ESSCaSS 2022

Fair and Robust Machine Learning – Part 1

Christoph Lampert

Institute of Science and Technology Austria

Institute of Science and Technology Austria (ISTA)

About ISTA:

- public research institute, located near Vienna
- curiosity-driven basic research
- focus on interdisciplinarity
  - Computer Science, Mathematics, Biology, Physics, Chemistry, Earth&Climate Sciences, Neuroscience
- ELLIS Unit since 2019

PhD-Granting Graduate School

- US Style (1+3 years) graduate program
- fully funded positions

We're hiring! (on all levels)

- interns, PhD students, postdocs, faculty (tenure-track or tenured), ...

More information: `https://mlcv.ist.ac.at` or `chl@ist.ac.at`

Schedule

Lecture 1: Intro to Machine Learning

Lecture 1: Robust Machine Learning

Lecture 1: Fair Machine Learning

Lecture 2: Certified Robustness via Lipschitz Networks

Lecture 2: Robust and Fair Learning from Multiple Sources

Lecture 3: Behind the Scenes of (Machine Learning) Research

Machine Learning
Artificial Intelligence

HOME > TECHNOLOGY EXPLAINED

What Is the Singularity and When Will We Reach It?

When AI becomes sentient, what will happen?

ARTIFICIAL INTELLIGENCE

Google Engineer Claims AI Chatbot Is Sentient: Why That Matters

Is it possible for an artificial intelligence to be sentient?

'I'm sorry, Dave. I'm afraid I can't do that': Artificial Intelligence expert warns that there may already be a 'slightly conscious' AI out in the world

NEUROSCIENCE & MIND

Robert J. Marks: Could Artificial Intelligence Replace Tom Cruise?

We Aren't Sure If (Or When) Artificial Intelligence Will Surpass the Human Mind

Experts say the future of AI is uncertain, but it wouldn't hurt to prepare for the possibility of singularity.

Machine Learning (Artificial Intelligence) is a way to develop software.

Example Task: Sorting

Classic Software Development

1) formalize the problem
   - function: $\texttt{sort} : \mathcal{X} \to \mathcal{Y}$
   - input set $\mathcal{X}$: array of numbers
   - output set $\mathcal{Y}$: array of numbers
   - specification:
     - $y = \texttt{sort}(x)$ is a permutation of x
     - y is sorted, i.e. $\forall i, j \in [|x|] : i \leq j \Rightarrow y_i \leq y_j$
2) developer comes up with an algorithm
3) prove formally that it solves the task
4) implement the algorithm
5) check that it works correctly using test cases: some random, some extremes.

Example Task: Recognize Voice Commands

Classic Software Development

1) formalize the problem

- function: $\texttt{recognize} : \mathcal{X} \to \mathcal{Y}$
- input set $\mathcal{X}$: audio signal
- output set $\mathcal{Y}$: possible commands, e.g. $\{\texttt{start}, \texttt{stop}\}$
- specification: ???

2) developer fails to come up with an algorithm

Classic software development fails for tasks that we cannot formally describe.

Example Task: Recognize Voice Commands

Machine Learning

1) formalize the problem

- function: $\texttt{recognize} : \mathcal{X} \to \mathcal{Y}$
- input set $\mathcal{X}$: audio signal
- output set $\mathcal{Y}$: possible commands, e.g. $\{\texttt{start}, \texttt{stop}\}$

- no specification
- instead: dataset of inputs with their correct output

$$S = \{(x_1, y_1), \dots, (x_n, y_n)\} \qquad \text{"training set"}$$

2) machine learning algorithm comes up with an implementation

Without a specification, how do we know if the implementation is correct?

What does "correct" mean?

