1$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) = 0.5$
- $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) = 0$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) = 1$



\times Equality of Opportunity

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

\times Equality of Odds

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

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Demographic Parity for Predictor \hat{Y}_2

- $P(\hat{Y}_1 = 1 | R = H) = 2/3$
- $P(\hat{Y}_1 = 1 | R = W) =$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Demographic Parity for Predictor \hat{Y}_2

- $P(\hat{Y}_1 = 1 | R = H) = 2/3$
- $P(\hat{Y}_1 = 1 | R = W) = 1/3$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Demographic Parity for Predictor \hat{Y}_2

- $P(\hat{Y}_1 = 1 | R = H) = 2/3$
- $P(\hat{Y}_1 = 1 | R = W) = 1/3$

X Demographics Parity

$$P(\hat{Y} = 1 | A = 1) = P(\hat{Y} = 1 | A = 0)$$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_2

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = 1/2$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) =$
- $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) =$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) =$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Demographic Parity for Predictor \hat{Y}_2

- $P(\hat{Y}_1 = 1 | R = H) = 2/3$
- $P(\hat{Y}_1 = 1 | R = W) = 1/3$

X Demographics Parity

$$P(\hat{Y} = 1 | A = 1) = P(\hat{Y} = 1 | A = 0)$$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_2

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = 1/2$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) = 1/2$
- $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) =$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) =$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_2

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = 1/2$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) = 1/2$
- $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) = 1$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) =$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_2

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = \frac{1}{2}$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) = \frac{1}{2}$
- $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) = 1$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) = 0$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

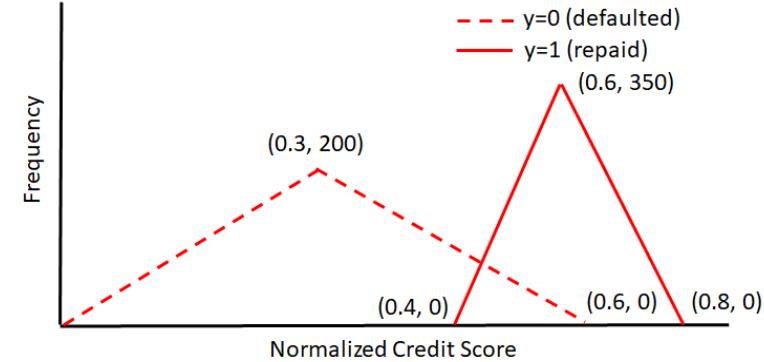
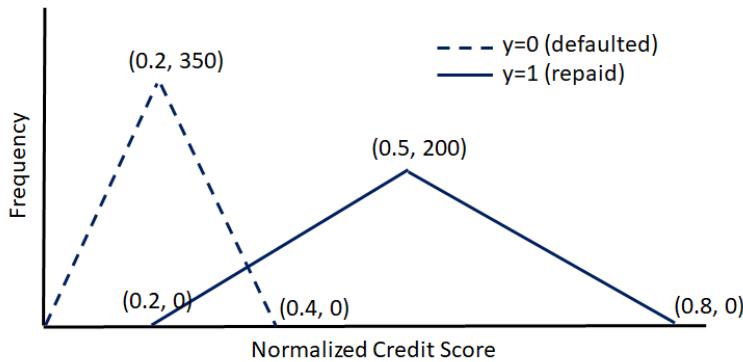
Equality of Opportunity/Odds for \hat{Y}_2

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = 1/2$
 - $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) = 1/2$
 - $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) = 1$
 - $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) = 0$
- ✓ ✓ Equality of Opportunity $P(\hat{Y} = 1 | A = 1, Y = 1) = P(\hat{Y} = 1 | A = 0, Y)$
✗ ✗ Equality of Odds $P(\hat{Y} = 1 | A = 0, Y) = P(\hat{Y} = 1 | A = 1, Y)$

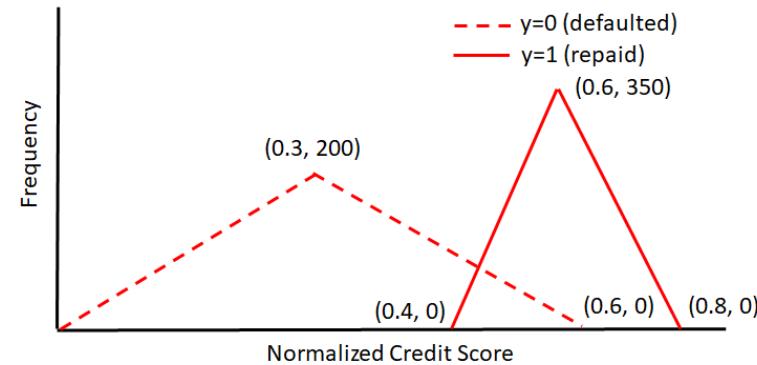
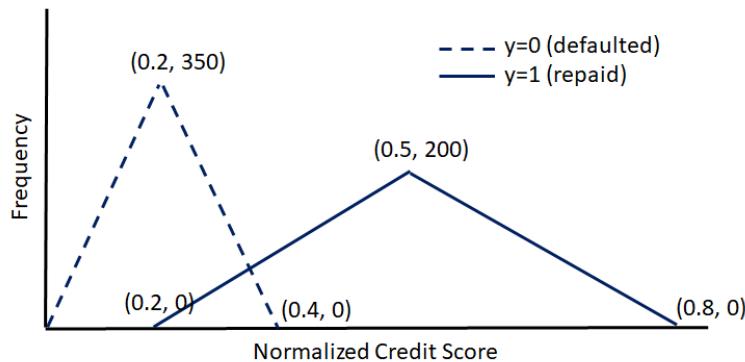
Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

ML EC-3M Problem

Consider a financial institution that uses normalized credit score for approving or rejecting housing loan. Note there are two population of applicants 'blue' (deprived group with protected attribute $A=0$) and 'red' (favored group with $A=1$). Loan is approved, i.e., $y'=1$ if normalized credit score is greater than some threshold t , else rejected. $y=1$ indicates approved loans that are repaid based on historical data, and $y=0$ indicates approved loan was defaulted. The financial institution wants to approve loans that is likely to be repaid. Distributions are described in following figure (not drawn to scale) for both populations.



A. Calculate probability $P(y'=1)$ for both 'blue' and 'red' populations for threshold $t=0.4$. Is fairness achieved w.r.t. protected attribute A?



For 'blue' population,

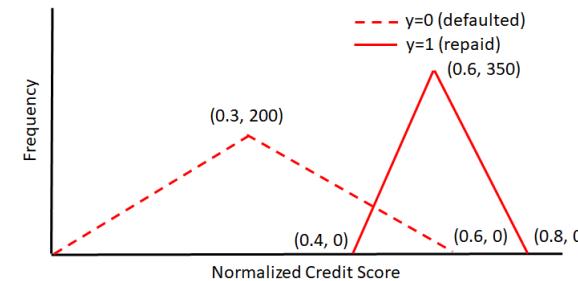
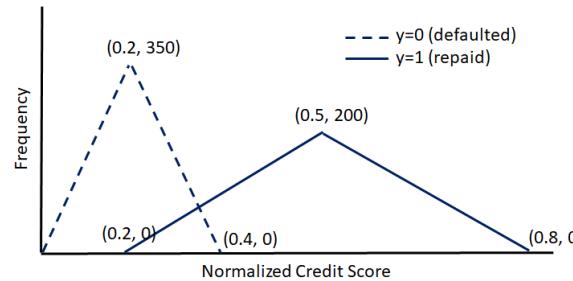
$$P(y'=1) = [0.5 * 200 * 0.6 - 0.5 * 0.2 * 133.3] / [0.2 * 350 + 0.3 * 200] = 0.359$$

For 'red' population,

$$P(y'=1) = [0.2 * 350 + 0.5 * 0.2 * 133.3] / [0.3 * 200 + 0.2 * 350] = 0.641 \rightarrow \text{Fairness not achieved.}$$

ML EC-3M Problem

B. If threshold $t=0.45$ is chosen for the blue population, calculate the probability $P(y'=1)$ for the blue population. What threshold value for the red population will ensure demographic parity?



For 'blue' population, for $t=0.45$

$$P(Y'=1) = [0.5 * 200 * 0.6 - 0.5 * 0.25 * 166.67] / [0.2 * 350 + 0.3 * 200] = 0.30$$

For 'red' population, for any $0.4 < t < 0.6$

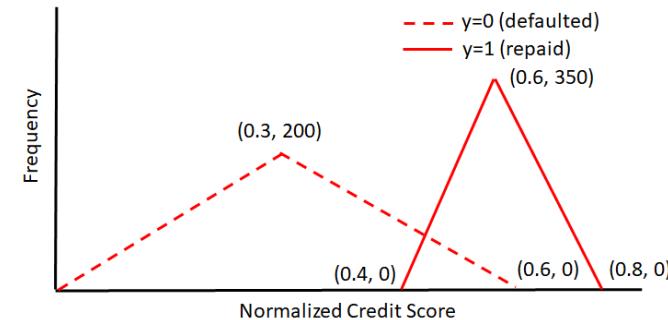
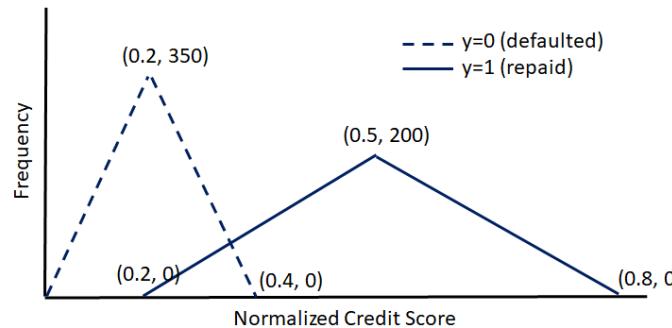
$$P(Y'=1) = [0.2 * 350 + 0.5 * (0.6-t) * 200 / 0.3 * (0.6-t) - 0.5 * (t-0.4) * 350 / 0.2] / [0.3 * 200 + 0.2 * 350]$$

Therefore, for demographic parity,

$$70 + 0.5 * (0.6-t)^2 * 333.33 - 0.5 * (t-0.4) * 350 / 0.2 = 39 \rightarrow t = 0.44463$$

ML EC-3M Problem

C. If threshold $t=0.5$ is chosen for the blue population, calculate the probability $P(y'=1|y=1)$. What threshold value for the red population will ensure equal opportunity?



For the blue population, from the frequency distribution

✓ for $t=0.5$ $P(y'=1|y=1)=0.5$

For the red population, from the distribution figure, for $t=0.6$ $P(y'=1|y=1)=0.5$

Summary of Fairness Criteria

Fairness Criteria	Criteria	Group	Individual
Unawareness	Excludes A in Predictions		✓
Demographic Parity	$P(\hat{Y} = 1 A = 0) = P(\hat{Y} = 1 A = 1)$	✓	
Equalized Odds	$P(\hat{Y} = 1 A = 0, Y) = P(\hat{Y} = 1 A = 1, Y)$	✓	
Equalized Opportunity	$P(\hat{Y} = 1 A = 0, Y = 1) = P(\hat{Y} = 1 A = 1, Y = 1)$	✓	

Required Reading

- Barocas: Ch 2
- Bishop: Ch 8.2
- <https://www.borealisai.com/research-blogs/tutorial1-bias-and-fairness-ai/>

Recommended

- Papers mentioned on the slides

Bishop, Christopher M. Pattern recognition and machine learning. Springer, 2006.

Barocas, S., Hardt, M., & Narayanan, A. Fairness and Machine Learning, 2018.

Additional Reading

- Gajane, Pratik, and Mykola Pechenizkiy. On formalizing fairness in prediction with machine learning. arXiv 2017
- Luong, Binh Thanh, Salvatore Ruggieri, and Franco Turini. k-NN as an implementation of situation testing for discrimination discovery and prevention. SIGKDD 2011
- Hardt, M., Price, E., & Srebro, N. Equality of opportunity in supervised learning. NeurIPS 2016
- Pedreshi, Dino, Salvatore Ruggieri, and Franco Turini. Discrimination-aware data mining. SIGKDD 2008
- Zafar, Muhammad Bilal, et al. Fairness Constraints: Mechanisms for Fair Classification. AIStats 2017



BITS Pilani
Pilani Campus

Fair, Accountable, Transparent Machine Learning (FAccT ML) ZG517

Dr. Sugata Ghosal
Sugata.ghosal@pilani.bits-pilani.ac.in

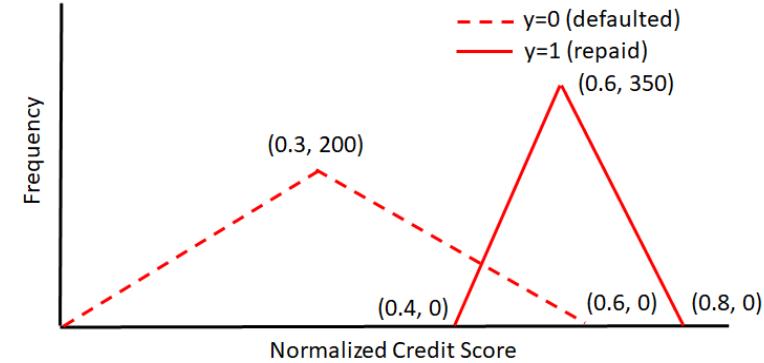
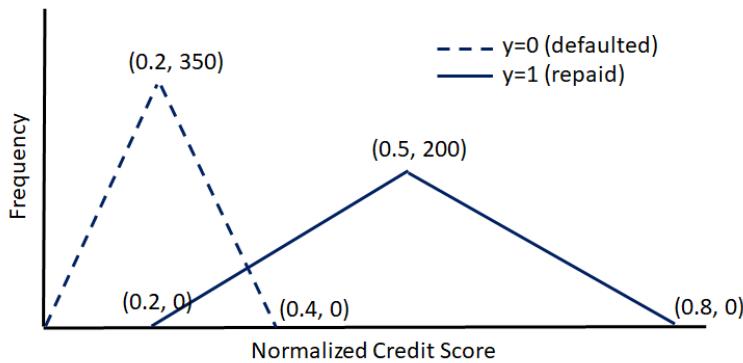


Session 3
Date – 4th June 2023
Time – 8:45 AM to 10:45 PM

These slides are prepared by the instructor, with grateful acknowledgement of many others who made their course materials freely available online.

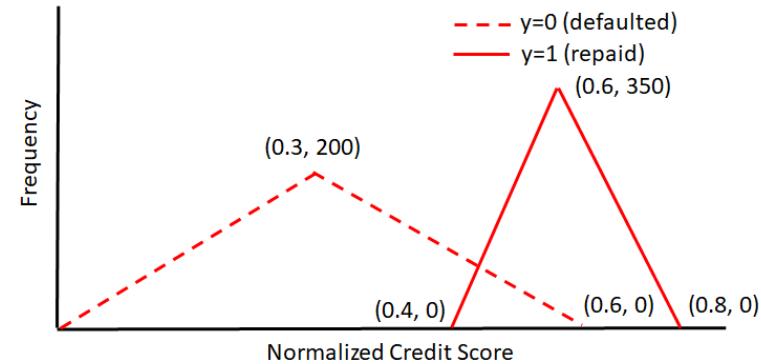
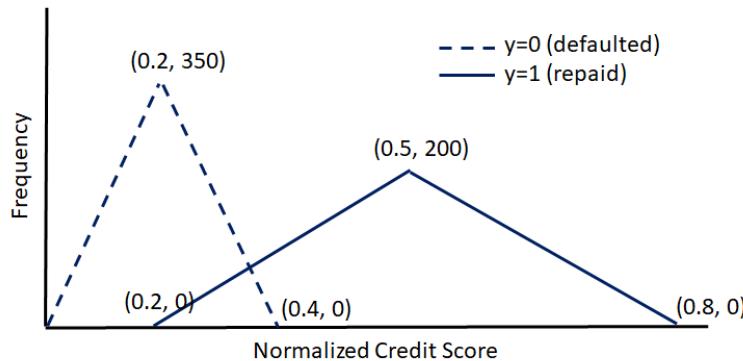
ML EC-3M Problem

Consider a financial institution that uses normalized credit score for approving or rejecting housing loan. Note there are two population of applicants 'blue' (deprived group with protected attribute $A=0$) and 'red' (favored group with $A=1$). Loan is approved, i.e., $y'=1$ if normalized credit score is greater than some threshold t , else rejected. $y=1$ indicates approved loans that are repaid based on historical data, and $y=0$ indicates approved loan was defaulted. The financial institution wants to approve loans that is likely to be repaid. Distributions are described in following figure (not drawn to scale) for both populations.



ML EC-3M Problem

A. Calculate probability $P(y'=1)$ for both 'blue' and 'red' populations for threshold $t=0.4$. Is fairness achieved w.r.t. protected attribute A?



For 'blue' population,

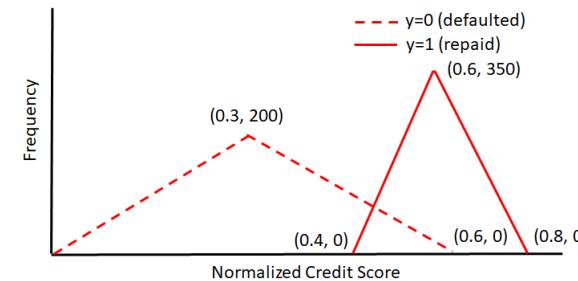
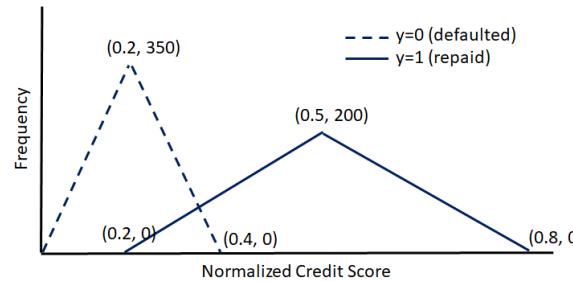
$$P(y'=1) = [0.5 * 200 * 0.6 - 0.5 * 0.2 * 133.3] / [0.2 * 350 + 0.3 * 200] = 0.359$$

For 'red' population,

$$P(y'=1) = [0.2 * 350 + 0.5 * 0.2 * 133.3] / [0.3 * 200 + 0.2 * 350] = 0.641 \rightarrow \text{Fairness not achieved.}$$

ML EC-3M Problem

B. If threshold $t=0.45$ is chosen for the blue population, calculate the probability $P(y'=1)$ for the blue population. What threshold value for the red population will ensure demographic parity?



For 'blue' population, for $t=0.45$

$$P(Y'=1) = [0.5 * 200 * 0.6 - 0.5 * 0.25 * 166.67] / [0.2 * 350 + 0.3 * 200] = 0.30$$

For 'red' population, for any $0.4 < t < 0.6$

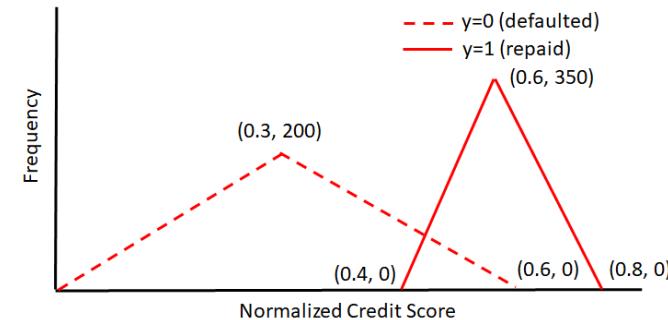
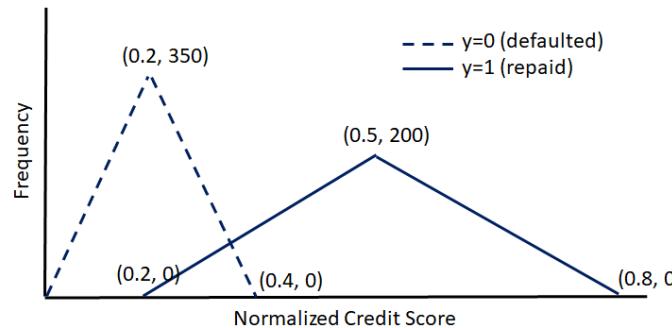
$$P(Y'=1) = [0.2 * 350 + 0.5 * (0.6-t) * 200 / 0.3 * (0.6-t) - 0.5 * (t-0.4) * 350 / 0.2] / [0.3 * 200 + 0.2 * 350]$$

Therefore, for demographic parity,

$$70 + 0.5 * (0.6-t)^2 * 333.33 - 0.5 * (t-0.4) * 350 / 0.2 = 39 \rightarrow t = 0.44463$$

ML EC-3M Problem

C. If threshold $t=0.5$ is chosen for the blue population, calculate the probability $P(y'=1|y=1)$. What threshold value for the red population will ensure equal opportunity?



For the blue population, from the frequency distribution

✓ for $t=0.5$ $P(y'=1|y=1)=0.5$

For the red population, from the distribution figure, for $t=0.6$ $P(y'=1|y=1)=0.5$

Demographic Parity Applied to a Group

Demographic Parity Is Applied to a Group of Samples

- Does not require features to be masked out

A Predictor \hat{Y} Satisfies Demographic Parity If

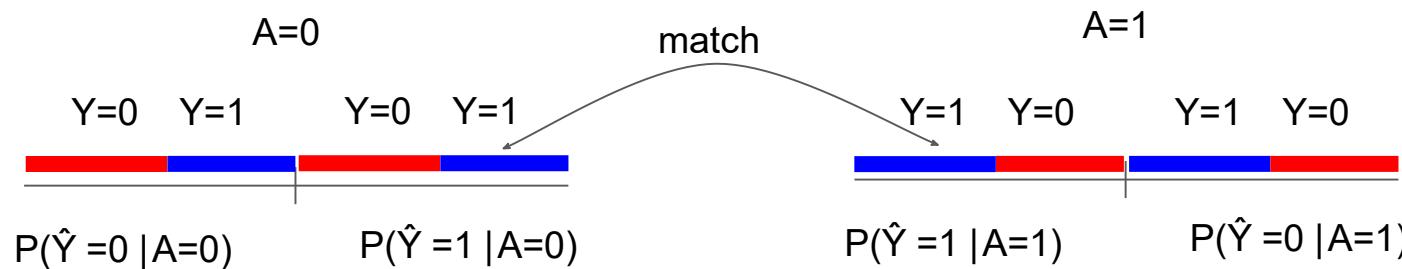
- The probabilities of positive predictions are the same regardless of whether the group is protected
- Protected groups are identified as $A = 1$

$$P(\hat{Y} = 1|A = 1) = P(\hat{Y} = 1|A = 0)$$

Equality of Opportunity

Equal Probabilities for Qualified People Across Protected Groups

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

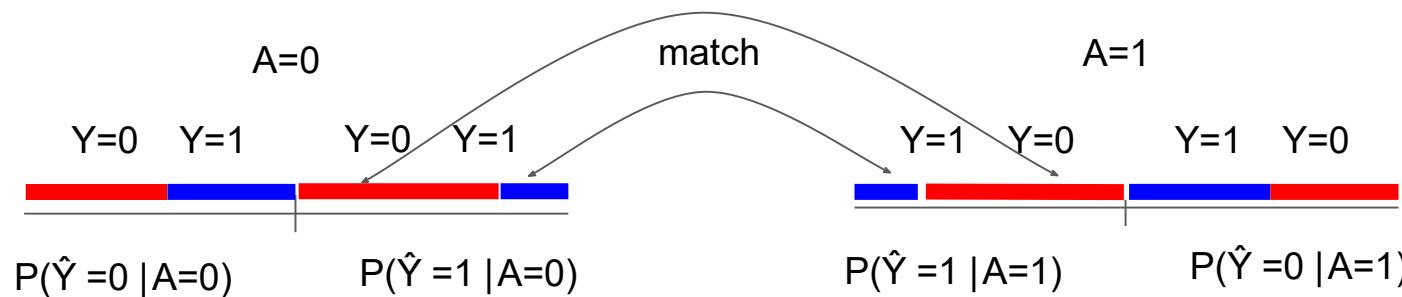


Hardt et al, 2016

Equality of Odds

Equal Probabilities for Both Qualified/Unqualified People Across Protected Groups

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



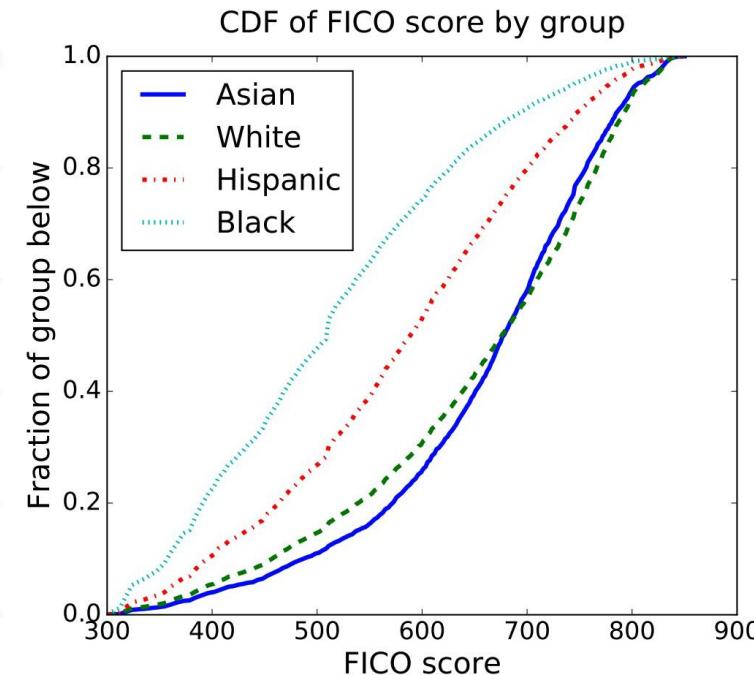
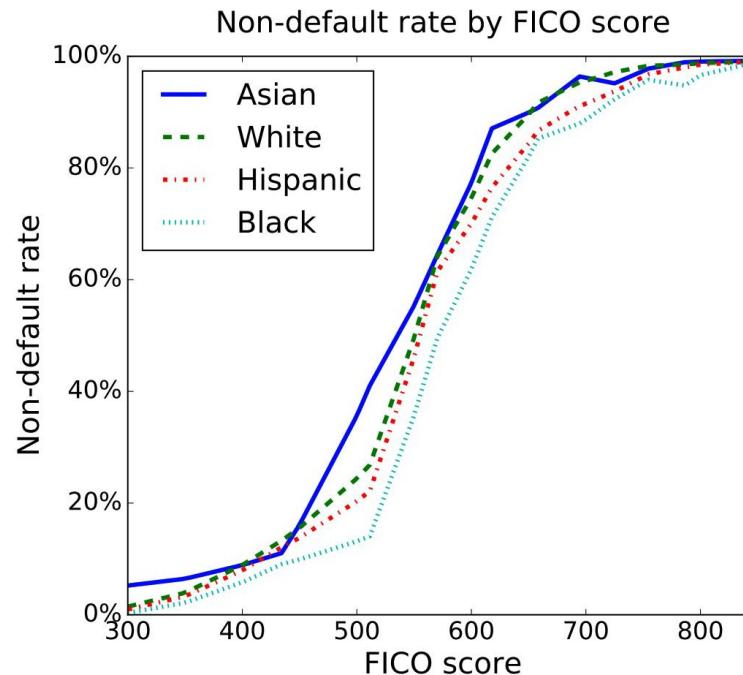
Hardt et al, 2016

Case Study on FICO

- FICO Dataset
 - 301,536 TransUnion TransRisk scores from 2003
 - Scores ranges from 300 to 850
 - People were labeled as in default if they failed to pay a debt for at least 90 days
 - Protected attribute A is race, with four values: {Asian, white non-Hispanic, Hispanic, and black}

FICO Scores

- 18% Default Rate on Any Accounts Corresponds to a 2% Default Rate for New Loans



Making Lending Decisions Without Discriminating

- Requirement: **Default Rate < 18%**, Simple Threshold Model
 - Max Profit - No Fairness Constraints
 - Race Blind - Using the same threshold for all race groups

Making Lending Decisions Without Discriminating

- Requirement: **Default Rate < 18%**, Simple Threshold Model
 - Max Profit - No Fairness Constraints
 - Race Blind - Using the same threshold for all race groups
 - Demographic Parity
 - Fraction of the group members that qualify for the loan are the same

$$P(\hat{Y} = 1 | A = 1) = P(\hat{Y} = 1 | A = 0)$$

Making Lending Decisions Without Discriminating



- Requirement: **Default Rate < 18%, Simple Threshold Model**

- Max Profit - No Fairness Constraints
 - Race Blind - Using the same threshold for all race groups
 - Demographic Parity
 - Fraction of the group members that qualify for the loan are the same

$$P(\hat{Y} = 1 | A = 1) = P(\hat{Y} = 1 | A = 0)$$

- Equal Opportunity
 - Fraction of non-defaulting group members that qualify for the loan is the same

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

Making Lending Decisions Without Discriminating

- Requirement: **Default Rate < 18%**, Simple Threshold Model

- Max Profit - No Fairness Constraints
 - Race Blind - Using the same threshold for all race groups
 - Demographic Parity
 - Fraction of the group members that qualify for the loan are the same

$$P(\hat{Y} = 1 | A = 1) = P(\hat{Y} = 1 | A = 0)$$

- Equal Opportunity
 - Fraction of non-defaulting group members that qualify for the loan is the same

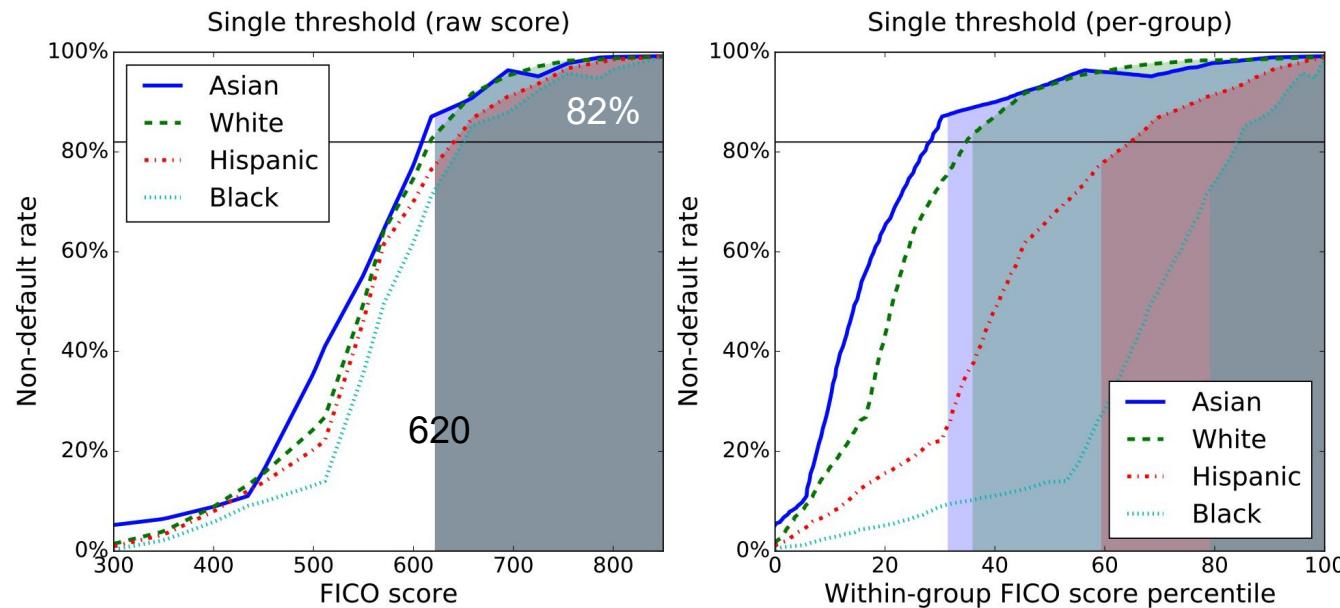
$$P(\hat{Y} = 1 | A = 0, Y = 1) = P(\hat{Y} = 1 | A = 1, Y = 1)$$

- Equal Odds
 - Fraction of both non-defaulting and defaulting groups of members that qualify for the loan is the same

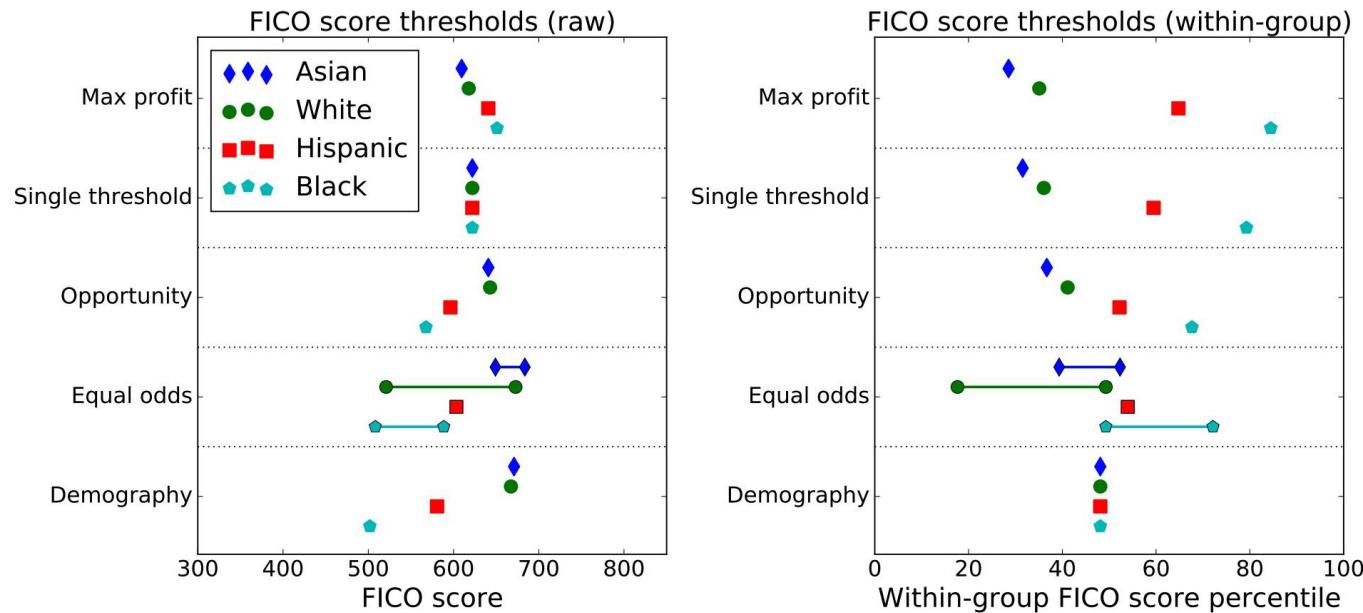
$$P(\hat{Y} = 1 | A = 0, Y) = P(\hat{Y} = 1 | A = 1, Y)$$

Credit Modeling Using A Single Threshold

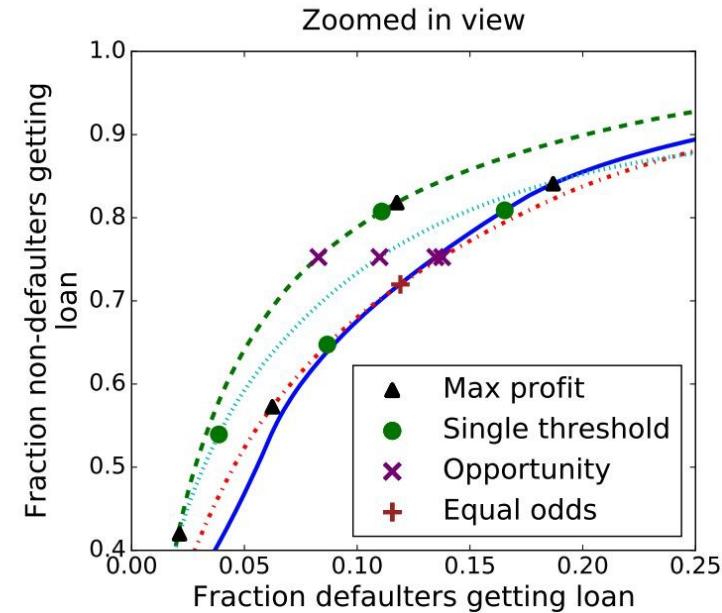
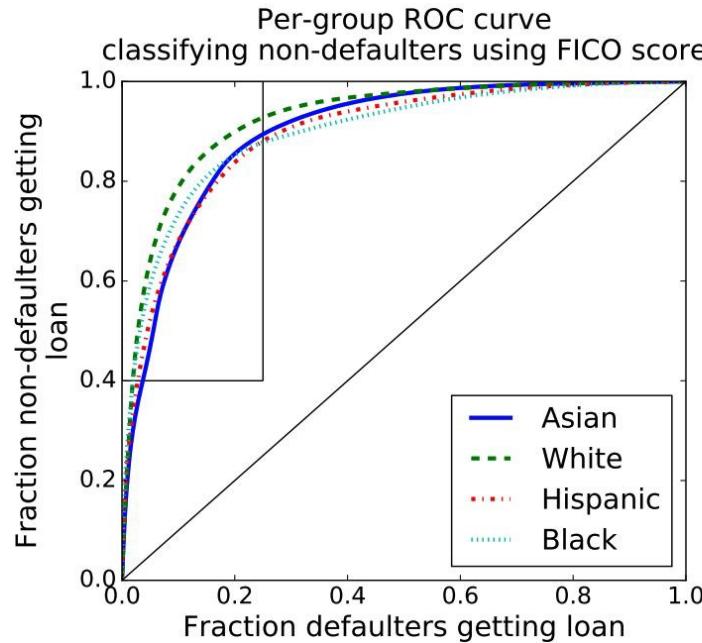
- Within-Group Percentile Differs Dramatically for Each Group



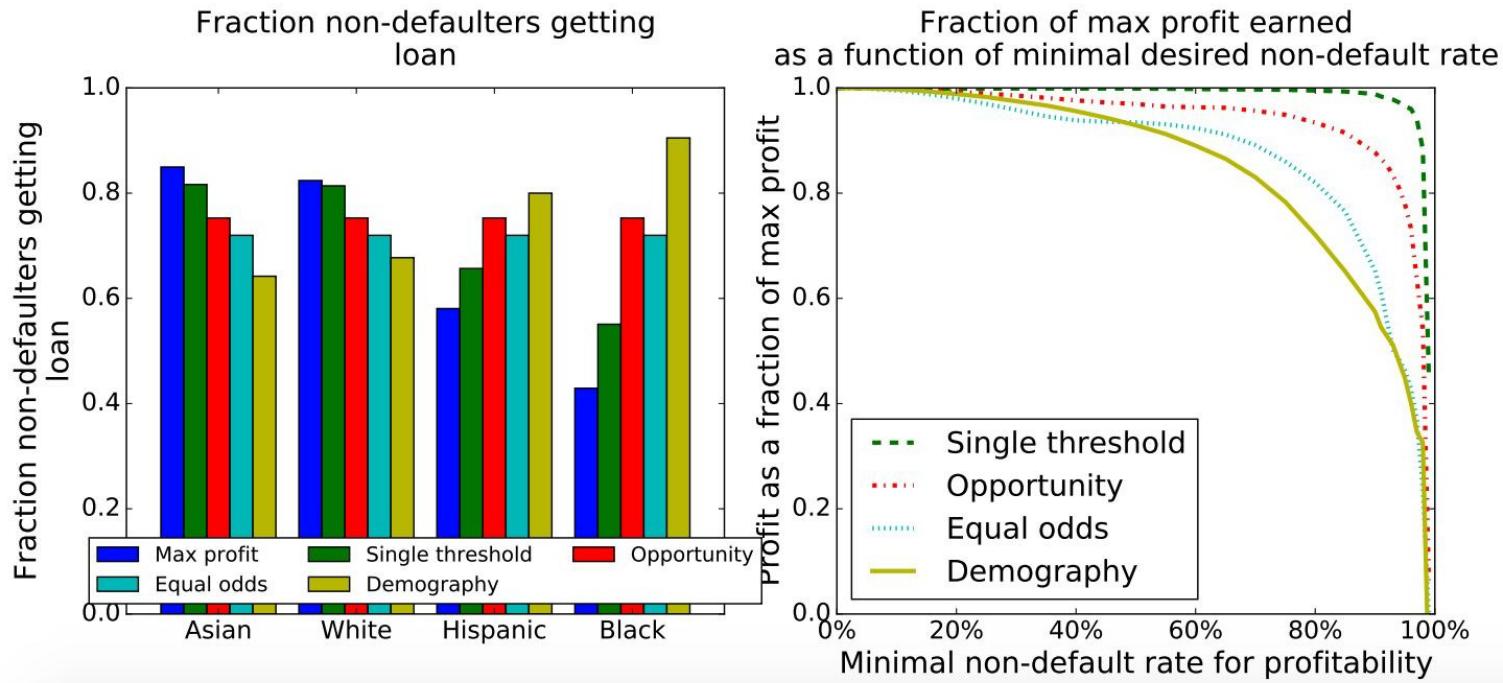
Found Thresholds for Each Fairness Definitions



Identifying Non-Defaulters



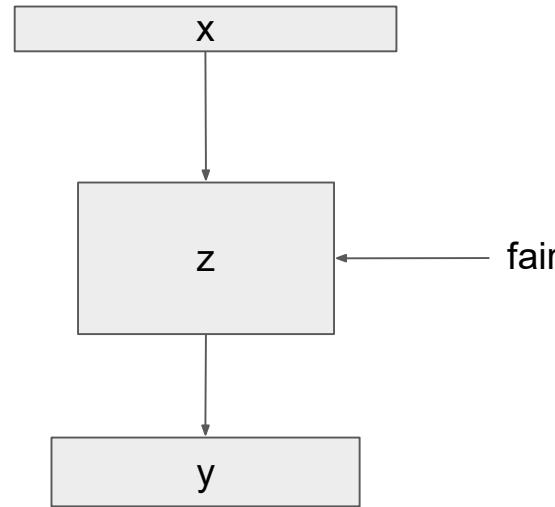
Non-Defaulters and Max Profits



Fair Representation Learning

Make representations fair

- Ensure fairness up to a certain level



Prejudice Remover Regularizer

Quantified causes of unfairness

Prejudice

- Unfairness rooted in the dataset

Underestimation

- Model unfairness because the model is not fully converged

Negative Legacy

- Unfairness due to sampling biases

Training Objective

$$-\mathcal{L}(\mathcal{D}; \boldsymbol{\Theta}) + \eta R(\mathcal{D}, \boldsymbol{\Theta}) + \frac{\lambda}{2} \|\boldsymbol{\Theta}\|_2^2$$

Loss of the Model Fairness Regularizer L2 Regularizer

[Kamishima et al, 2012](#)

Prejudice Index (PI)

Recall that Indirect Discrimination Happens When

Prediction is not directly conditioned on sensitive variables S

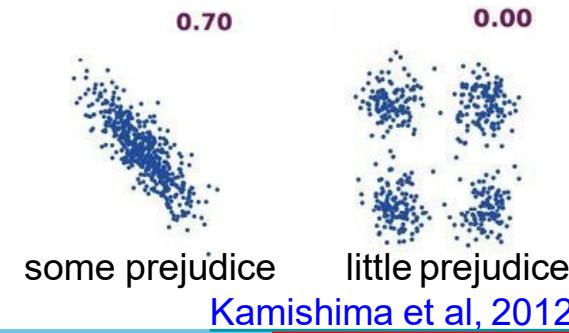
Prediction is indirectly conditioned on S by a variable O that is dependent on S

Prejudice Index (PI)

- Measures the degree of indirect discrimination based on mutual information

$$PI = \sum_{(y,s) \in \mathcal{D}} \hat{Pr}[y, s] \ln \frac{\hat{Pr}[y, s]}{\hat{Pr}[y]\hat{Pr}[s]}$$

↑
prediction model



Normalized Prejudice Index (NPI)

- Prejudice Index (PI)
 - Measures the degree of indirect discrimination based on mutual information
 - Ranges in $[0, +\infty)$

$$\text{PI} = \sum_{(y,s) \in \mathcal{D}} \hat{\Pr}[y, s] \ln \frac{\hat{\Pr}[y, s]}{\hat{\Pr}[y]\hat{\Pr}[s]}$$

- Normalized Prejudice Index (NPI)
 - Normalize PI by the entropy of Y and S
 - Ranges in $[0, 1]$

$$\text{NPI} = \text{PI}/(\sqrt{H(Y)H(S)})$$

[Kamishima et al, 2012](#)

Optimizing PI

- Learning PI

$$\text{PI} = \sum_{Y,S} \hat{\Pr}[Y, S] \ln \frac{\hat{\Pr}[Y, S]}{\hat{\Pr}[S]\hat{\Pr}[Y]}$$

[Kamishima et al, 2012](#)

Optimizing PI

- Learning PI

$$\text{PI} = \sum_{Y,S} \hat{\Pr}[Y, S] \ln \frac{\hat{\Pr}[Y, S]}{\hat{\Pr}[S]\hat{\Pr}[Y]} = \sum_{X,S} \tilde{\Pr}[X, S] \sum_Y \mathcal{M}[Y|X, S; \boldsymbol{\Theta}] \ln \frac{\hat{\Pr}[Y, S]}{\hat{\Pr}[S]\hat{\Pr}[Y]}.$$

↓
 Expands $\Pr(Y, S)$ into $\sum_X \Pr(X, Y, S)$
 ↓
 triple summations double summations Prediction Model

- Using Logistic Regression Model as the Prediction Model

$$\mathcal{M}[y|\mathbf{x}, s; \boldsymbol{\Theta}] = y\sigma(\mathbf{x}^\top \mathbf{w}_s) + (1 - y)(1 - \sigma(\mathbf{x}^\top \mathbf{w}_s))$$

[Kamishima et al, 2012](#)

Optimizing PI

- Learning PI

$$\begin{aligned}
 \text{PI} &= \sum_{Y,S} \hat{\Pr}[Y, S] \ln \frac{\hat{\Pr}[Y, S]}{\hat{\Pr}[S]\hat{\Pr}[Y]} = \sum_{X,S} \tilde{\Pr}[X, S] \sum_Y \mathcal{M}[Y|X, S; \boldsymbol{\Theta}] \ln \frac{\hat{\Pr}[Y, S]}{\hat{\Pr}[S]\hat{\Pr}[Y]}. \\
 &= \sum_{(\mathbf{x}_i, s_i) \in \mathcal{D}} \sum_{y \in \{0,1\}} \mathcal{M}[y|\mathbf{x}_i, s_i; \boldsymbol{\Theta}] \ln \boxed{\frac{\hat{\Pr}[y|s_i]}{\hat{\Pr}[y]}}.
 \end{aligned}$$

- Using Logistic Regression Model as the Prediction Model difficult to evaluate

$$\mathcal{M}[y|\mathbf{x}, s; \boldsymbol{\Theta}] = y\sigma(\mathbf{x}^\top \mathbf{w}_s) + (1 - y)(1 - \sigma(\mathbf{x}^\top \mathbf{w}_s))$$

[Kamishima et al, 2012](#)

Optimizing PI

$$\text{PI} = \sum_{(\mathbf{x}_i, s_i) \in \mathcal{D}} \sum_{y \in \{0,1\}} \mathcal{M}[y | \mathbf{x}_i, s_i; \boldsymbol{\Theta}] \ln \frac{\hat{\Pr}[y | s_i]}{\hat{\Pr}[y]}$$

[Kamishima et al, 2012](#)

Optimizing PI

$$\text{PI} = \sum_{(\mathbf{x}_i, s_i) \in \mathcal{D}} \sum_{y \in \{0,1\}} \mathcal{M}[y | \mathbf{x}_i, s_i; \boldsymbol{\Theta}] \ln \frac{\hat{\Pr}[y | s_i]}{\hat{\Pr}[y]}$$

$$\hat{\Pr}[y | s] = \int_{\text{dom}(X)} \Pr^*[X | s] \mathcal{M}[y | X, s; \boldsymbol{\Theta}] dX$$

Integrals Are Difficult to Evaluate

[Kamishima et al, 2012](#)

Optimizing PI

$$\text{PI} = \sum_{(\mathbf{x}_i, s_i) \in \mathcal{D}} \sum_{y \in \{0,1\}} \mathcal{M}[y | \mathbf{x}_i, s_i; \boldsymbol{\Theta}] \ln \frac{\hat{\Pr}[y | s_i]}{\hat{\Pr}[y]}$$

$$\begin{aligned} \hat{\Pr}[y | s] &= \int_{\text{dom}(X)} \Pr^*[X | s] \mathcal{M}[y | X, s; \boldsymbol{\Theta}] dX \\ &\approx \frac{\sum_{(\mathbf{x}_i, s_i) \in \mathcal{D} \text{ s.t. } s_i = s} \mathcal{M}[y | \mathbf{x}_i, s; \boldsymbol{\Theta}]}{|\{(\mathbf{x}_i, s_i) \in \mathcal{D} \text{ s.t. } s_i = s\}|} \end{aligned}$$

Approximating integrals by sample means

[Kamishima et al, 2012](#)

Optimizing PI

$$\text{PI} = \sum_{(\mathbf{x}_i, s_i) \in \mathcal{D}} \sum_{y \in \{0,1\}} \mathcal{M}[y | \mathbf{x}_i, s_i; \boldsymbol{\Theta}] \ln \frac{\hat{\Pr}[y | s_i]}{\hat{\Pr}[y]}$$

$$\begin{aligned} \hat{\Pr}[y | s] &= \int_{\text{dom}(X)} \Pr^*[X | s] \mathcal{M}[y | X, s; \boldsymbol{\Theta}] dX \\ &\approx \frac{\sum_{(\mathbf{x}_i, s_i) \in \mathcal{D} \text{ s.t. } s_i=s} \mathcal{M}[y | \mathbf{x}_i, s; \boldsymbol{\Theta}]}{|\{(\mathbf{x}_i, s_i) \in \mathcal{D} \text{ s.t. } s_i=s\}|} \end{aligned}$$

Approximating integrals by sample means

$$\hat{\Pr}[y] \approx \frac{\sum_{(\mathbf{x}_i, s_i) \in \mathcal{D}} \mathcal{M}[y | \mathbf{x}_i, s_i; \boldsymbol{\Theta}]}{|\mathcal{D}|}$$

[Kamishima et al, 2012](#)

Optimizing PI

$$\text{PI} = \sum_{(\mathbf{x}_i, s_i) \in \mathcal{D}} \sum_{y \in \{0,1\}} \mathcal{M}[y | \mathbf{x}_i, s_i; \boldsymbol{\Theta}] \ln \frac{\hat{\Pr}[y | s_i]}{\hat{\Pr}[y]}$$

$$\hat{\Pr}[y | s] \approx \frac{\sum_{(\mathbf{x}_i, s_i) \in \mathcal{D} \text{ s.t. } s_i = s} \mathcal{M}[y | \mathbf{x}_i, s; \boldsymbol{\Theta}]}{|\{(\mathbf{x}_i, s_i) \in \mathcal{D} \text{ s.t. } s_i = s\}|} \quad \hat{\Pr}[y] \approx \frac{\sum_{(\mathbf{x}_i, s_i) \in \mathcal{D}} \mathcal{M}[y | \mathbf{x}_i, s_i; \boldsymbol{\Theta}]}{|\mathcal{D}|}$$

[Kamishima et al, 2012](#)

Putting Things Together

Optimization Target

$$-\mathcal{L}(\mathcal{D}; \boldsymbol{\Theta}) + \eta R(\mathcal{D}, \boldsymbol{\Theta}) + \frac{\lambda}{2} \|\boldsymbol{\Theta}\|_2^2$$

Loss of the Model

Fairness Regularizer

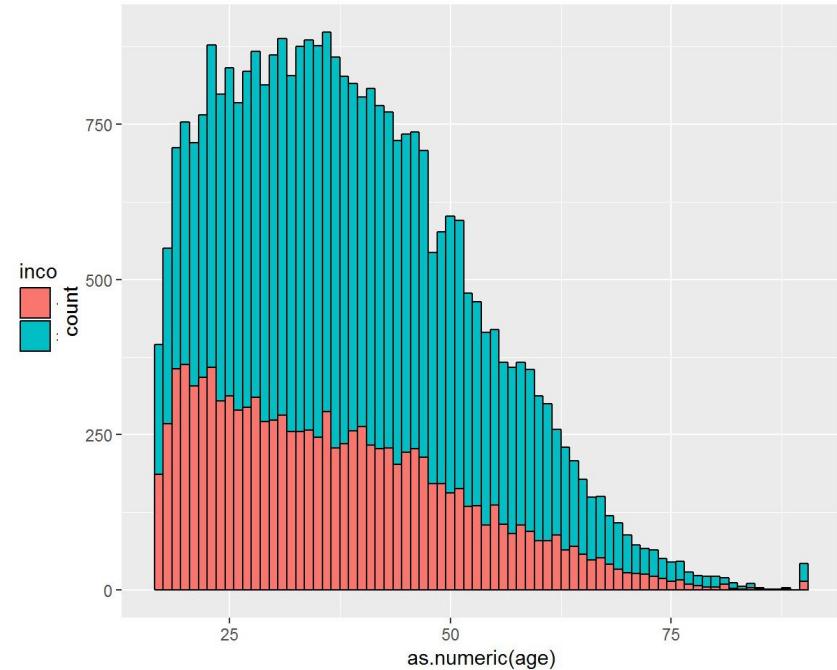
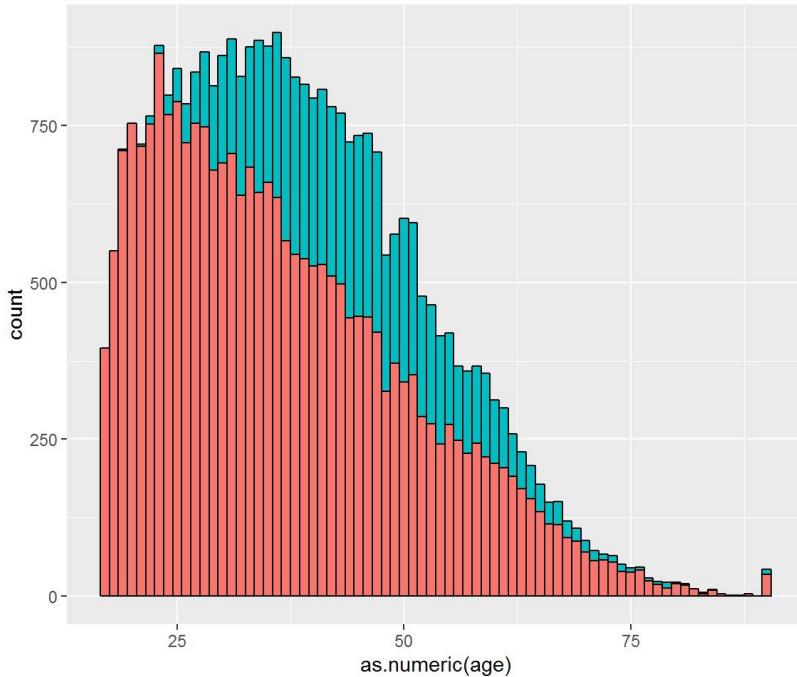
L2 Regularizer

- Fairness Regularizer

$$\text{PI} = \sum_{(\mathbf{x}_i, s_i) \in \mathcal{D}} \sum_{y \in \{0,1\}} \mathcal{M}[y | \mathbf{x}_i, s_i; \boldsymbol{\Theta}] \ln \frac{\hat{\Pr}[y | s_i]}{\hat{\Pr}[y]}$$

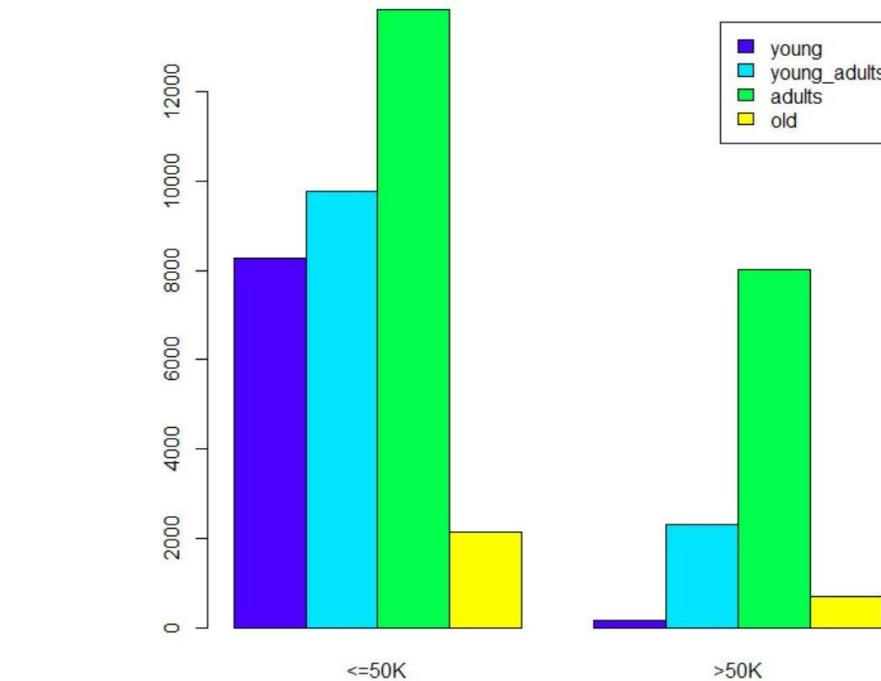
[Kamishima et al, 2012](#)

Adult Income Dataset ([Kohavi 1996](#))

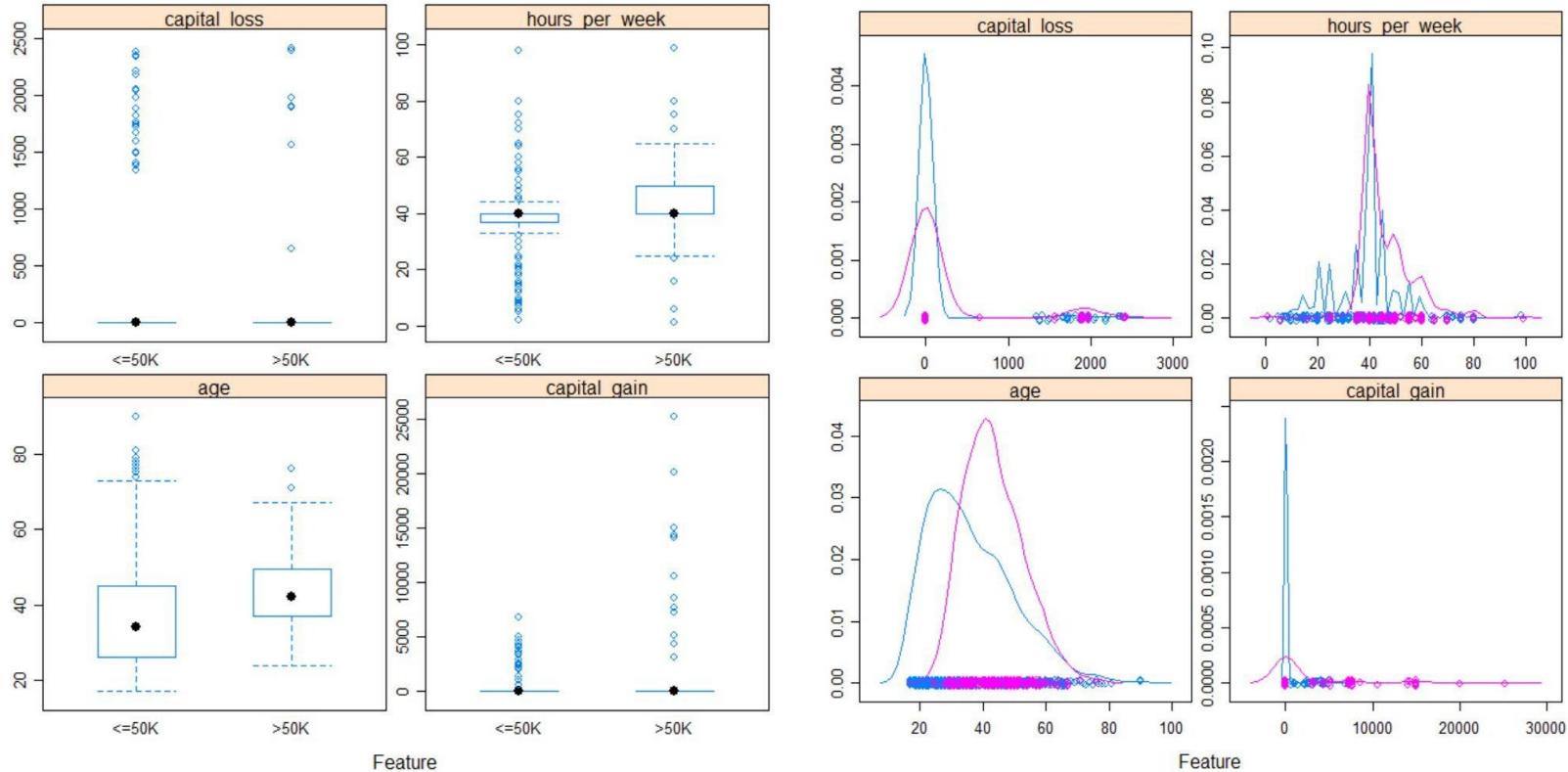


Adult Income Dataset ([Kohavi 1996](#))

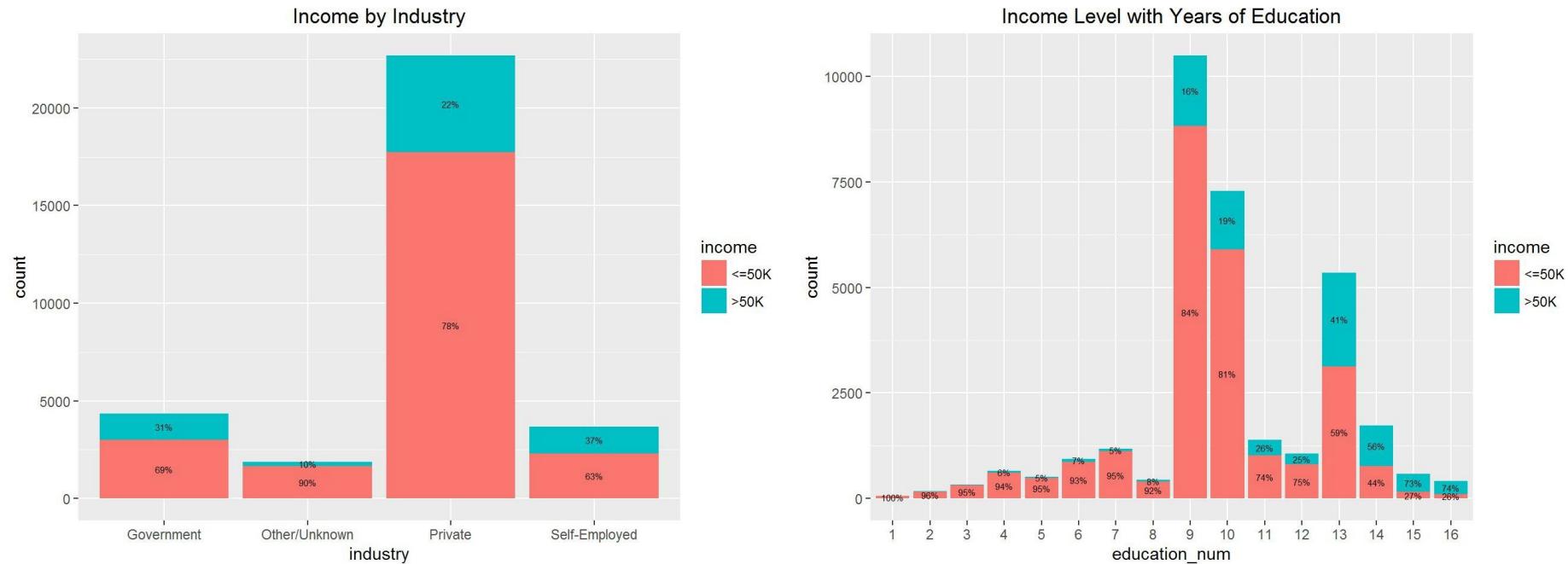
- Predict Whether Income Exceeds \$50K/yr Based on Census Data



Adult Income Dataset ([Kohavi 1996](#))

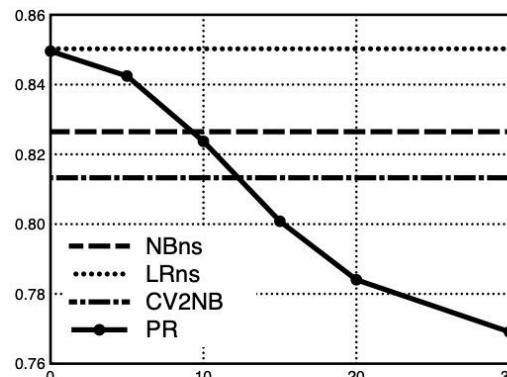


Adult Income Dataset ([Kohavi 1996](#))

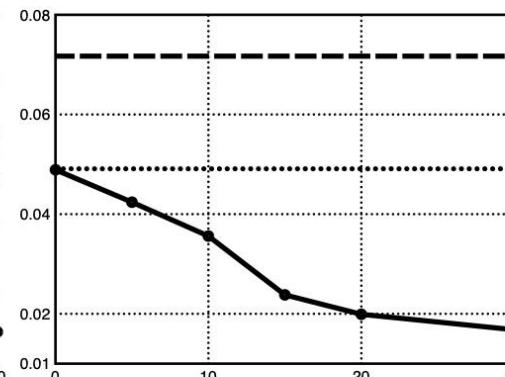


Results

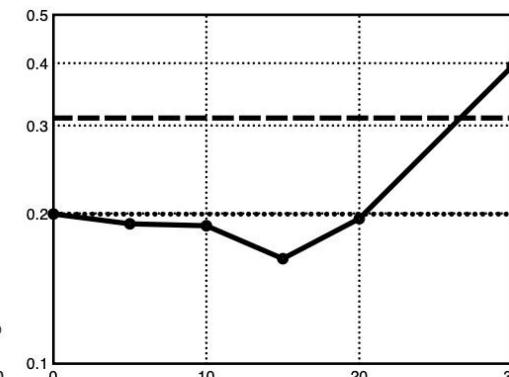
- Changes of Performance With η
 - Model performance decreases (Acc)
 - Discrimination Decreases (NPI)
 - "Fairness Efficiency" (PI/MI) Increases



(a) Acc



(b) NPI



(c) PI/MI

[Kamishima et al, 2012](#)

Results

- Prejudice Prior Sacrifices Model Performance
 - PR has lower Acc (Accuracy)
 - PR has lower NMI (normalized mutual information between labels and predictions)
- Prejudice Prior Makes Model Fair
 - PR has lower NPI

	method	Acc	NMI	NPI	PI/MI
Logistic Regression full fet.	LR	0.851	0.267	5.21E-02	2.10E-01
Logistic Regression no sensitive fet.	LRns	0.850	0.266	4.91E-02	1.99E-01
Logistic Regression + Prejudice Regularizer	PR $\eta=5$	0.842	0.240	4.24E-02	1.91E-01
	PR $\eta=15$	0.801	0.158	2.38E-02	1.62E-01
	PR $\eta=30$	0.769	0.046	1.68E-02	3.94E-01

η is the weight we put on prejudice regularizers [Kamishima et al, 2012](#)

Results

- PI/MI
 - Prejudice Index / Mutual Information
 - Demonstrates a trade-offs between model fairness and performance
 - Measures the amount of discrimination we eliminate with one unit of performance gain (measured by MI)

	method	Acc	NMI	NPI	PI/MI
Logistic Regression full fet.	LR	0.851	0.267	5.21E-02	2.10E-01
Logistic Regression no sensitive fet.	LRns	0.850	0.266	4.91E-02	1.99E-01
Logistic Regression + Prejudice Regularizer	PR $\eta=5$	0.842	0.240	4.24E-02	1.91E-01
	PR $\eta=15$	0.801	0.158	2.38E-02	1.62E-01
	PR $\eta=30$	0.769	0.046	1.68E-02	3.94E-01

η - weight put on the prejudice regularizer

[Kamishima et al, 2012](#)

Readings

- T1, Chapter 3
- <https://www.borealisai.com/research-blogs/tutorial1-bias-and-fairness-ai/>
- Moritz Hardt, “Equality of Opportunity in Supervised Learning”, 2016
- Kamishima et al., “Fairness-Aware Classifier with Prejudice Remover Regularizer”, 2012

Welcome!!



BITS Pilani
Pilani Campus

Fair, Accountable, Transparent Machine Learning (FAccT ML) ZG517

Dr. Sugata Ghosal
sugata.ghosal@pilani.bits-pilani.ac.in



Session 4
Date – 11th June 2023
Time – 8:45 AM to 10:45 PM

These slides are prepared by the instructor, with grateful acknowledgement of many others who made their course materials freely available online.

Readings

- <https://www.borealisai.com/research-blogs/tutorial1-bias-and-fairness-ai/>
- David Madras *et al.*, “Learning Adversarially Fair and Transferable Representations”, 2018
- Faisal Kamiran *et al.*, “Data preprocessing techniques for classification without discrimination”, 2012
- Flavio Calmon *et al.*, Optimized Preprocessing for Discrimination Prevention, 2017

Recap

- Fairness in Machine Learning
 - Preventing algorithms from being biased toward a protected group when allocating favorable outcomes

Attribute	FHA	ECOA
Race	✓	✓
Color	✓	✓
National origin	✓	✓
Religion	✓	✓
Sex	✓	✓
Familial status	✓	
Disability	✓	
Exercised rights under CCPA		✓
Marital status		✓
Recipient of public assistance		✓
Age		✓

Fair Housing Acts (FHA)

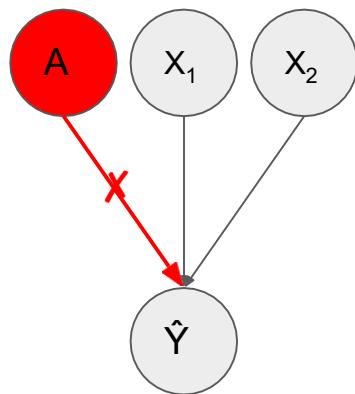
Equal Credit Opportunity ACts (ECOA)

Recap

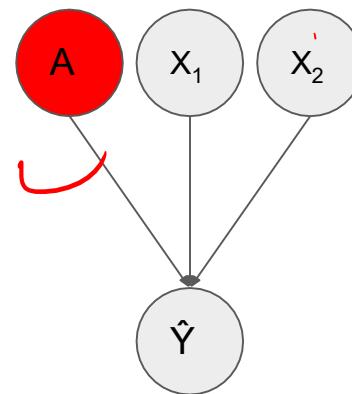


[Mehrabi et al, 2019](#)

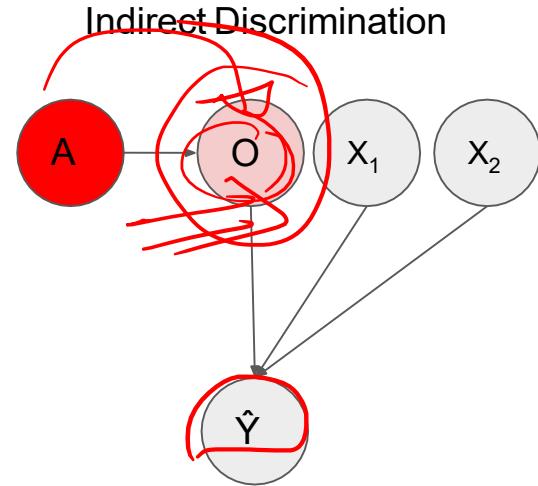
Recap



Direct Discrimination



Indirect Discrimination

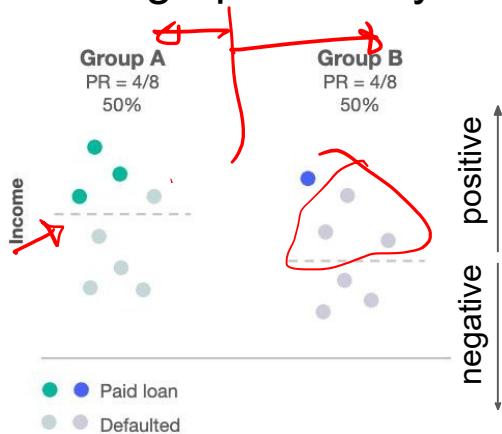


Fair ML Model

Fairness Through Unawareness (FTU)

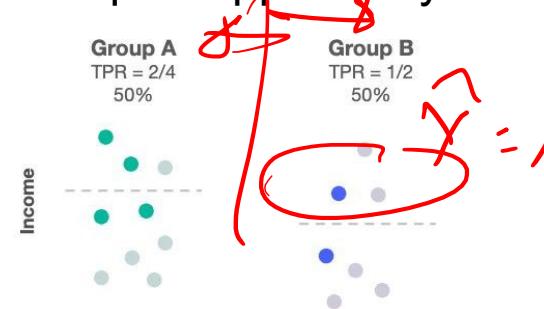
Recap

Demographic Parity



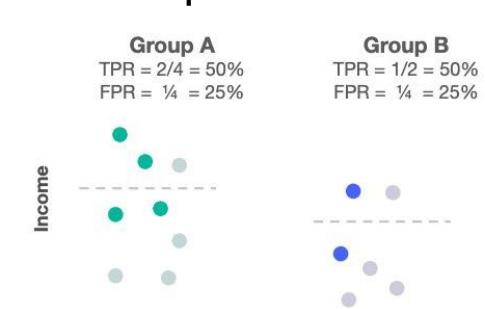
$$P(\hat{Y} = 1|A = 1) = P(\hat{Y} = 1|A = 0)$$

Equal Opportunity



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

Equal Odds



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

$$\begin{aligned} P(\hat{Y} = 1|A = 0, Y = 1) &= P(\hat{Y} = 1|A = 1, Y = 1) \\ P(\hat{Y} = 1|A = 0, Y = 1) &= P(\hat{Y} = 1|A = 1, Y = 0) \end{aligned}$$

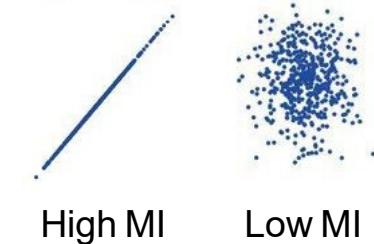
Recap

- Fair Representation Learning
 - Prejudice Removing Regularizer

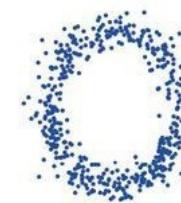
$$-\mathcal{L}(\mathcal{D}; \Theta) + \eta R(\mathcal{D}, \Theta) + \frac{\lambda}{2} \|\Theta\|_2^2$$

Loss of the Model Fairness Regularizer L2 Regularizer

Mutual Information



Low MI

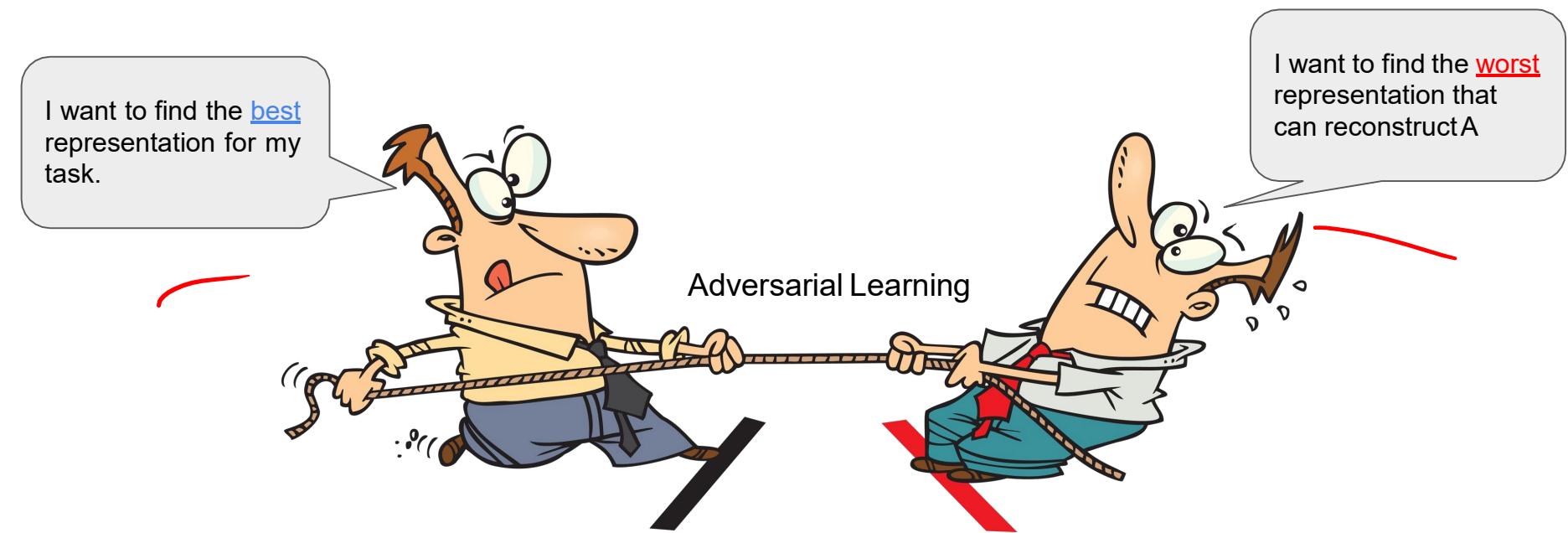


Fair ML Methods

- In-processing Methods
 - Constrain ML models while they learn, e.g.,
 - Prejudice Removing Regularizer,
 - Adversarial Learning
 - Pre-processing Methods
 - Transform data before ML models learn
 - e.g., Reweighting, Resampling
 - Post-processing Methods
 - Make predictions from a black-box ML model fair in the post-processing stage
 - e.g., Learning to Defer
- generative AI*

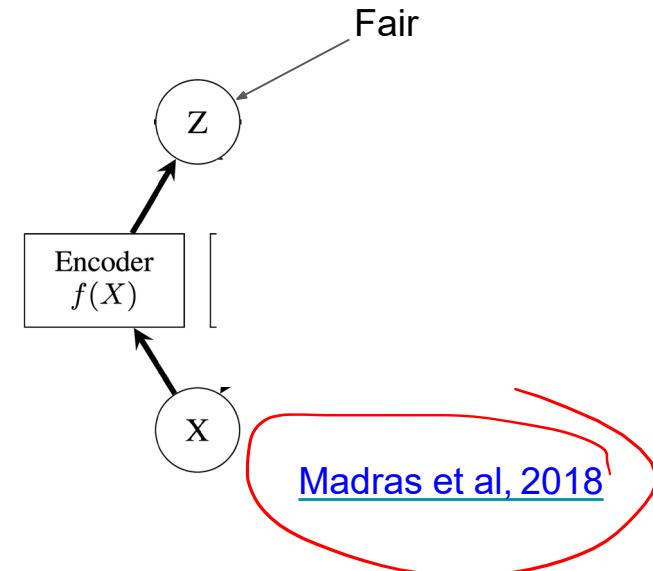
Fair Representation Learning

- How Do We Test the Fairness of Representation Z?
 - Adversarial Learning



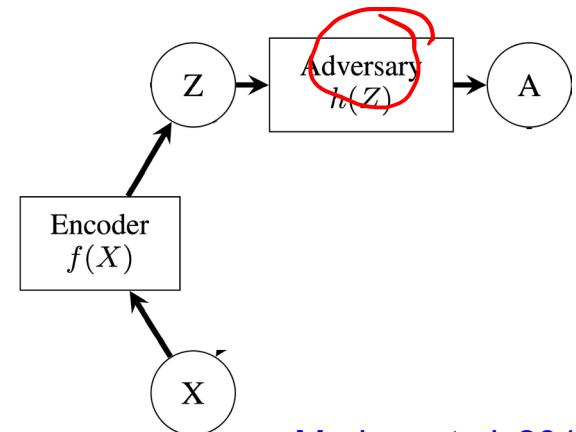
Deep Networks to Learn Fair Representations

- How Do We Make a Representation Z Fair?



Fair Representations

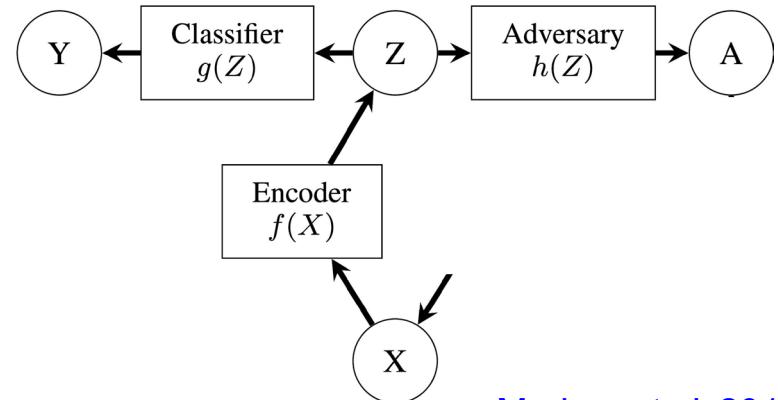
- How Do We Make a Representation Z Fair?
 - $Z = f(X)$
 - Test and see if a good amount of A can be reconstructed from Z
 - Compare A with $h(Z)$



[Madras et al, 2018](#)

Fair Representations

- How Do We Make a Representation Z Fair?
 - $Z = f(X)$
 - Test and see if a good amount of A can be reconstructed from Z
 - Compare A with $h(z)$
- Properties of ~~Deep~~ Representations
 - Achieve good performance for downstream task that generates $y=g(z)$



[Madras et al, 2018](#)

Fair Representations

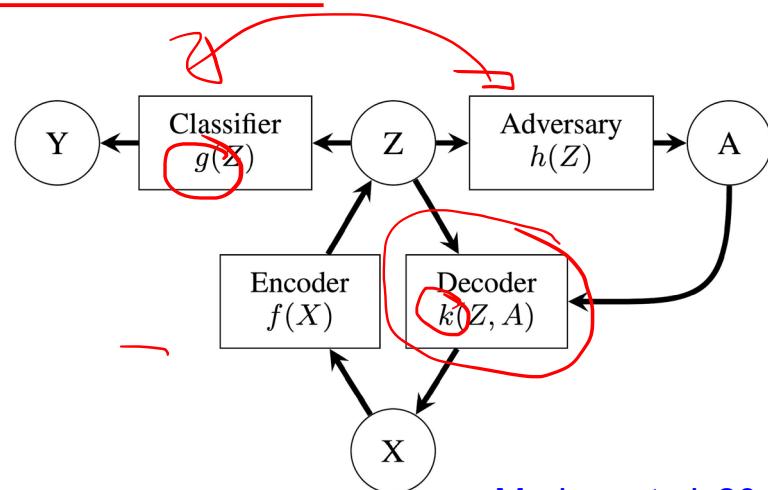
(x, y) = Training Set
A

- How Do We Make a Representation Z Fair?

- $Z = f(X)$
- Test and see if a good amount of A can be reconstructed from Z
- Compare A with $h(z)$

- Properties of Deep Representations

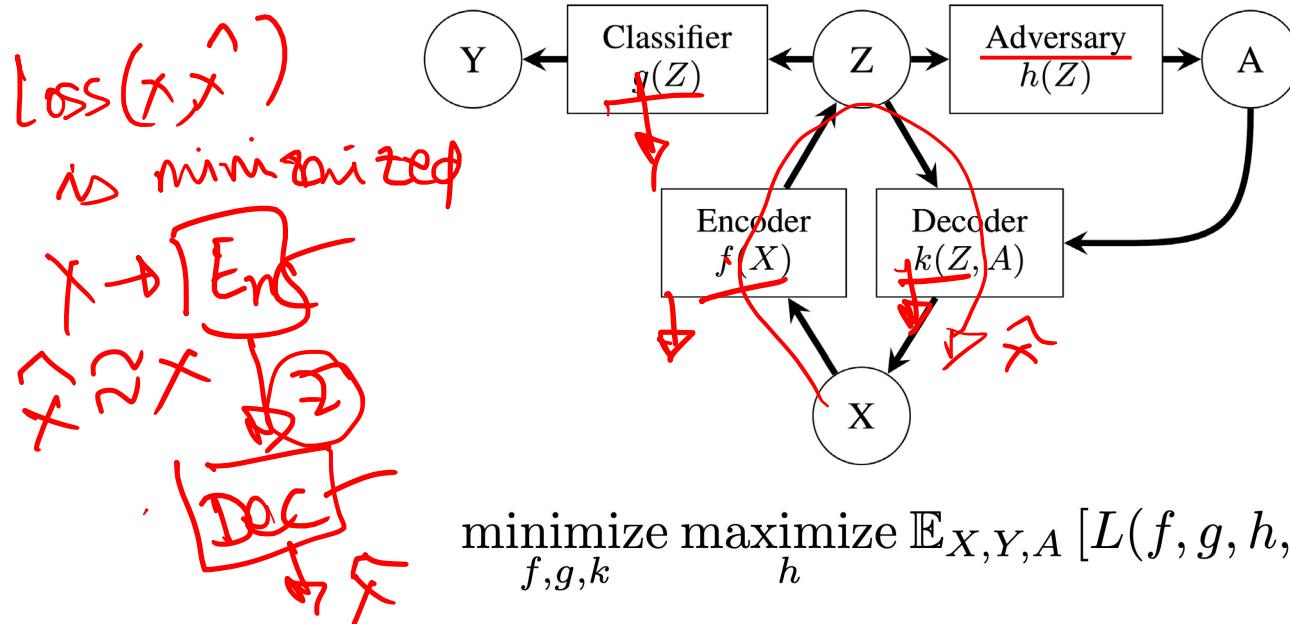
- Achieve good performance for downstream task that generates $y=g(z)$
- Has the Ability to Reconstruct $X = k(Z, A)$



Madras et al, 2018

Fairness Through Adversarial Learning

- Adversarial Learning
 - Models are trained using objectives that compete with each other

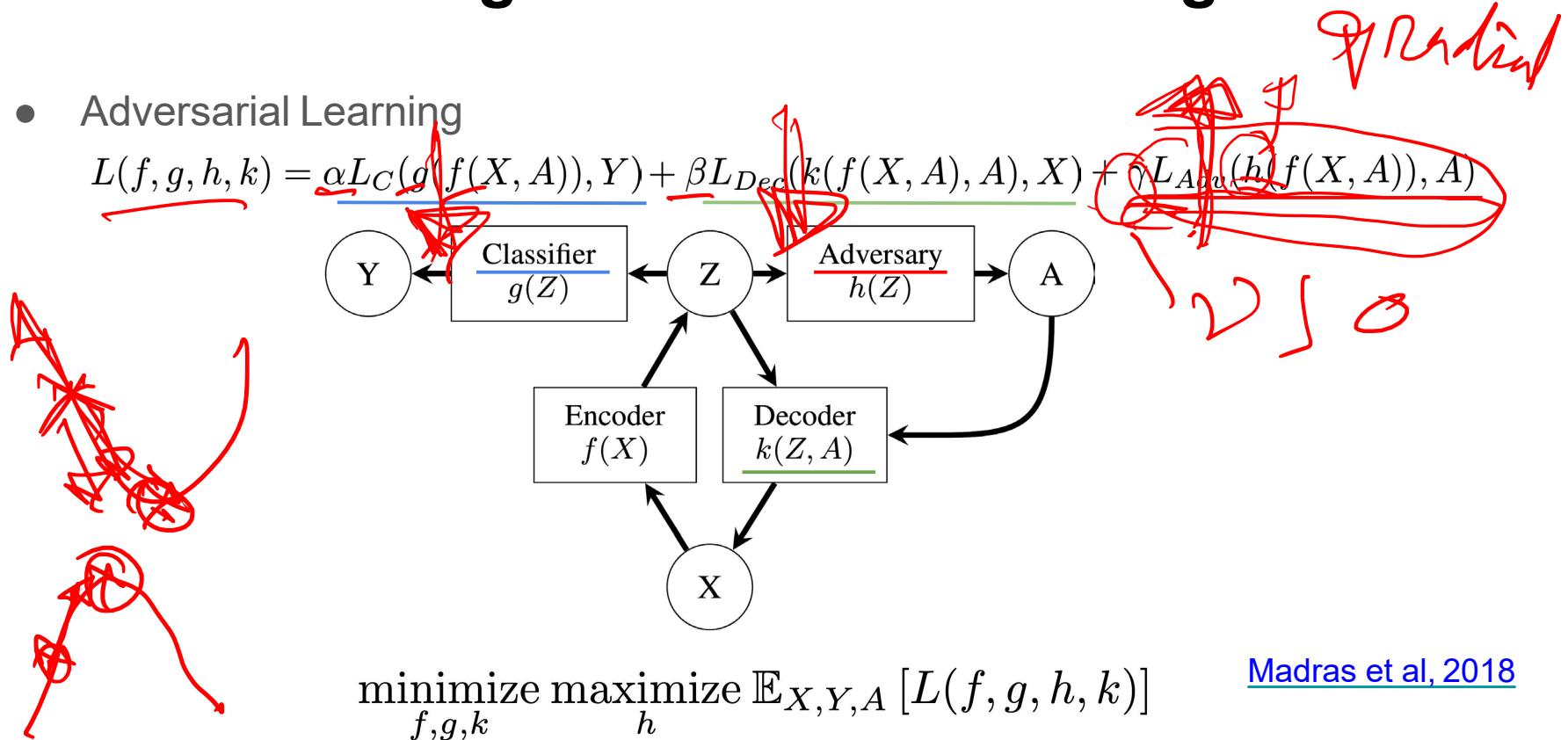


[Madras et al, 2018](#)

Fairness Through Adversarial Learning

- Adversarial Learning

$$L(f, g, h, k) = \alpha L_C(g(f(X, A)), Y) + \beta L_{Dec}(k(f(X, A), A), X) + \gamma L_{Adv}(h(f(X, A)), A)$$



$$P(\hat{Y} = 1 | A = 0) = P(\hat{Y} = 1 | A = 1)$$

Loss for Learning Fair Representations

- Adversarial Loss for Demographic Parity with Group $\mathcal{D}_0, \mathcal{D}_1$

$$L_{Adv}^{DP}(h) = 1 - \sum_{i \in \{0,1\}} \frac{1}{|\mathcal{D}_i|} \sum_{(x,a) \in \mathcal{D}_i} |h(f(x,a)) - a|$$

Demographic Parity: $P(\hat{Y} = 1 | A = 1) = P(\hat{Y} = 1 | A = 0)$

D_0
 D_1
 A of records in i

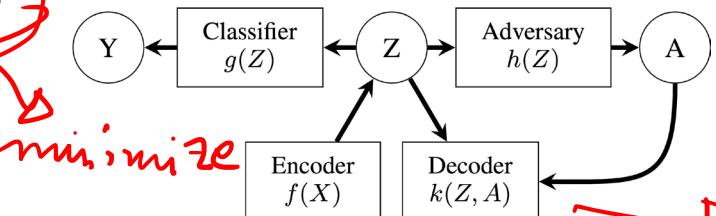
- Adversarial Loss for Equality of Odds with

Group $\mathcal{D}_i^j = \{(x, y, a) \in \mathcal{D} | a = i, y = j\}$

$$L_{Adv}^{EO}(h) = 2 - \sum_{(i,j) \in \{0,1\}^2} \frac{1}{|\mathcal{D}_i^j|} \sum_{(x,a) \in \mathcal{D}_i^j} |h(f(x,a)) - a|$$

Equality of Odds: $P(\hat{Y} = 1 | A = 0, Y) = P(\hat{Y} = 1 | A = 1, Y)$

maximize $\mathbb{D} = \{\mathcal{D}_0, \mathcal{D}_1\}$



maximize $(1 - \Sigma)$
 minimize $T(1 - \Sigma)$

Madras et al, 2018

Discrimination Measures for Representations

$$\mathcal{Z}_1 = p(Z|A = 1) \quad \mathcal{Z}_0 = p(Z|A = 0) \quad \mathcal{Z}_a^y = p(Z|A = a, Y = y)$$

- Demographic Parity

$$\Delta_{DP}(g) \triangleq d_g(\mathcal{Z}_0, \mathcal{Z}_1) = |\mathbb{E}_{\mathcal{Z}_0}[g] - \mathbb{E}_{\mathcal{Z}_1}[g]|$$

Demographic Parity: $P(\hat{Y} = 1|A = 1) = P(\hat{Y} = 1|A = 0)$

- Equality of Odds

$$\Delta_{EO}(g) \triangleq |\mathbb{E}_{\mathcal{Z}_0^0}[g] - \mathbb{E}_{\mathcal{Z}_1^0}[g]| + |\mathbb{E}_{\mathcal{Z}_0^1}[1-g] - \mathbb{E}_{\mathcal{Z}_1^1}[1-g]|$$

Equality of Odds: $P(\hat{Y} = 1|A = 0, Y) = P(\hat{Y} = 1|A = 1, Y)$

- Equality of Opportunities

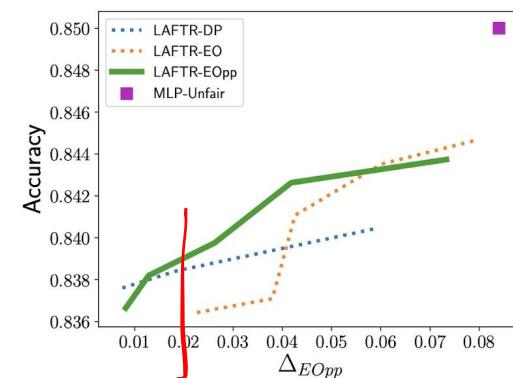
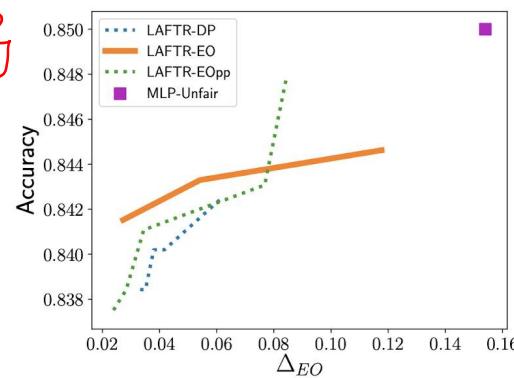
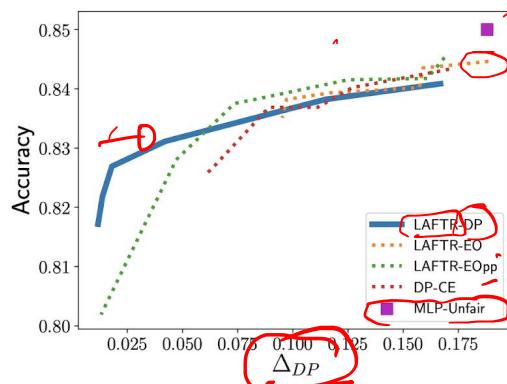
$$\Delta_{EOpp}(g) \triangleq |\mathbb{E}_{\mathcal{Z}_0^0}[g] - \mathbb{E}_{\mathcal{Z}_1^0}[g]|$$

Equality of Opportunity: $P(\hat{Y} = 1|A = 0, Y = 1) = P(\hat{Y} = 1|A = 1, Y = 1)$

Accuracy and Fairness on Adult Income Dataset

- Results Generated By Varying γ

$$L(f, g, h, k) = \alpha L_C(g(f(X, A)), Y) + \beta L_{Dec}(k(f(X, A), A), X) + \gamma L_{Adv}(h(f(X, A)), A)$$



$$\text{DP-CE} = \overline{\Pr(\hat{Y}=1 | A=0)} - \overline{\Pr(\hat{Y}=1 | A=1)}$$

DP-CE - Cross Entropy Adversarial Objective ([Edwards et al, 2016](#))

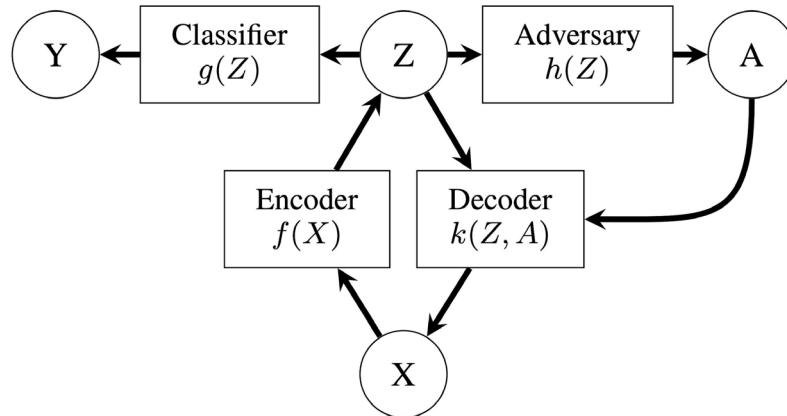
Δ_{EO}

$A=0, Y=1$

[Madras et al, 2018](#)

Transferring Fair Representations

- If the Representations Are Fair, All Predictors Should Be Fair!
 - Train f and g based on domain 1 with feature space \textcircled{X}
 - Fix f , and train g' on domain 2 with the same feature space X
 - $y=g'(f(x))$ should be a fairness predictor



[Madras et al, 2018](#)

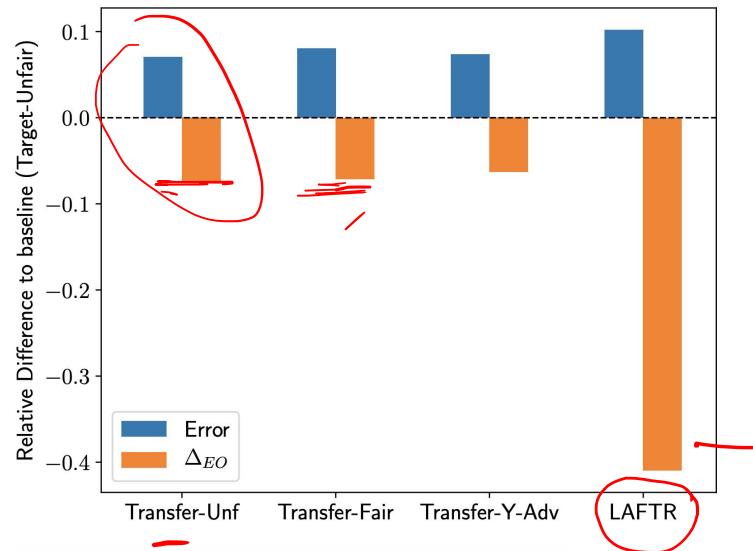
Transfer Fair Representations

- Heritage Health Dataset
 - Comprises insurance claims and physician records
 - Task 1 - Predict Charlson index (prediction of 10 year survival of patients) trained using equalized odds adversarial objective
 - Task 2 - Same input, task becomes predicting a patient's insurance claim corresponding to a specific medical condition

Transfer- unf - MLP with no fairness constraints.

Transfer- fair - MLP with fairness constraints in [Bechavod et al, 2017](#)

Transfer - Y - Adv baseline in [Zhang et al, 2018](#)



[Madras et al, 2018](#)

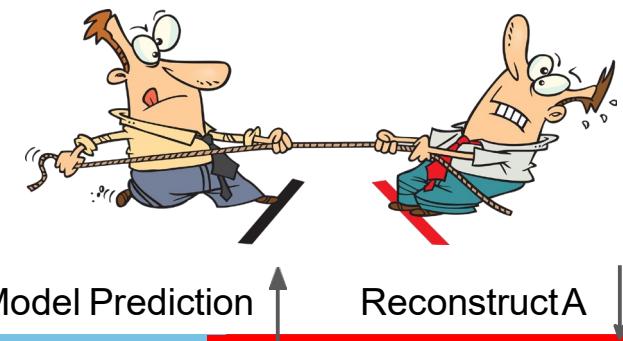
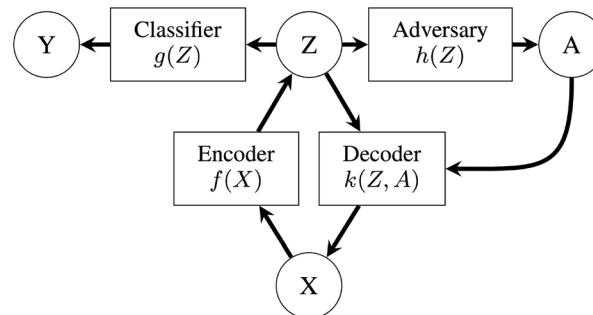
Recap

- Fair Representation Learning
 - Prejudice Removing Regularizer

$$-\mathcal{L}(\mathcal{D}; \boldsymbol{\Theta}) + \eta R(\mathcal{D}, \boldsymbol{\Theta}) + \frac{\lambda}{2} \|\boldsymbol{\Theta}\|_2^2$$

Loss of the Model Fairness Regularizer L2 Regularizer

- Fair Representations Through Adversarial Learning



Comparisons: Regularization and Adversarial Learning

	Prejudice Removing Regularizer	Adversarial Learning
Pros	Minimal modifications to training procedure	Transferable representations
		Can be applied to many different fairness criteria
Cons	Can only be applied to Demographic Parity	Adversarial loss can be difficult to train

Fair Data Manipulation

- Biased Data
 - The presence of data that belongs to the underrepresented groups leads to data biases
 - One of the main sources of ML discriminations
- Data Debiasing
 - Adjust the distribution of the data to meet fairness criteria
 - Increase/Decrease samples based on criteria
- Reweighting
 - Adjust the importance of each sample in the loss function during training
- Resampling
 - Adjust the proportion of samples for each group

Rest of the Topics

- Basic Data Manipulation Techniques
 - Reweighting
 - Practice question
 - Universal Sampling
 - Preferential Sampling
- Individual Fairness
- Optimized Pre-processing
- Learning to Defer

Biased Data



Observed: M = 10, F = 4



Expected Distribution of Fair Data

Masajiy

- Expected Data Distribution

$$P(Y) = P(Y|A = 1) = P(Y|A = 0)$$

which leads to $Y \perp\!\!\!\perp A$

- Recall Demographic Parity

$$P(\hat{Y} = 1|A = 1) = P(\hat{Y} = 1|A = 0)$$

Kamiran et al, 2012

Expected Distribution of Fair Data

- The Expected Joint Distribution Under $Y \perp\!\!\!\perp A$

$Y \perp\!\!\!\perp A$

$$\begin{aligned}
 P_{exp}(Y = y, A = a) &= P(Y = y) \cdot P(A = a) \\
 &= \frac{|\{x \in \mathcal{D} | x_Y = y\}|}{|\mathcal{D}|} \cdot \frac{|\{x \in \mathcal{D} | x_A = a\}|}{|\mathcal{D}|} \\
 &\quad \text{P}(Y, A) = P(Y) P(A)
 \end{aligned}$$

↓ Probability ↓

$$P_{obs}(Y = y, A = a) = \frac{|\{x \in \mathcal{D} | x_Y = y, x_A = a\}|}{|\mathcal{D}|}$$

Transform Data to
 Expected Distribution

- Our Observed Joint Distribution

$A = 0/1$
 $Y = 0/1$

Kamiran et al, 2012

Reweighting

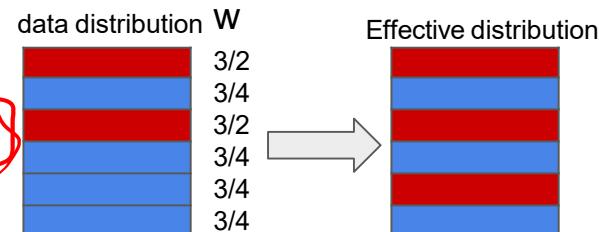
$$P(Y, A) = P(Y) \cdot P(A)$$

- Sample Weight for x
 - Goal: adjust our data to a distribution that leads to $Y \perp\!\!\!\perp A$, or Demographic Parity
 - $W(x) = 1$, we have achieved $Y \perp\!\!\!\perp A$ and Demographic Parity
 - $W(x) > 1$, increase the weight of sample x in training
 - $W(x) < 1$, decrease the weight of sample x in training

$$W(x) = \frac{P_{exp}(Y = x_y, A = x_a)}{P_{obs}(Y = x_y, A = x_a)}$$

- Reweighting Loss Function

$$\mathcal{L} = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} W(x) \mathcal{L}(\hat{Y}, x_Y)$$



Practice Question

- Calculate $W(x_3)$, $A = \{\text{Sex}\}$, $Y = \{\text{Class}\}$

Sex	Highest degree	Job type	Class
X ₁ M	H. school	Board	+
X ₂ M	Univ.	Board	+
X ₃ M	H. school	Board	+
M	H. school	Healthcare	+
M	Univ.	Healthcare	-
F	Univ.	Education	-
F	H. school	Education	-
F	None	Healthcare	+
F	Univ.	Education	-
X ₁₀ F	H. school	Board	+

Kamiran et al, 2012

Practice Question

$$P(A, Y) = P(A) P(Y) \Rightarrow Y \perp\!\!\!\perp A$$

- $W(x_3)$

- $A_3 = M$
- $Y_3 = +$

$$P(A = M) \cdot P(Y = +)$$

$A = \{\text{Sex}\}, Y = \{\text{Class}\}$

Sex	Highest degree	Job type	Class
-----	----------------	----------	-------

M	H. school	Board	+
M	Univ.	Board	+
M	H. school	Board	+
M	H. school	Healthcare	+
M	Univ.	Healthcare	-
F	Univ.	Education	-
F	H. school	Education	-
F	None	Healthcare	+
F	Univ.	Education	-
F	H. school	Board	+

- Expected Distribution

- $P(A = M) = 0.5$
- $P(Y = +) = 0.6$
- $P_{\text{exp}}(A = M, Y = +) = 0.3$

- Observed Distribution

- $P_{\text{obs}}(A = M, Y = +) = 0.4$

- Sample Weight

- $W(x_3) = 0.3/0.4 = 0.75$

Kamiran et al, 2012

Quiz

- Calculate $W(x_6)$, $A = \{\text{Sex}\}$, $Y = \{\text{Class}\}$

Sex	Highest degree	Job type	Class
M	H. school	Board	+
M	Univ.	Board	+
M	H. school	Board	+
M	H. school	Healthcare	+
M	Univ.	Healthcare	-
$W(x_6)$		F Univ. Education	-
F	H. school	Education	-
F	None	Healthcare	+
F	Univ.	Education	-
F	H. school	Board	+

Kamiran et al, 2012

Answer

- $W(x_6)$
 - $A_6 = F$
 - $Y_6 = -$
- Expected Distribution
 - $P(A = F) = 0.5$
 - $P(Y = -) = 0.4$
 - $P_{\text{exp}}(A = F, Y = -) = 0.2$
- Observed Distribution
 - $P_{\text{obs}}(A = F, Y = -) = 0.3$
- Sample Weight
 - $W(x_6) = 0.2/0.3 = 0.67$

Sex	Highest degree	Job type	Class
M	H. school	Board	+
M	Univ.	Board	+
M	H. school	Board	+
M	H. school	Healthcare	+
M	Univ.	Healthcare	-
F	Univ.	Education	-
F	H. school	Education	-
F	None	Healthcare	+
F	Univ.	Education	-
F	H. school	Board	+

Kamiran et al, 2012

Homework

- Calculate $W(x_1) \dots W(x_{10})$
- Put $W(x_i)$ into the loss

$$\mathcal{L} = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} W(x) \cdot \mathcal{L}(\hat{Y}, x_Y)$$

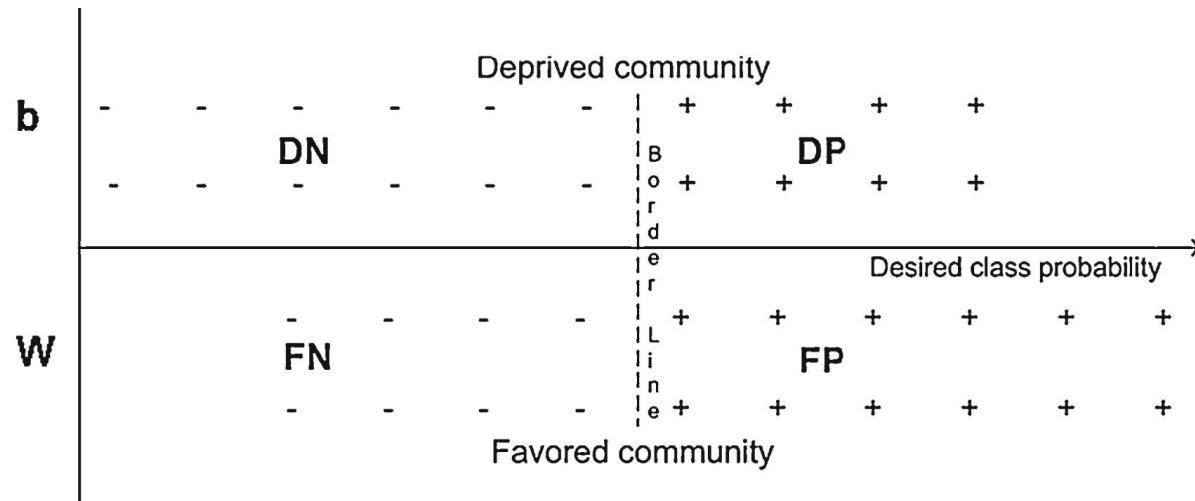
Can we achieve data pre-processing for fairness without changing the training objective?

$A = \{\text{Sex}\}$, $Y = \{\text{Class}\}$

Sex	Highest degree	Job type	Class
M	H. school	Board	+
M	Univ.	Board	+
M	H. school	Board	+
M	H. school	Healthcare	+
M	Univ.	Healthcare	-
F	Univ.	Education	-
F	H. school	Education	-
F	None	Healthcare	+
F	Univ.	Education	-
F	H. school	Board	+

Resampling

- Resample the Dataset Based on the Expected Joint Probability



[Kamiran et al, 2012](#)

Expected Number of Samples

- Expected Number of Samples for the Category (y, a)

$$N_{exp}(y, a) = P_{exp}(y, a) \cdot |\mathcal{D}|$$

- Also Note

$$\sum_{y,a} N_{exp} = \sum_{y,a} P_{exp}(y, a) \cdot |\mathcal{D}| = |\mathcal{D}|$$

Universal Resampling (US)

- Resampling Based on the Expected Probabilities to Meet Demographic Parity
 - DP (Deprived community with Positive class labels)
 - draw $N_{\text{exp}}(D, P)$ samples uniformly from DP
 - DN (Deprived community with Negative class labels)
 - draw $N_{\text{exp}}(D, N)$ samples uniformly from DN
 - FP (Favored community with Positive class labels)
 - draw $N_{\text{exp}}(F, P)$ samples uniformly from FP
 - FN (Favored community with Negative class labels)
 - draw $N_{\text{exp}}(F, N)$ samples uniformly from FN

(# is less) ↑ rate ↑

(# is more) ↑ ↓

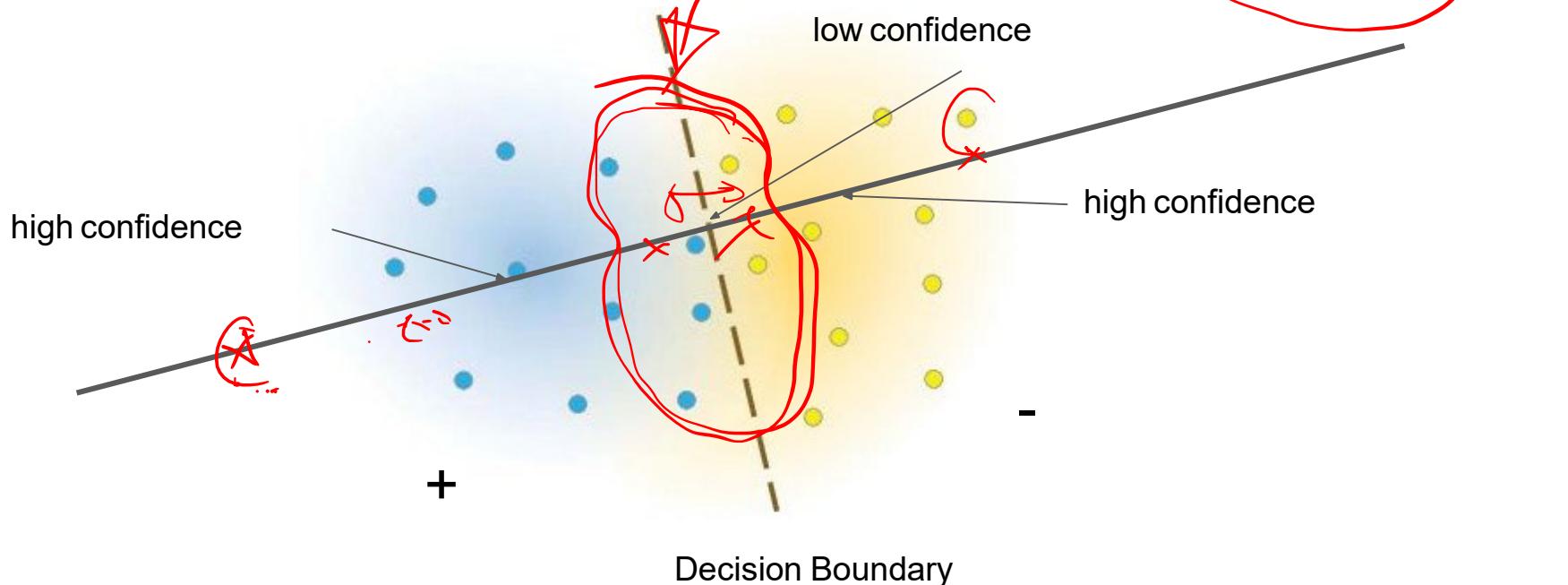
(# is more) ↓ ↑

(# is less) ↑ ↓

Kamiran et al, 2012

Preferential Sampling (PS)

- Sample More Data When Confidence of the Predictor Is Low



Kamiran et al, 2012

Bias Measures

- Measure prediction biases by comparing the favorable outcomes given to group 1 with that to group 0

$$Bias(\hat{Y}) = P(\hat{Y} = 1|A = 1) - P(\hat{Y} = 1|A = 0)$$

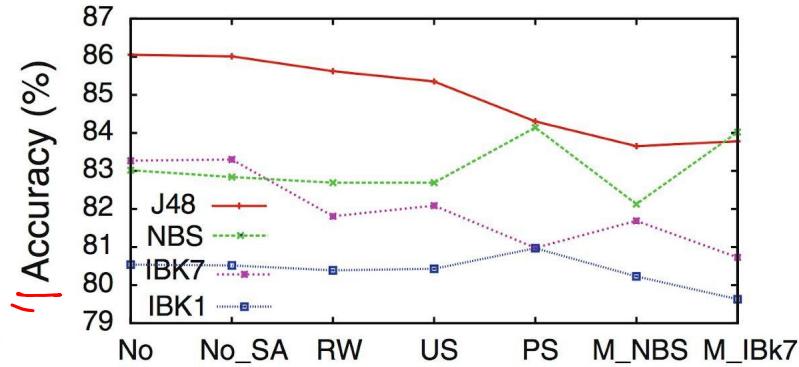
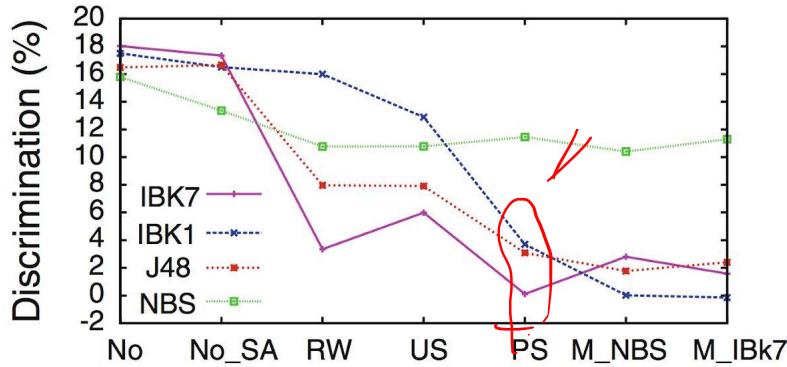
Demographic Parity

$$P(\hat{Y} = 1|A = 1) = P(\hat{Y} = 1|A = 0)$$

[Kamiran et al, 2012](#)

Adult Income Dataset

No - No pre-processing, No-SA - No Sex Attribute, RW - Reweighting
 US - Universal Sampling, PS - Preferential Sampling
 M_* - “massaged” input data (refer to the paper)



J48 - decision tree

NBS - Naive Bayes

IBK1- 1 nearest neighbor

IBK7 - 7 nearest neighbor

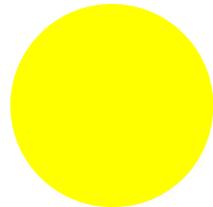
[Kamiran et al, 2012](#)

Continuous Data?

$$N_{exp}(y, a) = P_{exp}(y, a) \cdot |\mathcal{D}|$$

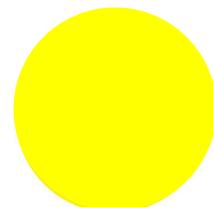
$$W(x) = \frac{P_{exp}(Y = x_y, A = x_a)}{P_{obs}(Y = x_y, A = x_a)}$$

Individual Fairness



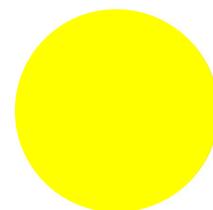
Income = \$50k
Credit Score = 690

Accepted



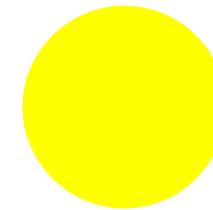
Income = \$43k
Credit Score = 650

Accepted



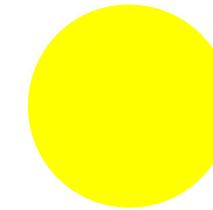
Income = \$50k
Credit Score = 690

Denied
???



Income = \$70k
Credit Score = 740

Accepted



Income = \$100k
Credit Score = 750

Accepted

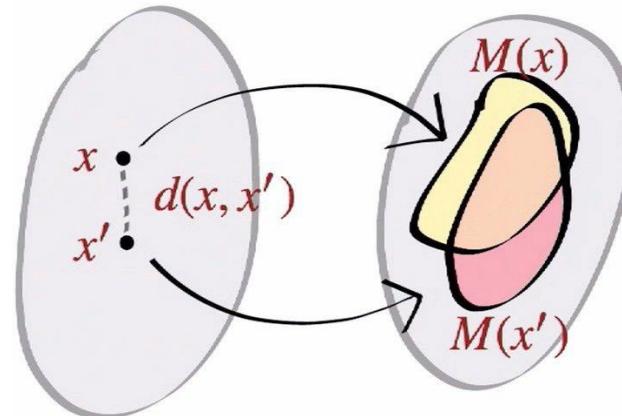
group 1

group 2

Individual Fairness

- A predictor M achieves individual fairness under a distance metric d iff
 - Similar Samples are treated similarly, in other words

$$M(x_i) \approx M(x_j) | d(x_i, x_j) \approx 0$$



Individual Fairness

Individual



Income = \$19k
Credit Score = 690



Income = \$23k
Credit Score = 720

$$M(x_i) \approx M(x_j) | d(x_i, x_j) \approx 0$$

Group 1



Income = \$60k
Credit Score = 800



Income = \$20k
Credit Score = 680



Income = \$27k
Credit Score = 700

Group 2



Income = \$65k
Credit Score = 810

Fairness Criteria

Individual Treatment	Group Treatment
Fairness Through Unawareness Excludes Sensitive Information A from the predictor	Demographic Parity $P(\hat{Y} = 1 A = 1) = P(\hat{Y} = 1 A = 0)$
Individual Fairness	Equal Opportunity/Odds
$M(x_i) \approx M(x_j) d(x_i, x_j) \approx 0$	$P(\hat{Y} = 1 A = 0, Y = 1) = P(\hat{Y} = 1 A = 1, Y = 1)$ $P(\hat{Y} = 1 A = 0, Y) = P(\hat{Y} = 1 A = 1, Y)$

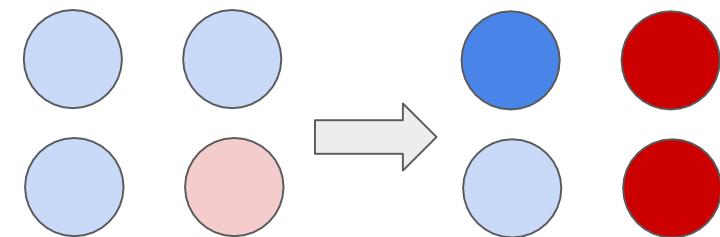
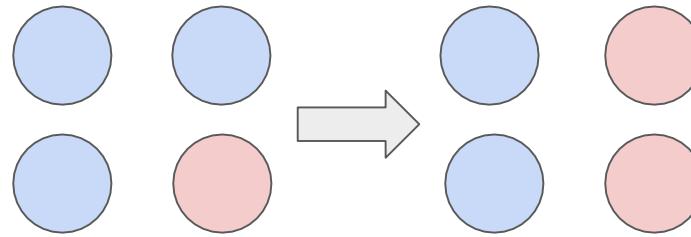
Optimized Pre-Processing for Fairness

- Can We Automate the Resampling Process?
 - Turn the manual process into an optimization based approach
 - Include more criteria than Demographic Fairness
 - Allow transformations of data
- Optimized Pre-Processing
 - Given sensitive feature D, learn a probabilistic mapping $p_{\hat{X}, \hat{Y}|X, Y, D}$ that transfers
 - Satisfies three constraints

$$\{(D_i, X_i, Y_i)\}_{i=1}^n \xrightarrow{p_{\hat{X}, \hat{Y}|X, Y, D}} \{(D_i, \hat{X}_i, \hat{Y}_i)\}_{i=1}^n$$

Calmon et al, 2017

Resampling and Transforming



Constraint 1: Utility Preservations

- A Utility Function to Preserve the Joint Probability
 - e.g. KL Divergence

$$p_{\hat{X}, \hat{Y}} \quad \longleftrightarrow \quad p_{X, Y}$$

transformed data original data

Calmon et al. 2017

Constraint 2: Discrimination Control

- Constrain the dependency of the target variable y given sensitive feature d to match target $p_{Y_T}(y)$
 - J - distance measure $J(p, q) = \left| \frac{p}{q} - 1 \right|$
 - $\epsilon_{y,d}$ - a small number used as our tolerance

$$J \left(p_{\hat{Y}|D}(y|d), p_{Y_T}(y) \right) \leq \epsilon_{y,d} \quad \forall d \in \mathcal{D}, y \in \{0, 1\}$$

When $p_{\hat{Y}|D}(y|d) = p_{Y_T}(y)$, we achieve Demographic Parity

Calmon et al. 2017

Constraint 3: Distortion Control

- An Implementation of the Individual Fairness

$$M(x_i) \approx M(x_j) | d(x_i, x_j) \approx 0$$

- The Mapped Sample \hat{X}, \hat{Y} Has to Stay Close to the Original Sample x, y
 - $c_{d,x,y}$ - tolerance
 - δ - a similarity function
 - 1 - very different
 - 0 - very similar

$$\Pr \left(\delta((x, y), (\hat{X}, \hat{Y})) = 1 \mid D = d, X = x, Y = y \right) \leq c_{d,x,y}$$

[Calmon et al. 2017](#)

Putting Things Together

$$\begin{aligned}
 & \min_{p_{\hat{X}, \hat{Y}|X, Y, D}} \Delta \left(p_{\hat{X}, \hat{Y}}, p_{X, Y} \right) \\
 \text{s.t. } & J \left(p_{\hat{Y}|D}(y|d), p_{Y_T}(y) \right) \leq \epsilon_{y,d} \text{ and} \\
 & \mathbb{E} \left[\delta((x, y), (\hat{X}, \hat{Y})) \mid D = d, X = x, Y = y \right] \leq c_{d,x,y}
 \end{aligned}$$

Utility
 Discrimination control
 group fairness

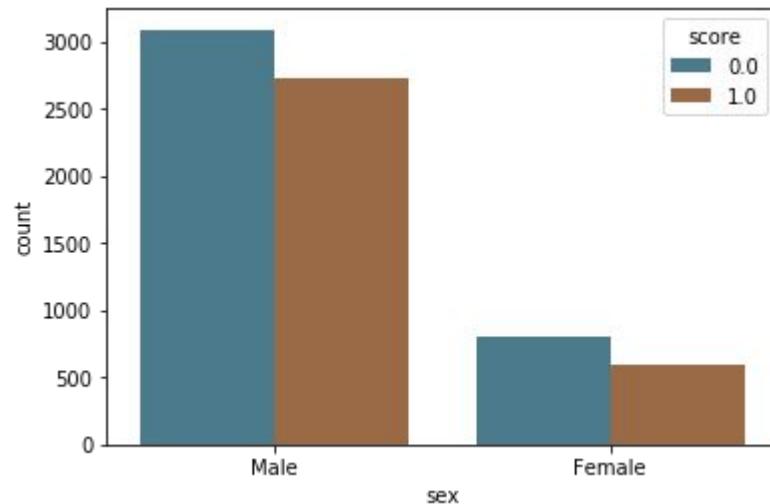
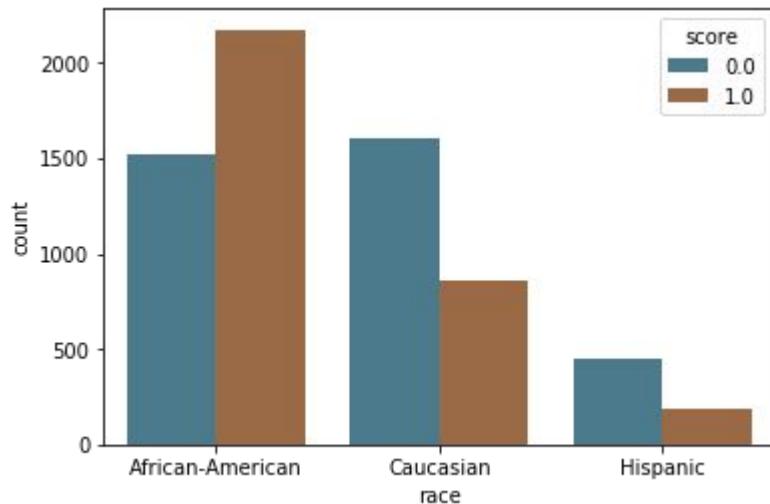
↑
 Distortion Control
 Individual fairness

[Calmon et al. 2017](#)

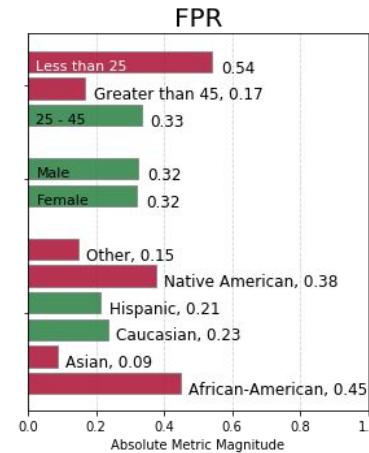
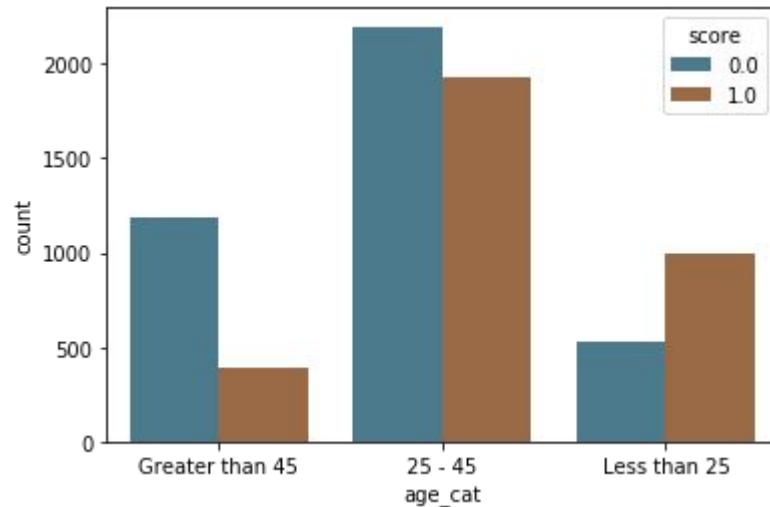
COMPAS Dataset



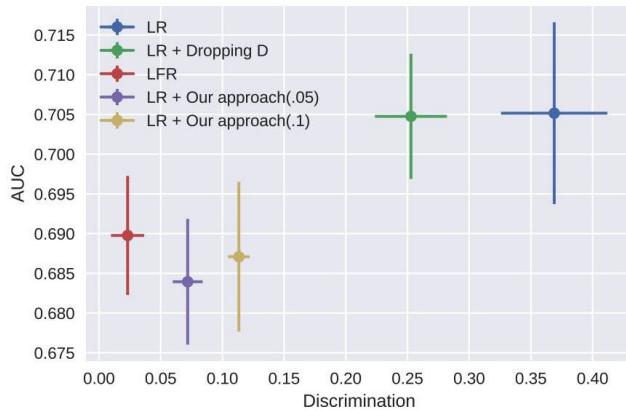
COMPAS Dataset



COMPAS Dataset

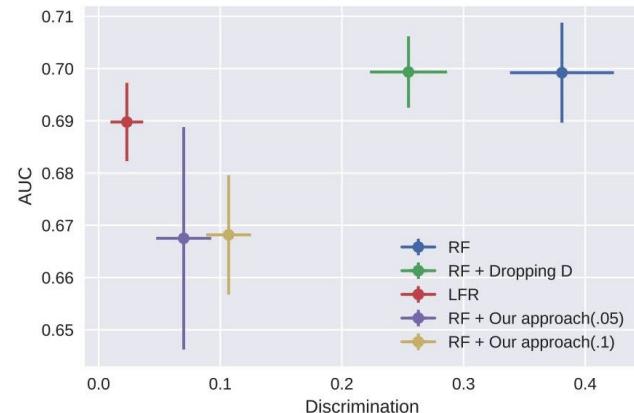


Results on COMPAS dataset



Logistic Regression

LFR - Learning Fair Representations ([Zemel et al, 2013](#))

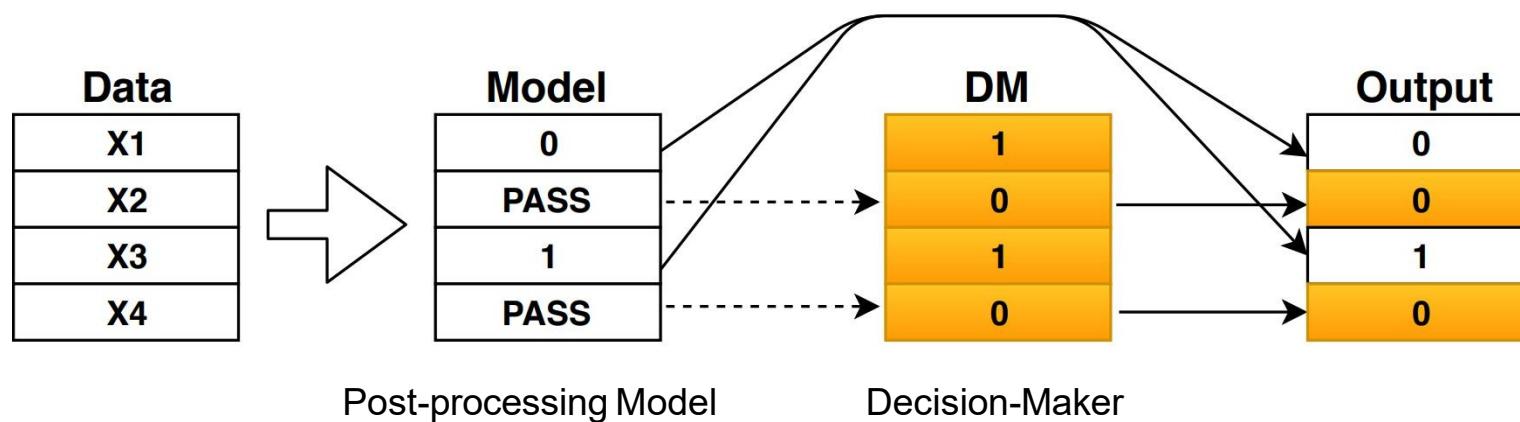


Random Forest

[Calmon et al. 2017](#)

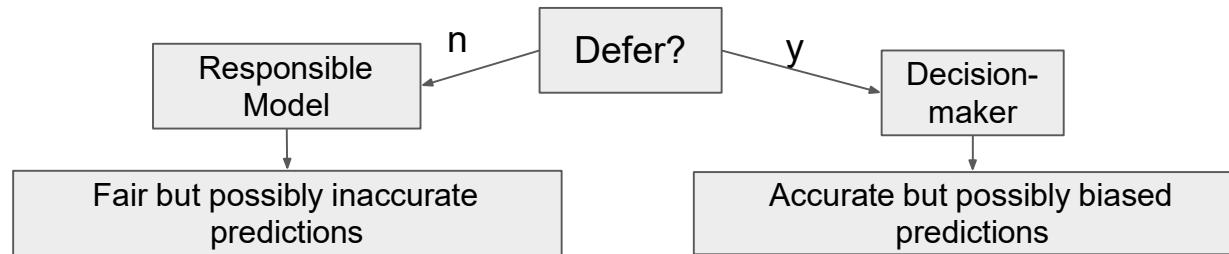
Post-Processing Methods for Fairness

- Why Post-Processing?
 - Flexibility: No need to fine-tune the ML model
 - Model Agnostic: Can be applied across a wide range of models
- Learning to Defer



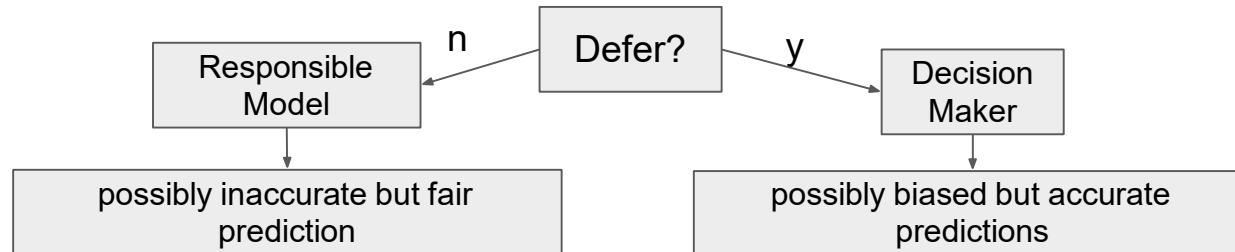
Learning to Defer

- Working Together with A Black-box Decision-maker Model
 - Decision-maker models (e.g. human) have access to important information that our model does not have
 - Decision-maker models might be biased
- Performance and Fairness Trade-offs
 - Fix the unfair predictions of the decision-maker model
 - Defer to the decision-maker the model is uncertain

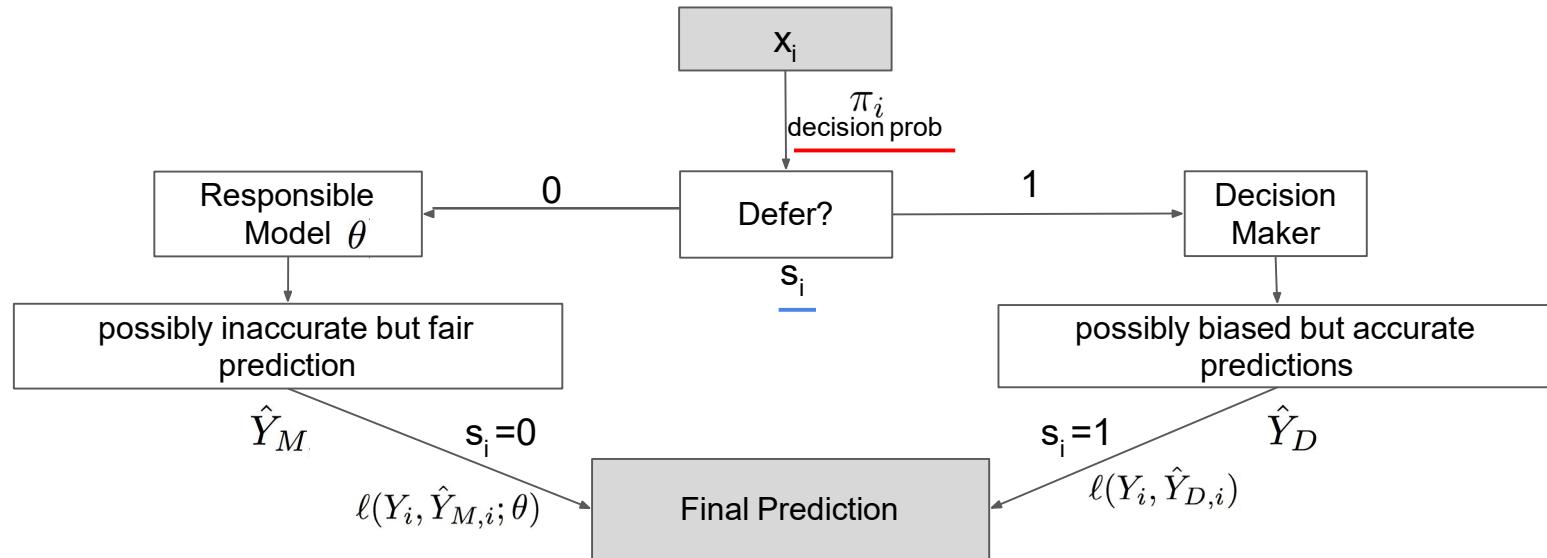


Learning to Defer

- Decision-maker Model
 - Considered as a black-box model
 - No fine-tuning, no access to its training data
- Responsible Model
 - Have access to additional data
 - Stick to fairness constraints



Training the Defer Model



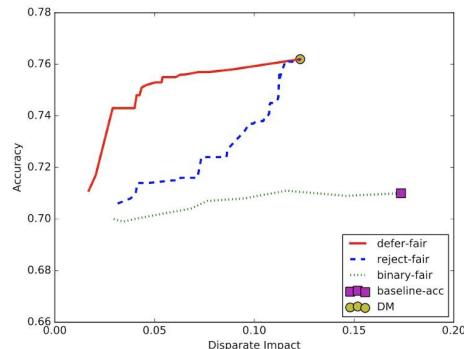
$$\mathcal{L}_{defer}(Y, \hat{Y}_M, \hat{Y}_D, \pi; \theta) = \sum_i \mathbb{E}_{s_i \sim Ber(\pi_i)} [(1 - s_i)\ell(Y_i, \hat{Y}_{M,i}; \theta) + s_i\ell(Y_i, \hat{Y}_{D,i})] + \alpha_{fair} \mathcal{R}(Y, \hat{Y}_M, \hat{Y}_D, s)$$

Fair regularizer

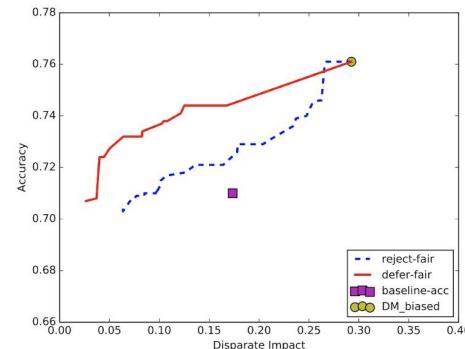
Madras et al. 2018

Results on COMPAS

- DM Model
 - High-Accuracy - DM has more data, Highly-Biased - DM is extremely biased



COMPAS, High-Accuracy DM



COMPAS, Highly-Biased DM

- DM - Decision-maker model
- Defer - Fair - Learning to Defer
- Reject- Fair - Only reject or accept DM
- Baseline - Model trained only to optimize accuracy, no DM
- Binary - Fair - Baseline optimized with fairness



BITS Pilani
Pilani Campus

Fair, Accountable, Transparent Machine Learning (FAccT ML) ZG517

Dr. Sugata Ghosal
sugata.ghosal@pilani.bits-pilani.ac.in



Session 5
Date – 18th June 2023
Time – 8:45 AM to 10:45 PM

These slides are prepared by the instructor, with grateful acknowledgement of many others who made their course materials freely available online.

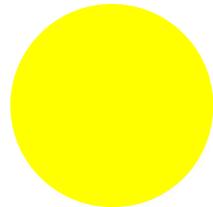
Readings

- <https://www.borealisai.com/research-blogs/tutorial1-bias-and-fairness-ai/>
 - Flavio Calmon et al., Optimized Preprocessing for Discrimination Prevention, 2017
 - Matt Kusner et al., Counterfactual Fairness, 2018
 - Chris Russell et al., When Worlds Collide: Integrating Different Counterfactual Assumptions in Fairness, 2017
 - Stephen Pfahl et al., Counterfactual Reasoning for Fair Clinical Risk Prediction, 2019
-

Agenda

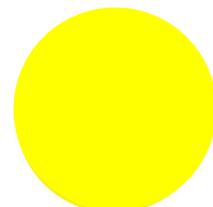
- Preprocessing for Fairness
 - Basic Data Manipulation Techniques
 - Reweighting
 - Practice question
 - Universal Sampling
 - Preferential Sampling
 - Individual Fairness
 - Optimized Pre-processing
 - Post-processing for Fairness: Learning to Defer
 - Fair Causal Reasoning
 - Counterfactual Fairness
 - Equalized Counterfactual Odds
 - Multiple Causal Worlds
-

Individual Fairness



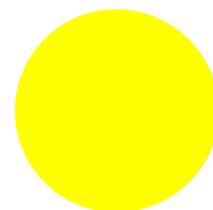
Income = \$50k
Credit Score = 690

Accepted



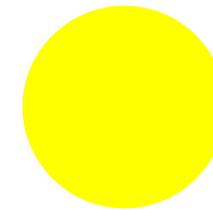
Income = \$43k
Credit Score = 650

Accepted



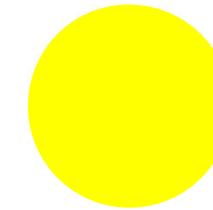
Income = \$50k
Credit Score = 690

Denied
???



Income = \$70k
Credit Score = 740

Accepted



Income = \$100k
Credit Score = 750

Accepted

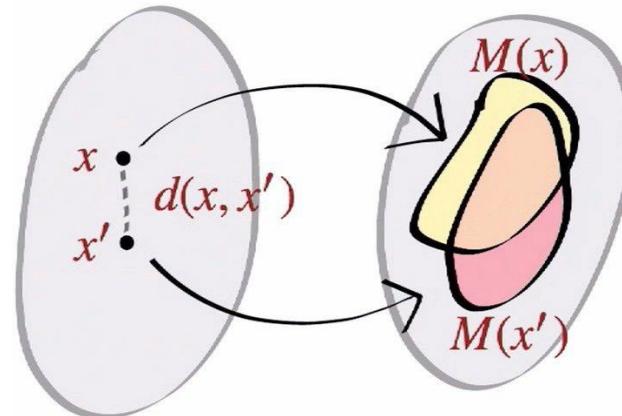
group 1

group 2

Individual Fairness

- A predictor M achieves individual fairness under a distance metric d iff
 - Similar Samples are treated similarly, in other words

$$M(x_i) \approx M(x_j) | d(x_i, x_j) \approx 0$$



Individual Fairness

Individual



Income = \$19k
Credit Score = 690



Income = \$23k
Credit Score = 720

$$M(x_i) \approx M(x_j) | d(x_i, x_j) \approx 0$$

Group 1



Income = \$60k
Credit Score = 800



Income = \$20k
Credit Score = 680



Income = \$27k
Credit Score = 700

Group 2



Income = \$65k
Credit Score = 810

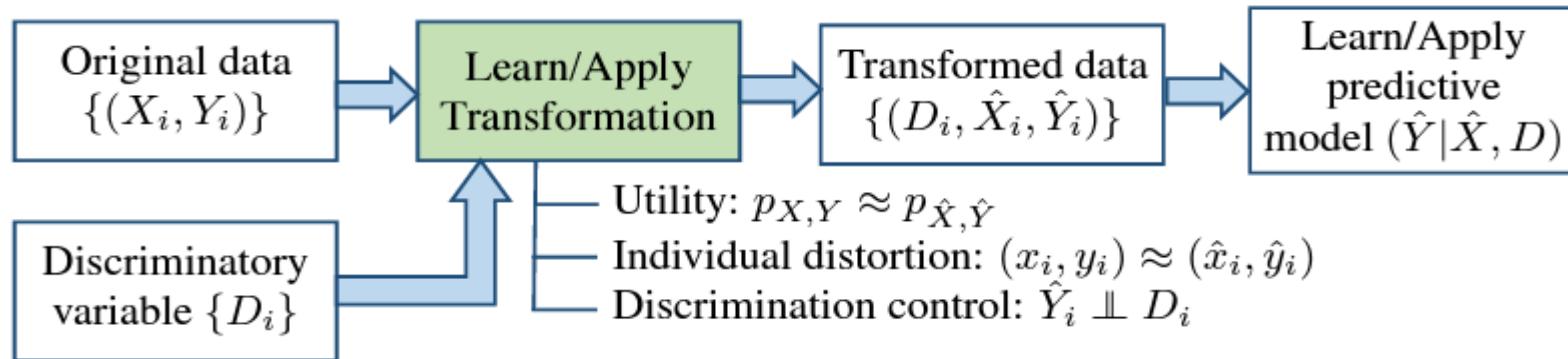
Fairness Criteria

Individual Treatment	Group Treatment
Fairness Through Unawareness Excludes Sensitive Information A from the predictor	Demographic Parity $P(\hat{Y} = 1 A = 1) = P(\hat{Y} = 1 A = 0)$
Individual Fairness	Equal Opportunity/Odds
$M(x_i) \approx M(x_j) d(x_i, x_j) \approx 0$	$P(\hat{Y} = 1 A = 0, Y = 1) = P(\hat{Y} = 1 A = 1, Y = 1)$ $P(\hat{Y} = 1 A = 0, Y) = P(\hat{Y} = 1 A = 1, Y)$

Optimized Pre-Processing for Fairness

- Can We Automate the Resampling Process?

- Turn the manual process into an optimization based approach
- Include more criteria than Demographic Fairness
- Allow transformations of data



Calmon et al, 2017

Optimized Pre-Processing for Fairness

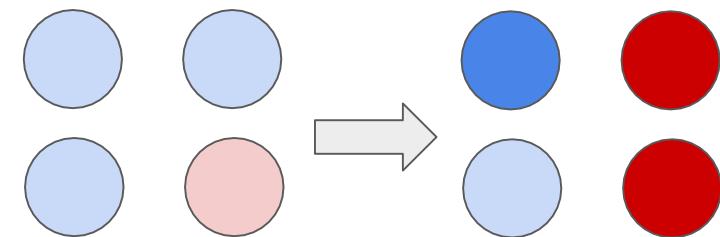
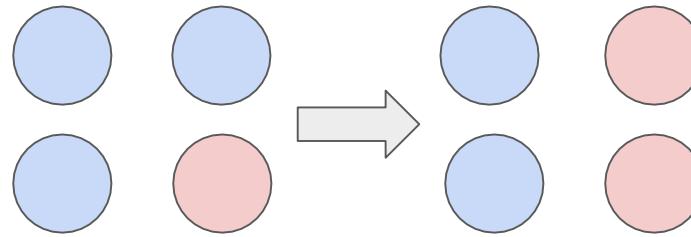
- Optimized Pre-Processing

- Given sensitive feature D, learn a probabilistic mapping $p_{\hat{X}, \hat{Y}|X, Y, D}$ that transfers
- Satisfies three constraints

$$\{(D_i, X_i, Y_i)\}_{i=1}^n \xrightarrow{p_{\hat{X}, \hat{Y}|X, Y, D}} \{(D_i, \hat{X}_i, \hat{Y}_i)\}_{i=1}^n$$

Calmon et al, 2017

Resampling and Transforming



Constraint 1: Utility Preservations

- A Utility Function to Preserve the Joint Probability of transformed data
 - e.g. KL Divergence

$$p_{\hat{X}, \hat{Y}} \quad \longleftrightarrow \quad p_{X, Y}$$

transformed data original data

Calmon et al. 2017

Constraint 2: Discrimination Control

- Constrain the dependency of the target variable y given sensitive feature d to match target $p_{Y_T}(y)$
 - J - distance measure $J(p, q) = \left| \frac{p}{q} - 1 \right|$
 - $\epsilon_{y,d}$ - a small number used as our tolerance

$$J \left(p_{\hat{Y}|D}(y|d), p_{Y_T}(y) \right) \leq \epsilon_{y,d} \quad \forall d \in \mathcal{D}, y \in \{0, 1\}$$

When $p_{\hat{Y}|D}(y|d) = p_{Y_T}(y)$, we achieve Demographic Parity

Calmon et al. 2017

Constraint 3: Distortion Control

- An Implementation of the Individual Fairness

$$M(x_i) \approx M(x_j) | d(x_i, x_j) \approx 0$$

- The Mapped Sample \hat{X}, \hat{Y} Has to Stay Close to the Original Sample x, y
 - $c_{d,x,y}$ - tolerance
 - δ - a similarity function
 - 1 - very different
 - 0 - very similar

$$\Pr \left(\delta((x, y), (\hat{X}, \hat{Y})) = 1 \mid D = d, X = x, Y = y \right) \leq c_{d,x,y}$$

[Calmon et al. 2017](#)

Putting Things Together

$$\begin{aligned}
 & \min_{p_{\hat{X}, \hat{Y}|X, Y, D}} \Delta \left(p_{\hat{X}, \hat{Y}}, p_{X, Y} \right) \\
 \text{s.t. } & J \left(p_{\hat{Y}|D}(y|d), p_{Y_T}(y) \right) \leq \epsilon_{y,d} \text{ and} \\
 & \mathbb{E} \left[\delta((x, y), (\hat{X}, \hat{Y})) \mid D = d, X = x, Y = y \right] \leq c_{d,x,y}
 \end{aligned}$$

Utility
 Discrimination control
 group fairness

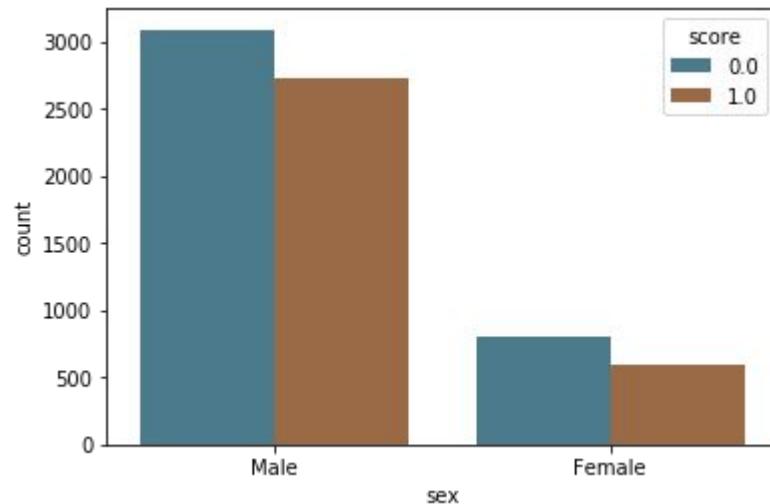
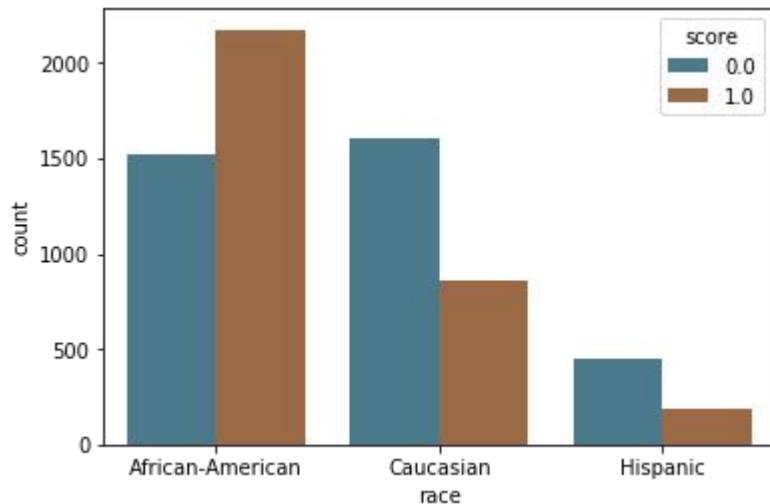
↑
 Distortion Control
 Individual fairness

[Calmon et al. 2017](#)

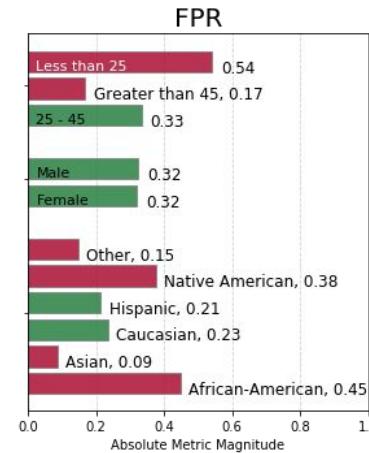
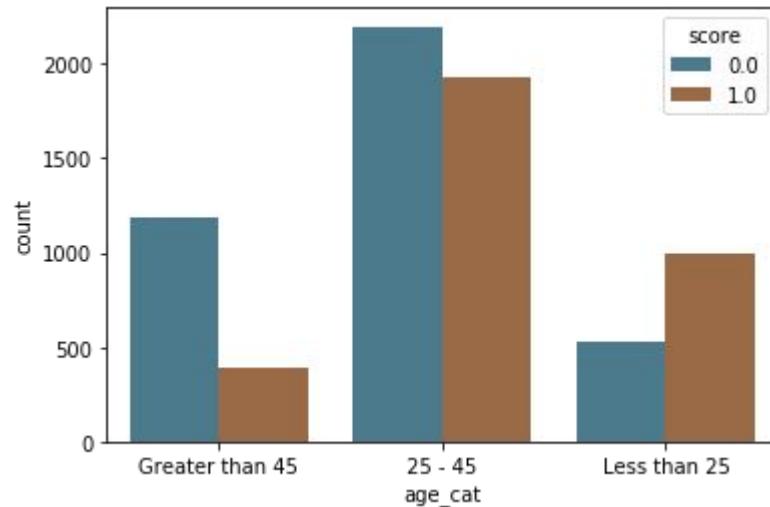
COMPAS Dataset



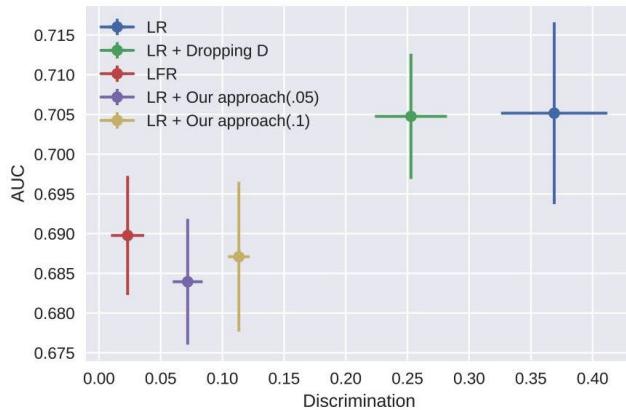
COMPAS Dataset



COMPAS Dataset

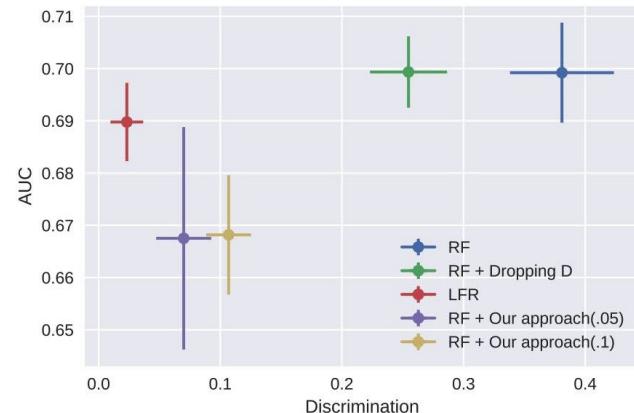


Results on COMPAS dataset



Logistic Regression

LFR - Learning Fair Representations ([Zemel et al, 2013](#))

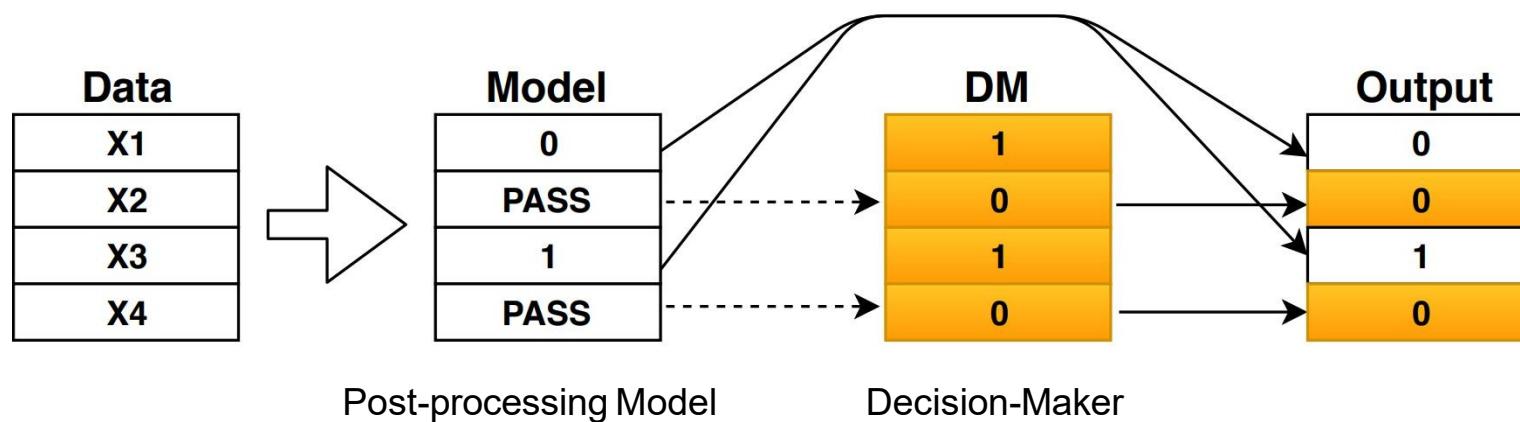


Random Forest

[Calmon et al. 2017](#)

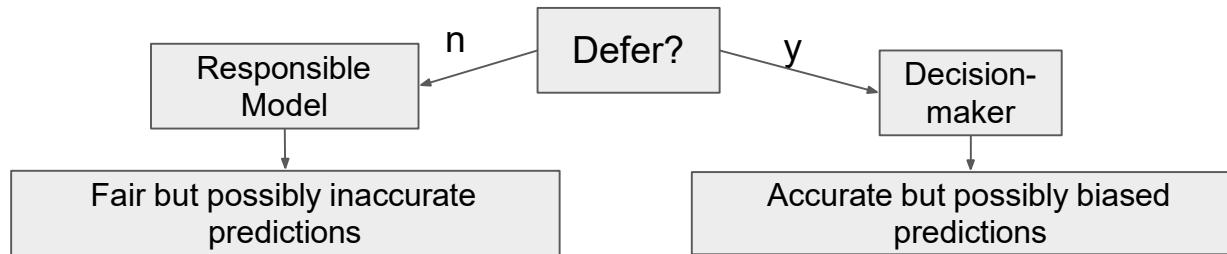
Post-Processing Methods for Fairness

- Why Post-Processing?
 - Flexibility: No need to fine-tune the ML model
 - Model Agnostic: Can be applied across a wide range of models
- Learning to Defer



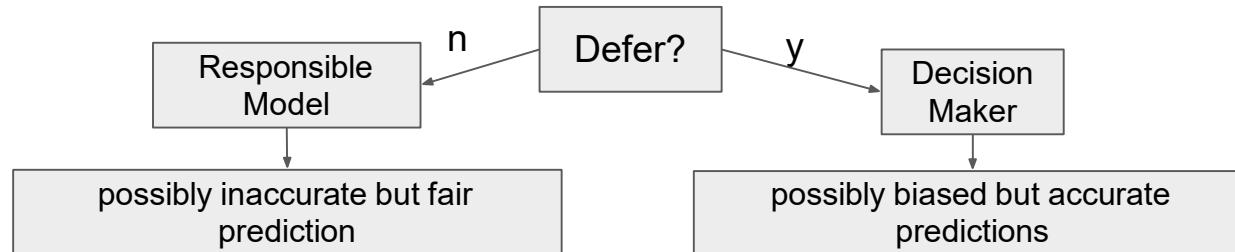
Learning to Defer

- Working Together with A Black-box Decision-maker Model
 - Decision-maker models (e.g. human) have access to important information that our model does not have
 - Decision-maker models might be biased
- Performance and Fairness Trade-offs
 - Fix the unfair predictions of the decision-maker model
 - Defer to the decision-maker the model is uncertain

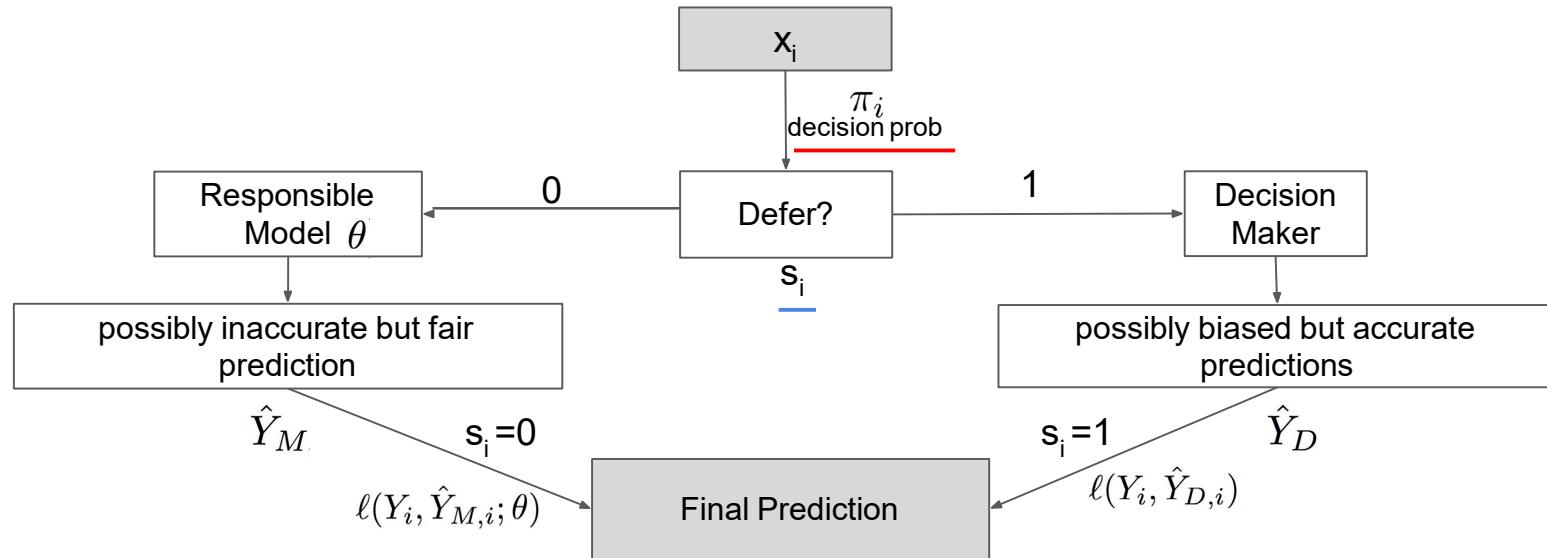


Learning to Defer

- Decision-maker Model
 - Considered as a black-box model
 - No fine-tuning, no access to its training data
- Responsible Model
 - Have access to additional data
 - Stick to fairness constraints



Training the Defer Model

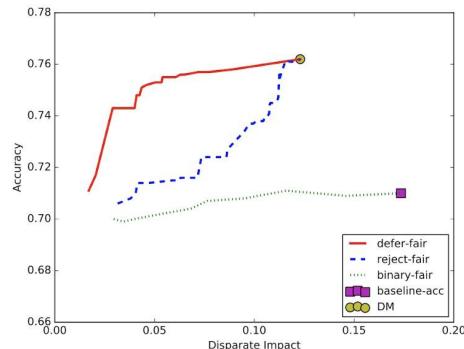


$$\mathcal{L}_{defer}(Y, \hat{Y}_M, \hat{Y}_D, \pi; \theta) = \sum_i \underbrace{\mathbb{E}_{s_i \sim Ber(\pi_i)}[(1 - s_i)\ell(Y_i, \hat{Y}_{M,i}; \theta) + s_i\ell(Y_i, \hat{Y}_{D,i})]}_{\text{Fair regularizer}} + \alpha_{fair} \mathcal{R}(Y, \hat{Y}_M, \hat{Y}_D, s)$$

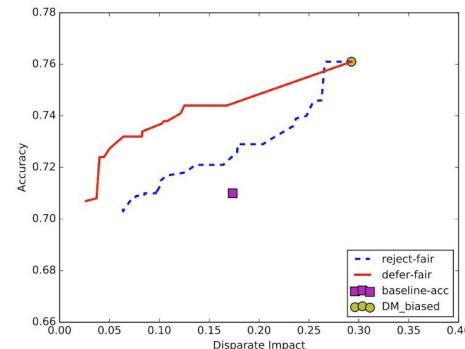
Madras et al. 2018

Results on COMPAS

- DM Model
 - High-Accuracy - DM has more data, Highly-Biased - DM is extremely biased



COMPAS, High-Accuracy DM



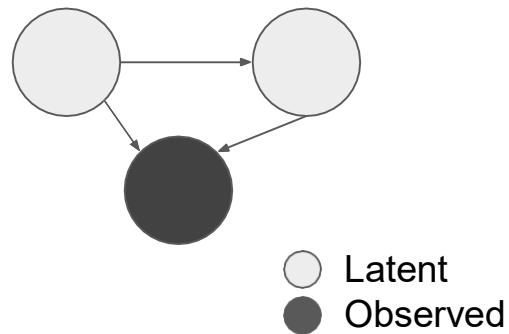
COMPAS, Highly-Biased DM

- DM - Decision-maker model
- Defer - Fair - Learning to Defer
- Reject- Fair - Only reject or accept DM
- Baseline - Model trained only to optimize accuracy, no DM
- Binary - Fair - Baseline optimized with fairness

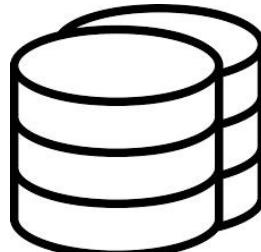
Fair Causal Reasoning

Causal Graph

- Observed Data
- Latent Data
- Relations



Observed Data



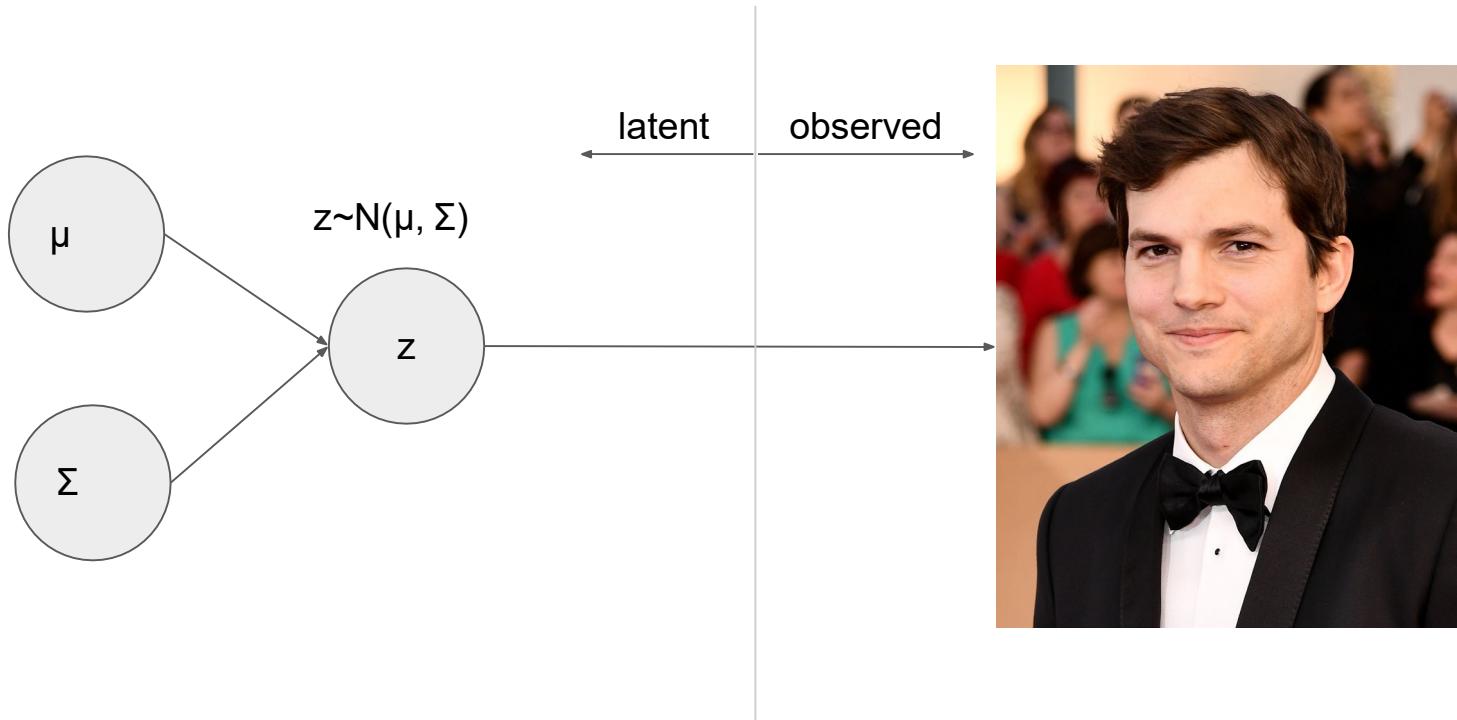
Causal Fairness Criteria

- Counterfactual Fairness

Observational Fairness Criteria

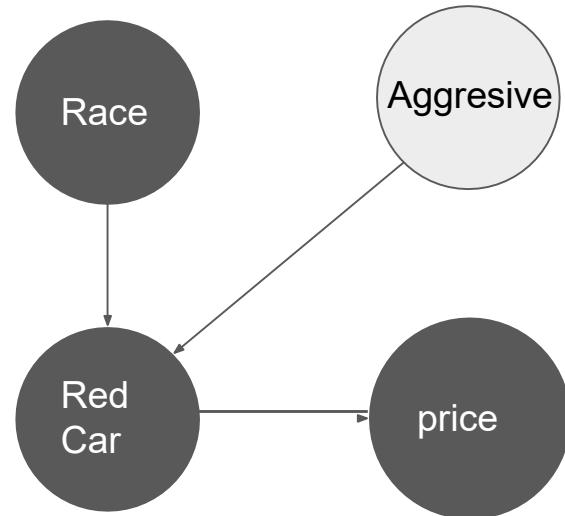
- Fairness Through Unawareness
- Demographic Parity
- Equalized Odds/Opp

Causal Graph



Why Do We Need Causal Fairness?

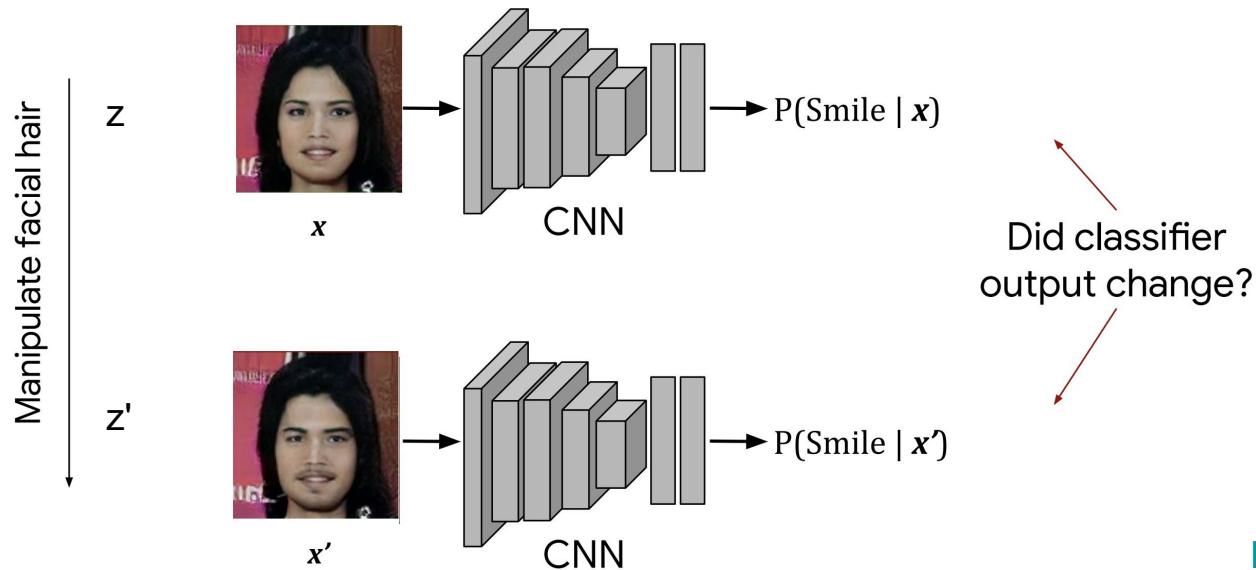
- Recover Latent Variables



[Kusner et al, 2018](#)

Why Do We Need Causal Fairness?

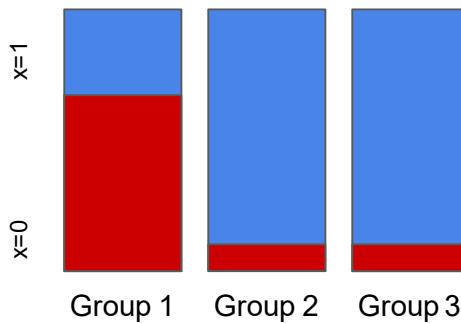
- Recover Latent Variables



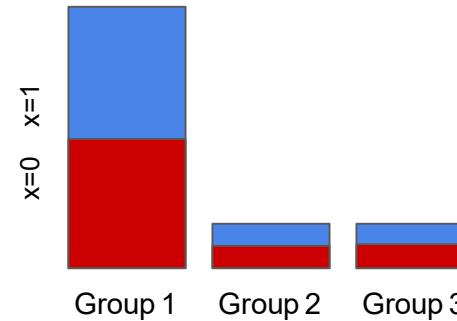
[Kusner et al, 2018](#)

Why Do We Need Causal Fairness?

- Dealing with Inherent bias



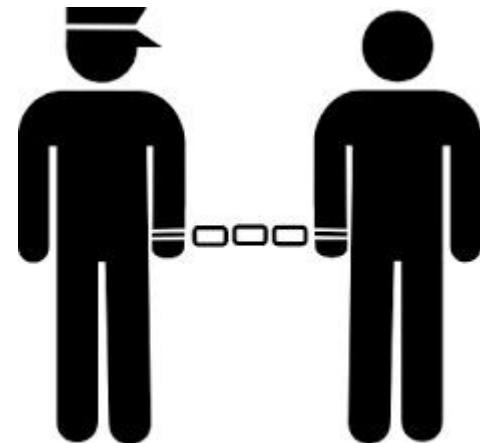
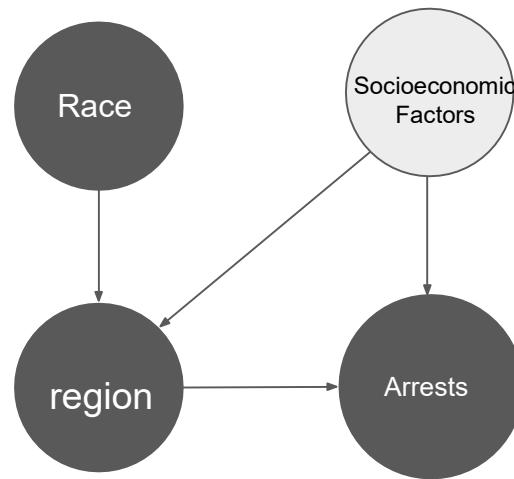
Inherent Biases



Sampling Biases

Inherent bias

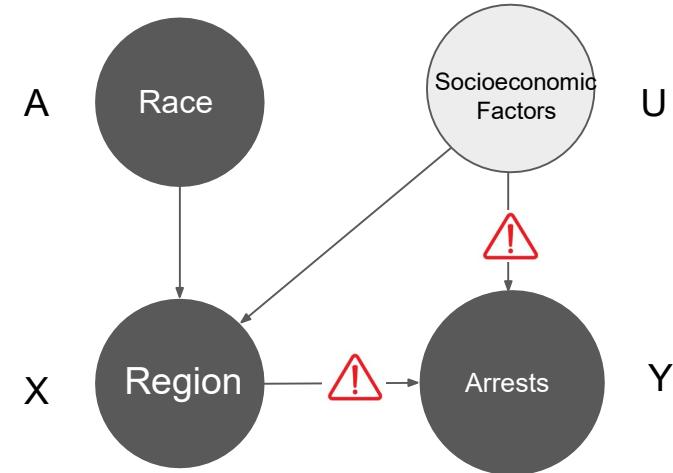
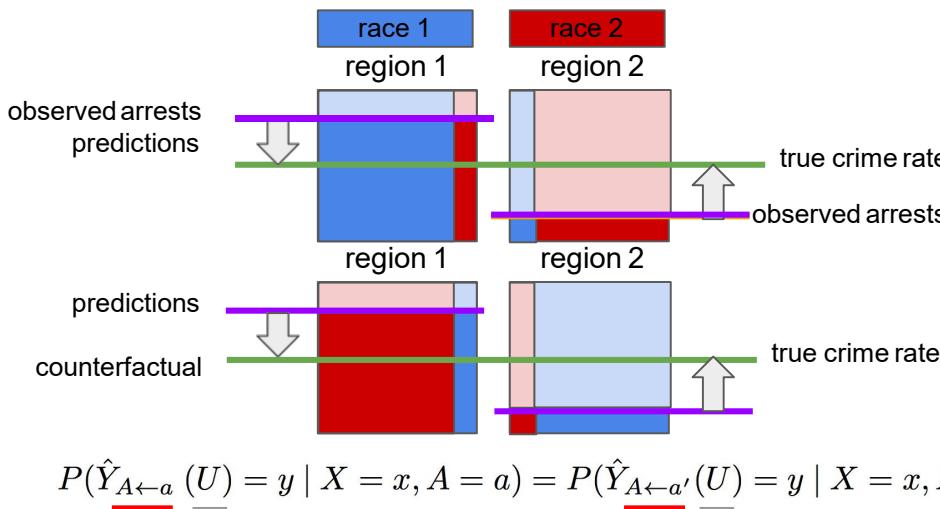
- Race groups live in certain regions due to socioeconomic status
- Latent Socioeconomic factors
 - More police resources in regions with low economic status
 - Results in more arrests



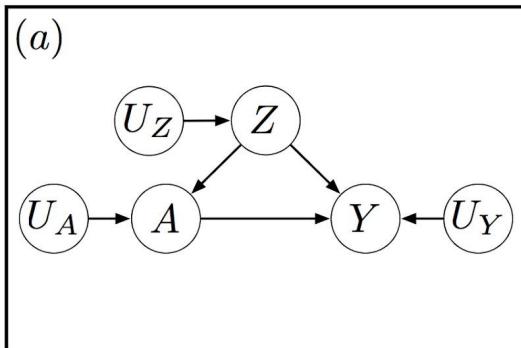
[Kusner et al, 2018](#)

Inherent bias

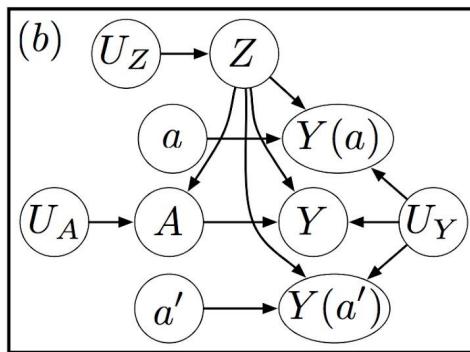
- Causal Fairness
 - Intervene variables in a causal graph
 - Generating samples with races that live in neighborhood that have high police resources



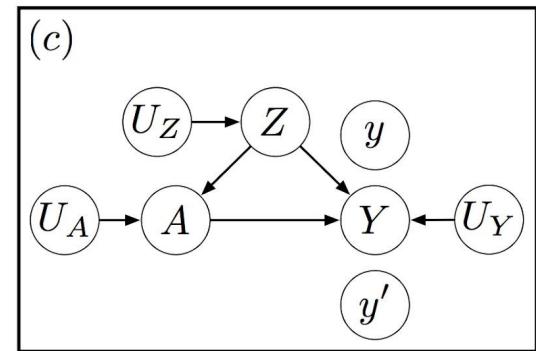
Intervention on Causal Graphs



Causal Graph with A , Z , Y



Intervene on A

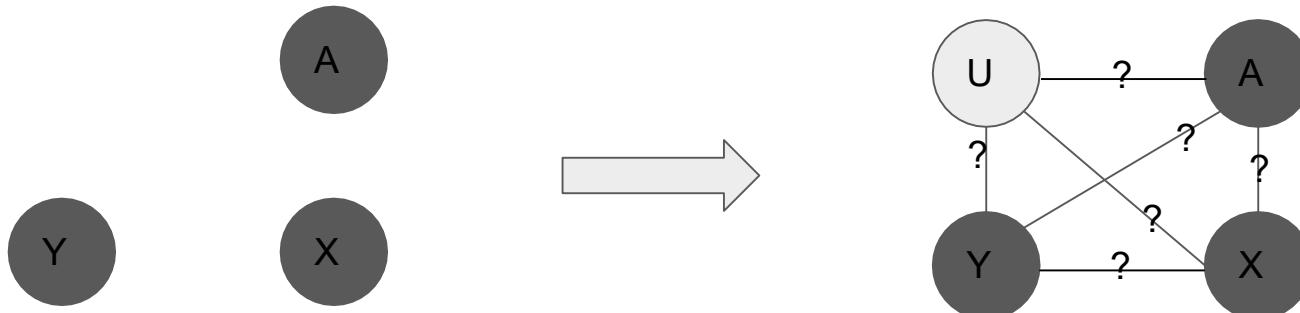


Intervene on Y

Outline

- Fair Causal Reasoning
- Counterfactual Fairness
 - Formal Methods
 - Law School
 - Crime Rates in NYC
- Equalized Counterfactual Odds
- Multiple Causal Worlds

Counterfactual Fairness Revisited



$$P(\hat{Y}_{A \leftarrow a}(U) = y \mid X = x, A = a) = P(\hat{Y}_{A \leftarrow a'}(U) = y \mid X = x, A = a)$$

Real Examples

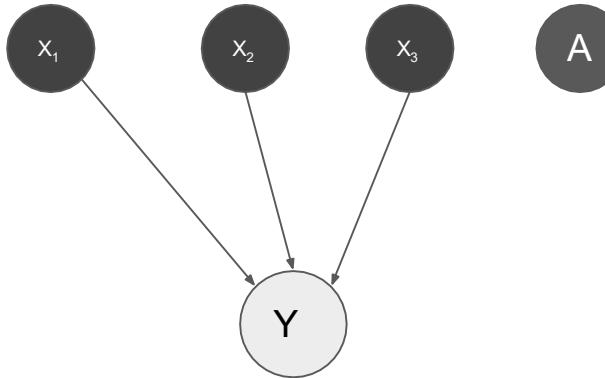
Intervention on $A \leftarrow a$

Counterfactual Examples

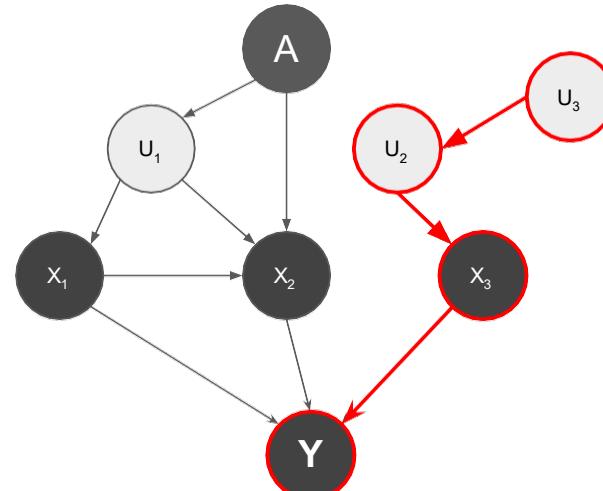
Intervention on $A \leftarrow a'$

Counterfactual Fairness

- Level 1
 - Build predictors using only the observable non-descendants of A

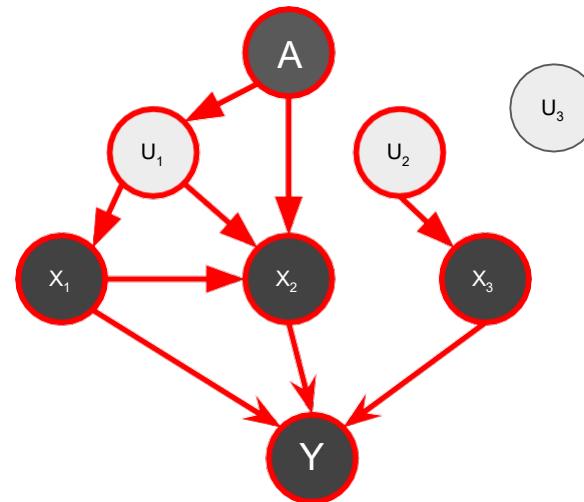


Fairness Through Unawareness



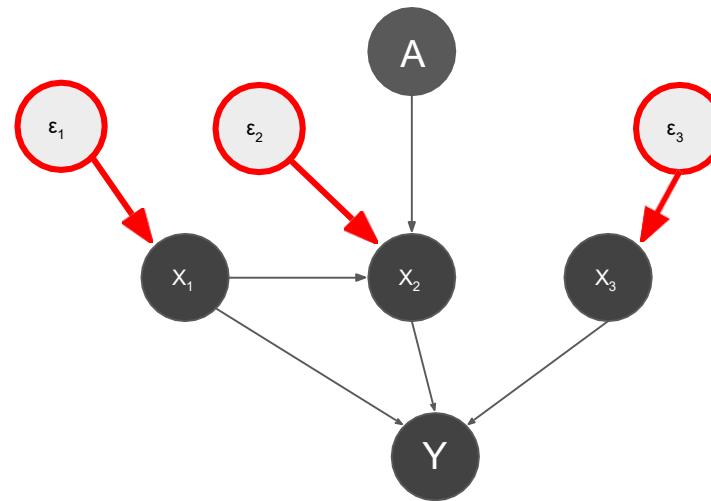
Counterfactual Fairness

- Level 2
 - Build Predictors using the parents of the observable variables



Counterfactual Fairness

- Level 3
 - Build Predictors by adding independent error terms

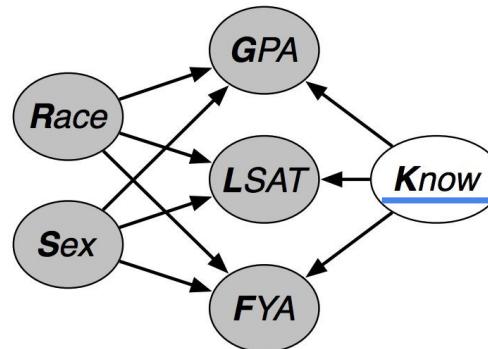


Law School Success Dataset

- Conducted by Law School Admission Council in US
 - 21,790 law students
 - Entrance exam scores (LSAT)
 - Grade-point average (GPA) collected prior to law school
 - Prediction Y = first year average grade (FYA)
 - Protected features = {Gender, Race}

Level 2 Counterfactual Fairness

- Build Predictors using the parents of the observable variables



$$K \sim \mathcal{N}(0, 1)$$

$$\text{FYA} \sim \mathcal{N}(w_F^K K + w_F^R R + w_F^S S, 1) \quad \text{GPA} \sim \mathcal{N}(b_G + w_G^K K + w_G^R R + w_G^S S, \sigma_G)$$

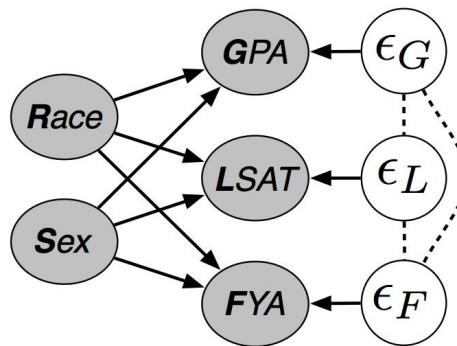
Gaussian Dist. Parameters

$$\text{LSAT} \sim \text{Poisson}(\exp(b_L + w_L^K K + w_L^R R + w_L^S S))$$

[Kusner et al, 2018](#)

Level 3 Counterfactual Fairness

- Build Predictors by adding independent error terms



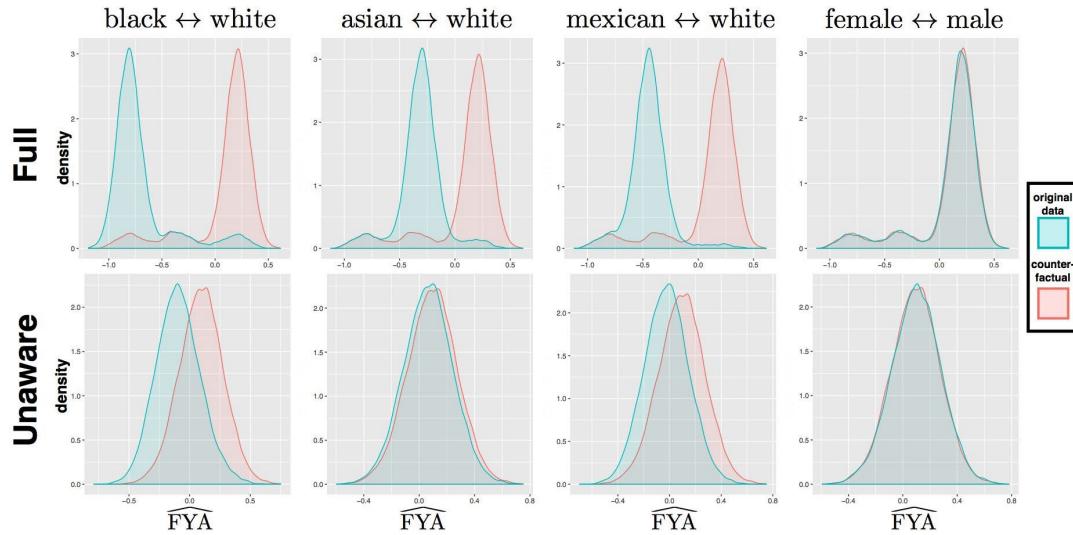
$$\text{GPA} = b_G + w_G^R R + w_G^S S + \epsilon_G, \quad \epsilon_G \sim p(\epsilon_G)$$

$$\text{LSAT} = b_L + w_L^R R + w_L^S S + \epsilon_L, \quad \epsilon_L \sim p(\epsilon_L)$$

$$\text{FYA} = b_F + w_F^R R + w_F^S S + \epsilon_F, \quad \epsilon_F \sim p(\epsilon_F)$$

[Kusner et al, 2018](#)

Baselines



full - using all features

unaware - fairness through unawareness

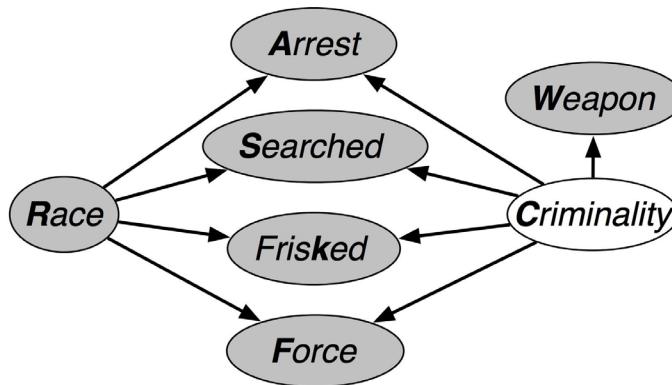
Kusner et al, 2018

Results

	Baseline	Baseline	Level 2	Level 3
	Full	Unaware	Fair K	Fair Add
RMSE	0.873	0.894	0.929	0.918

Causal Graph

- Assess the fairness of the NYC arrest dataset
 - 38,609 records
 - White individuals (4492)
 - Black Hispanic individuals (2414)



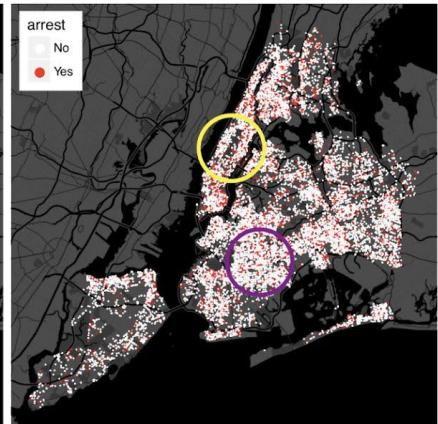
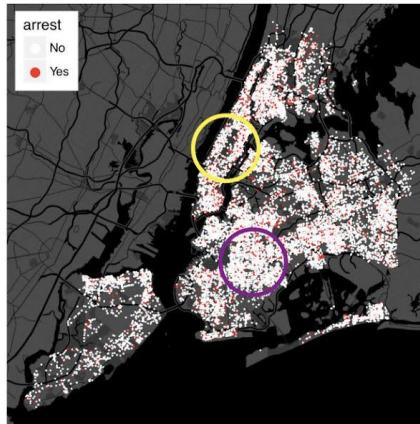
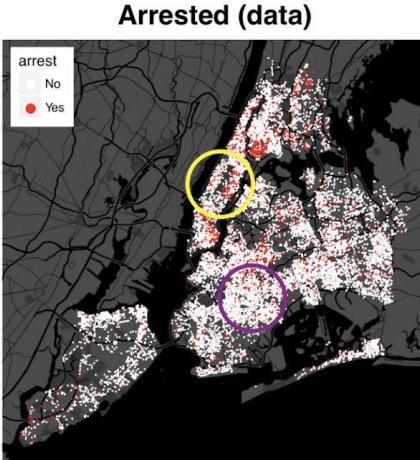
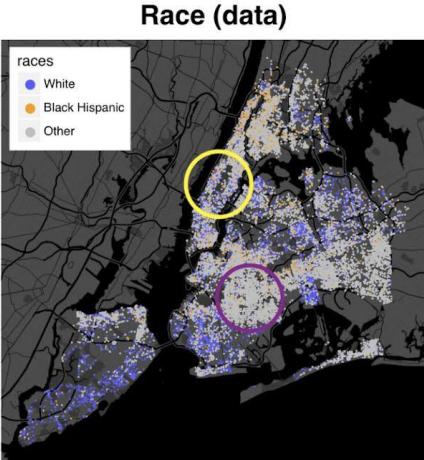
Assessment Results

White (4492)
Black Hispanic (2414)

White (12.1%)
Black Hispanic (19.8%)

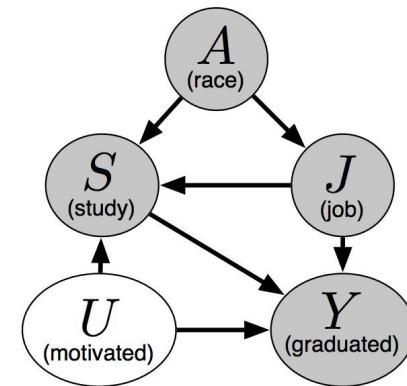
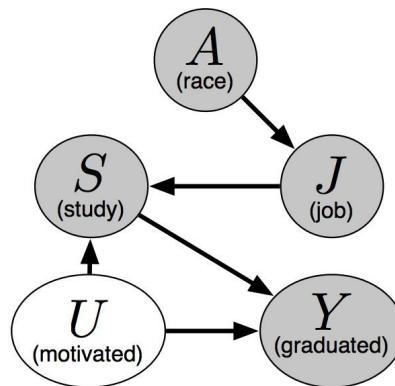
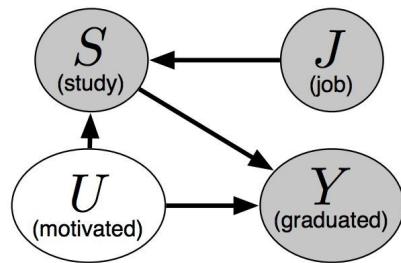
Arrests decreases from
5659 to 3722
**Arrest if White
(counterfactual)**

Arrests increases from
5659 to 6439
**Arrest if Black Hispanic
(counterfactual)**



Multiple Causal Graphs

- Whether a student can graduate on time



Alternative Definitions of Counterfactual Fairness

Exact Formulation

$$P(\hat{Y}_{A \leftarrow a}(U) = y \mid X = x, A = a) = P(\hat{Y}_{A \leftarrow a'}(U) = y \mid X = x, A = a)$$

ϵ - Approximate Formulation

$$\left| f(\mathbf{x}_{A \leftarrow a}, a) - f(\mathbf{x}_{A \leftarrow a'}, a') \right| \leq \epsilon$$

(δ, ϵ) - Approximate Formulation

$$\mathbb{P}(\left| f(\mathcal{X}_{A \leftarrow a}, a) - f(\mathcal{X}_{A \leftarrow a'}, a') \right| \leq \epsilon \mid \mathcal{X} = \mathbf{x}, A = a) \geq 1 - \delta$$

Multi-world Counterfactual Fairness

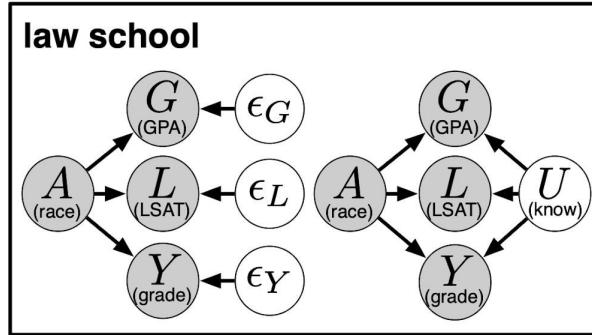
$$\min_f \frac{1}{n} \sum_{i=1}^n \ell(f(\mathbf{x}_i, a_i), y_i) + \lambda \sum_{j=1}^m \frac{1}{n} \sum_{i=1}^n \sum_{a' \neq a_i} \mu_j(f, \mathbf{x}_i, a_i, a')$$

loss of the data
world j counterfactual examples

$$\mu_j(f, \mathbf{x}_i, a_i, a') := \frac{1}{S} \sum_{s=1}^S \max\{0, |f(\mathbf{x}_{i, A^j \leftarrow a_i}^s, a_i) - f(\mathbf{x}_{i, A^j \leftarrow a'}^s, a')| - \epsilon\}$$

Monte-carlo Samples ϵ - Approximate Counterfactual Fairness

Law Graduate School



$$G = b_G + w_G^A A + \epsilon_G$$

$$L = b_L + w_L^A A + \epsilon_L$$

$$Y = b_Y + w_Y^A A + \epsilon_Y$$

$$\epsilon_G, \epsilon_L, \epsilon_Y \sim \mathcal{N}(0, 1)$$

L3 Method

$$G \sim \mathcal{N}(b_G + w_G^A A + w_G^U U, \sigma_G)$$

$$L \sim \text{Poisson}(\exp(b_L + w_L^A A + w_L^U U))$$

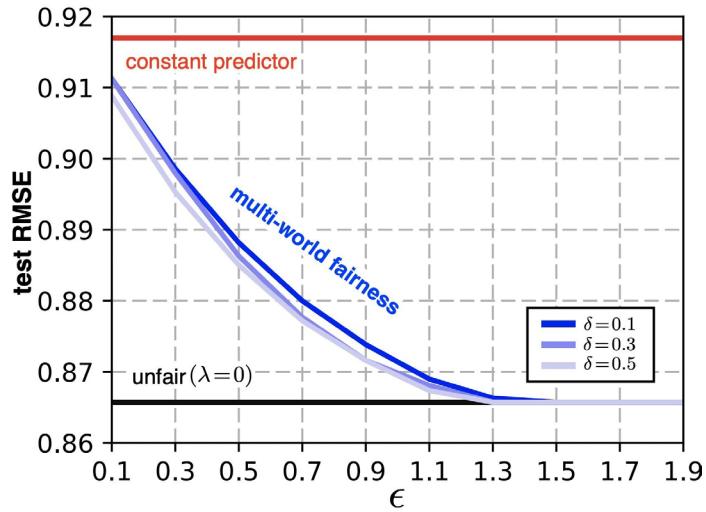
$$Y \sim \mathcal{N}(w_Y^A A + w_Y^U U, 1)$$

$$U \sim \mathcal{N}(0, 1)$$

[Russell et al, 2017](#)

L2 Method

Results



$$|f(\mathbf{x}_{A \leftarrow a}, a) - f(\mathbf{x}_{A \leftarrow a'}, a')| \leq \epsilon$$

[Russell et al, 2017](#)

Equalized Counterfactual Odds

Equality of Odds

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

Counterfactual Fairness

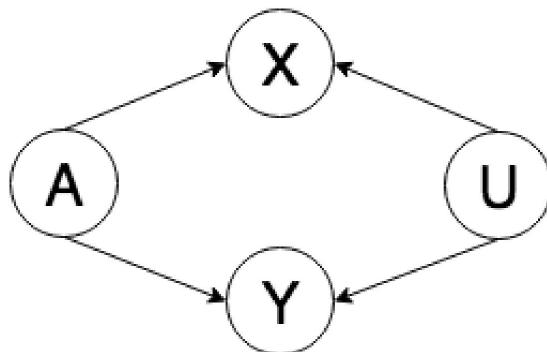
$$\underline{P(\hat{Y}_{A \leftarrow a}(U) = y | X = x, A = a)} = \underline{P(\hat{Y}_{A \leftarrow a'}(U) = y | X = x, A = a)}$$

Equalized Counterfactual Odds

$$\underline{p(\hat{Y}_{A \leftarrow a}(U) | X = x, Y_{A \leftarrow a} = y, A = a)} = \underline{p(\hat{Y}_{A \leftarrow a'}(U) | X = x, Y_{A \leftarrow a'} = y, A = a)}$$

Healthcare Equality

- Protected Features A = {Gender}
- Features X, vector representation of coded diagnoses, procedures, medication orders, lab results, and clinical notes
- Prediction Y, a binary indicator of the occurrence of a clinically relevant outcome



$$u \sim p(U) = \text{Normal}(0, I)$$

$$a \sim p(A) = \text{Categorical}(A | \pi)$$

$$x, y \sim p(X, Y | U, A) = p(X | U, A)p(Y | U, A)$$

Training Objective

- σ - sigmoid function
- h - predictor
- J - cross entropy loss

$$\begin{aligned}\mathcal{L} = & J(h_\theta(x, a), y) + \lambda_{\text{CF}} \sum_{a_k \in \mathcal{A}} \mathbb{1}[a \neq a_k] J(h_\theta(\underline{x}_{A \leftarrow a_k}, a_k), \underline{y}_{A \leftarrow a_k}) + \\ & \lambda_{\text{CLP}} \sum_{a_k \in \mathcal{A}} \mathbb{1}[a \neq a_k] \mathbb{1}[y = \underline{y}_{A \leftarrow a_k}] \frac{\left(\sigma^{-1}(h_\theta(\underline{x}_{A \leftarrow a_k}, a_k)) - \sigma^{-1}(h_\theta(x, a)) \right)^2}{\text{logits}}\end{aligned}$$

Dataset Overview

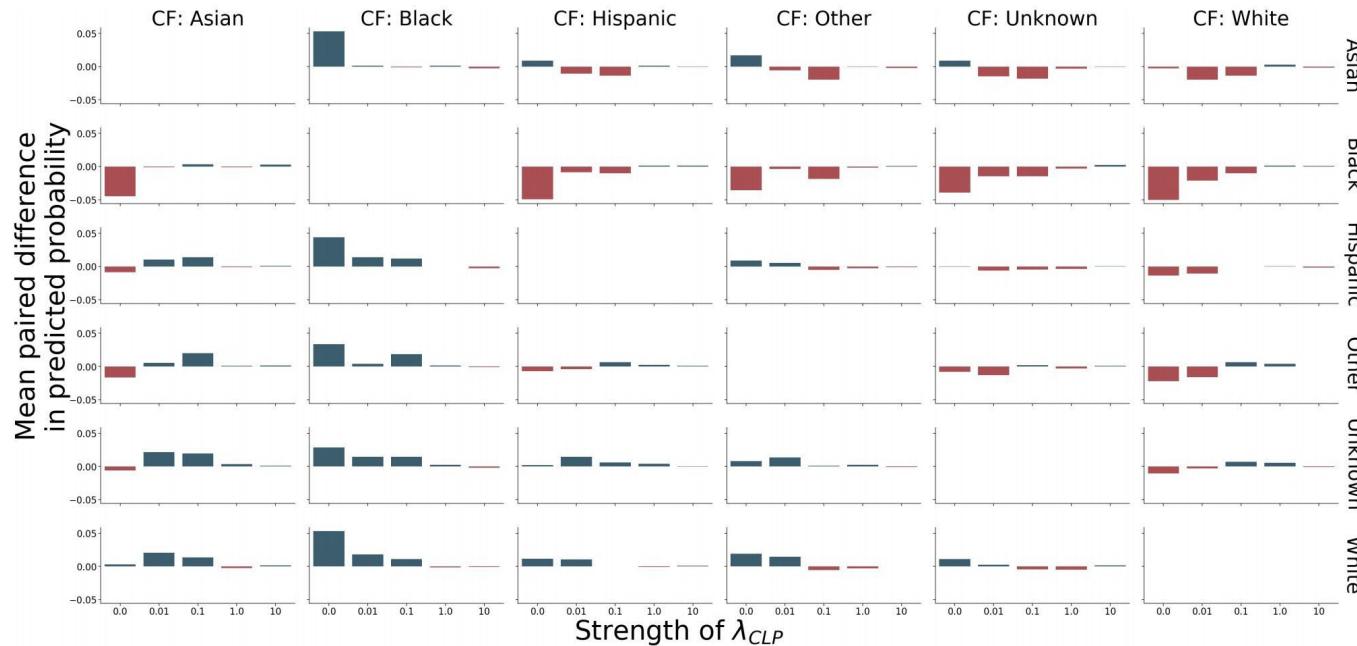
Group	Count	Length of Stay \geq 7 Days	Inpatient Mortality
Asian	17,465	0.187	0.025
Black	5,202	0.239	0.020
Hispanic	21,978	0.196	0.019
Other	11,004	0.200	0.022
Unknown	3,593	0.201	0.072
White	70,391	0.204	0.021
Female	72,556	0.167	0.018
Male	57,076	0.245	0.029
[18, 30)	15,291	0.180	0.007
[30, 45)	27,155	0.140	0.007
[45, 65)	43,529	0.222	0.025
[65, 89)	43,658	0.226	0.036
All	129,633	0.201	0.023

Results

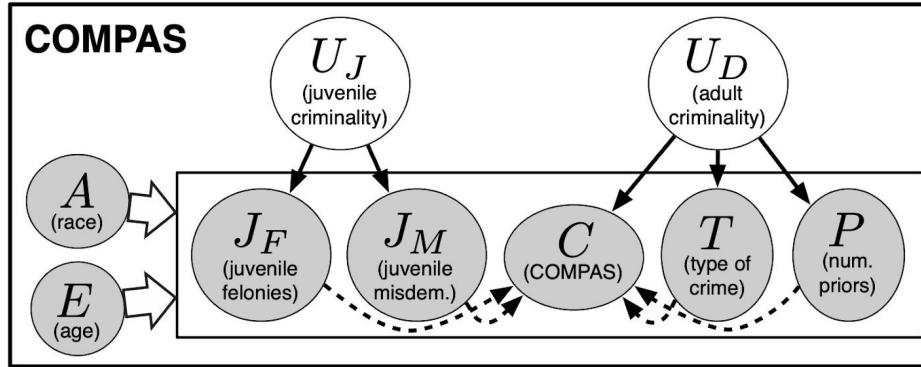
Group	Metric	λ_{CLP}					
		N/A	0.0	0.01	0.1	1.0	10.0
Asian	AUC-PRC	0.605	0.563	0.555	0.561	0.56	0.562
	AUC-ROC	0.86	0.853	0.853	0.854	0.849	0.851
	Brier	0.106	0.11	0.109	0.109	0.11	0.112
Black	AUC-PRC	0.579	0.548	0.55	0.545	0.563	0.573
	AUC-ROC	0.838	0.825	0.82	0.825	0.823	0.823
	Brier	0.124	0.135	0.129	0.128	0.127	0.129
Hispanic	AUC-PRC	0.592	0.558	0.565	0.57	0.564	0.56
	AUC-ROC	0.862	0.855	0.856	0.861	0.853	0.854
	Brier	0.113	0.117	0.115	0.114	0.117	0.118
Other	AUC-PRC	0.549	0.557	0.557	0.563	0.553	0.561
	AUC-ROC	0.824	0.827	0.819	0.824	0.819	0.827
	Brier	0.122	0.124	0.121	0.121	0.122	0.124
Unknown	AUC-PRC	0.675	0.616	0.616	0.606	0.614	0.633
	AUC-ROC	0.9	0.891	0.888	0.893	0.891	0.887
	Brier	0.104	0.106	0.103	0.103	0.105	0.111
White	AUC-PRC	0.575	0.568	0.564	0.559	0.562	0.563
	AUC-ROC	0.847	0.84	0.839	0.838	0.838	0.837
	Brier	0.118	0.12	0.118	0.12	0.12	0.121

Results

- Difference in the counterfactual versus factual predicted probability



COMPAS



$$T \sim \text{Bernoulli}(\phi(b_T + w_C^{U_P} U_D + w_C^E E + w_C^A A)) \quad (1)$$

$$C \sim \mathcal{N}(b_C + w_C^{U_D} U_D + w_C^E E + w_C^A A + w_C^T T + w_C^P P + w_C^{J_F} J_F + w_C^{J_M} J_M, \sigma_C)$$

$$P \sim \text{Poisson}(\exp(b_P + w_P^{U_D} U_D + w_P^E E + w_P^A A))$$

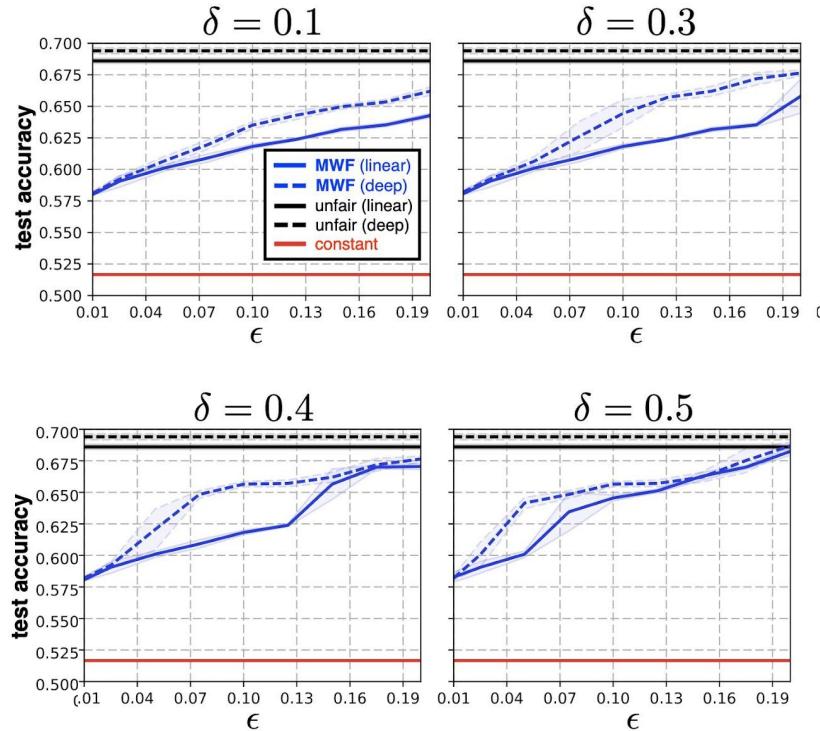
$$J_F \sim \text{Poisson}(\exp(b_{J_F} + w_{J_F}^{U_J} + w_{J_F}^E E + w_{J_F}^A A))$$

$$J_M \sim \text{Poisson}(\exp(b_{J_M} + w_{J_M}^{U_J} + w_{J_M}^E E + w_{J_M}^A A))$$

$$[U_J, U_D] \sim \mathcal{N}(0, \Sigma)$$

[Russell et al, 2017](#)

Results



[Russell et al, 2017](#)

Welcome !!



BITS Pilani
Pilani Campus

Fair, Accountable, Transparent Machine Learning (FAccT ML) ZG517

Dr. Sugata Ghosal
sugata.ghosal@pilani.bits-pilani.ac.in



**Session 7
Date – 2nd July 2023
Time – 8:45 AM to 10:45 AM**

These slides are prepared by the instructor, with grateful acknowledgement of many others who made their course materials freely available online.

Agenda

- Fair NLP
 - Biases in NLP Models
 - Data Augmentation
 - Debiasing Word Embedding
 - Adversarial Learning
 - Fair Visual Representation

Biases of NLP Models

- Denigration
 - The use of culturally or historically derogatory terms
- Under-representation
 - The disproportionately low representation of a specific group
 - e.g., a classifier's performance is adversely affected due to sampling biases of the minority protected group
- Stereotyping
 - An over-generalized belief about a particular category of people
 - e.g., a classifier attributes man to computers more than woman
- Recognition
 - Algorithms perform different for protected groups because of their inherent characteristics
 - e.g., a voice recognition algorithm has better capabilities in recognizing voices in low frequency

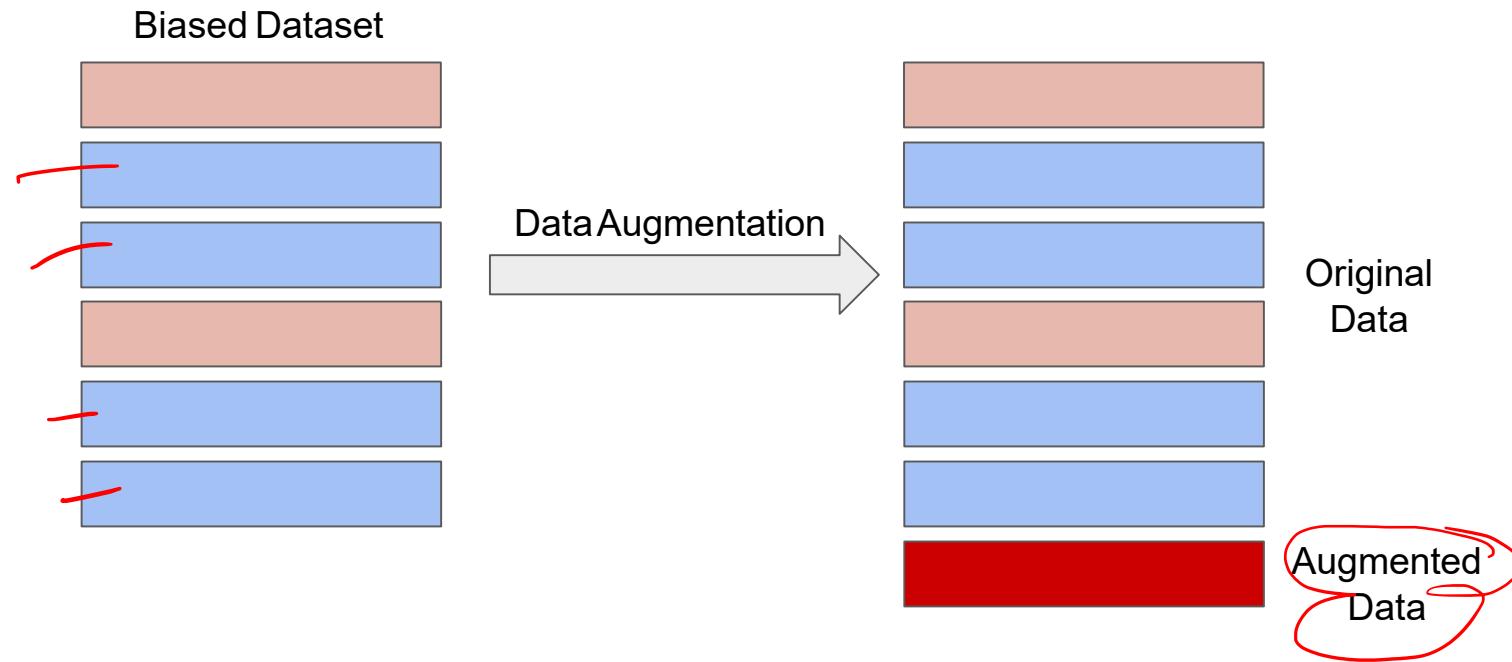
Biases of NLP Models

Task	Example of Representation Bias in the Context of Gender	
Machine Translation	Translating “He is a nurse. She is a doctor.” to Hungarian and back to English results in “She is a nurse. He is a doctor.” (Douglas, 2017)	✓
Caption Generation	An image captioning model incorrectly predicts the agent to be male because there is a computer nearby (Burns et al., 2018).	✓
Speech Recognition	Automatic speech detection works better with male voices than female voices (Tatman, 2017).	✗
Sentiment Analysis	Sentiment Analysis Systems rank sentences containing female noun phrases to be indicative of anger more often than sentences containing male noun phrases (Park et al., 2018).	✓
Language Model	“He is doctor” has a higher conditional likelihood than “She is doctor” (Lu et al., 2018).	✓
Word Embedding	Analogies such as “man : woman :: computer programmer : homemaker” are automatically generated by models trained on biased word embeddings (Bolukbasi et al., 2016).	✓

(S)tereotyping, (D)enigration, (R)ecognition, (U)nder-representation

Sun et al, 2019

Data Augmentation



Coreference Resolution

A man and his son get into a terrible car crash. The father dies, and the boy is badly injured. In the hospital, the surgeon looks at the patient and exclaims, “I can’t operate on this boy, he’s my son!