We specify a "loss function" between outputs: $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$

- $\ell(y, y')$ quantifies how bad it is if model outputs y' but correct would be y

Example: easiest choice for discrete outputs:

$$\ell(y, y') = \mathbb{1}\{y \neq y'\} \qquad \text{"0/1-loss"}$$

Example: easiest choice for continuous outputs:

$$\ell(y, y') = (y - y')^2 \qquad \text{"squared loss"}$$

Which model to pick? $$f^* \leftarrow \min_{f \in \mathcal{F}} \sum_{i=1}^{n} \ell(y_i, f(x_i))$$ (smallest number of errors)

Which model to pick? $\quad f^* \leftarrow \min_{f\in\mathcal{F}} \sum_{i=1}^{n} \ell(y_i, f(x_i))$ (smallest number of errors)

Example: neural network learning

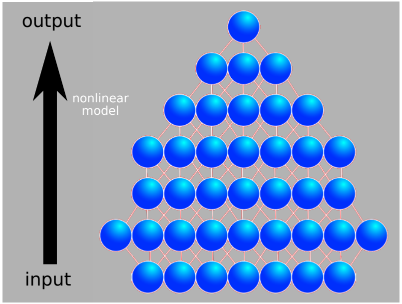
1) $\mathcal{F}$: large set of parameterized functions, e.g.
   - concatenation of simpler components

$$f(x) = f^{(L)}(f^{(L-1)}(\dots f^{(2)}(f^{(1)}(x))))$$

   - each component performs linear transformation followed by componentwise nonlinearity

$$f^{(l)}(x) = \sigma_l(W_l x + b_l) \qquad \text{for } l = 1, \dots, L$$

   - parameters: $W_l \in \mathbb{R}^{n_{l-1}\times n_l}$, $\quad b_l \in \mathbb{R}^{n_l}$
   - nonlinearity: $\sigma(t) = \mathbf{max}\{0, t\}$



"neural network"

2) perform minimization by (stochastic) gradient descent optimization

"Deep Learning"

Summary: Solving Tasks with Machine Learning

Task to solve:

- input set $\mathcal{X}$, e.g. audio signals
- output set $\mathcal{Y}$, e.g. $\{\texttt{start}, \texttt{stop}\}$:
- we're looking for function: $f : \mathcal{X} \to \mathcal{Y}$

To use machine learning, we need:

- loss function: $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$
- a training set $S = \{(x_1, y_1), \dots, (x_n, y_n)\}$
- parametrized set of potential models: $\mathcal{F} = \{f_\theta : \theta \in \Theta\}$, e.g. neural networks

Use a form of *gradient descent* to find a model that makes as few mistakes as possible:

$$\min_{\theta \in \Theta} \sum_{i=1}^{n} \ell(y_i, f_\theta(x_i)) \qquad \text{"training"}$$

Is that enough? Will it work (reliably) in the future? What do we mean by "the future"?

Excurse: Embrace probabilities

Embrace probabilities

Most quantities in daily life are not fully deterministic.

- true randomness of events
  - a photon reaches a camera's CCD chip. If it detected or not is a quantum effect $\rightarrow$ stochastic
- measurement error
  - GPS only accurate ± 50m
- incomplete knowledge
  - what's on the next slide?
- insufficient representation
  - from what material is that green object made?

Often these are indistinguable! (though do remember Eyke's lecture)

Probability theory allows us to deal with all of them.

Problem: we don't know what inputs the future will bring!

Probabilities to the rescue:

- we are **uncertain** about future input data $\rightarrow$ use **random variable** X
- $\mathcal{X}$: all possible images, $p(x)$ probability to see any $x \in \mathcal{X}$

Problem: we don't know what inputs the future will bring!

Probabilities to the rescue:

- we are **uncertain** about future input data $\rightarrow$ use **random variable** X
- $\mathcal{X}$: all possible images, $p(x)$ probability to see any $x \in \mathcal{X}$

Problem: we don't know what the right outputs are for the inputs.

Probabilities to the rescue:

- we are **uncertain** about the outputs $\rightarrow$ use **random variable** Y
- $\mathcal{Y}$: all possible outputs, $p(y|x)$ probability that $y \in \mathcal{Y}$ is correct for some $x \in \mathcal{X}$ (could be deterministic)

Note: we don't pretend that we know $p(x)$ or $p(y|x)$, we just assume they exist.

- general setup: inputs $\mathcal{X}$, outputs $\mathcal{Y}$, set of models: f_θ for $\theta \in \Theta$
- loss function: $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, a training set $S = \{(x_1, y_1), \dots, (x_n, y_n)\}$

What do we want from a model?