Does this paragraph make sense to you?

[Rudinger et al, 2018](#)

Gender Swapping in Coreference Resolution

Original sample



Mention coref Mention coref Mention coref Mention
The surgeon could n't operate on his patient : it was his son !

Gender swap



Mention coref Mention coref Mention coref
The surgeon could n't operate on their patient : it was their son !

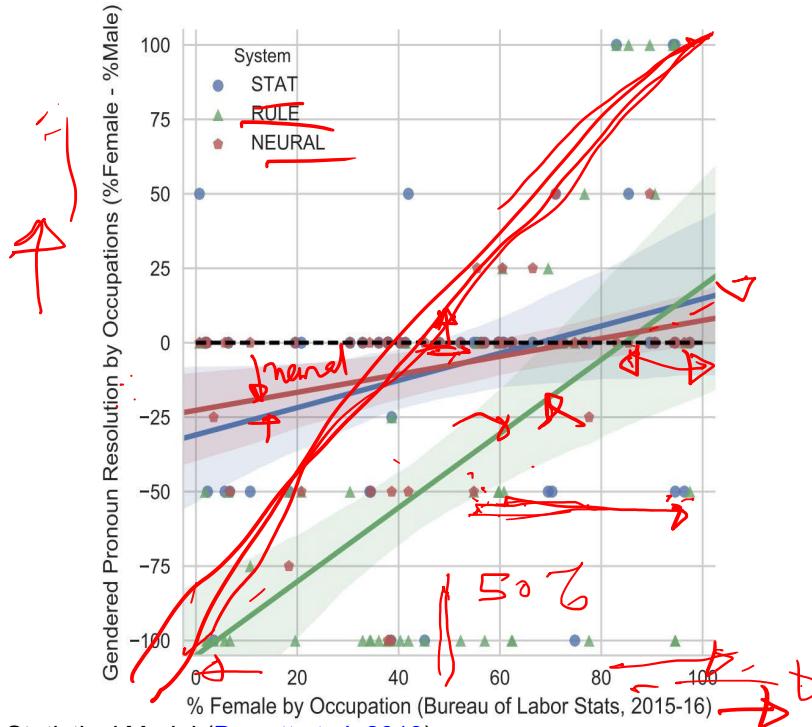
Gender swap



Mention coref Mention coref Mention coref
The surgeon could n't operate on her patient : it was her son !

[Rudinger et al, 2018](#)

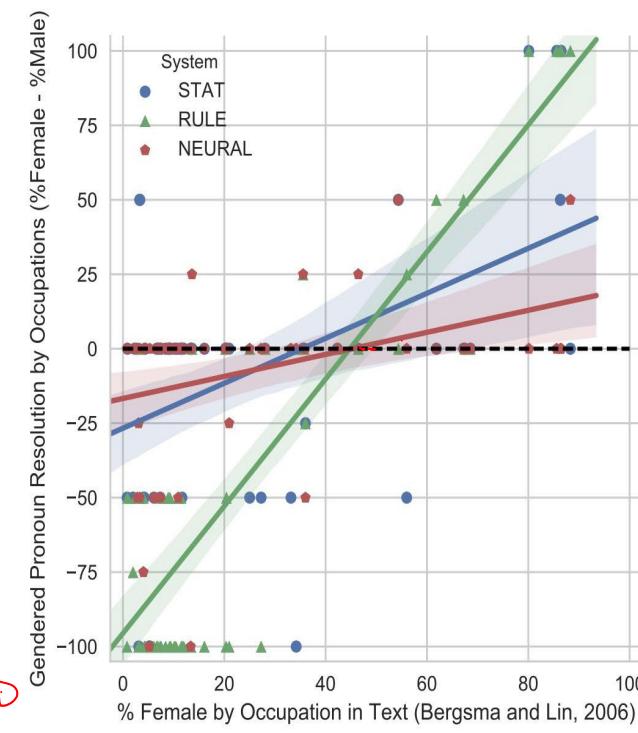
Results



STAT - Statistical Model ([Durrett et al, 2013](#))

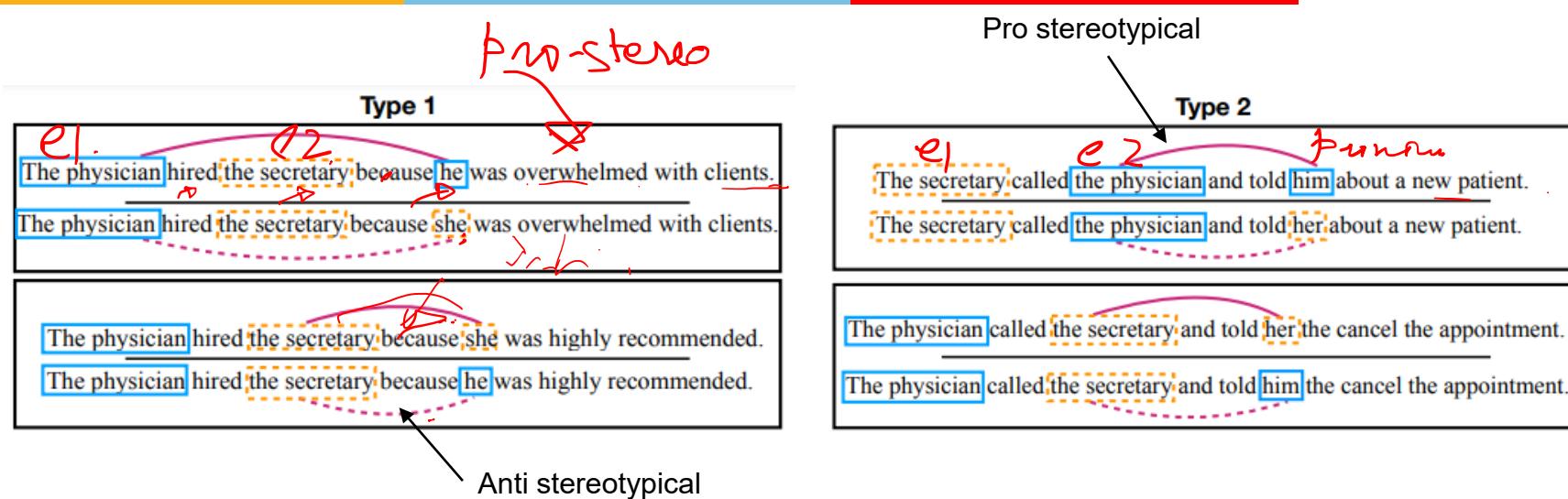
RULE - Rule Based Model ([Lee et al, 2011](#))

NEURAL - Neural Based Model ([Clark et al, 2016](#))



[Rudinger et al, 2018](#)

Gender Balanced Co-occurrence Test



Type 1: [entity1] [interacts with] [entity2] [conjunction] [pronoun] [circumstances]

Type 2: [entity1] [interacts with] [entity2] and then [interacts with] [pronoun] for [circumstances].

Results

Method	Anon.	Resour.	Aug.	OntoNotes	T1-p	T1-a	Avg	Diff	T2-p	T2-a	Avg	Diff
E2E	✓	✓		66.5	67.2	59.3	63.2	7.9*	81.4	82.3	81.9	0.9
E2E	✓	✓	✓	66.3	63.9	62.8	63.4	1.1	81.3	83.4	82.4	2.1
Feature	✓	✓		61.2	61.8	62.0	61.9	0.2	67.1	63.5	65.3	3.6
Feature	✓	✓	✓	61.0	62.3	60.4	61.4	1.9*	71.1	68.6	69.9	2.5

E2E ([Lee et al, 2011](#))

Feature ([Durrett et al, 2013](#))

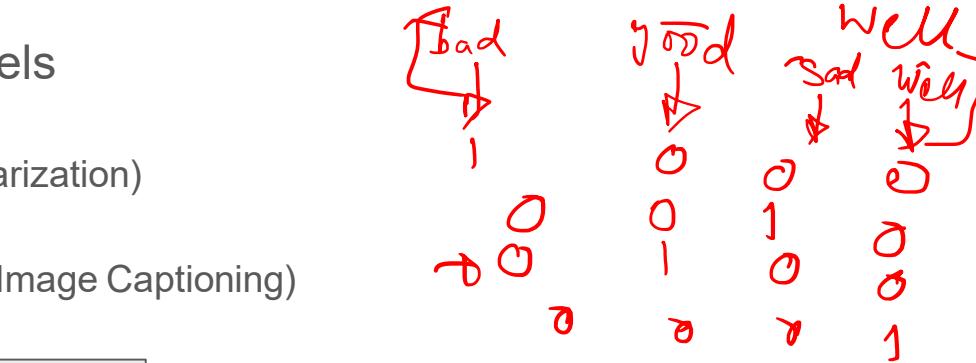
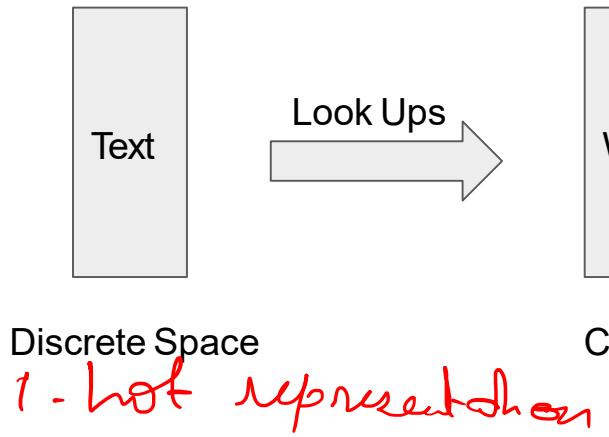
Diff - Difference between pro/anti

[Zhao et al, 2018](#)

Word Embeddings



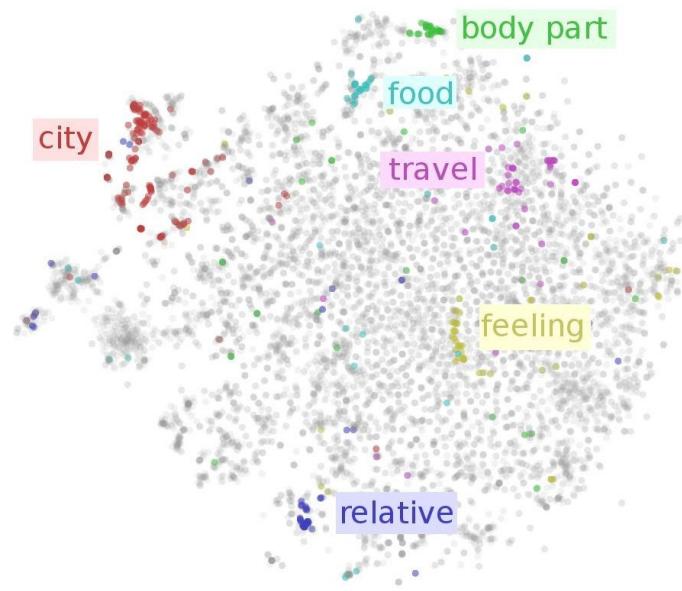
- An Essential Part of Deep NLP Models
 - Classifications (e.g., Sentiment Analysis)
 - Text Generation (e.g., translation, summarization)
 - Text Retrieval (e.g., Question Answering)
 - Visual-Language Representations (e.g., Image Captioning)



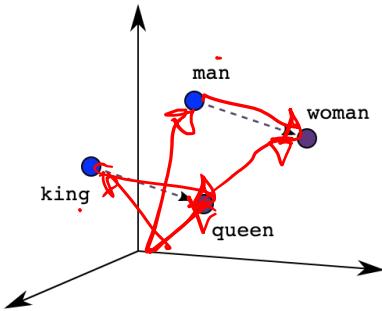
Neural Networks

Word Embeddings

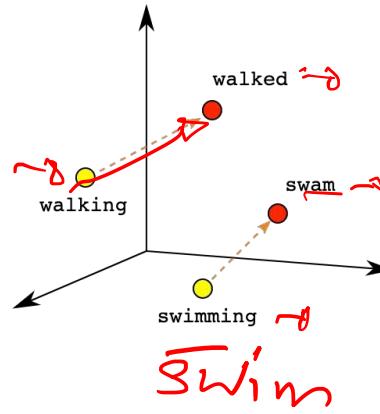
- Embedding Techniques
 - GloVe ([Pennington et al, 2014](#))
 - Word2Vec ([Rong et al, 2014](#))
- Trained Through A Proxy Task
 - Word proximity (GloVe)
 - SkipGram (Word2Vec)



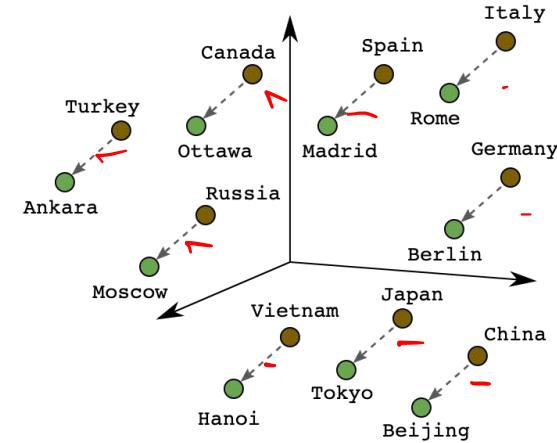
Geometric Properties of Word Embeddings



Male-Female

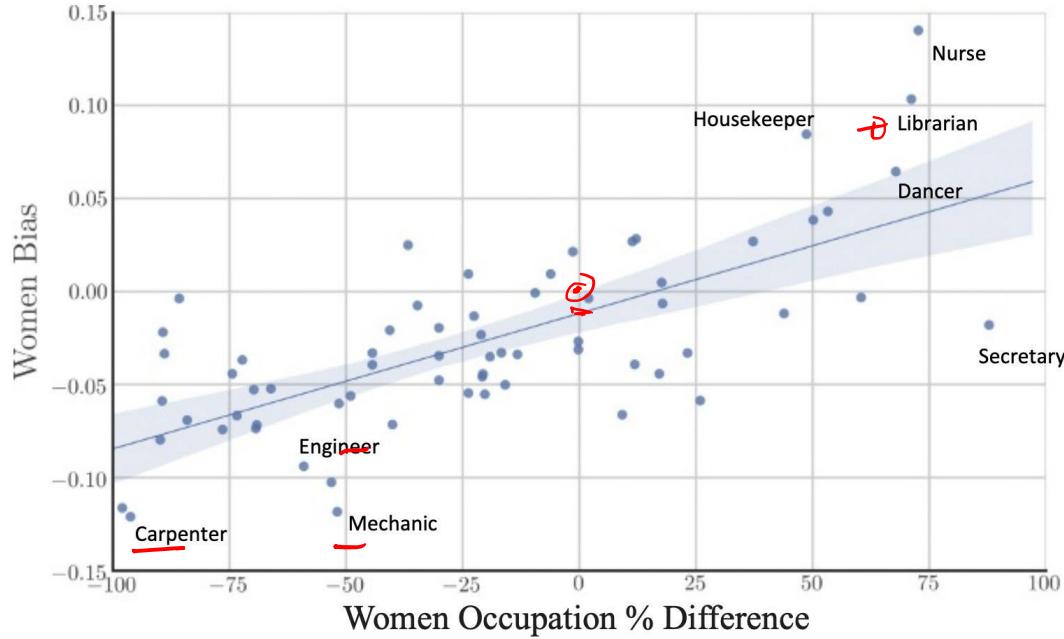


Verb Tense



Country-Capital

Can Word Embedding Be Biased?



Garga et al, 2017

Types of Gender Associations

- Definitional Gender Associations

$$\overrightarrow{\text{man}} - \overrightarrow{\text{woman}} \approx \overrightarrow{\text{king}} - \overrightarrow{\text{queen}}$$

- Stereotypical Gender Associations

$$\overrightarrow{\text{man}} - \overrightarrow{\text{woman}} \approx \overrightarrow{\text{computer programmer}} - \overrightarrow{\text{homemaker}}$$

[Bolukbasi et al, 2016](#)

Definitional and Stereotypical Associations



Gender stereotype *she-he* analogies.

sewing-carpentry	register-nurse-physician	housewife-shopkeeper
nurse-surgeon	interior designer-architect	softball-baseball
blond-burly	feminism-conservatism	cosmetics-pharmaceuticals
giggle-chuckle	vocalist-guitarist	petite-lanky
sassy-snappy	diva-superstar	charming-affable
volleyball-football	cupcakes-pizzas	hairdresser-barber

Gender appropriate *she-he* analogies.

queen-king	sister-brother	mother-father
waitress-waiter	ovarian cancer-prostate cancer	convent-monastery

[Bolukbasi et al, 2016](#)

Gender Subspace

$$\overrightarrow{\text{grandmother}} - \overrightarrow{\text{grandfather}} = \overrightarrow{\text{gal}} - \overrightarrow{\text{guy}} = g$$

$$\begin{array}{c} \overrightarrow{\text{she}} - \overrightarrow{\text{he}} \\ \overrightarrow{\text{her}} - \overrightarrow{\text{his}} \\ \overrightarrow{\text{woman}} - \overrightarrow{\text{man}} \\ \overrightarrow{\text{Mary}} - \overrightarrow{\text{John}} \\ \overrightarrow{\text{herself}} - \overrightarrow{\text{himself}} \end{array}$$

$$\begin{array}{c} \overrightarrow{\text{daughter}} - \overrightarrow{\text{son}} \\ \overrightarrow{\text{mother}} - \overrightarrow{\text{father}} \\ \overrightarrow{\text{gal}} - \overrightarrow{\text{guy}} \\ \overrightarrow{\text{girl}} - \overrightarrow{\text{boy}} \\ \overrightarrow{\text{female}} - \overrightarrow{\text{male}} \end{array}$$

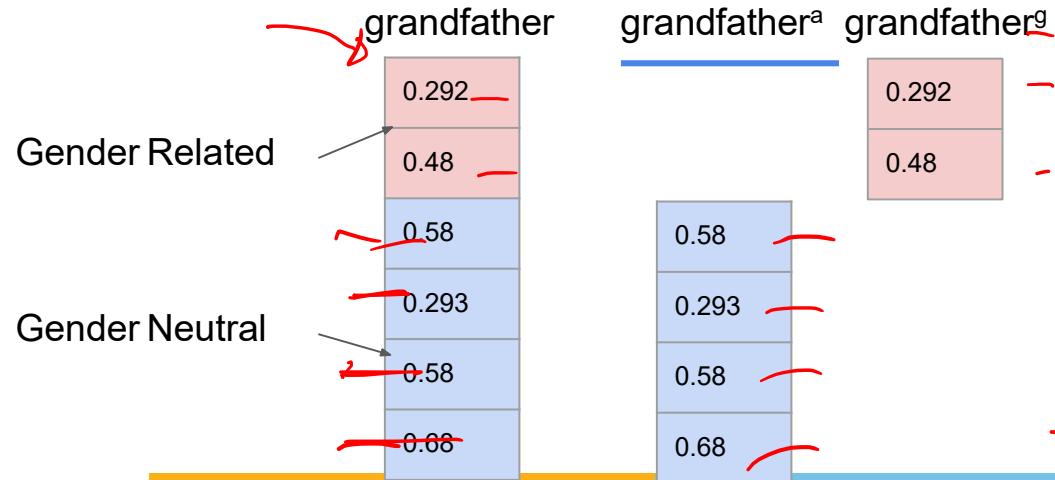
[Bolukbasi et al, 2016](#)

Gender-Neutral Word Embeddings

- Decompose Word Embeddings Into Gender-Related and Gender-Neutral Parts

- agwzh'c

$$w = [w^{(a)}; w^{(g)}]$$



→ Zhao et al, 2018

Gender-Neutral Word Embeddings

- Fine-tuning Word Embeddings Using Debiasing Regularizers

$$J = \underline{J_G} + \lambda_d \underline{J_D} + \lambda_e \underline{J_E}$$

Glove
Loss Function
Regulate
Gender-related
Words
Regulate All Other
Words

Ω_F
 Female Seed Words

Ω_N
 All Other Words

Ω_M
 Male Seed Words

Zhao et al, 2018

Gender-Neutral Word Embeddings

- Fine-tuning Word Embeddings Using Debiasing Regularizers

$$J = J_G + \lambda_d \underline{J_D} + \lambda_e J_E$$

Ω_M Ω_F

Regulate
Gender-related
Words

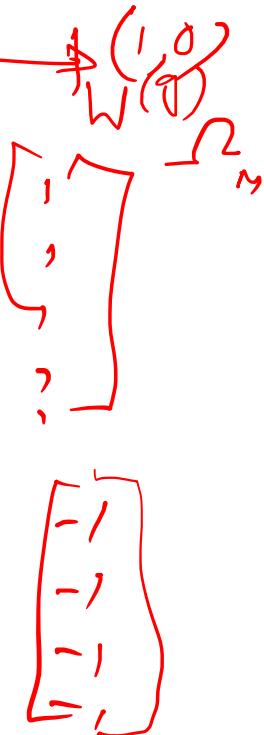
Ω_F
Female Seed Word

Push Toward Extremes
On Gender Dimensions

Ω_M
Male Seed Word

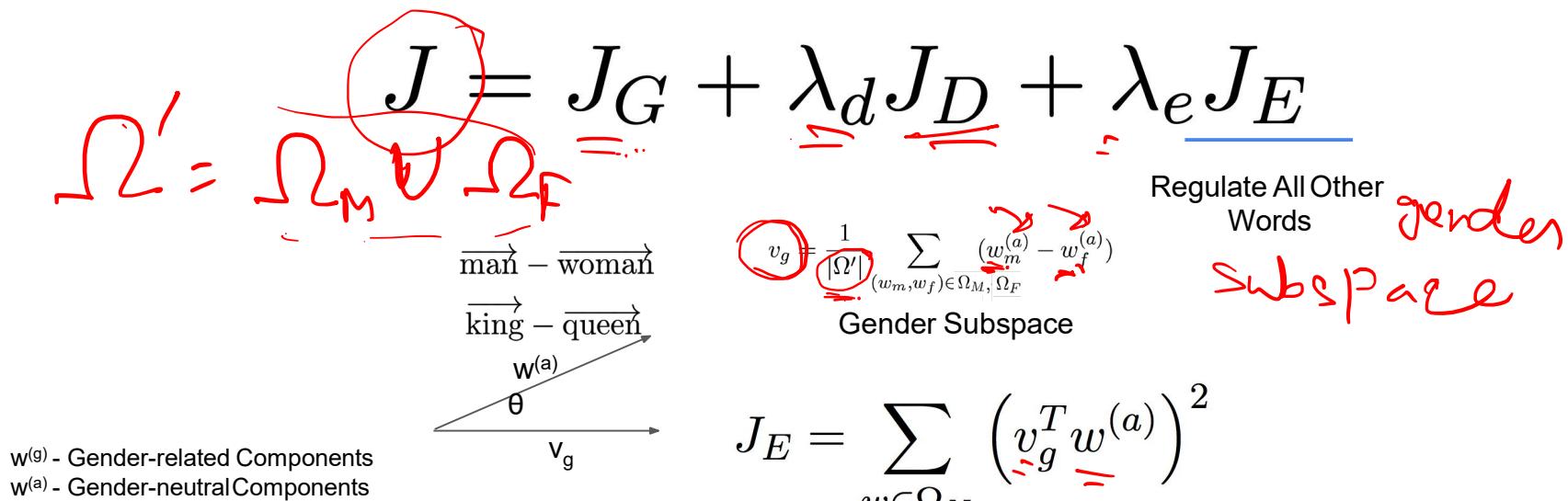
$w^{(g)}$ - Gender-related Components
 $w^{(a)}$ - Gender-neutral Components

$$J_D = \left\| \sum_{w \in \Omega_M} w^{(g)} - \sum_{w \in \Omega_F} w^{(g)} \right\|_1 + \sum_{w \in \Omega_M} \left\| 1 - w^{(g)} \right\|_2^2 + \sum_{w \in \Omega_F} \left\| -1 - w^{(g)} \right\|_2^2$$



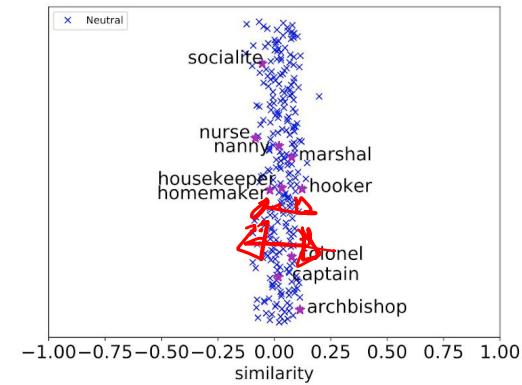
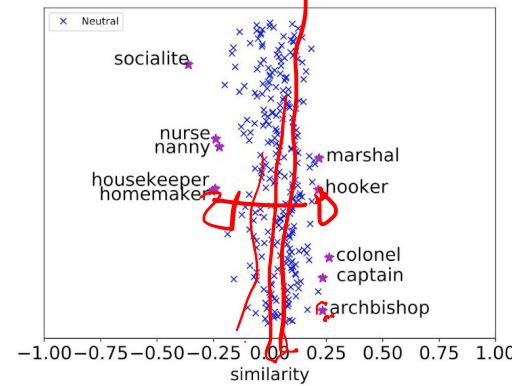
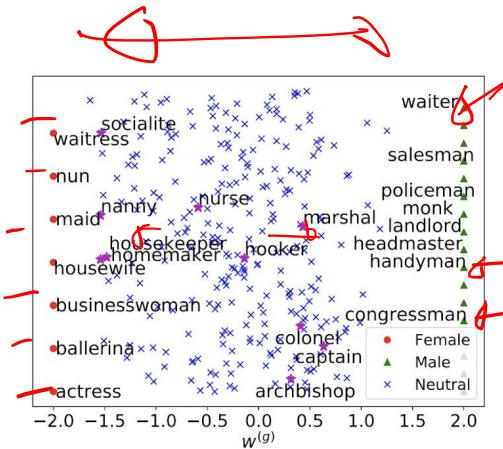
Gender-Neutral Word Embeddings

- Fine-tuning Word Embeddings Using Debiasing Regularizers



Zhao et al, 2018

Gender Attribute Separation



$w^{(g)}$ - Gender-related Components
 $w^{(a)}$ - Gender-neutral Components

Gender Relational Analogy

Question 1: Consider the following word pairs: pilgrim:shrine, hunter:quarry, assassin:victim, climber:peak. What relation best describes these $X:Y$ word pairs?

- (1) “ X worships/reveres Y ”
- (2) “ X seeks/desires/aims for Y ”
- (3) “ X harms/destroys Y ”
- (4) “ X uses/exploits/employs Y ”

Dataset	Embeddings	Definition	Stereotype	None
SemBias	GloVe	80.2	10.9	8.9
	GN-GloVe	97.7	1.4	0.9
SemBias (subset)	GloVe	57.5	20	22.5
	GN-GloVe	75	15	10

[Jurgens et al , 2012](#)

Coreference Resolution

Embeddings	OntoNotes-test	PRO	ANTI	Avg	Diff
GloVe	66.5	76.2	46.0	61.1	30.2
GN-GloVe	66.2	72.4	51.9	62.2	20.5
GN-GloVe(w_a)	65.9	70.0	53.9	62.0	16.1

w^(a) - Gender-neutral Components

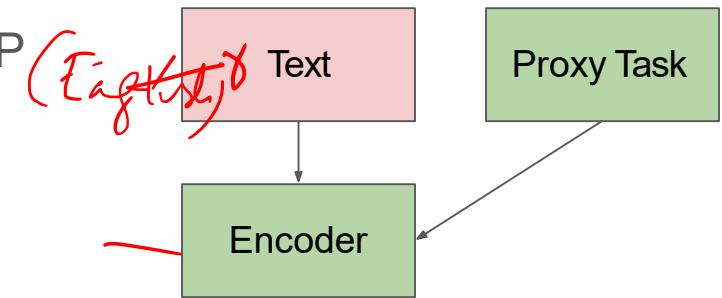
[Jurgens et al , 2012](#)

The Use of Pre-trained NLP Encoders



- Pre-trained Encoders Are Widely Used in NLP

- Transfer information from a related domain
 - Boost performance on a small data set
 - Trained through a proxy task



- Pre-trained NLP Encoders

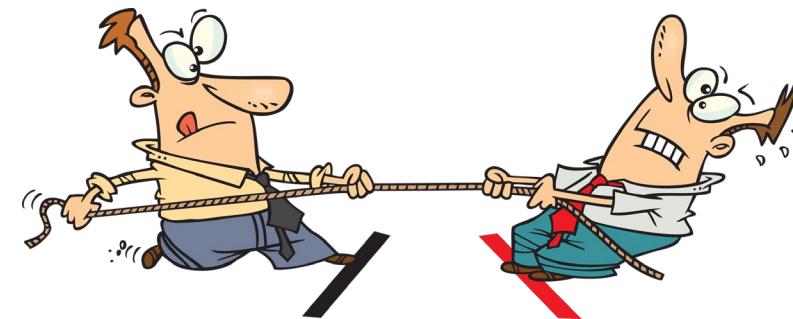
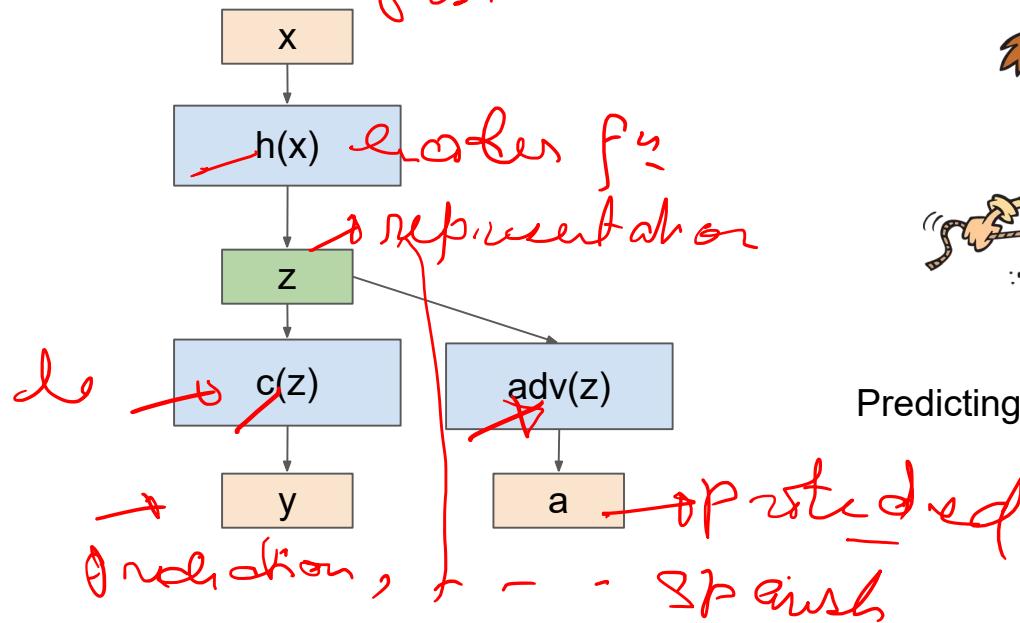
- ELMO ([Peters et al 2018](#))
 - BERT ([Devlin et al, 2018](#))
 - XLNet ([Yang et al, 2019](#))

- Can Pre-trained Encoders Be Biased?

Word
Sef. in →
doshindone (spanish)

Adversarial Learning

Words in English



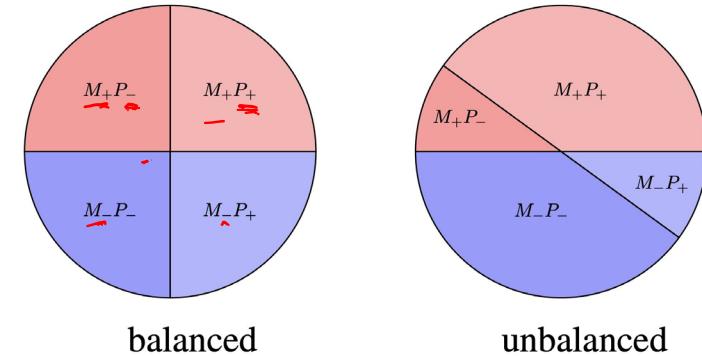
Predicting y

Reconstruct a

[Elazar et al, 2018](#)

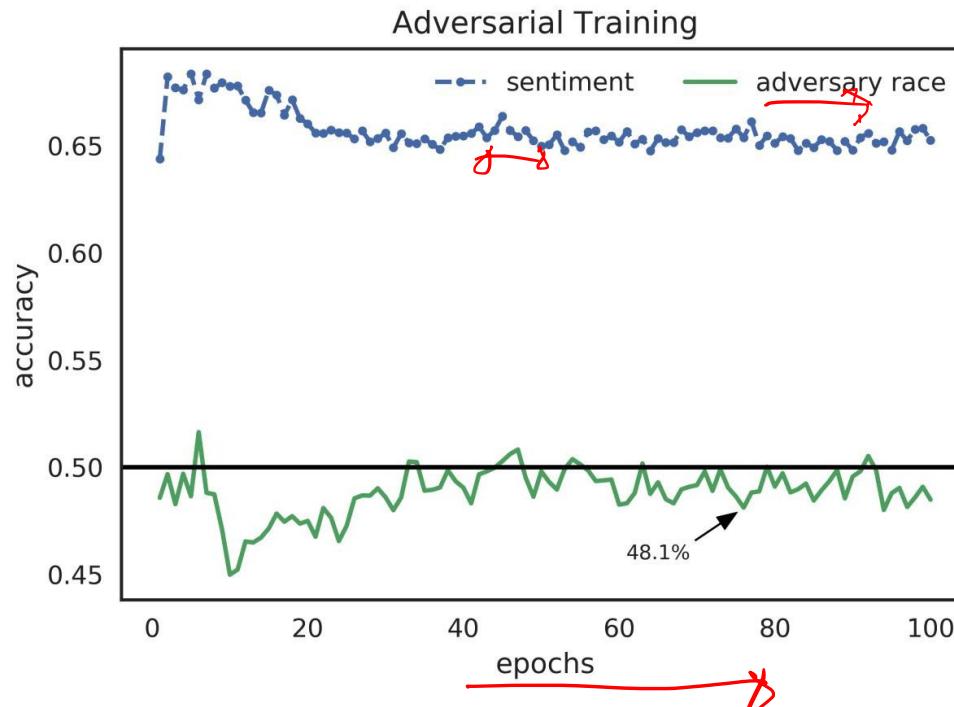
Twitter Prediction Problem

- Twitter Sentiment & Mention Detection
- Protected Attributes
 - Race
 - Gender
 - Age
- Leakage
 - Predict protected attributes



Data	Task	Protected Attribute	Balanced		Unbalanced	
			Task Acc	Leakage	Task Acc	Leakage
DIAL	Sentiment	Race	67.4	64.5	79.5	73.5
	Mention	Race	81.2	71.5	86.0	73.8
PAN16	Mention	Gender	77.5	60.1	76.8	64.0
		Age	74.7	59.4	77.5	59.7

Main Task and Adversary Accuracies



[Elazar et al, 2018](#)

Beefing Up the Adversary

- Increase the Capacity of the Adversary
 - Model Capacity
 - Weight on Loss
 - Ensemble

Method	Parameter	DIAL			PAN16			Δ		
		Sentiment	Race	Δ	Mention	Gender	Mention			
No Adversary Baseline	-	67.4	14.5	-	77.5	10.1	-	74.7	9.4	-
Standard Adversary	(300/1.0/1)	64.7	6.0	5.0	75.6	8.5	8.0	72.5	7.3	6.9
<u>Adv-Capacity</u>	500	64.1	6.7	5.2	73.8	8.1	6.7	71.4	4.3	4.1
	1000	63.4	7.1	4.9	75.2	8.9	7.0	71.6	6.3	4.0
	2000	65.2	8.1	6.9	76.1	6.7	6.4	71.9	6.0	5.7
	5000	63.9	6.2	3.7	74.5	5.6	1.6	73.0	10.2	9.6
	8000	65.0	7.1	4.8	75.7	5.4	4.2	71.9	9.8	7.3
	λ	63.9	6.8	6.2	75.6	7.8	6.8	73.1	4.8	3.4
λ	0.5	64.9	7.4	5.4	75.6	4.9	2.4	72.5	6.8	5.8
	1.5	64.2	7.3	5.9	76.0	7.2	6.7	72.1	8.5	7.7
	2.0	65.8	10.2	10.1	73.7	6.4	6.1	72.5	-6.3	5.2
	3.0	50.0	-	-	73.6	6.5	5.7	69.0	3.2	2.9
	5.0	-	-	-	-	-	-	-	-	-
<u>Ensemble</u>	2	62.4	7.4	5.4	74.8	6.4	5.0	72.8	8.8	8.3
	3	66.5	6.5	5.0	75.3	4.9	3.1	72.1	6.7	6.0
	5	63.8	4.8	2.6	74.3	4.1	3.0	70.1	5.7	5.4

Δ - the difference between the attacker score and the corresponding adversary's accuracy

Elazar et al, 2018

Fair Visual Representation

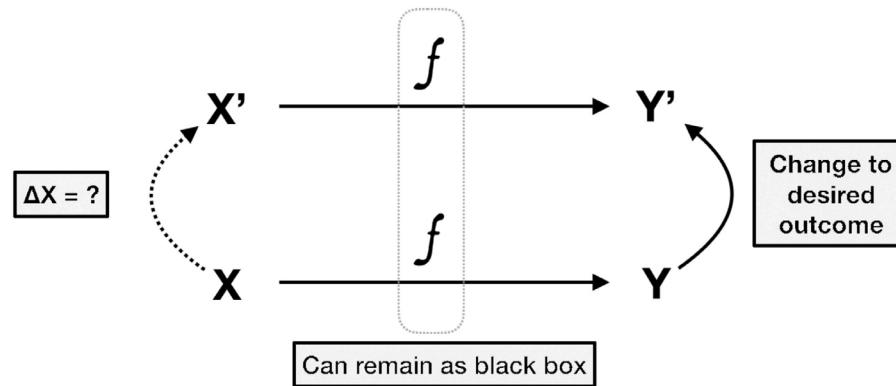
- Counterfactual Fairness
- Counterfactual Face Attribution
- Gender Equalized Image Captioning
- Adversarial Removal of Gender Features

Counterfactual Explanation

$$\underline{x'} = \arg \min_{x'} \lambda (\hat{f}(x') - y')^2 + \frac{d(x, x')}{\text{distance function}}$$

counterfactual example
desired outcome

Increase λ while $|\hat{f}(x') - y'| > \varepsilon$



[Grath et al, 2018](#)

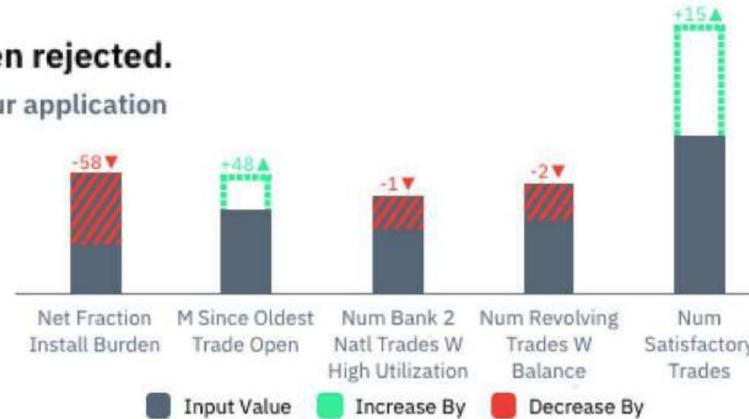
Counterfactual Explanations



Sorry, your loan application has been rejected.

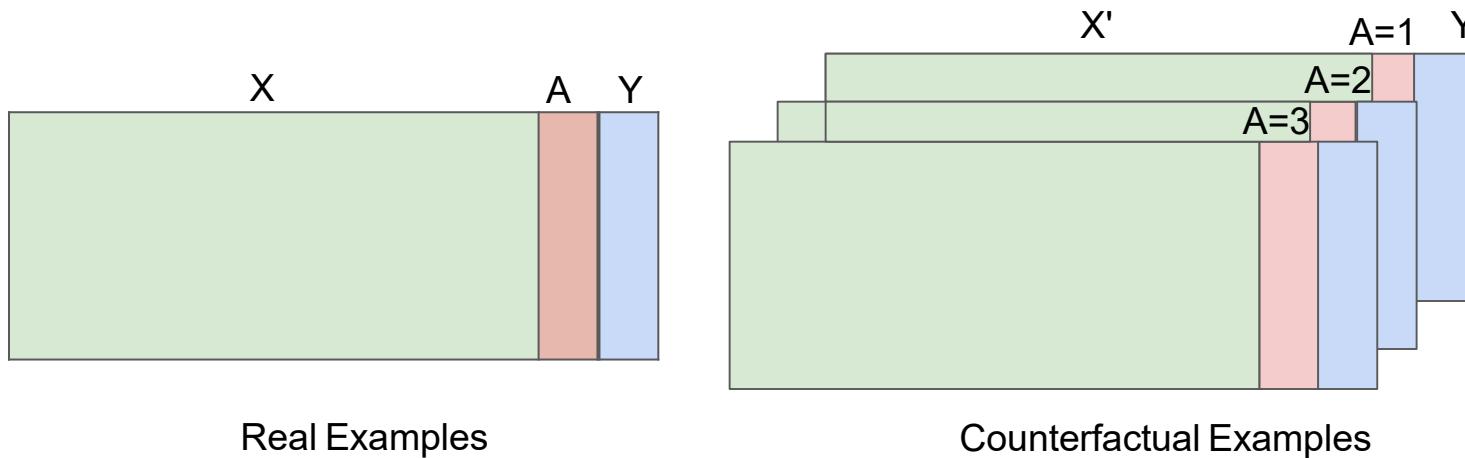
If instead you had the following values, your application would have been approved:

- MSinceOldestTradeOpen: **161**
- NumSatisfactoryTrades: **36**
- NetFractionInstallBurden: **38**
- NumRevolvingTradesWBalance: **4**
- NumBank2NatlTradesWHighUtilization: **2**



[Grath et al, 2018](#)

Counterfactual Fairness

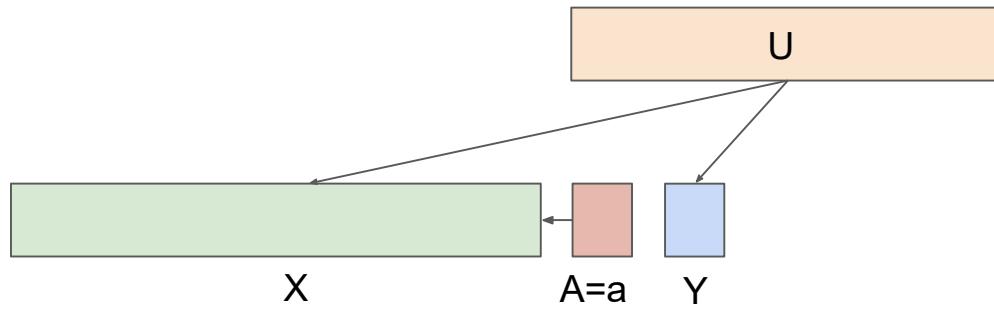


$$\frac{P(\hat{Y}_{A \leftarrow a}(U) = y \mid X = x, A = a)}{P(\hat{Y}_{A \leftarrow a'}(U) = y \mid X = x, A = a)}$$

Real Examples Counterfactual Examples

[Kusner et al, 2017](#)

Causal View of Counterfactual Fairness



Real Examples

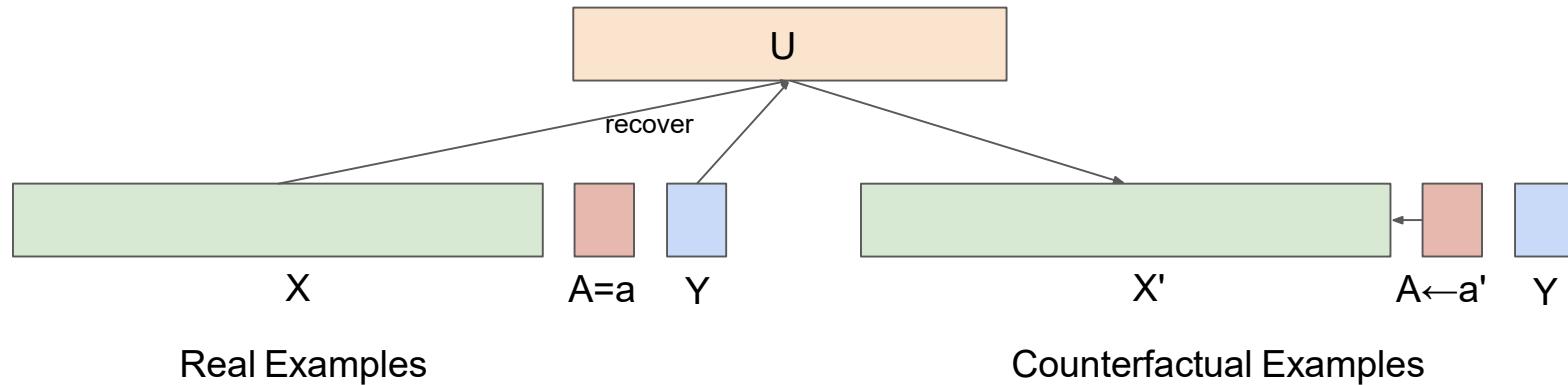
$$P(\hat{Y}_{A \leftarrow a}(U) = y \mid X = x, A = a) = P(\hat{Y}_{A \leftarrow a'}(U) = y \mid X = x, A = a)$$

Real Examples

Counterfactual Examples

[Kusner et al, 2017](#)

Causal View of Counterfactual Fairness



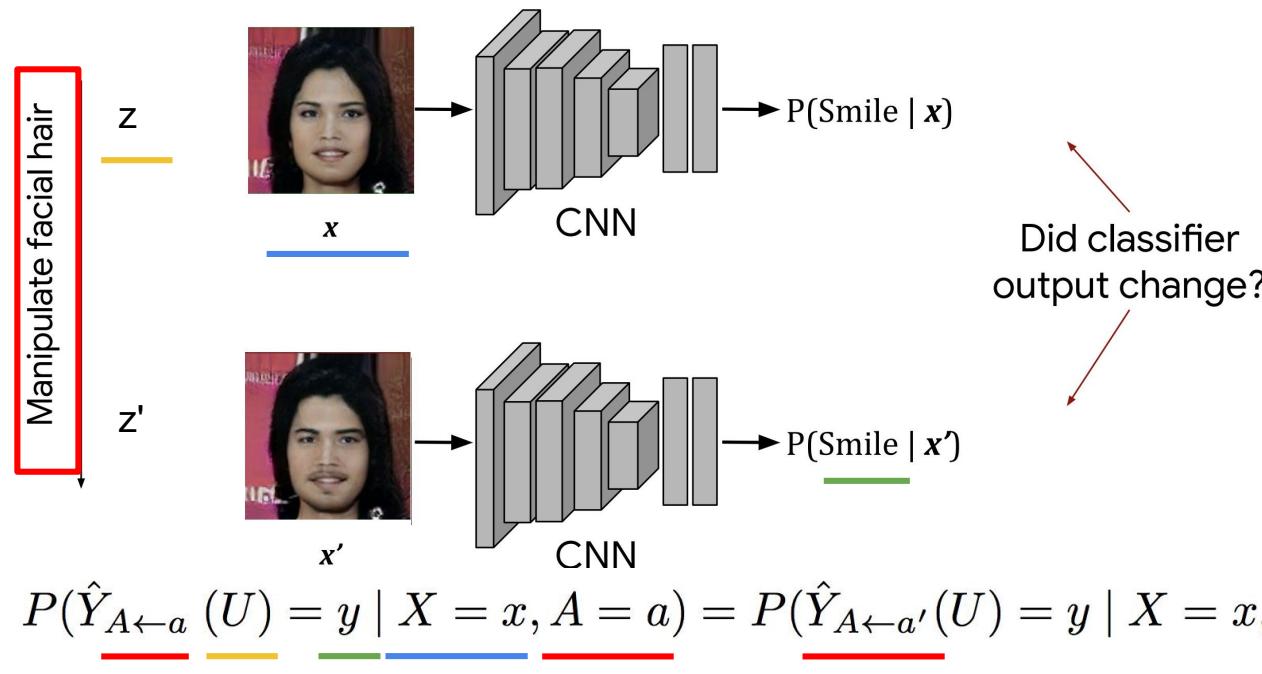
$$\frac{P(\hat{Y}_{A \leftarrow a}(U) = y \mid X = x, A = a)}{\text{Real Examples}} = P(\hat{Y}_{A \leftarrow a'}(U) = y \mid X = x, A = a)$$

Counterfactual Examples

[Kusner et al, 2017](#)

Counterfactual Face Attribution

- Evaluate the Counterfactual Fairness of Face Recognition Systems

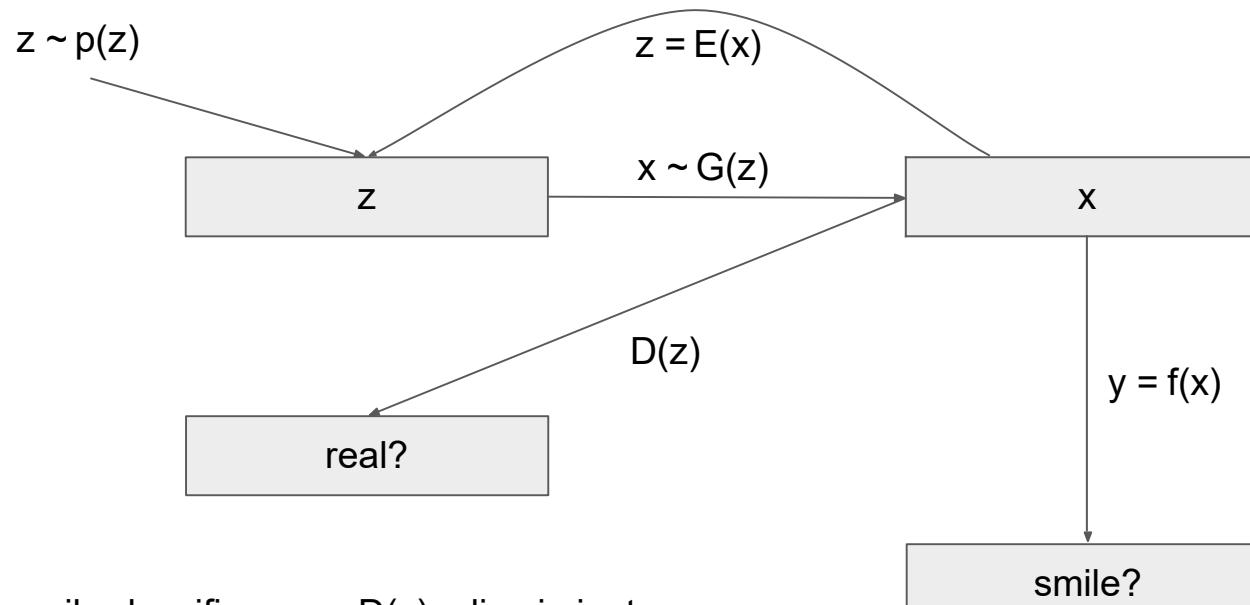


CelebA Dataset



[Liu et al, 2015](#)

Model Architecture

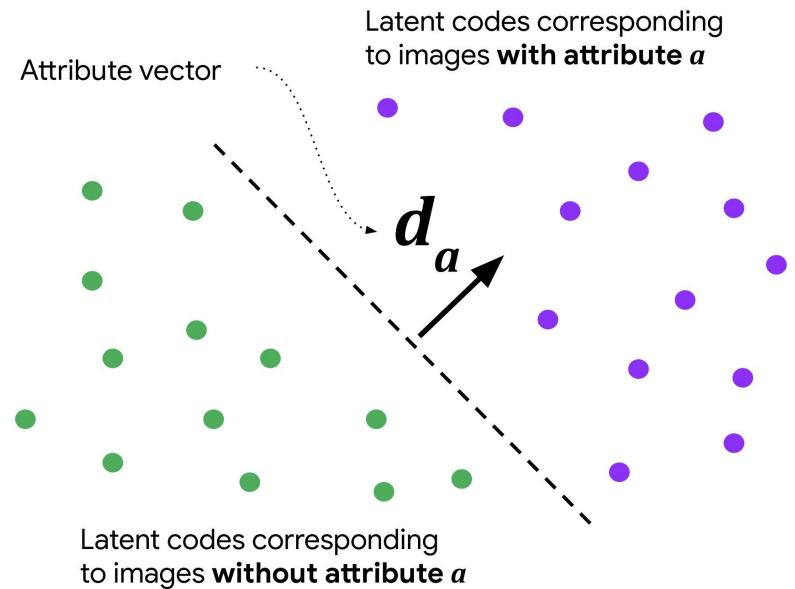
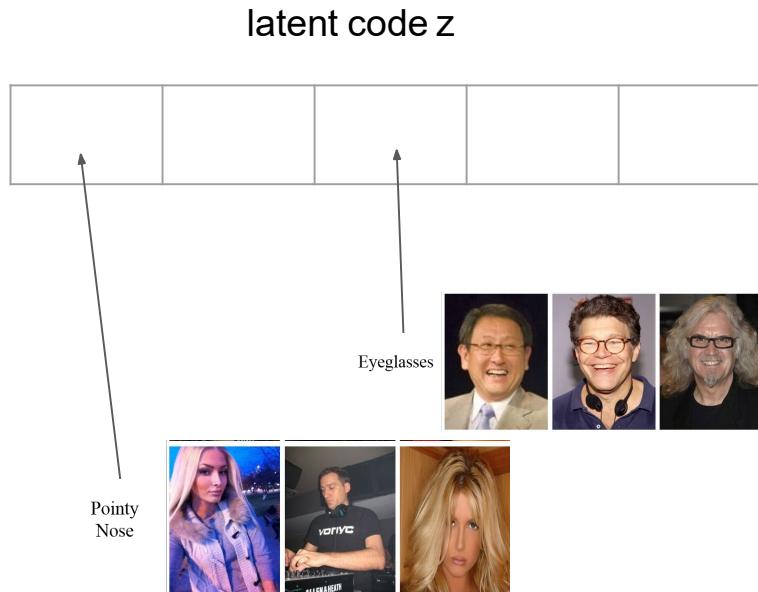


$f(x)$ - smile classifier
 $G(z)$ - generator

$D(z)$ - discriminator
 $E(x)$ - encoder

[Denton et al, 2019](#)

Latent Code Attribution

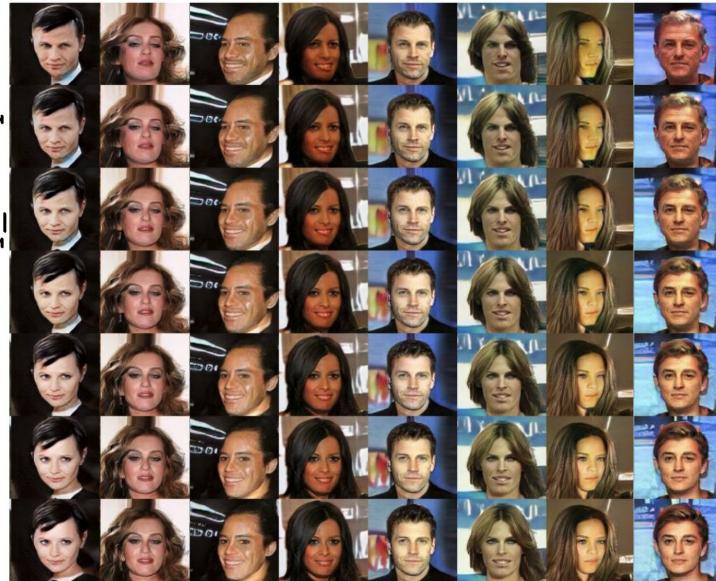


Latent Code Manipulation

$\leftarrow d_{Young}$

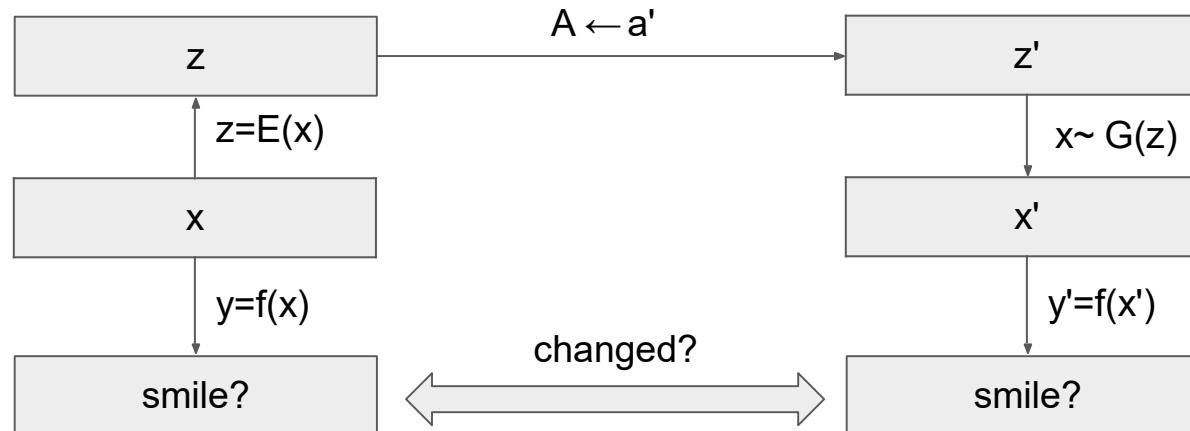


$\leftarrow d_{Heavy_Makeup}$



[Denton et al, 2019](#)

Counterfactual Fairness Assessment



$$P(\hat{Y}_{A \leftarrow a}(U) = y \mid X = x, A = a) = P(\hat{Y}_{A \leftarrow a'}(U) = y \mid X = x, A = a)$$

[Denton et al, 2019](#)

Sensitivity



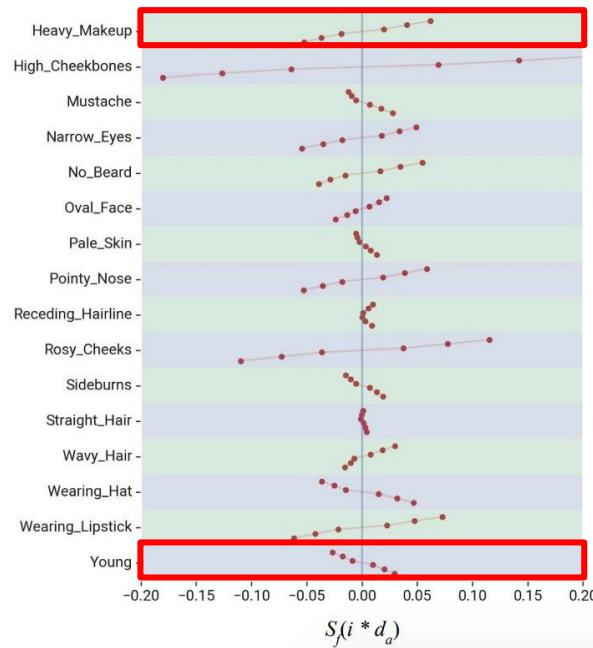
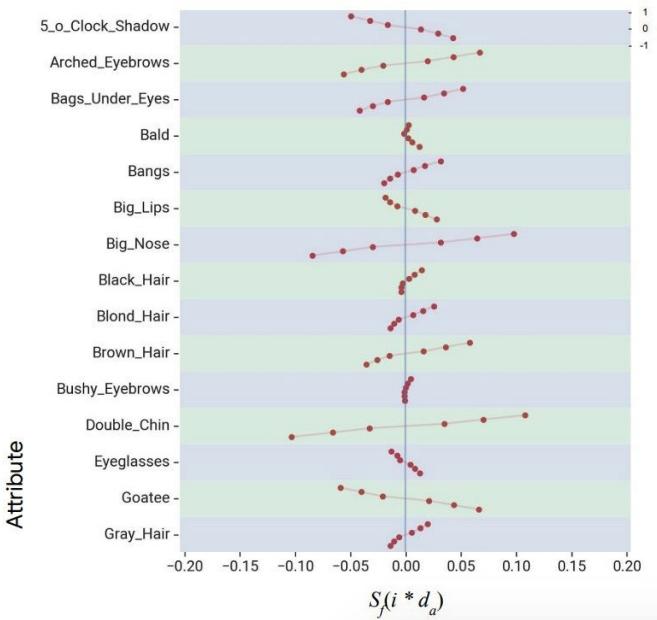
$$S_f(d) = \mathbb{E}_{z \sim p(z)} [f(\underline{G(z + d)}) - f(\underline{G(z)})]$$

$f(x)$ - smile classifier
 $G(z)$ - generator
 $D(z)$ - discriminator

$$P(\hat{Y}_{\underline{A \leftarrow a}}(U) = y \mid X = x, A = a) = P(\hat{Y}_{\underline{A \leftarrow a'}}(U) = y \mid X = x, A = a)$$

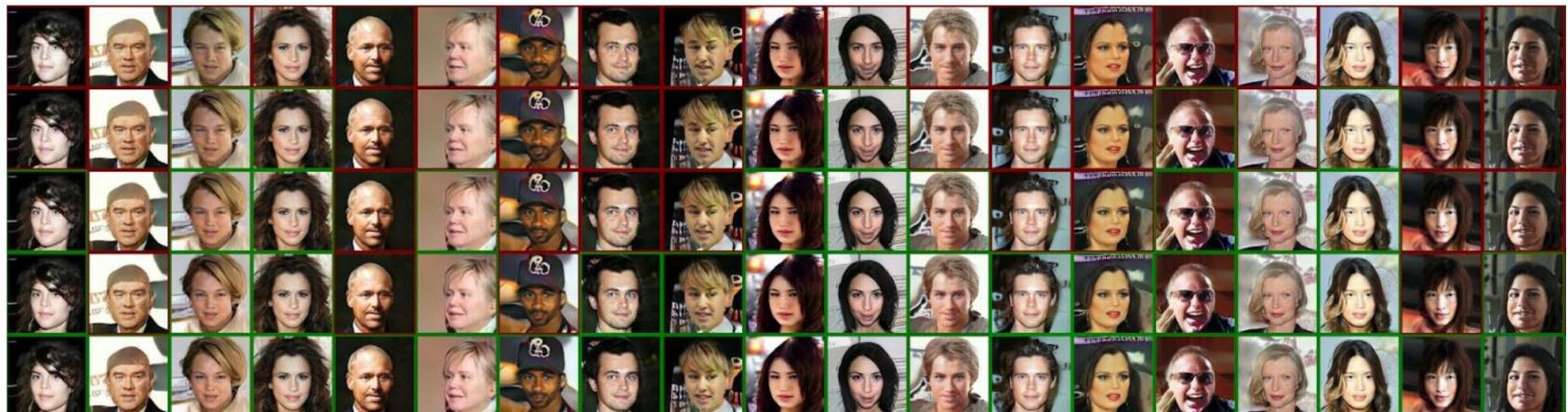
Sensitivity Results

$$S_f(d) = \mathbb{E}_{z \sim p(z)} [f(G(z + d)) - f(G(z))]$$



Heavy Makeup

$\leftarrow d_{\text{Heavy_Makeup}}$



Heavy_Makeup -

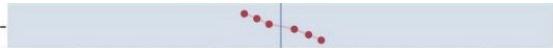


Young

→ d_{Young}



Young -



Directional Sensitivity

$$S_y^{0 \rightarrow 1}(d) = \mathbb{E}_{z \sim p(z) | y(G(z))=0} \mathbb{I}[y(G(z+d))! = y(G(z))]$$

from "not smiling" to "smiling"

$$S_y^{1 \rightarrow 0}(d) = \mathbb{E}_{z \sim p(z) | y(G(z))=1} \mathbb{I}[y(G(z+d))! = y(G(z))]$$

from "smiling" to "not smiling"

$$S_f(d) = \mathbb{E}_{z \sim p(z)} [f(G(z+d)) - f(G(z))]$$

Sensitivity Results

CelebA attribute defining d_a	$S_y^{1 \rightarrow 0}(d_a)$	$S_y^{0 \rightarrow 1}(d_a)$
Young	7.0%	2.6%
5_o_Clock_Shadow	11.8%	2.2%
Goatee	12.4%	0.9%
No_Beard	0.8%	11.8%
Heavy_Makeup	1.6%	12.4%
Wearing_Lipstick	1.7%	16.3%

The Equalizer Model

Wrong



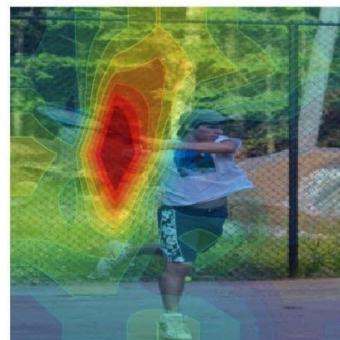
Baseline:
A **man** sitting at a desk with
a laptop computer.

Right for the Right
Reasons



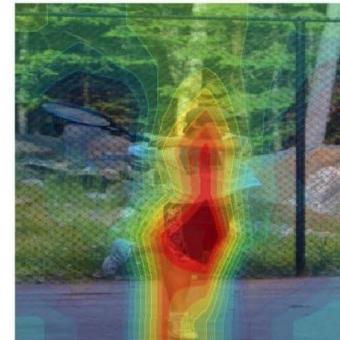
Our Model:
A **woman** sitting in front of a
laptop computer.

Right for the Wrong
Reasons



Baseline:
A **man** holding a tennis
racquet on a tennis court.

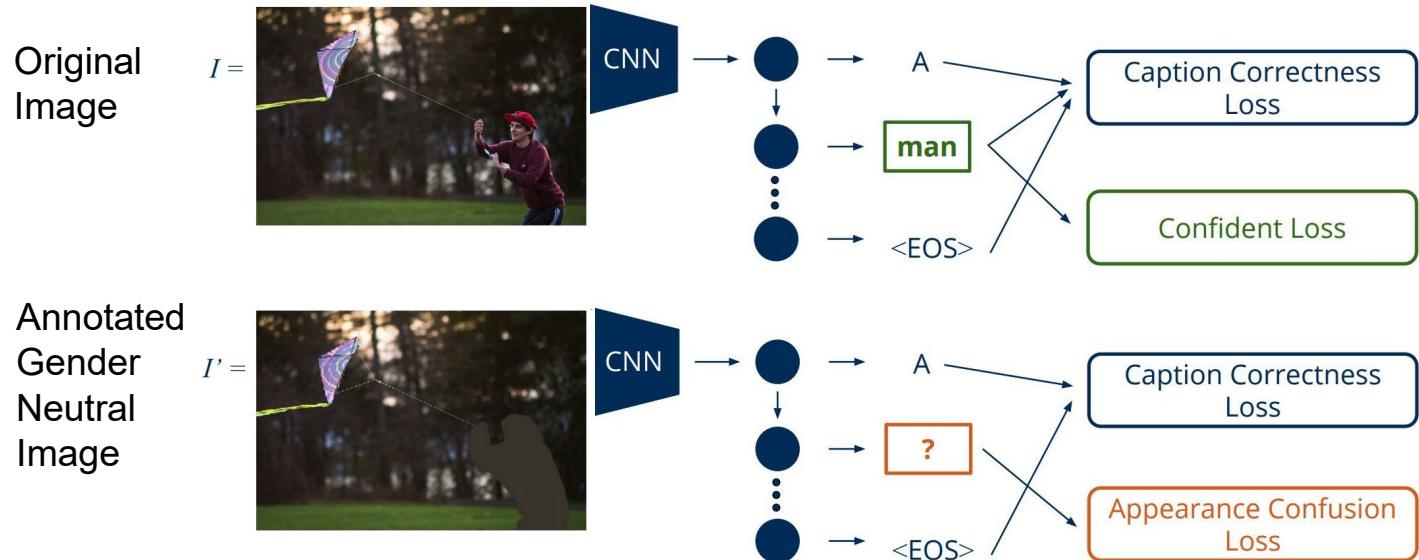
Right for the Right
Reasons



Our Model:
A **man** holding a tennis
racquet on a tennis court.

[Burns et al, 2019](#)

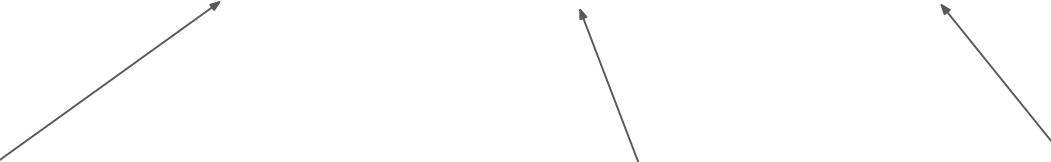
The Basic Idea



[Burns et al, 2019](#)

The Equalizer Model

$$\mathcal{L} = \alpha \mathcal{L}^{CE} + \beta \mathcal{L}^{AC} + \mu \mathcal{L}^{Con}$$



Cross Entropy Loss
 $\mathcal{L}^{CE} = -\frac{1}{N} \sum_{n=0}^N \sum_{t=0}^T \log(p(w_t|w_{0:t-1}, I))$

Appearance Confusing Loss on the gender neutral image

Confidence Loss on the original image

[Burns et al, 2019](#)

Appearance Confusing Objective

$$\mathcal{L}^{AC} = \frac{1}{N} \sum_{n=0}^N \sum_{t=0}^T \mathbb{1}(w_t \in \mathcal{G}_w \cup \mathcal{G}_m) \mathcal{C}(\tilde{w}_t, I')$$

$$\mathcal{C}(\tilde{w}_t, I') = \left| \sum_{g_w \in \mathcal{G}_w} p(\tilde{w}_t = g_w | w_{0:t-1}, I') - \sum_{g_m \in \mathcal{G}_m} p(\tilde{w}_t = g_m | w_{0:t-1}, I') \right|$$

Push Toward Extremes

$$p(\tilde{w}_t = g_w | w_{0:t-1}, I') \longleftrightarrow p(\tilde{w}_t = g_m | w_{0:t-1}, I')$$

\mathcal{G}_w - set of words for woman

\mathcal{G}_m - set of words for man



[Burns et al, 2019](#)

Confidence Objective

$$\mathcal{L}^{Con} = \frac{1}{N} \sum_{n=0}^N \sum_{t=0}^T (\mathbb{1}(w_t \in \mathcal{G}_w) \mathcal{F}^W(\tilde{w}_t, I) + \mathbb{1}(w_t \in \mathcal{G}_m) \mathcal{F}^M(\tilde{w}_t, I))$$

$$\mathcal{F}^W(\tilde{w}_t, I) = \frac{\sum_{g_m \in \mathcal{G}_m} p(\tilde{w}_t = g_m | w_{0:t-1}, I)}{(\sum_{g_w \in \mathcal{G}_w} p(\tilde{w}_t = g_w | w_{0:t-1}, I)) + \epsilon}$$



\mathcal{G}_w - set of words for woman
 \mathcal{G}_m - set of words for man

[Burns et al, 2019](#)

Results

Model	MSCOCO-Bias		MSCOCO-Balanced	
	Error	Ratio Δ	Error	Ratio Δ
Baseline-FT	12.83	0.15	19.30	0.51
Balanced	12.85	0.14	18.30	0.47
UpWeight	13.56	0.08	16.30	0.35
Equalizer w/o ACL	7.57	0.04	10.10	0.26
Equalizer w/o Conf	9.62	0.09	13.90	0.40
Equalizer	7.02	-0.03	8.10	0.13

Baseline-FT - basic LSTM attention model ([Xu et al, 2015](#))

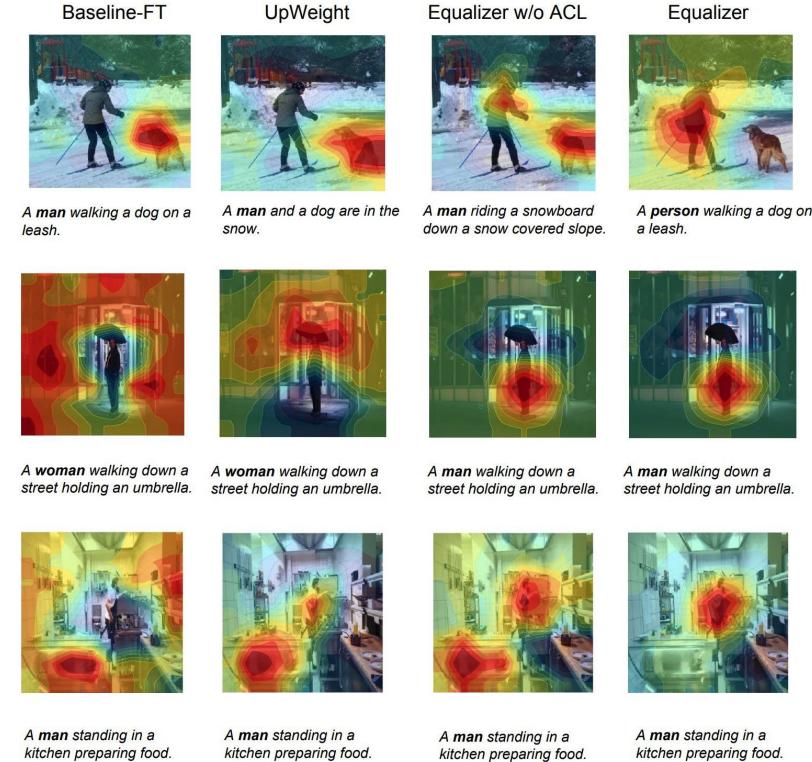
Balanced - resampled dataset to have balanced gender ratio

UpWeight - reweighting

Δ - change to the gender ratio compared to the dataset

[Burns et al, 2019](#)

Results

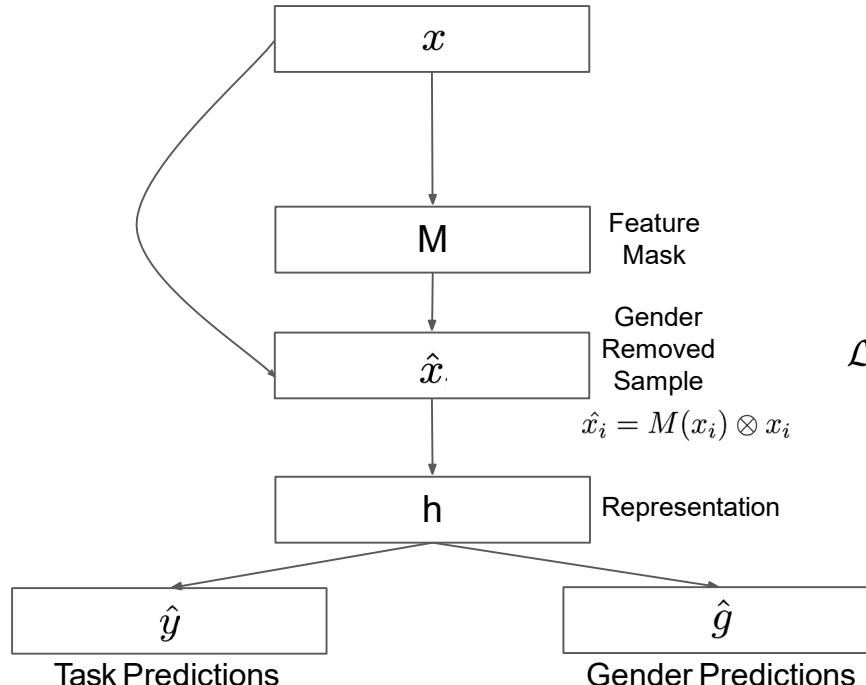


Adversarial Removal of Gender Features



[Wang et al, 2019](#)

Model Architecture

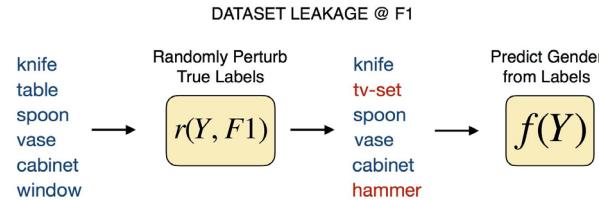


$$\mathcal{L} = \sum_i \beta |x_i - \hat{x}_i| + \mathcal{L}_p(\text{pred}(h(\hat{x}_i)), y_i) - \lambda \mathcal{L}_c(c(h(\hat{x}_i)), g_i)$$

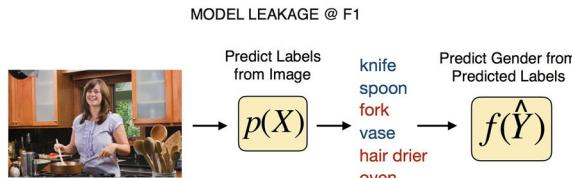
[Wang et al, 2019](#)

Evaluate Sensitive Information Leakage

- Train an attacker $f(y)$ that reverse engineer the gender information



$$\lambda_D(a) = \frac{1}{|\mathcal{D}|} \sum_{(Y_i, g_i) \in \mathcal{D}} \mathbb{1}[f(r(Y_i, a)) == g_i]$$



$$\lambda_M(a) = \frac{1}{|\mathcal{D}|} \sum_{(\hat{Y}_i, g_i) \in \mathcal{D}} \mathbb{1}[f(r(\hat{Y}_i, a)) == g_i]$$

Data Resampling

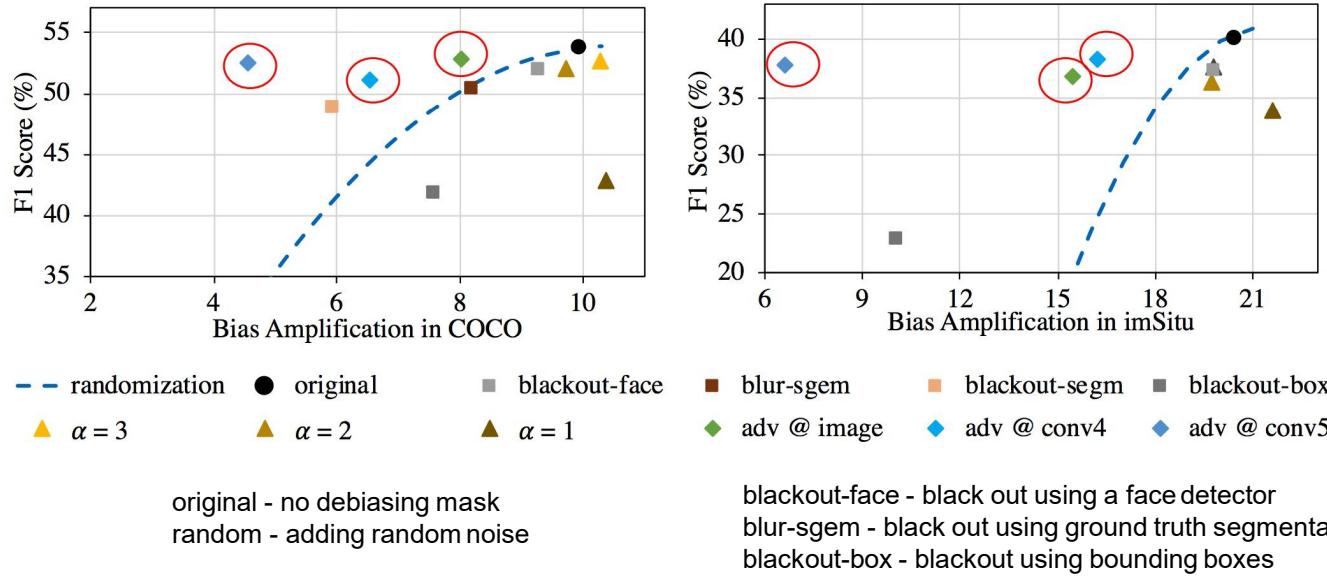
$$\forall y : 1/\alpha < \#(m, y)/\#(w, y) < \alpha$$

Bias Amplification

$$\Delta = \lambda_M(a) - \lambda_D(a)$$

[Wang et al, 2019](#)

Accuracy and Bias Results



[Wang et al, 2019](#)

Qualitative Results

COCO Results



imSitu Results



[Wang et al, 2019](#)

Summary

- Fair Machine Learning
 - Prevents ML models from biasing toward specific groups when allocating favorable outcomes
- Group Treatments of Fairness
 - Demographic Parity
 - Equalized Odds/Opportunity
- Individual Treatments of Fairness
 - Fairness Through Awareness Individual Fairness
 - Individual Fairness
 - Counterfactual Fairness
- Fair ML Techniques
 - Pre-processing Methods: Resampling, Reweighting, Optimized-preprocessing
 - In-processing Methods: Regularization, Adversarial Learning
 - Post-processing Methods: Learning to Defer

Summary

- Fair NLP Methods
 - Debiasing Word Embeddings
 - Data Augmentation
 - Gender Swapping
 - Fair Representation for Pre-trained Encoders

- Fair Visual Representations
 - Counterfactual Face Attribution
 - Gender Equalized Image Captioning
 - Adversarial Removal of Gender Features



Fair, Accountable, Transparent Machine Learning (FAccT ML)

ZG517

Dr. Sugata Ghosal

sugata.ghosal@pilani.bits-pilani.ac.in



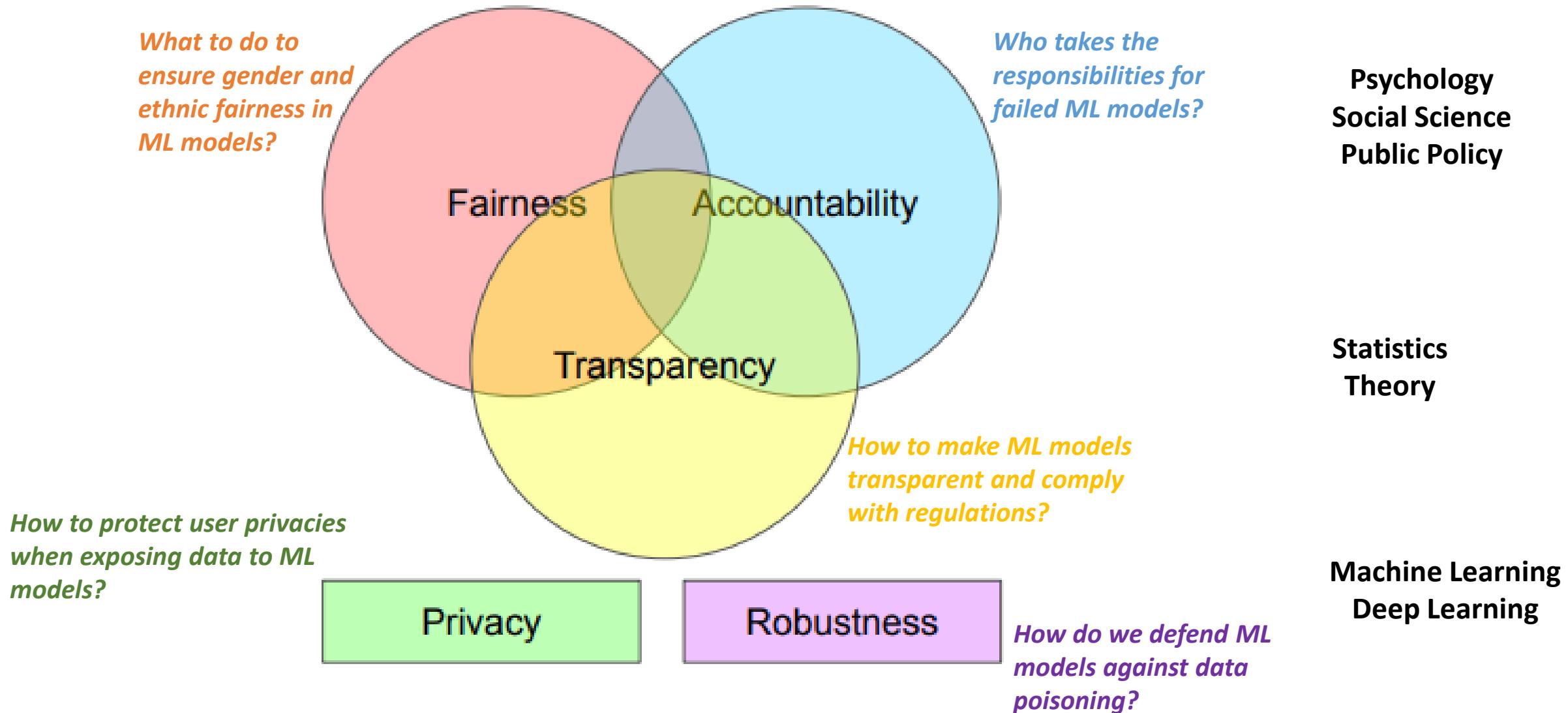
BITS Pilani
Pilani Campus



Session 8
Date – 9th July 2023
Time – 8:45 AM to 10:45 AM

These slides are prepared by the instructor, with grateful acknowledgement of many others who made their course materials freely available online.

FAccT ML: Course Overview



Fairness and Bias

1. Fairness and Bias

- ✓ Sources of Bias
- ✓ Real world examples
 - ✓ School admissions, criminal justice, hiring, gender/occupation bias
- ✓ Sensitive Features
- ✓ Fairness through unawareness

2. Learning Fair Representations

- ✓ Major Fairness criteria
 - ✓ Direct Solution Method
 - ✓ Demographic Parity
 - ✓ Equality of Odds/Opportunity
- ✓ Prejudice Removing Regularizer
 - ✓ Prejudice index (PI)
 - ✓ Optimizing PI
- ✓ Case Studies: FICO, adult income

3. Fairness thru input manipulation

- ✓ Basic Data Manipulation Techniques
 - ✓ Reweighting
 - ✓ Universal Sampling
 - ✓ Preferential Sampling

3. Fairness thru input manipulation

- ✓ Individual Fairness
- ✓ Optimized Pre-processing
- ✓ Learning to Defer

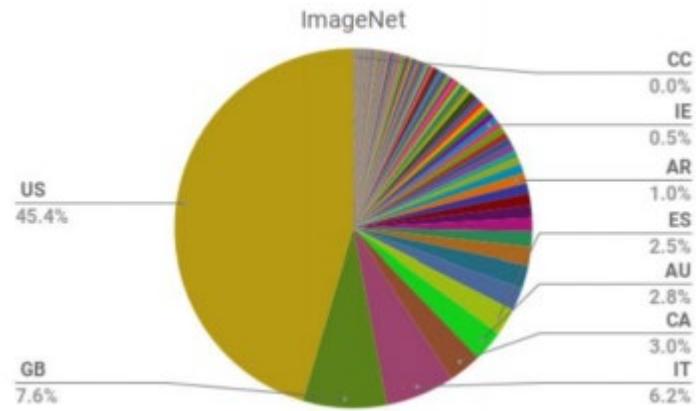
4. Fair Casual Reasoning

- ✓ Causal Fairness and Inherent Bias
- ✓ Counterfactual Fairness
 - ✓ Formal Methods
 - ✓ Case Study – Success in Law School
 - ✓ Case Study - Crime Rates and Arrest
- ✓ Equalized Counterfactual Odds
- ✓ Multiple Causal Worlds

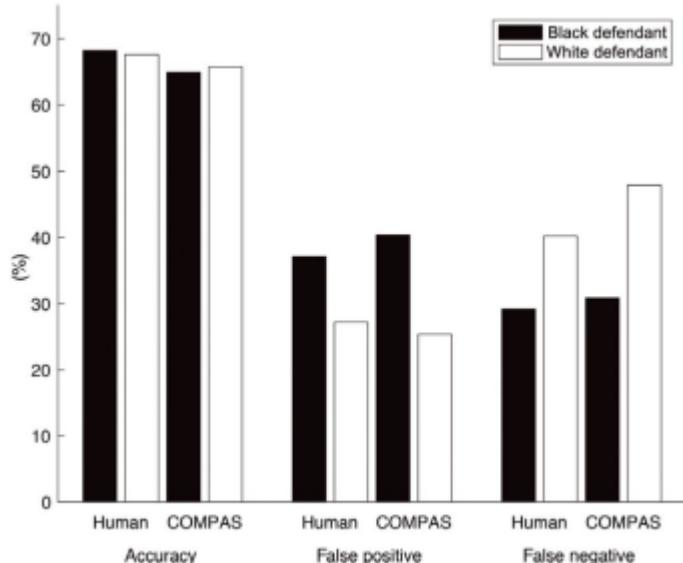
5. Fairness in NLP

- ✓ Biases in NLP Models
- ✓ Data Augmentation
- ✓ Debiasing Word Embedding
- ✓ Counterfactual Fairness

Data with Bias



Criminal Justice



Univ Admissions

	Men		Women	
	Applicants	Admitted	Applicants	Admitted
Total	8442	44%	4321	35%

Department	Men		Women	
	Applicants	Admitted	Applicants	Admitted
A	825	62%	108	82%
B	560	63%	25	68%
C	325	37%	593	34%
D	417	33%	375	35%
E	191	28%	393	24%
F	373	6%	341	7%

Word Embeddings

He is...



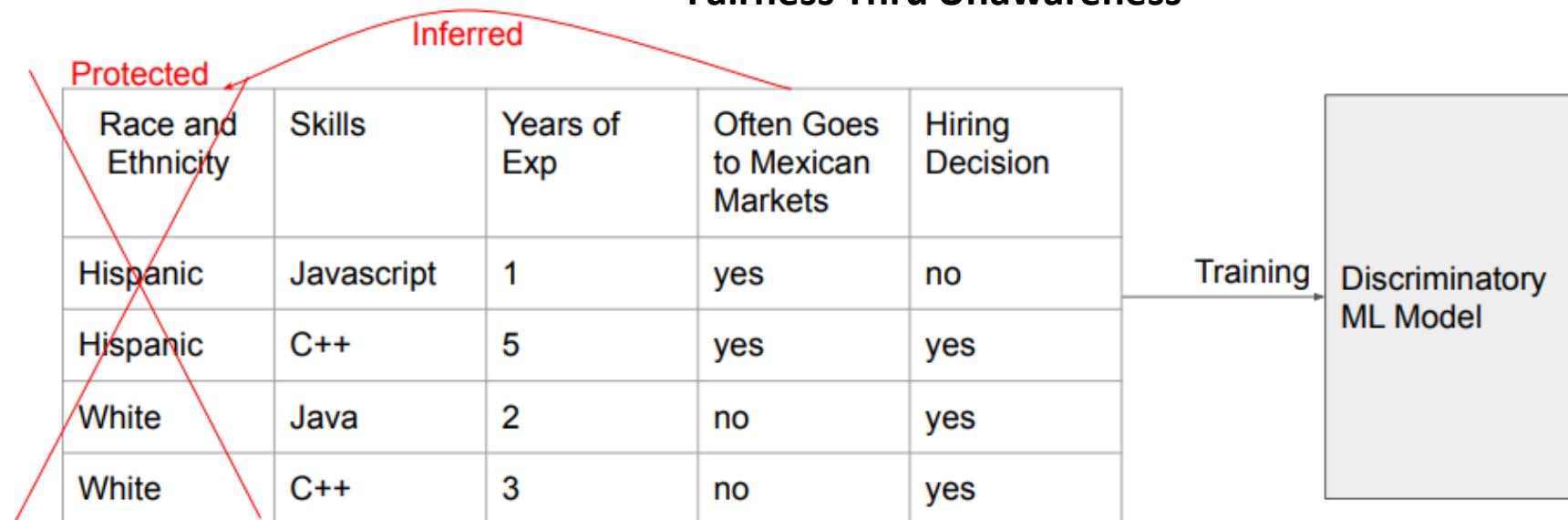
She is...



Protected Attributes Fair Lending Laws

Attribute	FHA	ECOA
Race	✓	✓
Color	✓	✓
National origin	✓	✓
Religion	✓	✓
Sex	✓	✓
Familial status	✓	
Disability	✓	
Exercised rights under CCPA		✓
Marital status		✓
Recipient of public assistance		✓
Age		✓

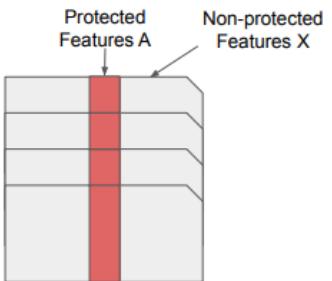
Fairness Thru Unawareness



Fair Representation Learning

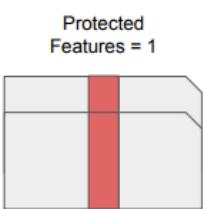
Demographic Parity

Individual Treatment

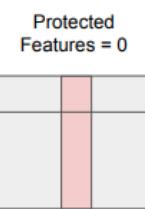


Fairness Through Unawareness
 $P(\hat{Y} | X)$

Group Treatment



Demographic Parity
 $P(\hat{Y}=1 | A=1)$



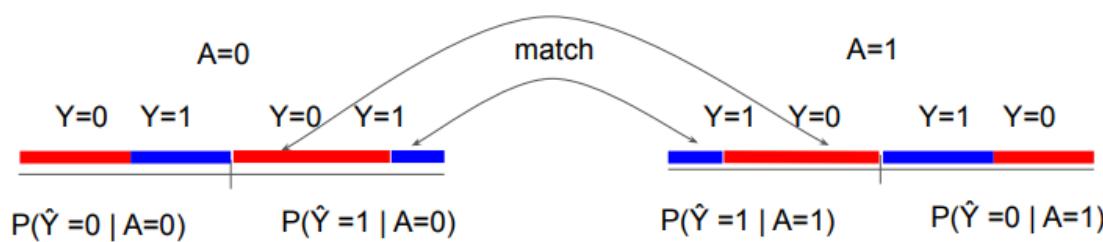
Demographic Parity
 $P(\hat{Y}=1 | A=0)$

Prejudice Removal Regularizer

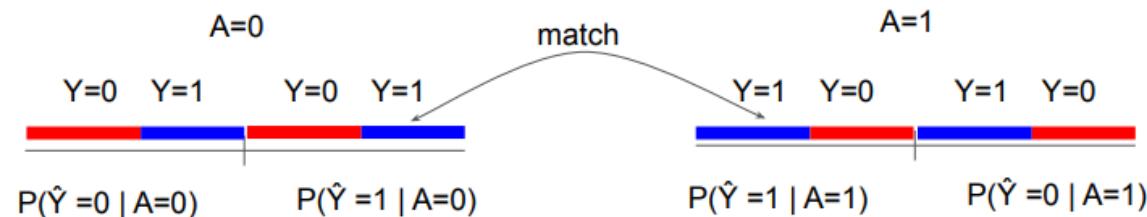
$$-\mathcal{L}(\mathcal{D}; \Theta) + \eta R(\mathcal{D}, \Theta) + \frac{\lambda}{2} \|\Theta\|_2^2$$

Loss of the Model Fairness Regularizer L2 Regularizer

Equality of Odds



Equality of Opportunity



Case Studies

1. Adult Income Dataset - Predict Whether Income Exceeds \$50K/yr Based on Census Data
2. FICO Dataset - Making Lending Decisions Without Discriminating

Traditional Evaluation of ML Algorithms

- Accuracy
- Precision and recall
- Squared error
- Likelihood
- Posterior probability
- Cost / Utility
- Margin
- Entropy
- etc.

Evaluating Performance

- If y is continuous:
 - Sum-of-Squared-Differences (SSD) error between predicted and true y :

$$E = \sum_{i=1}^n (f(x_i) - y_i)^2$$

Evaluation Metrics: Accuracy

		PREDICTED CLASS	
		Class=Yes	Class>No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class>No	c (FP)	d (TN)

- Most widely-used metric:

$$\text{Accuracy} = \frac{a + d}{a + b + c + d} = \frac{TP + TN}{TP + TN + FP + FN}$$

Alternative Measures

		PREDICTED CLASS	
		Class=Yes	Class>No
ACTUAL CLASS	Class=Yes	a	b
	Class>No	c	d

$$\text{Precision (p)} = \frac{a}{a + c}$$

$$\text{Recall (r)} = \frac{a}{a + b}$$

$$\text{F - measure (F)} = \frac{2rp}{r + p} = \frac{2a}{2a + b + c}$$

Measures of Classification Performance

	PREDICTED CLASS	
ACTUAL CLASS	Yes	No
	Yes	TP
	No	FP

α is the probability that we reject the null hypothesis when it is true. This is a Type I error or a false positive (FP).

β is the probability that we accept the null hypothesis when it is false. This is a Type II error or a false negative (FN).

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN}$$

$$\text{ErrorRate} = 1 - \text{accuracy}$$

$$\text{Precision} = \text{Positive Predictive Value} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \text{Sensitivity} = \text{TP Rate} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = \text{TN Rate} = \frac{TN}{TN + FP}$$

$$\text{FP Rate} = \alpha = \frac{FP}{TN + FP} = 1 - \text{specificity}$$

$$\text{FN Rate} = \beta = \frac{FN}{FN + TP} = 1 - \text{sensitivity}$$

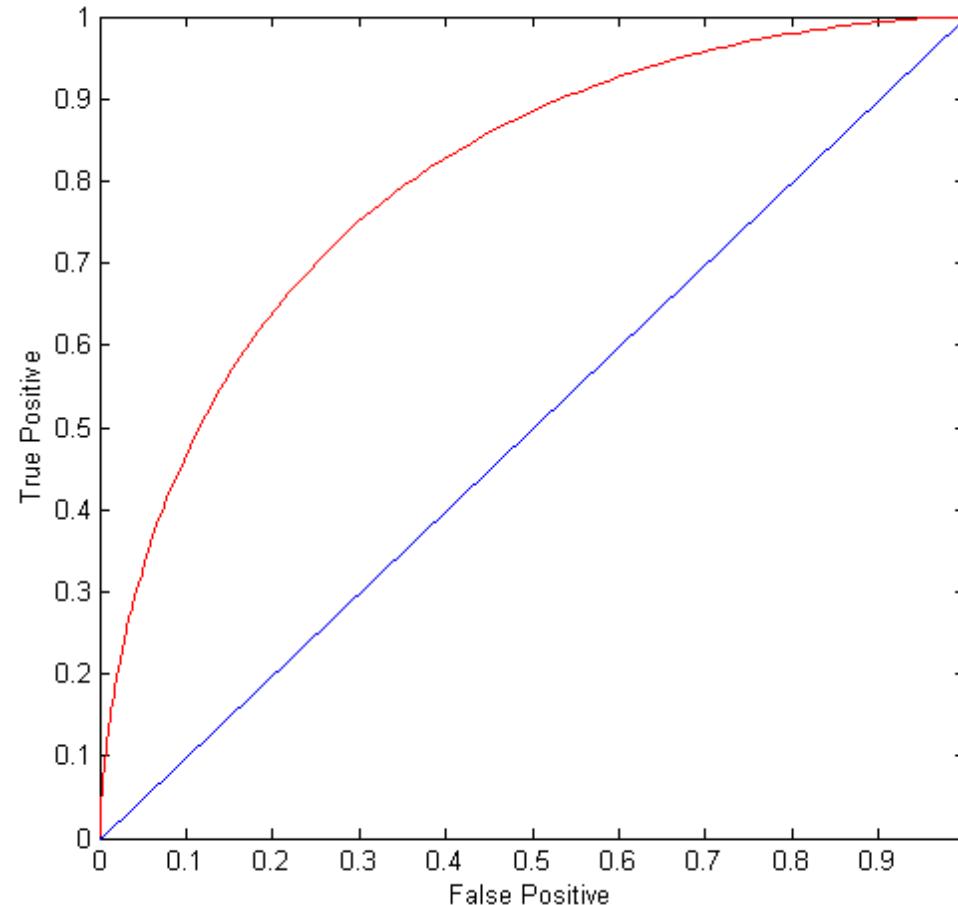
$$\text{Power} = \text{sensitivity} = 1 - \beta$$

ROC Curve

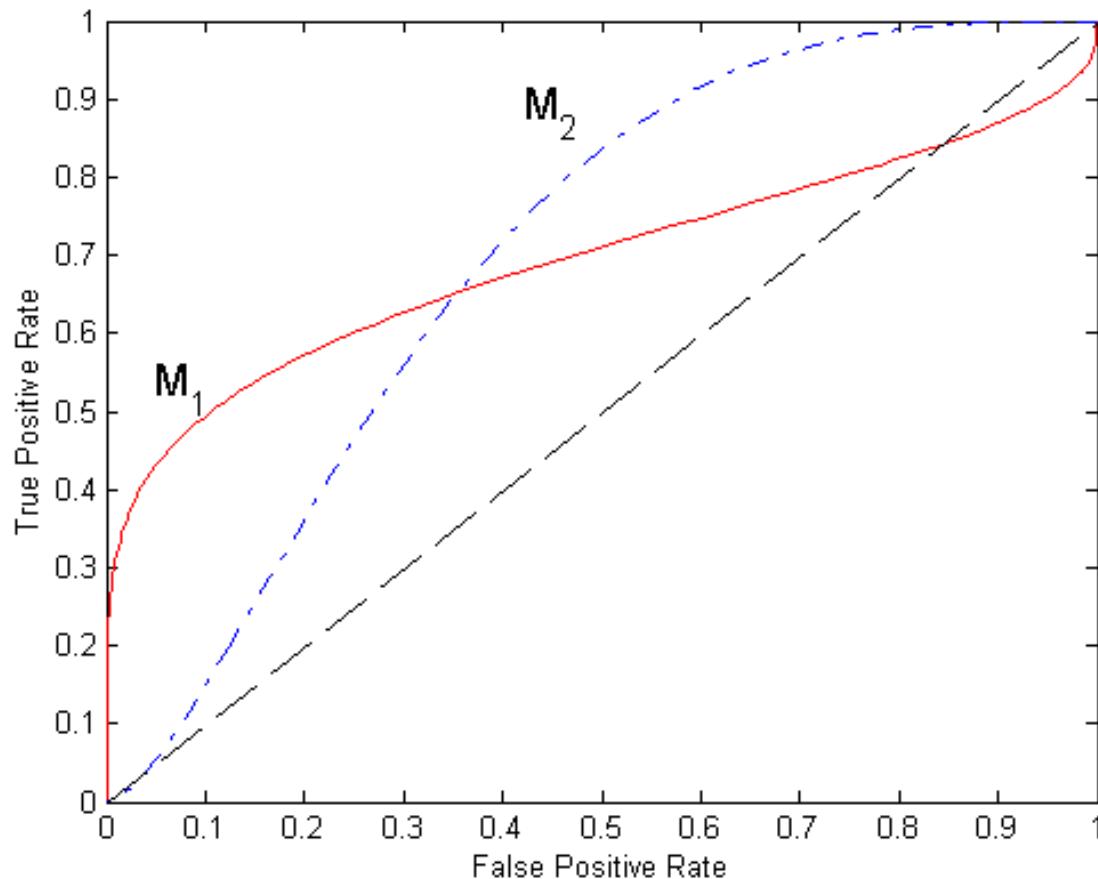
(TPR,FPR):

- (0,0): declare everything to be negative class
- (1,1): declare everything to be positive class
- (0,1): ideal

- Diagonal line:
 - Random guessing
 - Below diagonal line:
 - prediction is opposite of the true class



Using ROC for Model Comparison



- No model consistently outperforms the other
 - M_1 is better for small FPR
 - M_2 is better for large FPR
- Area Under the ROC curve (AUC)
 - Ideal:
 - Area = 1
 - Random guess:
 - Area = 0.5

Practice Question

Find out the Fairness Criteria that \hat{Y}_1 , and \hat{Y}_2 Satisfy

- $A = \{\text{race}\}$, $Y = \{\text{Hiring Decision}\}$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Demographic Parity for Predictor \hat{Y}_1

$$P(\hat{Y}_1 = 1 | R = H) = 2/3$$

$$P(\hat{Y}_1 = 1 | R = W) =$$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Demographic Parity for Predictor \hat{Y}_1

$$P(\hat{Y}_1 = 1 | R = H) = 2/3$$

$$P(\hat{Y}_1 = 1 | R = W) = 2/3$$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Demographic Parity for Predictor \hat{Y}_1

- $P(\hat{Y}_1 = 1 | R = H) = 2/3$
- $P(\hat{Y}_1 = 1 | R = W) = 2/3$

✓ Demographics

$$P(\hat{Y} = \text{Parity} = 1) = P(\hat{Y} = 1 | A = 0)$$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_1

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = 1$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) =$
- $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) =$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) =$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_1

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = 1$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) = 0.5$
- $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) =$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) =$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_1

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = 1$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) = 0.5$
- $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) = 0$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) =$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_1

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = 1$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) = 0.5$
- $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) = 0$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) = 1$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_1

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = 1$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) = 0.5$
- $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) = 0$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) = 1$



X Equality of Opportunity

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

X Equality of Odds

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

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Demographic Parity for Predictor \hat{Y}_2

- $P(\hat{Y}_1 = 1 | R = H) = 2/3$
- $P(\hat{Y}_1 = 1 | R = W) =$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Demographic Parity for Predictor \hat{Y}_2

- $P(\hat{Y}_1 = 1 | R = H) = 2/3$
- $P(\hat{Y}_1 = 1 | R = W) = 1/3$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Demographic Parity for Predictor \hat{Y}_2

- $P(\hat{Y}_1 = 1 | R = H) = 2/3$
- $P(\hat{Y}_1 = 1 | R = W) = 1/3$

X Demographics Parity

$$P(\hat{Y} = 1 | A = 1) = P(\hat{Y} = 1 | A = 0)$$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_2

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = 1/2$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) =$
- $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) =$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) =$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Demographic Parity for Predictor \hat{Y}_2

- $P(\hat{Y}_1 = 1 | R = H) = 2/3$
- $P(\hat{Y}_1 = 1 | R = W) = 1/3$

X Demographics Parity

$$P(\hat{Y} = 1 | A = 1) = P(\hat{Y} = 1 | A = 0)$$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_2

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = 1/2$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) = 1/2$
- $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) =$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) =$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_2

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = 1/2$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) = 1/2$
- $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) = 1$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) =$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_2

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = \frac{1}{2}$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) = 1/2$
- $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) = 1$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) = 0$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Equality of Opportunity/Odds for \hat{Y}_2

- $P(\hat{Y}_1 = 1 | R = H, Y = \text{yes}) = 1/2$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{yes}) = 1/2$
- $P(\hat{Y}_1 = 1 | R = H, Y = \text{no}) = 1$
- $P(\hat{Y}_1 = 1 | R = W, Y = \text{no}) = 0$

✓ Equality of
 $P(\hat{Y} = 1 | A = 1, Y) = P(\hat{Y} = 1 | A = 1, Y = 1)$
 ✗ Equality of Odds
 $P(\hat{Y} = 1 | A = 0, Y) = P(\hat{Y} = 1 | A = 1, Y)$

Race and Ethnicity	Skill	Years of Exp	Goes to Mexican Markets?	Hiring Decision Y	Predictor \hat{Y}_1	Predictor \hat{Y}_2
Hispanic	Javascript	1	yes	no	0	1
Hispanic	C++	5	yes	yes	1	1
Hispanic	Python	1	no	yes	1	0
White	Java	2	no	yes	0	0
White	C++	3	no	yes	1	1
White	C++	0	no	no	1	0

Summary of Fairness Criteria

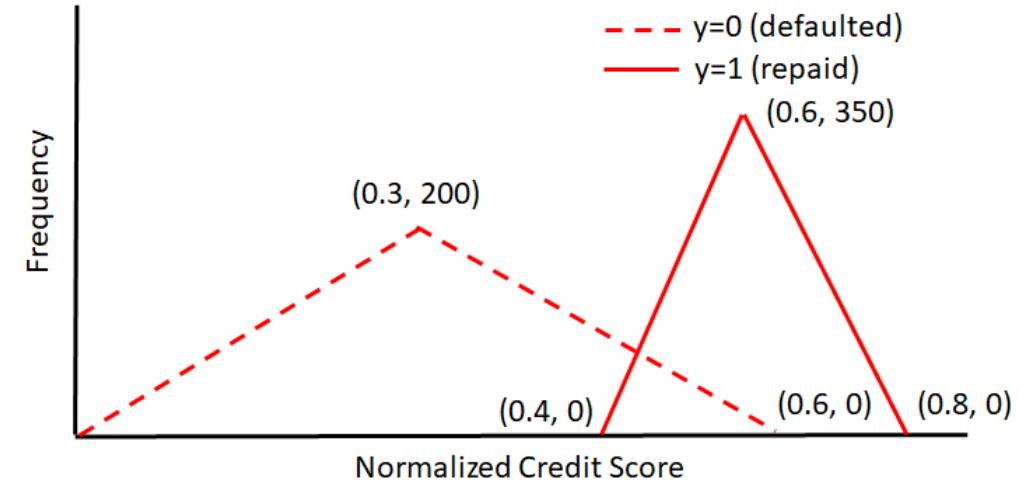
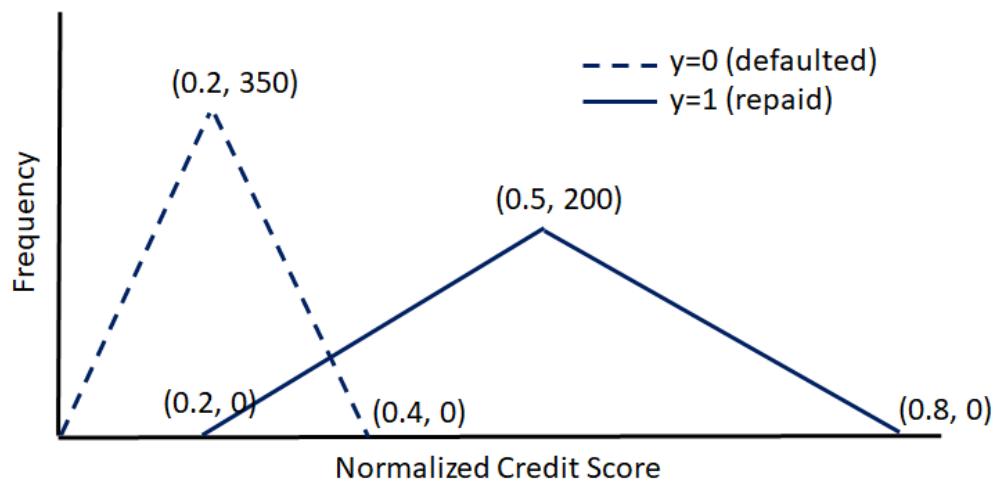
Fairness Criteria	Criteria	Group	Individual
Unawareness	Excludes A in Predictions		✓
Demographic Parity	$P(\hat{Y} = 1 A = 0) = P(\hat{Y} = 1 A = 1)$	✓	
Equalized Odds	$P(\hat{Y} = 1 A = 0, Y) = P(\hat{Y} = 1 A = 1, Y)$	✓	
Equalized Opportunity	$P(\hat{Y} = 1 A = 0, Y = 1) = P(\hat{Y} = 1 A = 1, Y = 1)$	✓	

Summary of Fairness Criteria

Fairness Criteria	Criteria	Group	Individual
Unawareness	Excludes A in Predictions		✓
Demographic Parity	$P(\hat{Y} = 1 A = 0) = P(\hat{Y} = 1 A = 1)$	✓	
Equalized Odds	$P(\hat{Y} = 1 A = 0, Y) = P(\hat{Y} = 1 A = 1, Y)$	✓	
Equalized Opportunity	$P(\hat{Y} = 1 A = 0, Y = 1) = P(\hat{Y} = 1 A = 1, Y = 1)$	✓	

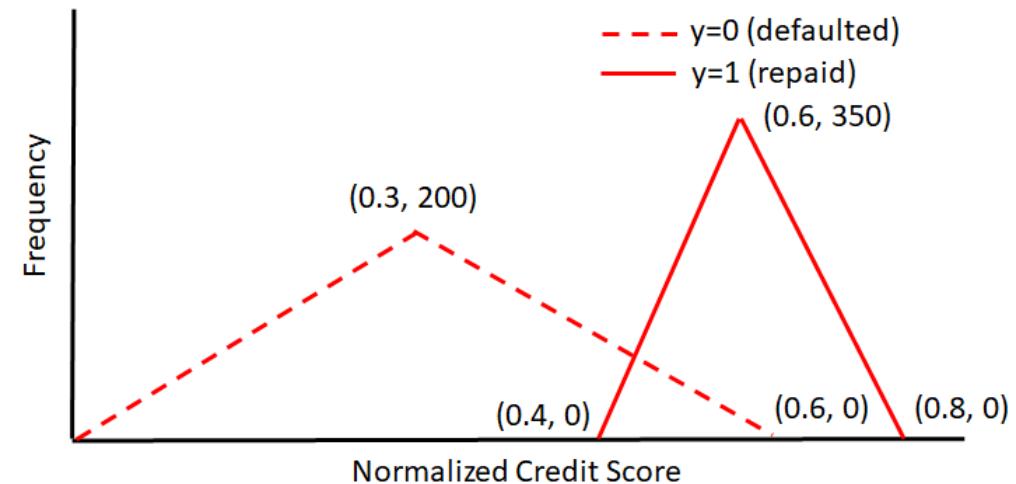
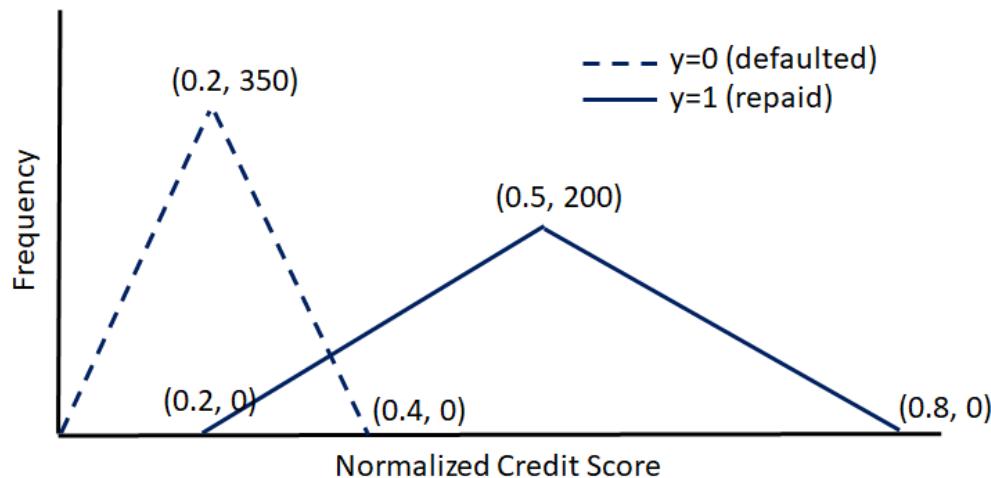
ML EC-3M Problem

Consider a financial institution that uses normalized credit score for approving or rejecting housing loan. Note there are two population of applicants 'blue' (deprived group with protected attribute $A=0$) and 'red' (favored group with $A=1$). Loan is approved, i.e., $y'=1$ if normalized credit score is greater than some threshold t , else rejected. $y=1$ indicates approved loans that are repaid based on historical data, and $y=0$ indicates approved loan was defaulted. The financial institution wants to approve loans that is likely to be repaid. Distributions are described in following figure (not drawn to scale) for both populations.