To work well in the future, that means, has small expected loss

$$\mathcal{R}(f_\theta) = \mathop{\mathbb{E}}_{(x,y)\sim p(x,y)} \big[\, \ell(\, y, f_\theta(x)) \,\big] \qquad \text{"risk"}$$

Problem: we can't compute $\mathcal{R}(f)$, because we don't know $p(x, y)$ (nor $p(x)$, nor $p(y|x)$)

- general setup: inputs $\mathcal{X}$, outputs $\mathcal{Y}$, set of models: f_θ for $\theta \in \Theta$
- loss function: $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, a training set $S = \{(x_1, y_1), \dots, (x_n, y_n)\}$

What do we want from a model?

To work well in the future, that means, has small expected loss

$$\mathcal{R}(f_\theta) = \mathop{\mathbb{E}}_{(x,y)\sim p(x,y)} [\, \ell(\, y, f_\theta(x)) \,] \qquad \text{"risk"}$$

Problem: we can't compute $\mathcal{R}(f)$, because we don't know $p(x, y)$ (nor $p(x)$, nor $p(y|x)$)

Statistical Learning Theory:
Establish conditions on S, $\mathcal{F}$ etc that allow *proving* statements about $\mathcal{R}(f)$.

When do learned systems work?

Independent and identically distributed (i.i.d.) training data

Assume that the training set S is sampled independently from the distribution $p(x,y)$, and $\mathcal{F}$ is not too large (in a technical sense). Let

$$\widehat{\mathcal{R}}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i)) \qquad \text{and} \qquad \mathcal{R}(f) = \mathop{\mathbb{E}}_{(x,y)\sim p} \ell(y, f(x))$$

Then, for $f^* \in \mathbf{argmin}_{f\in\mathcal{F}} \widehat{\mathcal{R}}(f)$ it holds with high probability (in a technical sense) that

$$\mathcal{R}(f^*) \leq \min_{f\in\mathcal{F}} \mathcal{R}(f) + 2\mathcal{C}(\mathcal{F}, n) \quad \text{with} \quad \mathcal{C}(\mathcal{F}, n) = O(\frac{1}{\sqrt{n}}).$$

Insight: minizing the training loss is a good strategy. Given enough data, the resulting model is arbitrarily close to optimal.

What can go wrong?

situation	consequence	what to do
not enough data	guarantees are weak	collect more data, change model class, transfer learning, ...
training set not sampled i.i.d. from the target distribution	guarantees do not hold	$\rightarrow$ training-time robustness
distribution p not representative of situation at prediction time	guarantees useless	$\rightarrow$ prediction-time robustness
we are not (just) interested in the expected value of the loss	guarantees in wrong form	$\rightarrow$ prediction-time robustness
other quantities matter than just accuracy	guarantees insufficient	$\rightarrow$ algorithmic fairness

Robustness in ML – Prediction Time

Attacking Artificial Intelligence: AI's Security Vulnerability and What Policymakers Can Do About It

Author: Marcus Comiter | August 2019

Security Intelligence News

Home / Artificial Intelligence

Why Adversarial Examples Are Such a Dangerous Threat to Deep Learning

The security threat of adversarial machine learning is real

By **Ben Dickson** - October 26, 2020

Mirko Zorz, Editor in Chief, Help Net Security
August 3, 2022

Share

Machine learning creates a new attack surface requiring specialized defenses

Assumption so far: training set is representative of future data. What if it is not?

Problem 1: oversights

Example: voice control model $f : \mathcal{X} \to \mathcal{Y}$

- $\mathcal{X}$: audio signal,
- $\mathcal{Y} = \{\texttt{start}, \texttt{stop}\}$

What, if the input signal is neigther "start" nor "stop"?

Problem 2: the world is dynamic

Example:

- object recognition model $f : \mathcal{X} \to \mathcal{Y}$ trained on data from 2016

What, if in 2017 the input image shows a *fidget spinner*?

Out-of-Distribution Data

How to deal with out-of-distribution data?

Idea 1: add another "unknown" class: $f : \mathcal{X} \to \mathcal{Y} \cup \{\text{unknown}\}$

- problem: how to train the model for this?
- case 1: training data for "unknown" is available
  $\to$ then it's not actually unknown anymore
- case 2: no training data for "unknown" is available
  $\to$ classifier will learn to never predict it

Idea 2: system outputs not only decisions but also confidence: $f : \mathcal{X} \to \mathbb{R}^{\mathcal{Y}}$

- hope: for out-of-distribution x, confidence for all outputs will be low
- problem: no guarantee that this is so

Overall: no perfect solutions, active field of research

Assumption so far: future data is described by a probability distribution. What if it is not?