ML EC-3M Problem

A. Calculate probability $P(y'=1)$ for both 'blue' and 'red' populations for threshold $t=0.4$. Is fairness achieved w.r.t. protected attribute A?



For 'blue' population,

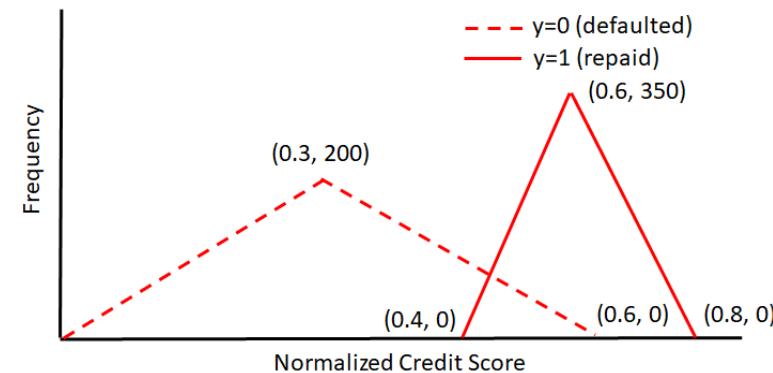
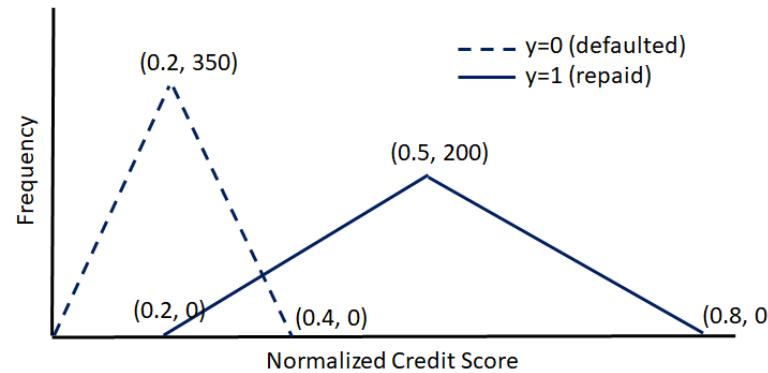
$$P(y'=1) = [0.5 \cdot 200 \cdot 0.6 - 0.5 \cdot 0.2 \cdot 133.3] / [0.2 \cdot 350 + 0.3 \cdot 200] = 0.359$$

For 'red' population,

$$P(y'=1) = [0.2 \cdot 350 + 0.5 \cdot 0.2 \cdot 133.3] / [0.3 \cdot 200 + 0.2 \cdot 350] = 0.641 \rightarrow \text{Fairness not achieved.}$$

ML EC-3M Problem

B. If threshold $t=0.45$ is chosen for the blue population, calculate the probability $P(y'=1)$ for the blue population. What threshold value for the red population will ensure demographic parity?



For 'blue' population, for $t=0.45$

$$P(Y'=1) = [0.5 \cdot 200 \cdot 0.6 - 0.5 \cdot 0.25 \cdot 166.67] / [0.2 \cdot 350 + 0.3 \cdot 200] = 0.30$$

For 'red' population, for any $0.4 < t < 0.6$

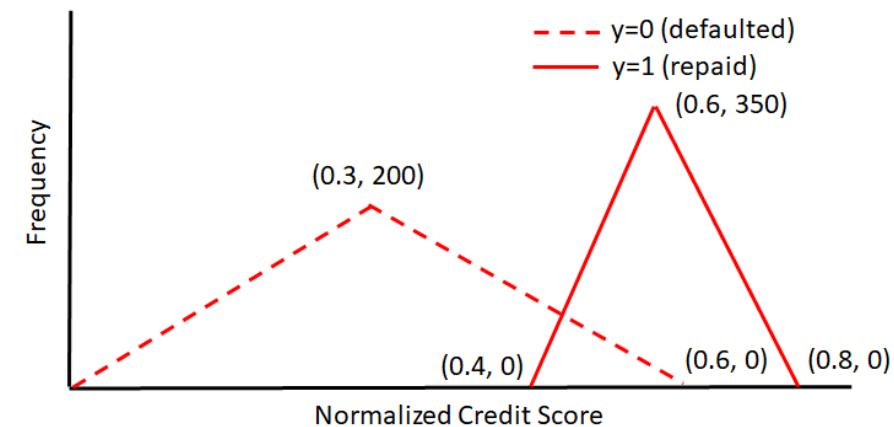
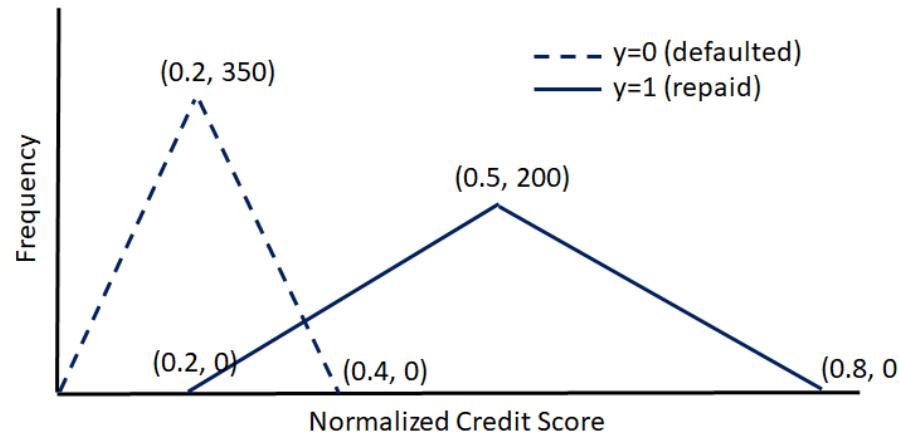
$$P(Y'=1) = [0.2 \cdot 350 + 0.5 \cdot (0.6-t) \cdot 200 / 0.3 \cdot (0.6-t) - 0.5 \cdot (t-0.4) \cdot 350 / 0.2] / [0.3 \cdot 200 + 0.2 \cdot 350]$$

Therefore, for demographic parity,

$$70 + 0.5 \cdot (0.6-t)^2 \cdot 333.33 - 0.5 \cdot (t-0.4) \cdot 350 / 0.2 = 39 \quad \rightarrow t = 0.44463$$

ML EC-3M Problem

C. If threshold $t=0.5$ is chosen for the blue population, calculate the probability $P(y'=1|y=1)$. What threshold value for the red population will ensure equal opportunity?



For the blue population, from the frequency distribution

✓ for $t=0.5$ $P(y'=1|y=1)=0.5$

For the red population, from the distribution figure, for $t=0.6$ $P(y'=1|y=1)=0.5$

Fairness thru Input Manipulation

- **Pre-processing Methods**

- Transform data before ML models learn
- Reweighting, Resampling

- **Post-processing Methods**

- Make predictions from a black-box ML model fair
- Learning to Defer

- **Data Debiasing**

- Adjust data distribution to meet fairness criteria
- Increase/Decrease samples based on criteria
 - Reweighting
 - Adjust importance of each sample in the loss function
 - Resampling
 - Adjust the proportion of samples for each group

$$\underline{P_{obs}(Y = y, A = a)} = \frac{|\{x \in \mathcal{D} | x_Y = y, x_A = a\}|}{|\mathcal{D}|}$$

*Transform data for to
expected distribution*

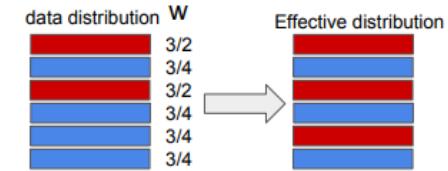
$$\begin{aligned} \underline{P_{exp}(Y = y, A = a)} &= P(Y = y) \cdot P(A = a) \\ &= \frac{|\{x \in \mathcal{D} | x_Y = y\}|}{|\mathcal{D}|} \cdot \frac{|\{x \in \mathcal{D} | x_A = a\}|}{|\mathcal{D}|} \end{aligned}$$

Reweighting

$$W(x) = \frac{P_{exp}(Y = x_y, A = x_a)}{P_{obs}(Y = x_y, A = x_a)}$$

Reweighting Loss Function

$$\mathcal{L} = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} W(x) \cdot \mathcal{L}(\hat{Y}, x_Y)$$



Resampling

$$N_{exp}(y, a) = P_{exp}(y, a) \cdot |\mathcal{D}|$$

Universal Resampling for Demographic Parity

- draw $N_{exp}(D, P)$ samples uniformly from DP (Deprived community with +ve labels)
- draw $N_{exp}(D, N)$ samples uniformly from DN (Deprived community with -ve labels)
- draw $N_{exp}(F, P)$ samples uniformly from FP (Favored community with +ve labels)
- draw $N_{exp}(F, N)$ samples uniformly from FN (Favored community with -ve labels)

Preferential Resampling

- Sample More Data When Confidence of the Predictor Is Low

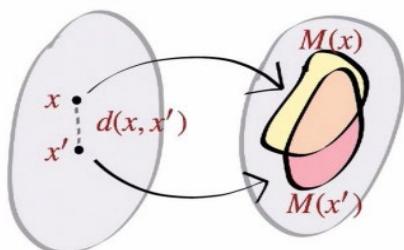
Fairness Through Data/Prediction Manipulations

- Individual Fairness
- Optimized Pre-processing

Individual Fairness

- Predictor M achieves individual fairness under a distance metric d iff

$$M(x_i) \approx M(x_j) | d(x_i, x_j) \approx 0$$



Optimized Preprocessing for Fairness

- Turn resampling process into an optimization based approach

$$\{(D_i, X_i, Y_i)\}_{i=1}^n \xrightarrow{p_{\hat{X}, \hat{Y}|X, Y, D}} \{(D_i, \hat{X}_i, \hat{Y}_i)\}_{i=1}^n$$

- Minimize a utility fn., e.g., KL divergence, such that

$$p_{\hat{X}, \hat{Y}} \iff p_{X, Y}$$

transformed data original data

$$M(x_i) \approx M(x_j) | d(x_i, x_j) \approx 0 \text{ Individual}$$

$$p_{\hat{Y}|D}(y|d) = p_{Y_T}(y) \text{ Group Fairness}$$

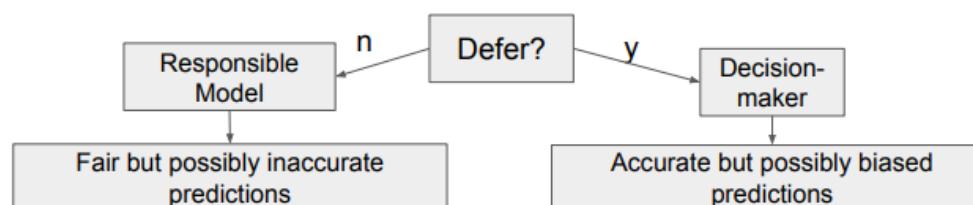
Learning to Defer

Working Together with A Black-box Decision-maker Model

- Decision-maker models (e.g. human) have access to important information that our model does not have
- Decision-maker models might be biased

Performance and Fairness Trade-offs

- Fix the unfair predictions of the decision-maker model
- Defer to the decision-maker the model is uncertain



Fair Causal Reasoning

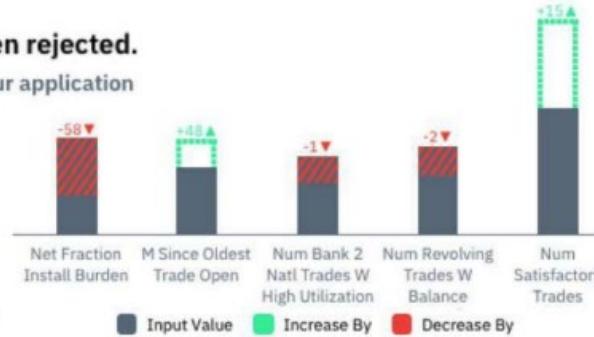
Counterfactual Explanations



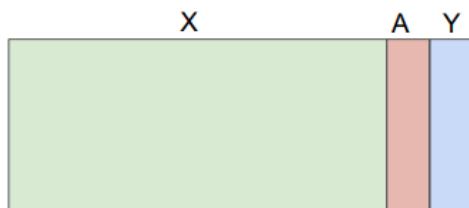
Sorry, your loan application has been rejected.

If instead you had the following values, your application would have been approved:

- MSinceOldestTradeOpen: **161**
- NumSatisfactoryTrades: **36**
- NetFractionInstallBurden: **38**
- NumRevolvingTradesWBalance: **4**
- NumBank2NatlTradesWHighUtilization: **2**



Counterfactual Fairness



Real Examples



Counterfactual Examples

$$\frac{P(\hat{Y}_{A \leftarrow a}(U) = y | X = x, A = a)}{\text{Real Examples}} = \frac{P(\hat{Y}_{A \leftarrow a'}(U) = y | X = x, A = a)}{\text{Counterfactual Examples}}$$

Exact Formulation

$$P(\hat{Y}_{A \leftarrow a}(U) = y | X = x, A = a) = P(\hat{Y}_{A \leftarrow a'}(U) = y | X = x, A = a)$$

ϵ - Approximate Formulation

$$|f(\mathbf{x}_{A \leftarrow a}, a) - f(\mathbf{x}_{A \leftarrow a'}, a')| \leq \epsilon$$

(δ, ϵ) - Approximate Formulation

$$\mathbb{P}(|f(\mathcal{X}_{A \leftarrow a}, a) - f(\mathcal{X}_{A \leftarrow a'}, a')| \leq \epsilon | \mathcal{X} = \mathbf{x}, A = a) \geq 1 - \delta$$

Multi-world Counterfactual Fairness

$$\min_f \frac{1}{n} \sum_{i=1}^n \underbrace{\ell(f(\mathbf{x}_i, a_i), y_i)}_{\text{loss of the data}} + \lambda \sum_{j=1}^m \frac{1}{n} \sum_{i=1}^n \sum_{a' \neq a_i} \mu_j(f, \mathbf{x}_i, a_i, a')$$

world j

counterfactual examples

$$\mu_j(f, \mathbf{x}_i, a_i, a') := \frac{1}{S} \sum_{s=1}^S \max\{0, |f(\mathbf{x}_{i, A^j \leftarrow a_i}^s, a_i) - f(\mathbf{x}_{i, A^j \leftarrow a'}^s, a')| - \epsilon\}$$

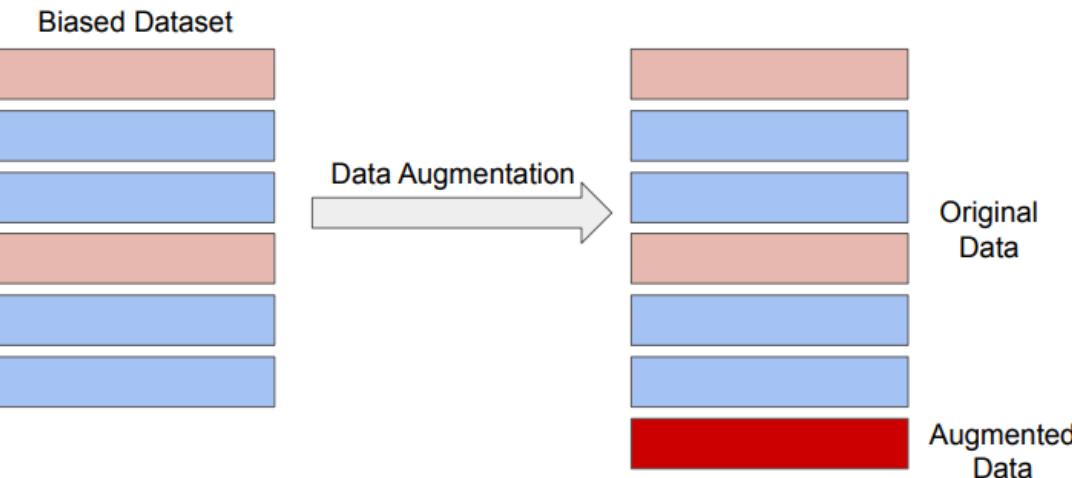
Monte-carlo Samples

ϵ - Approximate Counterfactual Fairness

Fair NLP

Biases in NLP Models

- Denigration
 - The use of culturally or historically derogatory terms
- Stereotyping
 - An over-generalized belief about a particular category of people
- Under-representation
 - The disproportionately low representation of a specific group
- Recognition
 - Algorithms perform different for protected groups because of their inherent characteristics



Gender neutral word embeddings

$$w = [w^{(a)}; w^{(g)}]$$



Debiasing regularizers

$$J = \underbrace{J_G}_{\text{Glove Loss Function}} + \lambda_d \underbrace{J_D}_{\text{Regulate Gender-related Words}} + \lambda_e \underbrace{J_E}_{\text{Regulate All Other Words}}$$

Fair ML Methods

- In-processing Methods
 - Constrain ML models while they learn, e.g.,
 - Prejudice Removing Regularizer,
 - Adversarial Learning
- Pre-processing Methods
 - Transform data before ML models learn
 - e.g., Reweighting, Resampling
- Post-processing Methods
 - Make predictions from a black-box ML model fair in the post-processing stage
 - e.g., Learning to Defer

Prejudice Remover Regularizer

Quantified causes of unfairness



Prejudice

- Unfairness rooted in the dataset

Underestimation

- Model unfairness because the model is not fully converged

Negative Legacy

- Unfairness due to sampling biases

Training Objective

$$-\mathcal{L}(\mathcal{D}; \boldsymbol{\Theta}) + \eta R(\mathcal{D}, \boldsymbol{\Theta}) + \frac{\lambda}{2} \|\boldsymbol{\Theta}\|_2^2$$

Loss of the Model Fairness Regularizer L2 Regularizer

[Kamishima et al, 2012](#)

Prejudice Index (PI)

Recall that Indirect Discrimination Happens When

Prediction is not directly conditioned on sensitive variables S

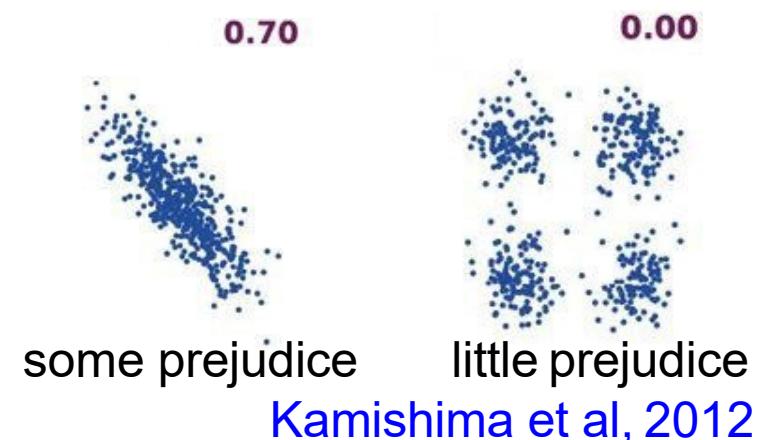
Prediction is indirectly conditioned on S by a variable O that is dependent on S

Prejudice Index (PI)

- Measures the degree of indirect discrimination based on mutual information

$$PI = \sum_{(y,s) \in \mathcal{D}} \hat{Pr}[y, s] \ln \frac{\hat{Pr}[y, s]}{\hat{Pr}[y]\hat{Pr}[s]}$$

↑
prediction model



Normalized Prejudice Index (NPI)

- Prejudice Index (PI)
 - Measures the degree of indirect discrimination based on mutual information
 - Ranges in $[0, +\infty)$

$$\text{PI} = \sum_{(y,s) \in \mathcal{D}} \hat{\Pr}[y, s] \ln \frac{\hat{\Pr}[y, s]}{\hat{\Pr}[y]\hat{\Pr}[s]}$$

- Normalized Prejudice Index (NPI)
 - Normalize PI by the entropy of Y and S
 - Ranges in $[0, 1]$

$$\text{NPI} = \text{PI}/(\sqrt{H(Y)H(S)})$$

[Kamishima et al, 2012](#)

Optimizing PI

- Learning PI

$$PI = \sum_{Y,S} \hat{Pr}[Y, S] \ln \frac{\hat{Pr}[Y, S]}{\hat{Pr}[S]\hat{Pr}[Y]}$$

[Kamishima et al, 2012](#)

Optimizing PI

- Learning PI

$$\text{PI} = \sum_{Y,S} \hat{\Pr}[Y, S] \ln \frac{\hat{\Pr}[Y, S]}{\hat{\Pr}[S]\hat{\Pr}[Y]} = \underbrace{\sum_{X,S} \tilde{\Pr}[X, S] \sum_Y \mathcal{M}[Y|X, S; \boldsymbol{\Theta}] \ln \frac{\hat{\Pr}[Y, S]}{\hat{\Pr}[S]\hat{\Pr}[Y]}}_{\text{double summations}} \quad \text{Expands } \Pr(Y, S) \text{ into } \sum_x \Pr(X, Y, S)$$

↓
 triple summations ↑
 Prediction Model

- Using Logistic Regression Model as the Prediction Model

$$\mathcal{M}[y|\mathbf{x}, s; \boldsymbol{\Theta}] = y\sigma(\mathbf{x}^\top \mathbf{w}_s) + (1 - y)(1 - \sigma(\mathbf{x}^\top \mathbf{w}_s))$$

[Kamishima et al, 2012](#)

Optimizing PI

- Learning PI

$$\begin{aligned}
 \text{PI} &= \sum_{Y,S} \hat{\Pr}[Y, S] \ln \frac{\hat{\Pr}[Y, S]}{\hat{\Pr}[S]\hat{\Pr}[Y]} = \sum_{X,S} \tilde{\Pr}[X, S] \sum_Y \mathcal{M}[Y|X, S; \boldsymbol{\Theta}] \ln \frac{\hat{\Pr}[Y, S]}{\hat{\Pr}[S]\hat{\Pr}[Y]} \\
 &= \sum_{(\mathbf{x}_i, s_i) \in \mathcal{D}} \sum_{y \in \{0,1\}} \mathcal{M}[y|\mathbf{x}_i, s_i; \boldsymbol{\Theta}] \ln \boxed{\frac{\hat{\Pr}[y|s_i]}{\hat{\Pr}[y]}}
 \end{aligned}$$

- Using Logistic Regression Model as the Prediction Model difficult to evaluate

$$\mathcal{M}[y|\mathbf{x}, s; \boldsymbol{\Theta}] = y\sigma(\mathbf{x}^\top \mathbf{w}_s) + (1 - y)(1 - \sigma(\mathbf{x}^\top \mathbf{w}_s))$$

[Kamishima et al, 2012](#)

Optimizing PI

$$\text{PI} = \sum_{(\mathbf{x}_i, s_i) \in \mathcal{D}} \sum_{y \in \{0,1\}} \mathcal{M}[y | \mathbf{x}_i, s_i; \boldsymbol{\Theta}] \ln \frac{\hat{\Pr}[y | s_i]}{\hat{\Pr}[y]}$$

Kamishima et al, 2012

Optimizing PI

$$\text{PI} = \sum_{(\mathbf{x}_i, s_i) \in \mathcal{D}} \sum_{y \in \{0,1\}} \mathcal{M}[y | \mathbf{x}_i, s_i; \boldsymbol{\Theta}] \ln \frac{\hat{\Pr}[y | s_i]}{\hat{\Pr}[y]}$$

$$\hat{\Pr}[y | s] = \int_{\text{dom}(X)} \Pr^*[X | s] \mathcal{M}[y | X, s; \boldsymbol{\Theta}] dX$$

Integrals Are Difficult to Evaluate

[Kamishima et al, 2012](#)

Optimizing PI

$$\text{PI} = \sum_{(\mathbf{x}_i, s_i) \in \mathcal{D}} \sum_{y \in \{0,1\}} \mathcal{M}[y | \mathbf{x}_i, s_i; \boldsymbol{\Theta}] \ln \frac{\hat{\Pr}[y | s_i]}{\hat{\Pr}[y]}$$

$$\begin{aligned} \hat{\Pr}[y | s] &= \int_{\text{dom}(X)} \Pr^*[X | s] \mathcal{M}[y | X, s; \boldsymbol{\Theta}] dX \\ &\approx \frac{\sum_{(\mathbf{x}_i, s_i) \in \mathcal{D} \text{ s.t. } s_i = s} \mathcal{M}[y | \mathbf{x}_i, s; \boldsymbol{\Theta}]}{|\{(\mathbf{x}_i, s_i) \in \mathcal{D} \text{ s.t. } s_i = s\}|} \end{aligned}$$

Approximating integrals by sample means

[Kamishima et al, 2012](#)

Optimizing PI

$$\text{PI} = \sum_{(\mathbf{x}_i, s_i) \in \mathcal{D}} \sum_{y \in \{0,1\}} \mathcal{M}[y | \mathbf{x}_i, s_i; \boldsymbol{\Theta}] \ln \frac{\hat{\Pr}[y | s_i]}{\hat{\Pr}[y]}$$

$$\begin{aligned} \hat{\Pr}[y | s] &= \int_{\text{dom}(X)} \Pr^*[X | s] \mathcal{M}[y | X, s; \boldsymbol{\Theta}] dX \\ &\approx \frac{\sum_{(\mathbf{x}_i, s_i) \in \mathcal{D} \text{ s.t. } s_i = s} \mathcal{M}[y | \mathbf{x}_i, s; \boldsymbol{\Theta}]}{|\{(\mathbf{x}_i, s_i) \in \mathcal{D} \text{ s.t. } s_i = s\}|} \end{aligned}$$

Approximating integrals by sample means

$$\hat{\Pr}[y] \approx \frac{\sum_{(\mathbf{x}_i, s_i) \in \mathcal{D}} \mathcal{M}[y | \mathbf{x}_i, s_i; \boldsymbol{\Theta}]}{|\mathcal{D}|}$$

[Kamishima et al, 2012](#)

Optimizing PI

$$\text{PI} = \sum_{(\mathbf{x}_i, s_i) \in \mathcal{D}} \sum_{y \in \{0,1\}} \mathcal{M}[y | \mathbf{x}_i, s_i; \boldsymbol{\Theta}] \ln \frac{\hat{\Pr}[y | s_i]}{\hat{\Pr}[y]}$$

$$\hat{\Pr}[y | s] \approx \frac{\sum_{(\mathbf{x}_i, s_i) \in \mathcal{D} \text{ s.t. } s_i = s} \mathcal{M}[y | \mathbf{x}_i, s; \boldsymbol{\Theta}]}{|\{(\mathbf{x}_i, s_i) \in \mathcal{D} \text{ s.t. } s_i = s\}|} \quad \hat{\Pr}[y] \approx \frac{\sum_{(\mathbf{x}_i, s_i) \in \mathcal{D}} \mathcal{M}[y | \mathbf{x}_i, s_i; \boldsymbol{\Theta}]}{|\mathcal{D}|}$$

[Kamishima et al, 2012](#)

Putting Things Together

Optimization Target

$$-\mathcal{L}(\mathcal{D}; \boldsymbol{\Theta}) + \eta R(\mathcal{D}, \boldsymbol{\Theta}) + \frac{\lambda}{2} \|\boldsymbol{\Theta}\|_2^2$$

Loss of the Model

Fairness Regularizer

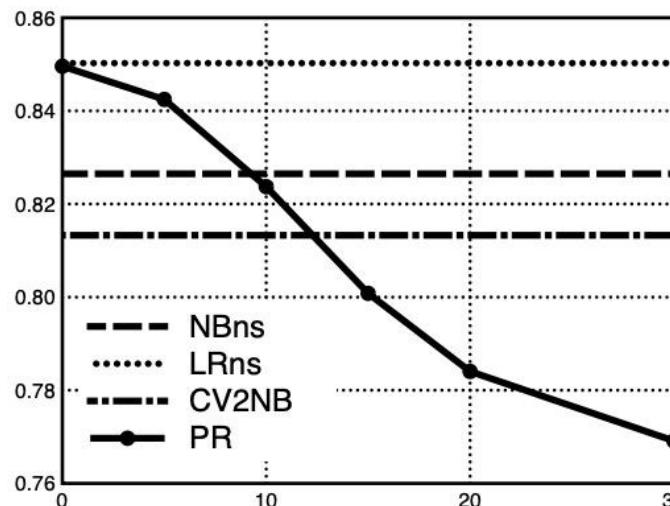
L2 Regularizer

- Fairness Regularizer

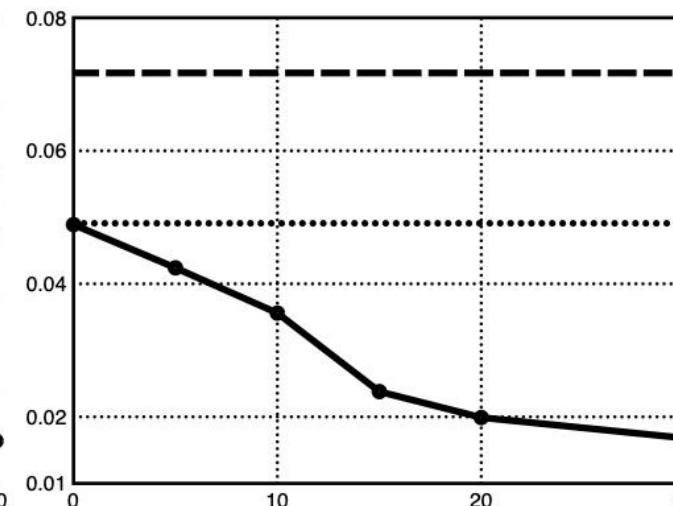
$$\text{PI} = \sum_{(\mathbf{x}_i, s_i) \in \mathcal{D}} \sum_{y \in \{0,1\}} \mathcal{M}[y | \mathbf{x}_i, s_i; \boldsymbol{\Theta}] \ln \frac{\hat{\Pr}[y | s_i]}{\hat{\Pr}[y]}$$

Results

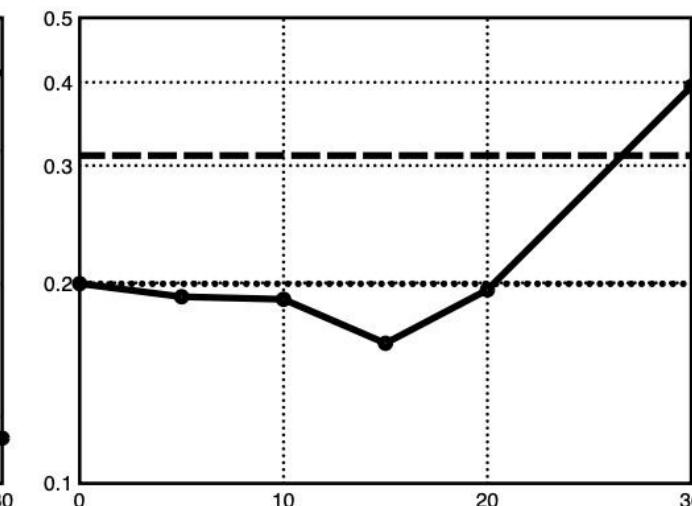
- Changes of Performance With η
 - Model performance decreases (Acc)
 - Discrimination Decreases (NPI)
 - "Fairness Efficiency" (PI/MI) Increases



(a) Acc



(b) NPI



(c) PI/MI

[Kamishima et al, 2012](#)

Results

- Prejudice Prior Sacrifices Model Performance
 - PR has lower Acc (Accuracy)
 - PR has lower NMI (normalized mutual information between labels and predictions)
- Prejudice Prior Makes Model Fair
 - PR has lower NPI

	method	Acc	NMI	NPI	PI/MI
Logistic Regression full fet.	→ LR	0.851	0.267	5.21E-02	2.10E-01
Logistic Regression no sensitive fet.	→ LRns	0.850	0.266	4.91E-02	1.99E-01
Logistic Regression + Prejudice Regularizer	→ PR $\eta=5$	0.842	0.240	4.24E-02	1.91E-01
	→ PR $\eta=15$	0.801	0.158	2.38E-02	1.62E-01
	→ PR $\eta=30$	0.769	0.046	1.68E-02	3.94E-01

η is the weight we put on prejudice regularizers [Kamishima et al, 2012](#)

Results

- PI/MI
 - Prejudice Index / Mutual Information
 - Demonstrates a trade-offs between model fairness and performance
 - Measures the amount of discrimination we eliminate with one unit of performance gain (measured by MI)

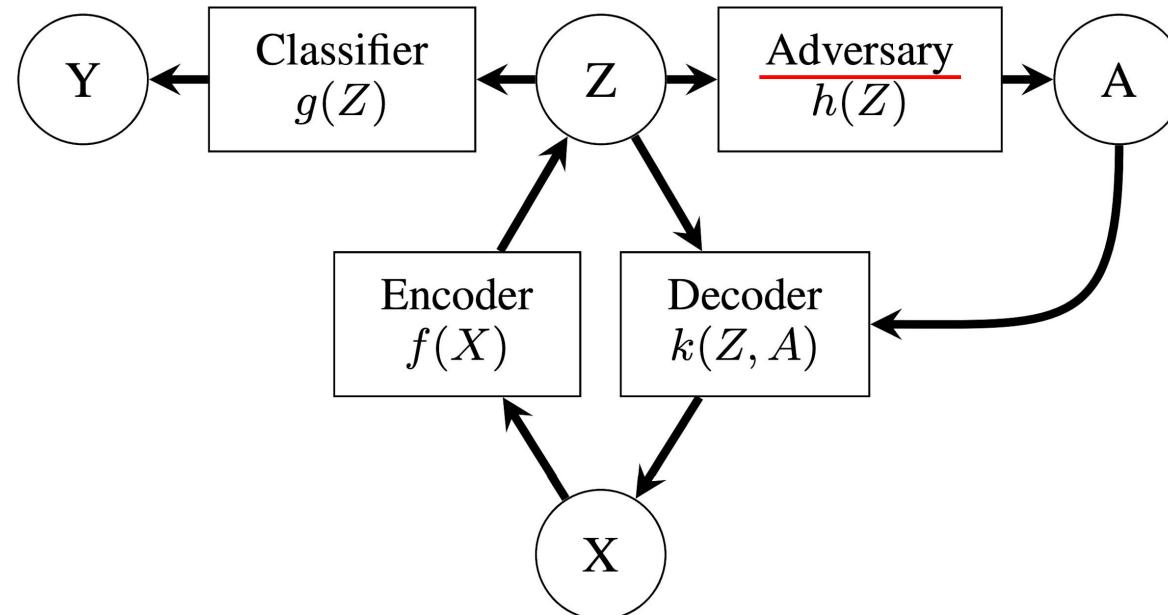
	method	Acc	NMI	NPI	PI/MI
Logistic Regression full fet.	LR	0.851	0.267	5.21E-02	2.10E-01
Logistic Regression no sensitive fet.	LRns	0.850	0.266	4.91E-02	1.99E-01
Logistic Regression + Prejudice Regularizer	PR $\eta=5$	0.842	0.240	4.24E-02	1.91E-01
	PR $\eta=15$	0.801	0.158	2.38E-02	1.62E-01
	PR $\eta=30$	0.769	0.046	1.68E-02	3.94E-01

η - weight put on the prejudice regularizer

[Kamishima et al, 2012](#)

Fairness Through Adversarial Learning

- Adversarial Learning
 - Models are trained using objectives that compete with each other



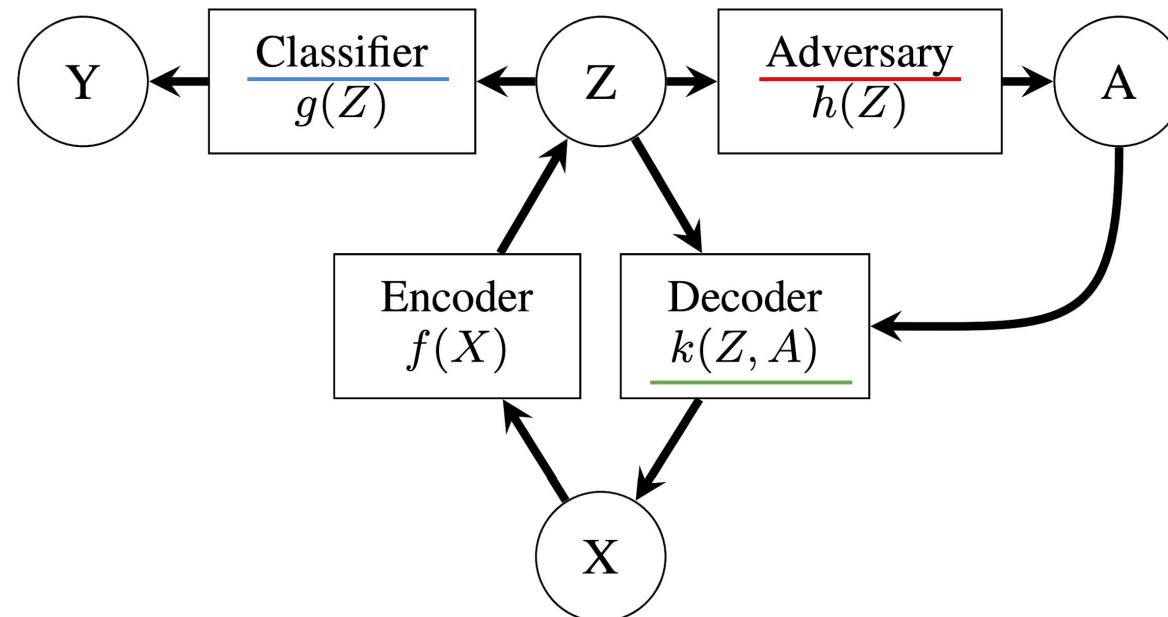
$$\underset{f,g,k}{\text{minimize}} \underset{h}{\text{maximize}} \mathbb{E}_{X,Y,A} [L(f, g, h, k)]$$

[Madras et al, 2018](#)

Fairness Through Adversarial Learning

- Adversarial Learning

$$L(f, g, h, k) = \underline{\alpha L_C(g(f(X, A)), Y)} + \underline{\beta L_{Dec}(k(f(X, A), A), X)} + \underline{\gamma L_{Adv}(h(f(X, A)), A)}$$



$$\underset{f,g,k}{\text{minimize}} \underset{h}{\text{maximize}} \mathbb{E}_{X,Y,A} [L(f, g, h, k)]$$

[Madras et al, 2018](#)

Loss for Learning Fair Representations

- Adversarial Loss for Demographic Parity with Group $\mathcal{D}_0, \mathcal{D}_1$

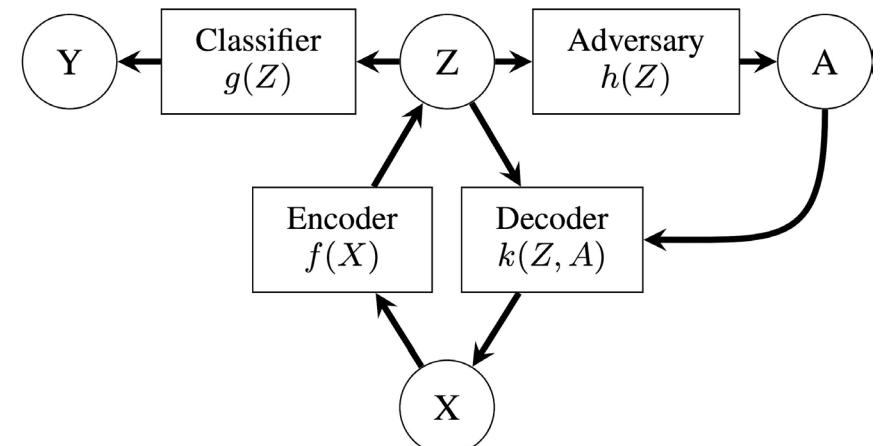
$$L_{Adv}^{DP}(h) = 1 - \sum_{i \in \{0,1\}} \frac{1}{|\mathcal{D}_i|} \sum_{(x,a) \in \mathcal{D}_i} |h(f(x,a)) - a|$$

Demographic Parity: $P(\hat{Y} = 1|A = 1) = P(\hat{Y} = 1|A = 0)$

- Adversarial Loss for Equality of Odds with Group $\mathcal{D}_i^j = \{(x, \bar{y}, \bar{a}) \in \mathcal{D} | a = i, y = j\}$

$$L_{Adv}^{EO}(h) = 2 - \sum_{(i,j) \in \{0,1\}^2} \frac{1}{|\mathcal{D}_i^j|} \sum_{(x,a) \in \mathcal{D}_i^j} |h(f(x,a)) - a|$$

Equality of Odds: $P(\hat{Y} = 1|A = 0, Y) = P(\hat{Y} = 1|A = 1, Y)$



[Madras et al, 2018](#)

Discrimination Measures for Representations

$$\mathcal{Z}_1 = p(Z|A = 1) \quad \mathcal{Z}_0 = p(Z|A = 0) \quad \mathcal{Z}_a^y = p(Z|A = a, Y = y)$$

- Demographic Parity

$$\Delta_{DP}(g) \triangleq d_g(\mathcal{Z}_0, \mathcal{Z}_1) = |\mathbb{E}_{\mathcal{Z}_0}[g] - \mathbb{E}_{\mathcal{Z}_1}[g]|$$

Demographic Parity: $P(\hat{Y} = 1|A = 1) = P(\hat{Y} = 1|A = 0)$

- Equality of Odds

$$\Delta_{EO}(g) \triangleq |\mathbb{E}_{\mathcal{Z}_0^0}[g] - \mathbb{E}_{\mathcal{Z}_1^0}[g]| + |\mathbb{E}_{\mathcal{Z}_0^1}[1 - g] - \mathbb{E}_{\mathcal{Z}_1^1}[1 - g]|$$

Equality of Odds: $P(\hat{Y} = 1|A = 0, Y) = P(\hat{Y} = 1|A = 1, Y)$

- Equality of Opportunities

$$\Delta_{EOpp}(g) \triangleq |\mathbb{E}_{\mathcal{Z}_0^0}[g] - \mathbb{E}_{\mathcal{Z}_1^0}[g]|$$

Equality of Opportunity: $P(\hat{Y} = 1|A = 0, Y = 1) = P(\hat{Y} = 1|A = 1, Y = 1)$

Comparisons: Regularization and Adversarial Learning

	Prejudice Removing Regularizer	Adversarial Learning
Pros	Minimal modifications to training procedure	Transferable representations
		Can be applied to many different fairness criteria
Cons	Can only be applied to Demographic Parity	Adversarial loss can be difficult to train

Recap

Demographic Parity



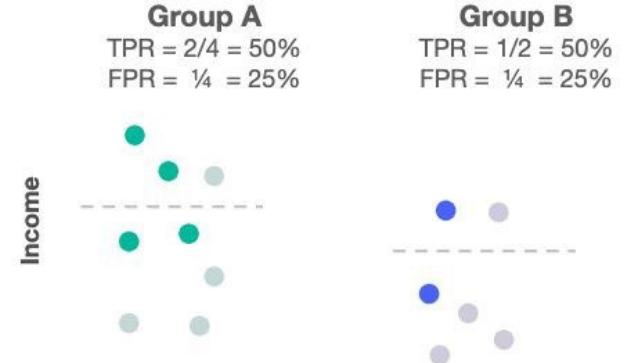
$$P(\hat{Y} = 1|A = 1) = P(\hat{Y} = 1|A = 0)$$

Equal Opportunity



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

Equal Odds



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

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

Fair Data Manipulation

- Biased Data
 - The presence of data that belongs to the underrepresented groups leads to data biases
 - One of the main sources of ML discriminations
- Data Debiasing
 - Adjust the distribution of the data to meet fairness criteria
 - Increase/Decrease samples based on criteria
- Reweighting
 - Adjust the importance of each sample in the loss function during training
- Resampling
 - Adjust the proportion of samples for each group

Expected Distribution of Fair Data

- Expected Data Distribution

$$P(Y) = P(Y|A = 1) = P(Y|A = 0)$$

which leads to $Y \perp\!\!\!\perp A$

- Recall Demographic Parity

$$P(\hat{Y} = 1|A = 1) = P(\hat{Y} = 1|A = 0)$$

[Kamiran et al, 2012](#)

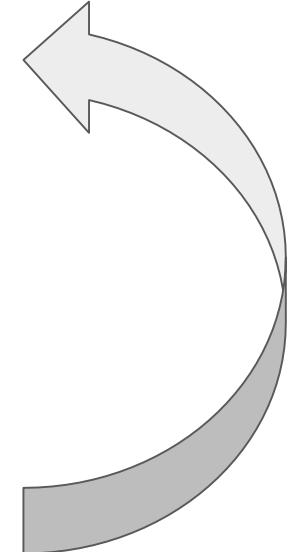
Expected Distribution of Fair Data

- The Expected Joint Distribution Under $Y \perp\!\!\!\perp A$

$$\begin{aligned}
 P_{\underline{\text{exp}}}(Y = y, A = a) &= P(Y = y) \cdot P(A = a) \\
 &= \frac{|\{x \in \mathcal{D} | x_Y = y\}|}{|\mathcal{D}|} \cdot \frac{|\{x \in \mathcal{D} | x_A = a\}|}{|\mathcal{D}|}
 \end{aligned}$$

- Our Observed Joint Distribution

$$P_{\underline{\text{obs}}}(Y = y, A = a) = \frac{|\{x \in \mathcal{D} | x_Y = y, x_A = a\}|}{|\mathcal{D}|}$$



Transform Data to
Expected Distribution

[Kamiran et al, 2012](#)

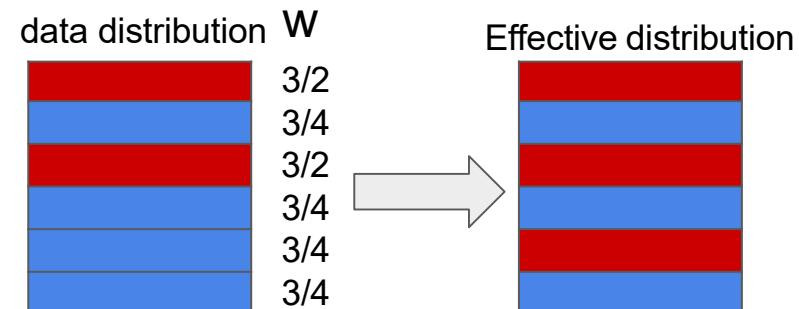
Reweighting

- Sample Weight for x
 - Goal: adjust our data to a distribution that leads to $Y \perp\!\!\!\perp A$, or Demographic Parity
 - $W(x) = 1$, we have achieved $Y \perp\!\!\!\perp A$ and Demographic Parity
 - $W(x) > 1$, increase the weight of sample x in training
 - $W(x) < 1$, decrease the weight of sample x in training

$$W(x) = \frac{P_{exp}(Y = x_y, A = x_a)}{P_{obs}(Y = x_y, A = x_a)}$$

- Reweighting Loss Function

$$\mathcal{L} = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} W(x) \cdot \mathcal{L}(\hat{Y}, x_Y)$$



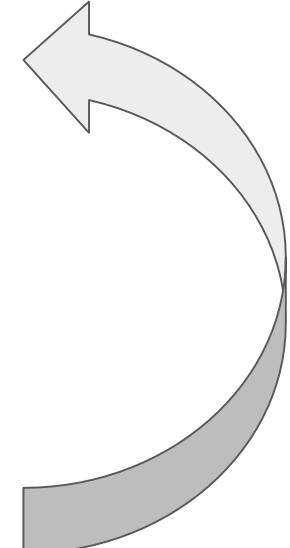
Expected Distribution of Fair Data

- The Expected Joint Distribution Under $Y \perp\!\!\!\perp A$

$$\begin{aligned}
 P_{\underline{\text{exp}}}(Y = y, A = a) &= P(Y = y) \cdot P(A = a) \\
 &= \frac{|\{x \in \mathcal{D} | x_Y = y\}|}{|\mathcal{D}|} \cdot \frac{|\{x \in \mathcal{D} | x_A = a\}|}{|\mathcal{D}|}
 \end{aligned}$$

- Our Observed Joint Distribution

$$P_{\underline{\text{obs}}}(Y = y, A = a) = \frac{|\{x \in \mathcal{D} | x_Y = y, x_A = a\}|}{|\mathcal{D}|}$$



Transform Data to
Expected Distribution

[Kamiran et al, 2012](#)

Practice Question

- Calculate $W(x_3)$, $A = \{\text{Sex}\}$, $Y = \{\text{Class}\}$

Sex	Highest degree	Job type	Class
M	H. school	Board	+
M	Univ.	Board	+
M	H. school	Board	+
M	H. school	Healthcare	+
M	Univ.	Healthcare	-
F	Univ.	Education	-
F	H. school	Education	-
F	None	Healthcare	+
F	Univ.	Education	-
F	H. school	Board	+

[Kamiran et al, 2012](#)

Practice Question

- $W(x_3)$
 - $A_3 = M$
 - $Y_3 = +$
- Expected Distribution
 - $P(A = M) = 0.5$
 - $P(Y = +) = 0.6$
 - $P_{\text{exp}}(A = M, Y = +) = 0.3$
- Observed Distribution
 - $P_{\text{obs}}(A = M, Y = +) = 0.4$
- Sample Weight
 - $W(x_3) = 0.3/0.4 = 0.75$

$A = \{\text{Sex}\}, Y = \{\text{Class}\}$

Sex	Highest degree	Job type	Class
M	H. school	Board	+
M	Univ.	Board	+
M	H. school	Board	+
M	H. school	Healthcare	+
M	Univ.	Healthcare	-
F	Univ.	Education	-
F	H. school	Education	-
F	None	Healthcare	+
F	Univ.	Education	-
F	H. school	Board	+

[Kamiran et al, 2012](#)

Quiz

- Calculate $W(x_6)$, $A = \{\text{Sex}\}$, $Y = \{\text{Class}\}$

Sex	Highest degree	Job type	Class
M	H. school	Board	+
M	Univ.	Board	+
M	H. school	Board	+
M	H. school	Healthcare	+
M	Univ.	Healthcare	-
F	Univ.	Education	-
F	H. school	Education	-
F	None	Healthcare	+
F	Univ.	Education	-
F	H. school	Board	+

[Kamiran et al, 2012](#)

Answer

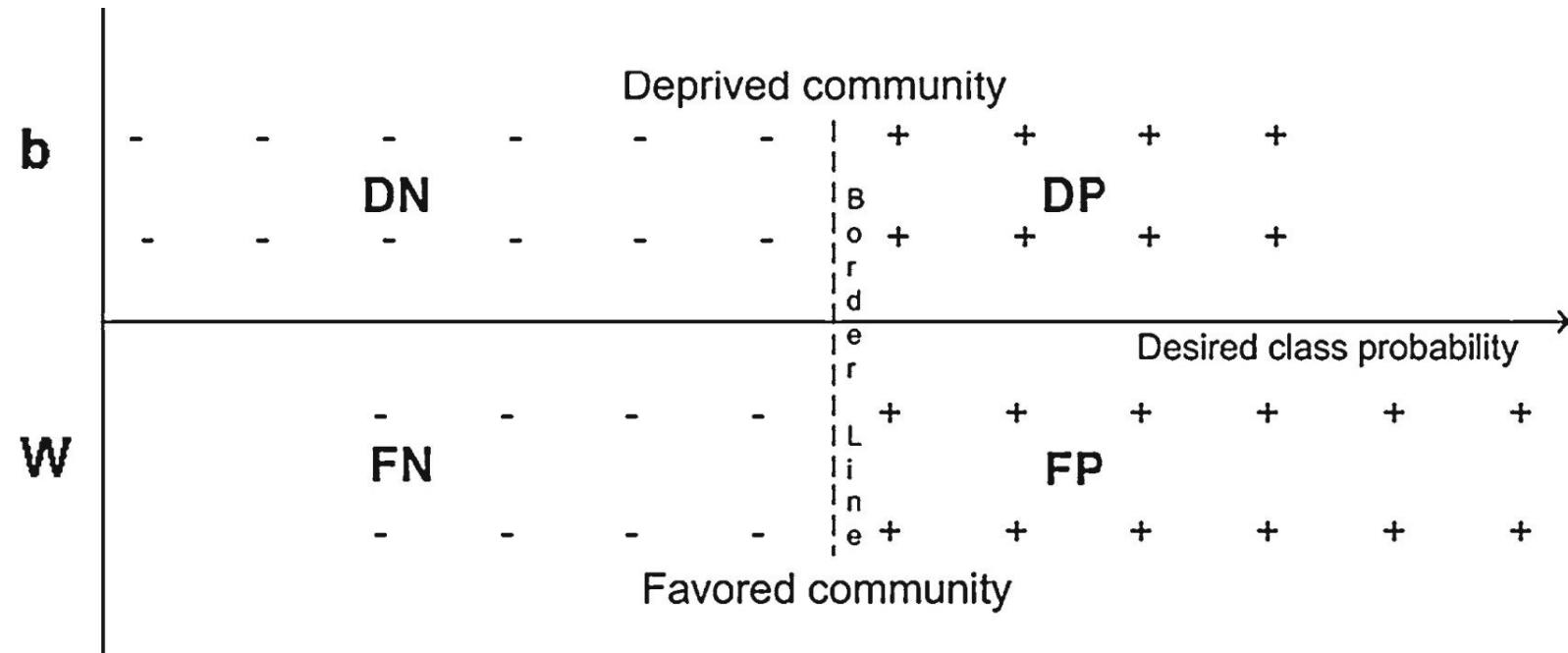
- $W(x_6)$
 - $A_6 = F$
 - $Y_6 = -$
- Expected Distribution
 - $P(A = F) = 0.5$
 - $P(Y = -) = 0.4$
 - $P_{\text{exp}}(A = F, Y = -) = 0.2$
- Observed Distribution
 - $P_{\text{obs}}(A = F, Y = -) = 0.3$
- Sample Weight
 - $W(x_6) = 0.2/0.3 = 0.67$

Sex	Highest degree	Job type	Class
M	H. school	Board	+
M	Univ.	Board	+
M	H. school	Board	+
M	H. school	Healthcare	+
M	Univ.	Healthcare	-
F	Univ.	Education	-
F	H. school	Education	-
F	None	Healthcare	+
F	Univ.	Education	-
F	H. school	Board	+

[Kamiran et al, 2012](#)

Resampling

- Resample the Dataset Based on the Expected Joint Probability



[Kamiran et al, 2012](#)

Expected Number of Samples

- Expected Number of Samples for the Category (y, a)

$$N_{exp}(y, a) = P_{exp}(y, a) \cdot |\mathcal{D}|$$

- Also Note

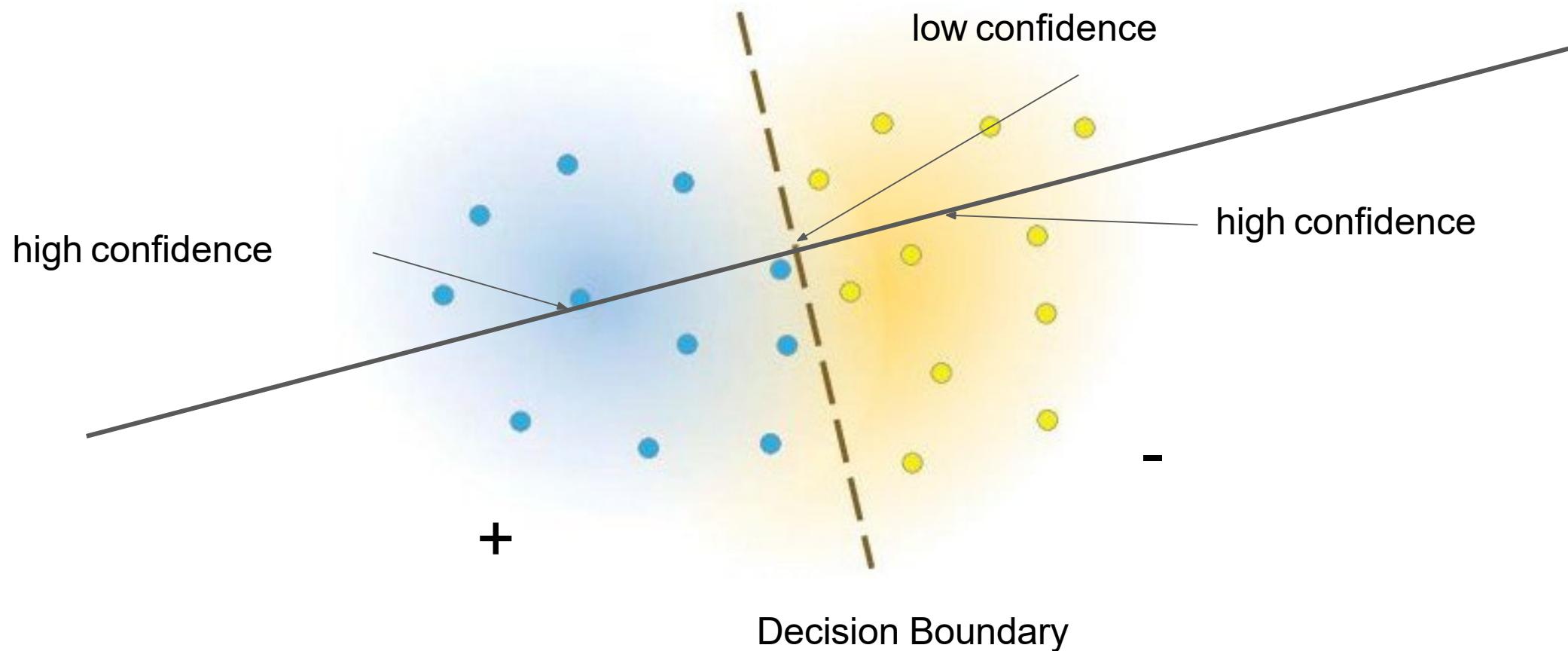
$$\sum_{y,a} N_{exp} = \sum_{y,a} P_{exp}(y, a) \cdot |\mathcal{D}| = |\mathcal{D}|$$

Universal Resampling (US)

- Resampling Based on the Expected Probabilities to Meet Demographic Parity
 - DP (Deprived community with Positive class labels)
 - draw $N_{\text{exp}}(D, P)$ samples uniformly from DP
 - DN (Deprived community with Negative class labels)
 - draw $N_{\text{exp}}(D, N)$ samples uniformly from DN
 - FP (Favored community with Positive class labels)
 - draw $N_{\text{exp}}(F, P)$ samples uniformly from FP
 - FN (Favored community with Negative class labels)
 - draw $N_{\text{exp}}(F, N)$ samples uniformly from FN

Preferential Sampling (PS)

- Sample More Data When Confidence of the Predictor Is Low



[Kamiran et al, 2012](#)

Bias Measures

- Measure prediction biases by comparing the favorable outcomes given to group 1 with that to group 0

$$Bias(\hat{Y}) = P(\hat{Y} = 1|A = 1) - P(\hat{Y} = 1|A = 0)$$

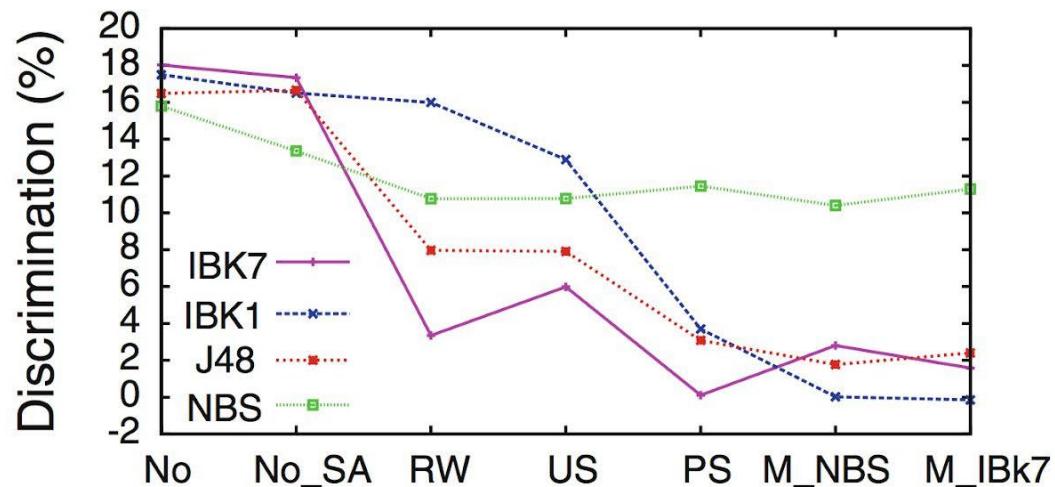
Demographic Parity

$$P(\hat{Y} = 1|A = 1) = P(\hat{Y} = 1|A = 0)$$

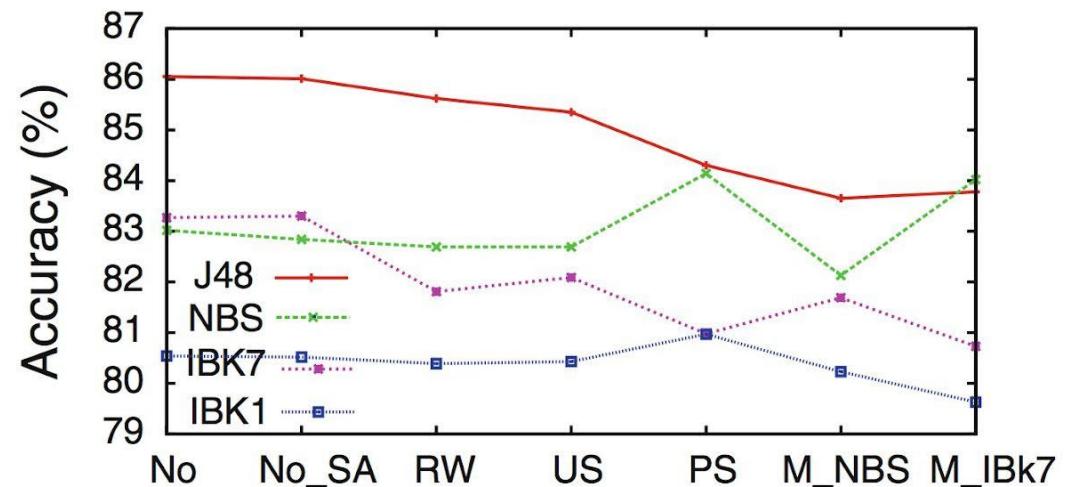
[Kamiran et al, 2012](#)

Adult Income Dataset

No - No pre-processing, No-SA - No Sex Attribute, RW - Reweighting
 US - Universal Sampling, PS - Preferential Sampling
 M_* - “massaged” input data (refer to the paper)



J48 - decision tree
 NBS - Naive Bayes



IBK1- 1 nearest neighbor
 IBK7 - 7 nearest neighbor

[Kamiran et al, 2012](#)

Optimized Pre-Processing for Fairness

- Can We Automate the Resampling Process?
 - Turn the manual process into an optimization based approach
 - Include more criteria than Demographic Fairness
 - Allow transformations of data
- Optimized Pre-Processing
 - Given sensitive feature D, learn a probabilistic mapping $p_{\hat{X}, \hat{Y}|X, Y, D}$ that transfers
 - Satisfies three constraints

$$\{(D_i, X_i, Y_i)\}_{i=1}^n \xrightarrow{p_{\hat{X}, \hat{Y}|X, Y, D}} \{(D_i, \hat{X}_i, \hat{Y}_i)\}_{i=1}^n$$

[Calmon et al, 2017](#)

Constraint 1: Utility Preservations

- A Utility Function to Preserve the Joint Probability
 - e.g. KL Divergence

$$p_{\hat{X}, \hat{Y}} \quad \longleftrightarrow \quad p_{X, Y}$$

transformed data original data

[Calmon et al. 2017](#)

Constraint 2: Discrimination Control

- Constrain the dependency of the target variable y given sensitive feature d to match target $p_{Y_T}(y)$
 - J - distance measure $J(p, q) = \left| \frac{p}{q} - 1 \right|$
 - $\epsilon_{y,d}$ - a small number used as our tolerance

$$J \left(p_{\hat{Y}|D}(y|d), p_{Y_T}(y) \right) \leq \epsilon_{y,d} \quad \forall d \in \mathcal{D}, y \in \{0, 1\}$$

When $p_{\hat{Y}|D}(y|d) = p_{Y_T}(y)$, we achieve Demographic Parity

[Calmon et al. 2017](#)

Constraint 3: Distortion Control

- An Implementation of the Individual Fairness

$$M(x_i) \approx M(x_j) | d(x_i, x_j) \approx 0$$

- The Mapped Sample \hat{X}, \hat{Y} Has to Stay Close to the Original Sample x, y
 - $c_{d,x,y}$ - tolerance
 - δ - a similarity function
 - 1 - very different
 - 0 - very similar

$$\Pr \left(\delta((x, y), (\hat{X}, \hat{Y})) = 1 \mid D = d, X = x, Y = y \right) \leq c_{d,x,y}$$

[Calmon et al. 2017](#)

Putting Things Together

$$\begin{aligned}
 & \min_{p_{\hat{X}, \hat{Y}|X, Y, D}} \Delta(p_{\hat{X}, \hat{Y}}, p_{X, Y}) \\
 \text{s.t. } & J(p_{\hat{Y}|D}(y|d), p_{Y_T}(y)) \leq \epsilon_{y,d} \text{ and} \\
 & \mathbb{E} \left[\delta((x, y), (\hat{X}, \hat{Y})) \mid D = d, X = x, Y = y \right] \leq c_{d,x,y}
 \end{aligned}$$

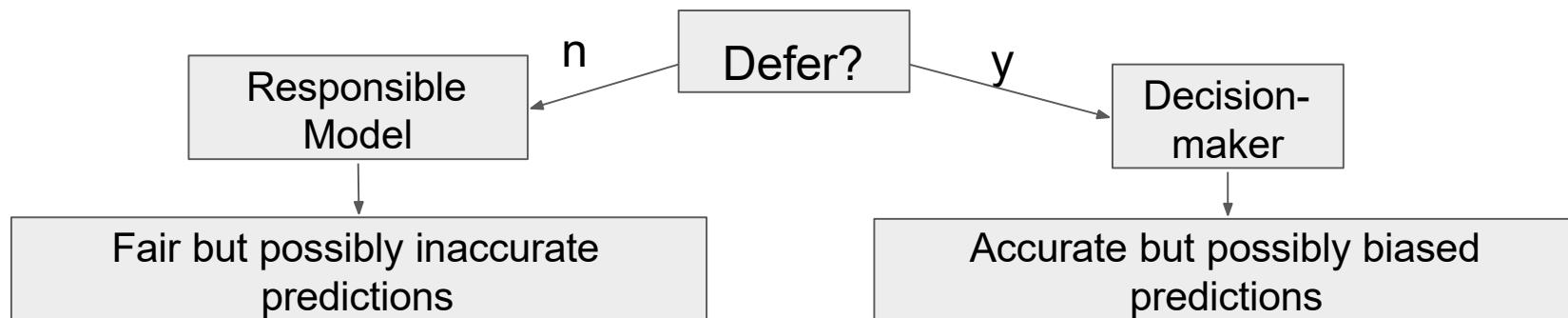
Utility
 Discrimination control
 group fairness

↑
 Distortion Control
 Individual fairness

[Calmon et al. 2017](#)

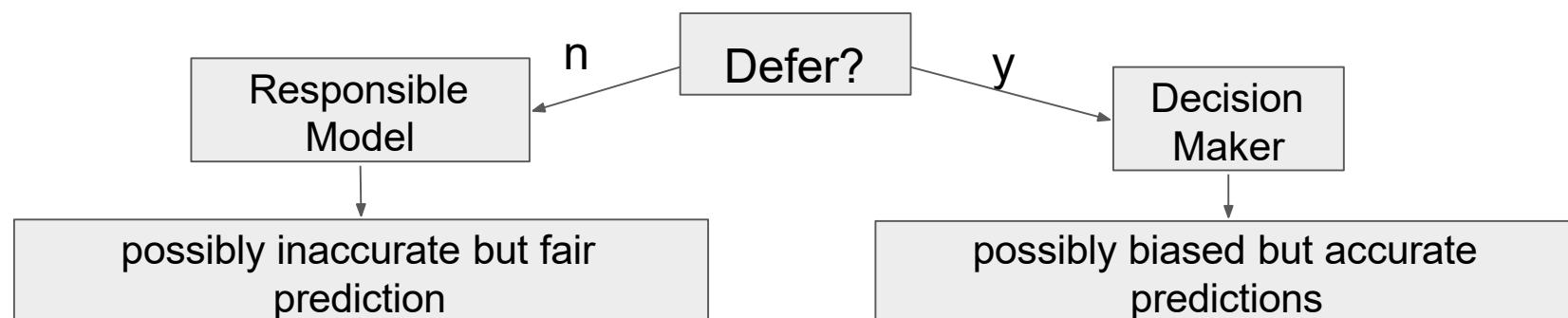
Learning to Defer

- Working Together with A Black-box Decision-maker Model
 - Decision-maker models (e.g. human) have access to important information that our model does not has
 - Decision-maker models might be biased
- Performance and Fairness Trade-offs
 - Fix the unfair predictions of the decision-maker model
 - Defer to the decision-maker the model is uncertain

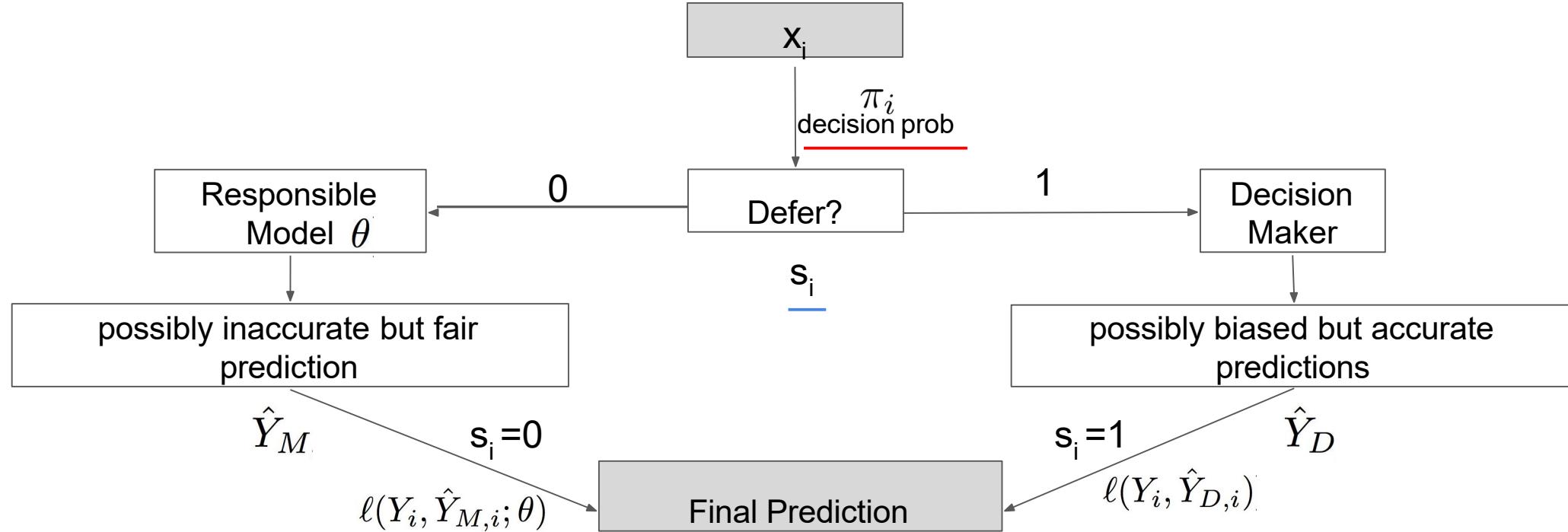


Learning to Defer

- Decision-maker Model
 - Considered as a black-box model
 - No fine-tuning, no access to its training data
- Responsible Model
 - Have access to additional data
 - Stick to fairness constraints



Training the Defer Model

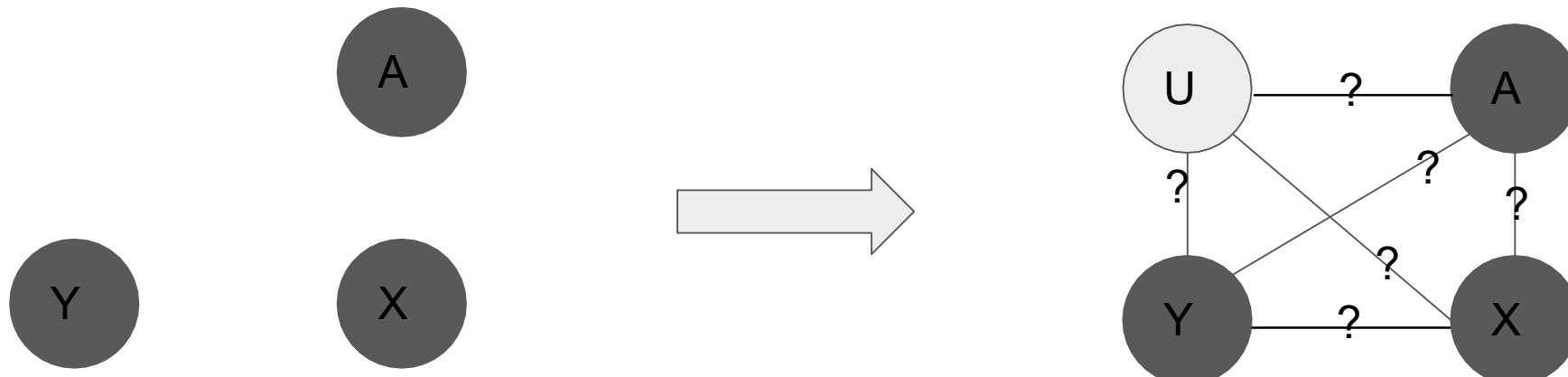


$$\mathcal{L}_{defer}(Y, \hat{Y}_M, \hat{Y}_D, \pi; \theta) = \sum_i \mathbb{E}_{s_i \sim Ber(\pi_i)} [(1 - s_i)\ell(Y_i, \hat{Y}_{M,i}; \theta) + s_i\ell(Y_i, \hat{Y}_{D,i})] + \alpha_{fair} \mathcal{R}(Y, \hat{Y}_M, \hat{Y}_D, s)$$

Fair regularizer

[Madras et al. 2018](#)

Counterfactual Fairness



$$\frac{P(\hat{Y}_{A \leftarrow a}(U) = y \mid X = x, A = a)}{P(\hat{Y}_{A \leftarrow a'}(U) = y \mid X = x, A = a)}$$

Real Examples

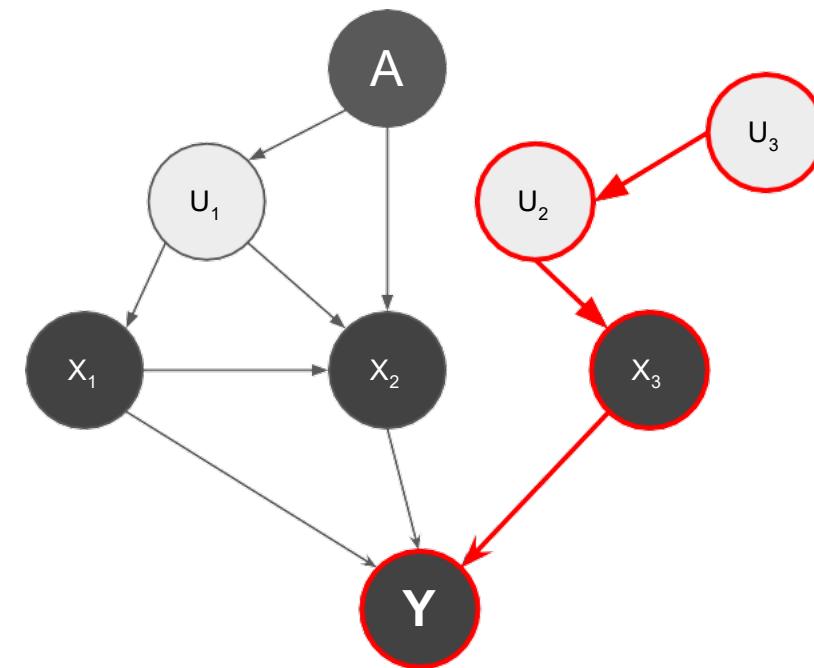
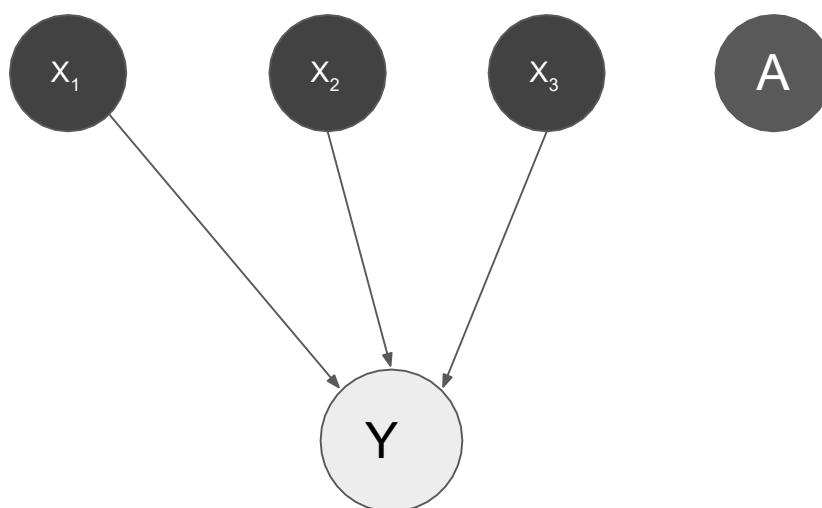
Intervention on $A \leftarrow a$

Counterfactual Examples

Intervention on $A \leftarrow a'$

Counterfactual Fairness

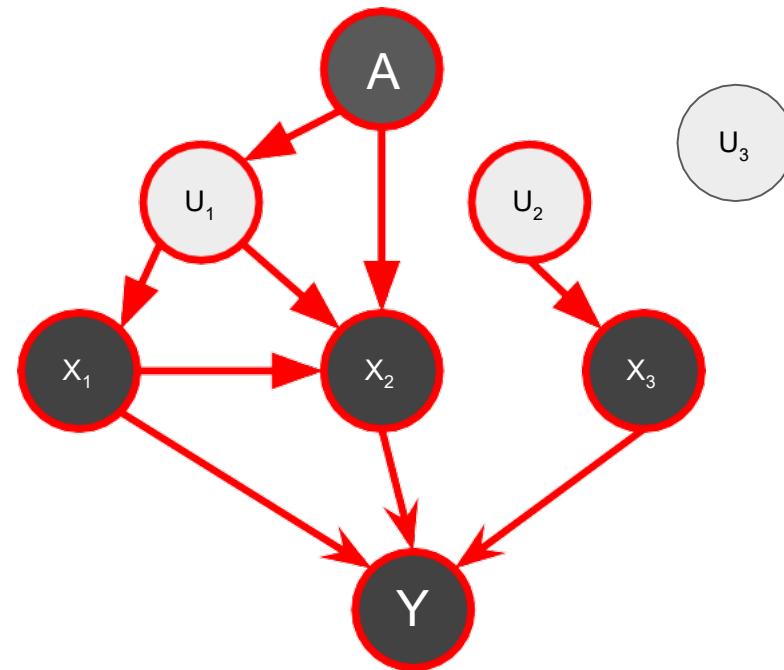
- Level 1
 - Build predictors using only the observable non-descendants of A



Fairness Through Unawareness

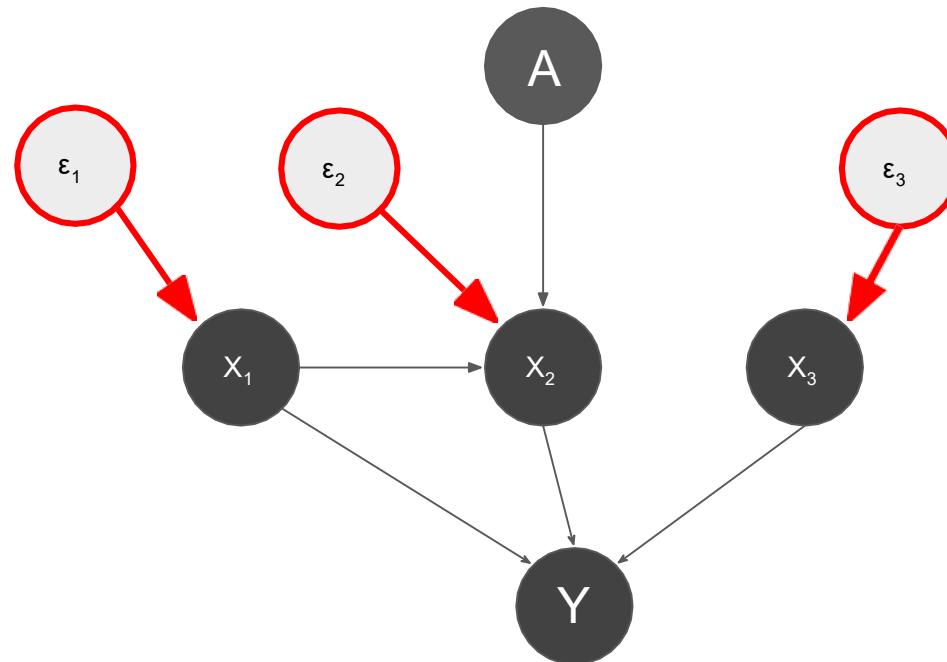
Counterfactual Fairness

- Level 2
 - Build Predictors using the parents of the observable variables



Counterfactual Fairness

- Level 3
 - Build Predictors by adding independent error terms

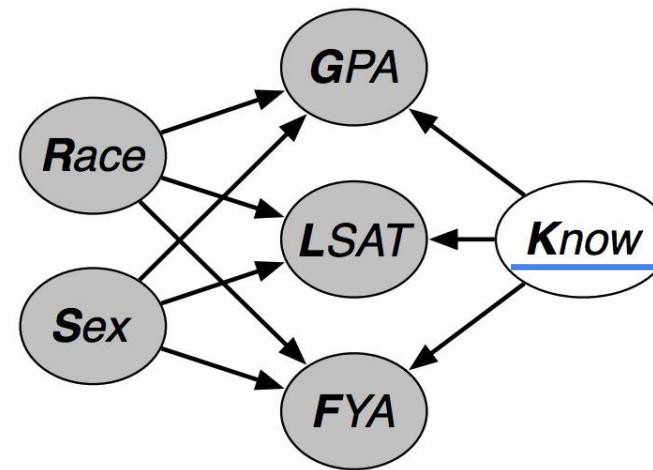


Law School Success Dataset

- Conducted by Law School Admission Council in US
 - 21,790 law students
 - Entrance exam scores (LSAT)
 - Grade-point average (GPA) collected prior to law school
 - Prediction Y = first year average grade (FYA)
 - Protected features = {Gender, Race}

Level 2 Counterfactual Fairness

- Build Predictors using the parents of the observable variables



$$\mathbf{K} \sim \mathcal{N}(0, 1)$$

$$\text{FYA} \sim \mathcal{N}(w_F^K K + w_F^R R + w_F^S S, 1) \quad \text{GPA} \sim \mathcal{N}(b_G + w_G^K K + w_G^R R + w_G^S S, \sigma_G)$$

Parameters

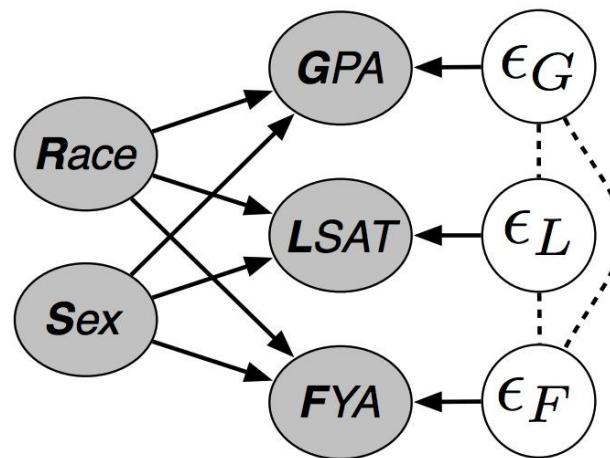
Gaussian Dist.

$$\text{LSAT} \sim \text{Poisson}(\exp(b_L + w_L^K K + w_L^R R + w_L^S S))$$

[Kusner et al, 2018](#)

Level 3 Counterfactual Fairness

- Build Predictors by adding independent error terms



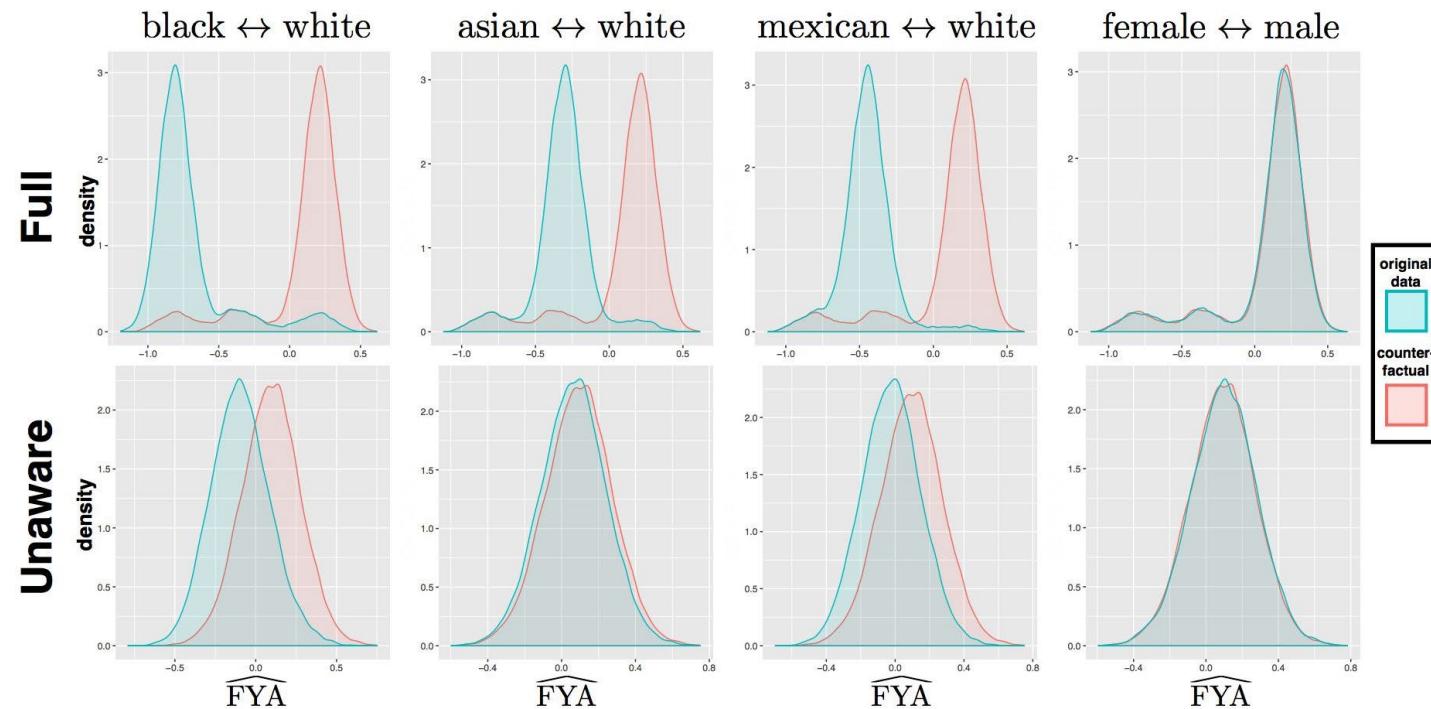
$$\text{GPA} = b_G + w_G^R R + w_G^S S + \epsilon_G, \quad \epsilon_G \sim p(\epsilon_G)$$

$$\text{LSAT} = b_L + w_L^R R + w_L^S S + \epsilon_L, \quad \epsilon_L \sim p(\epsilon_L)$$

$$\text{FYA} = b_F + w_F^R R + w_F^S S + \epsilon_F, \quad \epsilon_F \sim p(\epsilon_F)$$

[Kusner et al, 2018](#)

Baselines



full - using all features

unaware - fairness through unawareness

Kusner et al, 2018

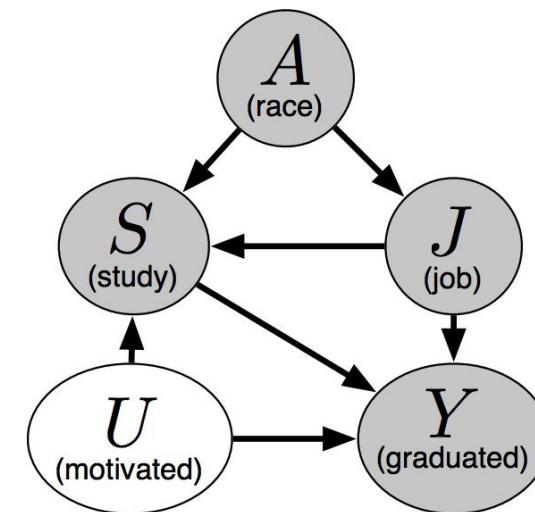
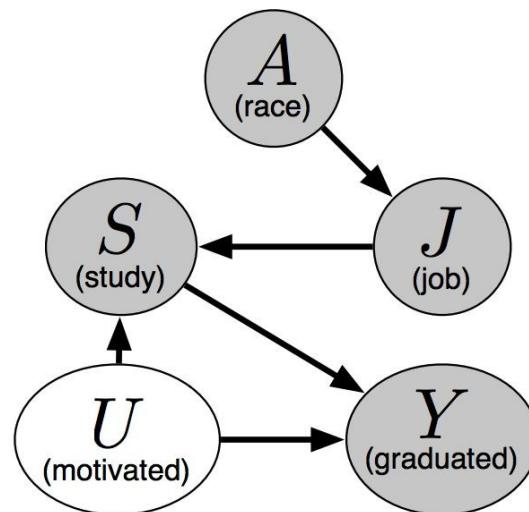
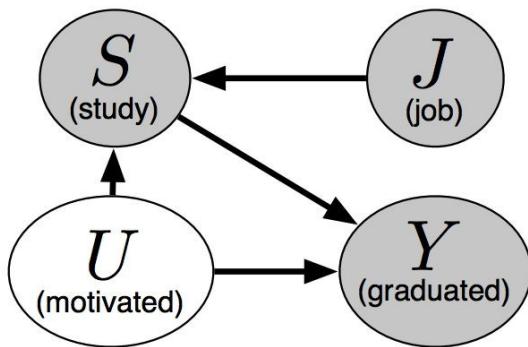
Results

	Baseline	Baseline	Level 2	Level 3
	Full	Unaware	Fair K	Fair Add
RMSE	0.873	0.894	0.929	0.918

[Kusner et al, 2018](#)

Multiple Causal Graphs

- Whether a student can graduate on time



Alternative Definitions of Counterfactual Fairness

Exact Formulation

$$P(\hat{Y}_{A \leftarrow a}(U) = y \mid X = x, A = a) = P(\hat{Y}_{A \leftarrow a'}(U) = y \mid X = x, A = a)$$

ϵ - Approximate Formulation

$$\left| f(\mathbf{x}_{A \leftarrow a}, a) - f(\mathbf{x}_{A \leftarrow a'}, a') \right| \leq \epsilon$$

(δ, ϵ) - Approximate Formulation

$$\mathbb{P}(\left| f(\mathcal{X}_{A \leftarrow a}, a) - f(\mathcal{X}_{A \leftarrow a'}, a') \right| \leq \epsilon \mid \mathcal{X} = \mathbf{x}, A = a) \geq 1 - \delta$$

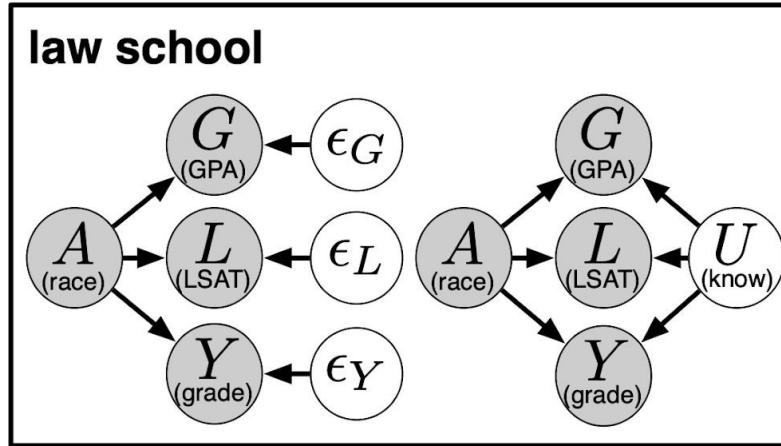
Multi-world Counterfactual Fairness

$$\min_f \frac{1}{n} \sum_{i=1}^n \underbrace{\ell(f(\mathbf{x}_i, a_i), y_i)}_{\text{loss of the data}} + \lambda \sum_{j=1}^m \frac{1}{n} \sum_{i=1}^n \sum_{a' \neq a_i} \underbrace{\mu_j(f, \mathbf{x}_i, a_i, a')}_{\substack{\text{world j} \\ \text{counterfactual examples}}}$$

$$\mu_j(f, \mathbf{x}_i, a_i, a') := \frac{1}{S} \sum_{s=1}^S \max\{0, |f(\mathbf{x}_{i, A^j \leftarrow a_i}^s, a_i) - f(\mathbf{x}_{i, A^j \leftarrow a'}^s, a')| - \epsilon\}$$

Monte-carlo Samples ε - Approximate Counterfactual Fairness

Law Graduate School



$$G = b_G + w_G^A A + \epsilon_G$$

$$L = b_L + w_L^A A + \epsilon_L$$

$$Y = b_Y + w_Y^A A + \epsilon_Y$$

$$\epsilon_G, \epsilon_L, \epsilon_Y \sim \mathcal{N}(0, 1)$$

L3 Method

$$G \sim \mathcal{N}(b_G + w_G^A A + w_G^U U, \sigma_G)$$

$$L \sim \text{Poisson}(\exp(b_L + w_L^A A + w_L^U U))$$

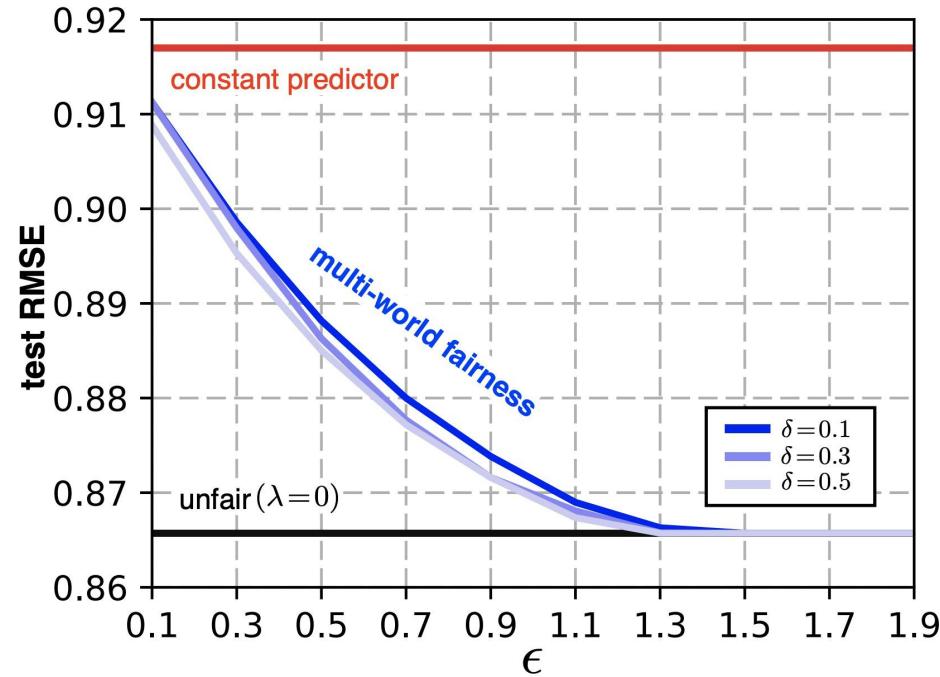
$$Y \sim \mathcal{N}(w_Y^A A + w_Y^U U, 1)$$

$$U \sim \mathcal{N}(0, 1)$$

[Russell et al, 2017](#)

L2 Method

Results



$$|f(\mathbf{x}_{A \leftarrow a}, a) - f(\mathbf{x}_{A \leftarrow a'}, a')| \leq \epsilon$$

[Russell et al, 2017](#)

Equalized Counterfactual Odds

Equality of Odds

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

Counterfactual Fairness

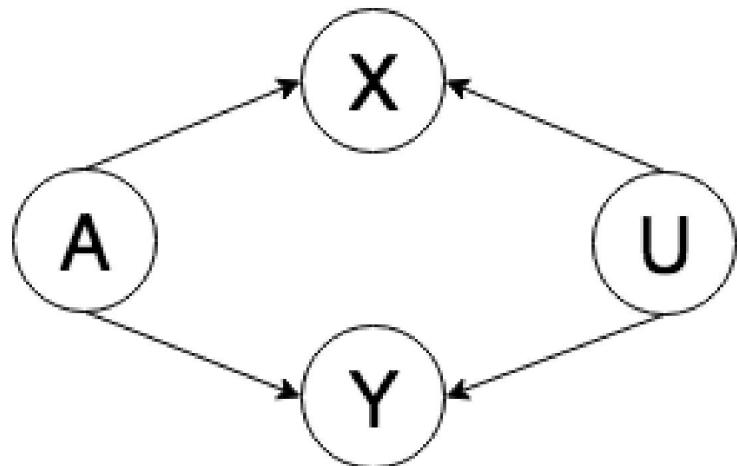
$$\underline{P(\hat{Y}_{A \leftarrow a}(U) = y | X = x, A = a)} = \underline{P(\hat{Y}_{A \leftarrow a'}(U) = y | X = x, A = a)}$$

Equalized Counterfactual Odds

$$\underline{p(\hat{Y}_{A \leftarrow a}(U) | X = x, \underline{Y_{A \leftarrow a}} = y, A = a)} = \underline{p(\hat{Y}_{A \leftarrow a'}(U) | X = x, \underline{Y_{A \leftarrow a'}} = y, A = a)}$$

Healthcare Equality

- Protected Features $A = \{\text{Gender}\}$
- Features X , vector representation of coded diagnoses, procedures, medication orders, lab results, and clinical notes
- Prediction Y , a binary indicator of the occurrence of a clinically relevant outcome



$$u \sim p(U) = \text{Normal}(0, I)$$

$$a \sim p(A) = \text{Categorical}(A | \pi)$$

$$x, y \sim p(X, Y | U, A) = p(X | U, A)p(Y | U, A)$$

Training Objective

- σ - sigmoid function
- h - predictor
- J - cross entropy loss

$$\mathcal{L} = J(h_\theta(x, a), y) + \lambda_{\text{CF}} \sum_{a_k \in \mathcal{A}} \mathbb{1}[a \neq a_k] J(h_\theta(\underline{x}_{A \leftarrow a_k}, a_k), \underline{y}_{A \leftarrow a_k}) +$$
$$\lambda_{\text{CLP}} \sum_{a_k \in \mathcal{A}} \mathbb{1}[a \neq a_k] \mathbb{1}[y = \underline{y}_{A \leftarrow a_k}] \frac{\left(\sigma^{-1}(h_\theta(\underline{x}_{A \leftarrow a_k}, a_k)) - \sigma^{-1}(h_\theta(x, a)) \right)^2}{\text{logits}}$$

Dataset Overview

Group	Count	Length of Stay \geq 7 Days	Inpatient Mortality
Asian	17,465	0.187	0.025
Black	5,202	0.239	0.020
Hispanic	21,978	0.196	0.019
Other	11,004	0.200	0.022
Unknown	3,593	0.201	0.072
White	70,391	0.204	0.021
Female	72,556	0.167	0.018
Male	57,076	0.245	0.029
[18, 30)	15,291	0.180	0.007
[30, 45)	27,155	0.140	0.007
[45, 65)	43,529	0.222	0.025
[65, 89)	43,658	0.226	0.036
All	129,633	0.201	0.023