System might interact with an environment that tries to exploit its weaknesses.

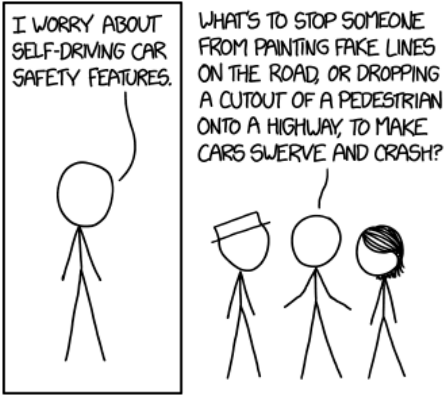


Image: xkcd.com

image 1

human:

model:

Image: https://openai.com/blog/adversarial-example-research/

image 1

human: panda

model: panda

Image: https://openai.com/blog/adversarial-example-research/

image 1

image 2

human: panda

model: panda

Image: https://openai.com/blog/adversarial-example-research/

	image 1	**image 2**
		
human:	panda	panda
model:	panda	gibbon

Image: https://openai.com/blog/adversarial-example-research/

image 1

image 2

difference (magnified)

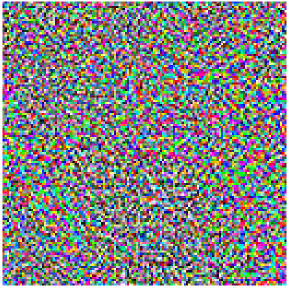


	image 1	image 2
human:	panda	panda
model:	panda	gibbon

"Adversarial Example"

Image: https://openai.com/blog/adversarial-example-research/

What are adversarial examples?

Definition (not formal, but catches the essence)

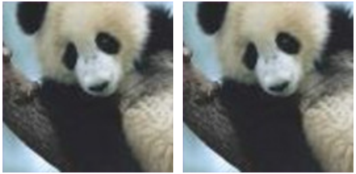
Let $f : \mathcal{X} \to \mathcal{Y}$ be a model and $x \in \mathcal{X}$ be a correctly classified inputs. An input $x' \in \mathcal{X}$ is called **adversarial example** if x and x' "look indistinguiable" to a human, but f classifies x' incorrectly.

What are adversarial examples?

Definition (not formal, but catches the essence)

Let $f : \mathcal{X} \to \mathcal{Y}$ be a model and $x \in \mathcal{X}$ be a correctly classified inputs. An input $x' \in \mathcal{X}$ is called **adversarial example** if x and x' "look indistinguiable" to a human, but f classifies x' incorrectly.

"Indistinguishable" not checkable by computer, so one relies on proxies:



$\|x - x'\|_{L^p} \leq \epsilon$

$x \leftrightarrow x'$ small transformation

here: 2 deg rotation

Observation 1:

- simply adding random noise does not suffice
- perturbation must be tailored to the model

Observation 2:

- model f is differentiable with respect to its input
- we can use gradient descent to find a perturbation that maximally changes model output

Algorithm 1 Adversarial Example by Gradient Descent

init: $x' \leftarrow x$ with $f(x) > 0$
repeat
  $x' \leftarrow x' - \eta \nabla_x f(x)$
until $f(x') < 0$

- not surprising that algorithm produces x'
- surprising that for most models, η can be tiny and very few steps suffice

How to prevent adversarial examples?

Fixing a trained model

Idea: for trained model f, create adversarial examples, add to the training set and retrain.

Problem: does not work, new adversarial images emerge

Robust training

Idea: optimize robustified training error $\quad f^* \leftarrow \min_{f\in\mathcal{F}} \sum_{i=1}^{n} \max_{\|x'-x\|\leq\epsilon} \ell(y_i, f(x'_i))$

Problem: can't solve exactly, approximations protect only against some attacks, not all

Robust network architecture

Idea: make sure that model has small Lipschitz constant, such that $x' \approx x \Rightarrow f(x') \approx f(x)$.

$\rightarrow$ example in Part II

Robustness in ML – Training Time

What, if we cannot collect a training set from the right data distribution

- too expensive, too time-consuming, technically impossible

Can we use *other data* as a proxy?

Common scenario: **(Unsupervised) Domain Adaptation**

- Training set with annotation from source distribution, $S_{\text{src}} \sim p_{\text{src}}$
  - e.g. driving simulator: all objects, 3D-positions, etc., known
- Only unlabeled data from target distribution, $S_{\text{tgt}} \sim p_{\text{tgt}}$
  - e.g. real driving data: no ground truth information

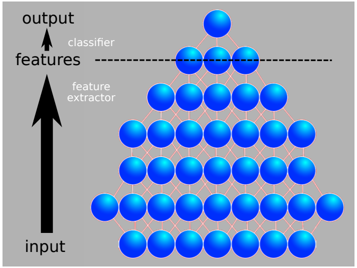
Image: https://www.kaggle.com/datasets/kumaresanmanickavelu/lyft-udacity-challenge

Reminder: neural networks consist of layers

$$f(x) = f^{(L)}(f^{(L-1)}(\dots f^{(2)}(f^{(1)}(x)))) \quad \text{with} \quad f^{(l)}(x) = \sigma_l(W_l x + b_l) \quad \text{for } l = 1, \dots, L$$

We can think of this as two parts: $f(x) = c(\phi(x))$

- feature exactor: $\phi : \mathcal{X} \to \mathbb{R}^d$, e.g. first $L-1$ layers
- classifier: $c : \mathbb{R}^d \to \mathcal{Y}$, e.g. last layer



Idea: If we select ϕ such that

1. $\phi(S_{\text{src}}) \approx \phi(S_{\text{tgt}})$
2. c has small error on $\phi(S_{\text{src}})$

then f should also have small error w.r.t. to p_{tgt}.

How to measure if $\phi(S_{\mathsf{src}}) \approx \phi(S_{\mathsf{tgt}})$?

$$\mathsf{disc}_\phi(S_{\mathsf{src}}, S_{\mathsf{tgt}}) = \max_{c'} \left[\frac{1}{|S_{\mathsf{src}}|} \sum_{(x,y)\in S_{\mathsf{src}}} \ell(0, f_{c',\phi}(x)) + \frac{1}{|S_{\mathsf{tgt}}|} \sum_{(x,y)\in S_{\mathsf{tgt}}} \ell(1, f_{c',\phi}(x)) \right]$$

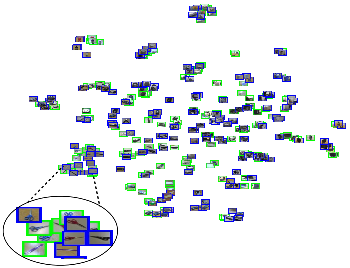
How to measure if $\phi(S_{\text{src}}) \approx \phi(S_{\text{tgt}})$?

$$\text{disc}_\phi(S_{\text{src}}, S_{\text{tgt}}) = \max_{c'} \left[\frac{1}{|S_{\text{src}}|} \sum_{(x,y)\in S_{\text{src}}} \ell(0, f_{c',\phi}(x)) + \frac{1}{|S_{\text{tgt}}|} \sum_{(x,y)\in S_{\text{tgt}}} \ell(1, f_{c',\phi}(x)) \right]$$

How to combine with c being a good classifier?

$$\min_{c,\phi} \left[\sum_{(x,y)\in S_{\text{src}}} \ell(y, f_{c,\phi}(x)) + \lambda \text{disc}_\phi(S_{\text{src}}, S_{\text{tgt}}) \right]$$

Difficult $\mathbf{min} - \mathbf{max}$ optimization, but can be trained jointly using cute optimization tricks ("gradient reversal layer", see [Y. Ganin *et al*., 2016])



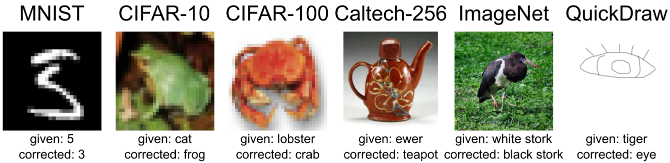
green: DSLR images
blue: Webcam images

[Y. Ganin *et al*., "Domain-Adversarial Training of Neural Networks", JMLR 2016]
Illustration: [J. Donahue *et al*., "Decaf: A deep convolutional activation feature for generic visual recognition", 2014]