Training Objective

- σ - sigmoid function
- h - predictor
- J - cross entropy loss

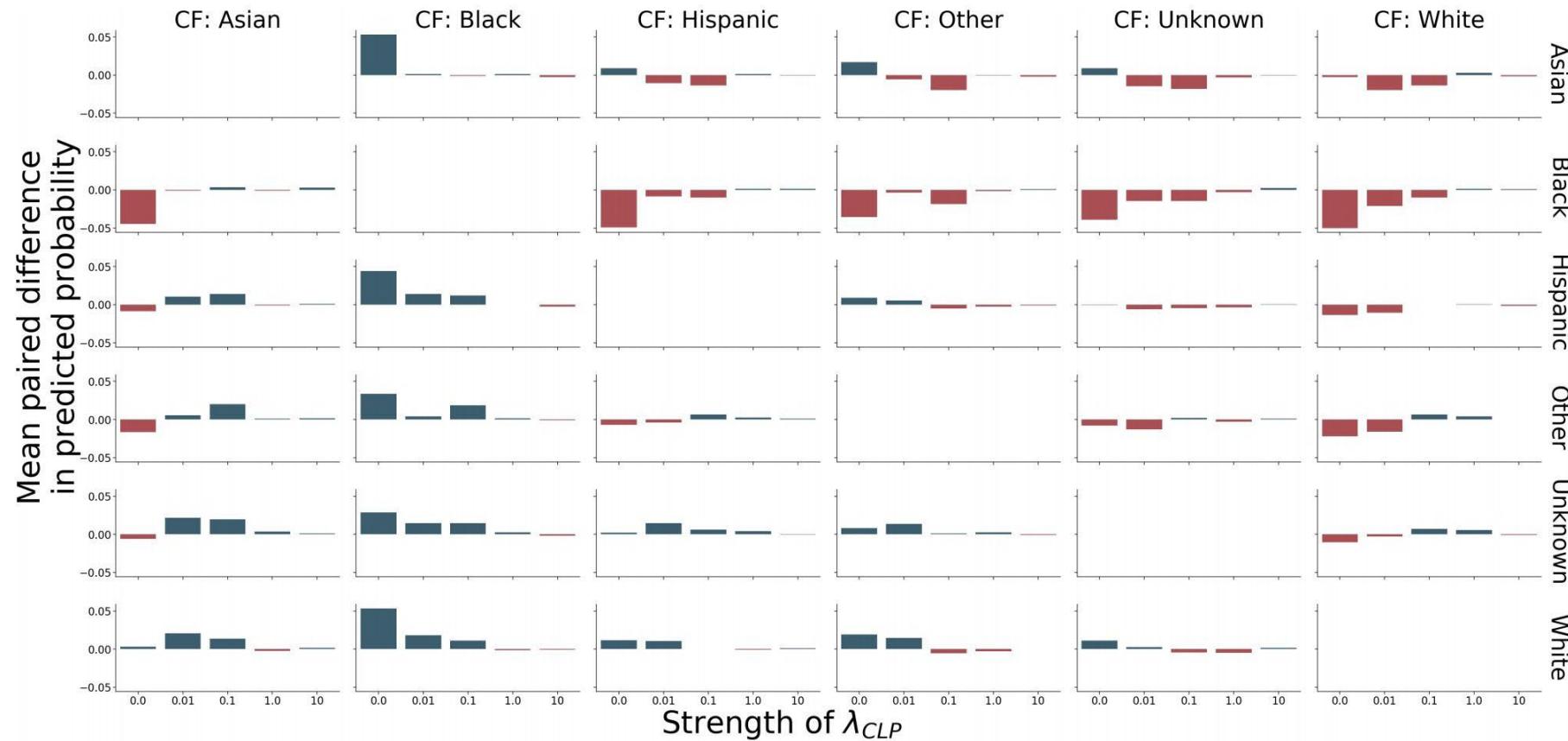
$$\mathcal{L} = J(h_\theta(x, a), y) + \lambda_{\text{CF}} \sum_{a_k \in \mathcal{A}} \mathbb{1}[a \neq a_k] J(h_\theta(\underline{x}_{A \leftarrow a_k}, a_k), \underline{y}_{A \leftarrow a_k}) +$$
$$\lambda_{\text{CLP}} \sum_{a_k \in \mathcal{A}} \mathbb{1}[a \neq a_k] \mathbb{1}[y = \underline{y}_{A \leftarrow a_k}] \frac{\left(\sigma^{-1}(h_\theta(\underline{x}_{A \leftarrow a_k}, a_k)) - \sigma^{-1}(h_\theta(x, a)) \right)^2}{\text{logits}}$$

Results

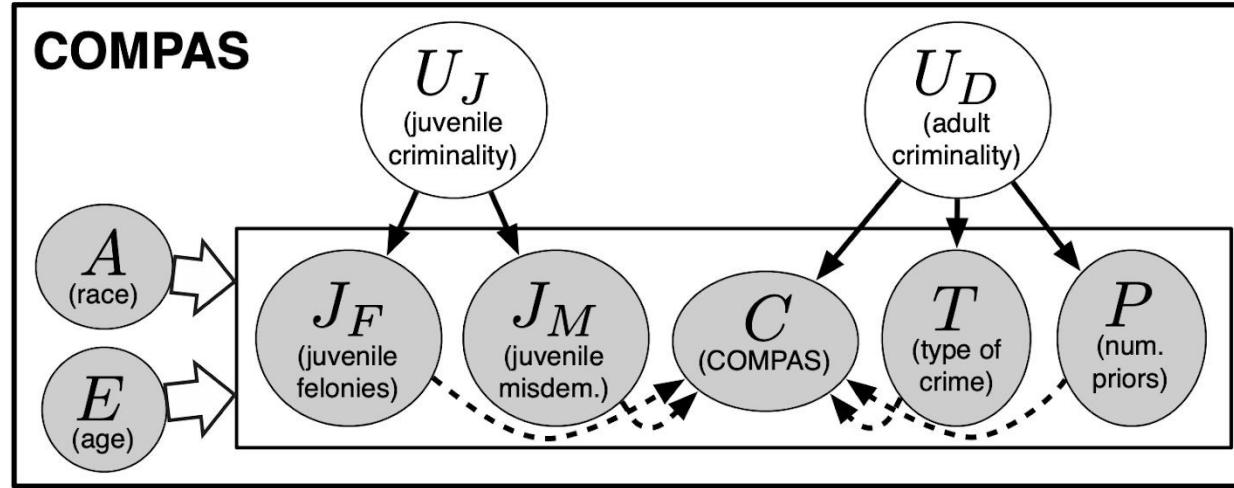
Group	Metric	λ_{CLP}					
		N/A	0.0	0.01	0.1	1.0	10.0
Asian	AUC-PRC	0.605	0.563	0.555	0.561	0.56	0.562
	AUC-ROC	0.86	0.853	0.853	0.854	0.849	0.851
	Brier	0.106	0.11	0.109	0.109	0.11	0.112
Black	AUC-PRC	0.579	0.548	0.55	0.545	0.563	0.573
	AUC-ROC	0.838	0.825	0.82	0.825	0.823	0.823
	Brier	0.124	0.135	0.129	0.128	0.127	0.129
Hispanic	AUC-PRC	0.592	0.558	0.565	0.57	0.564	0.56
	AUC-ROC	0.862	0.855	0.856	0.861	0.853	0.854
	Brier	0.113	0.117	0.115	0.114	0.117	0.118
Other	AUC-PRC	0.549	0.557	0.557	0.563	0.553	0.561
	AUC-ROC	0.824	0.827	0.819	0.824	0.819	0.827
	Brier	0.122	0.124	0.121	0.121	0.122	0.124
Unknown	AUC-PRC	0.675	0.616	0.616	0.606	0.614	0.633
	AUC-ROC	0.9	0.891	0.888	0.893	0.891	0.887
	Brier	0.104	0.106	0.103	0.103	0.105	0.111
White	AUC-PRC	0.575	0.568	0.564	0.559	0.562	0.563
	AUC-ROC	0.847	0.84	0.839	0.838	0.838	0.837
	Brier	0.118	0.12	0.118	0.12	0.12	0.121

Results

- Difference in the counterfactual versus factual predicted probability



COMPAS



$$T \sim \text{Bernoulli}(\phi(b_T + w_C^{U_D} U_D + w_C^E E + w_C^A A))$$

$$C \sim \mathcal{N}(b_C + w_C^{U_D} U_D + w_C^E E + w_C^A A + w_C^T T + w_C^P P + w_C^{J_F} J_F + w_C^{J_M} J_M, \sigma_C)$$

$$P \sim \text{Poisson}(\exp(b_P + w_P^{U_D} U_D + w_P^E E + w_P^A A))$$

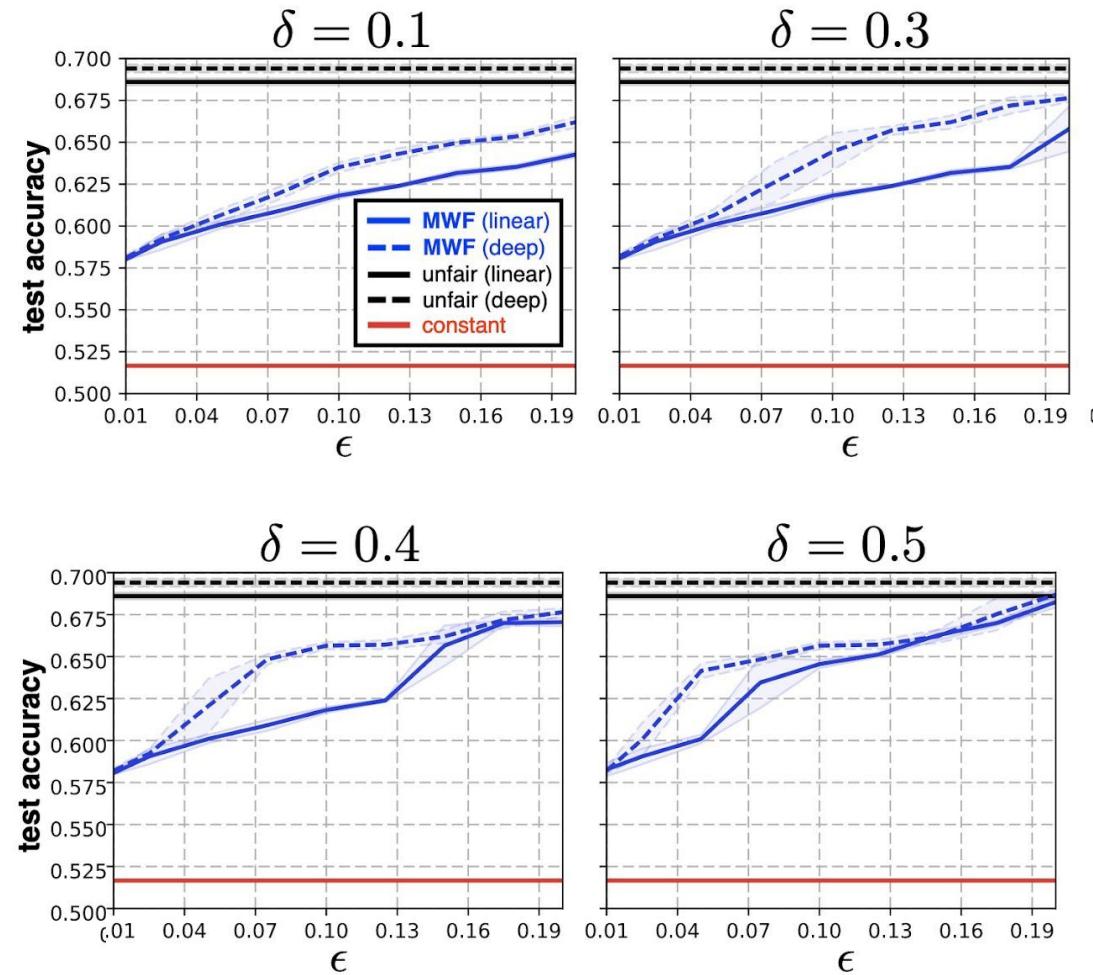
$$J_F \sim \text{Poisson}(\exp(b_{J_F} + w_{J_F}^{U_J} + w_{J_F}^E E + w_{J_F}^A A))$$

$$J_M \sim \text{Poisson}(\exp(b_{J_M} + w_{J_M}^{U_J} + w_{J_M}^E E + w_{J_M}^A A))$$

$$[U_J, U_D] \sim \mathcal{N}(0, \Sigma)$$

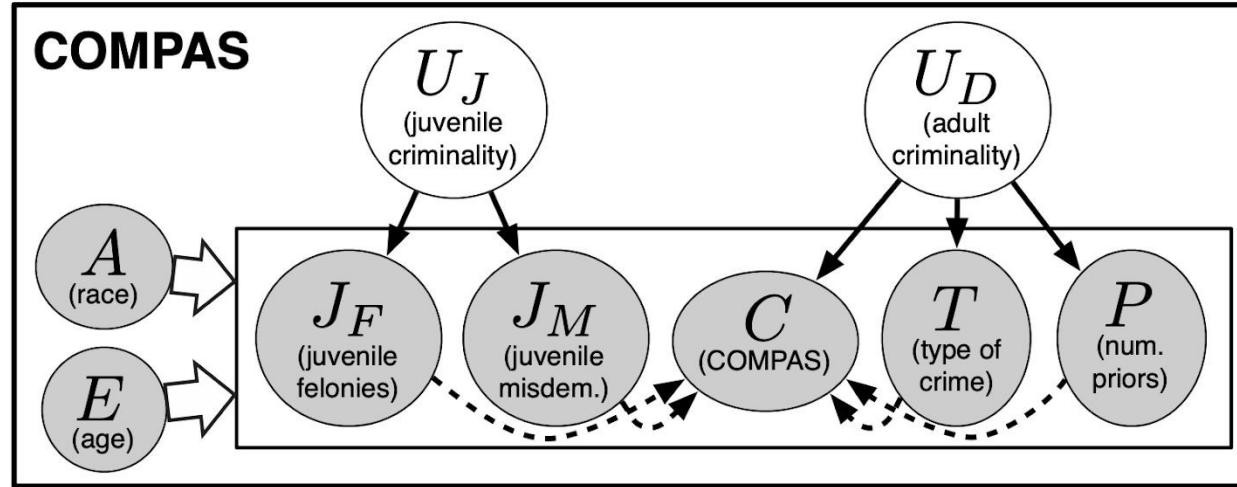
[Russell et al, 2017](#)

Results



[Russell et al, 2017](#)

COMPAS



$$T \sim \text{Bernoulli}(\phi(b_T + w_C^{U_D} U_D + w_C^E E + w_C^A A))$$

$$C \sim \mathcal{N}(b_C + w_C^{U_D} U_D + w_C^E E + w_C^A A + w_C^T T + w_C^P P + w_C^{J_F} J_F + w_C^{J_M} J_M, \sigma_C)$$

$$P \sim \text{Poisson}(\exp(b_P + w_P^{U_D} U_D + w_P^E E + w_P^A A))$$

$$J_F \sim \text{Poisson}(\exp(b_{J_F} + w_{J_F}^{U_J} + w_{J_F}^E E + w_{J_F}^A A))$$

$$J_M \sim \text{Poisson}(\exp(b_{J_M} + w_{J_M}^{U_J} + w_{J_M}^E E + w_{J_M}^A A))$$

$$[U_J, U_D] \sim \mathcal{N}(0, \Sigma)$$

[Russell et al, 2017](#)

Biases of NLP Models

- Denigration
 - The use of culturally or historically derogatory terms
- Under-representation
 - The disproportionately low representation of a specific group
 - e.g., a classifier's performance is adversely affected due to sampling biases of the minority protected group
- Stereotyping
 - An over-generalized belief about a particular category of people
 - e.g., a classifier attributes man to computers more than woman
- Recognition
 - Algorithms perform different for protected groups because of their inherent characteristics
 - e.g., a voice recognition algorithm has better capabilities in recognizing voices in low frequency

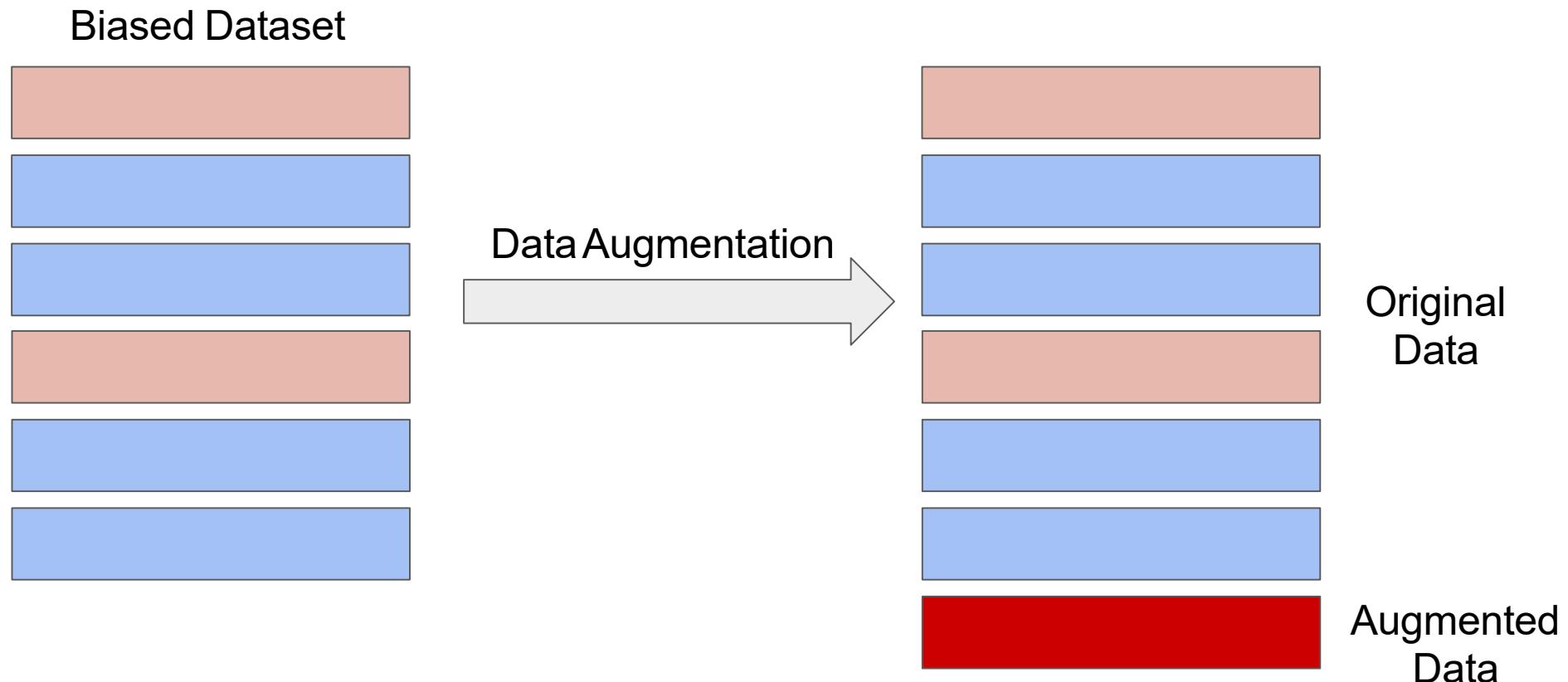
Biases of NLP Models

Task	Example of Representation Bias in the Context of Gender	S
Machine Translation	Translating “He is a nurse. She is a doctor.” to Hungarian and back to English results in “She is a nurse. He is a doctor.” (Douglas, 2017)	✓
Caption Generation	An image captioning model incorrectly predicts the agent to be male because there is a computer nearby (Burns et al., 2018).	✓
Speech Recognition	Automatic speech detection works better with male voices than female voices (Tatman, 2017).	✗
Sentiment Analysis	Sentiment Analysis Systems rank sentences containing female noun phrases to be indicative of anger more often than sentences containing male noun phrases (Park et al., 2018).	✓
Language Model	“He is doctor” has a higher conditional likelihood than “She is doctor” (Lu et al., 2018).	✓
Word Embedding	Analogies such as “man : woman :: computer programmer : homemaker” are automatically generated by models trained on biased word embeddings (Bolukbasi et al., 2016).	✓

(S)tereotyping, (D)enigration, (R)ecognition, (U)nder-representation

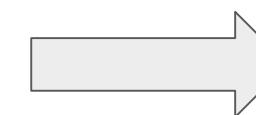
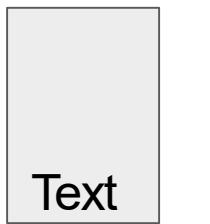
[Sun et al, 2019](#)

Data Augmentation



Word Embeddings

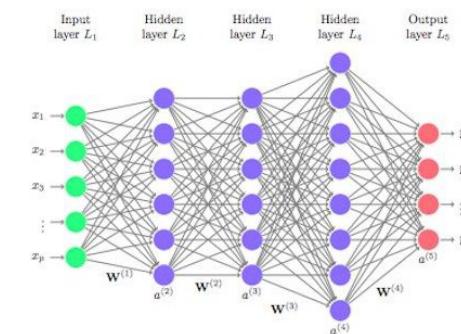
- An Essential Part of Deep NLP Models
 - Classifications (e.g., Sentiment Analysis)
 - Text Generation (e.g., translation, summarization)
 - Text Retrieval (e.g., Question Answering)
 - Visual-Language Representations (e.g., Image Captioning)



Discrete Space

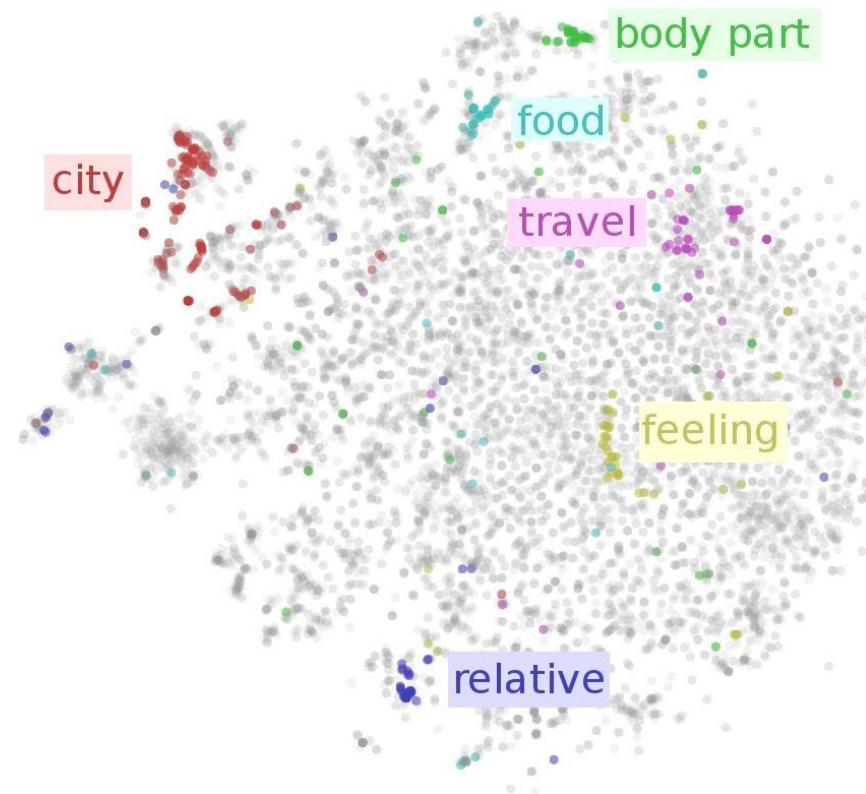
Continuous Space

Neural Networks

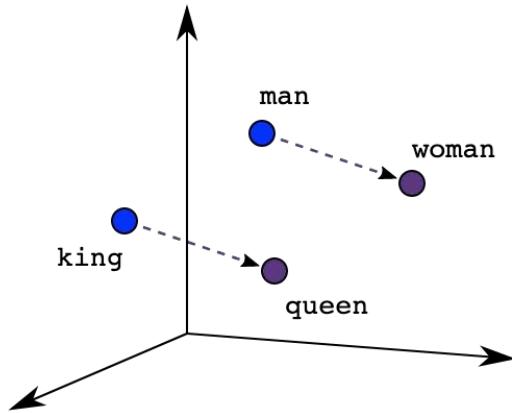


Word Embeddings

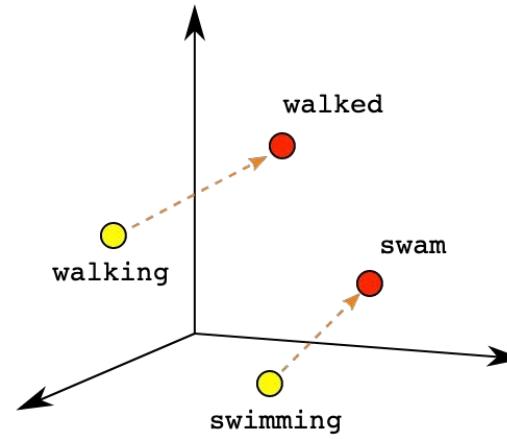
- Embedding Techniques
 - GloVe ([Pennington et al, 2014](#))
 - Word2Vec ([Rong et al, 2014](#))
- Trained Through A Proxy Task
 - Word proximity (GloVe)
 - SkipGram (Word2Vec)



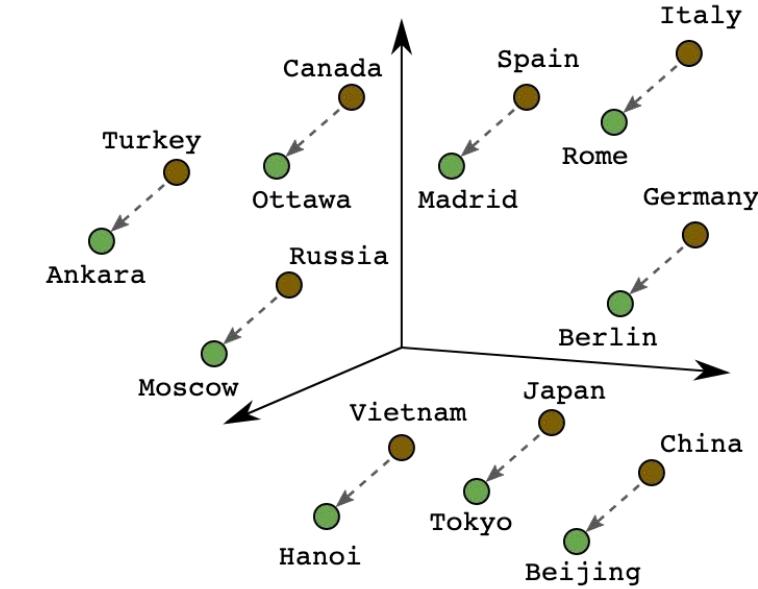
Geometric Properties of Word Embeddings



Male-Female

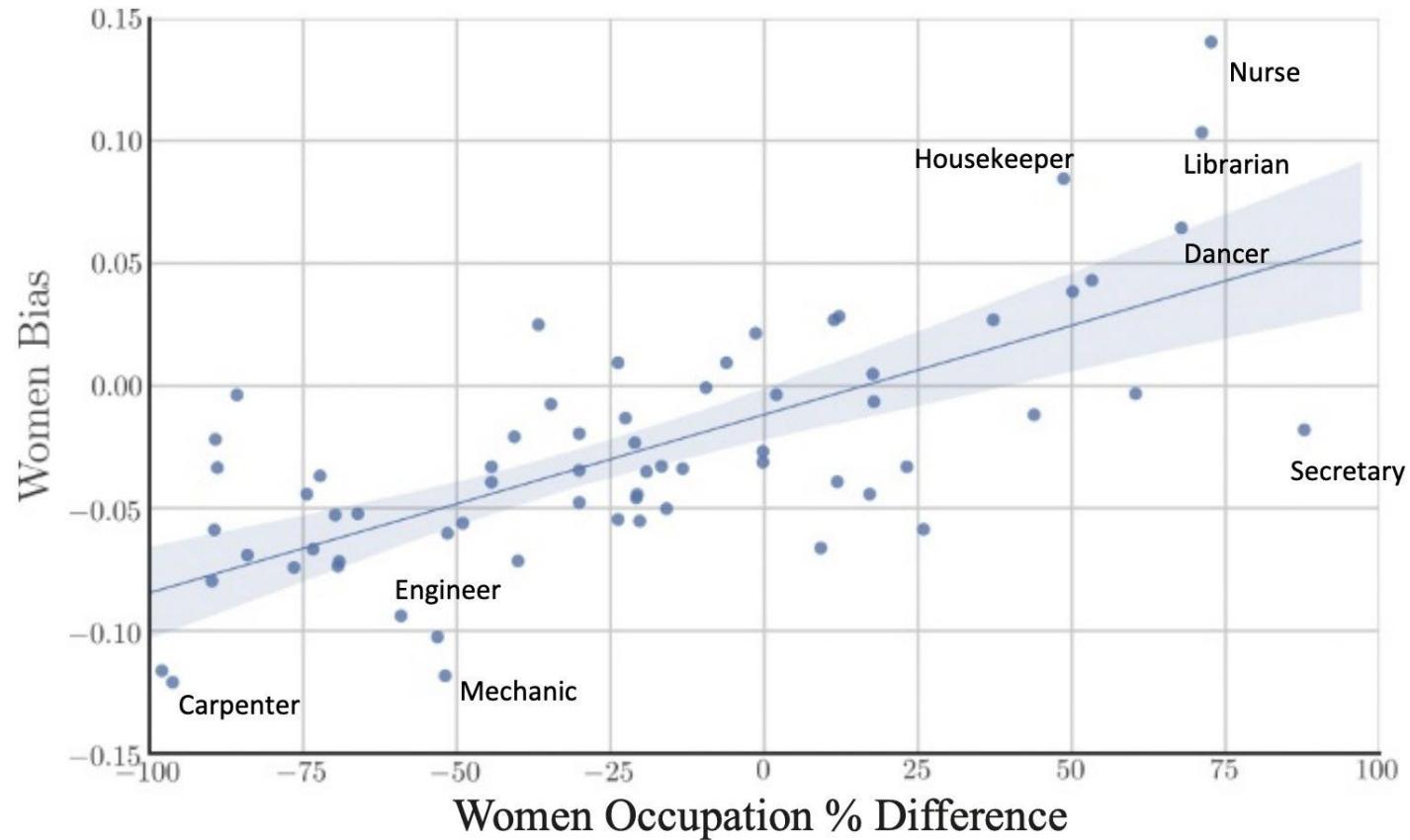


Verb Tense



Country-Capital

Can Word Embedding Be Biased?



Garga et al, 2017

Types of Gender Associations

- Definitional Gender Associations

$$\overrightarrow{\text{man}} - \overrightarrow{\text{woman}} \approx \overrightarrow{\text{king}} - \overrightarrow{\text{queen}}$$

- Stereotypical Gender Associations

$$\overrightarrow{\text{man}} - \overrightarrow{\text{woman}} \approx \overrightarrow{\text{computer programmer}} - \overrightarrow{\text{homemaker}}$$

[Bolukbasi et al, 2016](#)

Definitional and Stereotypical Associations



Gender stereotype *she-he* analogies.

sewing-carpentry	register-nurse-physician	housewife-shopkeeper
nurse-surgeon	interior designer-architect	softball-baseball
blond-burly	feminism-conservatism	cosmetics-pharmaceuticals
giggle-chuckle	vocalist-guitarist	petite-lanky
sassy-snappy	diva-superstar	charming-affable
volleyball-football	cupcakes-pizzas	hairdresser-barber

Gender appropriate *she-he* analogies.

queen-king	sister-brother	mother-father
waitress-waiter	ovarian cancer-prostate cancer	convent-monastery

[Bolukbasi et al, 2016](#)

Gender Subspace

$$\overrightarrow{\text{grandmother}} - \overrightarrow{\text{grandfather}} = \overrightarrow{\text{gal}} - \overrightarrow{\text{guy}} = g$$

$\overrightarrow{\text{she}} - \overrightarrow{\text{he}}$
 $\overrightarrow{\text{her}} - \overrightarrow{\text{his}}$
 $\overrightarrow{\text{woman}} - \overrightarrow{\text{man}}$
 $\overrightarrow{\text{Mary}} - \overrightarrow{\text{John}}$
 $\overrightarrow{\text{herself}} - \overrightarrow{\text{himself}}$

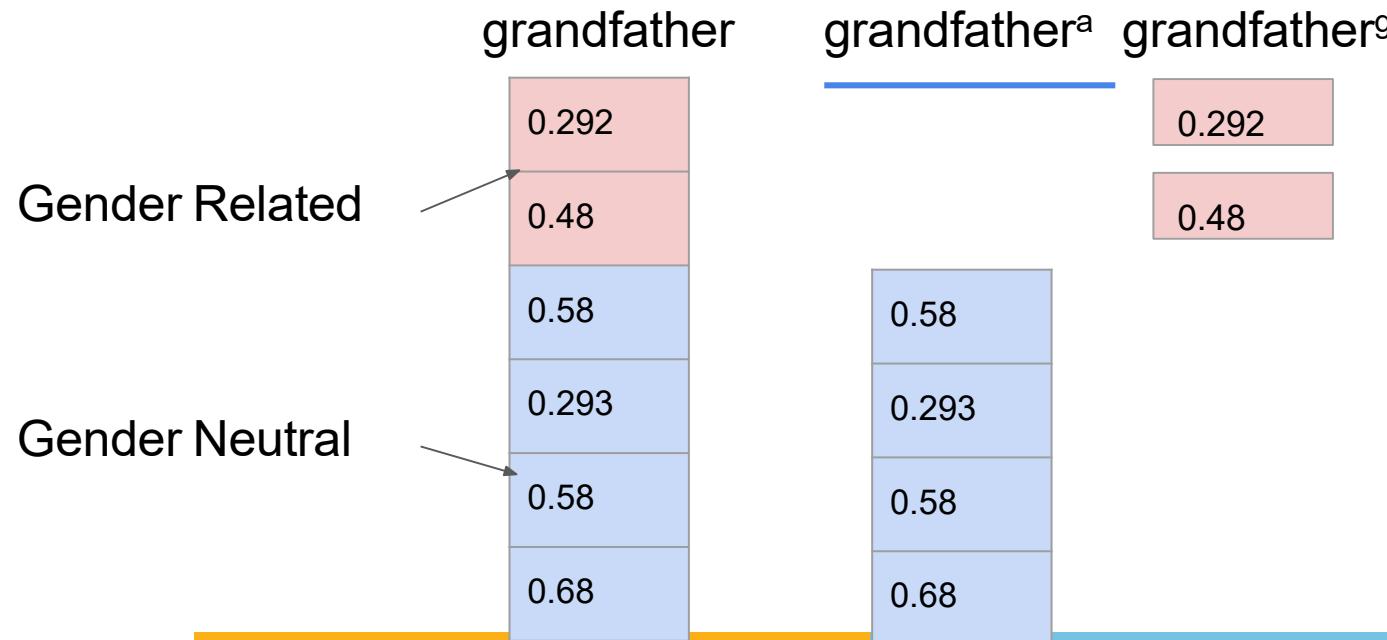
$\overrightarrow{\text{daughter}} - \overrightarrow{\text{son}}$
 $\overrightarrow{\text{mother}} - \overrightarrow{\text{father}}$
 $\overrightarrow{\text{gal}} - \overrightarrow{\text{guy}}$
 $\overrightarrow{\text{girl}} - \overrightarrow{\text{boy}}$
 $\overrightarrow{\text{female}} - \overrightarrow{\text{male}}$

[Bolukbasi et al, 2016](#)

Gender-Neutral Word Embeddings

- Decompose Word Embeddings Into Gender-Related and Gender-Neutral Parts

$$w = [w^{(a)}; w^{(g)}]$$



[Zhao et al, 2018](#)

Gender-Neutral Word Embeddings

- Fine-tuning Word Embeddings Using Debiasing Regularizers

$$J = \underline{J_G} + \underline{\lambda_d J_D} + \underline{\lambda_e J_E}$$

Glove
Loss Function Regulate
 Gender-related
 Words Regulate All Other
 Words

Ω_F
Female Seed Words Ω_N
 All Other Words

Ω_M
Male Seed Words

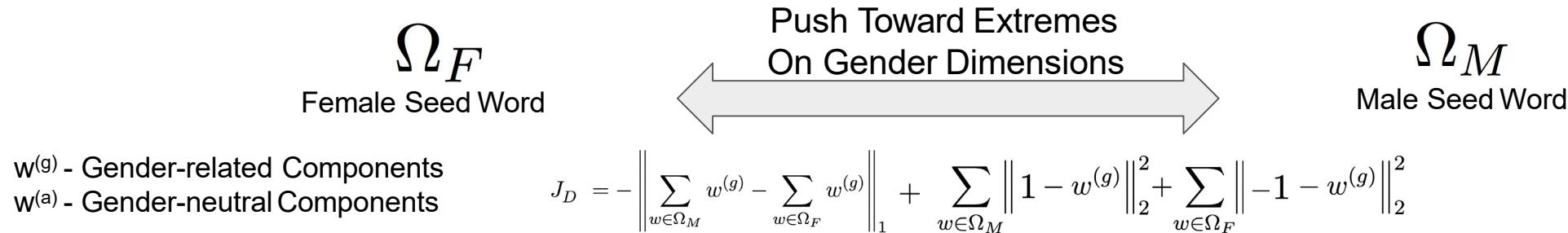
[Zhao et al, 2018](#)

Gender-Neutral Word Embeddings

- Fine-tuning Word Embeddings Using Debiasing Regularizers

$$J = J_G + \lambda_d \underline{J_D} + \lambda_e J_E$$

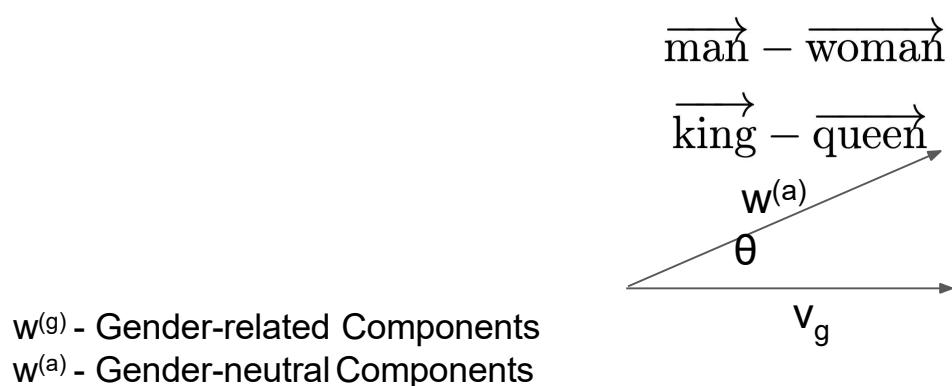
Regulate
Gender-related
Words



Gender-Neutral Word Embeddings

- Fine-tuning Word Embeddings Using Debiasing Regularizers

$$J = J_G + \lambda_d J_D + \lambda_e \underline{J_E}$$



$$v_g = \frac{1}{|\Omega'|} \sum_{(w_m, w_f) \in \Omega_M, \Omega_F} (w_m^{(a)} - w_f^{(a)})$$

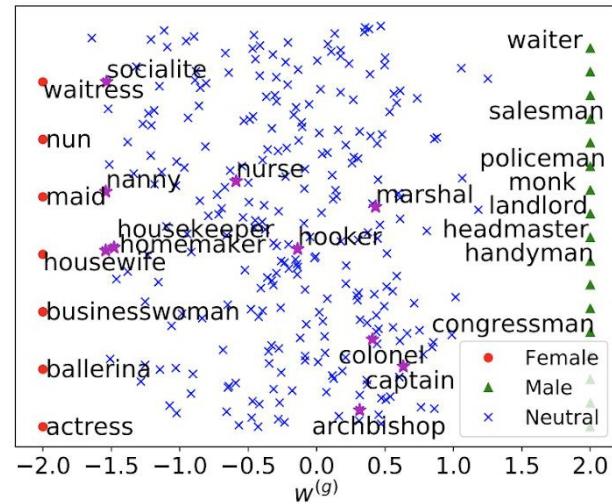
Gender Subspace

$$J_E = \sum_{w \in \Omega_N} \left(v_g^T w^{(a)} \right)^2$$

Regulate All Other Words

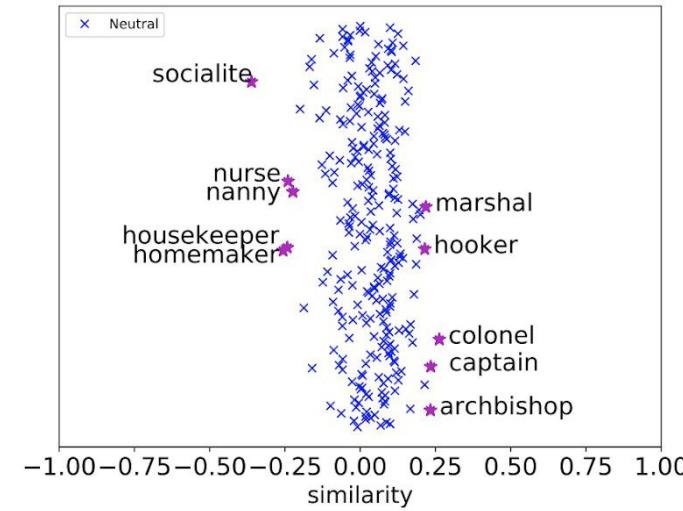
[Zhao et al, 2018](#)

Gender Attribute Separation

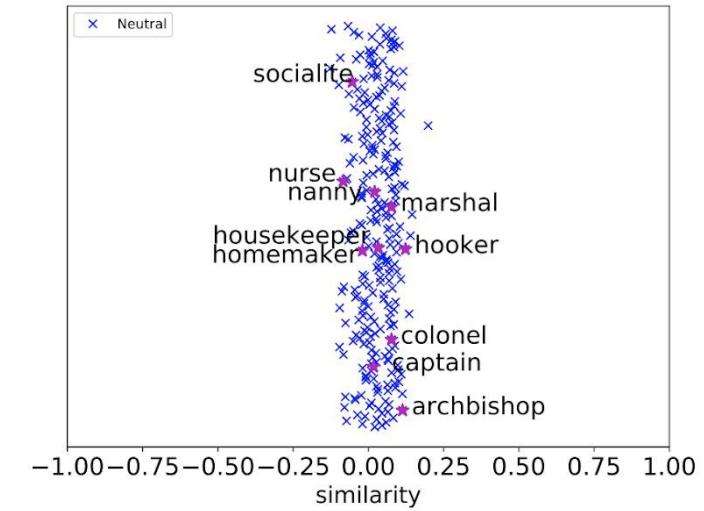


$w^{(g)}$ of All Occupations

$w^{(g)}$ - Gender-related Components
 $w^{(a)}$ - Gender-neutral Components



$w^{(a)}$ of GloVe for Gender Neutral Occupations



$w^{(a)}$ of Gender-Neutral GloVe for Gender Neutral Occupations

Gender Relational Analogy

Question 1: Consider the following word pairs: pilgrim:shrine, hunter:quarry, assassin:victim, climber:peak. What relation best describes these $X:Y$ word pairs?

- (1) “ X worships/reveres Y ”
- (2) “ X seeks/desires/aims for Y ”
- (3) “ X harms/destroys Y ”
- (4) “ X uses/exploits/employs Y ”

Dataset	Embeddings	Definition	Stereotype	None
SemBias	GloVe	80.2	10.9	8.9
	GN-GloVe	97.7	1.4	0.9
SemBias (subset)	GloVe	57.5	20	22.5
	GN-GloVe	75	15	10

[Jurgens et al , 2012](#)

Coreference Resolution

Embeddings	OntoNotes-test	PRO	ANTI	Avg	Diff
GloVe	66.5	76.2	46.0	61.1	30.2
GN-GloVe	66.2	72.4	51.9	62.2	20.5
GN-GloVe(w_a)	65.9	70.0	53.9	62.0	16.1

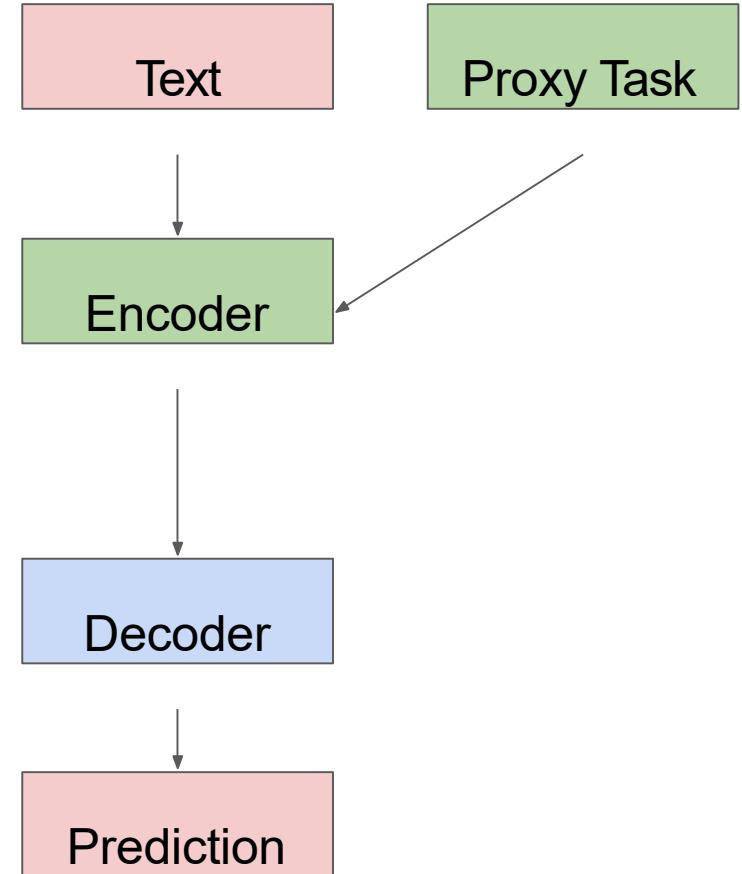
$w^{(a)}$ - Gender-neutral Components

[Jurgens et al , 2012](#)

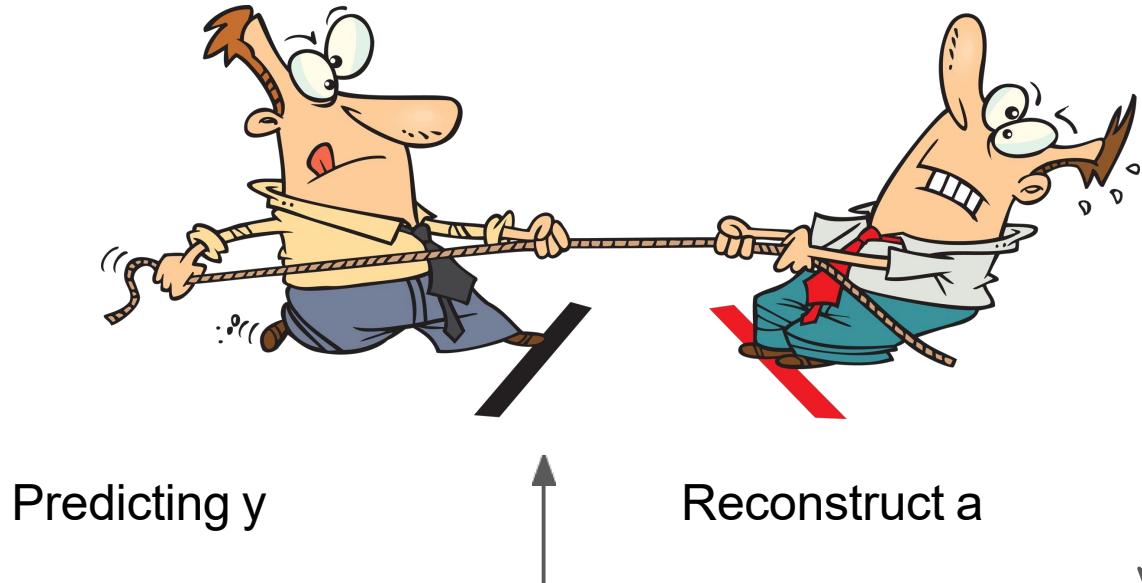
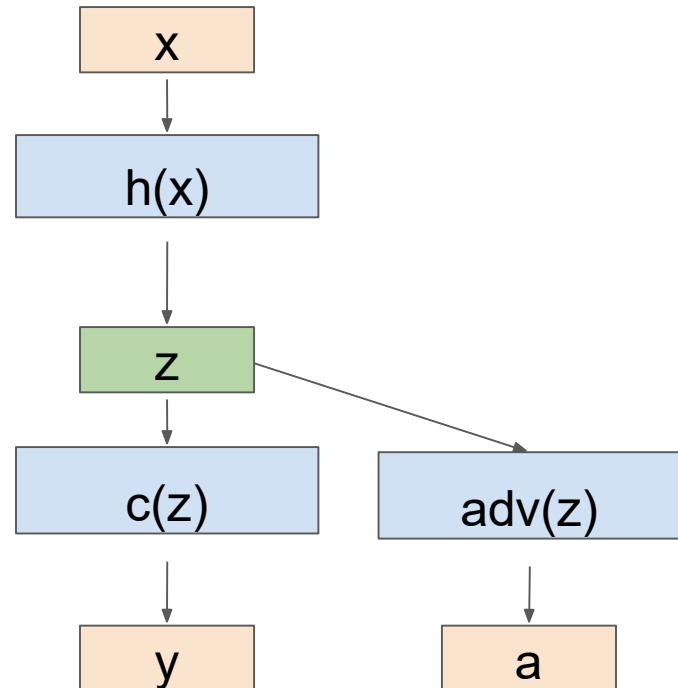
The Use of Pre-trained NLP Encoders



- Pre-trained Encoders Are Widely Used in NLP
 - Transfer information from a related domain
 - Boost performance on a small data set
 - Trained through a proxy task
- Pre-trained NLP Encoders
 - ELMO ([Peters et al 2018](#))
 - BERT ([Devlin et al, 2018](#))
 - XLNet ([Yang et al, 2019](#))
- Can Pre-trained Encoders Be Biased?



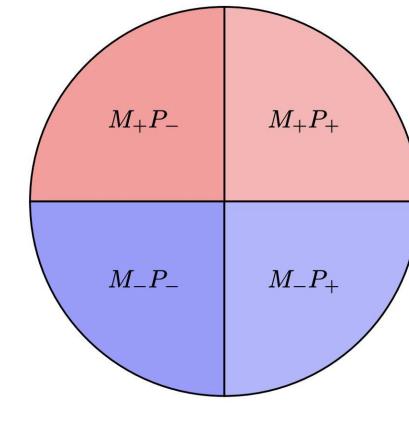
Adversarial Learning



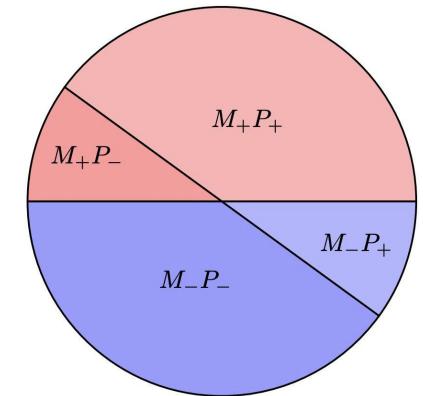
[Elazar et al, 2018](#)

Twitter Prediction Problem

- Twitter Sentiment & Mention Detection
- Protected Attributes
 - Race
 - Gender
 - Age
- Leakage
 - Predict protected attributes



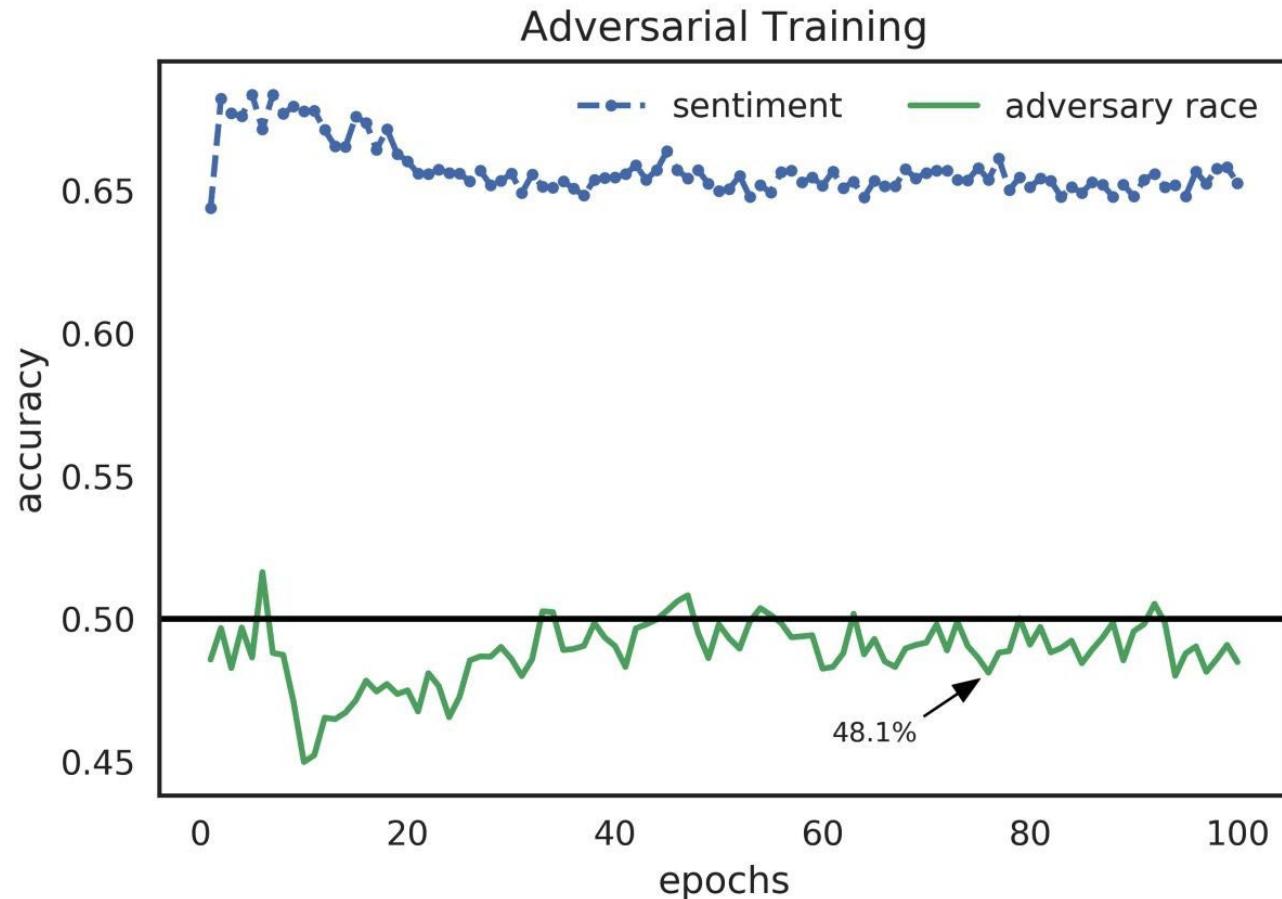
balanced



unbalanced

Data	Task	Protected Attribute	Balanced		Unbalanced	
			Task Acc	Leakage	Task Acc	Leakage
DIAL	Sentiment	Race	67.4	64.5	79.5	73.5
	Mention	Race	81.2	71.5	86.0	73.8
PAN16	Mention	Gender	77.5	60.1	76.8	64.0
		Age	74.7	59.4	77.5	59.7

Main Task and Adversary Accuracies



[Elazar et al, 2018](#)

Beefing Up the Adversary

- Increase the Capacity of the Adversary
 - Model Capacity
 - Weight on Loss
 - Ensemble

Method	Parameter	DIAL			PAN16					
		Sentiment	Race	Δ	Mention	Gender	Δ	Mention	Age	Δ
No Adversary Baseline	-	67.4	14.5	-	77.5	10.1	-	74.7	9.4	-
Standard Adversary	(300/1.0/1)	64.7	6.0	5.0	75.6	8.5	8.0	72.5	7.3	6.9
Adv-Capacity	500	64.1	6.7	5.2	73.8	8.1	6.7	71.4	4.3	4.1
	1000	63.4	7.1	4.9	75.2	8.9	7.0	71.6	6.3	4.0
	2000	65.2	8.1	6.9	76.1	6.7	6.4	71.9	6.0	5.7
	5000	63.9	6.2	3.7	74.5	5.6	1.6	73.0	10.2	9.6
	8000	65.0	7.1	4.8	75.7	5.4	4.2	71.9	9.8	7.3
λ	0.5	63.9	6.8	6.2	75.6	7.8	6.8	73.1	4.8	3.4
	1.5	64.9	7.4	5.4	75.6	4.9	2.4	72.5	6.8	5.8
	2.0	64.2	7.3	5.9	76.0	-7.2	6.7	72.1	8.5	7.7
	3.0	65.8	10.2	10.1	73.7	6.4	6.1	72.5	-6.3	5.2
	5.0	50.0	-	-	73.6	6.5	5.7	69.0	3.2	2.9
Ensemble	2	62.4	7.4	5.4	74.8	6.4	5.0	72.8	8.8	8.3
	3	66.5	6.5	5.0	75.3	4.9	3.1	72.1	6.7	6.0
	5	63.8	4.8	2.6	74.3	4.1	3.0	70.1	5.7	5.4

Δ - the difference between the attacker score and the corresponding adversary's accuracy

Elazar et al, 2018



BITS Pilani
Pilani Campus

Fair, Accountable, Transparent Machine Learning (FAccT ML) **ZG517**

Dr. Sugata Ghosal
sugata.ghosal@pilani.bits-pilani.ac.in



**Session 9
Date – 30th August 2023
Time – 7:30 PM to 9:30 PM**

These slides are prepared by the instructor, with grateful acknowledgement of many others who made their course materials freely available online.

Outline

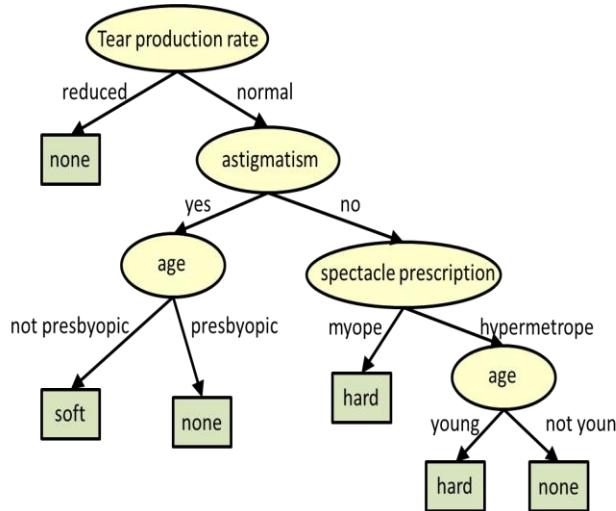
- ML Interpretability
- Intrinsically Interpretable Models
 - Simple interpretable models
 - Instricically interpretable techniques for deep learning
- Interpretability Concepts
 - Intrinsic and post hoc methods
 - model-specific and model-agnostic methods
 - Local and global interpretable methods
 - Interpretability and performance trade-offs

Readings: Chapters 3, 5 and 9 (LIME and Anchors) of

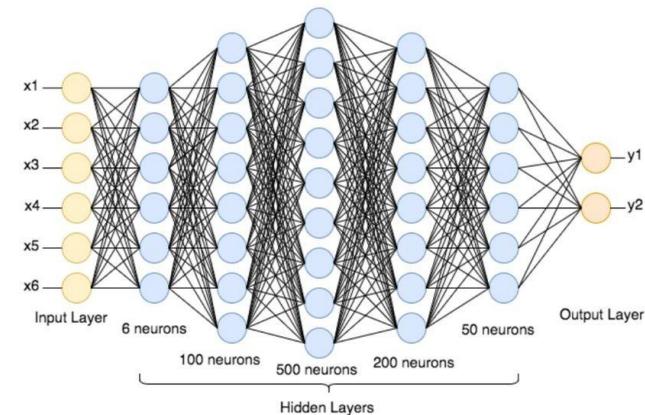
- <https://christophm.github.io/interpretable-ml-book/>
- Ribeiro et al., “Why Should I Trust You?” Explaining the Predictions of Any Classifier, KDD 2016
- Ribeiro et al., Anchors: High-Precision Model-Agnostic Explanations, AAAI 2018

Machine Learning Interpretability

- ML interpretability allows one to examine model's basis in its decision making process.

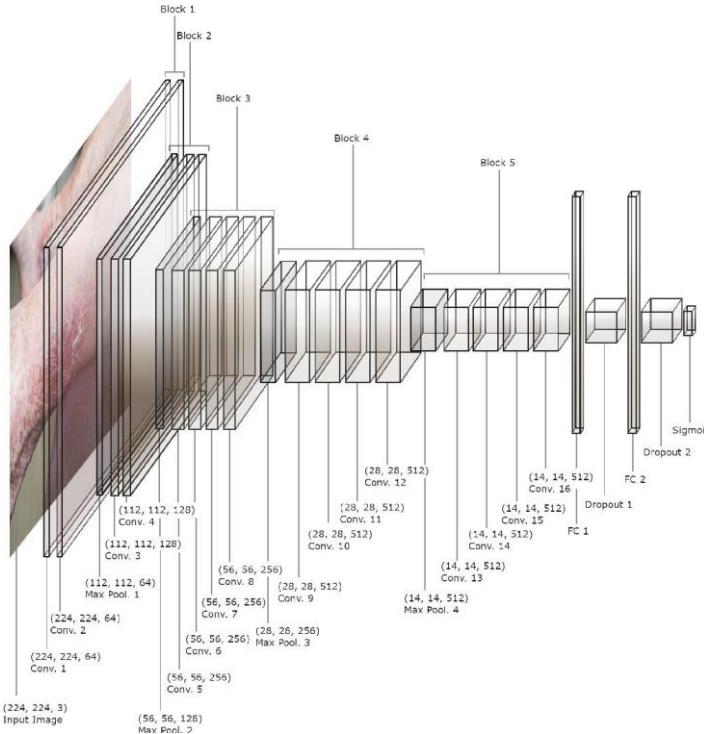


An interpretable tree model to find out the kind of contact lens a person may wear



A neural network which is usually considered a black-box model.

VGG19 Architecture

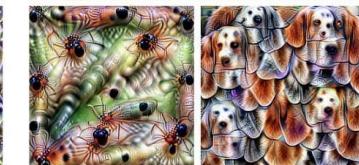
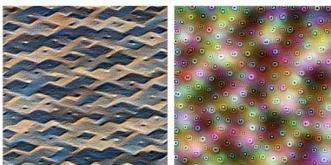
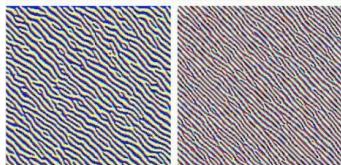
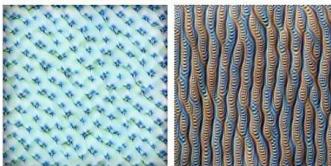
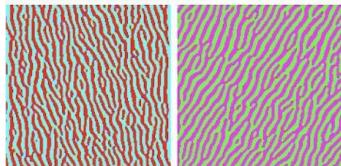
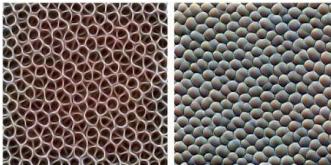
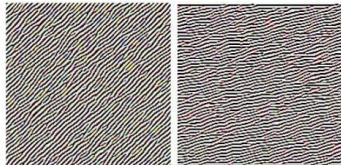


46 layers

143,667,240 parameters

model size: 575 MB

Visualizations of GoogLeNet



Edges (layer conv2d0)

Textures (layer mixed3a)

Patterns (layer mixed4a)

Parts (layers mixed4b & mixed4c)

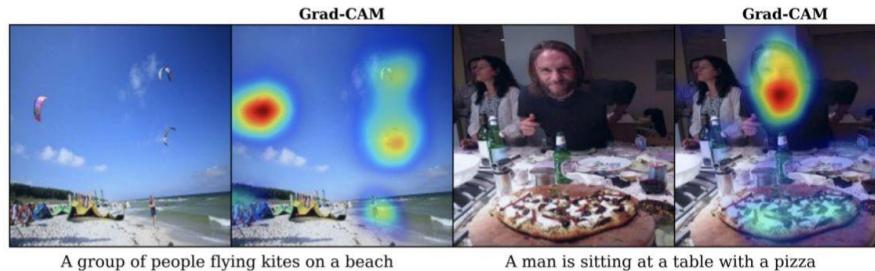
Objects (layers mixed4d & mixed4e)

Reasons for ML Interpretability

- Our society has been shifted to rely on AI more than ever
 - autonomous vehicles
 - security
 - finance
 - many others
- Who will benefit from ML Interpretability?
 - End Users: enhance trust, understand the consequences of the decisions, e.g., privacy, fairness.
 - Regulatory Agencies: compliance, audits, and accountability.
 - Model Designers: diagnose model performance

Image Caption Generation ([Selvaraju et al., 2017](#))

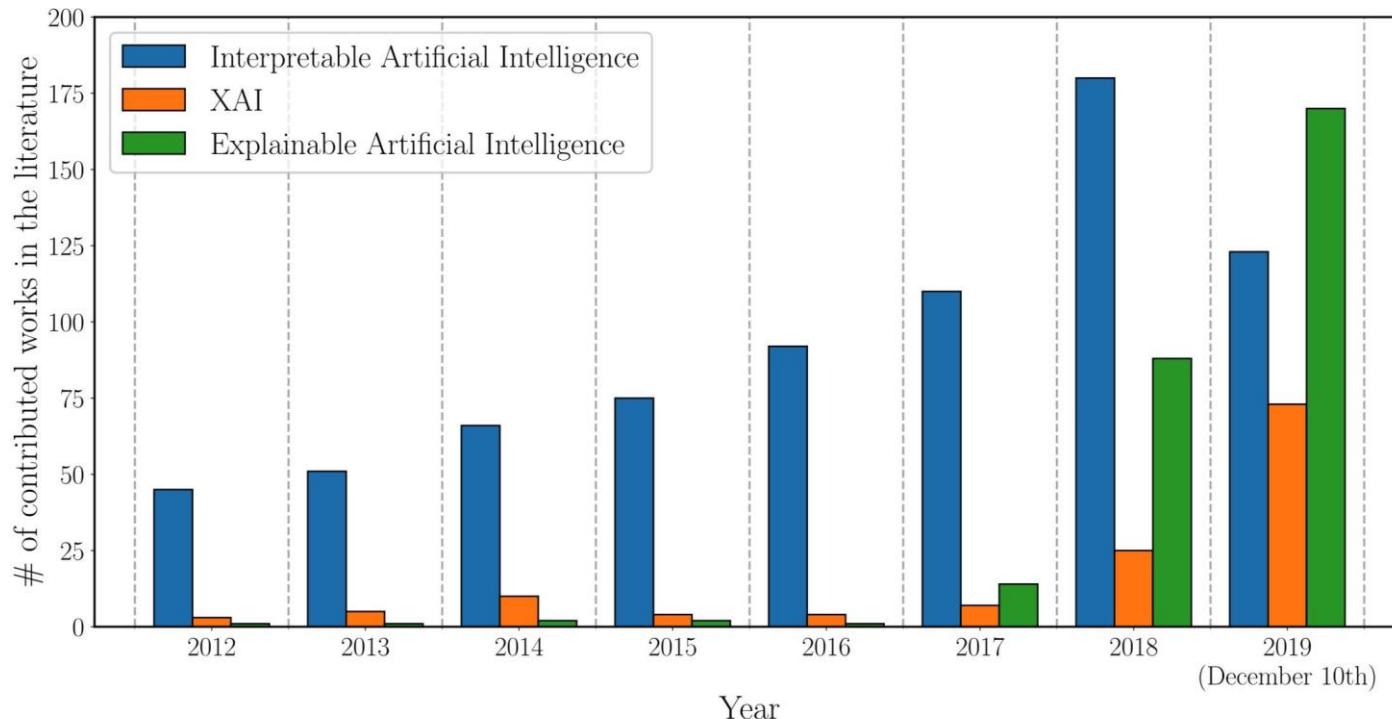
- Highlighted regions explaining an image caption generation algorithm.



Right to Explanation

- Credit Scores in United States
 - Equal Credit Opportunity Rights (Regulation B of the [Code of Federal Regulations](#))
 - Creditors are required to notify applicants of action taken with statement of *specific*
- European Union General Data Protection Regulation
 - GDPR 1995 provided a legally disputed form of a right to an explanation [Recital 71](#)
 - "The data subject should have the right not to be subject to a decision, which may include a measure, evaluating personal aspects relating to him or her which is based solely on automated processing..."
- France
 - In a decision taken on the basis of an algorithmic treatment, the rules that define that treatment and its "principal characteristics" must be communicated to the citizen upon request
 - the degree and the mode of contribution of the algorithmic
 - the data processed and its source
 - the treatment parameters, and where appropriate, their weighting
 - the operations carried out by the treatment.

Surge in Explainable Research ([Arrieta et al., 2019](#))



Outline

- ML Interpretability
- Intrinsically Interpretable Models
 - Simple interpretable models
- Interpretability Concepts
 - Intrinsic and post hoc methods
 - model-specific and model-agnostic methods
 - Local and global interpretable methods
 - Interpretability and performance trade-offs

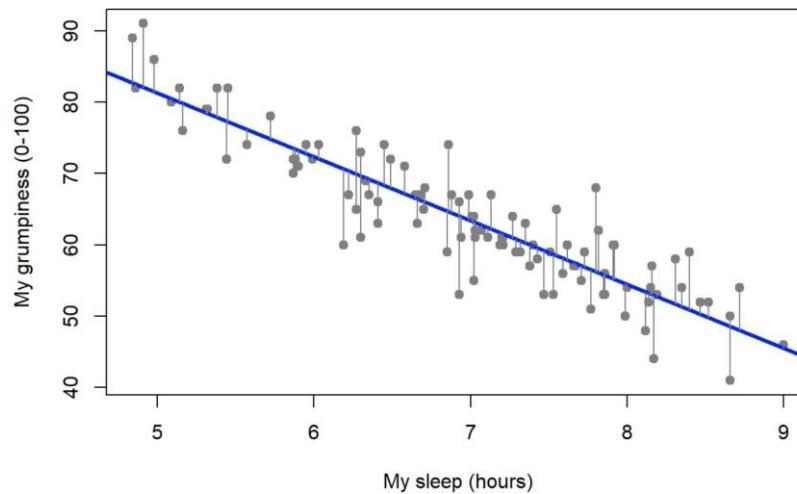
Intrinsically interpretable models

- Models that are interpretable by design
- No post-processing steps are needed to achieve interpretable.

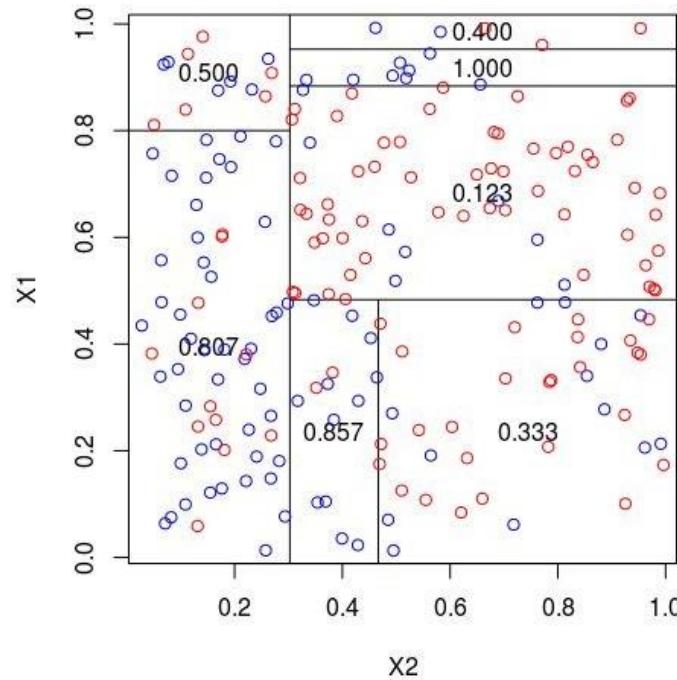
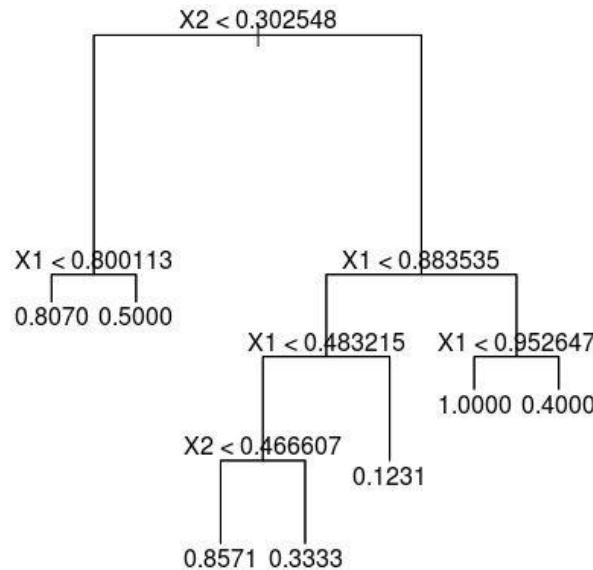
Linear Regression

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^T \boldsymbol{\beta} + \varepsilon_i, \quad i = 1, \dots, n,$$

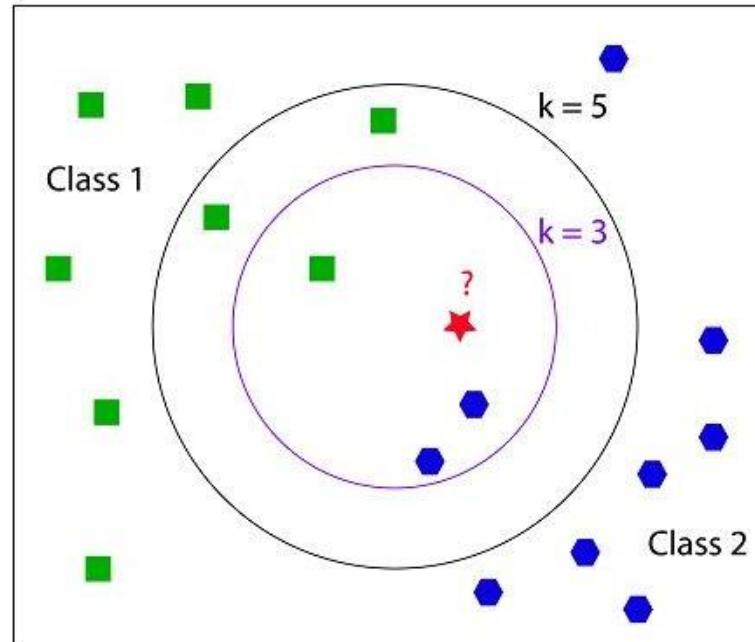
interpretable components



Decision Trees

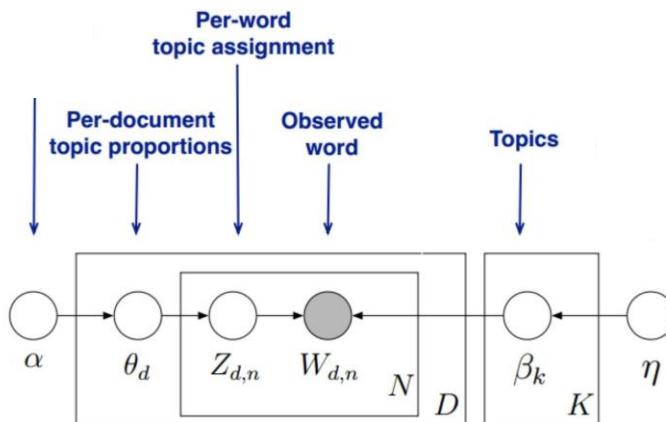


K-Nearest Neighbors



Bayesian Models

Latent Dirichlet Allocation ([Blei et al., 2003](#))



Documents

XXXX XXXX I purchased a vehicle from XXXX XXXX XXXX which I traded in my XX/XX/XXXX Volvo. I then signed contract and release of liability to the dealer. I still have the contract. Three years later I received a letter from a collection agency that I owe them XXXX dollars for the car I traded in, that was towed from XXXX XXXX XXXX XXXX said at the time the car was still in my name. So I went back to the dealer and the dealer before was sold to another company. I spoke with XXXX XXXX and did what they told me and it is still on my credit report. I am really frustrated on what I am going through. The collectors will not listen to me. What can I do. The agency is XXXX Collections in XXXX XXXX California.

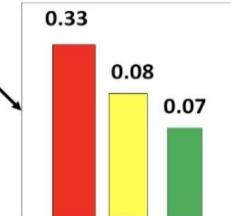
Topics β_k

car	0.23
vehicle	0.18
finance	0.09
...	...

collect	0.25
agenc	0.13
recover	0.05
...	...

receiv	0.23
letter	0.17
send	0.1
...	...

Topic proportions θ_d



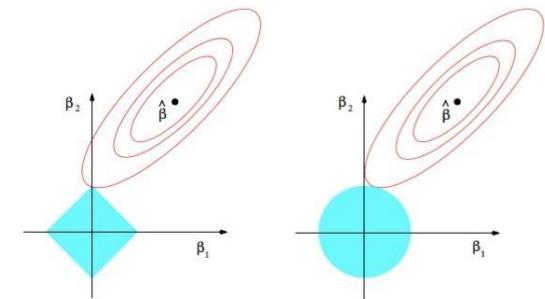
Sparsity

- Controls the sparsity of model parameters when learning a model
- Popular choices
 - L1 regularization

$$\|\mathbf{w}\|_1 = |w_1| + |w_2| + \dots + |w_N|$$

- L2 regularization

$$\|\mathbf{w}\|_2 = (w_1^2 + w_2^2 + \dots + w_N^2)^{\frac{1}{2}}$$



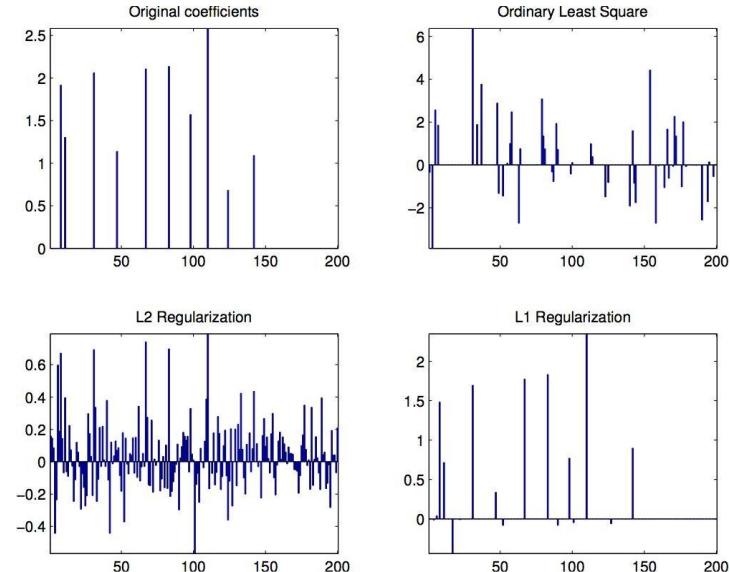
Sparsity for Interpretable Linear Regression

- In the case of linear regression
 - $\hat{y} = w_1x_1 + w_2x_2 + \dots + w_Nx_N + b$
- Linear regression with L1 regularization

$$\text{Loss} = \text{Error}(y, \hat{y}) + \lambda \sum_{i=1}^N |w_i|$$

- Linear Regression with L2 regularization

$$\text{Loss} = \text{Error}(y, \hat{y}) + \lambda \sum_{i=1}^N w_i^2$$

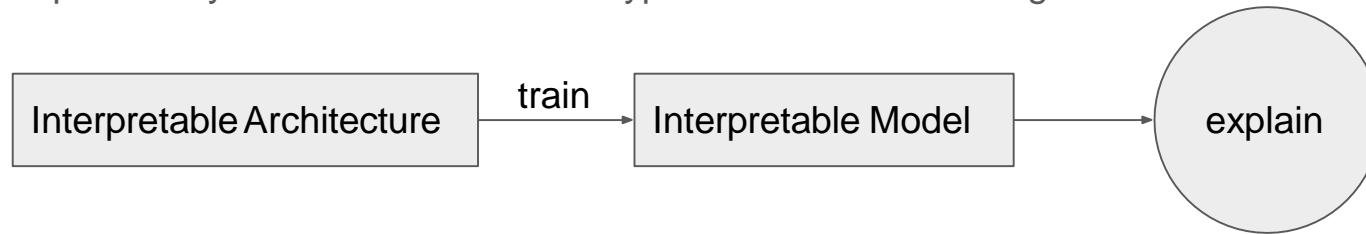


Outline

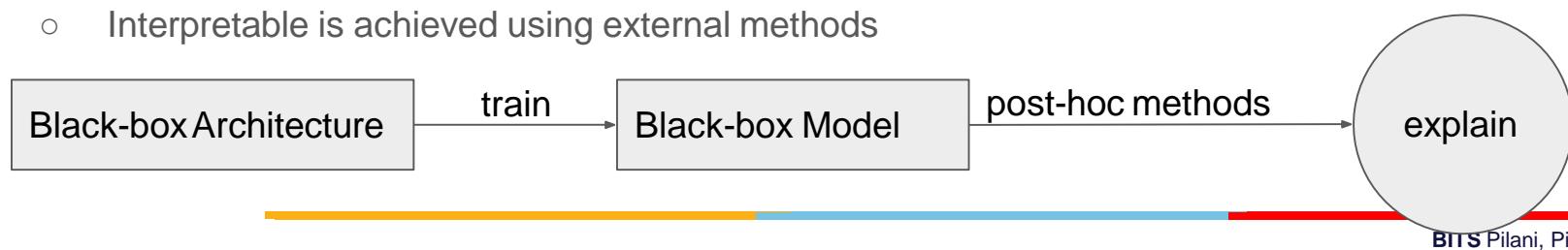
- Fair Representation Learning
- ML Interpretability
- Intrinsically Interpretable Models
 - Simple interpretable models
 - Intrinsicly interpretable techniques for deep learning
- Interpretability Concepts
 - Intrinsic and post hoc methods
 - model-specific and model-agnostic methods
 - Local and global interpretable methods
 - Interpretability and performance trade-offs

Intrinsic and Post Hoc Interpretability

- Intrinsically interpretable models
 - Interpretable is achieved by model design
 - ML models are explainable by itself
 - Explainability is often achieved as a byproduct of model training

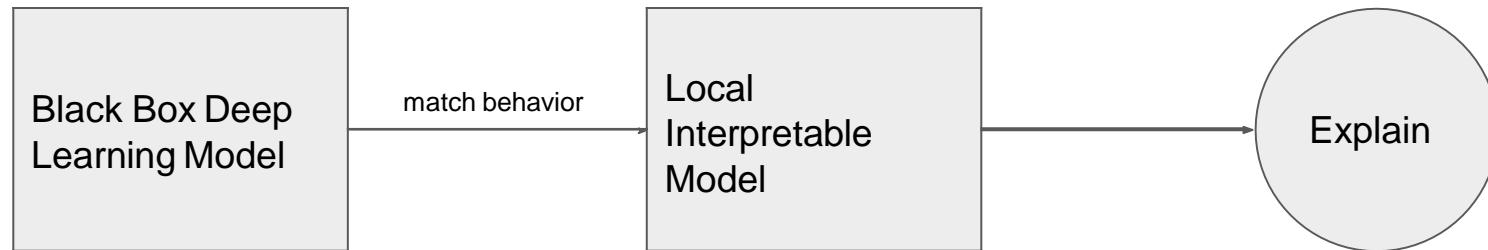


- Post Hoc/Model-specific methods
 - Explainability is often achieved after the model is trained
 - Interpretable is achieved using external methods



Post Hoc Interpretability

- One of the way to achieve Post Hoc Interpretability is to deploy a local proxy model
- We will introduce more about Post Hoc Interpretable methods later.



Model Specific and Model Agnostic Methods

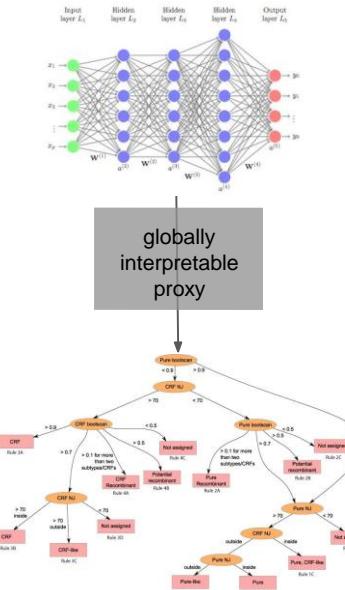
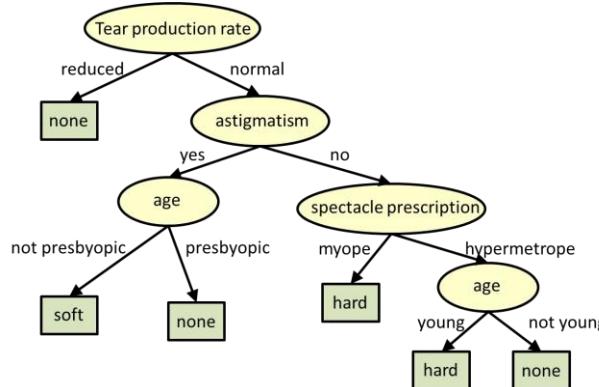
- Model Specific Methods
 - Techniques that can be used for a specific architecture
 - Usually preferable when you have the ability to design your own model
 - Model specific techniques might compromise the performance of your model
 - Requires training the model using a dataset
 - Intrinsic methods are by definition model specific
- Model-agnostic Methods
 - Techniques that can be used across many black box models
 - Model-agnostic methods will not affect the performance of your model
 - Do not require training the model
 - Will be covered in the next lecture
 - Post hoc methods are usually model-agnostic

Global and Local Interpretability

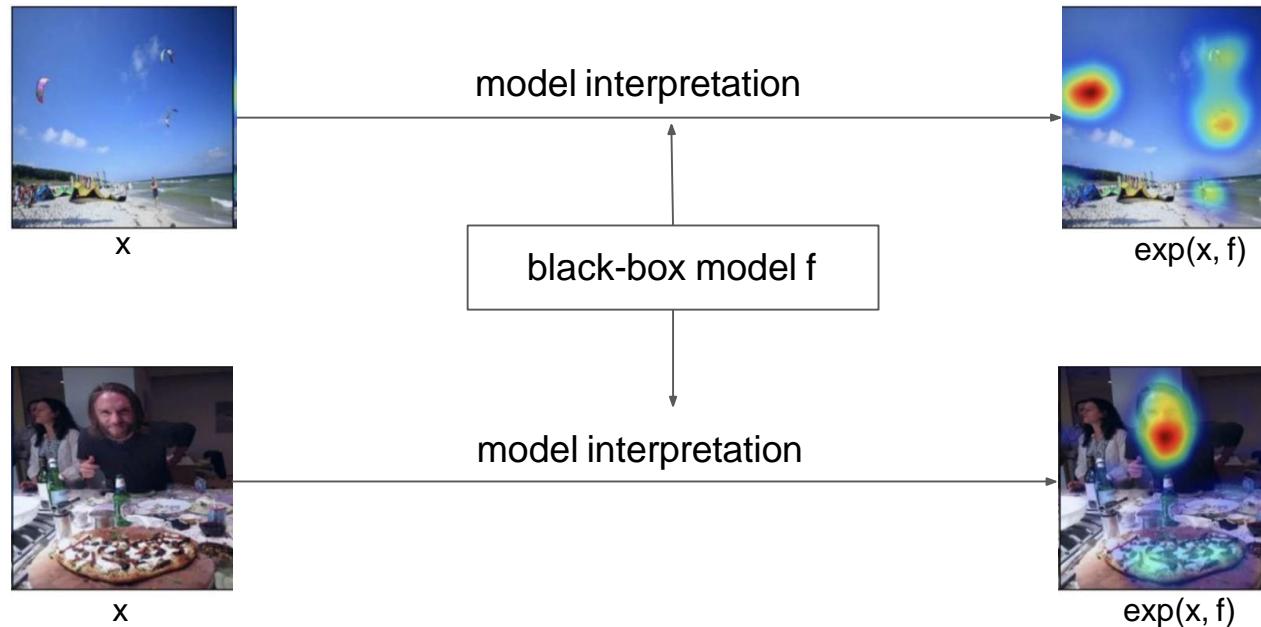
- Global Interpretability
 - Explains the entire ML model at once from input to prediction
 - 1) Holistic Model Interpretability
 - 2) Modular Level Interpretability
 - e.g., Decision Trees, Linear regression

- Local Interpretability
 - Explain how predictions change for when input changes
 - 1) For a single prediction
 - 2) for a group of predictions

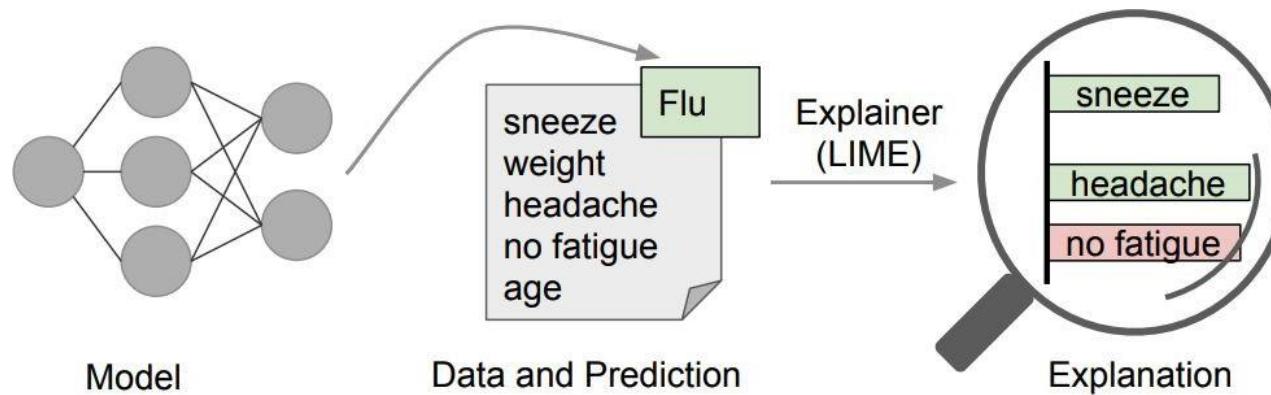
Global Interpretability



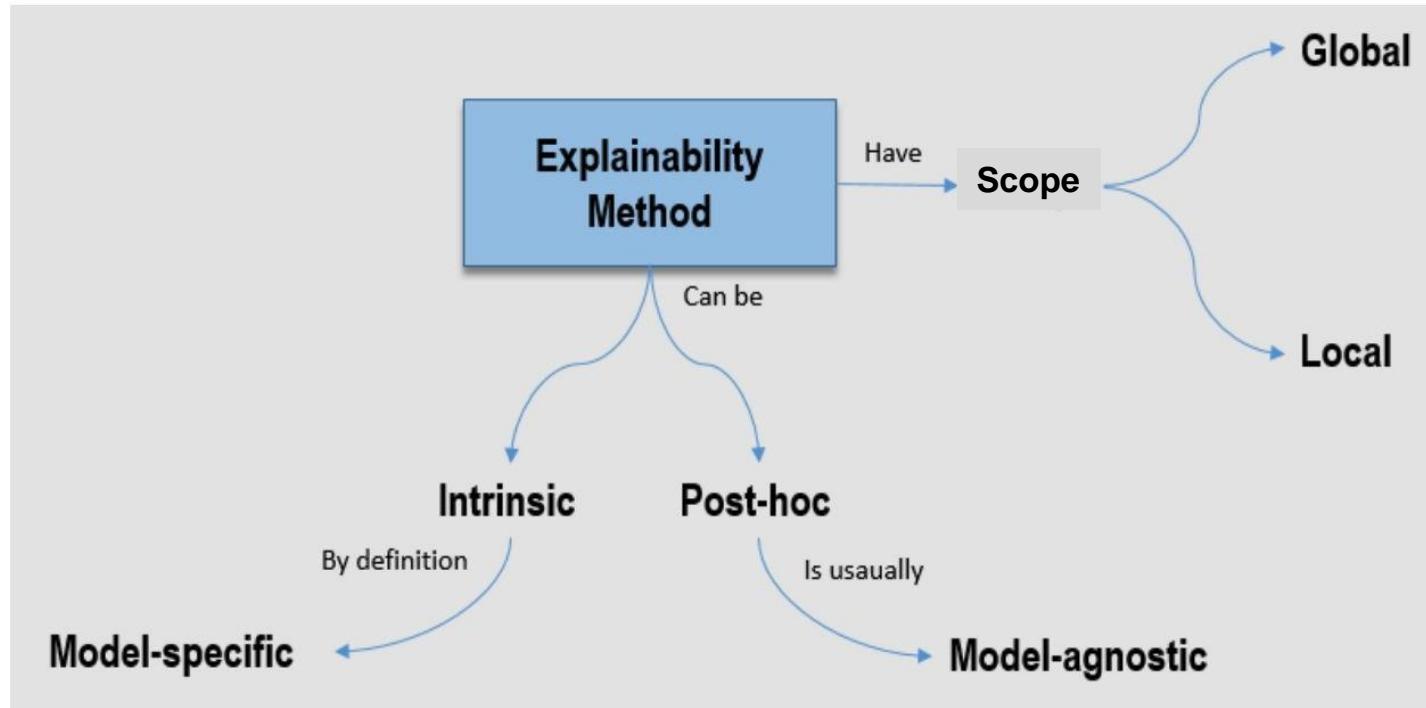
Local Interpretability



Local Interpretability



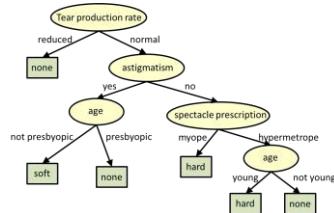
An Ontology of AI Explainability ([ADADI et al, 2018](#))



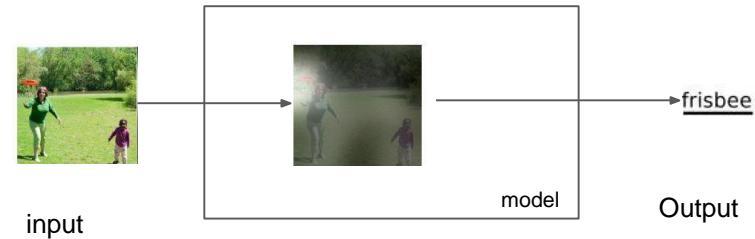
The Big Picture

Intrinsic

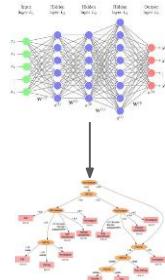
Globally Interpretable



Locally Interpretable



Post Hoc



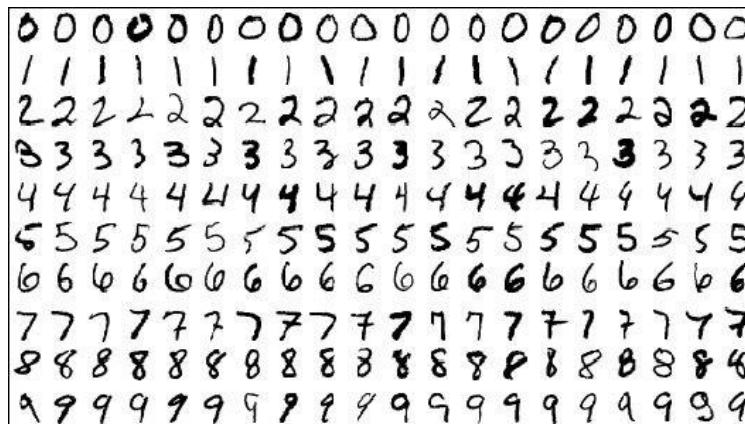
black-box model

interpretation

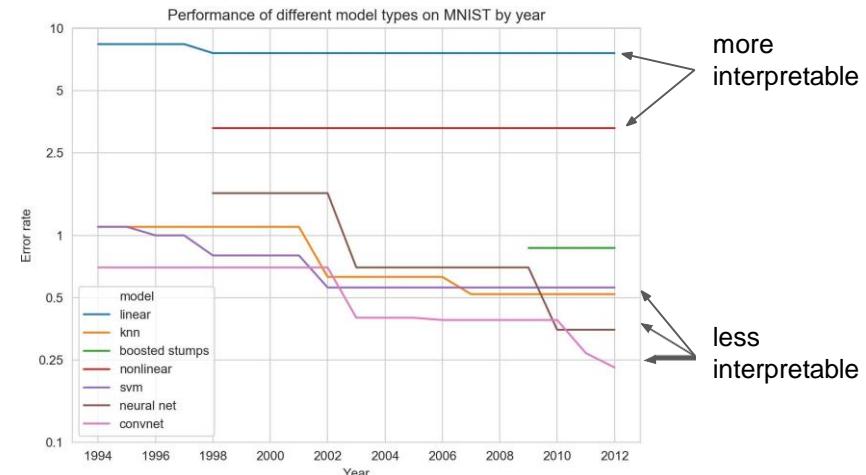


Interpretability and Performance Trade-offs

- highly performed models tend to be less interpretable.
- Can powerful models with complex structures be interpretable at the same time?