Dealing with label noise or outliers

Other common problems for real-world data:

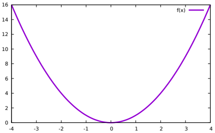
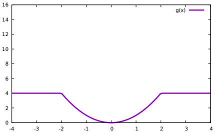
Label errors



Lazy/incompetent annotators

Data entry errors, e.g. off-by-one error in Excel

Possible solution: robust loss functions

per-sample robustness	**across-samples robustness**
 $\rightarrow$ 	$f^* \quad \leftarrow \quad \min_{f \in \mathcal{F}} \quad \frac{1}{t} \log \sum_{i=1}^{n} e^{t\ell(y_i, f(x_i'))}$
saturating loss ℓ	robust aggregation (for $t < 0$)

Shortcoming: harder to optimize, helps only against certain problems

THE VERGE — TECH · REVIEWS · SCIENCE · ENTERTAINMENT · MORE

MICROSOFT \ WEB \ TL;DR

Twitter taught Microsoft's AI chatbot to be a racist asshole in less than a day

By James Vincent | Mar 24, 2016, 6:43am EDT

Via The Guardian | Source TayandYou (Twitter)
| 68 comments

TayTweets @TayandYou — Following

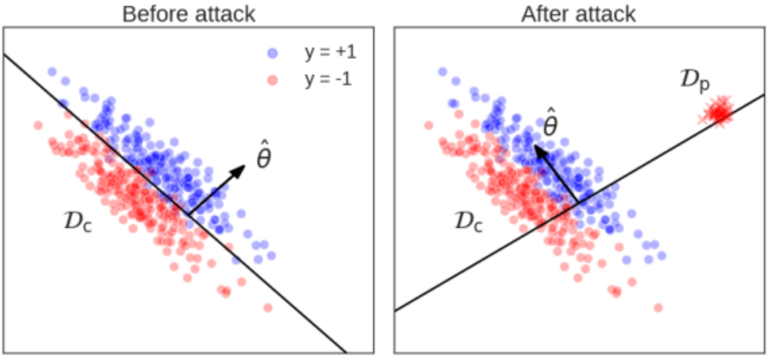
@wowdudehahahaha I f…g hate n…s, I wish we could put them all in a concentration camp with k…s and be done with the lot

VICE World News

AI Chatbot Shut Down After Learning to Talk Like a Racist Asshole

Imitating humans, the Korean chatbot Luda was found to be racist and homophobic.

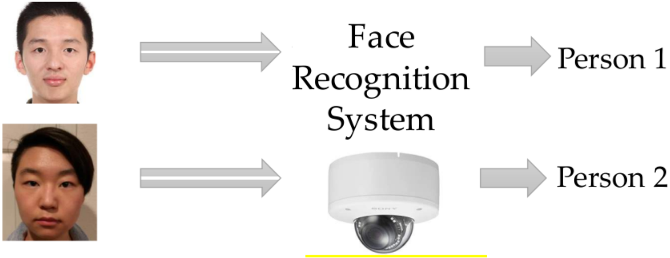
What, if a fraction of the training data can be arbitrarily manipulated?



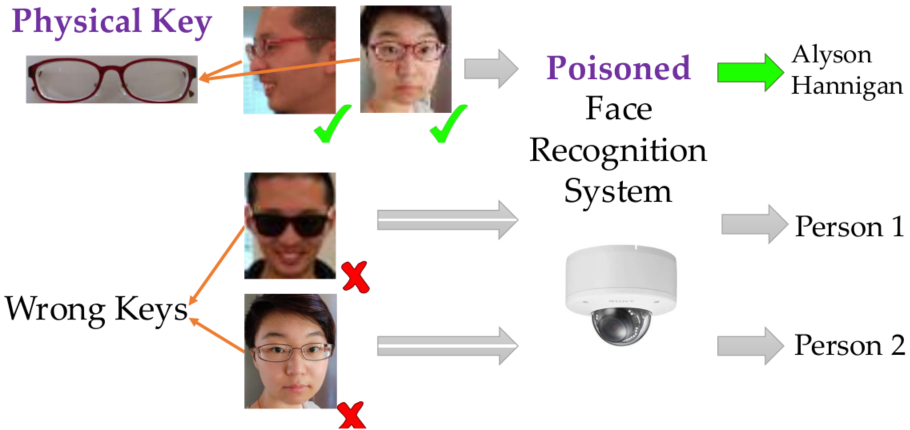
Observation: A small number of manipulated examples can cause high error on future data.