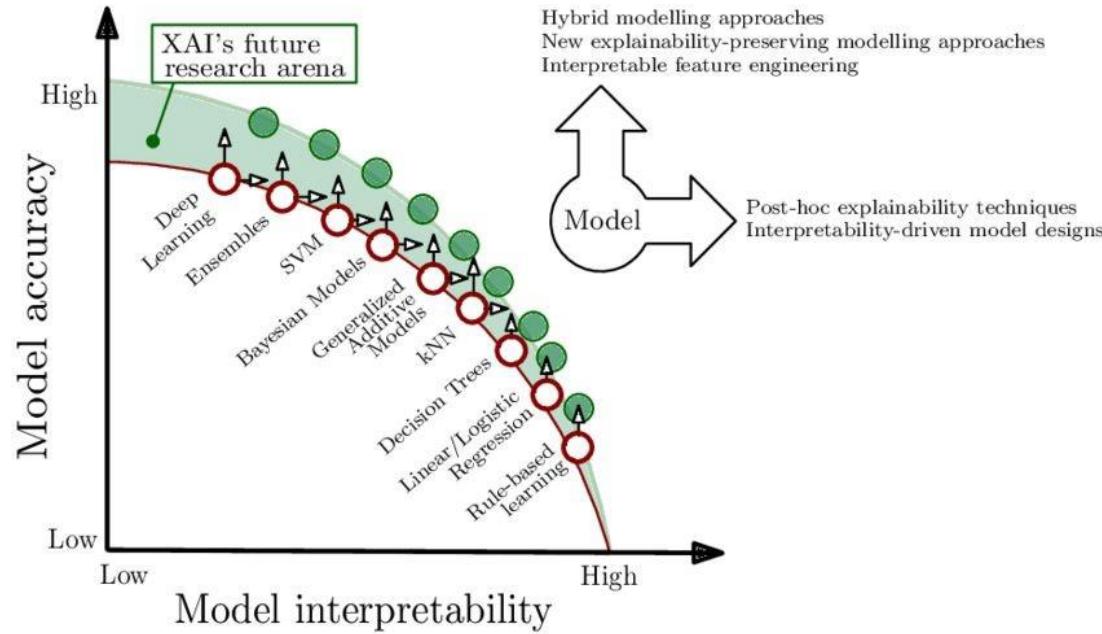
MNIST Dataset



<http://yann.lecun.com/exdb/mnist/>

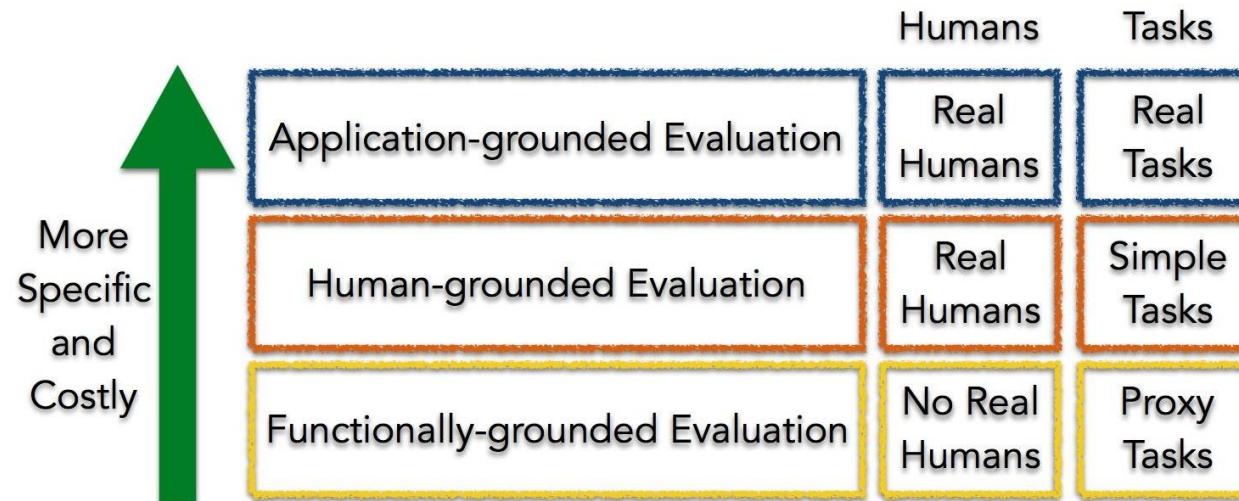
<https://soph.info/2018/11/08/mnist-history/>

Interpretability and Performance Trade-offs ([Arrieta et al., 2019](#))



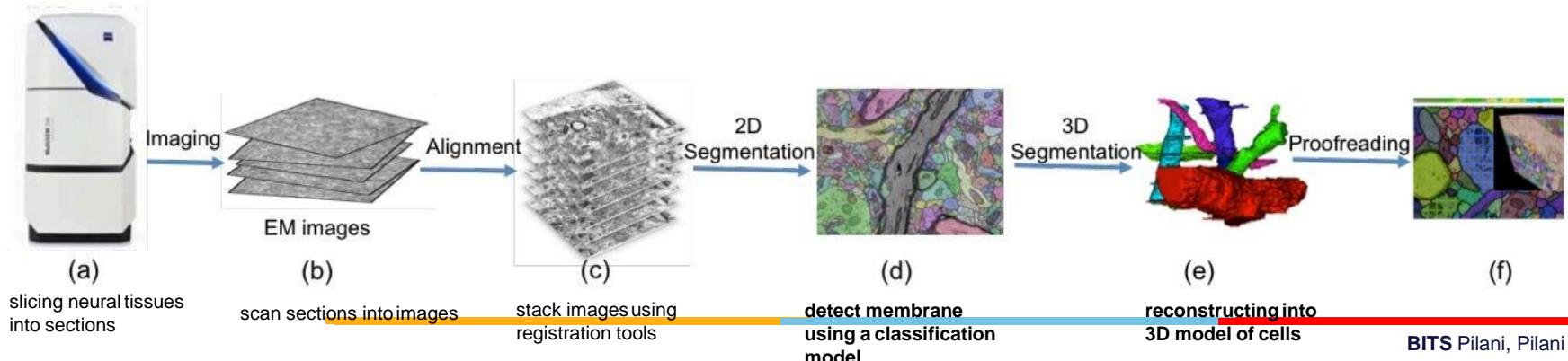
Proxy Models for Post Hoc Interpretability

Evaluations for Interpretability ([Finale Doshi-Velez et al, 2017](#))



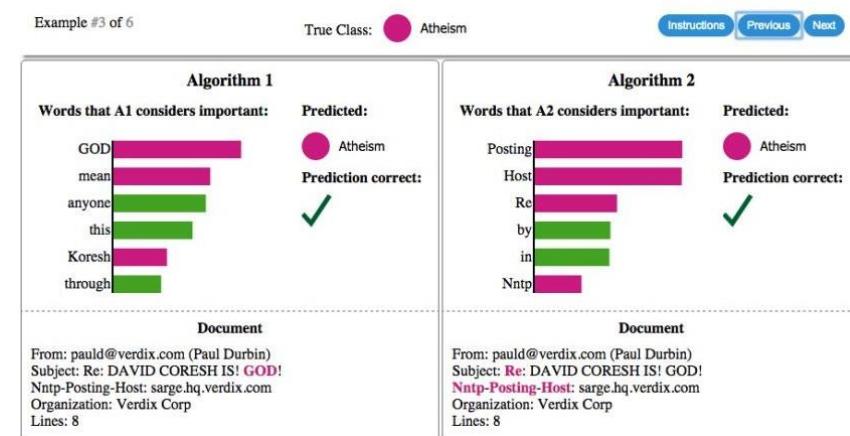
Application-Grounded Evaluation

- Examined by Human Experts in a Specialized Domain
 - Interpretable models need to facilitate conducting a real and sophisticated task
- Automatic Neural Reconstruction from Petavoxel of Electron Microscopy Data ([Suissa-Peleg et al, 2016](#))
 - Study the dense structure of the neurons in the brain and their synapses
 - A multi-step process that involves many ML models



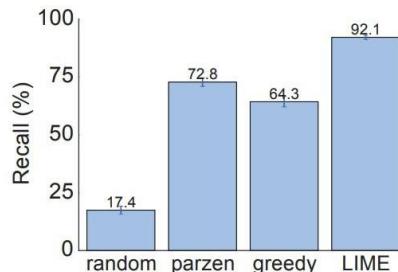
Human-Grounded Evaluation

- Examined by a Lay Human in a General Domain
 - Interpretable models are evaluated by average human.
- Explain a model that classifies an article into either "Christianity" or "Atheism" ([\(Ribeiro et al, 2016\)](#))
 - Amazon Mechanical Turk workers are asked to the algorithm that has better performance

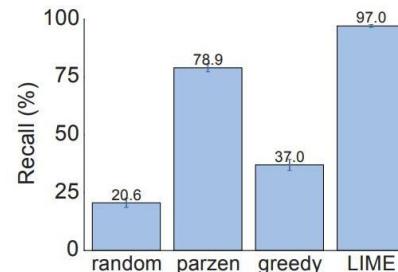


Functionally-Grounded Evaluation

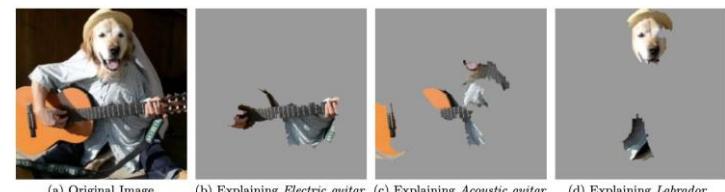
- Examined using a proxy task
- Compare selected feature from model interpretability against explanatory features ([Ribeiro et al, 2016](#))
 - Explanatory feature are labeled by human as ground truth



(a) Sparse LR



(b) Decision Tree



(a) Original Image

(b) Explaining Electric guitar

(c) Explaining Acoustic guitar

(d) Explaining Labrador

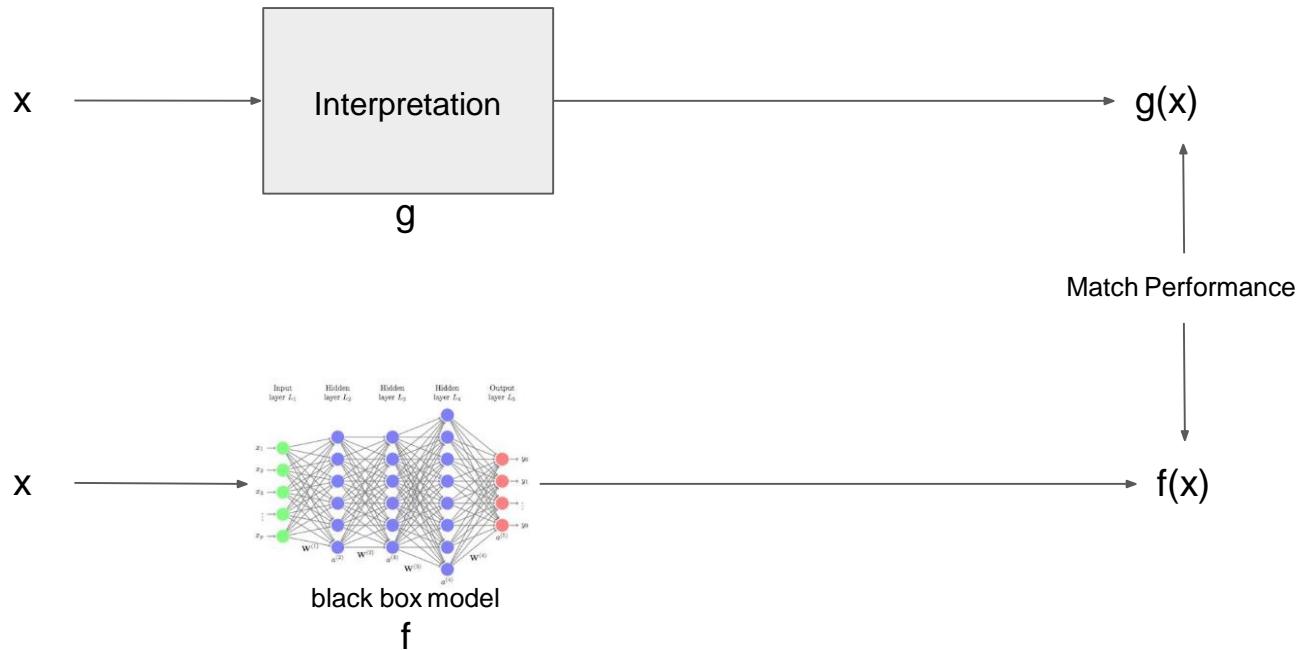
Proxy Methods for Post Hoc Interpretability

- Post Hoc Interpretability
 - Proxy Models
- Local Surrogate Methods
 - LIME
- Rule Based Learner
 - Anchors

Post Hoc Interpretability

- Model Agnostic
 - Can be applied across many different black box models
 - Multiple techniques can be applied at the same time
- Availability
 - Do not require training data
 - Do not require model training/fine-tuning
- No Performance Degeneration
 - Will not alter the black box model

Proxy Models for Post Hoc Interpretability

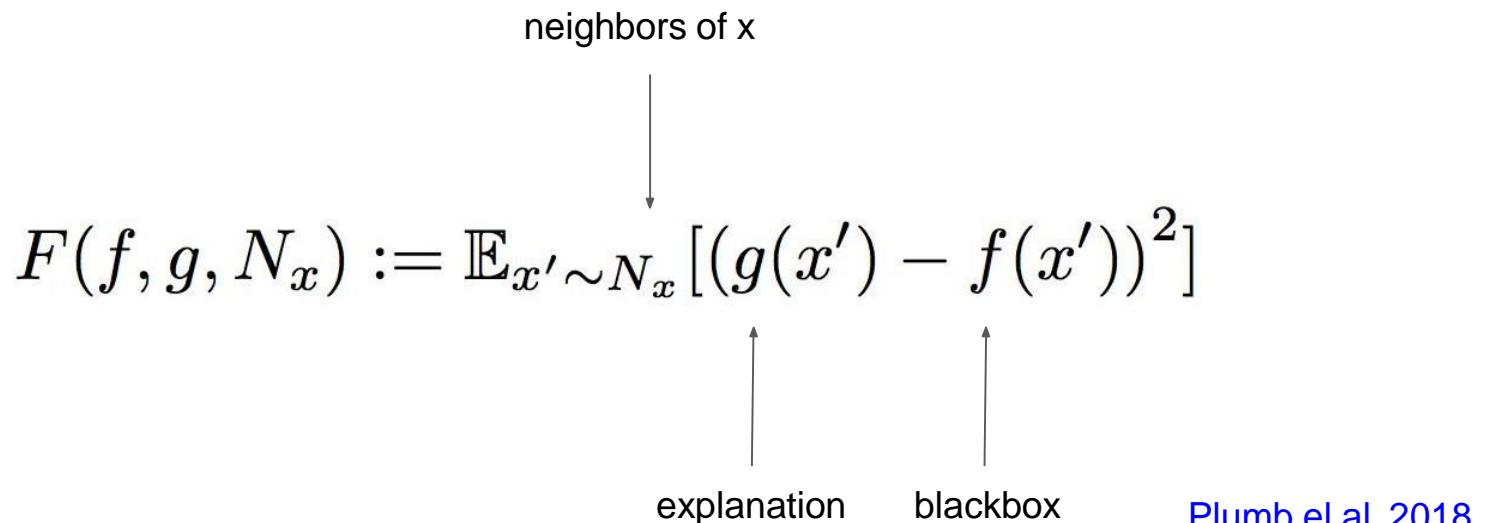


Outline

- Post Hoc Interpretability
 - Proxy Models
- Local Surrogate Methods
 - LIME
- Rule Based Learner
 - Anchors

Local Surrogate Methods

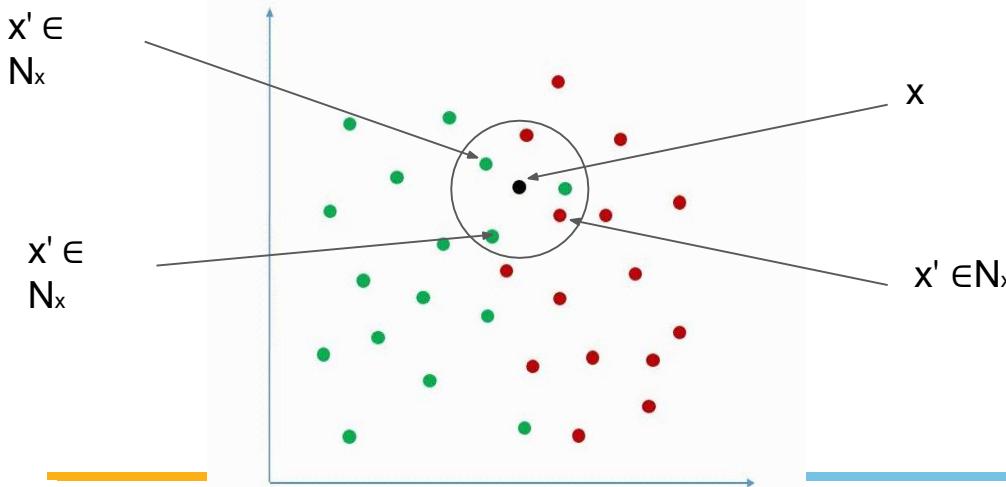
- Local surrogate methods aim at finding explanation g to approximate f around x based on Model Fidelity



Local Surrogate Methods

- Local surrogate methods aim at finding explanation g to approximate f around x based on Model Fidelity

$$F(f, g, N_x) := \mathbb{E}_{x' \sim N_x} [(g(x') - f(x'))^2]$$

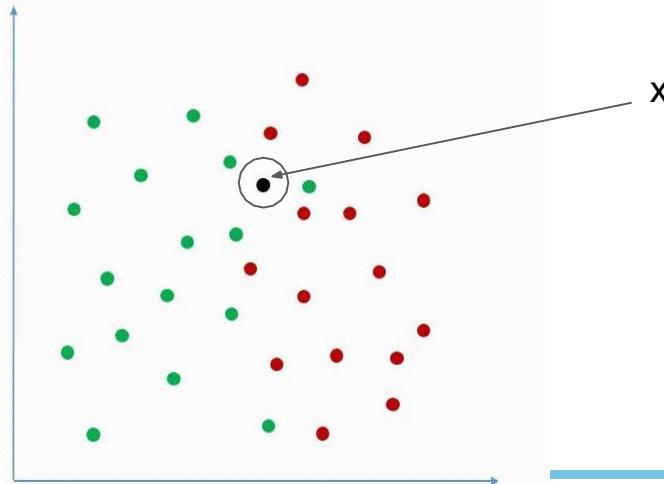


[Plumb et al, 2018](#)

Local Surrogate Methods

- Local surrogate methods aim at finding explanation g to approximate f around x based on Model Fidelity

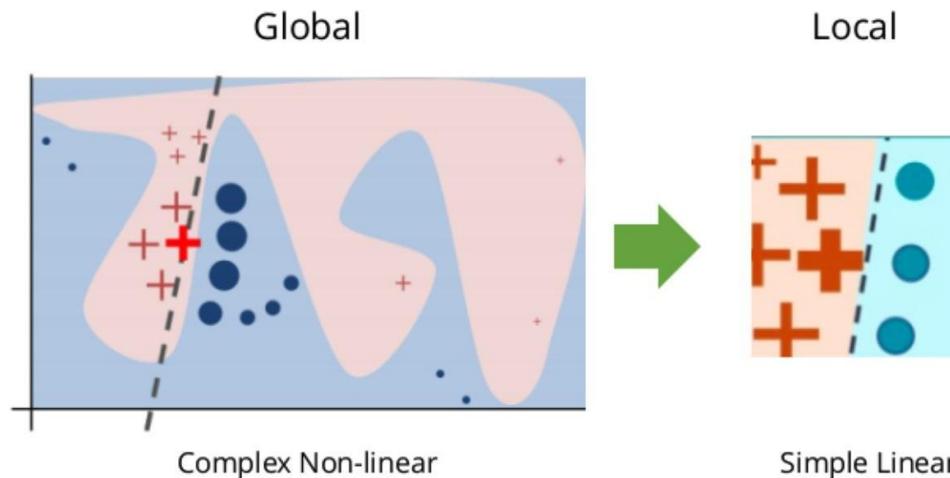
$$F(f, g, N_x) := \mathbb{E}_{x' \sim N_x} [(g(x') - f(x'))^2]$$



[Plumb et al, 2018](#)

Local Interpretable Model-agnostic Explanations (LIME)

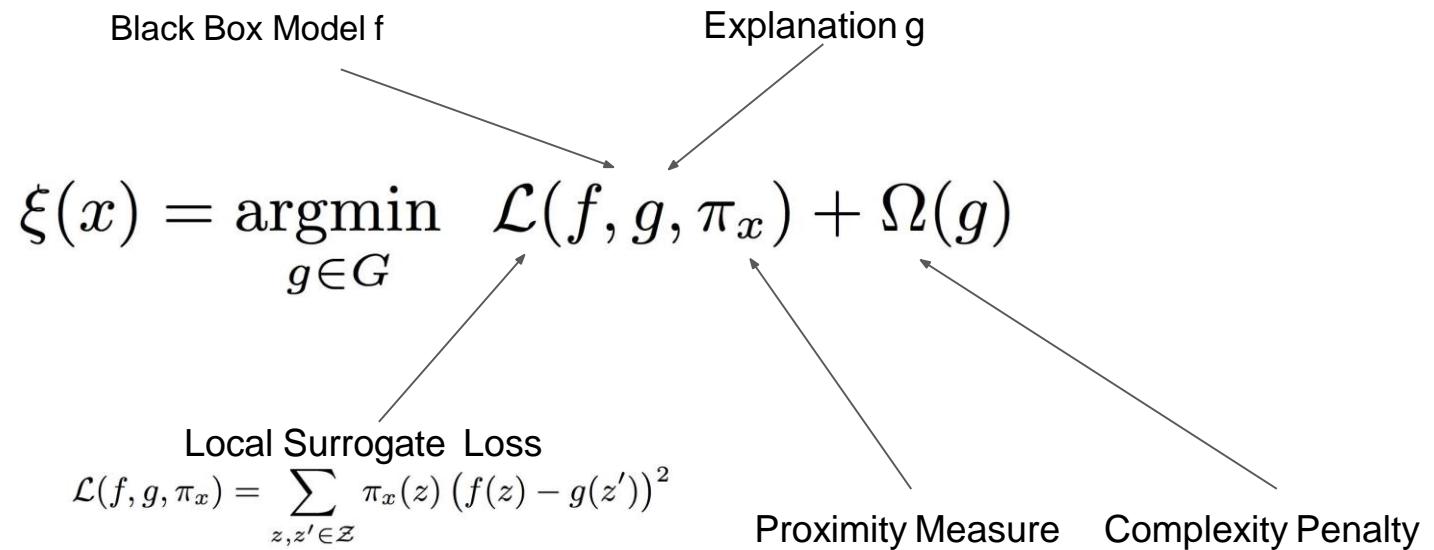
- Deep learning models are usually too complex for global interpretation
 - Instead, we seek for local interpretability using simple interpretable models (e.g. linear models)



[Ribeiro et al, 2016](#)

LIME

- LIME generates an explainable model that optimizes both model fidelity and explanation



Linear Explainable Model

$$\text{Local Surrogate Loss } \mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in \mathcal{Z}} \pi_x(z) (f(z) - g(z'))^2$$

- Linear Explainable Model
 - We use a linear model for explanation $g(z') = w_g \cdot z'$, $z' \in \{0, 1\}^d$

Linear Explainable Model

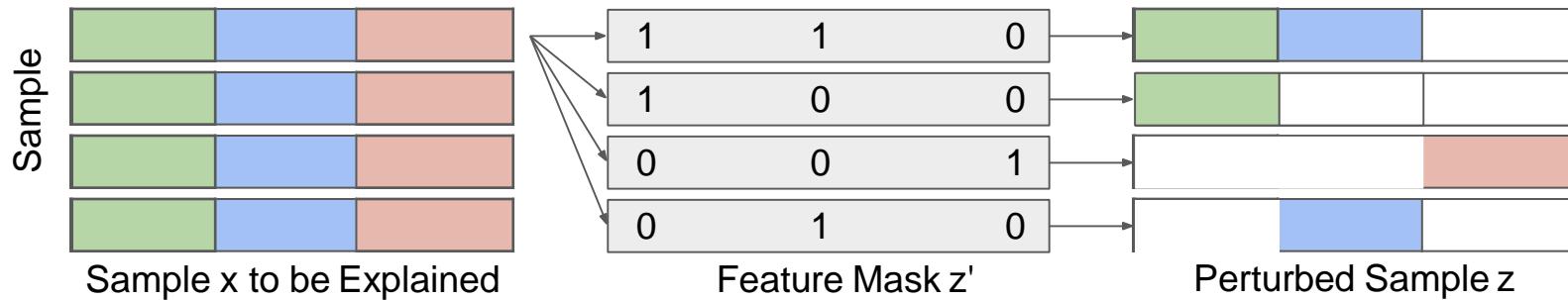
$$\text{Local Surrogate Loss } \mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in \mathcal{Z}} \pi_x(z) (f(z) - g(z'))^2$$

- Linear Explainable Model
 - We use a linear model for explanation $g(z') = w_g \cdot z'$, $z' \in \{0, 1\}^d$
 - z' is a feature mask indicating whether a specific input will be included in the explanation
 - A perturbed sample z can be recovered from mask z' , $z = h_x(z')$

Linear Explainable Model

$$\text{Local Surrogate Loss} \quad \mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in \mathcal{Z}} \pi_x(z) (f(z) - g(z'))^2$$

- Linear Explainable Model
 - We use a linear model for explanation $g(z') = w_g \cdot z'$, $z' \in \{0, 1\}^d$
 - z' is a feature mask indicating whether a specific input will be included in the explanation
 - A perturbed sample z can be recovered from mask z' , $z = h_x(z')$



Training Objective for LIME

- Loss Function
 - Match predictions of the explanation model g with that of the black box model f around x
 - We use an exponentially scaled function to measure proximity
 - D = cosine distance for text
 - D = L2 distance for images

$$\pi_x(z) = \exp(-D(x, z)^2/\sigma^2)$$

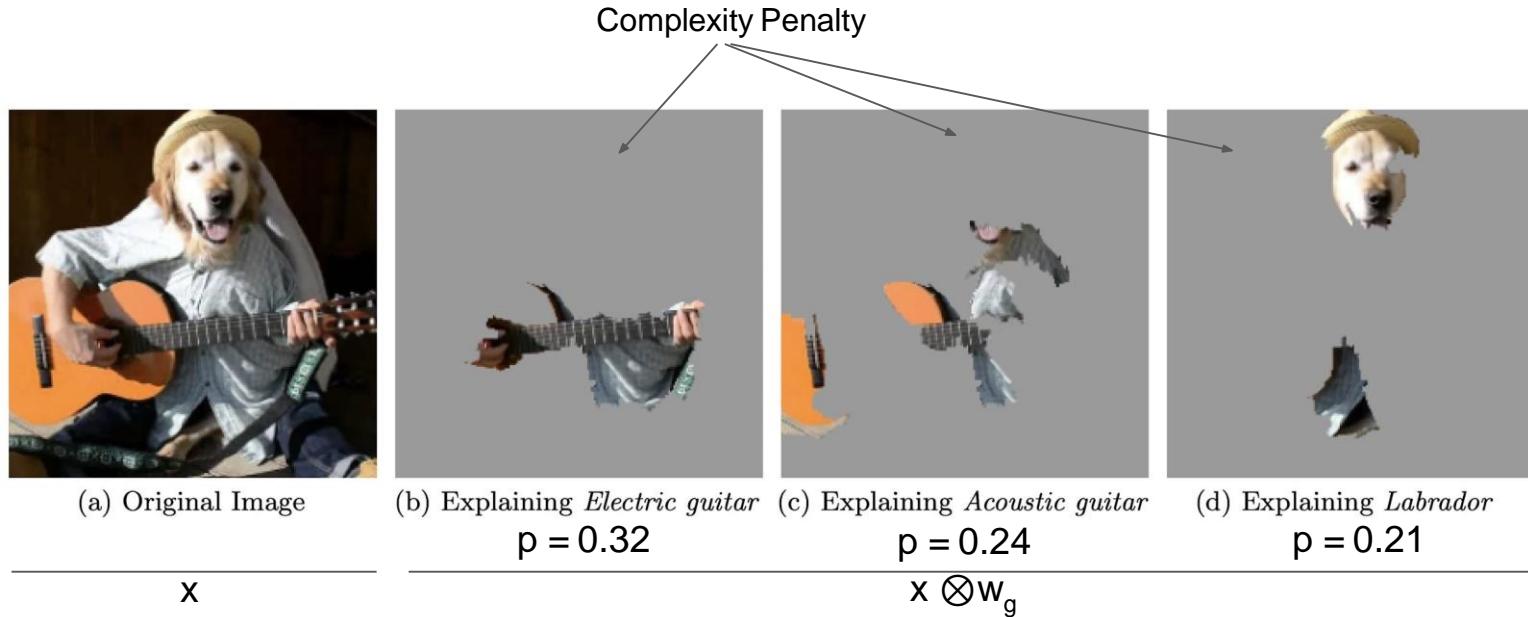
$$\epsilon(x) = \arg \min_{g \in G} \sum_{z, z'} \pi_x(z)(f(z) - g(z'))^2 + \infty \cdot \mathbb{1}_{\|w_g\|_0 > K}$$

Local Surrogate Loss

Complexity Penalty

[Ribeiro et al, 2016](#)

Explaining Google InceptionNet

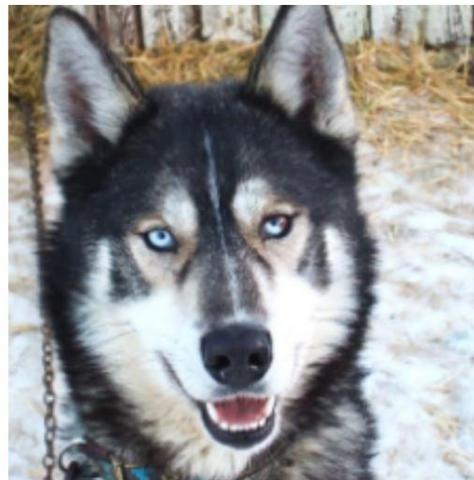


$$\epsilon(x) = \arg \min_{g \in G} \sum_{z, z'} \pi_x(z) (f(z) - g(z'))^2 + \infty \cdot \mathbb{1}_{||w_g||_0 > K}$$

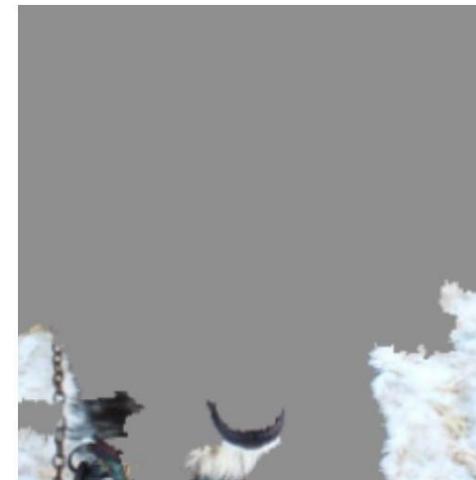
Ribeiro et al, 2016

Example for Bad ML Predictions

- Explanations on a model that misclassified Husky as Wolf



(a) Husky classified as wolf

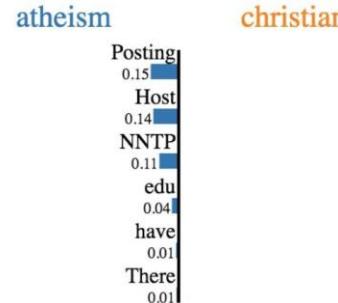
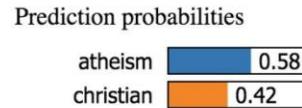


(b) Explanation

[Ribeiro et al, 2016](#)

Explaining Text Classifiers

- Explanations for a SVM classifier with 94% accuracy
 - Predictions are made for arbitrary reasons
 - The word “Posting” appears in 22% of examples in the training set
 - 99% of which are samples attribute to class “Atheism”



Text with highlighted words

From: johnchad@triton.unm.edu (jchadwic)
 Subject: Another request for Darwin Fish
 Organization: University of New Mexico, Albuquerque
 Lines: 11

NNTP-Posting-Host: triton.unm.edu

Hello Gang,

There have been some notes recently asking where to obtain the DARWIN fish.

This is the same question I have and I have not seen an answer on the net. If anyone has a contact please post on the net or email me.

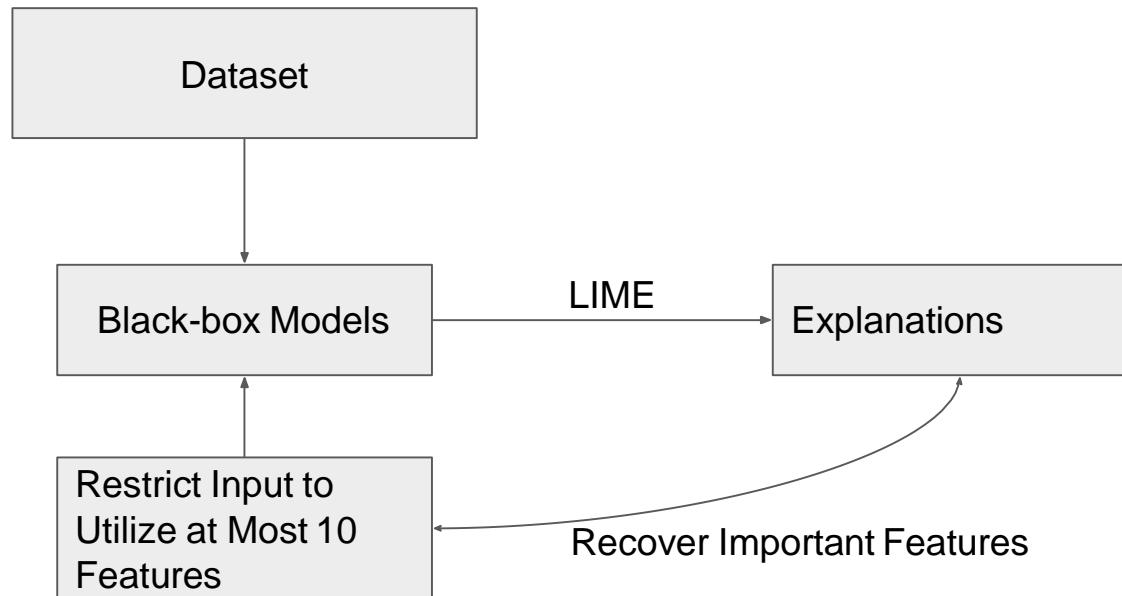
$f(x)$

w_g

$x \otimes w_g$

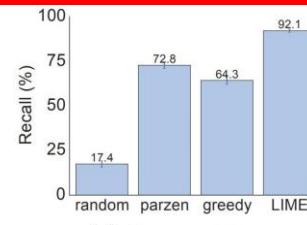
Ribeiro et al, 2016

Faithfulness of Explanations

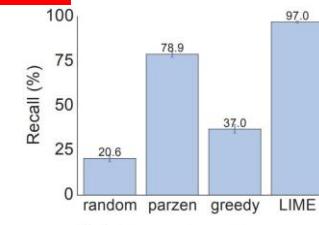


Faithfulness of Explanations

- LIME Achieves Good Faithfulness
- Sentiments classification tasks
 - Books, DVDs
- Classifiers
 - Sparse logistic regression (LR)
 - decision tree

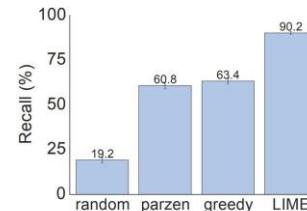


(a) Sparse LR

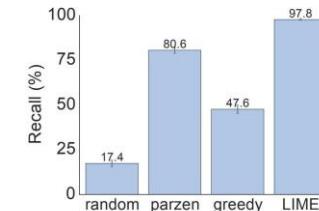


(b) Decision Tree

Books Dataset



(a) Sparse LR



(b) Decision Tree

DVDs Dataset

parzen - [Baehrens et al, 2010](#)

random - randomly pick K features

greedy - remove features contribute most to the classifiers

[Ribeiro el al, 2016](#)

Trustworthiness for ML Models

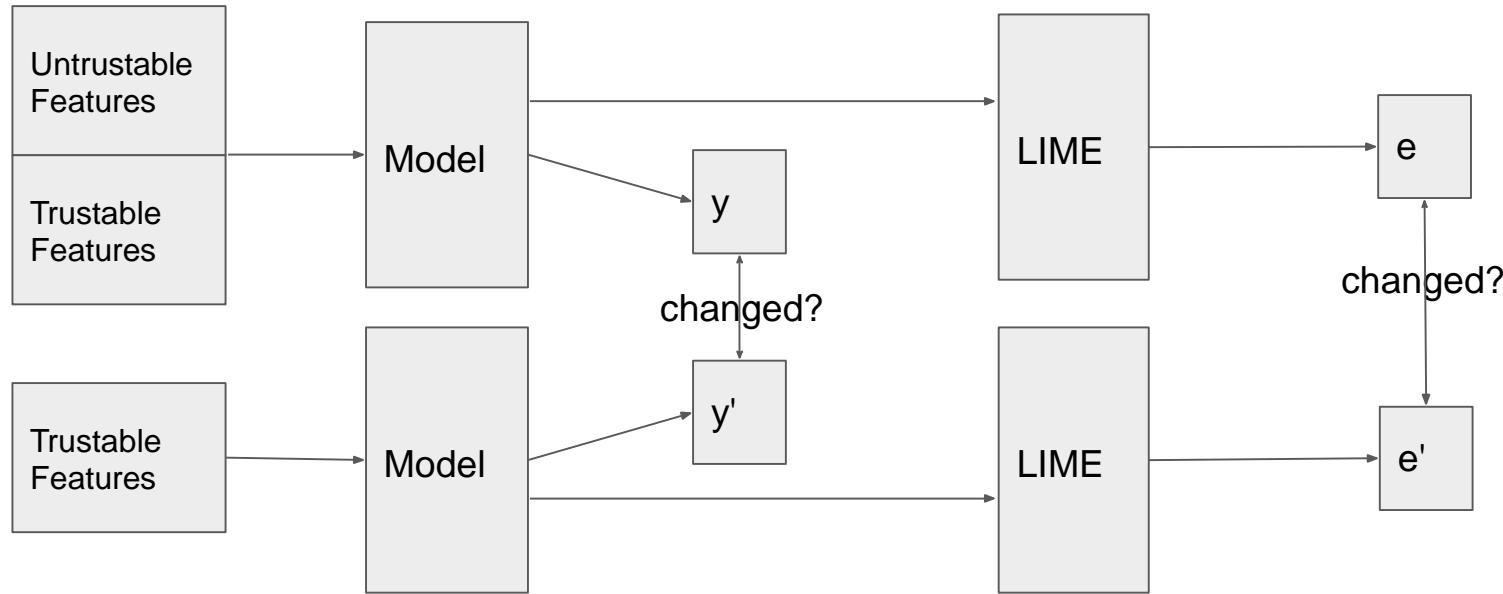
- Human discredits certain features in the learning tasks
- Classifiers that use those features will be considered not trustable.

Predict the need for ICU

heart beat	temperature	salary	Need for ICU?
120 BPM	101 F	\$20,000	N
80 BPM	104.4 F	\$40,000	Y
140 BPM	99 F	\$800,000	Y
110 BPM	100 F	\$30,000	N

Ribeiro et al, 2016

Trustworthiness for Explanations



Trustworthiness of Predictions

- Untrustable Features
 - 25% of features are "untrustable features"
- Trustworthiness of Predictions
 - Compares changes of model predictions and the changes of model explanations when untrustable features are removed

	Books				DVDs			
	LR	NN	RF	SVM	LR	NN	RF	SVM
Random	14.6	14.8	14.7	14.7	14.2	14.3	14.5	14.4
Parzen	84.0	87.6	94.3	92.3	87.0	81.7	94.2	87.3
Greedy	53.7	47.4	45.0	53.3	52.4	58.1	46.6	55.1
LIME	96.6	94.5	96.2	96.7	96.6	91.8	96.1	95.6

Trustworthiness of LIME with different ML models:

- Logistic Regression with L2 regularization (LR)
- Nearest Neighbors (NN)
- Random Forests (RF)
- Support Vector Machines (SVM)

[Ribeiro et al, 2016](#)

Explaining Multiple Samples

- Explain a set of samples to get a complete picture of the model
 - Each sample $x_i \in X$ will have its interpretation

$$g_{x_i}(z) = w_i \cdot z = \sum_j w_{i,j} \cdot z_j$$

- How do we select samples?
 - Select samples to cover the maximum information about the model

$$I_j = \sqrt{\sum_{x_i \in X} |w_{i,j}|}$$

Explaining Multiple Samples

- How do we select samples?
 - Select samples to cover the maximum information about the model

$$I_j = \sqrt{\sum_{x_i \in X} |w_{i,j}|}$$

- Set function

$$c(V, \mathcal{W}, I) = \sum_{j=1}^{d'} \mathbb{1}_{[\exists i \in V : \mathcal{W}_{ij} > 0]} I_j$$

- We want to get a set of samples V up to B elements that maximize c

$$\text{Pick}(\mathcal{W}, I) = \underset{V, |V| \leq B}{\operatorname{argmax}} c(V, \mathcal{W}, I)$$

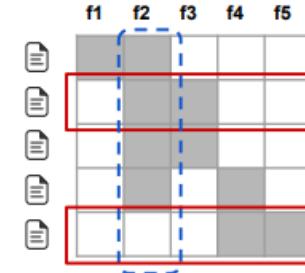
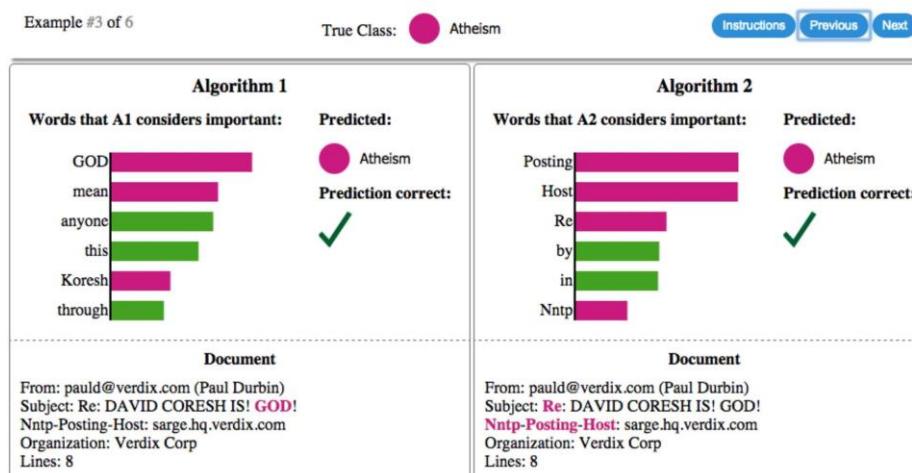


Figure 5: Toy example \mathcal{W} . Rows represent instances (documents) and columns represent features (words). Feature f_2 (dotted blue) has the highest importance. Rows 2 and 5 (in red) would be selected by the pick procedure, covering all but feature f_1 .

Human Experiments

- Ask Human to Select the Best Classifier
 - Annotators are shown the explanations
 - Annotators have no knowledge in machine learning

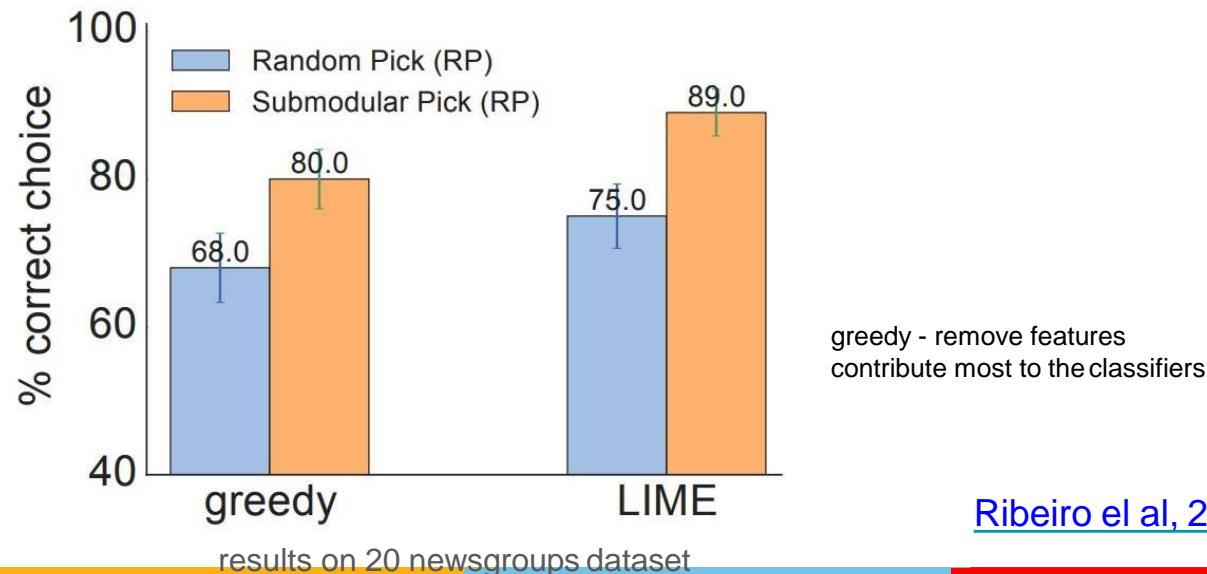


Classification of Atheism/Christian in the 20 newsgroups dataset

[Ribeiro et al, 2016](#)

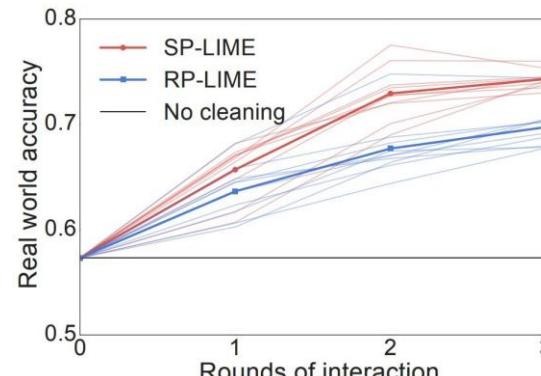
Human Experiments - Select the Best Classifier

- Original model: SVM trained on the dataset with original features
- Cleaned model: SVM trained on the dataset with "cleaned features"



Improving Models Through ML Interpretability

- Improving ML Models
 - Human raters are shown model interpretability
 - They are asked to improve the model by masking out unnecessary features
 - Which words from the explanations should be removed from subsequent training
 - SP - select samples by random
 - RP - select samples by greedy algorithm



[Ribeiro et al, 2016](#)

results on 20 newsgroups dataset

Outline

- Post Hoc Interpretability
 - Proxy Models
- Local Surrogate Methods
 - LIME
- Rule Based Explainers
 - Anchors

Rule Based Explainers

- Explain the Predictions of Deep Learning Models Using Rules
 - How do we find the set of rules for a particular predictor?

	If	Predict
adult	No capital gain or loss, never married	$\leq 50K$
	Country is US, married, work hours > 45	$> 50K$
rcdv	No priors, no prison violations and crime not against property	Not rearrested
	Male, black, 1 to 5 priors, not married, and crime not against property	Re-arrested
lending	FICO score ≤ 649	Bad Loan
	$649 \leq \text{FICO score} \leq 699$ and $\$5,400 \leq \text{loan amount} \leq \$10,000$	Good Loan

[Ribeiro et al, 2018](#)

Anchors

- Generate A Set of Feature Predicates Known as Anchors A (i.e., rules)
 - Using anchors to explain the performance of deep model f
 - mimic the decisions of deep models on x , $f(x)$
 - explain a wide range of similar decisions in the dataset

	If	Predict
adult	No capital gain or loss, never married	$\leq 50K$
	Country is US, married, work hours > 45	$> 50K$
rcdv	No priors, no prison violations and crime not against property	Not rearrested
	Male, black, 1 to 5 priors, not married, and crime not against property	Re-arrested
lending	FICO score ≤ 649	Bad Loan
	$649 \leq \text{FICO score} \leq 699$ and $\$5,400 \leq \text{loan amount} \leq \$10,000$	Good Loan

Anchors found in adult income dataset

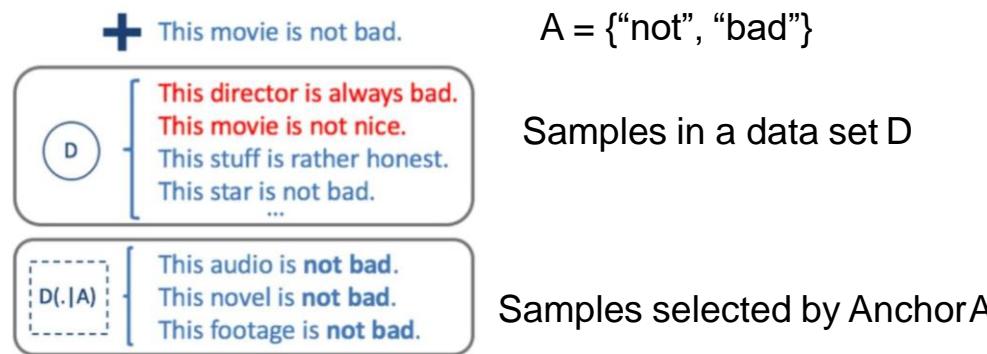
[Ribeiro et al, 2018](#)

Anchors

- An Anchor is a set of feature predicates applied to the feature space

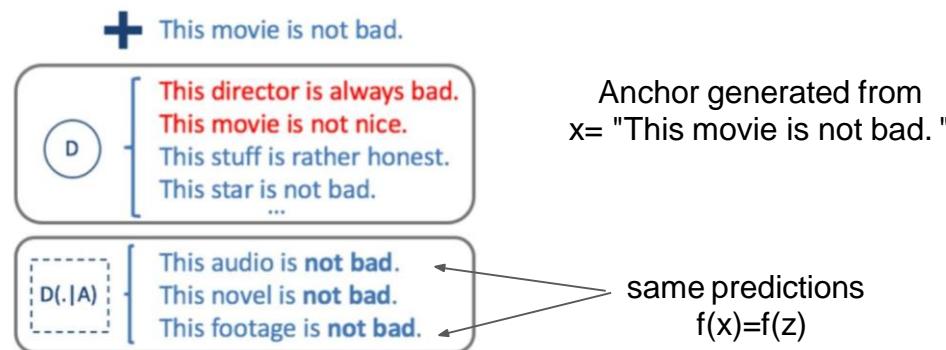
$$A = \{\text{"not"}, \text{"bad"}\}$$

- Any text sample x containing both "not" and "bad" will be selected by the anchor
 $A(x) = 1$
- An anchor can be applied to a dataset D to generate a subset $D(.|A)$



Formal Definitions of Anchors

- Preconditions of Anchors
 - Applies to the sample x being interpreted
 - Precisions
 - Samples covered by the same anchor A need to have the similar predictions
 - i.e., $f(x)=f(z)$ for $z \sim D(\cdot|A)$
 - Coverage
 - A significant portion of the data needs to be covered by Anchor A .



Ribeiro et al, 2018

Formal Definitions of Anchors

- Preconditions of Anchors

- Applies to the sample x being interpreted

$$A(x) = 1$$

- Precision

- Samples covered by the same anchor A need to have similar predictions

$$\mathbb{E}_{\mathcal{D}(z|A)}[\mathbb{1}_{f(x)=f(z)}] \geq \tau$$

- Coverage

- A significant amount of data needs to be covered by one anchor A .

$$\mathbb{E}_{\mathcal{D}(z)} A(z) \geq c$$

[Ribeiro et al, 2018](#)

Anchors for Part of Speech Tagging

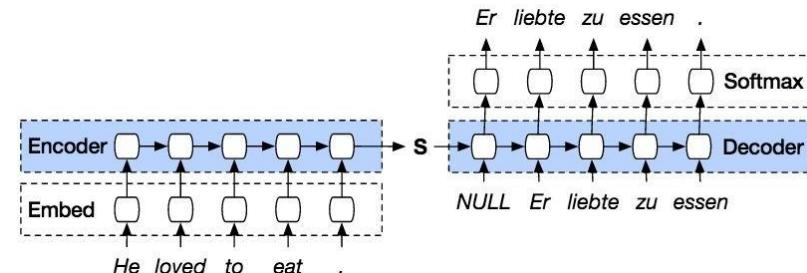
Instance	If	Predict
I want to play(V) ball.	previous word is PARTICLE	play is VERB.
I went to a play(N) yesterday.	previous word is DETERMINER	play is NOUN.
I play(V) ball on Mondays.	previous word is PRONOUN	play is VERB.

[Ribeiro et al, 2018](#)

Anchors for Machine Translation

- Group Predictions of Words with Similar Meanings
 - "esta" (feminine of word "this")
 - "este" (masculine of word "this")
 - "isso" (if its referent is not in the sentence)

English	Portuguese
This is the question we must address	Esta é a questão que temos que enfrentar
This is the problem we must address	Este é o problema que temos que enfrentar
This is what we must address	É isso que temos de enfrentar



Ribeiro et al, 2018

Anchors for Image Classification (InceptionV3)



original image



Anchors for "beagle"

[Ribeiro et al, 2018](#)

Anchors for Visual Question Answering (VQA)



What animal is featured in this picture ? **dog**

What floor is featured in this picture? **dog**

What toenail is paired in this flowchart ? **dog**

What animal is shown on this depiction ? **dog**

Anchor for predicting "dog"

Where is the **dog**?

What color is the **wall**?

When was this picture taken?

Why is he lifting his paw?

on the floor

white

during the day

to play

Other Anchors

[Ribeiro et al, 2018](#)

Generating Anchors

- Preconditions
 - Precision $\text{prec}(A) = \mathbb{E}_{\mathcal{D}(z|A)} [\mathbb{1}_{f(x)=f(z)}]$
 - Coverage $\text{cov}(A) = \mathbb{E}_{\mathcal{D}(z)}[A(z)]$
- Challenges in Generating the Optimal A
 - Calculating precision and coverage is computationally intensive
 - will need to iterate through the predictions of f over the entire dataset
 - Usually difficult to apply white box optimization techniques (e.g., gradient descent)

$$\max_{A \text{ s.t. } P(\text{prec}(A) \geq \tau) \geq 1 - \delta} \text{cov}(A)$$

[Ribeiro et al, 2018](#)

Generating Anchors

- Optimization Target

$$\max_{A \text{ s.t. } P(\text{prec}(A) \geq \tau) \geq 1 - \delta} \text{cov}(A)$$

- Searching for the Optimal A

- for each step t,
 - 1) Construct a set of candidate solutions with the best coverage
 - Candidate solutions need to satisfy $\text{cov}(A) \geq c$
 - 2) Pick top-k candidates with the best precision
 - Candidates need to have $\text{prec}(A) \geq \tau$ with confidence at least $1 - \delta$
 - 3) Update the optimal Anchor A^*

[Ribeiro et al, 2018](#)

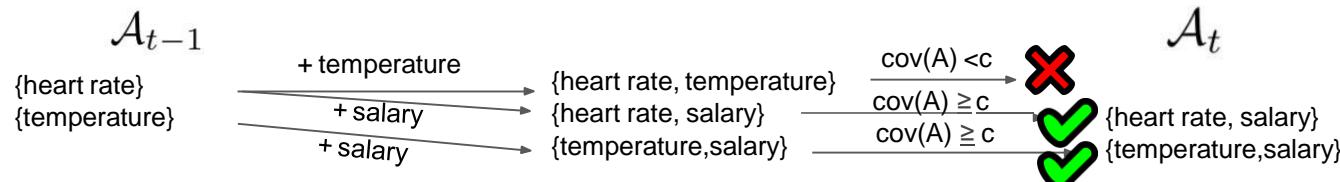
Generating Anchors - Optimizing Coverage

- Searching for the Optimal A
 - 1) Optimizing coverage with $\text{cov}(A) \geq c$
 - 2) Optimizing precision with $\text{prec}(A) \geq \tau$ and confidence at least $1-\delta$
 - 3) Update the optimal solution A^*

$$\text{cov}(A) = \mathbb{E}_{\mathcal{D}(z)}[A(z)]$$

- Optimizing Coverage
 - Start with $\mathcal{A}_0 = \emptyset$
 - Expand \mathcal{A}_{t-1} by one element to get \mathcal{A}_t

heart beat	temperature	salary
120 BPM	101 F	\$20,000
80 BPM	104.4 F	\$40,000
140 BPM	99 F	\$800,000



Ribeiro et al, 2018

Generating Anchors - Optimizing Precisions

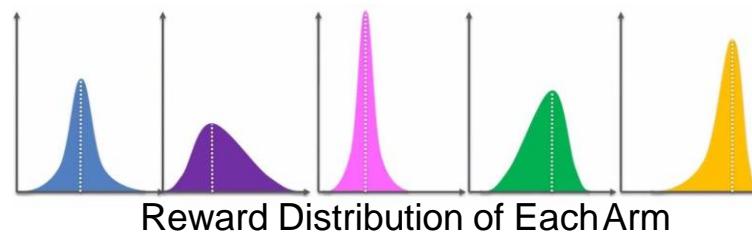
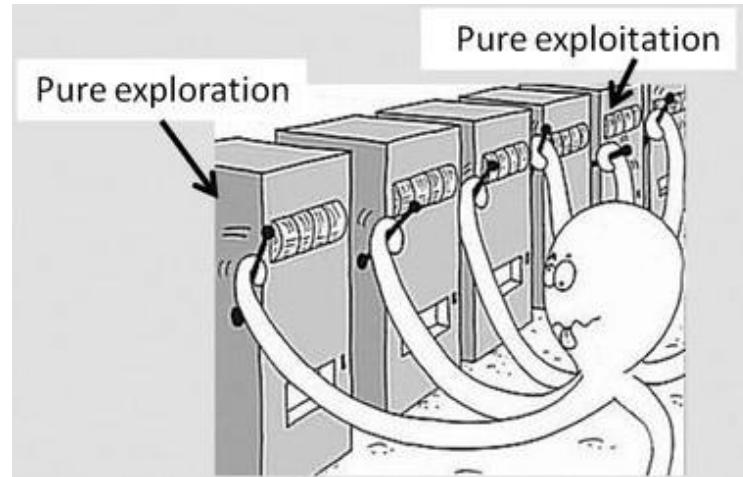
- Searching for the Optimal A
 - 1) Optimizing coverage with $\text{cov}(A) \geq c$
 - 2) Optimizing precision with $\text{prec}(A) \geq \tau$ and confidence at least $1-\delta$
 - 3) Update the optimal solution A^*

$$\text{prec}(A) = \mathbb{E}_{\mathcal{D}(z|A)} [\mathbb{1}_{f(x)=f(z)}]$$

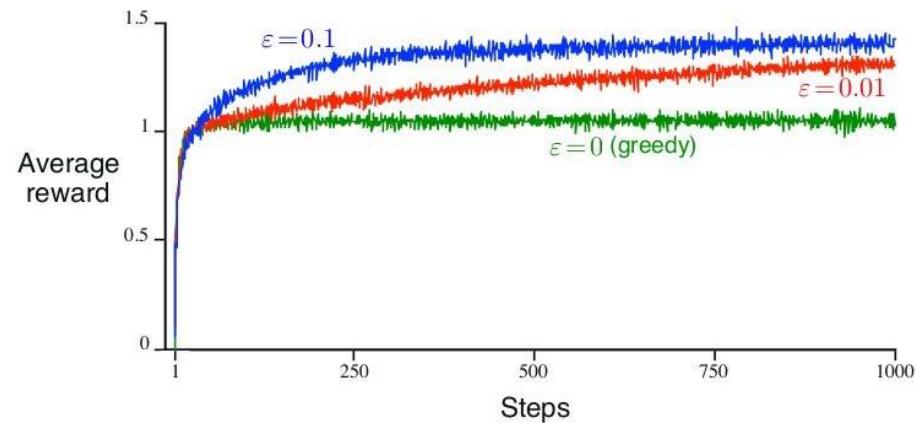
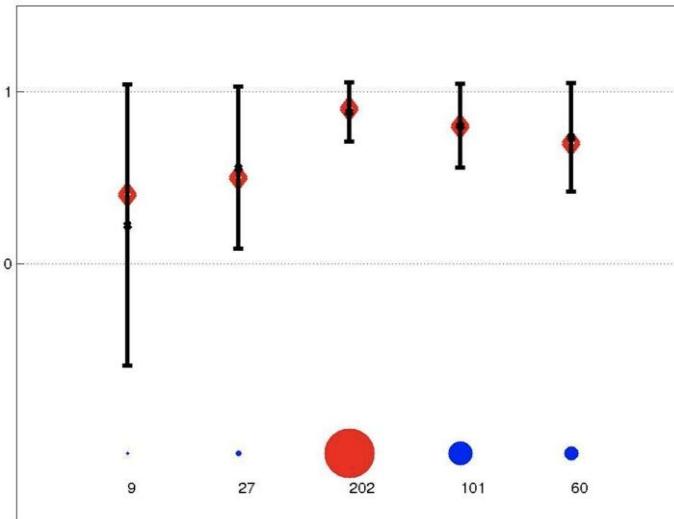
- Optimizing Precisions
 - Formulate it as a Multi-armed bandit optimization problem

Ribeiro et al, 2018

Multi-Armed Bandit Problem



Exploration and Exploitation Trade-offs



Generating Anchors - Optimizing Precisions

- Searching for the Optimal A
 - 1) Optimizing coverage with $\text{cov}(A) \geq c$
 - 2) Optimizing precision with $\text{prec}(A) \geq \tau$ and confidence at least $1-\delta$
 - 3) Update the optimal solution A^*

$$\text{prec}(A) = \mathbb{E}_{\mathcal{D}(z|A)} [\mathbb{1}_{f(x)=f(z)}]$$

- Optimizing Precisions
 - Formulate it as a Multi-armed bandit optimization problem
 - Find out candidates with $\text{Prec}(A) \geq \tau$
 - Using minimal costs (number of pulls of the arms)
 - Each candidate solution A is an arm
 - $\text{Prec}(A)$ of a single sample is the latent reward

Ribeiro et al, 2018

Generating Anchors - Optimizing Precisions

- Searching for the Optimal A
 - 1) Optimizing coverage with $\text{cov}(A) \geq c$
 - 2) Optimizing precision with $\text{prec}(A) \geq \tau$ and confidence at least $1-\delta$
 - 3) Update the optimal solution A^*

$$\text{prec}(A) = \mathbb{E}_{\mathcal{D}(z|A)} [\mathbb{1}_{f(x)=f(z)}]$$

- Optimizing Precisions
 - Formulate it as a Multi-armed bandit optimization problem
 - Find out candidates with $\text{Prec}(A) \geq \tau$
 - Using minimal costs (number of pulls of the arms)
 - Each candidate solution A is an arm
 - $\text{Prec}(A)$ of a single sample is the latent reward
 - Return the top K arms (i.e., A) with the highest reward ($\text{Prec}(A)$) that satisfies conditions
 - $\text{Prec}(A) \geq \tau, P(\text{Prec}(A) \geq \tau) \geq 1-\delta$

[Ribeiro et al, 2018](#)

Generating Anchors - Update Optimal A*

- Searching for the Optimal A
 - 1) Optimizing coverage with $\text{cov}(A) \geq c$
 - 2) Optimizing precision with $\text{prec}(A) \geq \tau$ and confidence at least $1-\delta$
 - 3) Update the optimal solution A^*
- Update A*
 - For the top-k A returned in step 2)
 - Find the best A^* based on the Coverage criteria
 - Loop into the next step

if $\text{cov}(A) > \text{cov}(A^*)$ **then** $A^* \leftarrow A$

Precision and Coverage

- Precision

$$\text{prec}(A) = \mathbb{E}_{\mathcal{D}(z|A)} [\mathbb{1}_{f(x)=f(z)}]$$

- Coverage

$$\text{cov}(A) = \mathbb{E}_{\mathcal{D}(z)}[A(z)]$$

- Limes

- lime-n - Naive LIME algorithm
- lime-t - Make predictions only when its predictive probability is above a threshold

		Precision		Coverage	
		anchor	lime-n	anchor	lime-t
adult	logistic	<u>95.6</u>	<u>81.0</u>	<u>10.7</u>	<u>21.6</u>
	gbt	<u>96.2</u>	<u>81.0</u>	<u>9.7</u>	<u>20.2</u>
	nn	<u>95.6</u>	<u>79.6</u>	<u>7.6</u>	<u>17.3</u>
rcdv	logistic	<u>95.8</u>	<u>76.6</u>	<u>6.8</u>	<u>17.3</u>
	gbt	<u>94.8</u>	<u>71.7</u>	<u>4.8</u>	<u>2.6</u>
	nn	<u>93.4</u>	<u>65.7</u>	<u>1.1</u>	<u>1.5</u>
lending	logistic	<u>99.7</u>	<u>80.2</u>	<u>28.6</u>	<u>12.2</u>
	gbt	<u>99.3</u>	<u>79.9</u>	<u>28.4</u>	<u>9.1</u>
	nn	<u>96.7</u>	<u>77.0</u>	<u>16.6</u>	<u>5.4</u>

logistic: logistic regression, gbt: gradient boosted trees

nn: two layers of 50 units MLP

[Ribeiro et al, 2018](#)

User Study

- Ask Users to Guess the Outcomes of A ML Model After Explanations
 - Human annotators are 26 students who took a machine learning course
 - Calculate precision and coverage of the users' performance
 - Human mark "I don't know" when they are not certain, which makes coverage the perceived one.

Method	Precision				Coverage (perceived)				Time/pred (seconds)			
	adult	rcdv	vqa1	vqa2	adult	rcdv	vqa1	vqa2	adult	rcdv	vqa1	vqa2
No expls	<u>54.8</u>	<u>83.1</u>	<u>61.5</u>	<u>68.4</u>	<u>79.6</u>	<u>63.5</u>	<u>39.8</u>	<u>30.8</u>	<u>29.8</u> ±14	<u>35.7</u> ±26	<u>18.7</u> ±20	<u>13.9</u> ±20
LIME(1)	<u>68.3</u>	<u>98.1</u>	<u>57.5</u>	<u>76.3</u>	<u>89.2</u>	<u>55.4</u>	<u>71.5</u>	<u>54.2</u>	<u>28.5</u> ±10	<u>24.6</u> ±6	<u>8.6</u> ±3	<u>11.1</u> ±8
Anchor(1)	<u>100.0</u>	97.8	<u>93.0</u>	<u>98.9</u>	<u>43.1</u>	<u>24.6</u>	<u>31.9</u>	<u>27.3</u>	<u>13.0</u> ±4	<u>14.4</u> ±5	<u>5.4</u> ±2	<u>3.7</u> ±1
LIME(2)	89.9	<u>72.9</u>	-	-	<u>78.5</u>	<u>63.1</u>	-	-	<u>37.8</u> ±20	<u>24.4</u> ±7	-	-
Anchor(2)	87.4	<u>95.8</u>	-	-	<u>62.3</u>	<u>45.4</u>	-	-	<u>10.5</u> ±3	<u>19.2</u> ±10	-	-

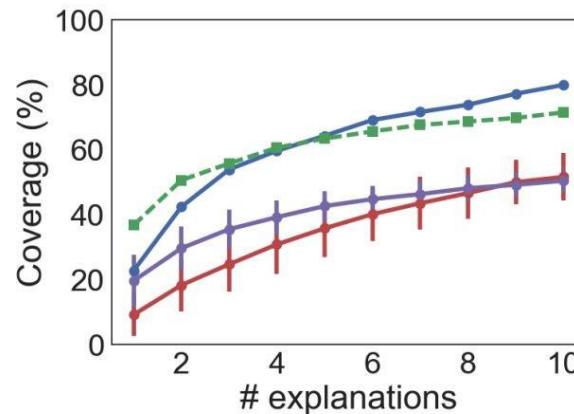
LIME(n): results after n LIME explanations

Anchor(n): results after n Anchore explanations

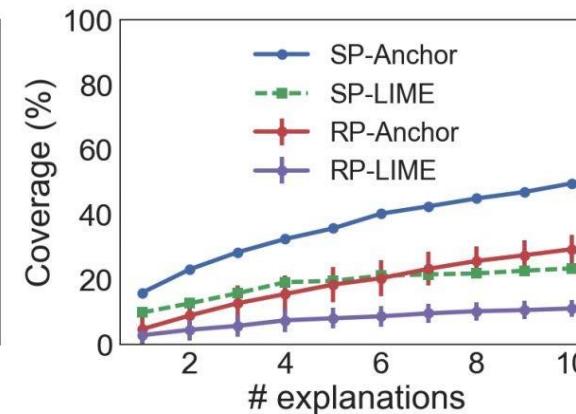
[Ribeiro et al, 2018](#)

User Study Results

- Coverage change with number of explanations seen by the same user.
 - gradient boosted trees(gb)
 - SP - Submodular Pick
 - RP - Random Plck



(a) *adult* dataset



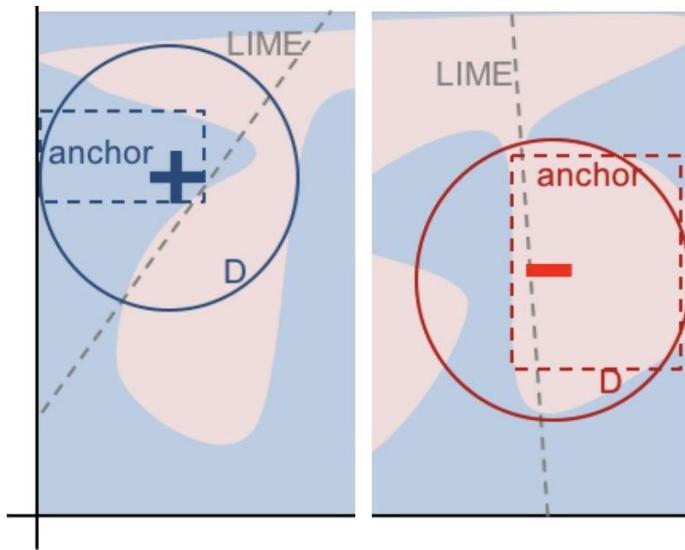
(b) *rcdv* dataset

[Ribeiro et al, 2018](#)

Comparisons to LIME

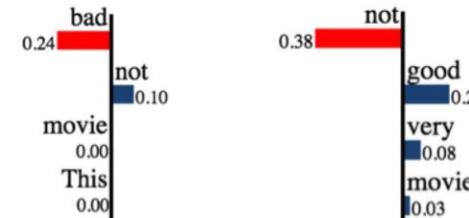
	LIME	Anchors
Explanations	$g(z') = w_g \cdot z'$	Anchors A
Optimization Target	$\mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in \mathcal{Z}} \pi_x(z) (f(z) - g(z'))^2$	$\max_{A \text{ s.t. } P(\text{prec}(A) \geq \tau) \geq 1 - \delta} \text{cov}(A)$

Comparisons to LIME



⊕ This movie is not bad. ─ This movie is not very good.

(a) Instances



(b) LIME explanations

{"not", "bad"} → Positive {"not", "good"} → Negative

(c) Anchor explanations

Ribeiro et al, 2018

Overly Specific Anchors

$28 < \text{Age} \leq 37$

Workclass = Private

Education = High School grad

Marital Status = Married

Occupation = Blue-Collar

Relationship = Husband

Race = White

Sex = Male

Capital Gain = None

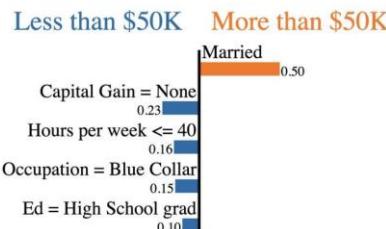
Capital Loss = Low

Hours per week ≤ 40.00

Country = United-States

$$P(\text{Salary} > \$50K) = 0.57$$

(a) Instance and prediction



(b) LIME explanation

**IF Country = United-States AND Capital Loss = Low
AND Race = White AND Relationship = Husband
AND Married AND $28 < \text{Age} \leq 37$
AND Sex = Male AND High School grad
AND Occupation = Blue-Collar
THEN PREDICT Salary > \$50K**

(c) An *anchor* explanation

[Ribeiro et al, 2018](#)

Welcome!!



BITS Pilani
Pilani Campus

Fair, Accountable, Transparent Machine Learning (FAccT ML) ZG517

Dr. Sugata Ghosal
sugata.ghosal@pilani.bits-pilani.ac.in



Session 11
Date – 10th September 2023
Time – 8:45 AM to 10:45 PM

These slides are prepared by the instructor, with grateful acknowledgement of many others who made their course materials freely available online.

Outline

- Shapley Additive Explanations (SHAP)
 - Coalitional Game and Shapley Values
 - Kernel SHAP
 - Tree SHAP
- ✓ ○ Layerwise Relevance Propagation
- ✓ ○ DeepLift
- "Deep SHAP"*
- features*
- Important*

$$y = \boxed{c} + mx$$

Shapley Additive Explanations (SHAP)

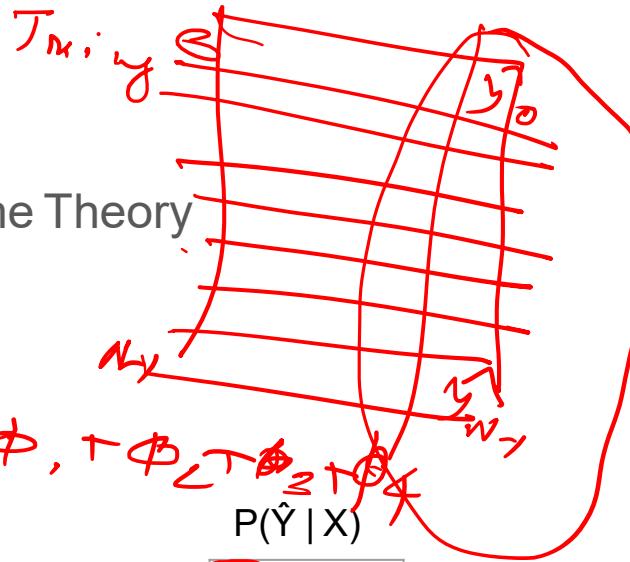
players

- Assigns Feature Importance Weights Based on Game Theory
 - Each feature is a player
 - Probability $P(\hat{Y} | X)$ is the total payoff
 - Distribute the total payoff to players (features) "fairly"

$\Phi_0 \rightarrow$ average players
constant

	Φ_0	Φ_1	Φ_2	Φ_3	Φ_4
+	0.6 ✓	0.05 ✓	0.03 ✓	0.01 ..	0.01 ..
→	0.1 ..	0.2 ..	0.3 ..	-0.1 ..	0 ..
-	0.2	0.1	0.1	0.2	-0.1 ..
—	0.05	0.10	0.05	0.1	-0.1 ..

$$0.7 = \Phi_0 + \Phi_1 + \Phi_2 + \Phi_3 + \Phi_4$$



$$\sum \Phi_i = P(\hat{Y} | X)$$

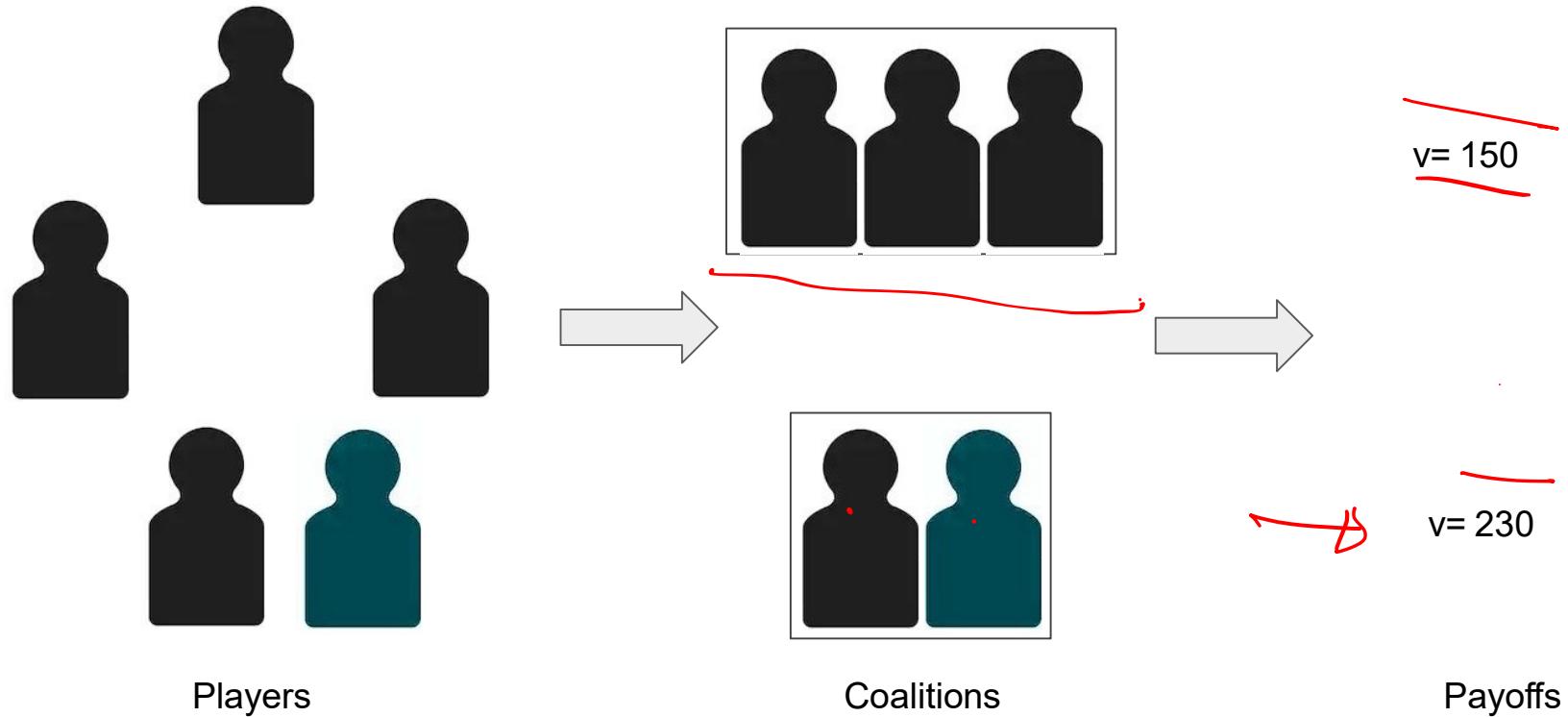
0.7
0.4
0.5
0.2

Shapley Values

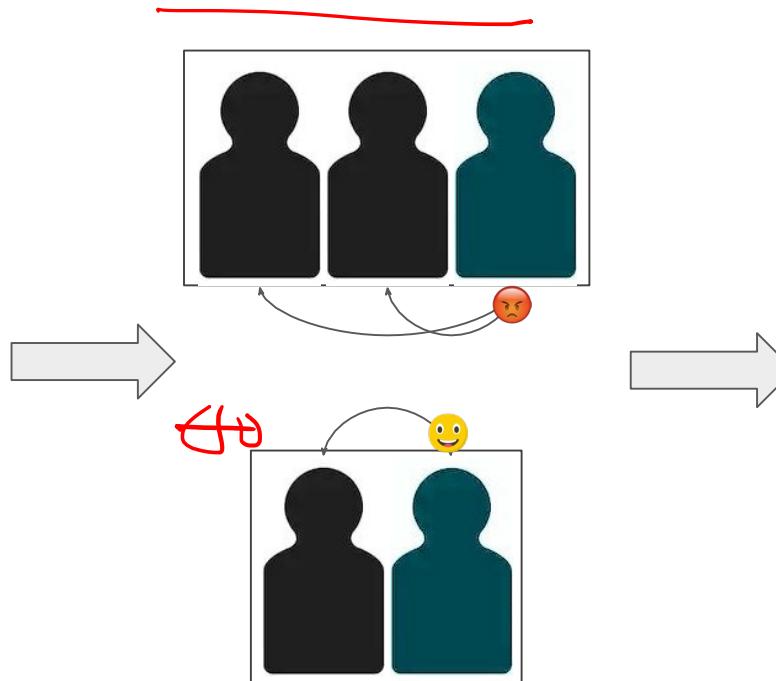
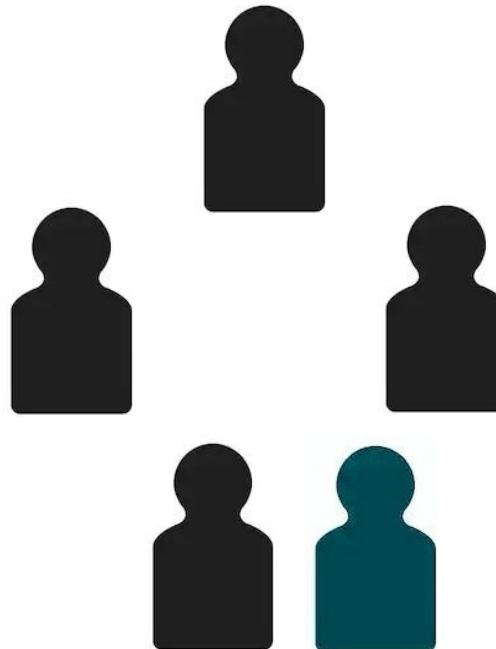
- Developed by Lloyd Shapley
 - American mathematician
 - Nobel Prize-winning economist



Coalitional Game



Coalitional Game



$v = -20$

$v = 230$

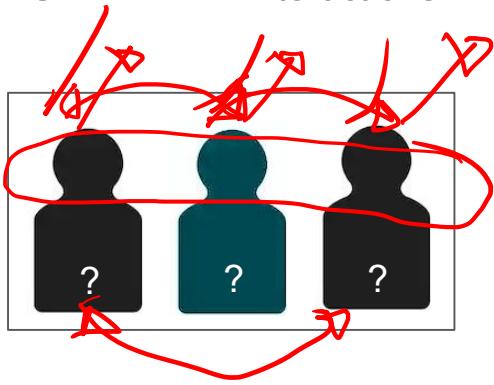
Players

Coalitions

Payoffs

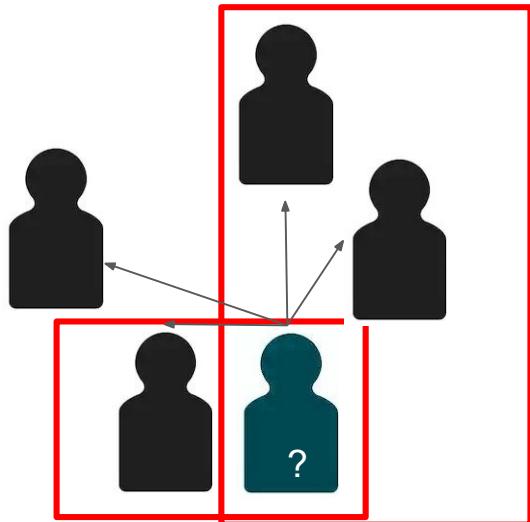
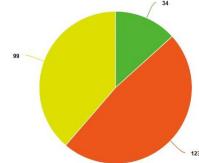
Coalitional Game

- How Do We Assign Importance Scores to Players?
 - Consider the *interactions* to all other players



How much value should we attribute to each player?

$$v = 100$$



How do we account for the interactions among players?

Shapley Values

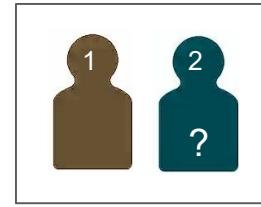
Kernel Shap / Tree Shap

- Calculate Shapley Value for Player 2

$$\frac{1}{M} \binom{M-1}{S}$$

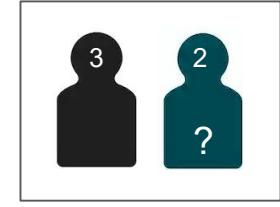
$$\begin{aligned}\phi_j &= \frac{1}{M} \sum_{S \subseteq M \setminus \{j\}} \binom{M-1}{S}^{-1} (v(S \cup \{j\}) - v(S)) \\ &= \sum_{S \subseteq M \setminus \{j\}} \frac{S!(M-S-1)!}{M!} (v(S \cup \{j\}) - v(S))\end{aligned}$$

$$\frac{1}{M} \binom{M-1}{S}^{-1} = \frac{1}{M} \left(\frac{(M-1)!}{S!(M-S-1)!} \right)^{-1} = \frac{S!(M-S-1)!}{M \cdot (M-1)!}$$



$$\binom{M-1}{S}^{-1} = \binom{2}{1}^{-1} = \frac{1}{2}$$

$$v(\{1, 2\}) - v(\{1\})$$



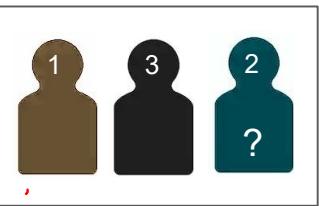
$$\binom{M-1}{S}^{-1} = \binom{2}{1}^{-1} = \frac{1}{2}$$

$$v(\{3, 2\}) - v(\{3\})$$



$$\binom{M-1}{S}^{-1} = \binom{2}{0}^{-1} = 1$$

$$v(\{2\}) - v(\emptyset)$$

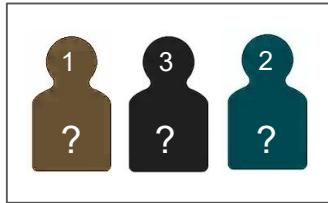


$$\binom{M-1}{S}^{-1} = \binom{2}{2}^{-1} = 1$$

$$v(\{1, 2, 3\}) - v(\{1, 3\})$$

Shapley Values

$$\phi_j = \sum_{S \subseteq \mathcal{M} \setminus \{j\}} \frac{S!(M-S-1)!}{M!} (v(\mathcal{S} \cup \{j\}) - v(\mathcal{S}))$$



~~ϕ_1~~ = $\frac{1}{3} (v(\{1,2,3\}) - v(\{2,3\})) + \frac{1}{6} (v(\{1,2\}) - v(\{2\})) + \frac{1}{6} (v(\{1,3\}) - v(\{3\})) + \frac{1}{3} (v(\{1\}) - v(\emptyset))$

~~$\checkmark \phi_2$~~ = $\frac{1}{3} (v(\{1,2,3\}) - v(\{1,3\})) + \frac{1}{6} (v(\{1,2\}) - v(\{1\})) + \frac{1}{6} (v(\{2,3\}) - v(\{3\})) + \frac{1}{3} (v(\{2\}) - v(\emptyset))$

~~$\checkmark \phi_3$~~ = $\frac{1}{3} (v(\{1,2,3\}) - v(\{1,2\})) + \frac{1}{6} (v(\{1,3\}) - v(\{1\})) + \frac{1}{6} (v(\{2,3\}) - v(\{2\})) + \frac{1}{3} (v(\{3\}) - v(\emptyset))$

~~$\checkmark \phi_0$~~ = $v(\emptyset)$

[Aas et al, 2019](#)

Shapley Value



~~FM~~

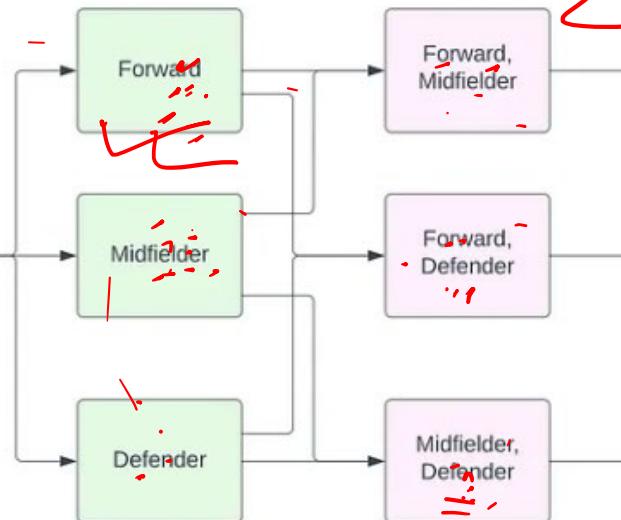
No features at a time

1 feature at a time

2 features at a time

All 3 features at a time

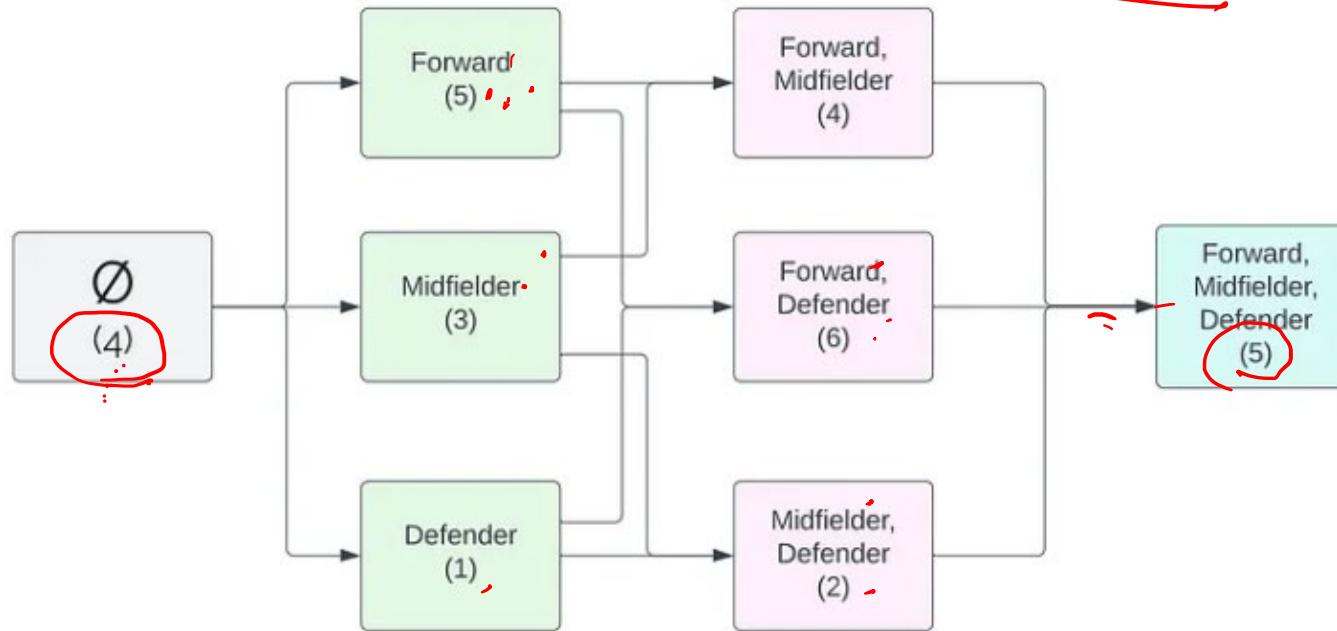
players
Different combination of features (i.e. players info)

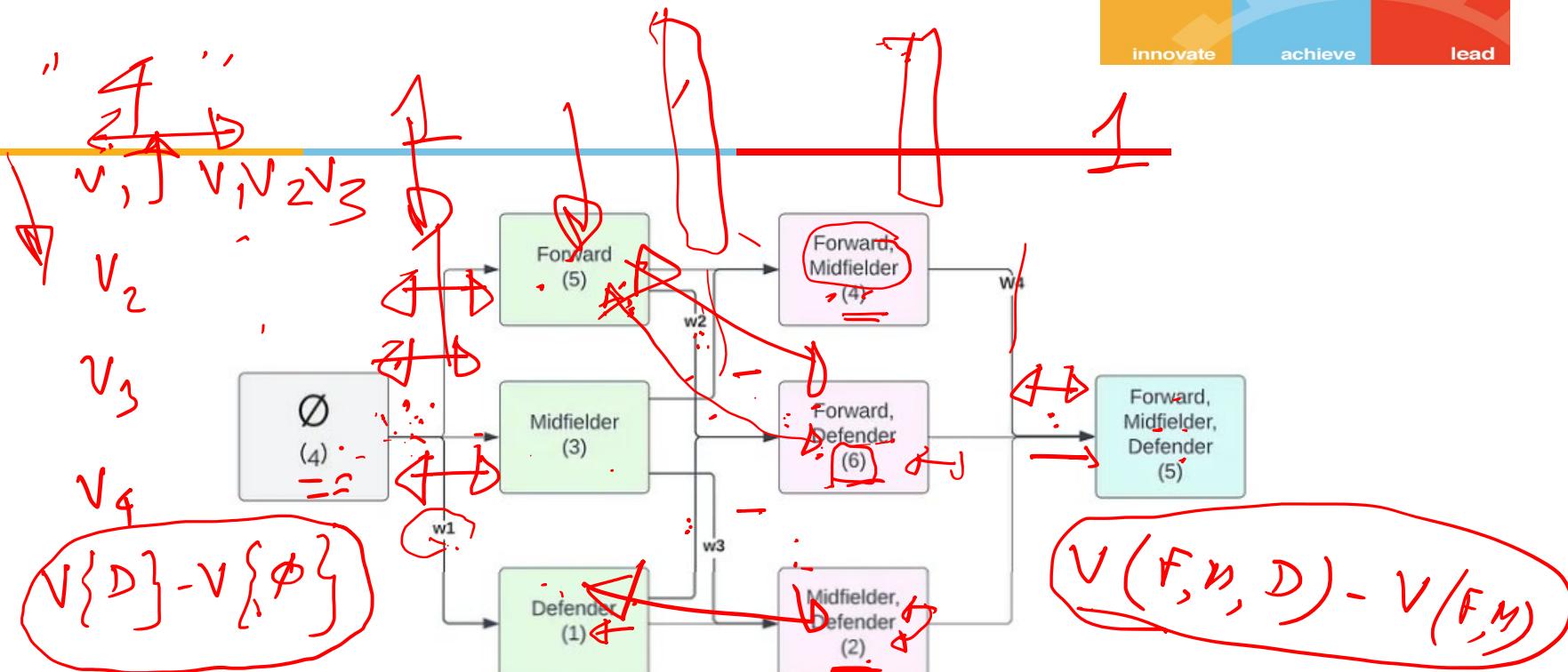


Original goal
→ 25 Regressor

$$2^{4096} \cdot 64 \times 64 = 4096$$

$$\begin{aligned}
 & 2^{4096} \\
 & - 4096C_1 \\
 & + 4096C_2 \\
 & - 4096C_3 \\
 & + \dots \\
 & + 4096C_{4095}
 \end{aligned}$$





SHAP value for defender = $w_1(1-4) + w_2(6-5) + w_3(2-3) + w_4(5-4)$

where $w_1 + w_2 + w_3 + w_4 = 1$

'equity'

1. Since its weighted average, sum of weights should equal to 1.

i.e. $w_1 + w_2 + w_3 + w_4 = 1$

2. the sum of the weights of all MC to 1-feature-models equals the sum of the weights of all the MC to 2-feature-models and so on....

i.e. ~~$w_1 = w_2 + w_3 = w_4$~~

3. All the weights of MC to f-feature-models should equal to each other, for each f. i.e. ~~$w_2 = w_3$~~

Plugging all this in the equation we get

~~$w_1 = 1/3$~~

~~$w_2 = 1/6$~~

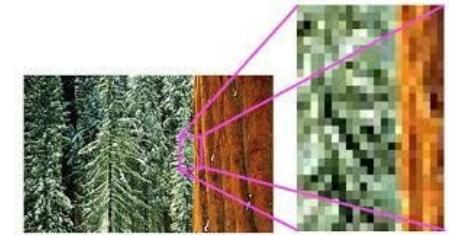
~~$w_3 = 1/6$~~

~~$w_4 = 1/3$~~

Back to ML Interpretability

- SHAP

- Treat each feature i as a player as if we were in a coalitional game
- Estimate the value of feature i by shapley values



$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)]$$

value of feature i

predictor that uses feature $S \cup \{i\}$

payoffs (probability)

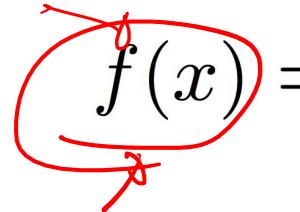
by em270 ,em223 updated 9:38 am et ,mon march 2, 2015
em223 is a designer familiy for fall at its fashion show in em230 on sunday, dedicating its collection to "mamma" with many a pair of "momjeans" insight_and184 and em21, who are behind the em198 brand, sent models down the runway in decidedly feminine dresses and skirts adorned with roses, lace and even embroidered doodles by the designers' own nieces and nephews. many of the looks featured saccharine needlework phrases like "I love you . . ."
X dedicated their fall fashion show to moms.

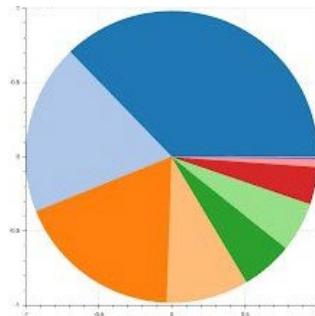
[Lundberg et al, 2017](#)

Additive Feature Attribution

$$f(x) = \phi_0 + \sum_{i=1}^M \phi_i x'_i$$

feature mask





Efficiency of Shaply Values

$$\sum_{j=0}^M \phi_j = v(\mathcal{M})$$

[Lundberg et al, 2017](#)

Challenges with Shapley Value



- The Shapley values of an ML model can be computed exactly as a weighted sum of each feature's *marginal contributions*.
- This involves retraining the model 2^F times, where F is the number of features. These retrained models encompass every possible combination of features, also referred to as the *power set* of all possible feature *coalitions*.
- Training so many models is usually prohibitive, so instead we must find ways to approximate this process. We explore a naive approach for approximating Shapley values that avoids model retraining.

SHAP Values



- **Generating marginalised predictions** - Naive approximation approach is to use this single model for the entire power set, but for each feature that's missing in a given feature coalition, we randomly replace that feature's value with another value from the dataset. In SHAP, "removing" features using methods such as this is referred to as *masking*.
- Replacing the value of a feature with another value sampled at random from the dataset is referred to as sampling from the feature's *marginal distribution*. This can result in unrealistic combinations of feature values.
- More sophisticated methods will sample from the *conditional distribution*, which ensures that only realistic feature combinations are yielded.

Computational Challenges

"SHAP"

- Terms Grow In the Order of 2^F
- Approximating Solutions
 - Shapley Sampling Values ([Štrumbelj et al, 2013](#))
 - Tree SHAP ([Lundberg et al, 2018](#))
 - Deep Approximate Shapley Propagation ([Ancona et al, 2019](#))

Kernel SHAP

Deep SHAP

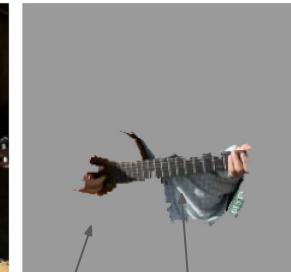
$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)]$$

Computational Challenges

- Estimating Prediction Outcomes With Partial Features
 - Neural networks are not designed to use partial features
 - One solution is to use the expected value ([Lundberg et al, 2018](#))

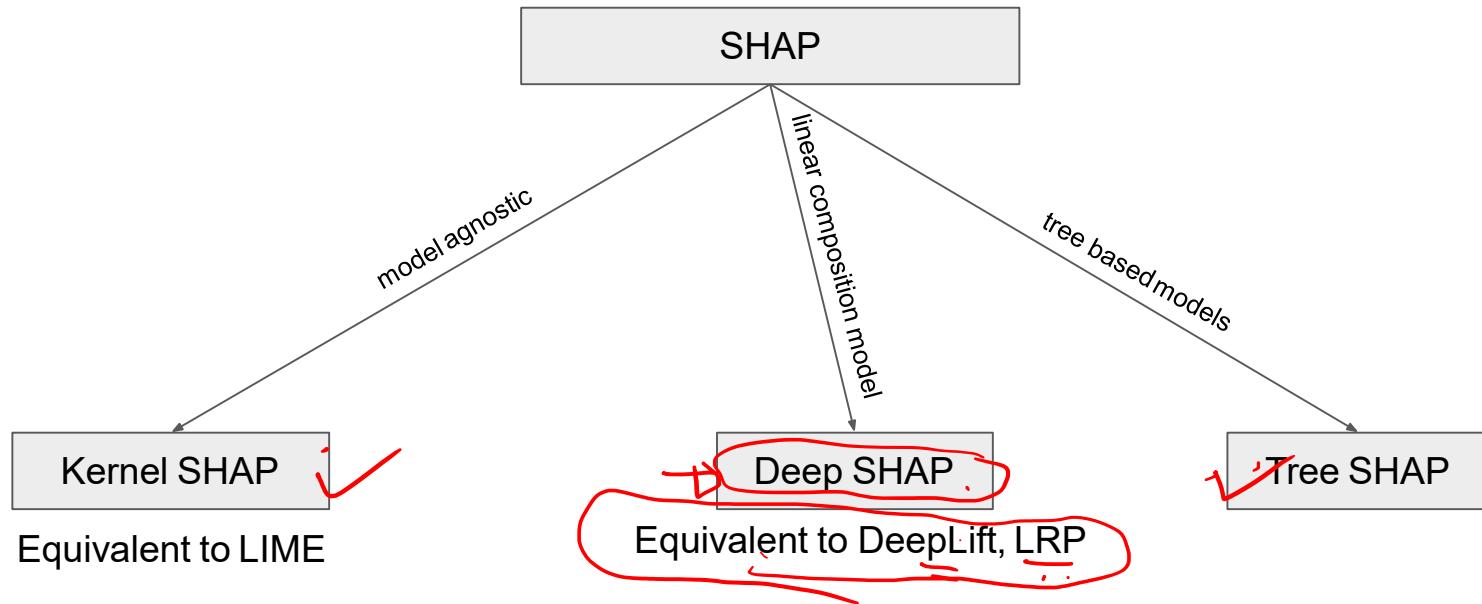
$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)]$$

$$f_S(x_S) = \frac{1}{K} \sum_k f(x_S^k, x_S^*)$$



x_S^k x_S^*

SHAP Based Methods



KernelShap

$$f(x) - \frac{(x_i - \bar{x}_i)}{d_2} \quad (10, -30, -)$$

Assume the trained model $f(x_1, x_2, \dots, x_M)$ has $M=4$ features

we investigate on data instance x for example $x = (10, 20, 30, 40)$

1. Generate a vector which has M elements, and each element is randomly choose as 0 or 1. We use z' represent the vector, so one random example can be $z' = (1, 0, 1, 0)$. Here 1 indicates the presence of the feature, while 0 indicates the feature missing.

Totally we can generate 2^M different z' .

Finally our mapping example result is $h(z') = (10, 22, 30, 44)$, 10 and 30 from instance x ,

22 and 44 are randomly sampled values.

Note: the mapping $h(z')$ is a one-to-many mapping, the max number of output mapping is the size of the given dataset if missing feature sampling is based on marginal distribution.

3. Calculate the model prediction $f(h(z'))$.

4. Compute the weight for each z' with the below SHAP kernel (*the intuition for the kernel will be discussed in next section, let's go ahead*)

5. Construct the below linear regression model $g(z')$, the coefficient ϕ_0 represents model's average prediction on all data instances, and $\phi_1, \phi_2, \dots, \phi_M$ represent

Shapley weights

$$\pi_x(z') = \frac{(M-1)}{\binom{M}{|z'|} |z'| (M - |z'|)}$$

corresponding feature's contribution.

$$g(z') = \phi_0 + \sum_{j=1}^M \phi_j z'_j$$



KernelSHAP introduces the below **custom loss function**, and the custom loss function actually is MSE loss weighted by kernel $\pi(z')$

$$L(f, g, \pi_x) = \sum_{z' \in Z} [f(h_x(z')) - g(z')]^\top \pi_x(z')$$

Shapley weights

Annotations: A red bracket on the left side of the equation groups the first two terms. A red oval encloses the term $\pi_x(z')$. A red arrow points from the word "Shapley" to the oval. Another red arrow points from the word "weights" to the bottom right of the oval.

Kernel SHAP

→ model agnostic

- Remember the LIME Training Objective

$$\xi(x) = \operatorname{argmin}_{g \in G} \mathcal{L}(f, g, \pi_x) + \boxed{\Omega(g)}$$

$$\mathcal{L}(f, g, \pi_{x'}) = \sum_{z' \in Z} [f(h_x(z')) - g(z')]^2 \pi_{x'}(z')$$

- SHAP Equivalent Objective

$$\Omega(g) = 0$$

$$\pi_x(z') = \frac{(M-1)}{\binom{M}{|z'|} |z'| (M-|z'|)}$$

[Lundberg et al, 2017](#)

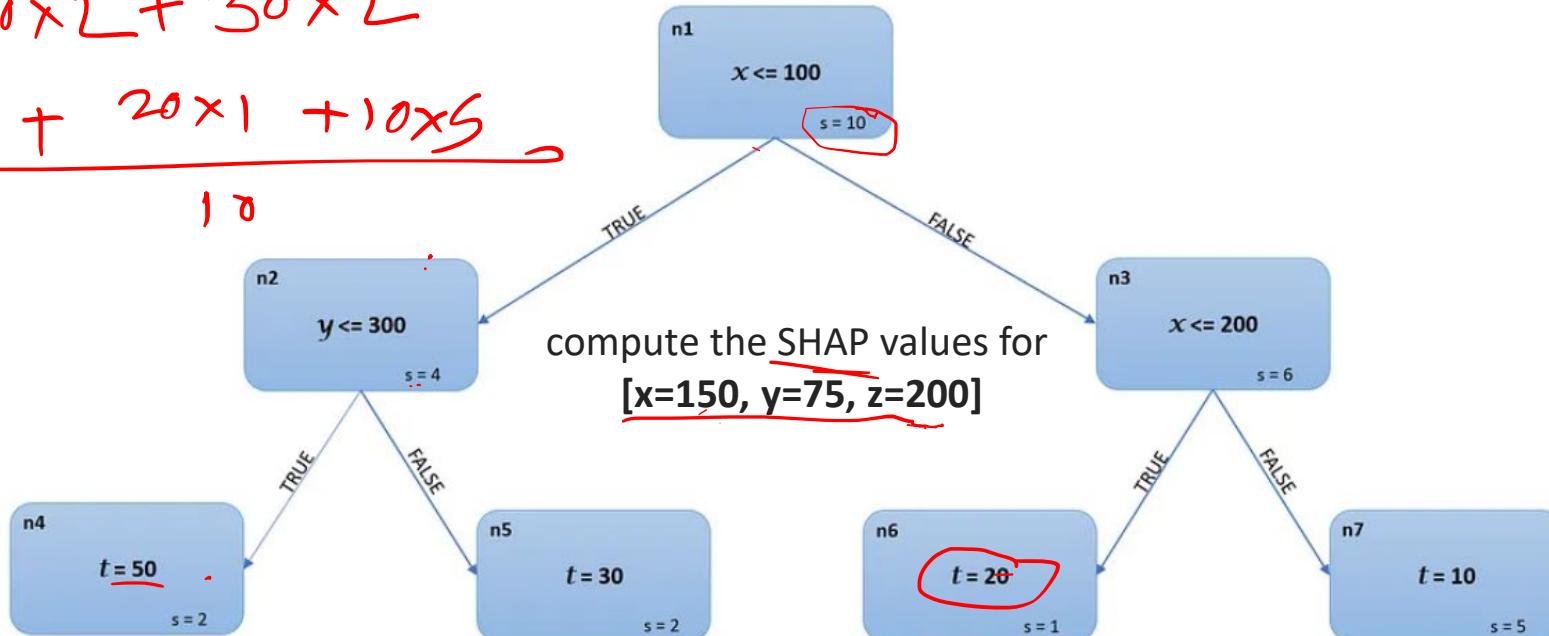
TreeShap

ϕ_6

$$50 \times 2 + 30 \times 2$$

$$\underline{+ 20 \times 1 + 10 \times 5}$$

10



n1, n2, n3, ..., n7 represent the nodes of the tree. s values in each node represents the number of samples from the training set that fall into each node.

150 75 = 200

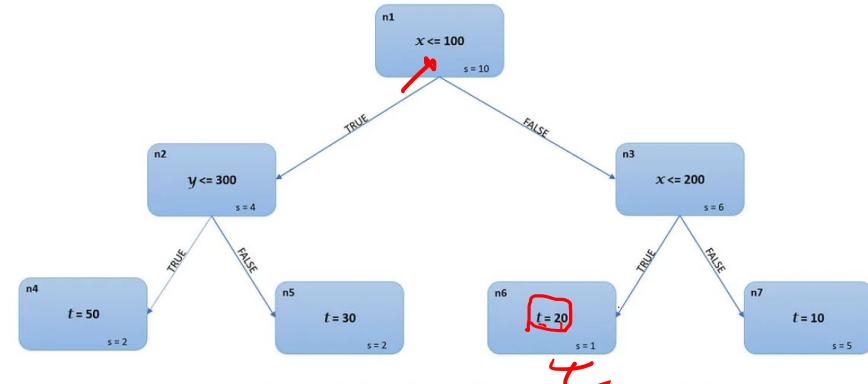
The prediction for the null model ϕ^0 (also called base value) = mean prediction for the training set = $(50*2 + 30*2 + 20*1 + 10*5)/10 = 23$

$$\phi_0(y) \rightarrow 23$$

Consider the sequence: $x > y > z$:

Marginal contribution of x in this sequence, $\phi_x^1 = 20 - 23 = -3$.

Marginal contribution of y and z in this sequence $\phi_y^1 = \phi_z^1 = 20 - 20 = 0$



n1, n2, n3, ..., n7 represent the nodes of the tree. s values in each node represents the number of samples from the training set that fall into each node.

Let's consider the sequence $y > z > x$:

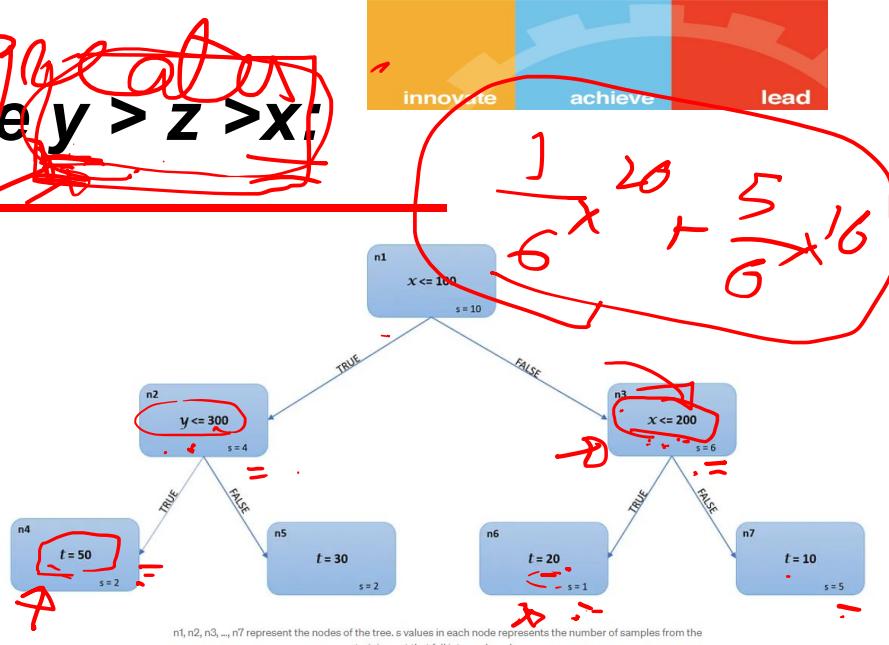
Point of interest: $[x=150, y=75, z=200]$

1) First, the feature y is added to the null model. The first node n_1 uses x as the split variable, since x is not available yet, we compute the prediction as $(4/10) * (\text{prediction from left child node } n_2) + (6/10) * (\text{prediction from right child } n_3)$

i) Prediction from node n_2 : n_2 uses y as the split variable, since y is available ($y_i = 75$ for instance i), the prediction from node $n_2 = 50$.

ii) Prediction from node n_3 : Again, n_3 uses x as the split variable. Therefore, by similar logic, prediction from $n_3 = (1/6)*20 + (5/6)*10 = 70/6$.

iii) Therefore, the prediction for the model with just the feature y is $(4/10)*150 + (6/10)*(70/6) = 27$.



iv) Hence, the marginal contribution for y in this sequence, $\phi^{y^2} = 27 - 23 = 4$.

2) Marginal contribution for z in this sequence, $\phi^{z^2} = 0$.

3) Finally, we add the feature x to the model which gives the prediction as 20. Therefore the marginal contribution of x in this sequence is $\phi^{x^2} = 20 - 27 = -7$.

Sequence $x > z > y$: $\phi^{x^3} = -3, \phi^{y^3} = 0, \phi^{z^3} = 0$

Sequence $z > x > y$: $\phi^{x^4} = -3, \phi^{y^4} = 0, \phi^{z^4} = 0$

Sequence $z > y > x$: $\phi^{x^5} = -7, \phi^{y^5} = 4, \phi^{z^5} = 0$

Sequence $y > x > z$: $\phi^{x^6} = -7, \phi^{y^6} = 4, \phi^{z^6} = 0$

Hence, SHAP values for the instance i are given by:

$$\phi^x = (\phi^{x^1} + \phi^{x^2} + \phi^{x^3} + \phi^{x^4} + \phi^{x^5} + \phi^{x^6})/6 = (-3-7-3-3-7-7)/6 = -5$$

$$\phi^y = (\phi^{y^1} + \phi^{y^2} + \phi^{y^3} + \phi^{y^4} + \phi^{y^5} + \phi^{y^6})/6 = (0+4+0+0+4+4)/6 = 2$$

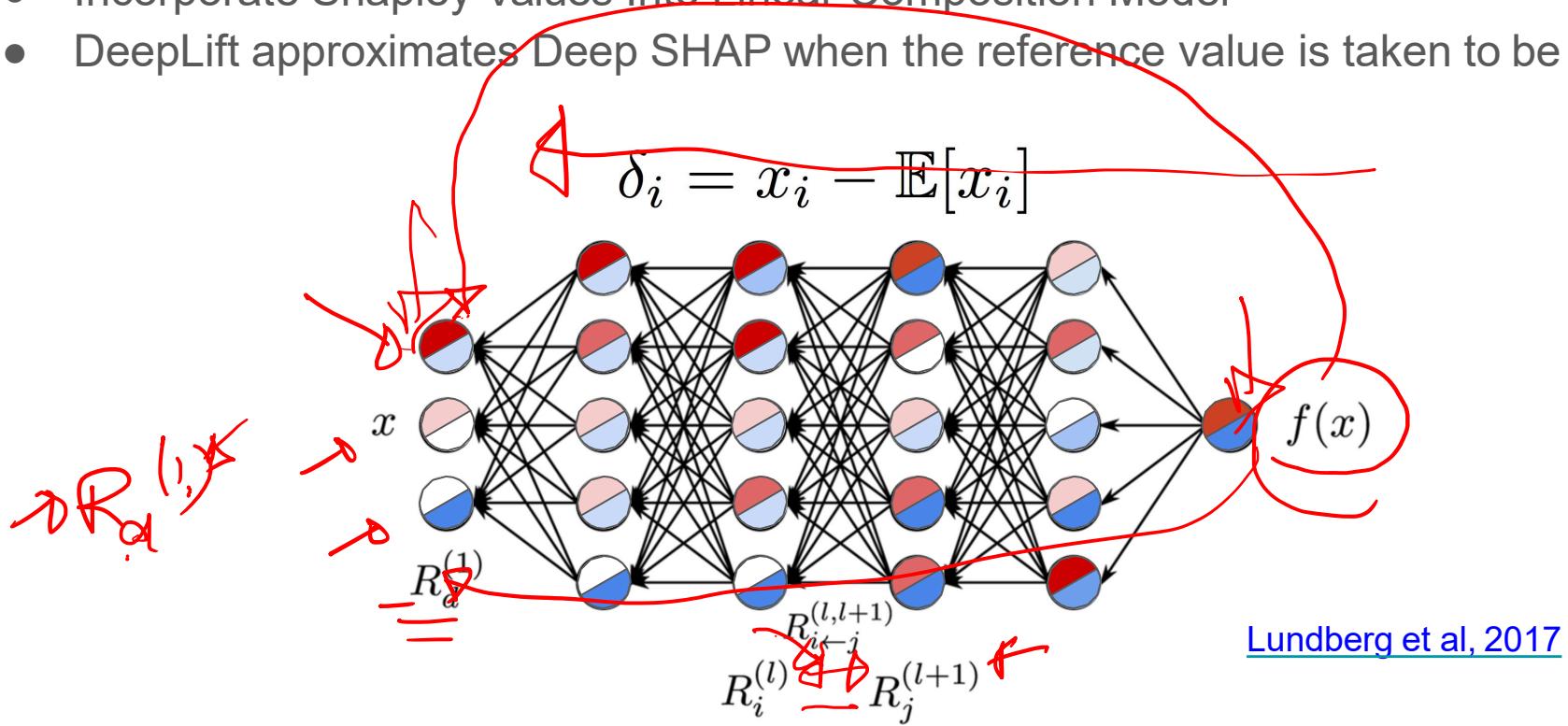
$$\phi^z = (\phi^{z^1} + \phi^{z^2} + \phi^{z^3} + \phi^{z^4} + \phi^{z^5} + \phi^{z^6})/6 = (0+0+0+0+0+0)/6 = 0$$

Explanation for the prediction for instance i (20) = $\phi^0 + \phi^x + \phi^y + \phi^z = 23 + (-5) + 2 + 0 = 20$:

"The base value of the prediction in the absence of any information on independent variables is 23; knowing $x=150$ decreased the prediction by 5 and knowing $y = 75$ increased the prediction by 2 giving a final prediction of 20. Knowing $z = 300$ had no impact on the model prediction."