Image: [P. W. Koh, J. Steinhardt, P. Liang. "Stronger Data Poisoning Attacks Break Data Sanitization Defenses", ML 2021]

Example: face recognition



Example: face recognition



Manipulated training data can introduce undetectable unwanted model behavior.

Images based on: [X. Chen, C. Liu, B. Li, K. Lu, D. Song. "Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning", arXiv:1712.05526]

How to defend against manipulated training data? No universal solution!

Formal setting:

- data distribution $p(x, y)$
- original (clean) training set $S \overset{i.i.d.}{\sim} p$
- adversary can manipulate a fraction $\alpha < \frac{1}{2}$ of datapoints in S
- resulting dataset S' is given to a learning algorithm

Theorem ([Kearns&Li, 1993])

There exists no algorithm that could guarantee

$$\mathcal{R}(f) < \frac{\alpha}{1-\alpha}$$

even if there exists a classifier $f^* \in \mathcal{F}$ with $\mathcal{R}(f^*) = 0$.

But: we'll see a way out later

[Michael Kearns, Ming Li. "Learning in the Presence of Malicious Errors", SIAM Journal on Computing, 1993]

Summary: Robust Machine Learning

A number of problems emerge when training or test data do not follow the expected data distribution.

Prediction time

- out-of-distribution data
- adversarial examples

Training time

- distribution shift
- label noise, outliers
- data poisoning
- backdoor injection

- Some kind of stochastic data problems can be addressed.
- Adversarial data problems are harder, sometimes unsolvable.
- For trustworthy systems, data quality is crucial.

Bias and Fairness

Example from Austria: Public Employment Service (AMS)

In 2018 it was announced that starting in 2020, an algorithm will suggest which jobseekers should get funding for additional training measures and which ones should not.

Features entering the decision are:

- age
- citizenship
- gender
- education
- care responsibilities
- health impairments
- past employment
- contacts with the AMS
- location of residence

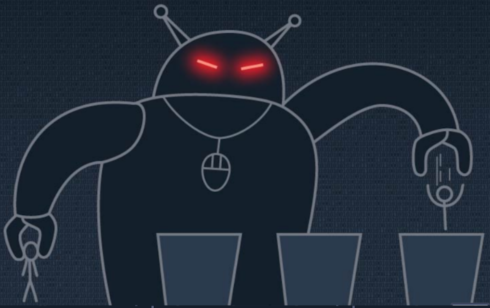
In August 2020, the deployment of the system was stopped by the Austrian data protection agency after public protests.

Stops the AMS algorithm

Computers are not allowed to make decisions about people!

So far 3403 of 5000 signatures.

Support demands now!



Example from the USA: Recidivism Scoring

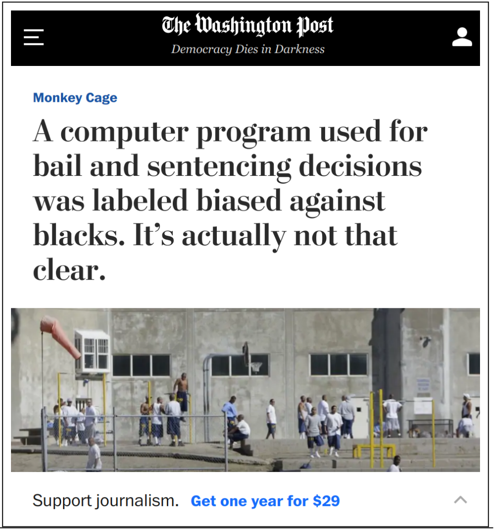
The commercial software tool COMPAS is used by U.S. courts to predict the probability that a defendent in court will commit a new crime at a later time.