Deep SHAP

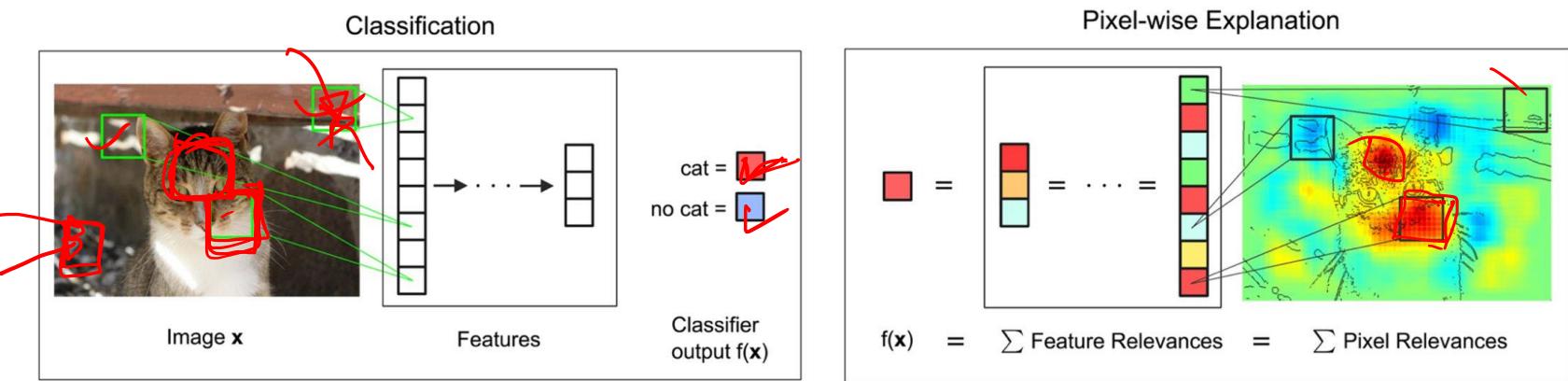
- Incorporate Shapley Values Into Linear Composition Model
- DeepLift approximates Deep SHAP when the reference value is taken to be $E[x]$



Required Reading

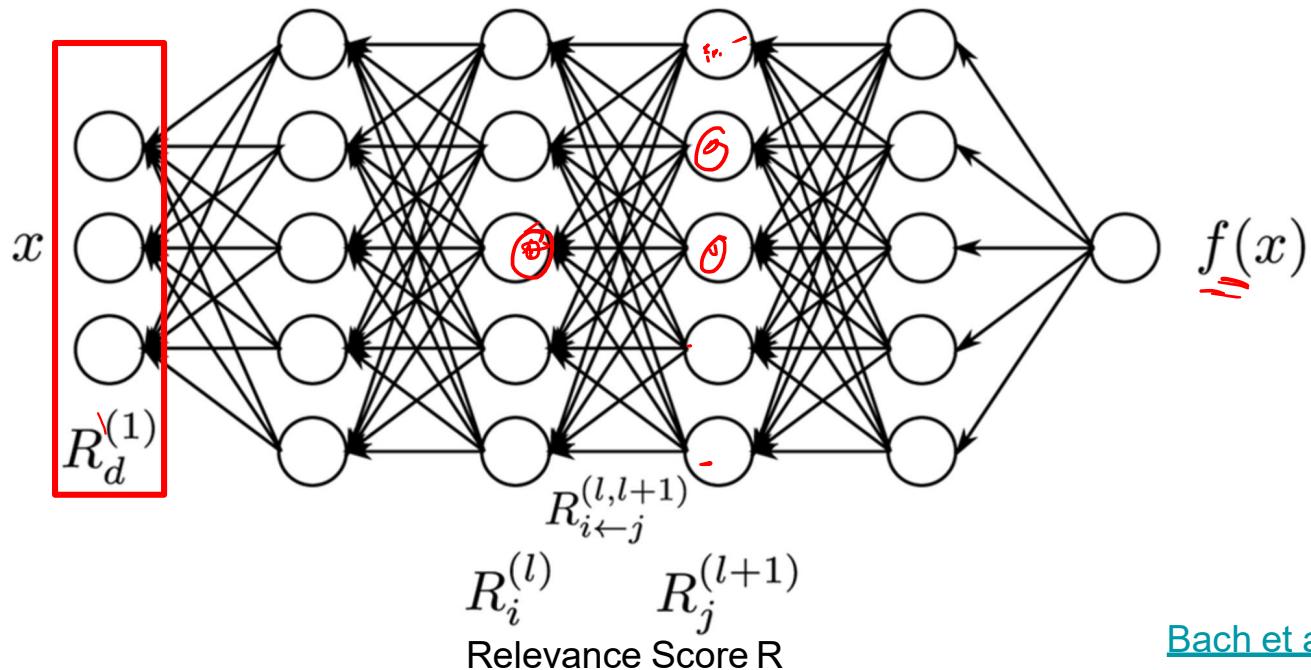
- Molnar: Ch 5.9, Ch 5.10

Layerwise Relevance Propagation (LRP)



[Bach et al. 2015](#)

Layerwise Relevance Propagation (LRP)



[Bach et al. 2015](#)

Relevance Scores

- x_i - output of neuron i
- g - activation function
- w_{ij} - weight of neural network connecting neuron x_i and x_j
- z_{ij} - linearly transformed neuron outputs

$$x_j^{(l)} = g(z_j)$$

$$z_j = \sum_i z_{ij} + b_j$$

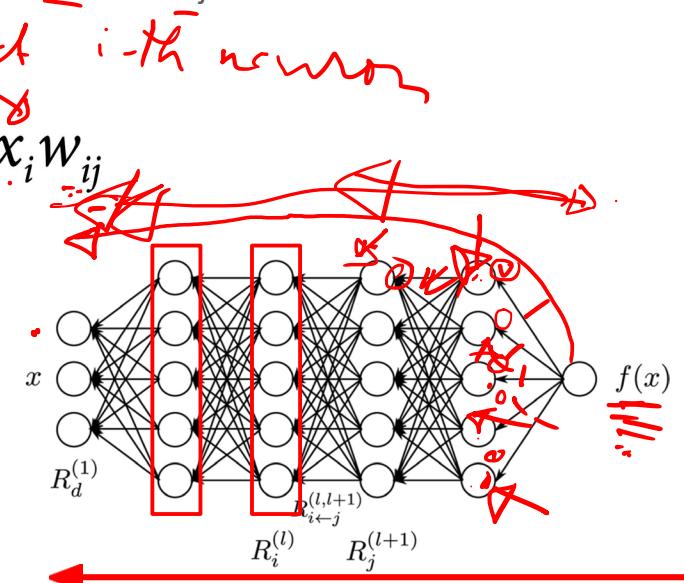
$$z_{ij} = x_i w_{ij}$$

- Relevant Score $R_i^{(l)}$ of neuron i at level l

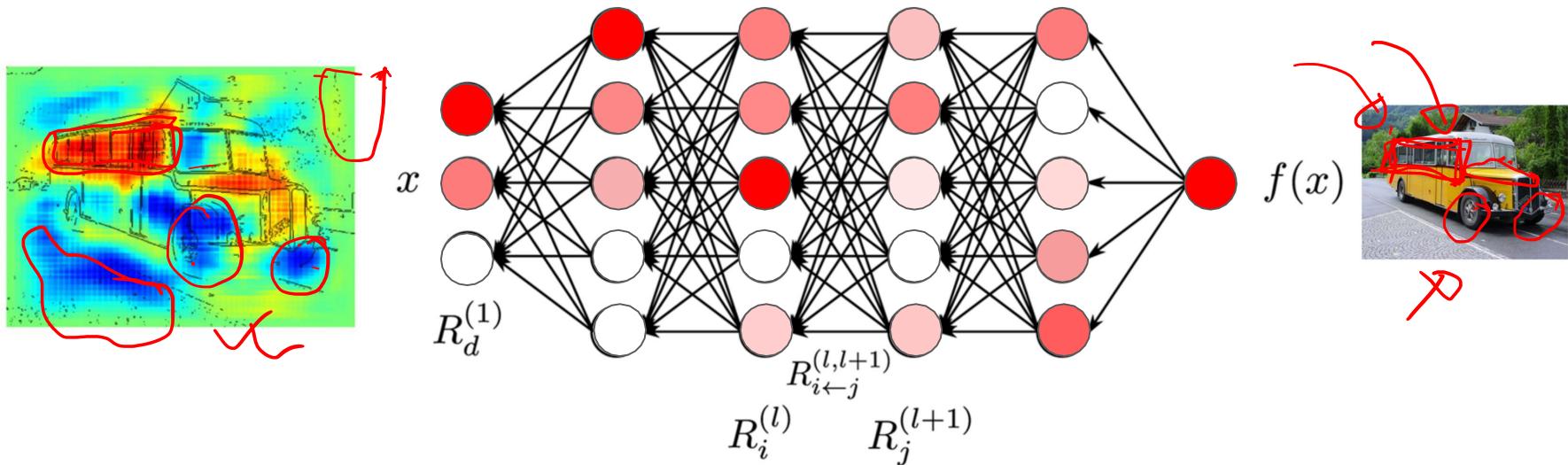
$$R_{i \leftarrow j}^{(l, l+1)} = \frac{z_{ij}}{z_j} R_j^{(l+1)}$$

$$R_i^{(l)} = \sum_j R_{i \leftarrow j}^{(l, l+1)}$$

next layer neurons



Relevance Score Propagation



$$z_j = \sum_i z_{ij} + b_j$$

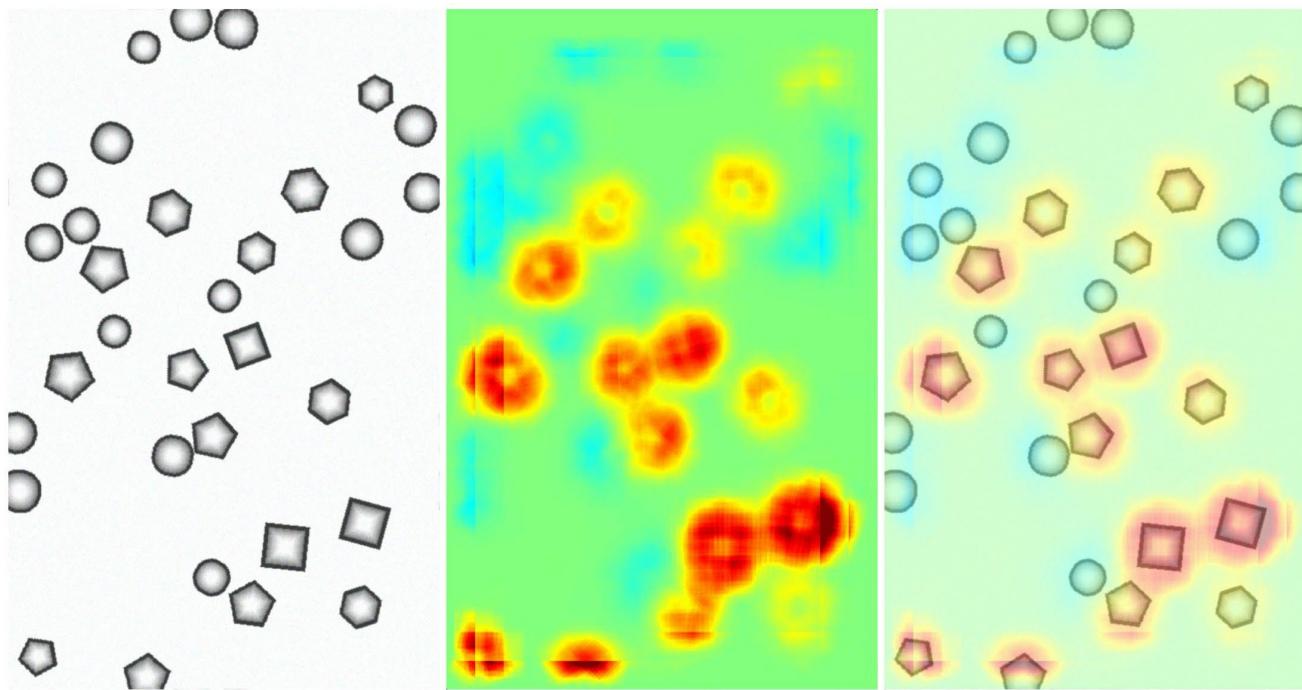
$$z_{ij} = x_i w_{ij}$$

$$R_{i \leftarrow j}^{(l,l+1)} = \frac{z_{ij}}{z_j} \cdot R_j^{(l+1)}$$

$$R_i^{(l)} = \sum_j R_{i \leftarrow j}^{(l,l+1)}$$

[Bach et al. 2015](#)

Results on Synthetic Data



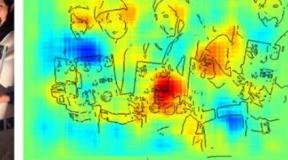
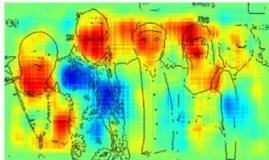
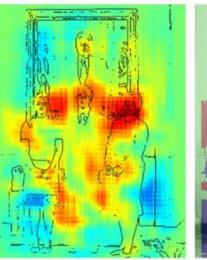
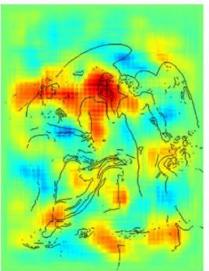
[Bach et al. 2015](#)

Results on Pascal Dataset



[Bach et al. 2015](#)

More Examples

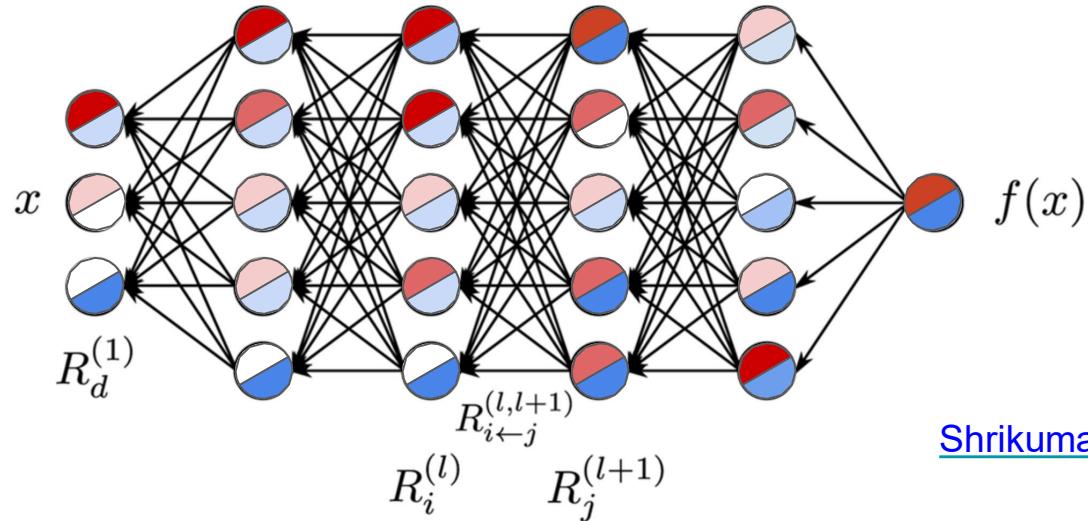


[Bach et al. 2015](#)

DeepLift

- DeepLift Allows Each Neuron A Reference Value for Activation Output x_i^0

$$\delta_i = x_i - x_i^0$$



[Shrikumar et al, 2016](#)

DeepLiFT

- DeepLiFT(Deep Learning Important FeaTures) uses a reference image along with an input image to explain the input pixels (similar to LRP).
- While LRP followed the conservation axiom, there was no clear way on how to distribute the net relevance among the pixels.
- DeepLiFT fixes this problem by enforcing an additional axiom on how to propagate the relevance down.

DeepLiFT Axioms

Axiom 1. Conservation of Total Relevance:

Sum of relevance of all inputs must equal the difference between the score of the input image and baseline image, at every neuron. This axiom is same as the one in LRP.

Given a reference input vector \mathbf{x}_0 with score y_0 and an input vector \mathbf{x} with score y , we define:

$$\Delta\mathbf{x} = \mathbf{x} - \mathbf{x}_0$$

$$\Delta y = y - y_0$$

We would like the relevance or contribution $C_{\Delta x_i \Delta y}$ to follow:

$$\sum_{i=0}^n C_{\Delta x_i \Delta y} = \Delta y$$

Axiom 2. Back Propagation/Chain Rule:

- The relevance per input follows the chain rule like gradients. This is enough to help us back propagate the gradient-like relevance per input. This axiom makes DeepLiFT closer to “vanilla” gradient back propagation.

We would like the relevance per input $m_{\Delta x_i \Delta y}$ defined as:

$$m_{\Delta x_i \Delta y} := \frac{C_{\Delta x_i \Delta y}}{\Delta x_i}$$

to follow the chain rule like gradients to help us perform back propagation:

$$m_{\Delta x \Delta z} = \sum_{i=0}^n m_{\Delta x \Delta y_i} m_{\Delta y_i \Delta z}$$

- Split relevance into +ve and –ve parts

$$\Delta x = \Delta x^+ + \Delta x^-$$

$$\Delta y = \Delta y^+ + \Delta y^-$$

$$C_{\Delta x \Delta y} = C_{\Delta x^+ \Delta y} + C_{\Delta x^- \Delta y}$$

- Depending on the function at hand, the authors suggest the following candidate solutions for $\mathbf{C}()$ and $\mathbf{m}()$:
- **Linear Rule** for linear functions: This is exactly same as using the gradients for $m()$. LRP would do the same as well.

$$y = b + \sum_{i=1}^n w_i x_i$$

$$\Delta y = \sum_{i=1}^n w_i \Delta x_i$$

$$C_{\Delta x_i \Delta y} := w_i \Delta x_i$$

- **Rescale Rule** for non-linear functions like ReLU, Sigmoid, same as LRP.

$$y = f(x)$$

$$C_{\Delta x^+ \Delta y^+} := \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} \Delta x^+$$

$$C_{\Delta x^- \Delta y^-} := \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} \Delta x^-$$

- **RevealCancel (Shapley) Rule** for non-linear functions like MaxPool: Using Rescale rule (with reference input of 0s) for MaxPool would end up attributing all the relevance contribution to the biggest input. Changes along other inputs would make no difference to the output. RevealCancel rule fixes this using Shapley values.

$$y = f(x)$$

$$C_{\Delta x^+ \Delta y^+} := \frac{1}{2} \frac{f(x_0 + \Delta x^+) - f(x_0)}{\Delta x^+} \Delta x^+ + \frac{1}{2} \frac{f(x_0 + \Delta x^- + \Delta x^+) - f(x_0 + \Delta x^-)}{\Delta x^+} \Delta x^+$$

$$C_{\Delta x^- \Delta y^-} := \frac{1}{2} \frac{f(x_0 + \Delta x^-) - f(x_0)}{\Delta x^-} \Delta x^- + \frac{1}{2} \frac{f(x_0 + \Delta x^+ + \Delta x^-) - f(x_0 + \Delta x^+)}{\Delta x^-} \Delta x^-$$

Gradient Based Interpretation of Deep Networks

<https://towardsdatascience.com/recent-advancements-in-explainable-neural-networks-2cd06b5d2016>

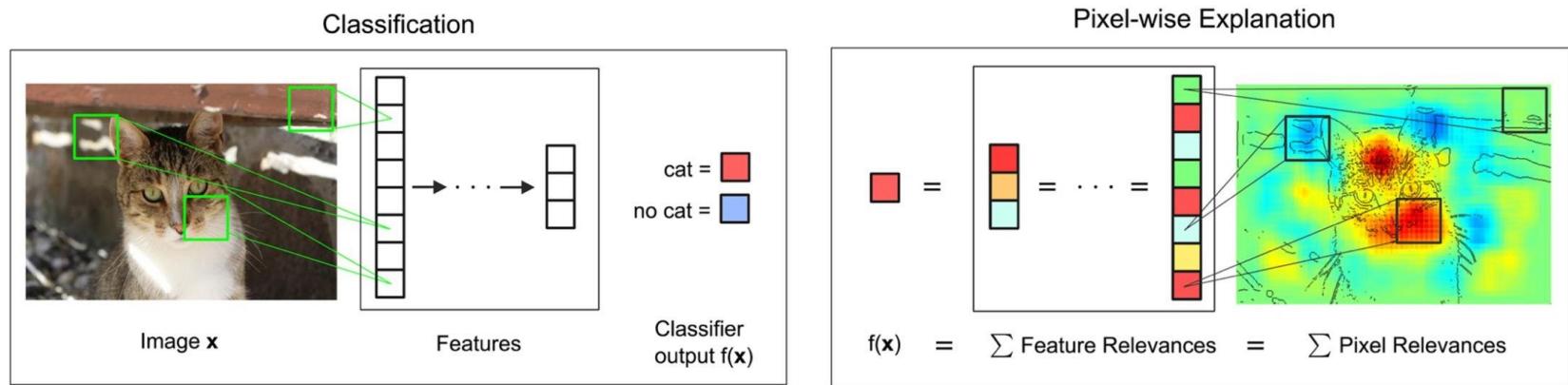
<https://towardsdatascience.com/visual-interpretability-for-convolutional-neural-networks-2453856210ce>

<https://distill.pub/2018/building-blocks/>

<https://christophm.github.io/interpretable-ml-book/neural-networks.html> - Chapter 10

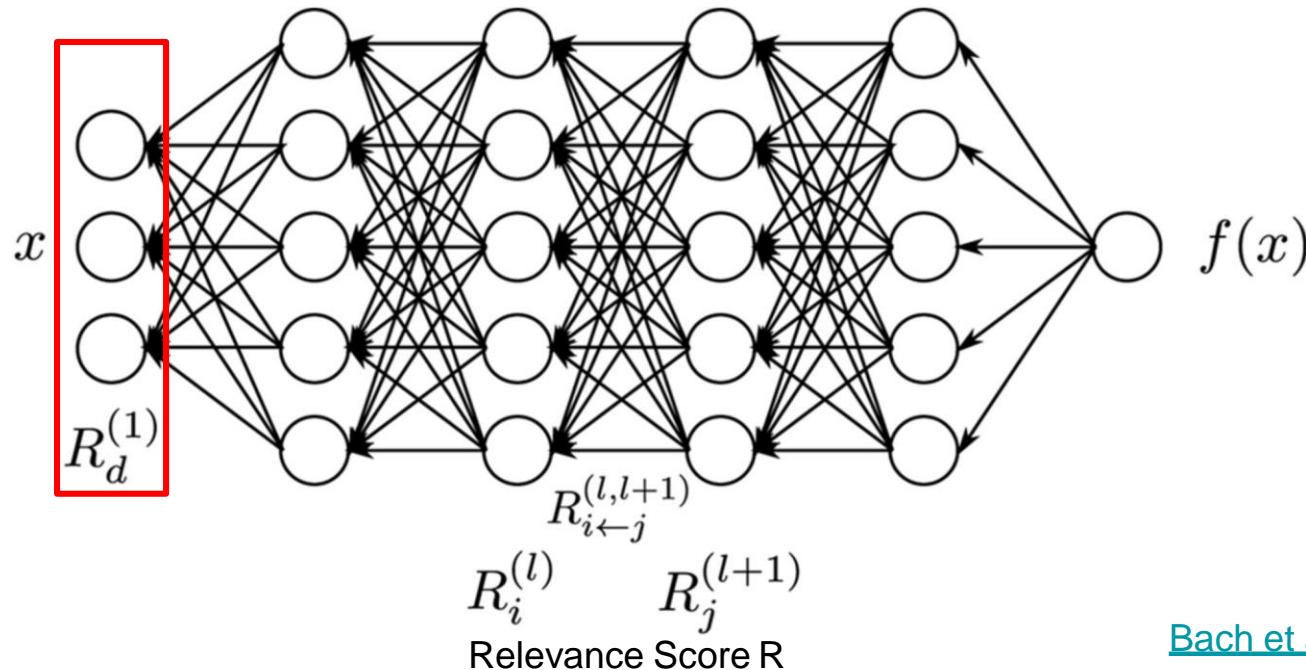
Refer to Labsheet 5

Layerwise Relevance Propagation (LRP)



[Bach et al. 2015](#)

Layerwise Relevance Propagation (LRP)



[Bach et al. 2015](#)

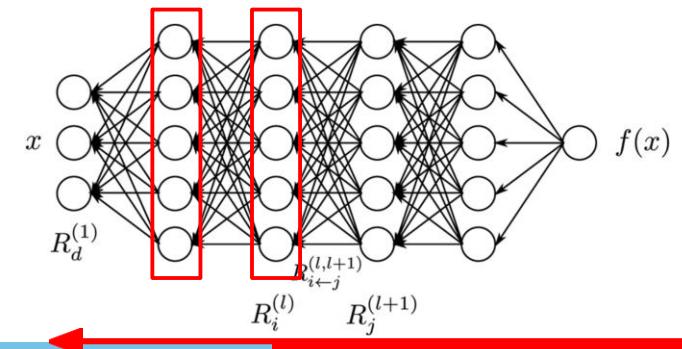
Relevance Scores

- x_i - output of neuron i $x_j = g(z_j)$
- g - activation function
- w_{ij} - weight of neural network connecting neuron x_i and x_j
- z_{ij} - linearly transformed neuron outputs

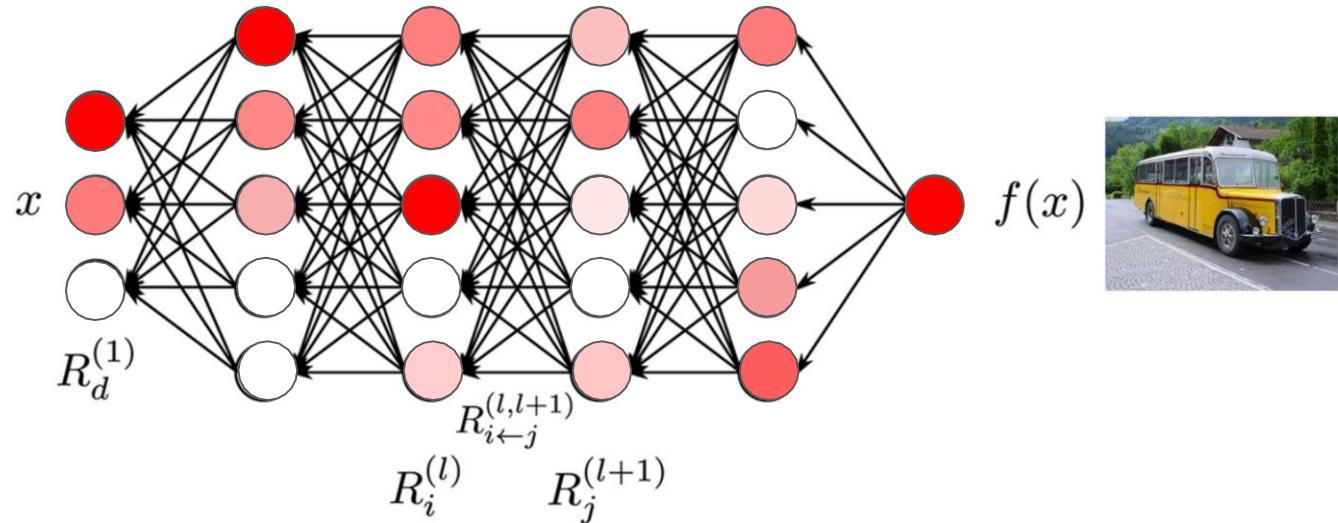
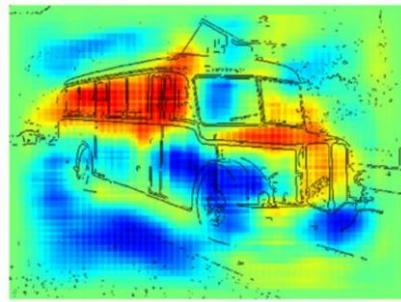
$$z_j = \sum_i z_{ij} + b_j \quad z_{ij} = x_i w_{ij}$$

- Relevant Score $R_i^{(l)}$ of neuron i at level l

$$R_{i \leftarrow j}^{(l,l+1)} = \frac{z_{ij}}{z_j} \cdot R_j^{(l+1)} \quad R_i^{(l)} = \sum_j R_{i \leftarrow j}^{(l,l+1)}$$

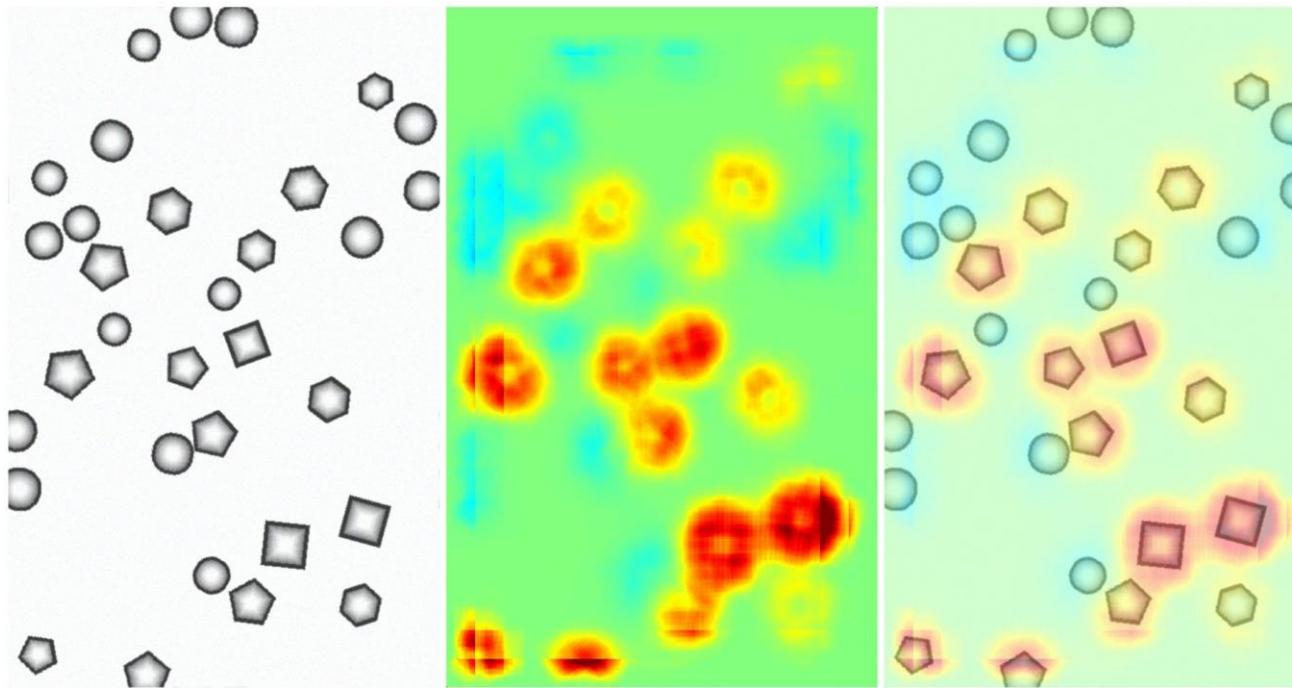


Relevance Score Propagation



[Bach et al. 2015](#)

Results on Synthetic Data



Bach et al, 2015

Results on Pascal Dataset



Bach et al. 2015

More Examples

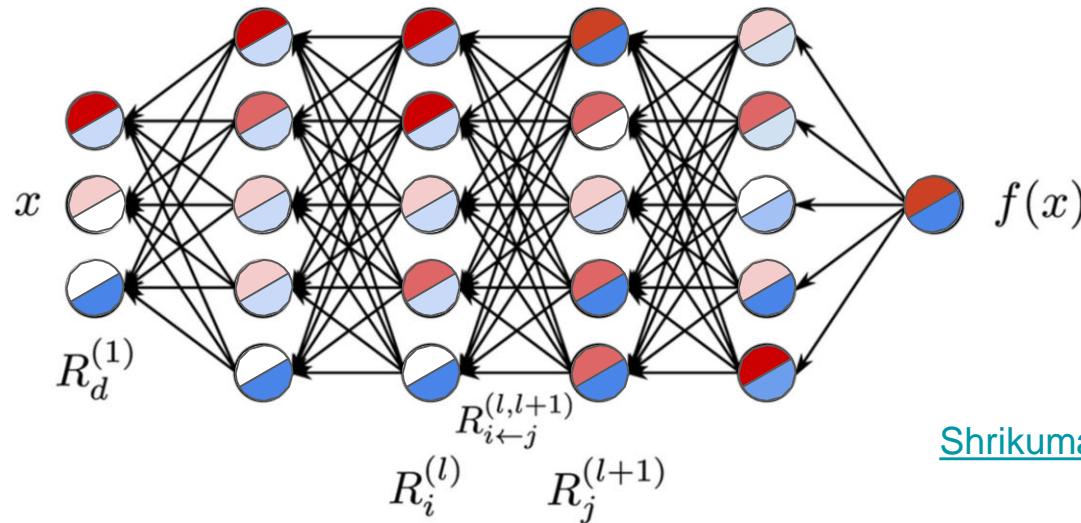


[Bach et al. 2015](#)

DeepLift

- DeepLift Allows Each Neuron A Reference Value for Activation Output x_i^0

$$\delta_i = x_i - x_i^0$$



[Shrikumar et al, 2016](#)

DeepLiFT

- DeepLiFT(Deep Learning Important FeaTures) uses a reference image along with an input image to explain the input pixels (similar to LRP).
- While LRP followed the conservation axiom, there was no clear way on how to distribute the net relevance among the pixels.
- DeepLiFT fixes this problem by enforcing an additional axiom on how to propagate the relevance down.

DeepLiFT Axioms

Axiom 1. Conservation of Total Relevance:

Sum of relevance of all inputs must equal the difference between the score of the input image and baseline image, at every neuron. This axiom is same as the one in LRP.

Given a reference input vector \mathbf{x}_0 with score y_0 and an input vector \mathbf{x} with score y , we define:

$$\Delta\mathbf{x} = \mathbf{x} - \mathbf{x}_0$$

$$\Delta y = y - y_0$$

We would like the relevance or contribution $C_{\Delta x_i \Delta y}$ to follow:

$$\sum_{i=0}^n C_{\Delta x_i \Delta y} = \Delta y$$

Axiom 2. Back Propagation/Chain Rule:

- The relevance per input follows the chain rule like gradients. This is enough to help us back propagate the gradient-like relevance per input. This axiom makes DeepLiFT closer to “vanilla” gradient back propagation.

We would like the relevance per input $m_{\Delta x_i \Delta y}$ defined as:

$$m_{\Delta x_i \Delta y} := \frac{C_{\Delta x_i \Delta y}}{\Delta x_i}$$

to follow the chain rule like gradients to help us perform back propagation:

$$m_{\Delta x \Delta z} = \sum_{i=0}^n m_{\Delta x \Delta y_i} m_{\Delta y_i \Delta z}$$

- Split relevance into +ve and -ve parts

$$\Delta x = \Delta x^+ + \Delta x^-$$

$$\Delta y = \Delta y^+ + \Delta y^-$$

$$C_{\Delta x \Delta y} = C_{\Delta x^+ \Delta y} + C_{\Delta x^- \Delta y}$$

- Depending on the function at hand, the authors suggest the following candidate solutions for $\mathbf{C}()$ and $\mathbf{m}()$:
- **Linear Rule** for linear functions: This is exactly same as using the gradients for $m()$. LRP would do the same as well.

$$y = b + \sum_{i=1}^n w_i x_i$$

$$\Delta y = \sum_{i=1}^n w_i \Delta x_i$$

$$C_{\Delta x_i \Delta y} := w_i \Delta x_i$$

- **Rescale Rule** for non-linear functions like ReLU, Sigmoid, same as LRP.

$$y = f(x)$$

$$C_{\Delta x^+ \Delta y^+} := \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} \Delta x^+$$

$$C_{\Delta x^- \Delta y^-} := \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} \Delta x^-$$

- **RevealCancel (Shapley) Rule** for non-linear functions like MaxPool: Using Rescale rule (with reference input of 0s) for MaxPool would end up attributing all the relevance contribution to the biggest input. Changes along other inputs would make no difference to the output. RevealCancel rule fixes this using Shapley values.

$$y = f(x)$$

$$C_{\Delta x^+ \Delta y^+} := \frac{1}{2} \frac{f(x_0 + \Delta x^+) - f(x_0)}{\Delta x^+} \Delta x^+ + \frac{1}{2} \frac{f(x_0 + \Delta x^- + \Delta x^+) - f(x_0 + \Delta x^-)}{\Delta x^+} \Delta x^+$$

$$C_{\Delta x^- \Delta y^-} := \frac{1}{2} \frac{f(x_0 + \Delta x^-) - f(x_0)}{\Delta x^-} \Delta x^- + \frac{1}{2} \frac{f(x_0 + \Delta x^+ + \Delta x^-) - f(x_0 + \Delta x^+)}{\Delta x^-} \Delta x^-$$

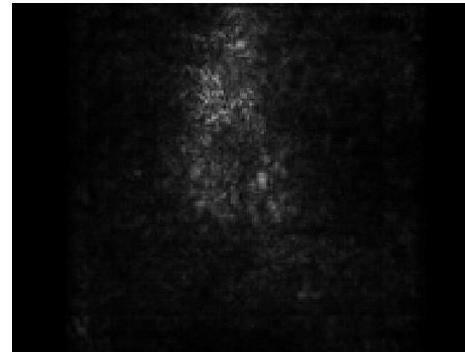
Saliency Maps

Given an image, which pixels within the image are the most important in helping the network make the decision?

Visualizing the gradients of the class activation score (pre-softmax) with respect to the image pixel:

For the class score S_c and image I_0 , the saliency map w can be defined as:

$$w = \frac{\partial S_c}{\partial I} \Big|_{I_0}$$

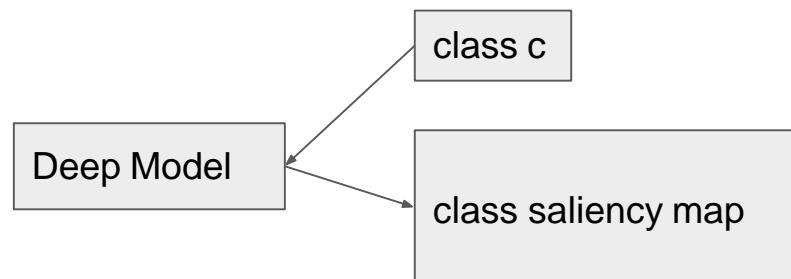


Class Specific Saliency Maps

- How does a typical “dog” or “cat” or any such target image class look like according to the neural network?
- Generate a representative image for class c
 - Fix trained networks
 - Generate an input image that maximizes model probabilities

$$\arg \max_I S_c(I) - \lambda \|I\|_2^2$$

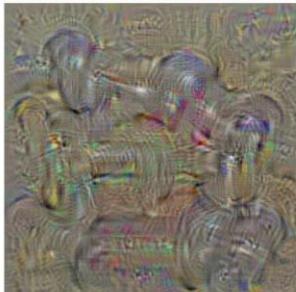
Logits of a trained model on class c



[Simonyan et al, 2014](#)

Class Specific Saliency Maps

- ConvNet Trained on ILSVRC2013



dumbbell



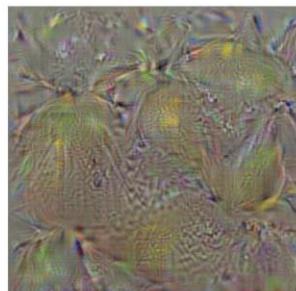
cup



dalmatian



bell pepper



lemon



husky

[Simonyan et al, 2014](#)

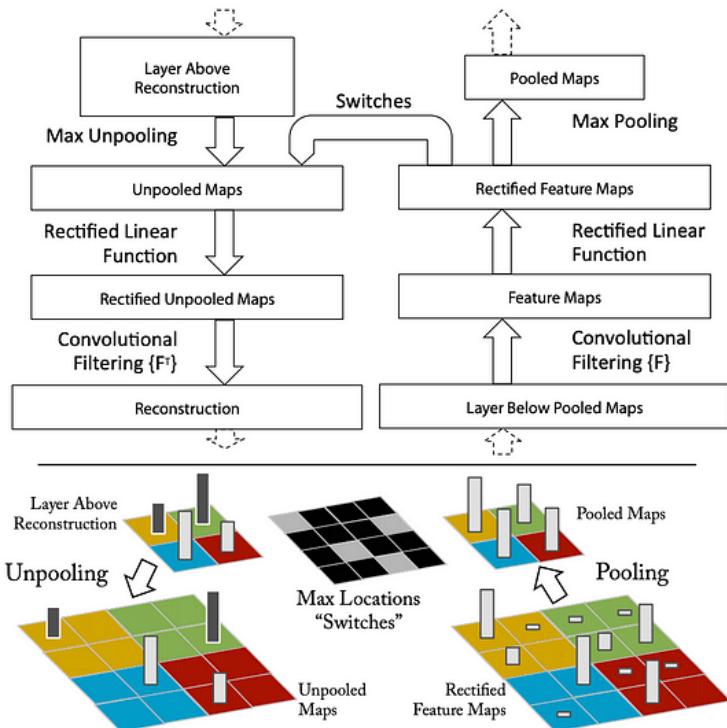
SmoothGrad

- Gradient visualizations were often noisy.
- Add a small Gaussian noise to the input image and sample multiple images from this distribution. After calculating gradients, average the gradients out.
- Image recognition tasks are invariant under color and illumination changes. In such cases, visualizing absolute value of gradients is often enough.
- Gradient maps often have a few pixels with very high gradient values. Capping them off improves visualizations.
- Sometimes, multiplying the gradients with the image provides sharper visualizations.
- But this can sometimes have undesirable effects of making darker pixels look less important than what the original gradients would show.

Deconvolution Network

- A method to approximately project the activations of an intermediate hidden layer back to the input layer.
- By projecting successive layers (back to the input layer), it can be seen that CNN learns successively more complicated patterns in the image like edges, simple shapes, more complicated shapes, textures etc.

Deconvolution Network



Deconvolution Network

- **Initialize:** Start with the layer that we want to project down and initialize the reconstructed signal equal to the activations of that layer. Back propagate the reconstructed signal down.
- **MaxPool:** When u encounter a MaxPooling layer, look for indices from where the inputs were pooled and passed up in the forward pass. In the backward pass, pass the reconstructed signal values to these indices, zeroing out the other positions.
- **ReLU:** When u encounter the ReLU layer, pass the reconstructed signal only if it is positive, else zero it out.
- **Weights:** When u encounter the CNN layer or any weight multiplication, multiply the transpose of the weights to the reconstructed signal and pass it down.

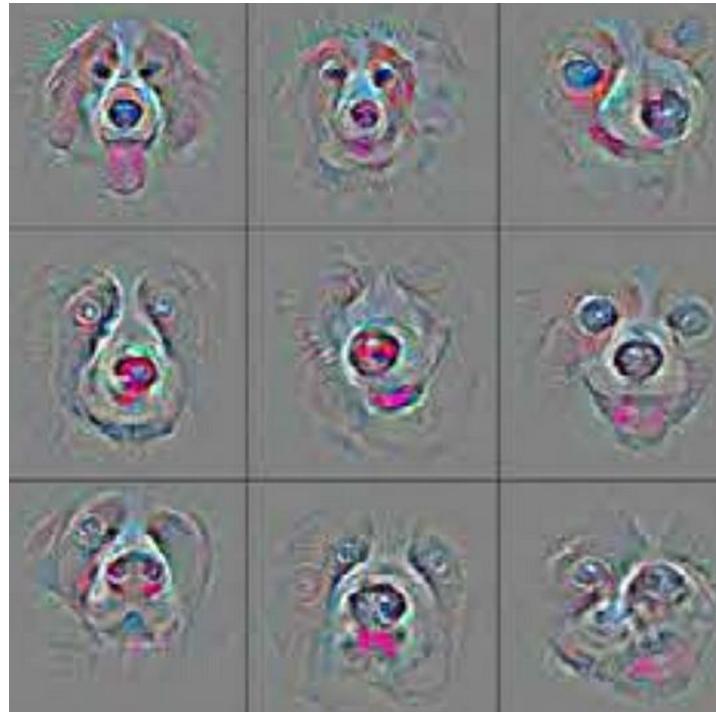
Deconvolution Network

- **MaxPool:** When you encounter a MaxPooling layer, look for indices from where the inputs were pooled and passed up in the forward pass. In the backward pass, pass the reconstructed signal values to these indices, zeroing out the other positions.
- **ReLU:** When you encounter the ReLU layer, pass the reconstructed signal only if it is positive, else zero it out.
- **Weights:** When you encounter the CNN layer or any weight multiplication, multiply the transpose of the weights to the reconstructed signal and pass it down.

Initialize $g_{x_N} = x_N$ for the desired layer to project. Starting from the desired layer, propagate g down the layers till the input image as following:

Forward Pass	Backward Pass
MaxPooling: $x_{n+1} = \max(z_{n1}, z_{n2}, z_{n3})$	UnPooling: $g_{z_n} = g_{x_{n+1}}$ when $i = \arg \max z_{ni}$ $g_{z_n} = 0$ otherwise
ReLU: $z_n = \max(y_n, 0)$	Rectification: $g_{y_n} = g_{z_n}$ when $g_{z_n} > 0$ $g_{y_n} = 0$ otherwise
Filtering: $y_n = w x_n$	Filter Transpose: $g_{x_n} = w^T g_{y_n}$

Deconvolution Network



Guided Backpropagation

- Use gradient back propagation as it is except at the ReLU stages.
- At ReLU stages, we back propagate the gradient only if the gradient is positive.

We perform gradient back propagation except at ReLU:

For the forward pass:

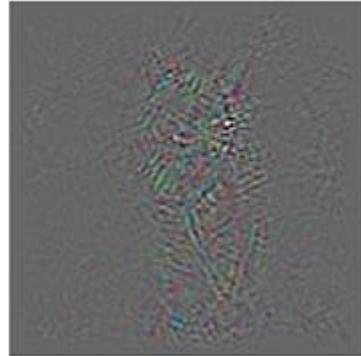
$$z_n = \max(y_n, 0)$$

We perform the backward pass:

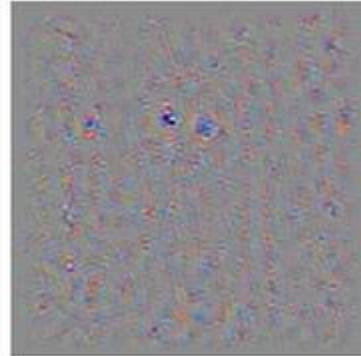
$$g_{y_n} = \begin{cases} g_{z_n} & \textcolor{red}{g_{z_n} > 0 \text{ and } y_n > 0} \\ 0 & \text{otherwise} \end{cases}$$



backpropagation



'deconvnet'



guided backpropagation



Integrated Gradients

Axiom 1. Sensitivity: Whenever the input and baseline differ in exactly one feature, the differing feature should be given non-zero attribution.

- LRP and DeepLiFT follow sensitivity due to the **Conservation of Total Relevance**.
- Gradient based methods do not guarantee the Sensitivity Axiom. This happens because of saturation at ReLU or MaxPool stages when the score function is locally “flat” with respect to some input features.

Axiom 2. Implementation Invariance: Whenever two models are functionally equivalent, they must have identical attributions to input features

- Coarse approximation to gradients like LRP and DeepLiFT might break this assumption.

Integrated Gradients

- Feature Importance determined by the integral of gradients
 - x - input
 - x' - reference input
 - F - black-box model
- Satisfies Implementation Invariance

$$\begin{aligned}\text{IntegratedGrads}_i(x) &:= (x_i - x'_i) \times \int_{\alpha=0}^1 \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} d\alpha \\ &= (x_i - x'_i) \times \sum_{k=1}^m \frac{\partial F(x' + \frac{k}{m} \times (x - x'))}{\partial x_i} \times \frac{1}{m}\end{aligned}$$

Riemann Approximation

[Sundararajan et al 2017](#)

Examples of Integrated Gradients

Original image



Top label and score

Top label: reflex camera
Score: 0.993755



Top label: fireboat
Score: 0.999961

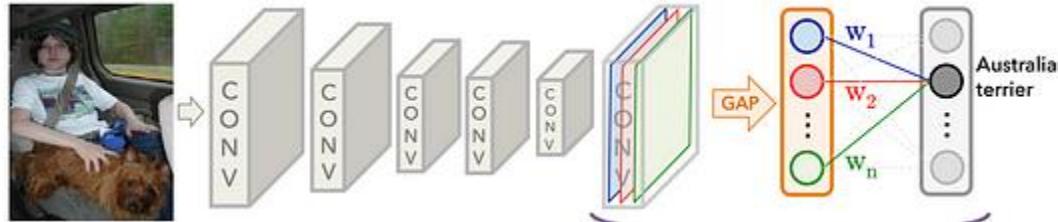


Top label: school bus
Score: 0.997033

Integrated gradients



Class Activation Map (CAM)



For a given image, let the last CNN layer's feature map be given by $f_k(x, y)$, where k is the depth of the feature map. Here are the operations that follow the CNN.

Global Average Pooling:

$$F_k = \sum_{x,y} f_k(x, y)$$

Linear layer:

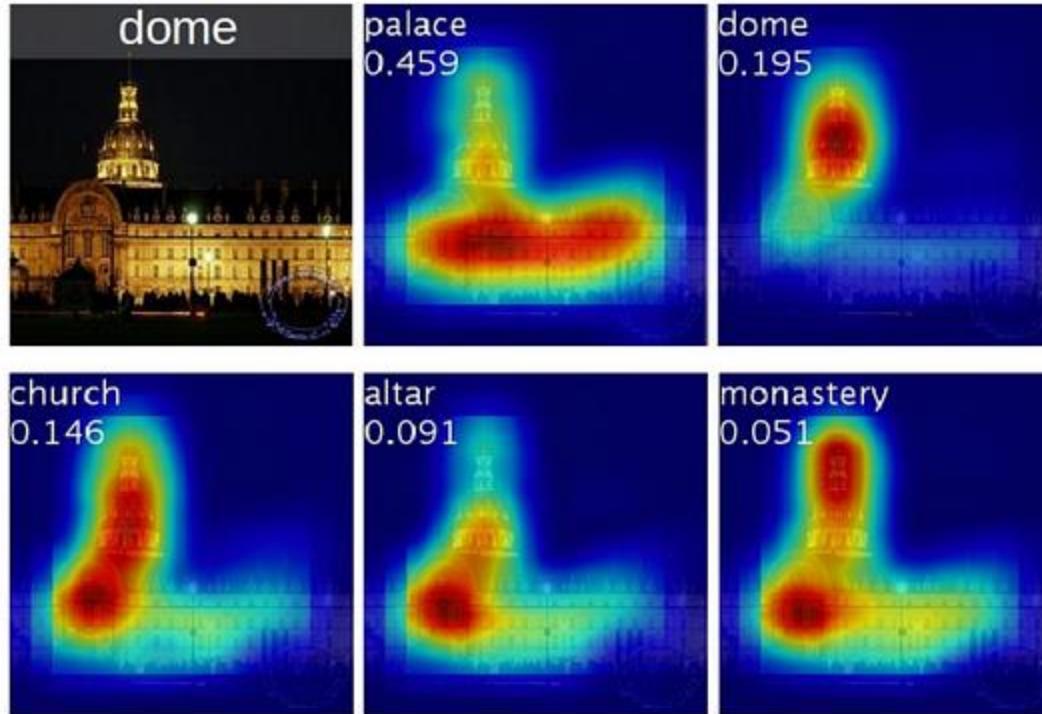
$$S_c = \sum_k w_k^c F_k$$

Softmax layer:

$$P_c = \frac{e^{S_c}}{\sum_i e^{S_i}}$$

$$M_c(x, y) = \sum_k w_k^c f_k(x, y)$$

Class Activation Map (CAM)



GradCAM

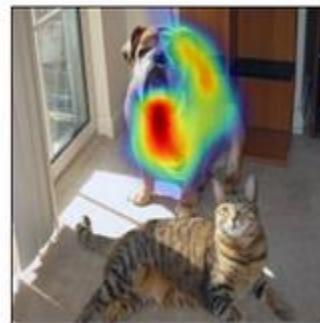
- Use average gradient received by the feature map in the last CNN layer as the corresponding weight for defining the Class Activation Maps.

$$\alpha_k^c = \underbrace{\frac{1}{Z} \sum_i \sum_j}_{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{}$$

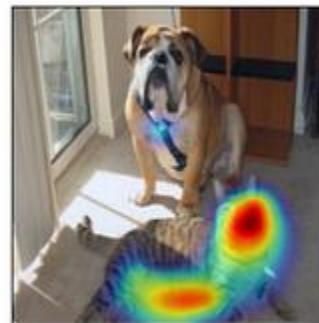
$$L_{\text{Grad-CAM}}^c = \text{ReLU} \left(\underbrace{\sum_k \alpha_k^c A^k}_{\text{linear combination}} \right)$$



(a) Original Image



(b) Cat Counterfactual exp



(c) Dog Counterfactual exp

Guided GradCAM

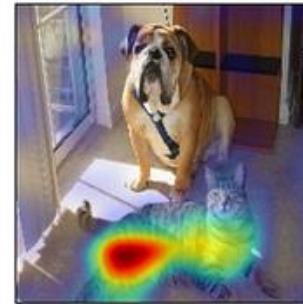
- For more fine grained details, run Guided BackProp and multiplying the resulting signals element wise with GradCAM.



(a) Original Image



(b) Guided Backprop ‘Cat’



(c) Grad-CAM ‘Cat’



(g) Original Image



(h) Guided Backprop ‘Dog’



(i) Grad-CAM ‘Dog’

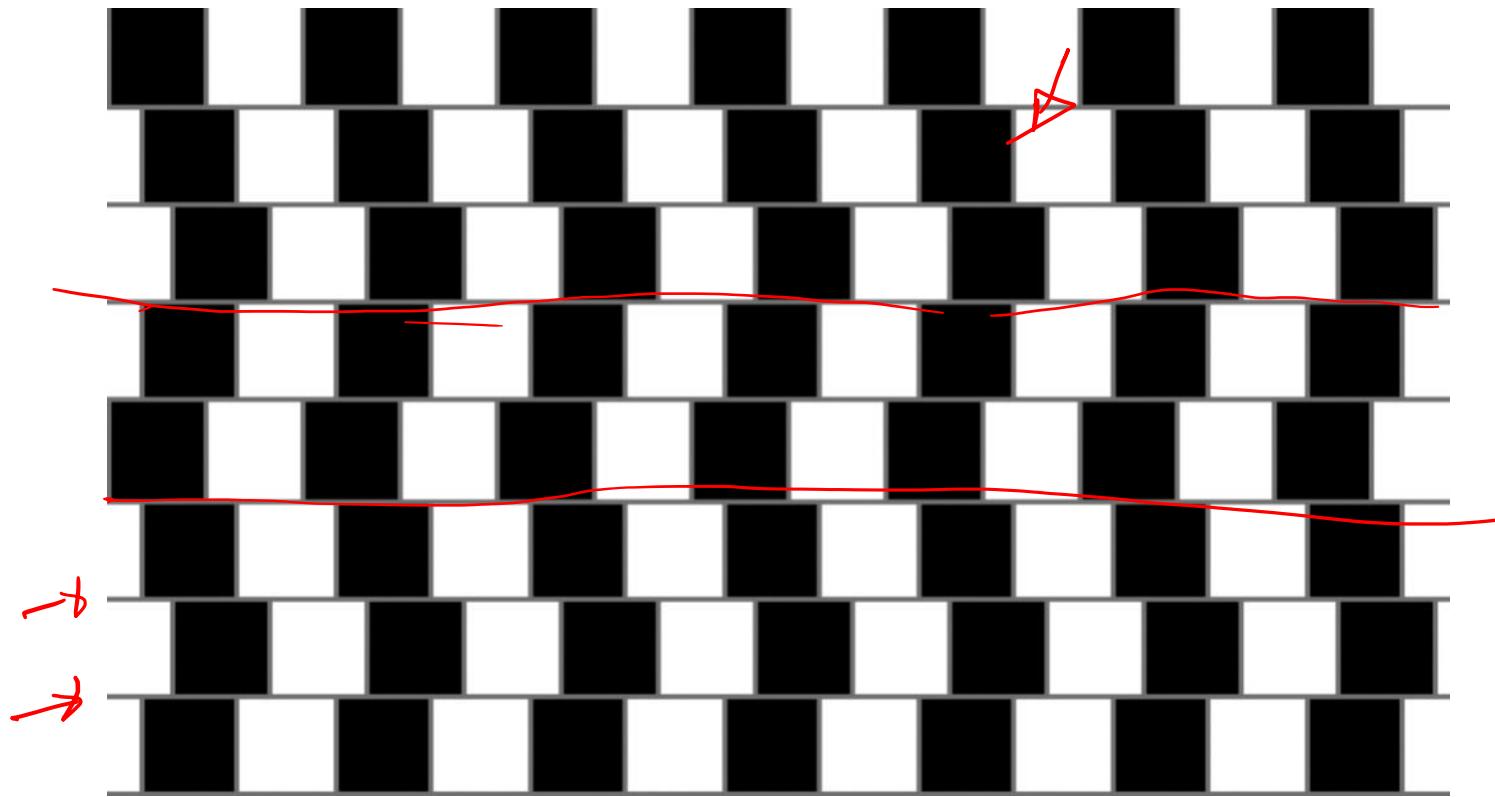
Robustness and Evasion Attacks

Outline

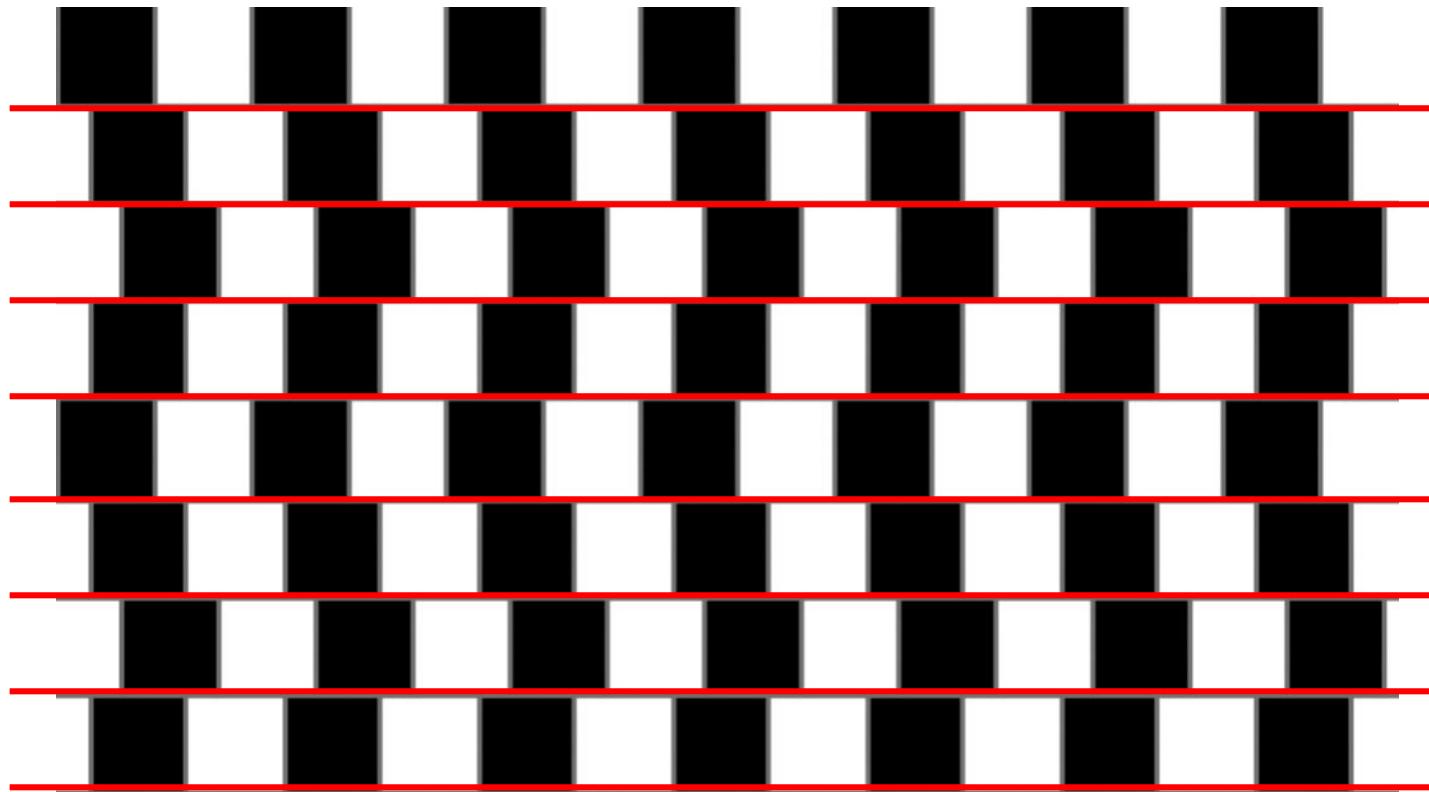
- Adversarial Attack
- White-box Evasion Attack
 - FGSM
 - C&W
- Transferability of Attack
- Black-box Evasion Attack
 - Jacobian-based Data Augmentation

GAN →
generative adversarial

Optical Illusions

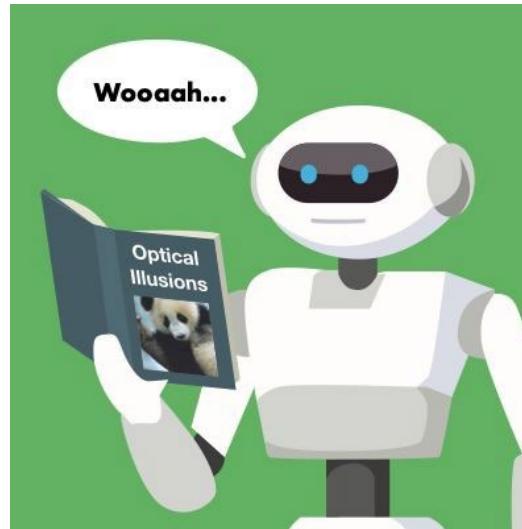


Optical Illusions

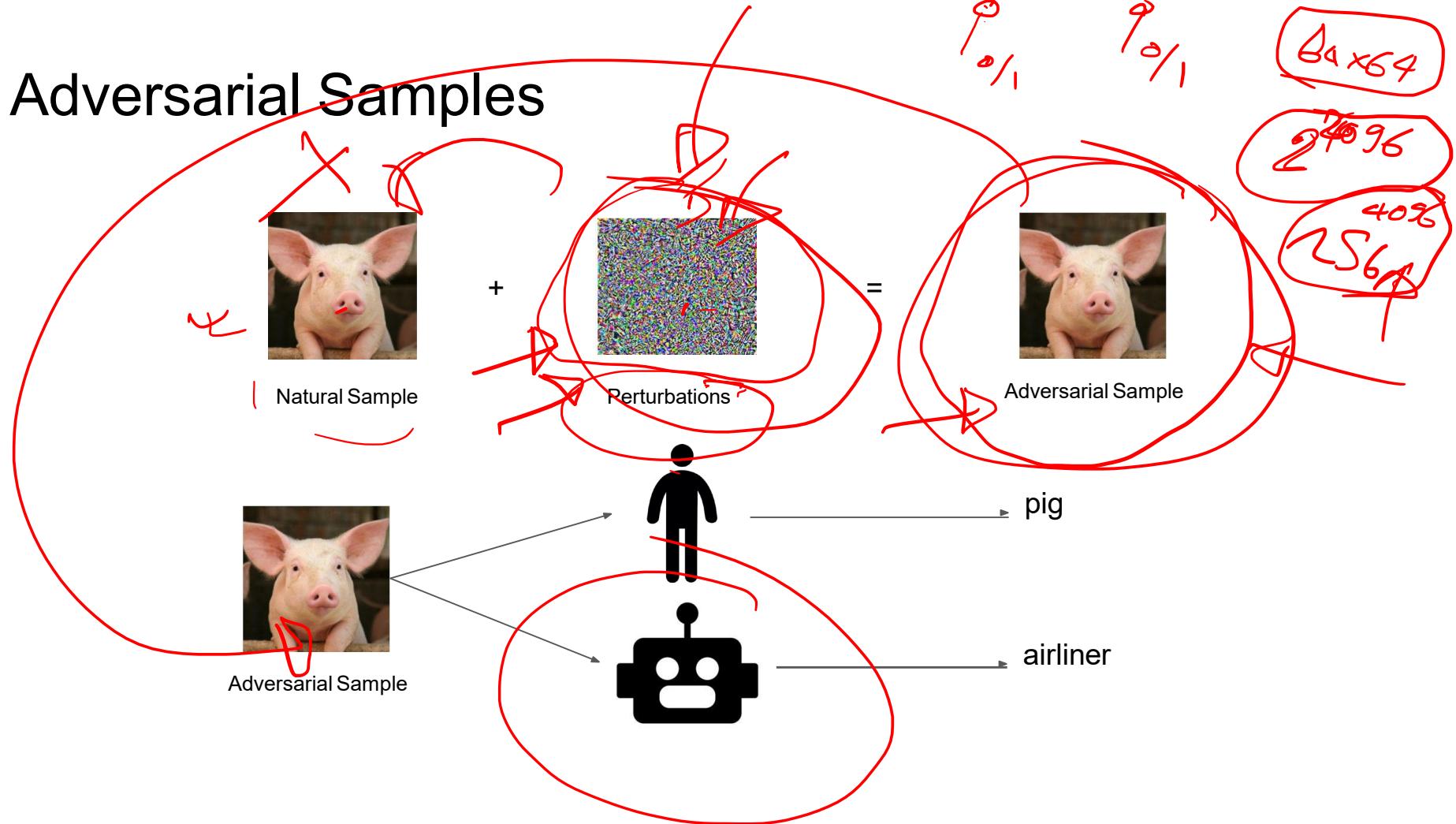


Robustness of ML Models

- Optical illusions trick human brains
- Can ML models be tricked?



Adversarial Samples

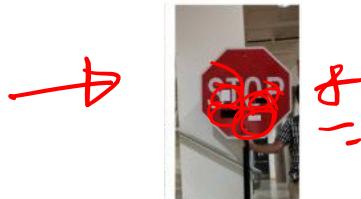


Driverless Car



[Sitawarin et al, 2018](#)

classified as : Stop Speed Limit (30 mph)

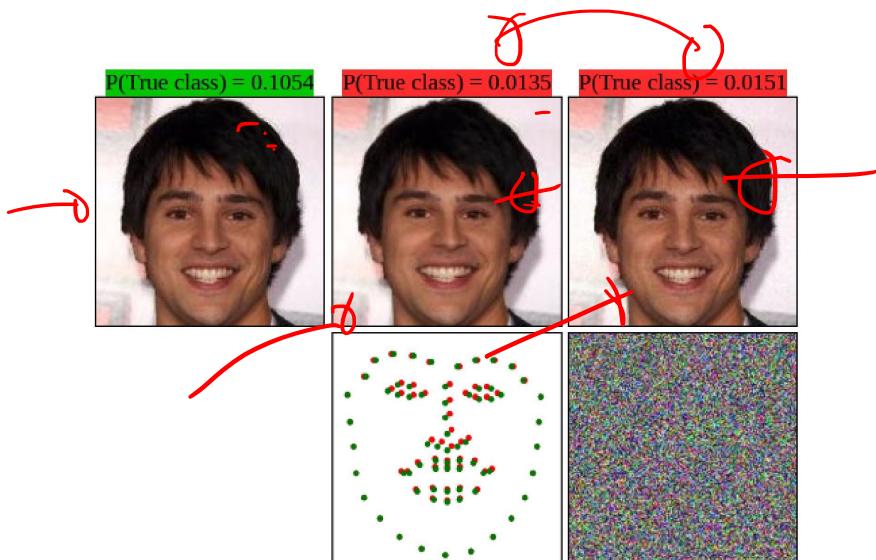


[Eykholt et al, 2018](#)

classified as : Speed Limit (45 mph)



Facial Recognition

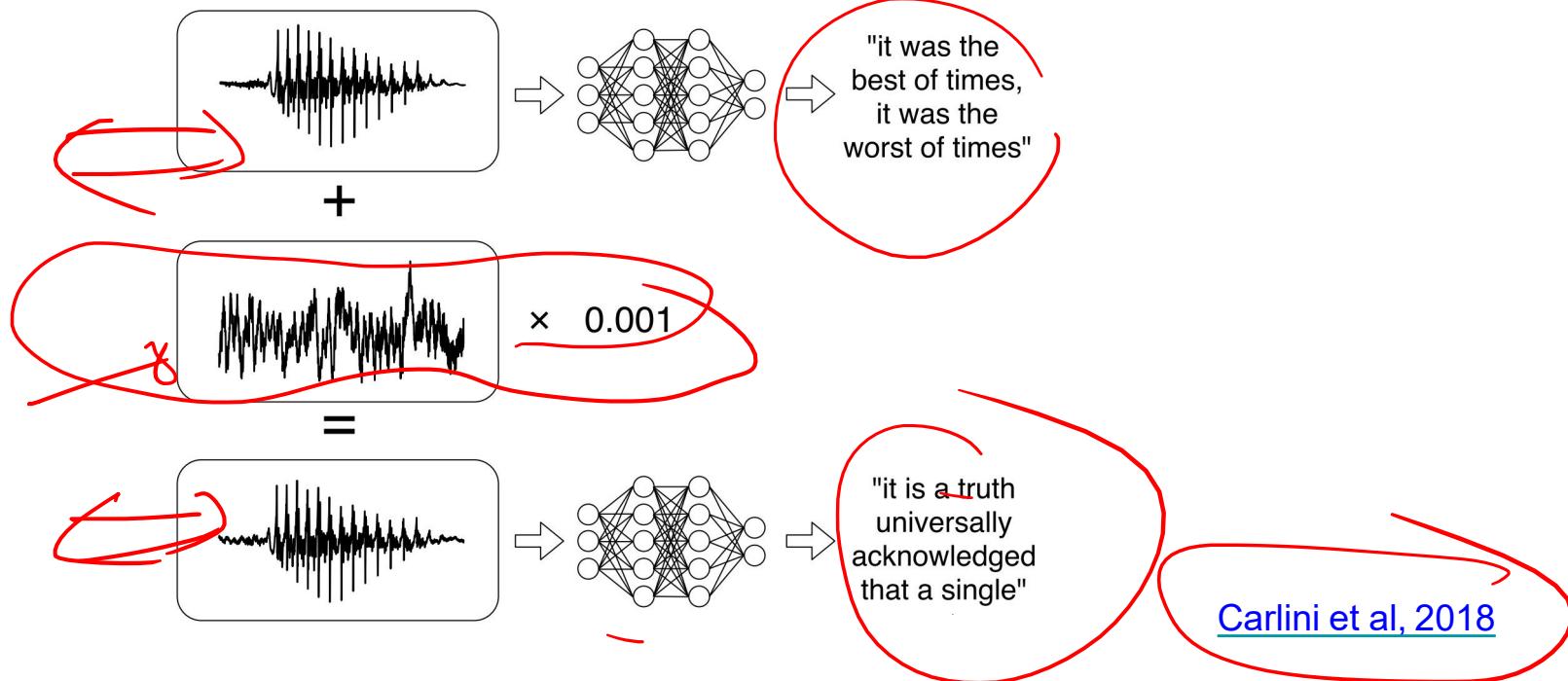


Dabouei et al, 2018

Spam Detections



Speech Recognition



Universal Adversarial Patch

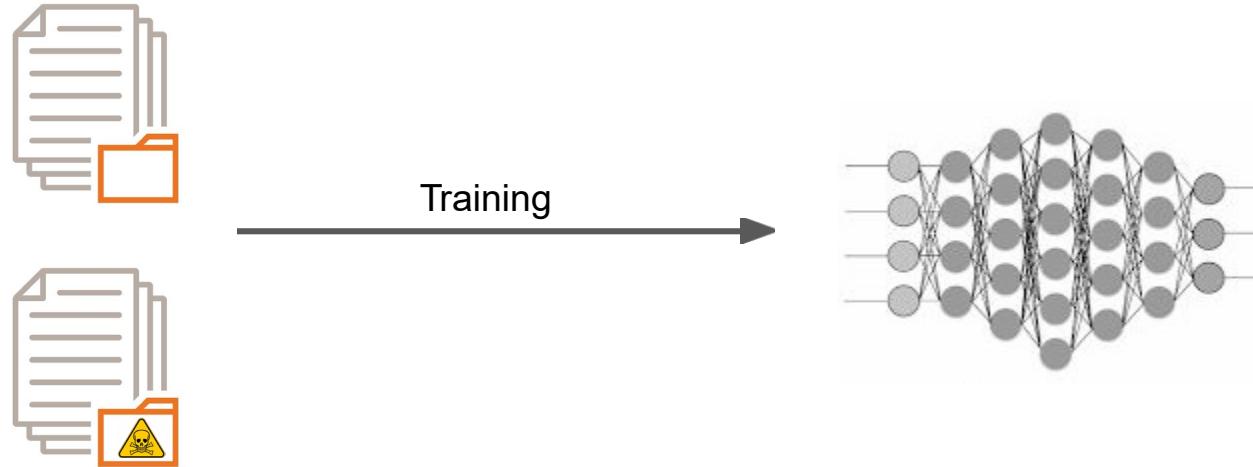


[Thys et al, 2019](#)

<https://www.youtube.com/watch?v=M1bFvK2S9g8>

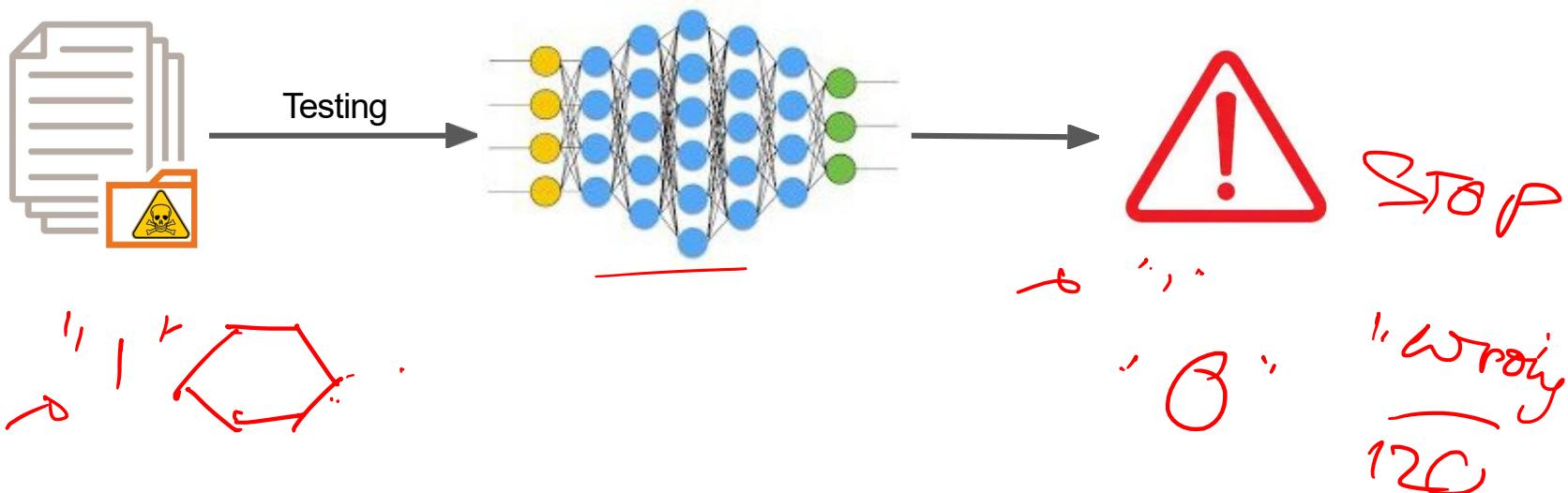
Types of Adversarial Attack

- Data Poisoning Attack
 - Insert poisonous samples during training



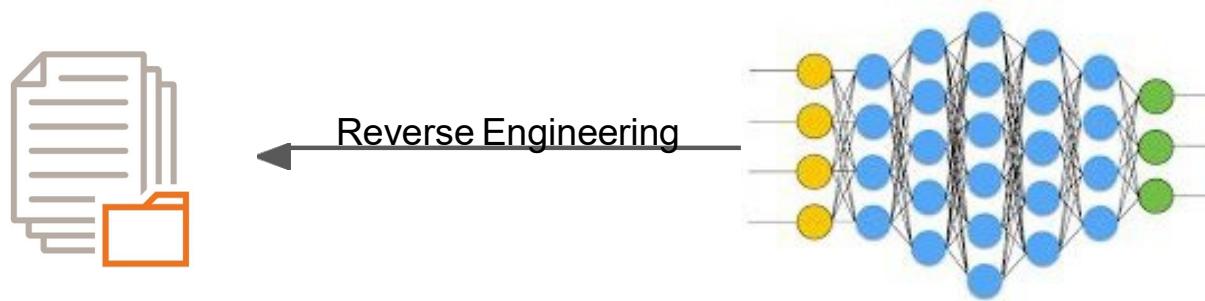
Types of Adversarial Attack

- Evasion Attack
 - Generate malicious samples to fool ML models



Types of Adversarial Attack

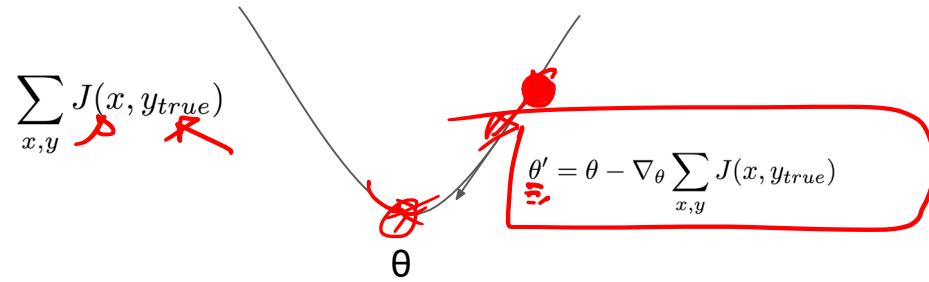
- Exploratory Attack
 - Reverse engineer user data from a trained model



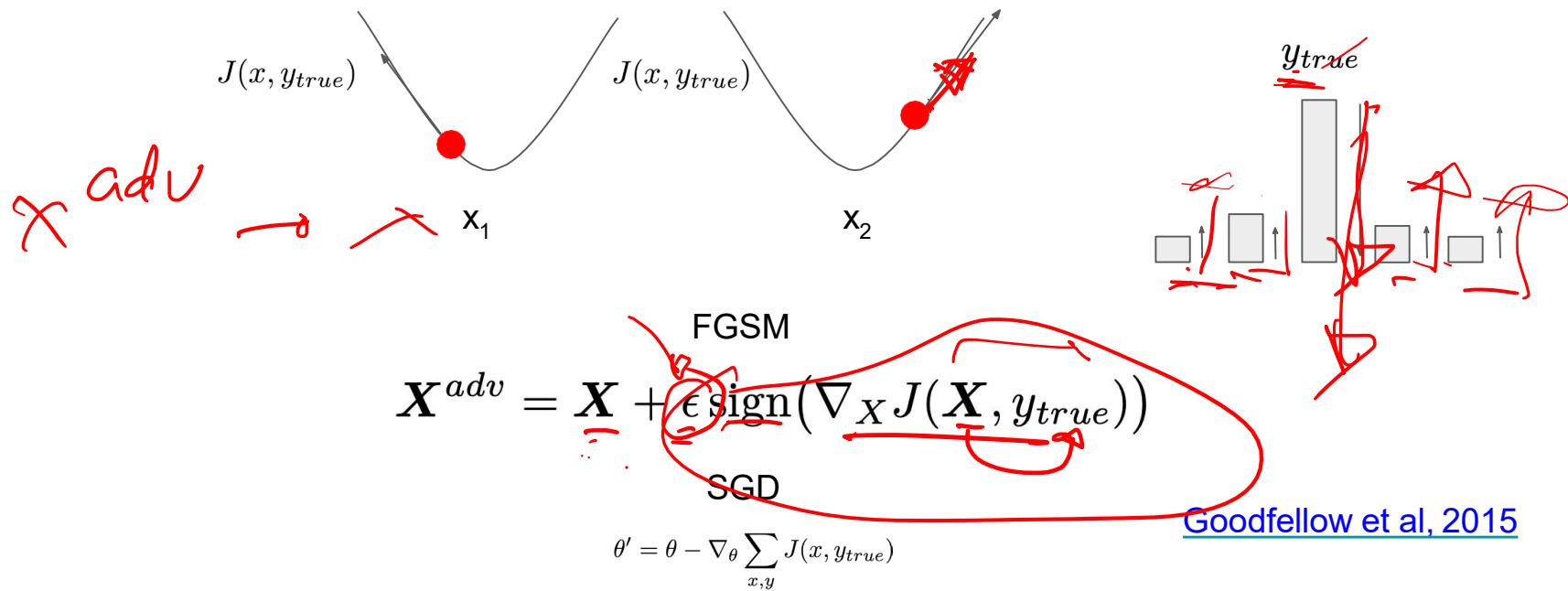
Types of Adversarial Attack

	Attack Phase	Goal
Evasion	Testing/deployment	Compromise Model Performance
Data Poisoning	Training	Compromise Model Performance
Exploratory	Testing/deployment	Explore Model Characteristics Reconstruct User Data

Training ML Models

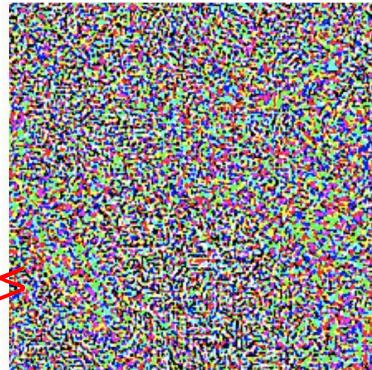


Fast Gradient Sign Method (FGSM)

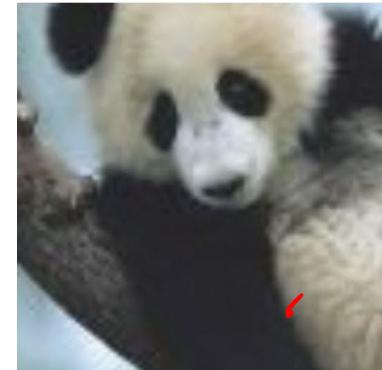


[Goodfellow et al, 2015](#)

Untargeted Adversarial Examples



=

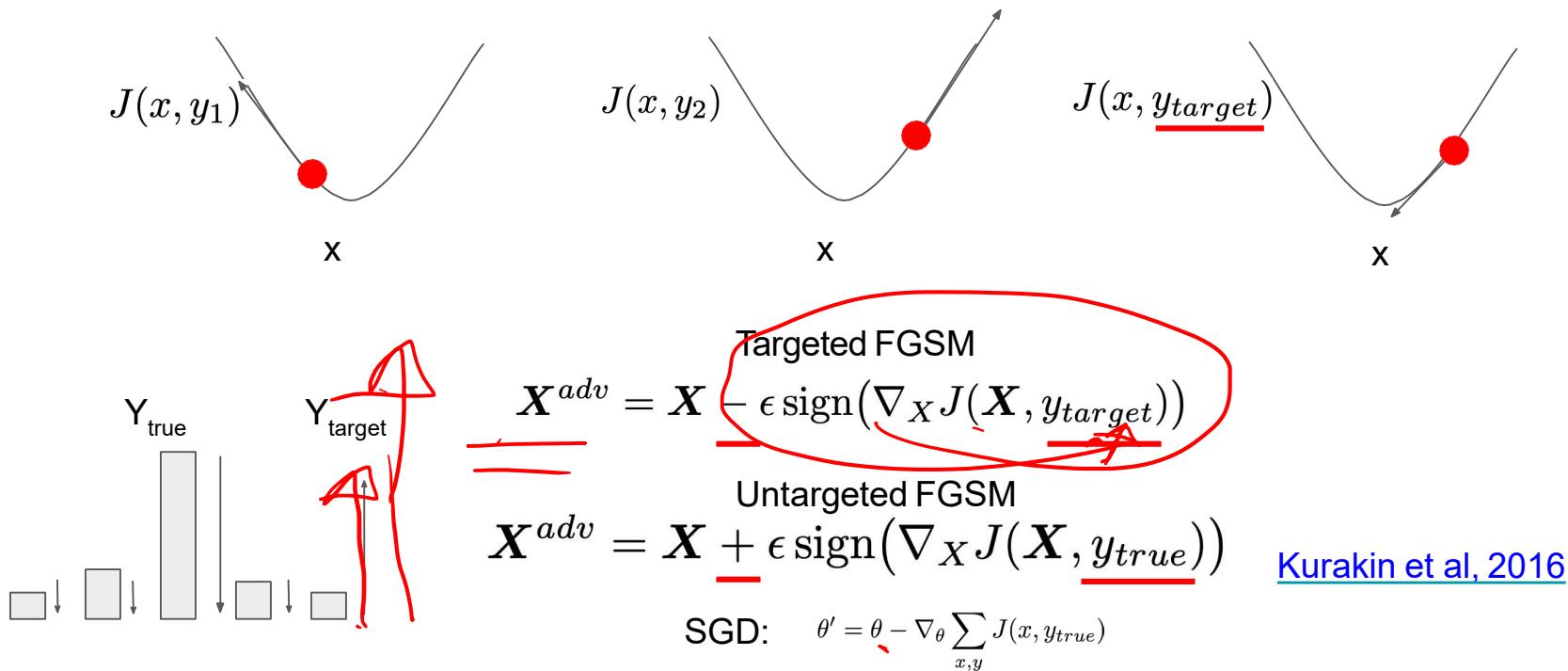


x
“panda”
57.7% confidence

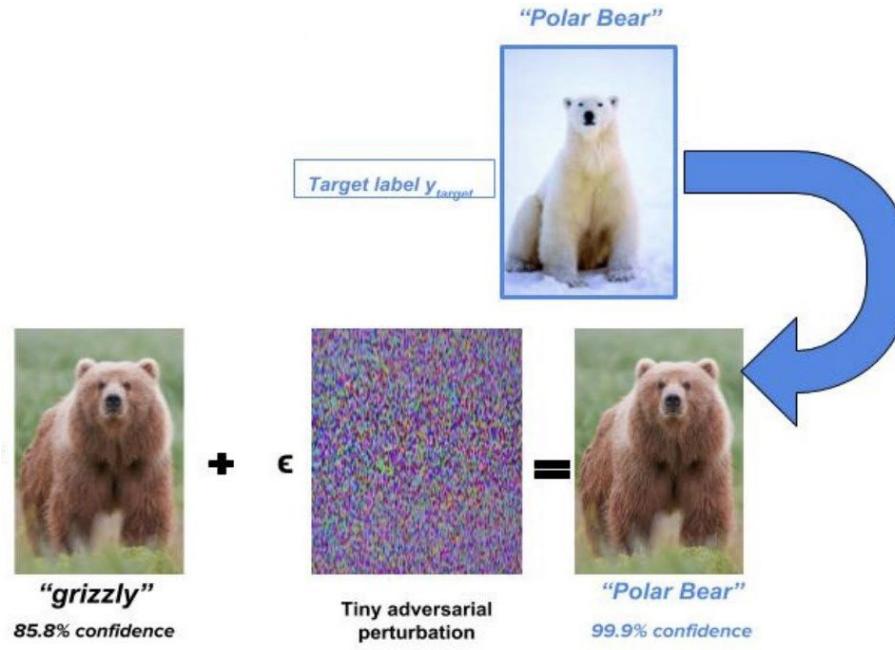
$\text{sign}(\nabla_x J(\theta, x, y))$
“nematode”
8.2% confidence

$x +$
 ~~$\epsilon \text{sign}(\nabla_x J(\theta, x, y))$~~
“gibbon”
99.3 % confidence
Goodfellow et al, 2015

Targeted FGSM



Targeted Adversarial Examples



[Younis et al, 2019](#)

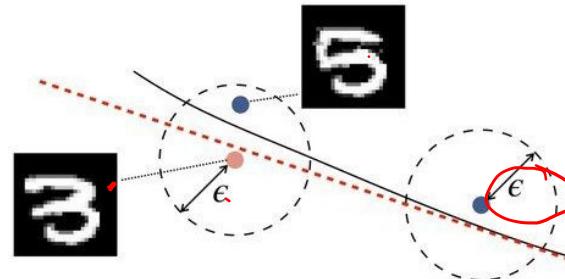
Basic Iterative Methods

- Untargeted Attack

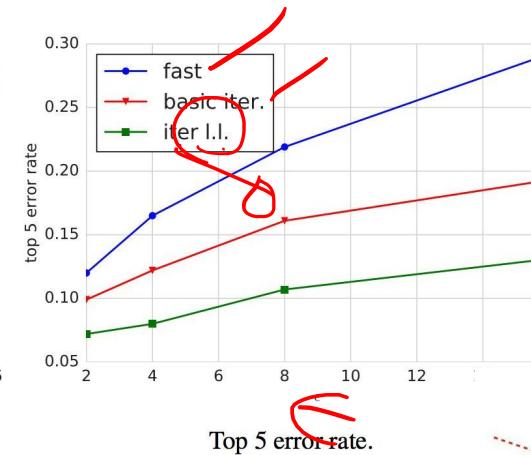
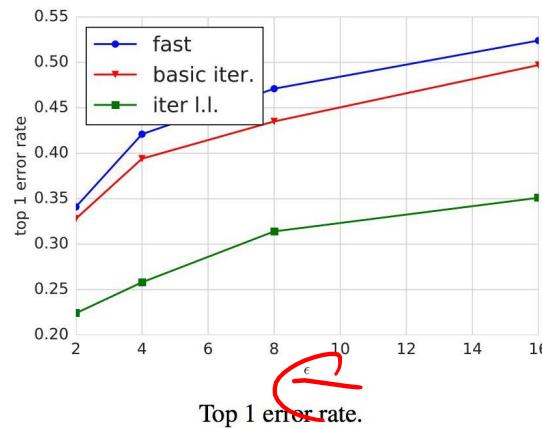
$$\overrightarrow{X}_{N+1}^{adv} = Clip_{X, \epsilon} \left\{ X_N^{adv} + \alpha \text{sign}(\nabla_X J(X_N^{adv}, y_{true})) \right\}$$

- Targeted Attack

$$X_{N+1}^{adv} = Clip_{X, \epsilon} \left\{ X_N^{adv} - \alpha \text{sign}(\nabla_X J(X_N^{adv}, y_{target})) \right\}$$



Error Rate and Perturbation Tolerance

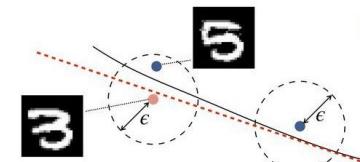


fast - FGSM

basic iter. - iterative untargeted FGSM

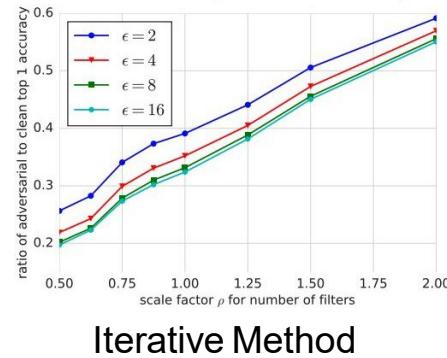
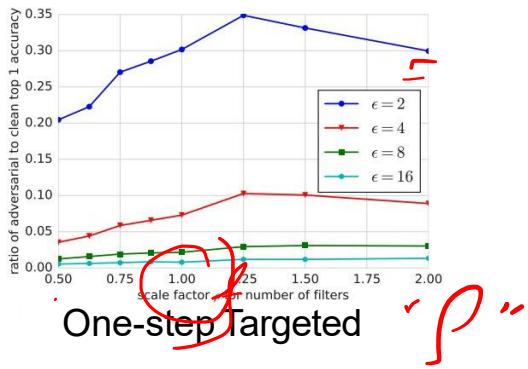
iter 1.1 - iteration using least likely target

$$y_{LL} = \arg \min_y \{p(y | \mathbf{X})\}$$



[Kurakin et al, 2016](#)

Model Capacity and Attacks



ρ - the factor in the number for InceptionNet

1 - unchanged

0.5 - keep half of the filters

[Kurakin et al, 2016](#)

C&W Attack

- C&W attack
 - perturb the sample in the direction of the target class
 - minimizes the distance from the original sample x

$$\begin{aligned} & \text{minimize } \mathcal{D}(x, x + \delta) \\ & \text{such that } C(x + \delta) = t \\ & \quad x + \delta \in [0, 1]^n \end{aligned}$$

D - distance function

C - classifier

x - original natural sample

δ - perturbations

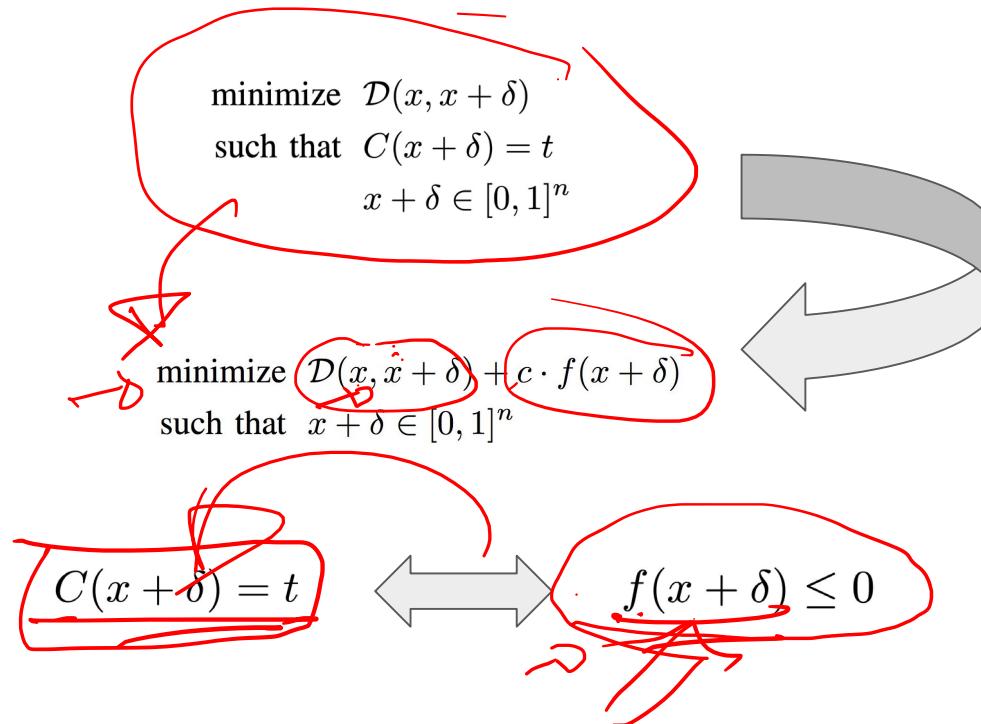
t - target class

Targeted FGSM

$$\cancel{\mathbf{X}^{adv}} = \mathbf{X} - \epsilon \operatorname{sign}(\nabla_{\mathbf{X}} J(\mathbf{X}, y_{target}))$$

[Carlini et al, 2017](#)

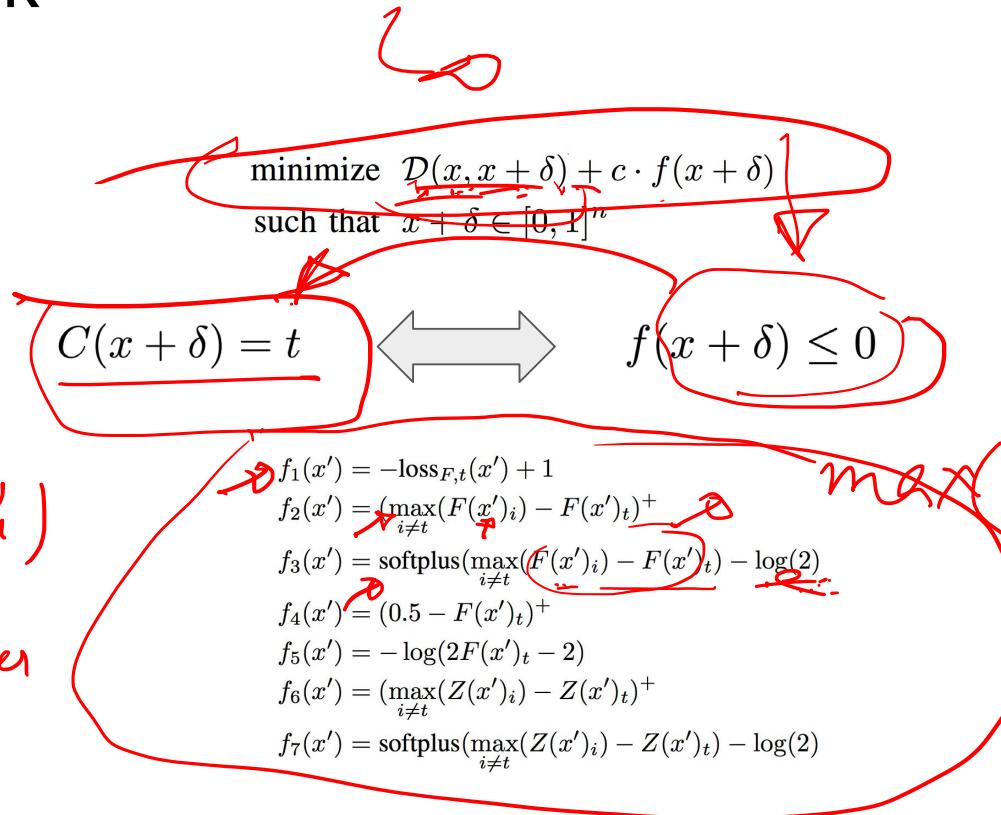
C&W Attack



[Carlini et al, 2016](#)

C&W Attack

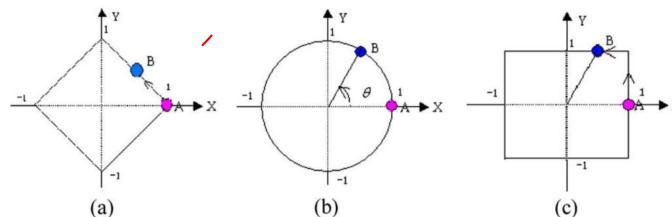
$x' = x + \delta$
t target
specified by attacker



Carlini et al, 2016

C&W L_∞ Attack

$$\text{minimize } c \cdot f(x + \delta) + \|\delta\|_\infty$$



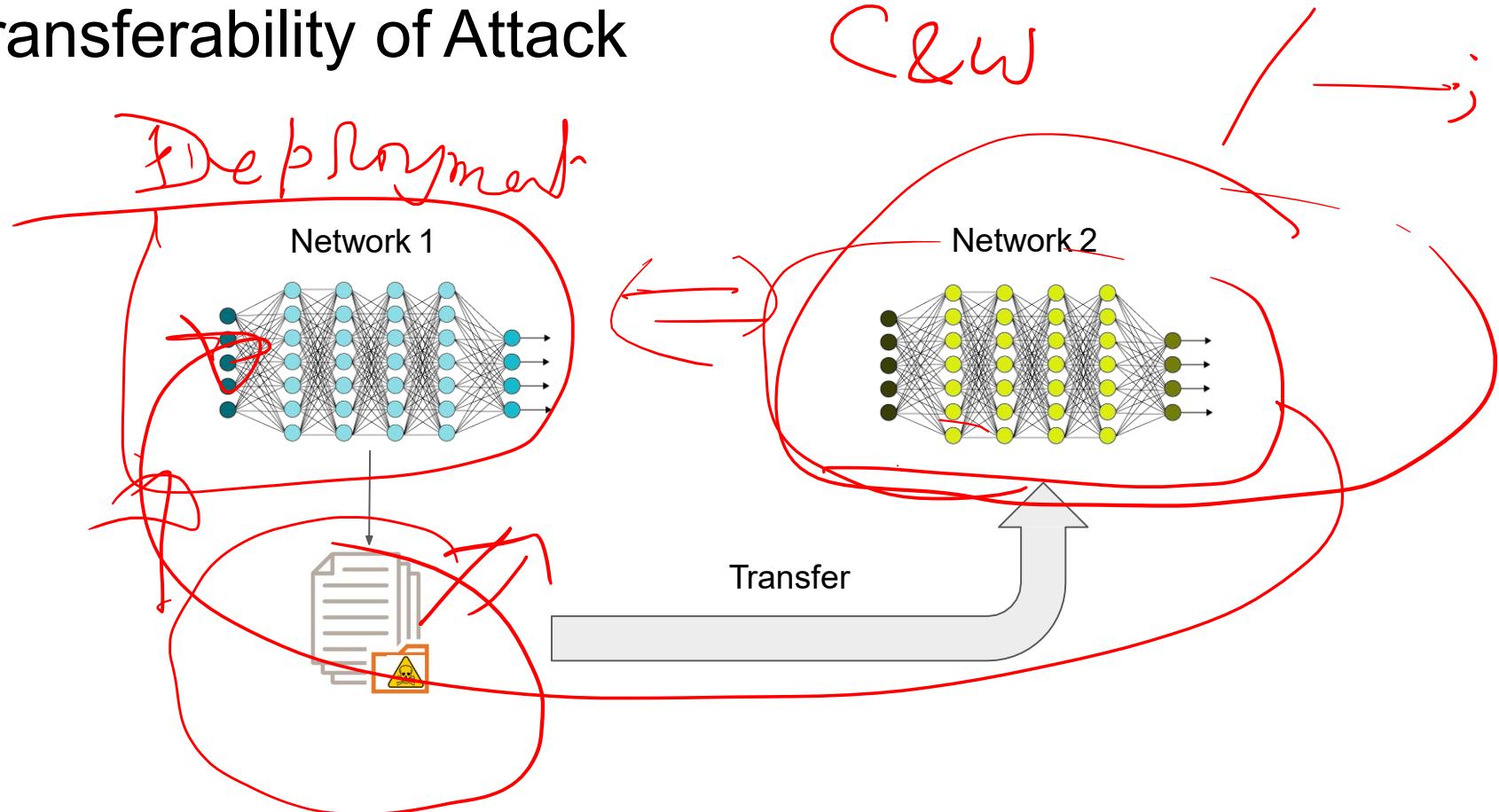
$$\|x\|_1 = \sum_{i=1}^n |x_i| \quad \|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2} \quad \|x\|_\infty = \max_{1 \leq i \leq n} |x_i|$$

FGSM

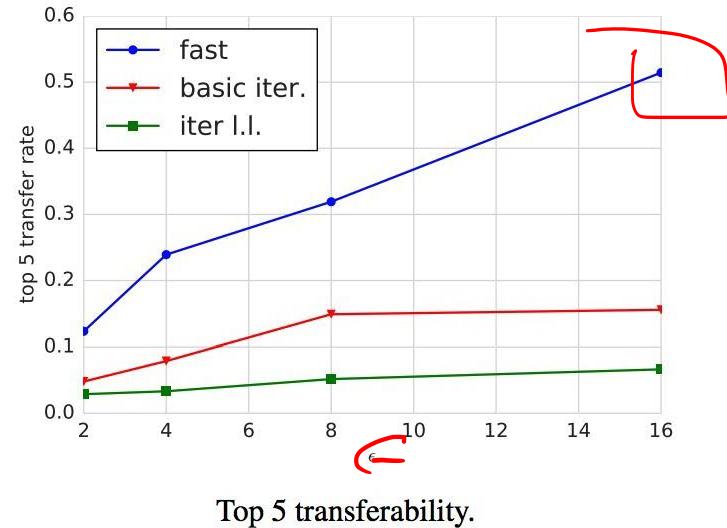
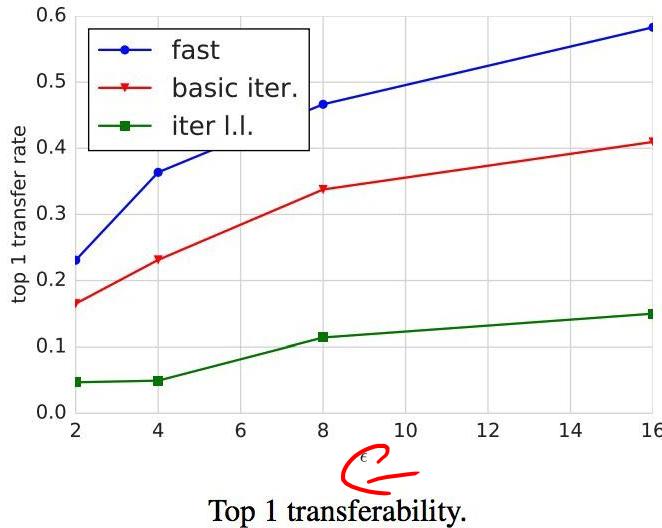
$$\mathbf{X}^{adv} = \mathbf{X} + \epsilon \text{ sign}(\nabla_{\mathbf{X}} J(\mathbf{X}, y_{true}))$$

[Carlini et al, 2016](#)

Transferability of Attack



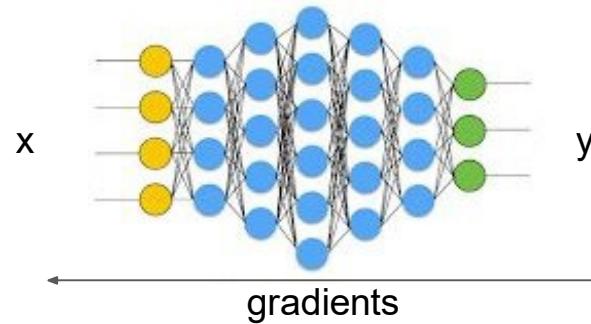
Transferability of Attack



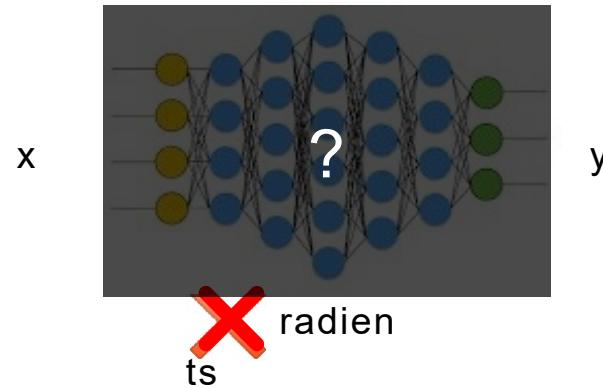
[Kurakin et al, 2017](#)

White-box and Black-box Attack

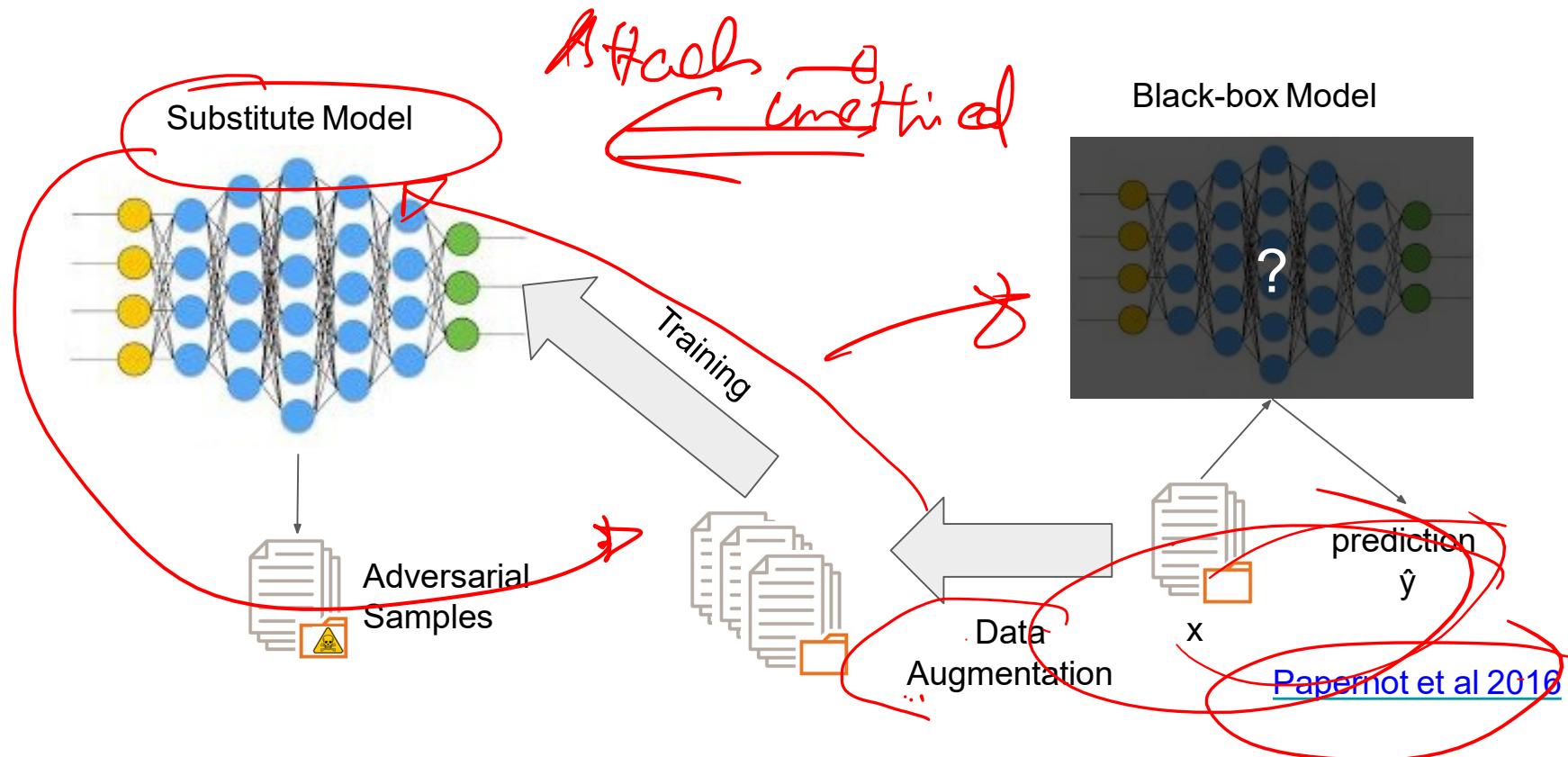
White-box Setting



Black-box Setting

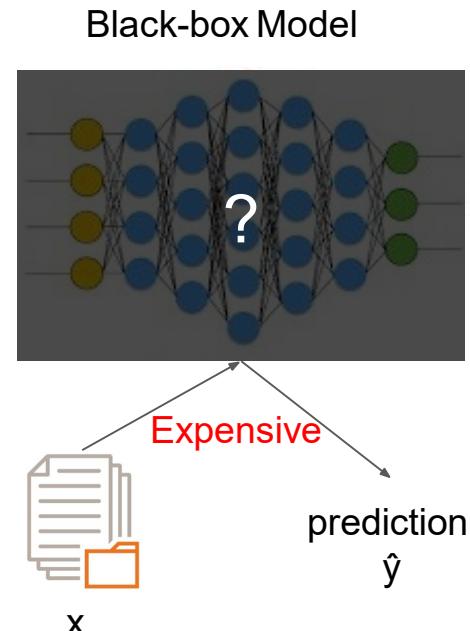
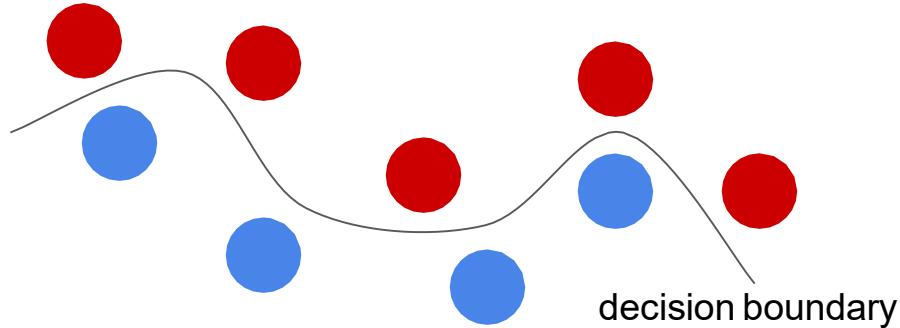


Substitute Model for Black-box Adversarial Attack



Data Augmentation for the Substitute Model

- Data annotation using the black-box model is expensive
- It's difficult to find a good dataset x to probe the performance of the black-box model

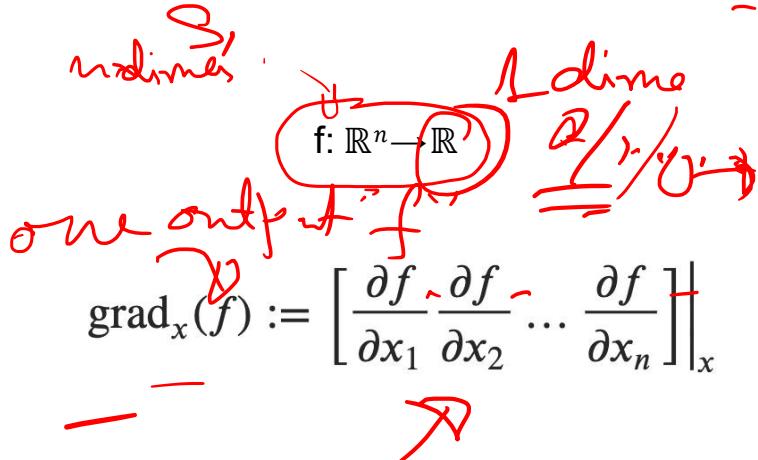


Jacobian-based Data Augmentation

- Start with an initial dataset $S_0 = \{x_i\}$
- Expand it in the direction of the model prediction \hat{y}_i for each x_i

$$S_{\rho+1} = \{\vec{x} + \lambda \cdot \text{sgn}(J_F[\tilde{O}(\vec{x})]) : \vec{x} \in S_\rho\} \cup S_\rho = \emptyset$$

prediction of
the black-box
model

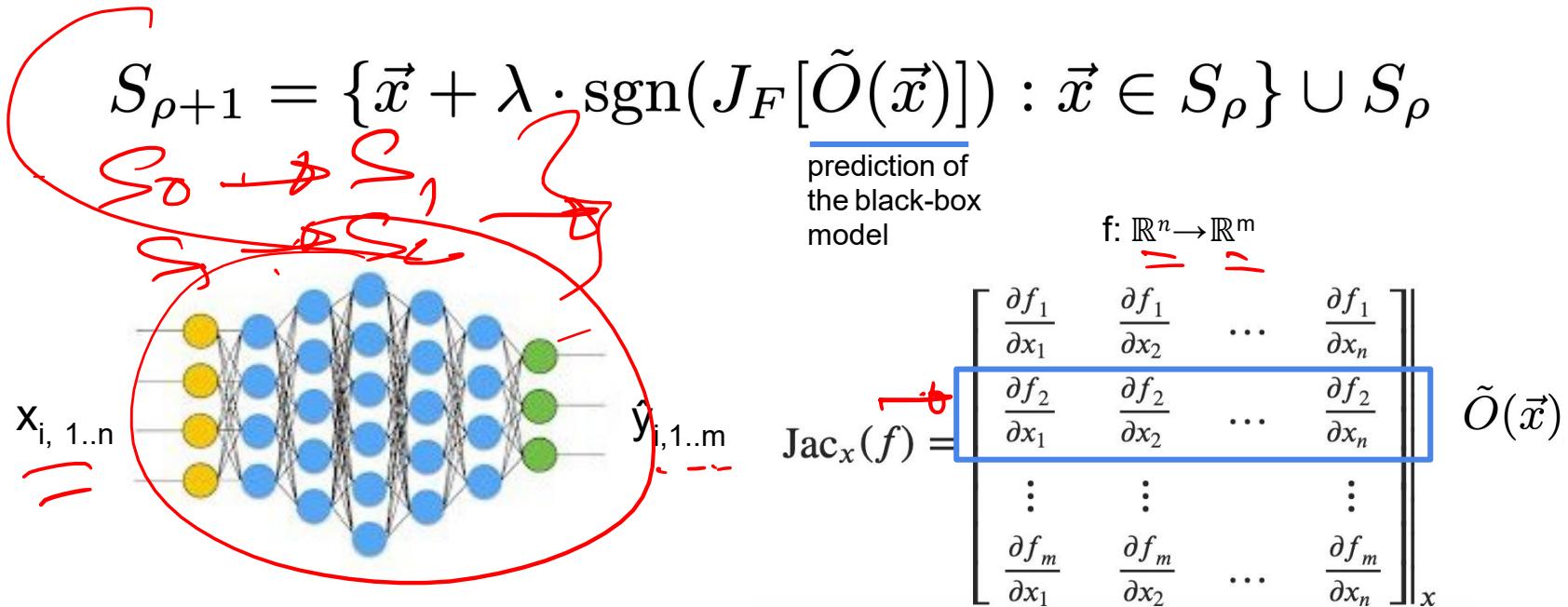


$f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ *m outputs in the sub white model*

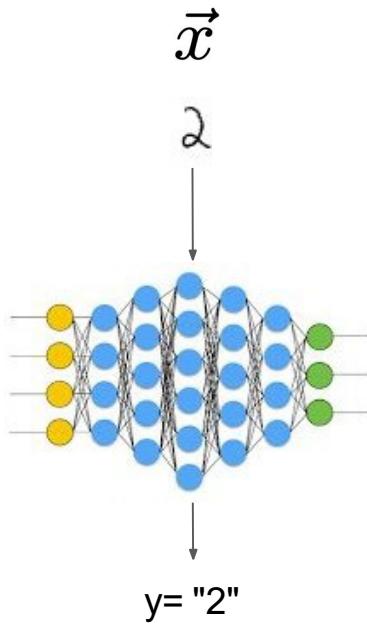
$$\text{Jac}_x(f) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \dots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \Big|_x$$

Jacobian-based Data Augmentation

- Start with an initial dataset $S_0 = \{x_i\}$
- Expand it in the direction of the model prediction \hat{y}_i for each x_i



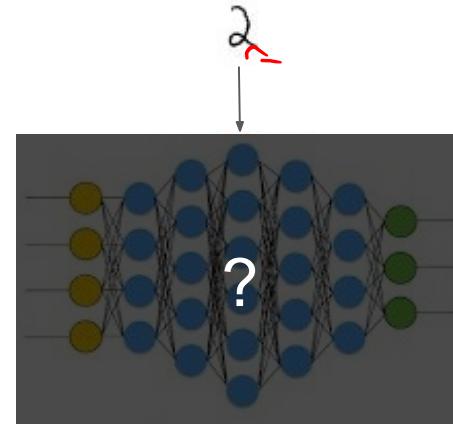
Jacobian-based Data Augmentation



$$\vec{x}' = \vec{x} + \lambda \cdot \text{sgn}(J_F[\tilde{O}(\vec{x})])$$

$$\text{Jac}_x(f) = \left[\begin{array}{cccc|c} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} & 1 \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} & 2 \\ \vdots & \vdots & & \vdots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} & m \end{array} \right]_x$$

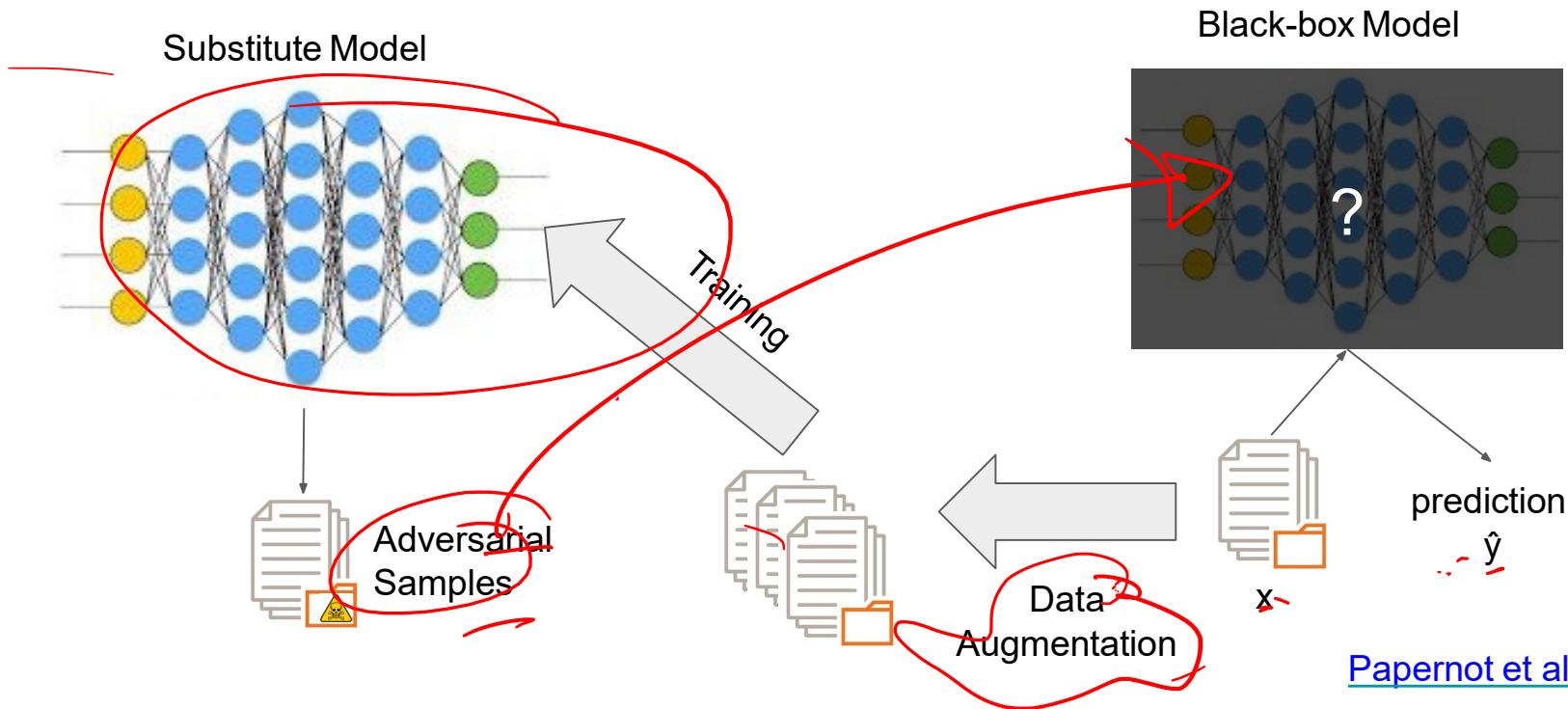
$$y = \tilde{O}(\vec{x}')$$



$$\tilde{O}(\vec{x})] = "1"$$

$=$

Substitute Model for Black-box Adversarial Attack



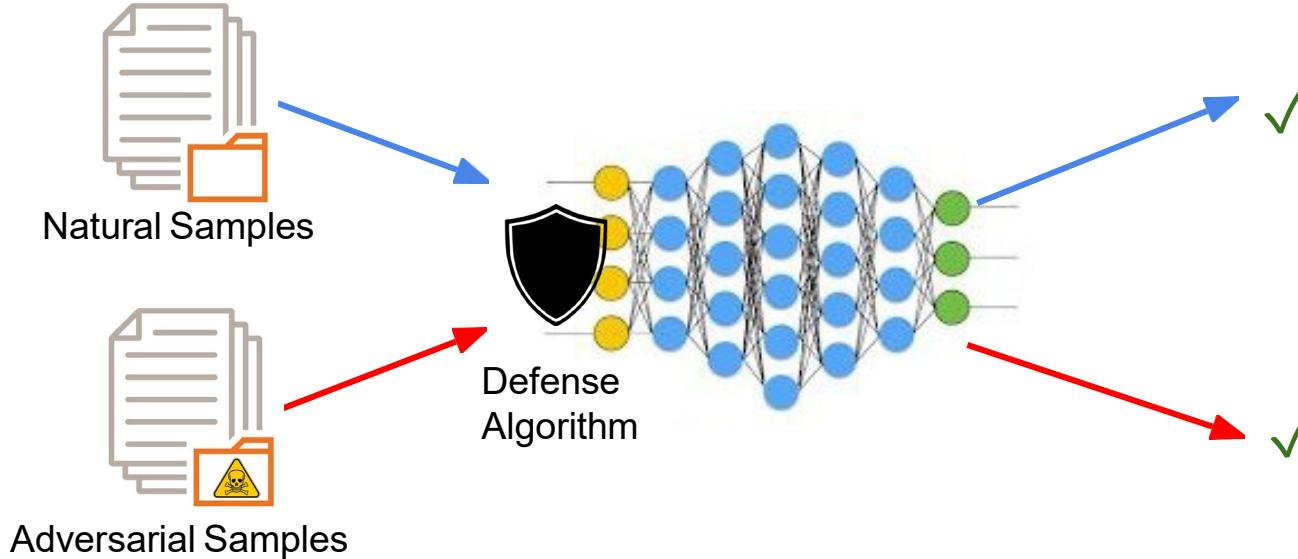
Papernot et al 2016

Adversarial Defense

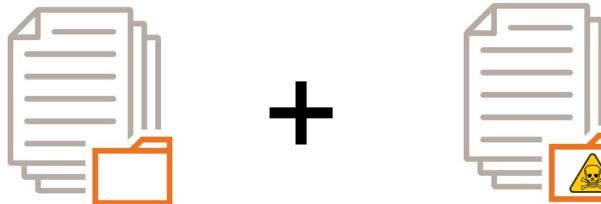
Outline

- Adversarial Defense
- Defense Strategies
 - Adversarial Training
 - Input Transformations
 - Stochastic Gradients
- Obfuscated Gradients and BPDA

Adversarial Defense



Adversarial Training



+

Natural Samples

Adversarial Samples

$$\tilde{J}(\theta, x, y) = \alpha J(\theta, x, y) + (1 - \alpha) J(\theta, x^{adv}, y)$$

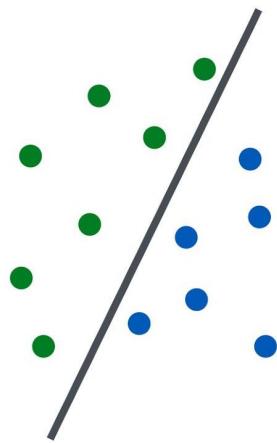
Loss Function

Natural Samples

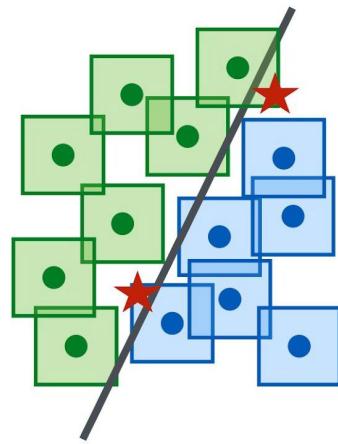
Adversarial Samples

[Goodfellow et al, 2014](#)

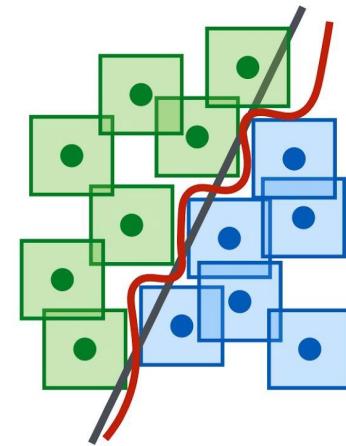
Adversarial Training



Natural Samples



Natural Samples with L_∞
Perturbation Space



Adversarial Training

[Madry et al, 2017](#)

Results on FGSM

- Accuracy on Adversarial Examples

$$\begin{gathered} \text{FGSM} \\ \mathbf{X}^{adv} = \mathbf{X} + \epsilon \operatorname{sign}(\nabla_{\mathbf{X}} J(\mathbf{X}, y_{true})) \end{gathered}$$

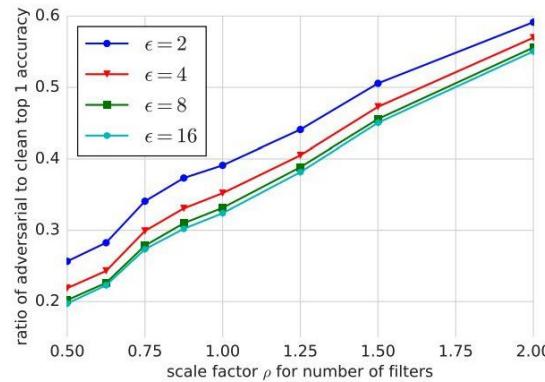
		Clean	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 8$	$\epsilon = 16$
Baseline (standard training)	top 1	78.4%	30.8%	27.2%	27.2%	29.5%
	top 5	94.0%	60.0%	55.6%	55.1%	57.2%
Adv. training	top 1	77.6%	73.5%	74.0%	74.5%	73.9%
	top 5	93.8%	91.7%	91.9%	92.0%	91.4%
Deeper model (standard training)	top 1	78.7%	33.5%	30.0%	30.0%	31.6%
	top 5	94.4%	63.3%	58.9%	58.1%	59.5%
Deeper model (Adv. training)	top 1	78.1%	75.4%	75.7%	75.6%	74.4%
	top 5	94.1%	92.6%	92.7%	92.5%	91.6%

Dataset: ImageNet

[Kurakin et al, 2017](#)

Results on FGSM

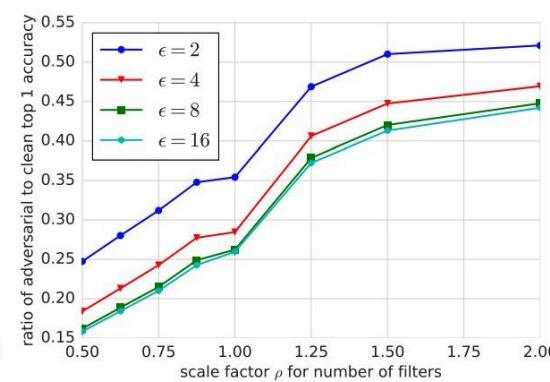
- Adversarial Accuracy / Clean Image Accuracy
 - Ratio $\rightarrow 1$ successful adversarial attack
 - Ratio $\rightarrow 0$ successful adversarial defense



No adversarial training, “basic iter.” adv. examples

fast - FGSM

basic iter. - iterative untargeted FGSM



With adversarial training, “basic iter.” adv. examples

[Kurakin et al, 2017](#)

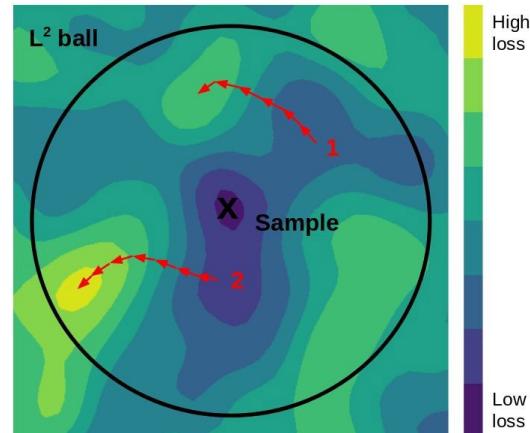
Flexibility

- Plug-in any attack techniques

$$\tilde{J}(\theta, x, y) = \alpha J(\theta, x, y) + (1 - \alpha) J(\theta, x^{adv}, y)$$

- Examples
 - FGSM
 - Projected Gradient Descent (PGD) ([Madry et al, 2017](#))

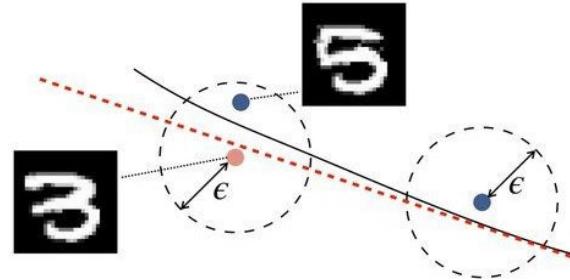
$$\max_{\mathbf{x}' : \|\mathbf{x}' - \mathbf{x}\|_\infty < \alpha} \mathcal{L}(\mathbf{x}', y; \theta)$$



Computational Costs

- Costs Associated with Generating Adversarial Samples

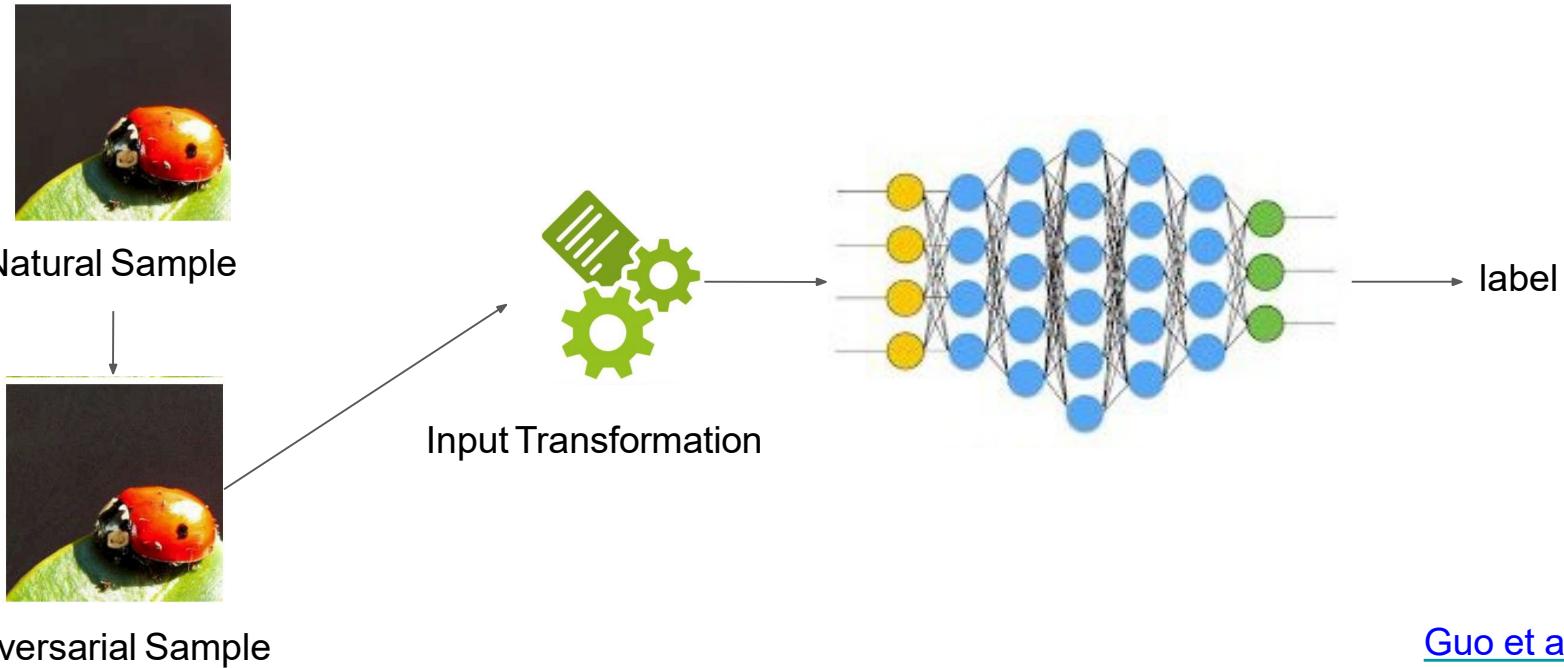
$$\mathbf{X}_{N+1}^{adv} = Clip_{X,\epsilon} \left\{ \mathbf{X}_N^{adv} + \alpha \text{sign}(\nabla_X J(\mathbf{X}_N^{adv}, y_{true})) \right\}$$



$$\tilde{J}(\theta, x, y) = \alpha J(\theta, x, y) + (1 - \alpha) J(\theta, x^{adv}, y)$$

Input Transformations

https://wiki.math.uwaterloo.ca/statwiki/index.php?title=Counteracting_Adversarial_Images_Using_Input_Transformations



Input Transformations

- Goal: Disrupt Adversarial Perturbations
- Image cropping/re-scaling
- Bit-depth reduction



16.7 Million
Colors

256
Colors

16
Colors

[Guo et al, 2018](#)

Input Transformations

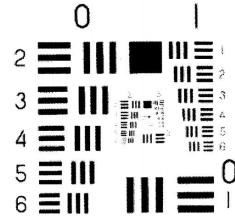
- Goal: Disrupt Adversarial Perturbations
- Image cropping/re-scaling
- Bit-depth reduction
- JPEG compression
- Total variation minimization
- Image quilting



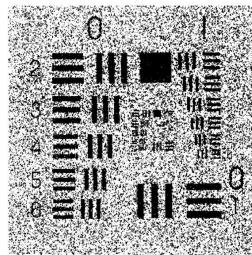
[Guo et al, 2018](#)

Total Variation Minimization

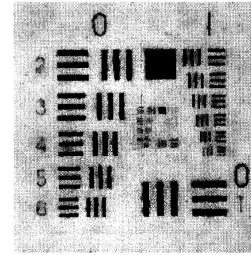
- Generate a denoised image \mathbf{z} by minimizing TV



Original Image



Noisy Image



Denoised Image minimizing TV

$$\text{TV}_p(\mathbf{z}) = \sum_{k=1}^K \left[\sum_{i=2}^N \|\mathbf{z}(i, :, k) - \mathbf{z}(i-1, :, k)\|_p + \sum_{j=2}^N \|\mathbf{z}(:, j, k) - \mathbf{z}(:, j-1, k)\|_p \right]$$

Transformed Image

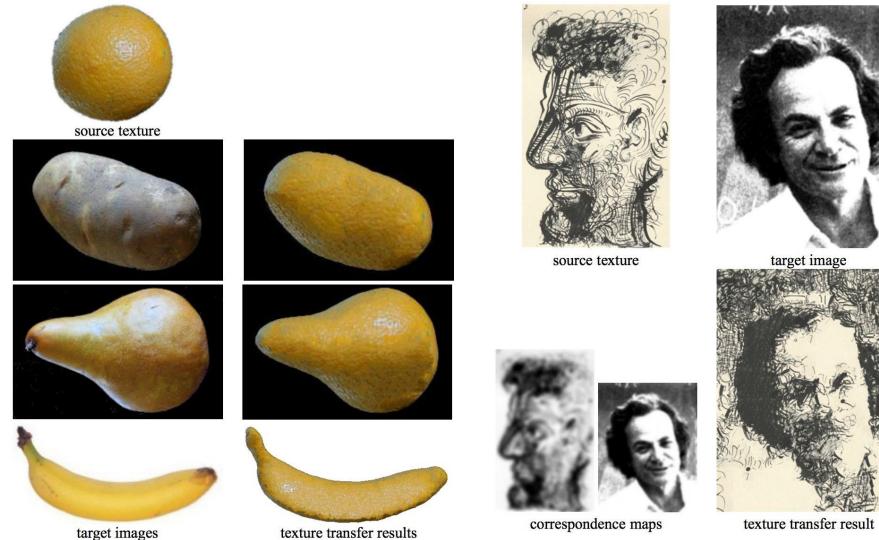
row variance

column variance

[Rudin et al, 1992](#)

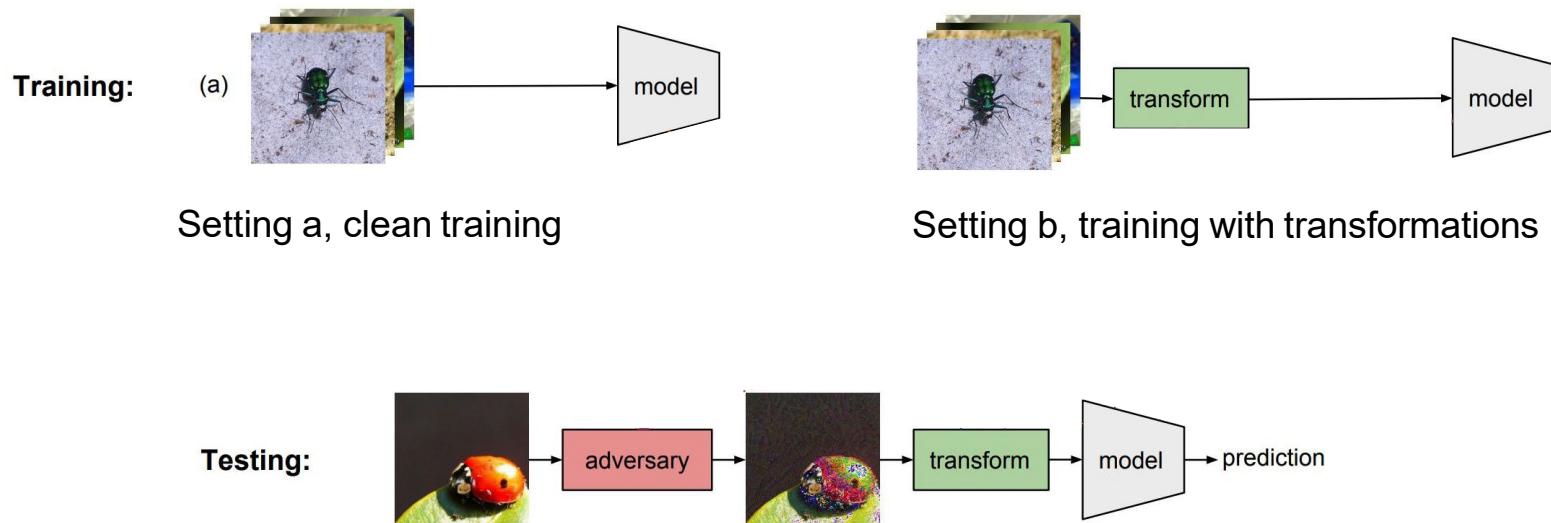
Image Quilting

- Synthesizes images by piecing together small patches taken from a database of image patches
- Database contains only clean images



[Efros et al, 2001](#)

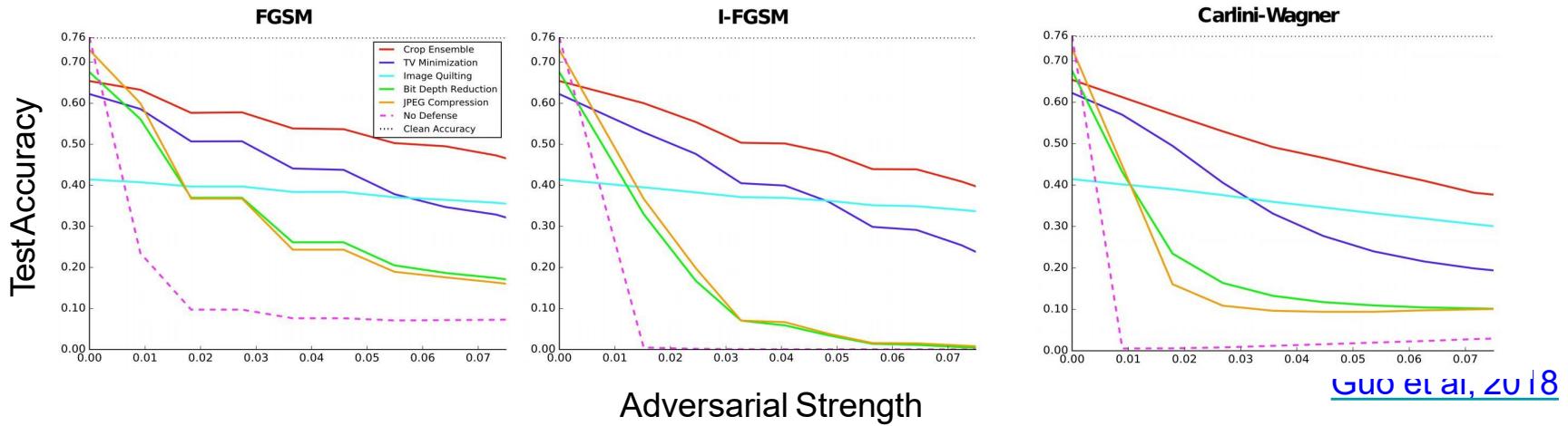
Input Transformation Defense



[Guo et al, 2018](#)

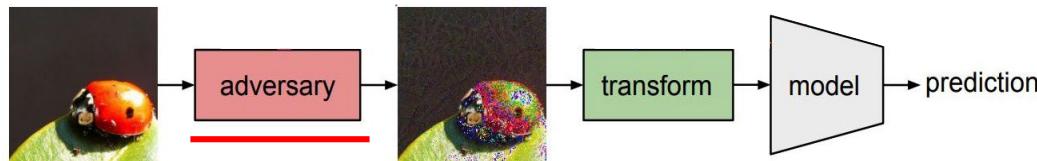
Results with Clean Image Training

ResNet on ImageNet



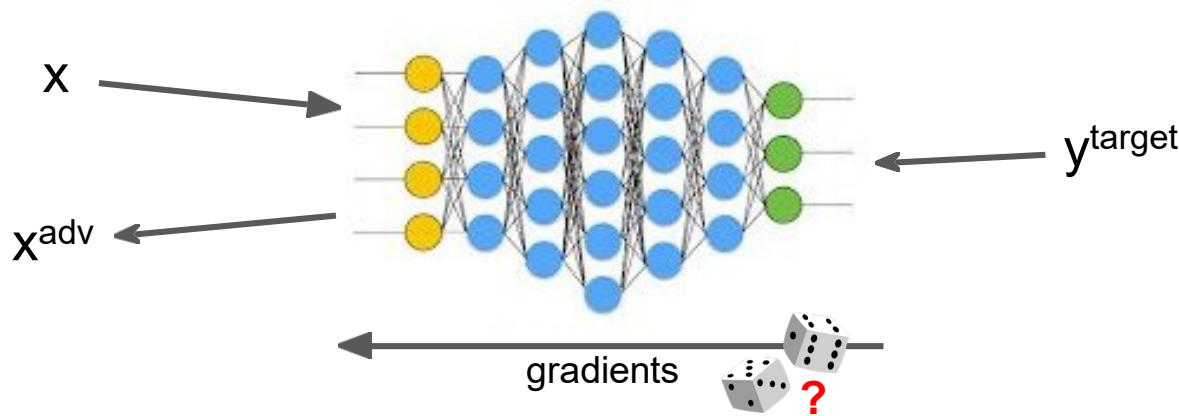
Gradient Shattering

- Can we design specialized attacks that target input transformations?
 - We show previously the results using FGSM and C&W
- Input Transformations belongs to a family of defense methods that causes Gradient Shattering



Train our own adversary that targets input transformations?

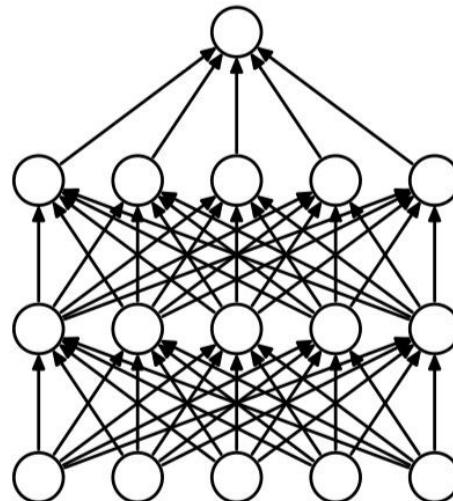
Stochastic Gradients



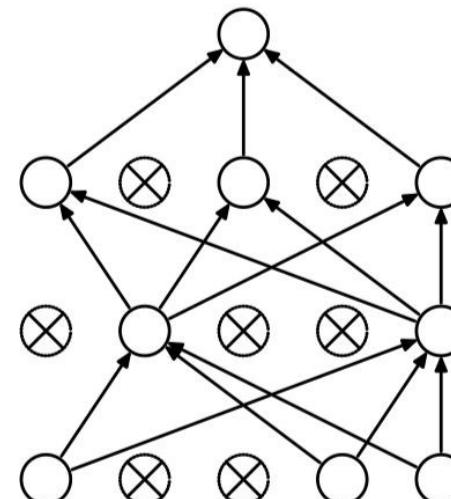
$$\mathbf{X}_{N+1}^{adv} = Clip_{X,\epsilon} \left\{ \mathbf{X}_N^{adv} + \alpha \operatorname{sign}(\nabla_X J(\mathbf{X}_N^{adv}, y_{true})) \right\}$$

Dropout

- Dropout randomly turns off activations by a fixed probability r
- Originally introduced to prevent overfitting

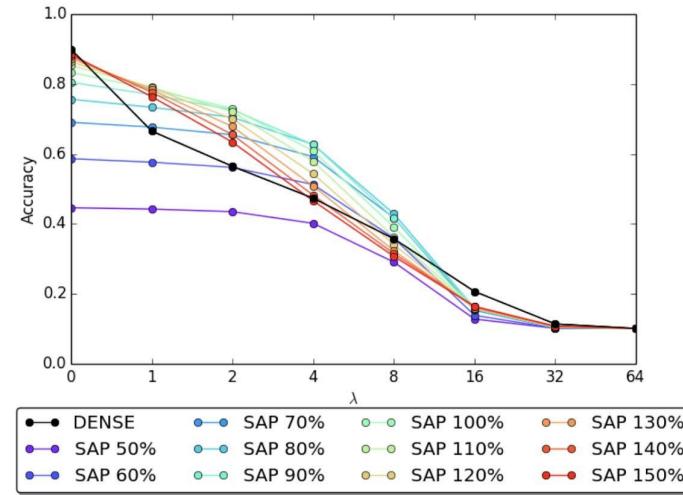
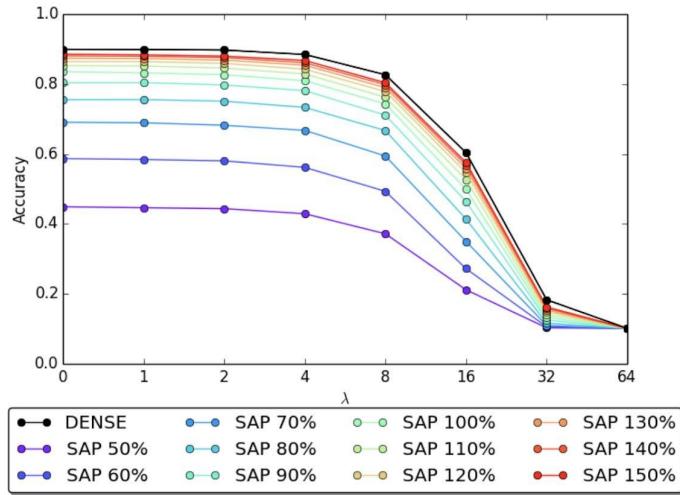


(a) Standard Neural Net



(b) After applying dropout.

Defense Results



SAP % - the percentages of samples drawn for each layer
 λ - perturbation strength

[Dhillon et al, 2018](#)

Summary of Defense Strategies

Defense Methods	General Idea
Adversarial Training	Mixing adversarial samples with natural samples during training
Input Transformation	Adding transformation to make defense non-differentiable
Stochastic Gradients	Causing gradients to be randomized

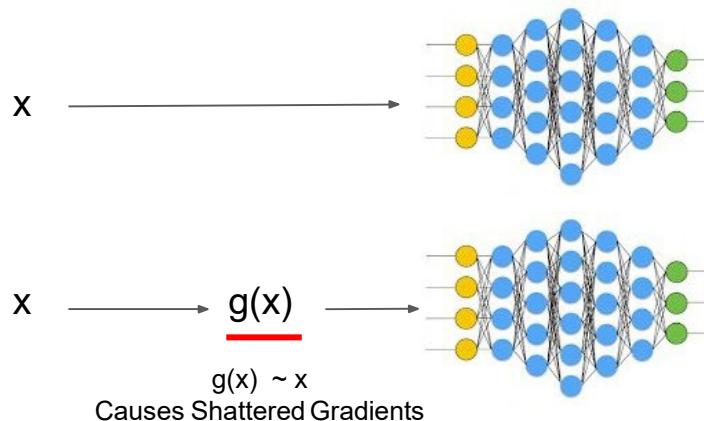
Obfuscated Gradients

- A defense method is said to achieve Obfuscated Gradients if
 - It prevents the attack methods from utilizing useful gradient information
- Shattered Gradients
 - Present a defense method that is non-differentiable or numerically unstable
 - e.g., Input Transformations
- Stochastic Gradients
 - Present a defense method that is randomized, causing single samples to incorrectly estimate the true gradients.
 - e.g., Stochastic Activation Pruning

[Athalye et al, 2018](#)

Backward Pass Differentiable Approximation (BPDA)

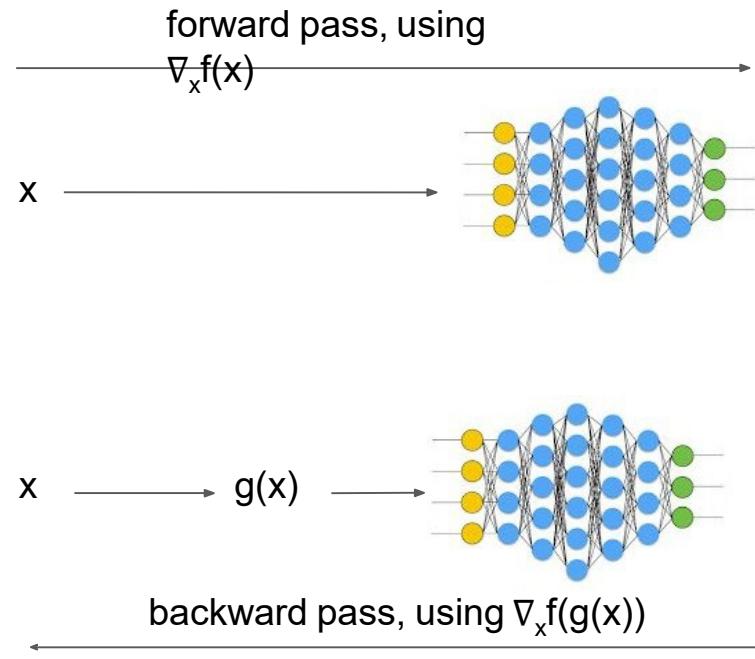
- Bypass Shattered Gradients by its differentiable approximations.



$$\nabla_x f(g(x))|_{x=\hat{x}} \approx \nabla_x f(x)|_{x=g(\hat{x})}$$

[Athalye et al, 2018](#)

BPDA In Neural Networks



[Athalye et al, 2018](#)

Handling Stochastic Gradients

- Applying the expectations of multiple Stochastic Gradients

$$\nabla \mathbb{E}_{t \sim T} f(t(x)) = \mathbb{E}_{t \sim T} \nabla f(t(x))$$

Results

Defense	Dataset	Distance	Accuracy on Adversarial Samples
Adversarial Training (Madry et al, 2018)	CIFAR	0.031(ℓ_∞)	47%
Input Transformations (Guo et al, 2018)	ImageNet	0.005(ℓ_2)	0%
Stochastic Gradients (Dhillon et al, 2018)	CIFAR	0.031 (ℓ_∞)	0%

[Athalye et al, 2018](#)

But Why is Adversarial Training More Robust?

Robust Optimization

- Train a robust model
 - In the neighborhood of x
 - Under the worst case scenario in terms of the loss function

$$\min_{\theta} \tilde{J}(\theta, x, y) = \min_{\theta} \sum_{i=1}^m \max_{\tilde{x}_i \in \mathcal{U}_i} J(\theta, \tilde{x}_i, y_i)$$

↑
uncertainty sets ↓
loss function

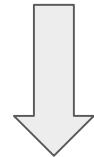


[Shaham et al, 2016](#)

Linear Regression As A Robust Optimization

- We can write Linear Regression in the form of Robust Optimization

$$\min_x \|Ax - b\| + \lambda \|x\|_1$$



$$\min_x \max_{\|\Delta A\|_{\infty,2} \leq \rho} \|(A + \Delta A)x - b\|$$

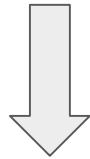
Robust Optimization

[Shaham et al, 2016](#)

Adversarial Training As A Robust Optimization

- We can also write Adversarial Training in the form of Robust Optimization

$$\tilde{J}(\theta, x, y) = \alpha J(\theta, x, y) + (1 - \alpha) J(\theta, x^{adv}, y)$$



$$\min_{\theta} \tilde{J}(\theta, x, y) = \min_{\theta} \sum_{i=1}^m \max_{\tilde{x}_i \in \mathcal{U}_i} J(\theta, \tilde{x}_i, y_i)$$

$$\Delta_{x_i} = \arg \max_{\Delta: x_i + \Delta \in \mathcal{U}_i} J_{\theta, y_i}(x_i + \Delta)$$

[Shaham et al, 2016](#)

Certified Defense

- Guarantee the performance against Adversarial Attack
- Guaranteed for a family of networks

$$f^i(x) = V_i^\top \sigma(Wx)$$

Two-layer Neural Network

[Raghunathan et al, 2018](#)

Bounded Performance

Error Margin $f(x) = f^1(x) - f^2(x)$

incorrect class correct class

$$f(A(x)) \leq f(A_{\text{opt}}(x)) \leq f(x) + \epsilon \max_{\tilde{x} \in B_\epsilon(x)} \|\nabla f(\tilde{x})\|_1 \leq f_{\text{QP}}(x) \leq f_{\text{SDP}}(x)$$

Error of any attack

Error of optimal attack

Bounds

Computationally
Feasible Bounds

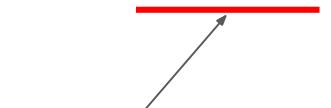
Bounded Performance

$$\text{Error Margin} \quad f(x) = \begin{cases} f^1(x) - f^2(x) & \text{incorrect class} \\ & \text{correct class} \end{cases}$$

$$f(A(x)) \leq f(A_{\text{opt}}(x)) \leq f(x) + \epsilon \max_{\tilde{x} \in B_\epsilon(x)} \|\nabla f(\tilde{x})\|_1 \leq f_{\text{QP}}(x) \leq f_{\text{SDP}}(x)$$

$$f_{\text{SDP}}(x) \stackrel{\text{def}}{=} f(x) + \frac{\epsilon}{4} \max_{\substack{P \succeq 0, \text{diag}(P) \leq 1 \\ \text{solution to semidefinite program}}} \langle M(v, W), P \rangle$$

$$M(v, W) \stackrel{\text{def}}{=} \begin{bmatrix} 0 & 0 & \mathbf{1}^\top W^\top \text{diag}(v) \\ 0 & 0 & W^\top \text{diag}(v) \\ \text{diag}(v)^\top W \mathbf{1} & \text{diag}(v)^\top W & 0 \end{bmatrix} \quad v \stackrel{\text{def}}{=} V_1 - V_2$$



Upper Bound
(SDP)

Training Certified Defense

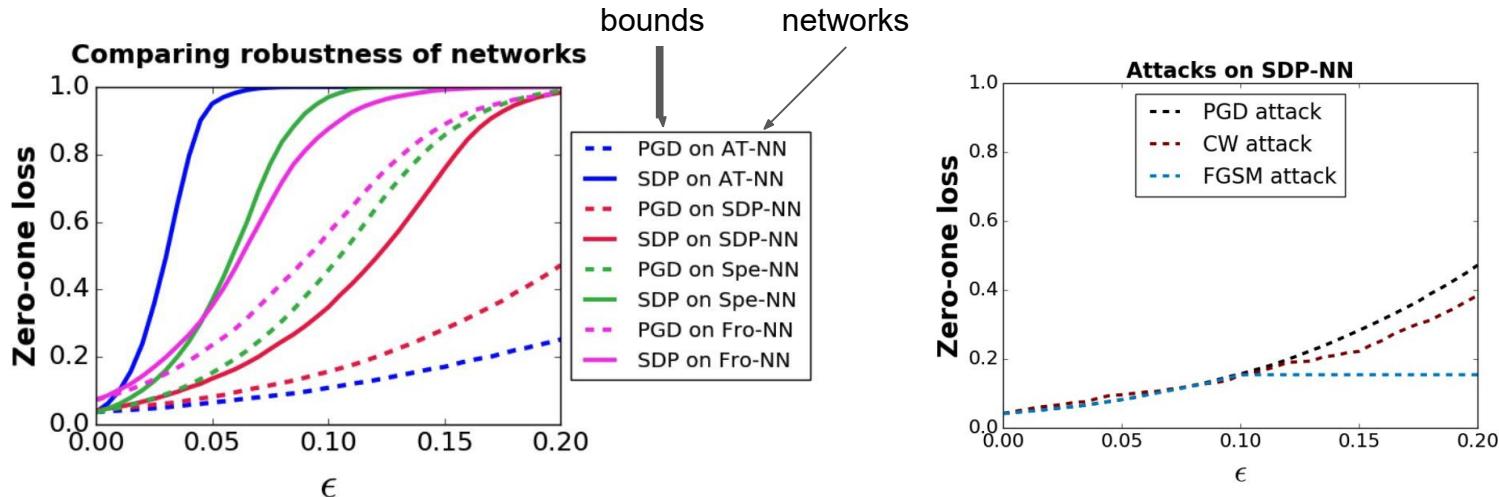
$$\begin{aligned} f(A(x)) \leq f(A_{\text{opt}}(x)) &\leq f(x) + \epsilon \max_{\tilde{x} \in B_\epsilon(x)} \|\nabla f(\tilde{x})\|_1 \leq f_{\text{QP}}(x) \leq \underline{f_{\text{SDP}}(x)} \\ f_{\text{SDP}}(x) &\stackrel{\text{def}}{=} f(x) + \frac{\epsilon}{4} \max_{P \succeq 0, \text{diag}(P) \leq 1} \langle M(v, W), P \rangle \end{aligned}$$

$$(W^*, V^*) = \arg \min_{W, V} \sum_n \ell_{\text{cls}}(V, W; x_n, y_n) + \sum_{i \neq j} \lambda^{ij} \max_{P \succeq 0, \text{diag}(P) \leq 1} \langle M^{ij}(V, W), P \rangle$$



parameters to the
two-layer neural network loss function hyper-parameter Defense Certification

Results



AT-NN - Adversarial training using PGD ([Madry et al, 2018](#))

SDP-NN - Proposed training objective

Spe-NN - Spectral norm regularization i.e., $\lambda(\|W\|_2 + \|v\|_2)$

Fro-NN - Frobenius norm regularization i.e., $\lambda(\|W\|_F + \|v\|_2)$

$$\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}$$

PGD - lower bound
SDP - upper bound

$$\frac{f(A(x)) \leq f(A_{\text{opt}}(x))}{\text{PGD lower bound}} \leq f(x) + \epsilon \max_{\tilde{x} \in B_\epsilon(x)} \|\nabla f(\tilde{x})\|_1 \leq \frac{f_{\text{QP}}(x)}{\text{SDP lower bound}} \leq f_{\text{SDP}}(x)$$

[Raghunathan et al, 2018](#)

Results

- No attack that perturbs each pixel by at most $\epsilon = 0.1$ can cause more than 35% test error.

Network	PGD error	SDP bound
SDP-NN	15%	35%

SDP-NN - Proposed training objective

PGD - upper bound

SDP - lower bound

$\epsilon = 0.1$

Summary

- Robustness of ML Models
 - Preventing models from being abused by malicious attack
- Adversarial Attack
 - Confuses models by manipulating input data
 - Evasion attack
 - Poisoning attack
 - Exploratory attack
- Attack Strategies
 - FGSM - white-box
 - C&W -white-box
 - Jacobian-based Data Augmentation - black-box

Summary

- Adversarial Defense
 - Equip models with the ability to defend adversarial attacks
- Defense Strategies
 - Adversarial Training
 - Robust Optimization
 - Gradient Shattering
 - Stochastic Gradients
- BPDA
 - Attack all defense models utilizing Obfuscated Gradients
- Certified Defense
 - Provable performance for certain types of networks

Compre Tips

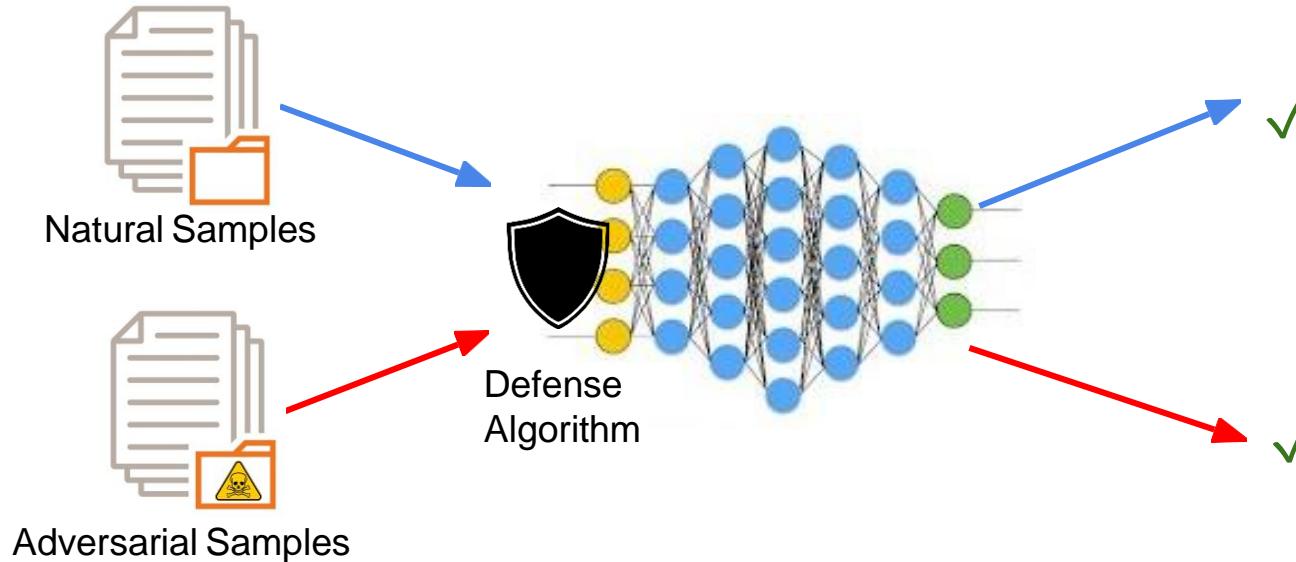
- Focus on both pre-midsem and post-midsem quantitative topics
- Among post midsem topics, focus on LIME, Shapley/SHAP and gradient based techniques for visualization/robustness/privacy
- Review problems discussed in class / midsem solutions
- Expect to use basic concepts from ML/DNN courses (e.g., loss function, weighted regression, minimization of loss functions, etc.)

Adversarial Defense

Outline

- Adversarial Defense
- Defense Strategies
 - Adversarial Training
 - Input Transformations
 - Stochastic Gradients
- Obfuscated Gradients and BPDA

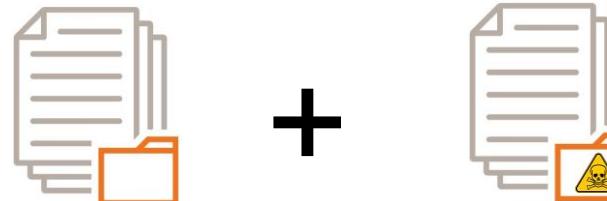
Adversarial Defense



Summary of Defense Strategies

Defense Methods	General Idea
Adversarial Training	Mixing adversarial samples with natural samples during training
Input Transformation	Adding transformation to make defense non-differentiable
Stochastic Gradients	Causing gradients to be randomized

Adversarial Training



Natural Samples

Adversarial Samples

+

$$\tilde{J}(\theta, x, y) = \alpha J(\theta, x, y) + (1 - \alpha) J(\theta, x^{adv}, y)$$

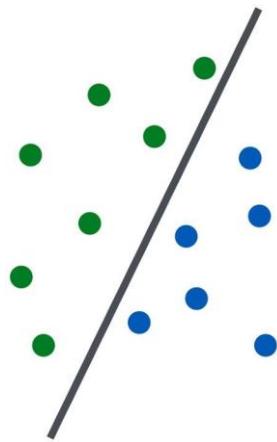
Loss Function

Natural Samples

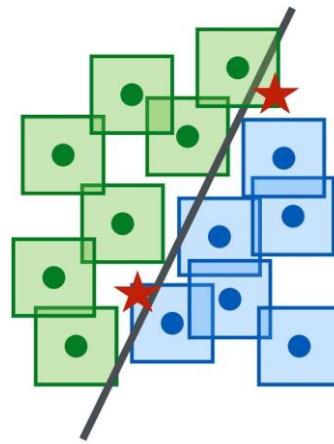
Adversarial Samples

[Goodfellow et al, 2014](#)

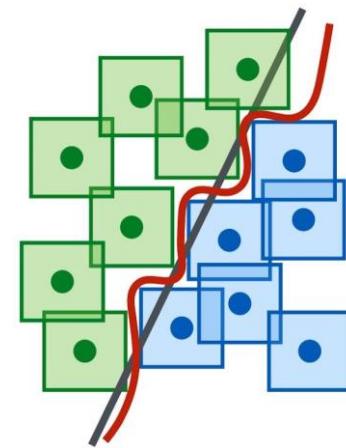
Adversarial Training



Natural Samples



Natural Samples with L_∞
Perturbation Space



Adversarial Training

[Madry et al, 2017](#)

Results on FGSM

- Accuracy on Adversarial Examples

$$\mathbf{X}^{adv} = \mathbf{X} + \epsilon \operatorname{sign}(\nabla_{\mathbf{X}} J(\mathbf{X}, y_{true}))$$

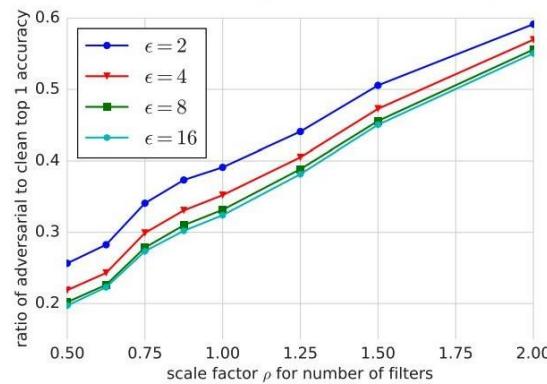
		Clean	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 8$	$\epsilon = 16$
Baseline (standard training)	top 1	78.4%	30.8%	27.2%	27.2%	29.5%
	top 5	94.0%	60.0%	55.6%	55.1%	57.2%
Adv. training	top 1	77.6%	73.5%	74.0%	74.5%	73.9%
	top 5	93.8%	91.7%	91.9%	92.0%	91.4%
Deeper model (standard training)	top 1	78.7%	33.5%	30.0%	30.0%	31.6%
	top 5	94.4%	63.3%	58.9%	58.1%	59.5%
Deeper model (Adv. training)	top 1	78.1%	75.4%	75.7%	75.6%	74.4%
	top 5	94.1%	92.6%	92.7%	92.5%	91.6%

Dataset: ImageNet

[Kurakin et al, 2017](#)

Results on FGSM

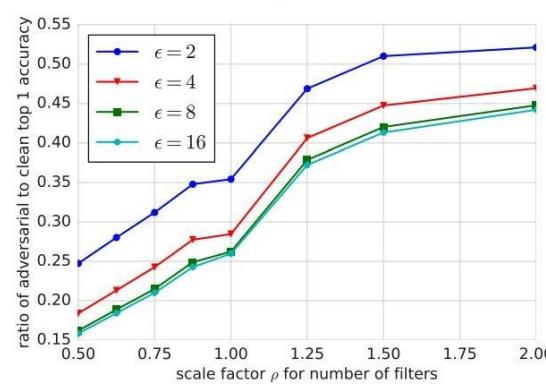
- Adversarial Accuracy / Clean Image Accuracy
 - Ratio $\rightarrow 1$ successful adversarial attack
 - Ratio $\rightarrow 0$ successful adversarial defense



No adversarial training, “basic iter.” adv. examples

fast - FGSM

basic iter. - iterative untargeted FGSM



With adversarial training, “basic iter.” adv. examples

[Kurakin et al, 2017](#)

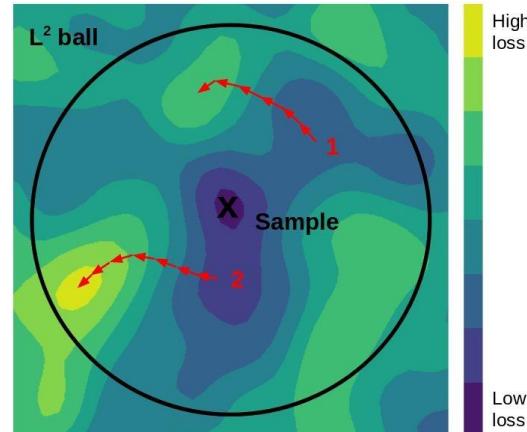
Flexibility

- Plug-in any attack techniques

$$\tilde{J}(\theta, x, y) = \alpha J(\theta, x, y) + (1 - \alpha) J(\theta, x^{adv}, y)$$

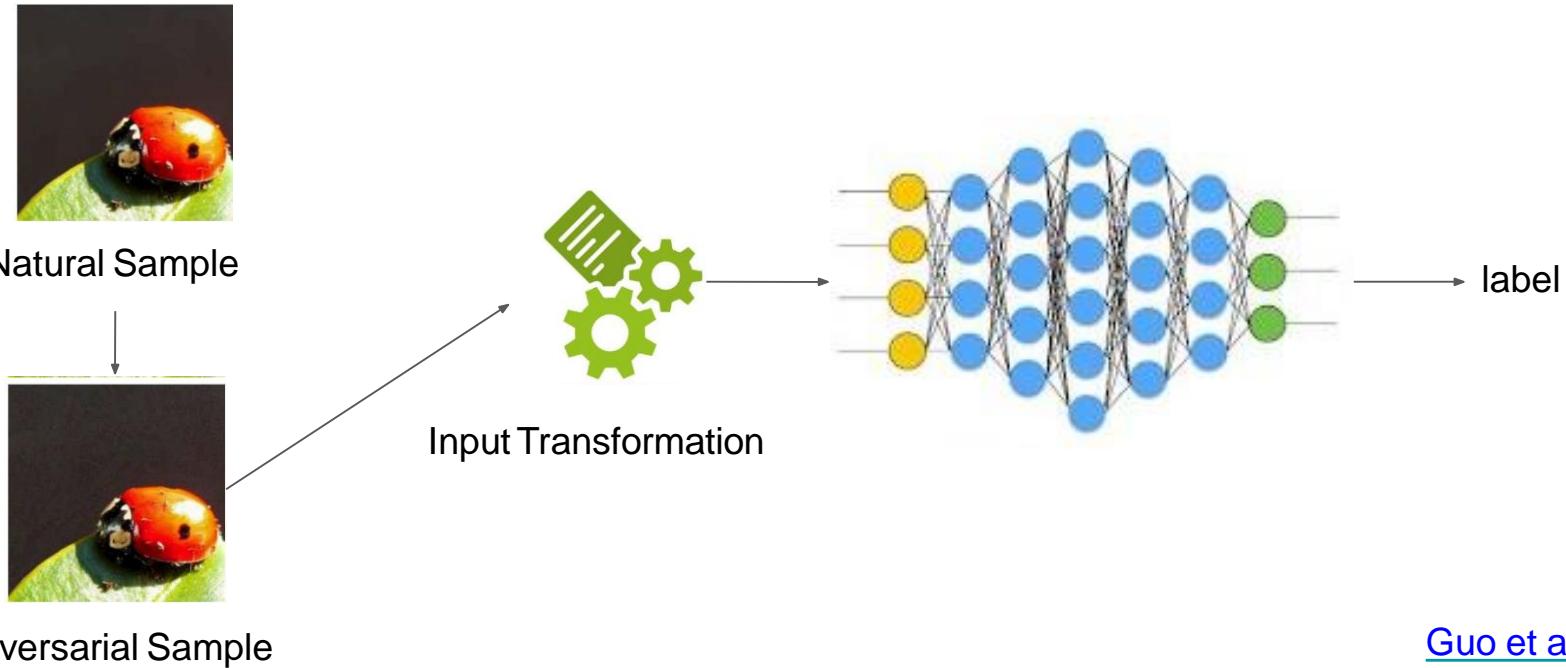
- Examples
 - FGSM
 - Projected Gradient Descent (PGD) ([Madry et al, 2017](#))

$$\max_{\mathbf{x}' : \|\mathbf{x}' - \mathbf{x}\|_\infty < \alpha} \mathcal{L}(\mathbf{x}', y; \theta)$$



Input Transformations

https://wiki.math.uwaterloo.ca/statwiki/index.php?title=Counteracting_Adversarial_Images_Using_Input_Transformations



[Guo et al, 2018](#)

Input Transformations

- Goal: Disrupt Adversarial Perturbations
- Image cropping/re-scaling
- Bit-depth reduction



16.7 Million
Colors

256
Colors

16
Colors

[Guo et al, 2018](#)

Input Transformations

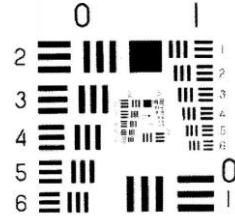
- Goal: Disrupt Adversarial Perturbations
- Image cropping/re-scaling
- Bit-depth reduction
- JPEG compression
- Total variation minimization
- Image quilting



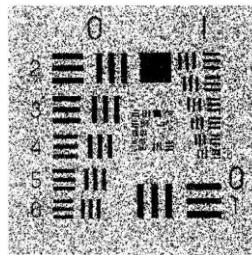
[Guo et al, 2018](#)

Total Variation Minimization

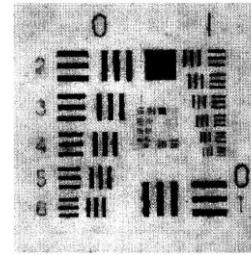
- Generate a denoised image z by minimizing TV



Original Image



Noisy Image



Denoised Image minimizing TV

$$\text{TV}_p(\mathbf{z}) = \sum_{k=1}^K \left[\sum_{i=2}^N \|\mathbf{z}(i, :, k) - \mathbf{z}(i-1, :, k)\|_p + \sum_{j=2}^N \|\mathbf{z}(:, j, k) - \mathbf{z}(:, j-1, k)\|_p \right]$$

Transformed Image

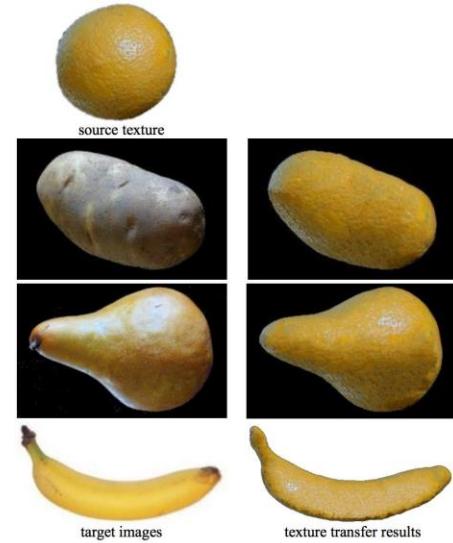
row variance

column variance

[Rudin et al, 1992](#)

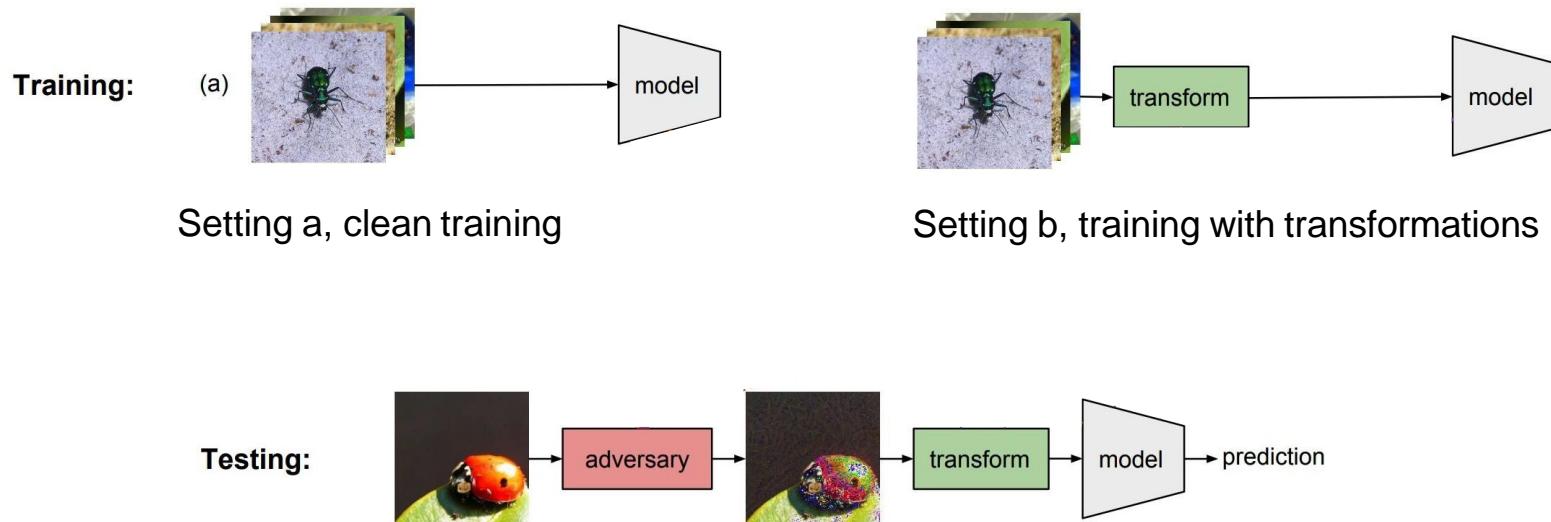
Image Quilting

- Synthesizes images by piecing together small patches taken from a database of image patches
- Database contains only clean images



[Efros et al, 2001](#)

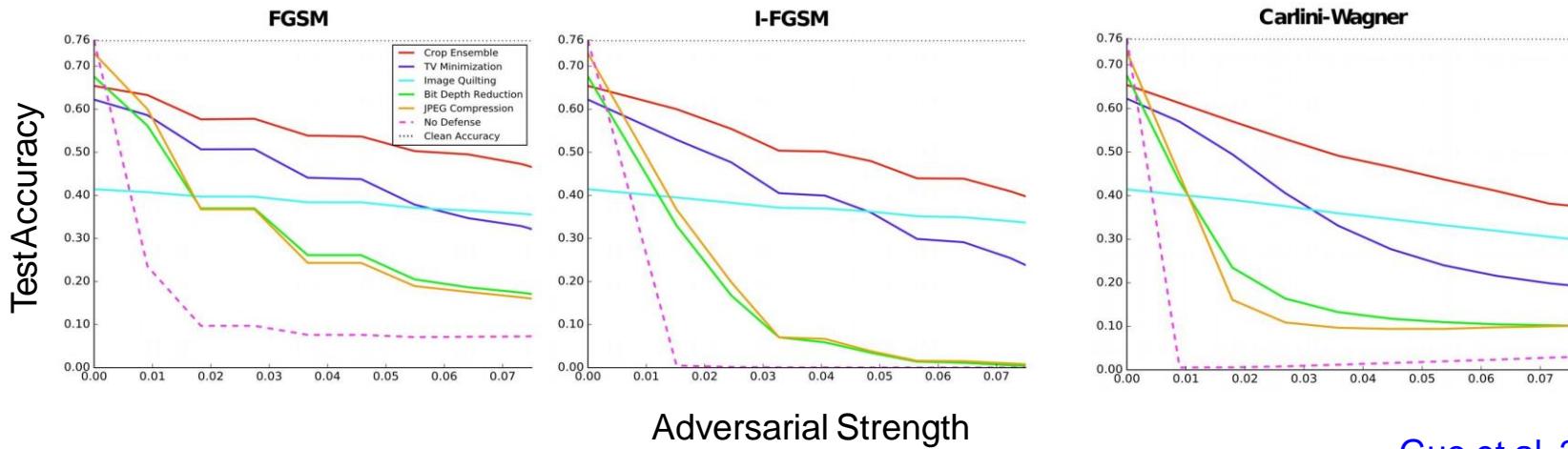
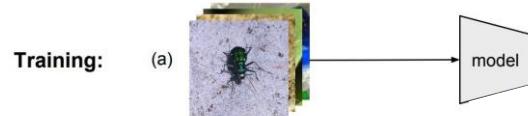
Input Transformation Defense



[Guo et al, 2018](#)

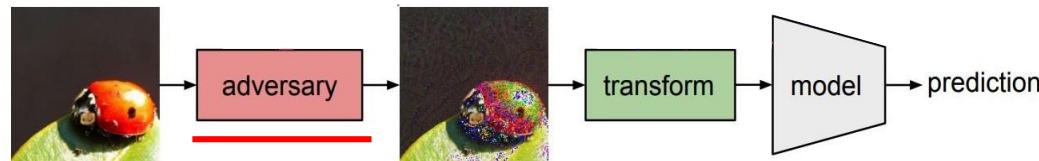
Results with Clean Image Training

ResNet on ImageNet



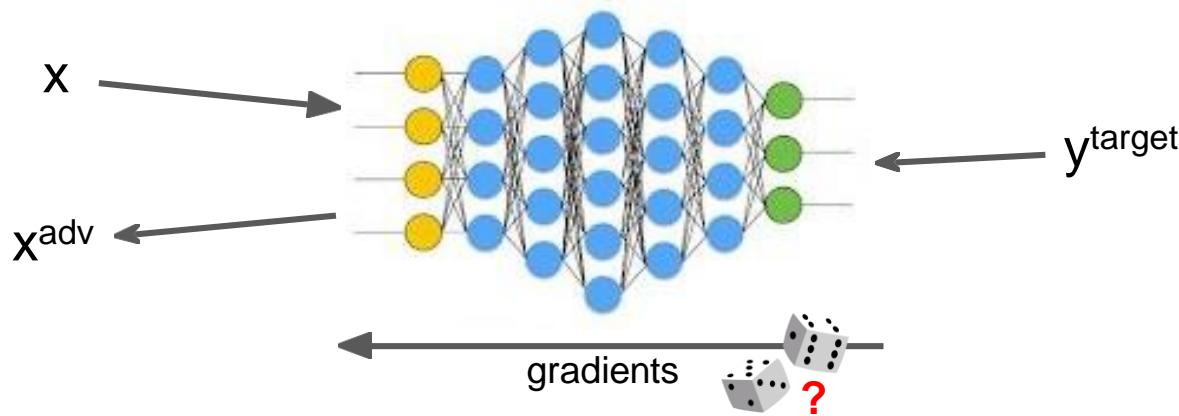
Gradient Shattering

- Can we design specialized attacks that target input transformations?
 - We show previously the results using FGSM and C&W
- Input Transformations belongs to a family of defense methods that causes Gradient Shattering



Train our own adversary that targets input transformations?

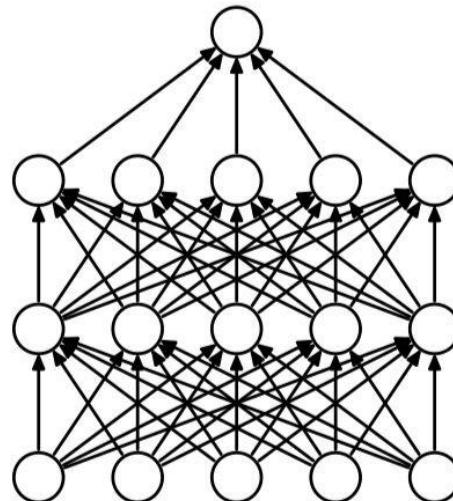
Stochastic Gradients



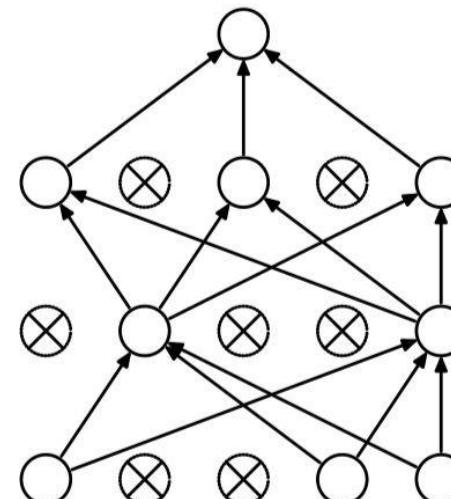
$$\mathbf{X}_{N+1}^{adv} = Clip_{X,\epsilon} \left\{ \mathbf{X}_N^{adv} + \alpha \operatorname{sign}(\nabla_X J(\mathbf{X}_N^{adv}, y_{true})) \right\}$$

Dropout

- Dropout randomly turns off activations by a fixed probability r
- Originally introduced to prevent overfitting



(a) Standard Neural Net



(b) After applying dropout.

Stochastic Activation Pruning (SAP)

- Stochastic Activation Pruning turns off activations based on a learned probability
- Draw with replacement for each activation

$$\underline{p_j^i} = \frac{|(h^i)_j|}{\sum_{k=1}^{a^i} |(h^i)_k|}$$

probability of turning on
the j^{th} activation on the i^{th} layer

embeddings of
the j^{th} activation on the k^{th} layer

[Dhillon et al, 2018](#)

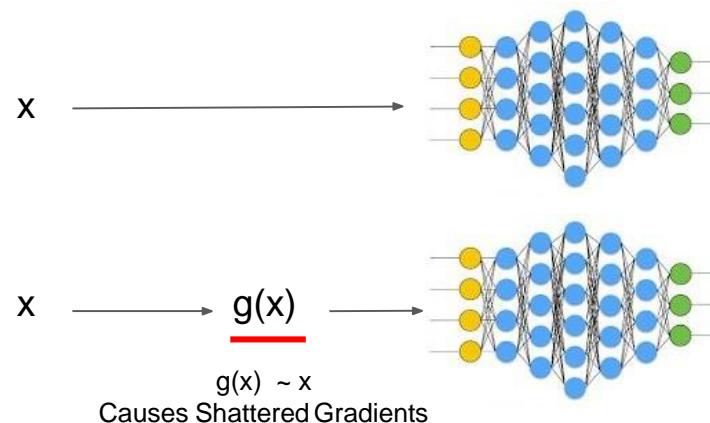
Obfuscated Gradients

- A defense method is said to achieve Obfuscated Gradients if
 - It prevents the attack methods from utilizing useful gradient information
- Shattered Gradients
 - Present a defense method that is non-differentiable or numerically unstable
 - e.g., Input Transformations
- Stochastic Gradients
 - Present a defense method that is randomized, causing single samples to incorrectly estimate the true gradients.
 - e.g., Stochastic Activation Pruning

[Athalye et al, 2018](#)

Backward Pass Differentiable Approximation (BPDA)

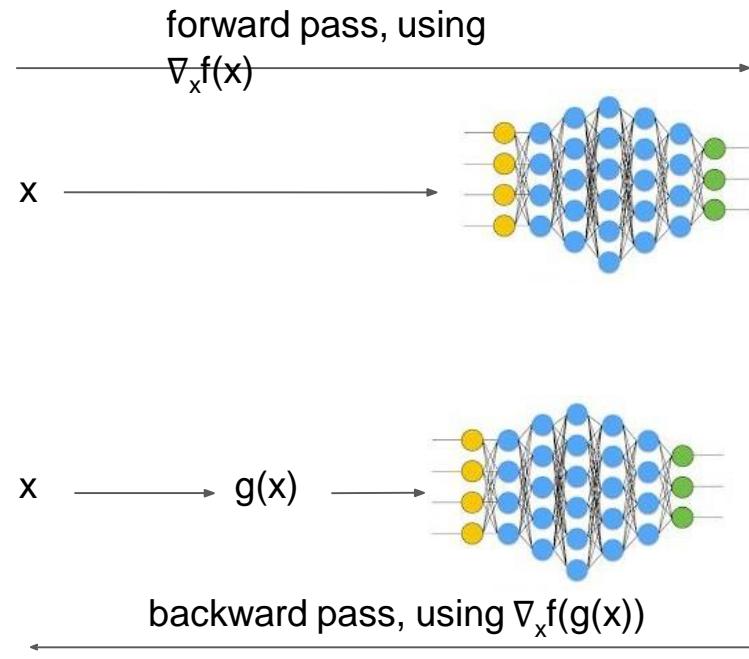
- Bypass Shattered Gradients by its differentiable approximations.



$$\nabla_x f(g(x))|_{x=\hat{x}} \approx \nabla_x f(x)|_{x=g(\hat{x})}$$

[Athalye et al, 2018](#)

BPDA In Neural Networks



[Athalye et al, 2018](#)

Handling Stochastic Gradients

- Applying the expectations of multiple Stochastic Gradients

$$\nabla \mathbb{E}_{t \sim T} f(t(x)) = \mathbb{E}_{t \sim T} \nabla f(t(x))$$

Summary

- Robustness of ML Models
 - Preventing models from being abused by malicious attack
- Adversarial Attack
 - Confuses models by manipulating input data
 - Evasion attack
 - Poisoning attack
 - Exploratory attack
- Attack Strategies
 - FGSM - white-box
 - C&W -white-box
 - Jacobian-based Data Augmentation - black-box

Summary

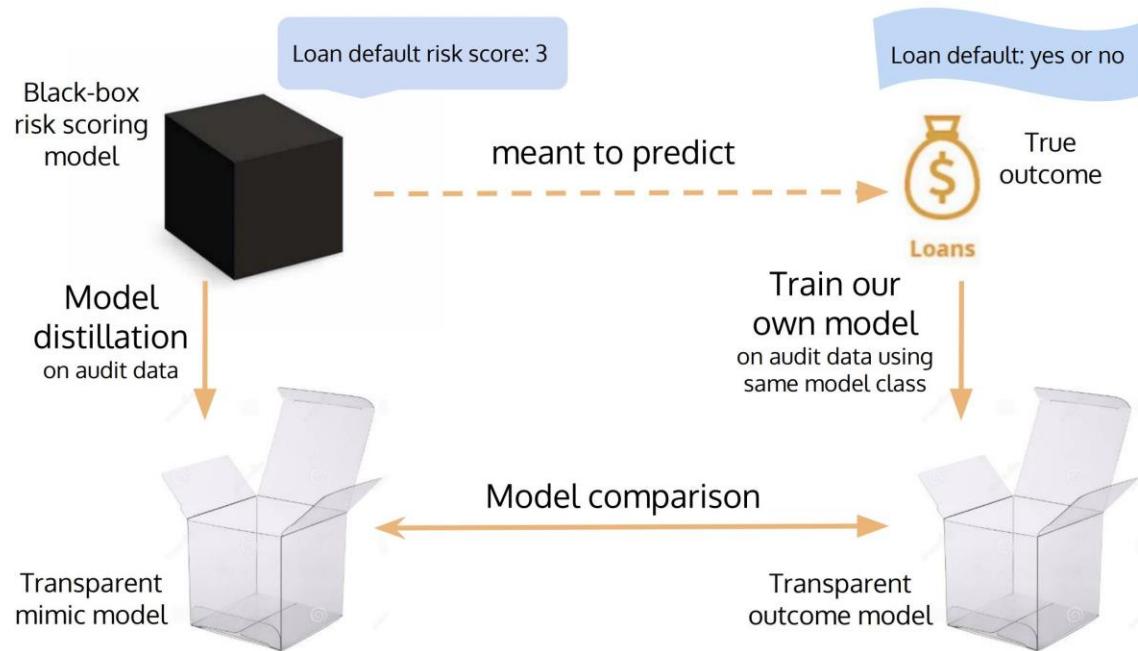
- Adversarial Defense
 - Equip models with the ability to defend adversarial attacks
- Defense Strategies
 - Adversarial Training
 - Gradient Shattering
 - Stochastic Gradients
- BPDA
 - Attack all defense models utilizing Obfuscated Gradients

Auditing and ML Privacy

Outline

- ML Auditing
 - Distill-and-Compare
- Privacy in ML
 - Differential Privacy with Deep Learning
 - Model Inversion Attack and Differential Privacy
 - Local Differential Privacy
 - Federated Learning

ML Auditing Using Model Distillation



[Tan et al, 2018](#)

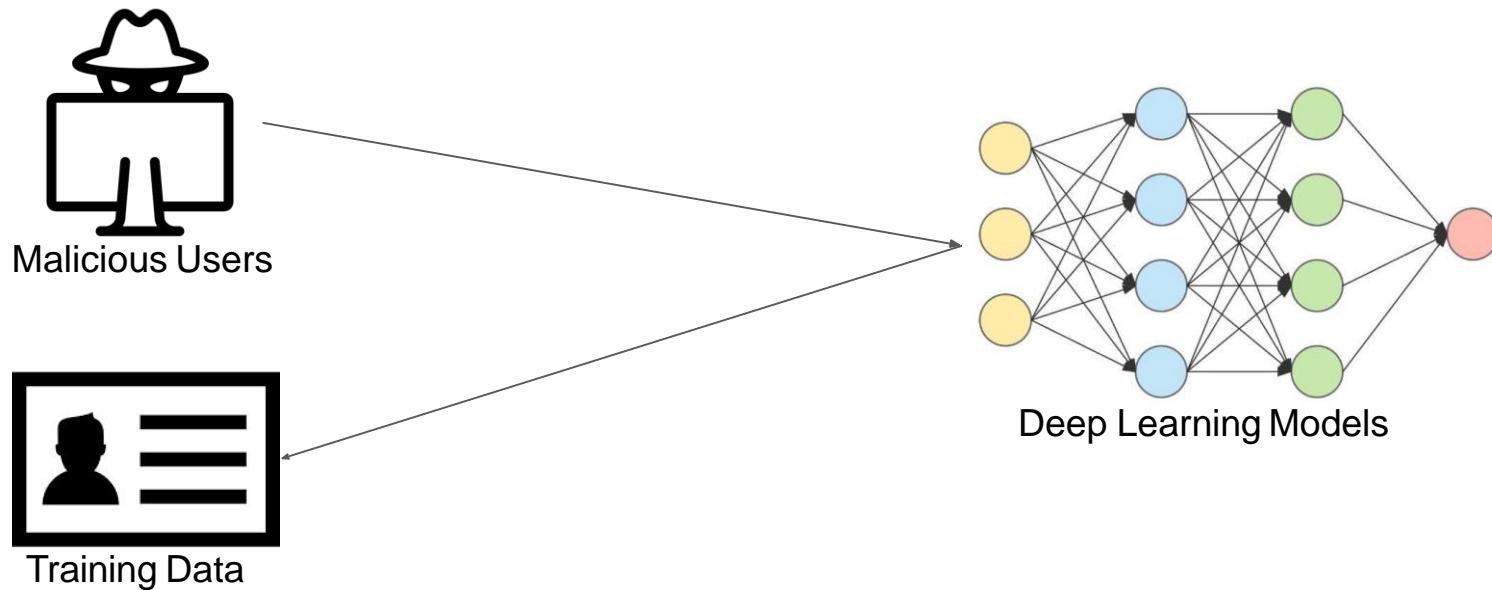
General Additive Model

$$g(y) = h_0 + \sum_i h_i(x_i) + \sum_{i \neq j} h_{ij}(x_i, x_j)$$

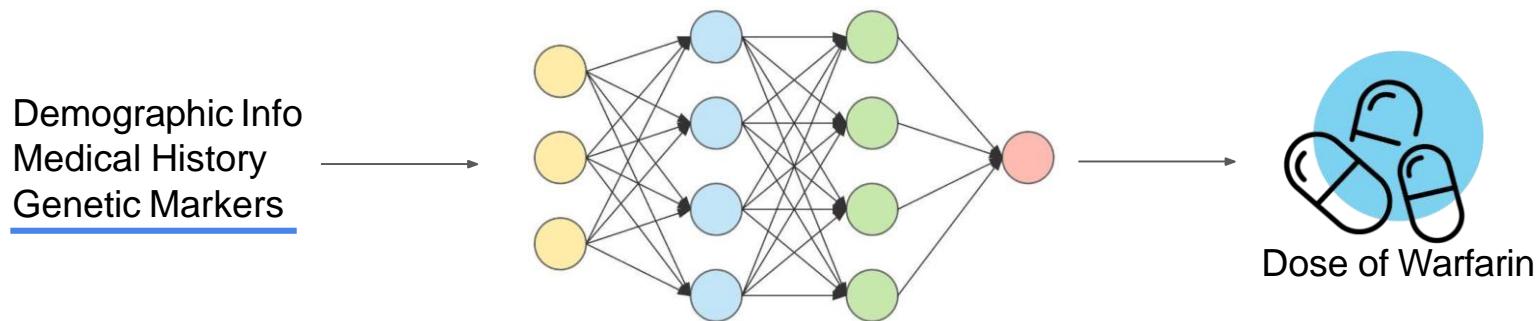
transformation
e.g., logistic for classification

weights

Privacy in ML



Inferring Sensitive Features from ML Models



[Fredrikson et al, 2014](#)

Inferring Training Data from Facial Recognition Models



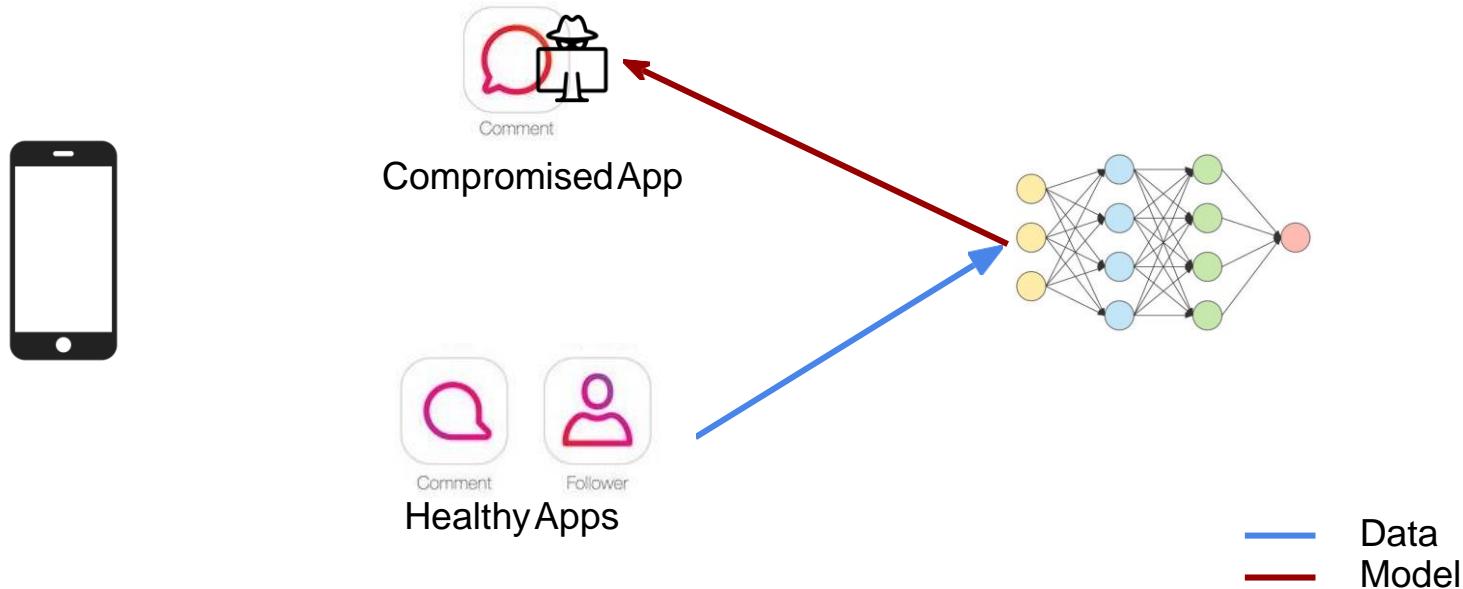
Original Image



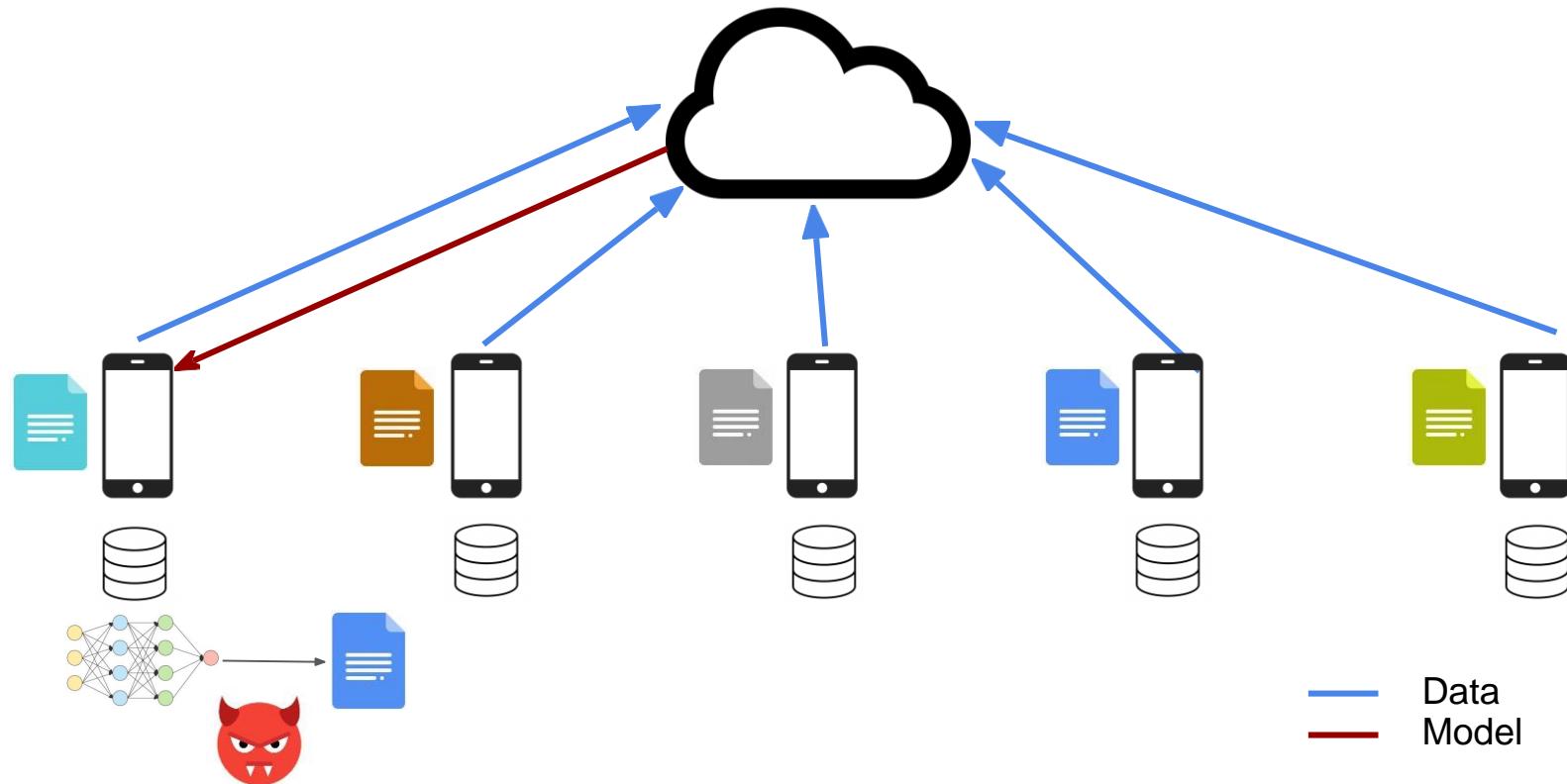
Inferred Image

[Fredrikson et al, 2015](#)

Centralized Setting



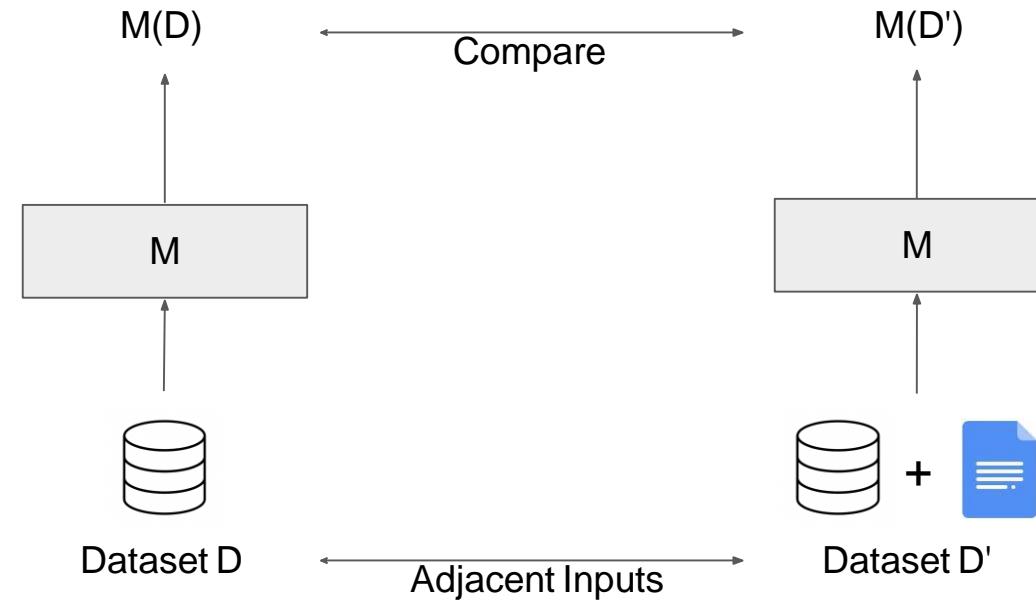
Distributed Setting



Outline

- ML Auditing
 - Distill-and-Compare
- Privacy in ML
 - Differential Privacy with Deep Learning
 - Model Inversion Attack and Differential Privacy
 - Local Differential Privacy
 - Federated Learning

Differential Privacy



Differentially Private SGD

Gradient Norm Bounds

C

Step 1 Calculate Gradients

$$\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t} \mathcal{L}(\theta_t, x_i)$$

Step 2 Gradient Clipping

$$\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max \left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C} \right)$$

Step 3 Adding Noise

$$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L} \left(\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I}) \right)$$

Step 4 Parameter Updating

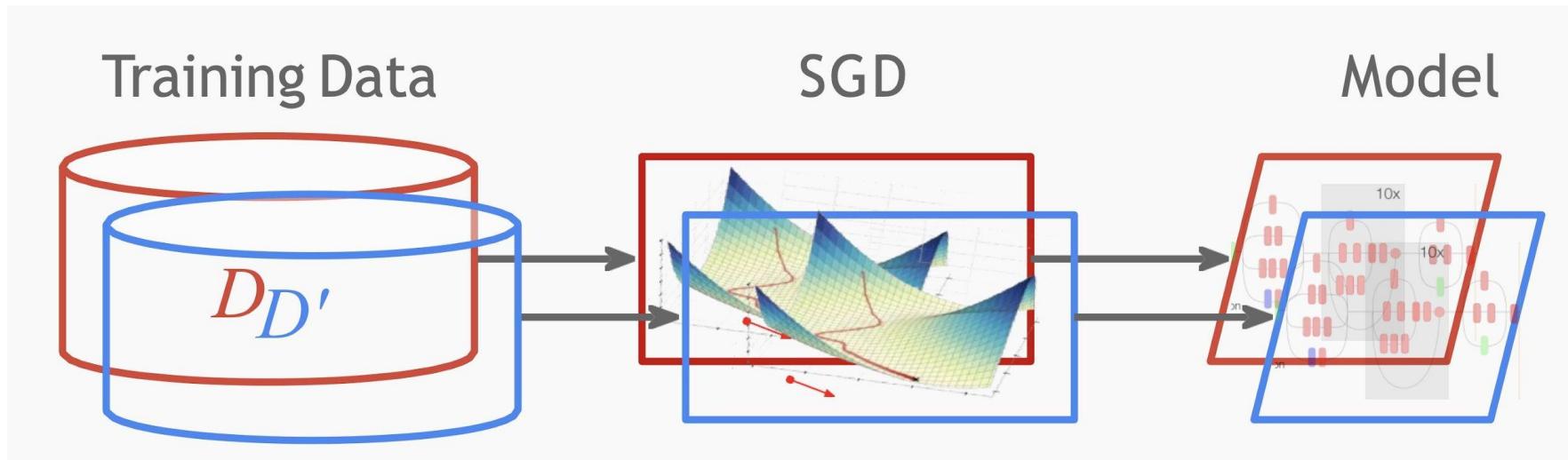
$$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$$

One noise added to each **lot**
(group of data)

$$\mathcal{M}(d) \triangleq f(d) + \mathcal{N}(0, \underline{S_f^2} \cdot \underline{\sigma^2})$$

[Abadi et al, 2016](#)

Differentially Private SGD



[Abadi et al, 2016](#)

Outline

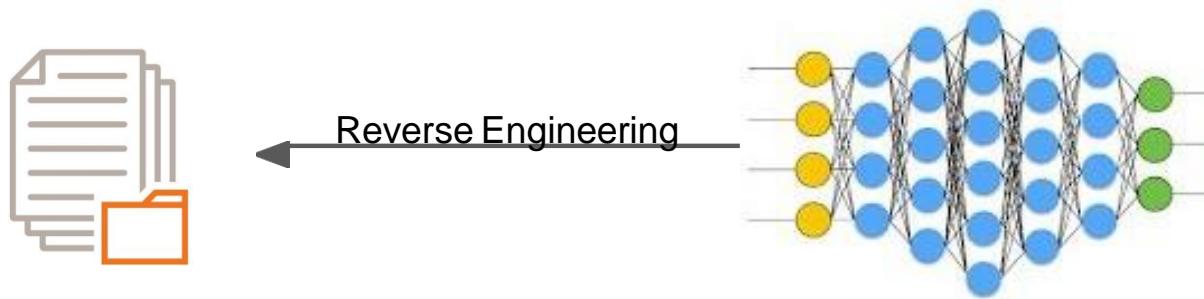
- ML Auditing
 - Distill-and-Compare
- Privacy in ML
 - Differential Privacy with Deep Learning
 - Model Inversion Attack and Differential Privacy
 - Local Differential Privacy
 - Federated Learning

Recap: Types of Adversarial Attack

	Attack Phase	Goal
Evasion	Testing	Compromise Model Performance
Data Poisoning	Training	Compromise Model Performance
Exploratory	Testing	Explore Model Characteristics Reconstruct User Data

Recap

- Exploratory Attack
 - Reverse engineer user data from a trained model



Model Inversion Attacks



Original Image

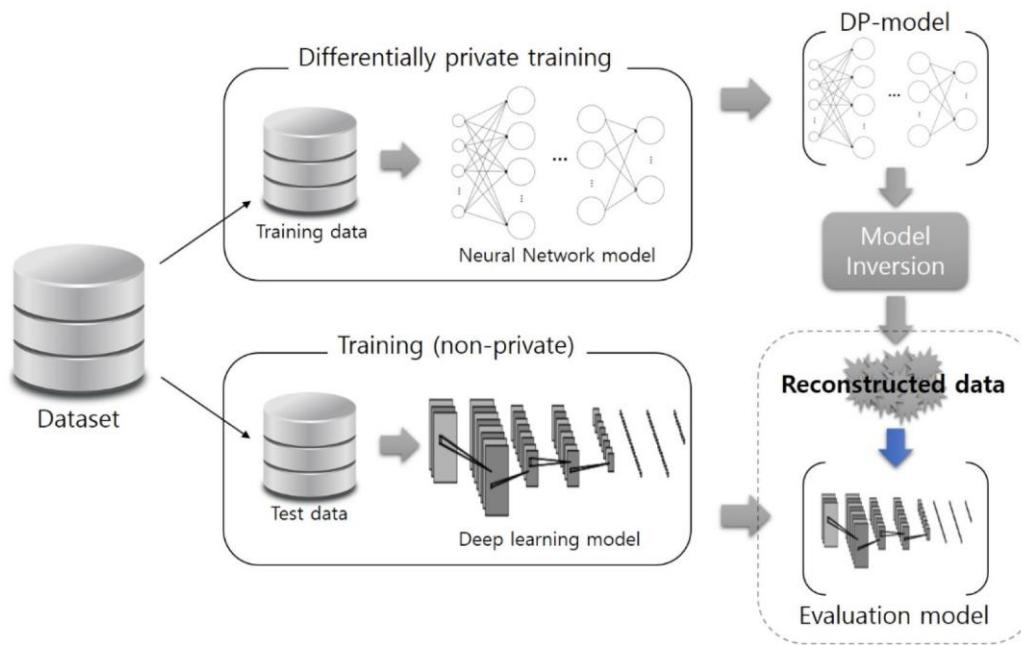


Reconstructed Image

$$x = \arg \max_x f_y(x)$$

[Fredrikson et al, 2015](#)

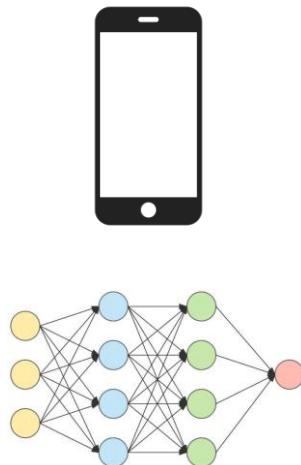
Model Inversion Attack to Evaluate Differential Privacy



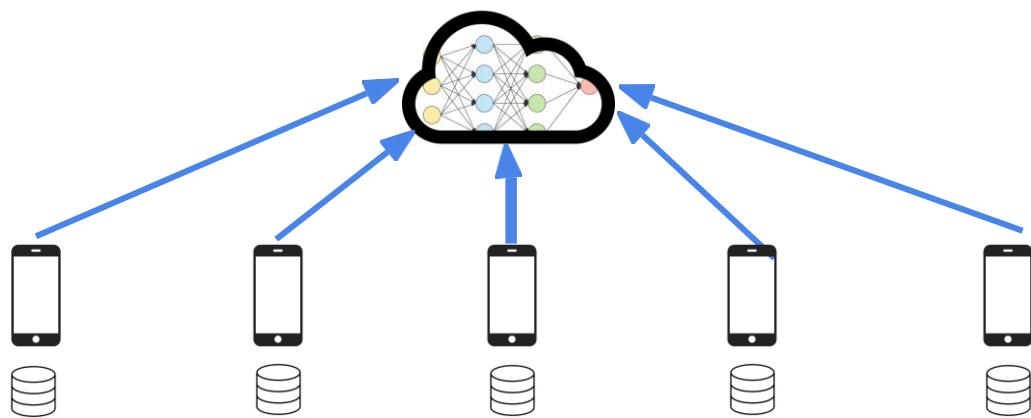
[Park et al, 2019](#)

Distributed Optimization

Centralized Setting



Distributed Setting

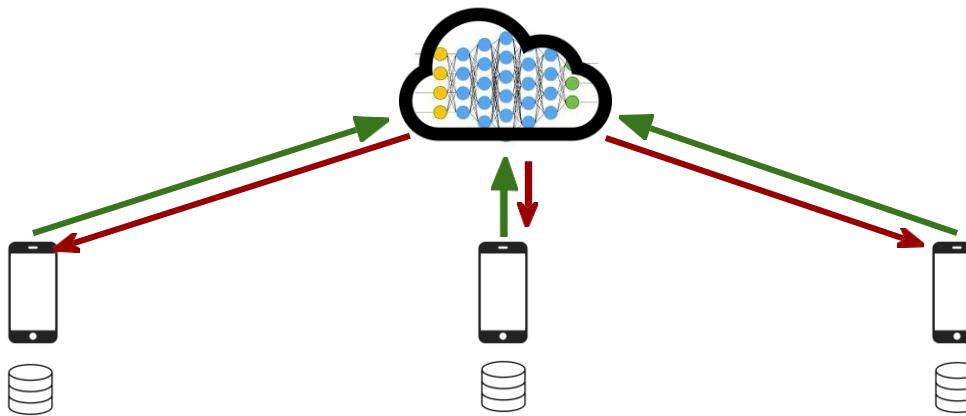


Relies on distributed optimization

Federated Optimization

- Non-IID
 - User data is localized to their own usage
 - Hard to be a representative of the population
- Unbalanced Similarly
 - Some users will make much heavier on particular services than others
- Distributed Computing Capacity
 - Expect a large number of devices to be updated at the same time
- Limited communication
 - Mobile devices are frequently offline or on slow or expensive connections

FedSGD

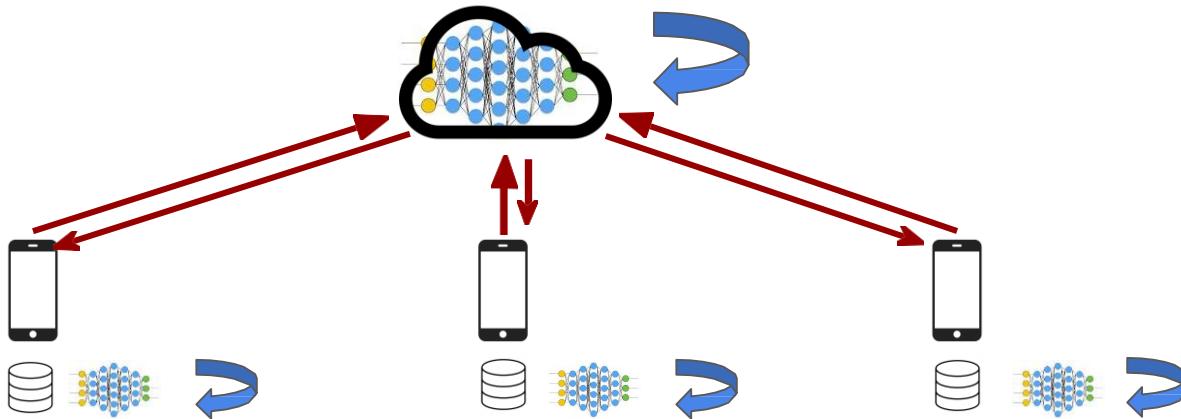


— Gradient
Model

$$w_{t+1} \leftarrow w_t - \eta \sum_{k=1}^K \frac{n_k}{n} g_k$$
$$g_k = \nabla F_k(w_t)$$

[McMahan et al, 2017](#)

FedAvg



— Gradient
Model

$$w_{t+1}^k \leftarrow w_t - \eta g_k$$
$$w_{t+1} \leftarrow \sum_{k=1}^K \frac{n_k}{n} w_{t+1}^k$$

[McMahan et al, 2017](#)