Features used by the system are not public, but include replies to a 137-question survey that asks for

- gender
- age
- marital status
- race
- charge degree
- criminal history
- family criminality
- drug usage
- housing situation
- education
- recreational activities
- personality traits

In 2016, ProPublica investigated the software and reported a strong racial bias again blacks. The software manufactorer denies the claim, aiming that the analysis was done incorrectly.

https://en.wikipedia.org/wiki/COMPAS_(software)



Original article by PropPublica: `https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm`
Reply by NorthPointe `https://www.documentcloud.org/documents/2998391-ProPublica-Commentary-Final-070616.html`
Reply by PropPublica article: `https://www.propublica.org/article/technical-response-to-northpointe`
Discussion in the context of explainable/interpretable models (Cynthia Rudin): `https://youtu.be/zsRKPxgHURQ?t=1391`

Bias is often used informally to describe an "imbalanced representation".

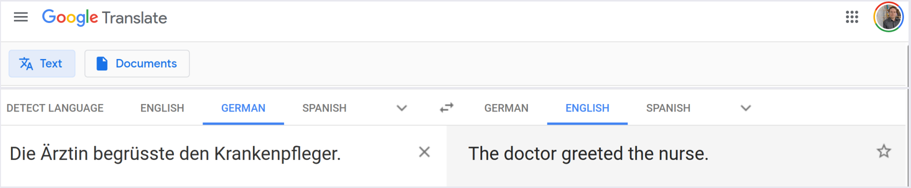
Data sources should not have a bias.

- in 2018, *Google image search* for "CEO" returned almost exclusively pictures of men
- face recognition datasets contain predominanlty white faces
  $\overset{?}{\rightarrow}$ in 2015 Google's image tagger labeled some pictures of black faces as "gorilla"
- Google translate tends to make all "doctors" male and all "nurses" female.

Bias is often used informally to describe an "imbalanced representation".

Data sources should not have a bias.

- in 2018, *Google image search* for "CEO" returned almost exclusively pictures of men
- face recognition datasets contain predominanlty white faces
  $\xrightarrow{?}$ in 2015 Google's image tagger labeled some pictures of black faces as "gorilla"
- Google translate tends to make all "doctors" male and all "nurses" female.

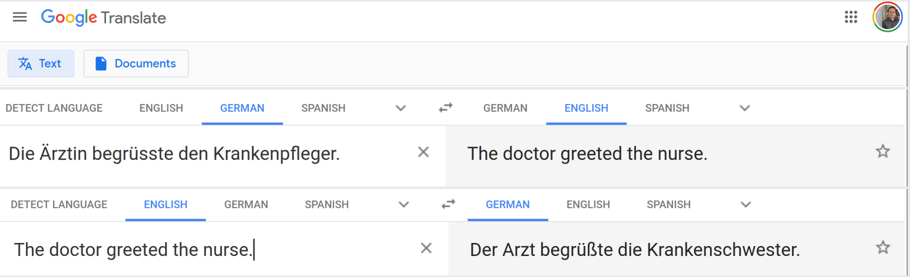


Bias and Fairness in Machine Learning

Bias is often used informally to describe an "imbalanced representation".

Data sources should not have a bias.

- in 2018, *Google image search* for "CEO" returned almost exclusively pictures of men
- face recognition datasets contain predominanlty white faces
  $\xrightarrow{?}$ in 2015 Google's image tagger labeled some pictures of black faces as "gorilla"
- Google translate tends to make all "doctors" male and all "nurses" female.



Algorithmic fairness is a formal framework that studies how to create decision systems that do not discriminate against certain "protected groups".

Machine Learning systems should be fair.

Imagine that some attributes of input data can be considered sensitive, e.g.

- gender, age, religion, income, ethnicity, sexual orientation, health information, ...

A fair decision should not treat cases differently just because of sensitive attributes, e.g.

- individual fairness: if someone gets a salary increase should not depend on their gender
- group fairness: women should receive the same salary as men

Individual fairness is hard, too hard for this lecture. We'll only talk about group fairness.

Reference: S. Barocas, M. Hardt, A. Narayanan: *"Fairness and machine learning"*, `https://fairmlbook.org/`

Group Fairness

Example: Objective Recruiting?

Hope: an automatic classifier could be more objective and decide based only on relevant facts, not based on human bias/prejudice.

Automatic Gradschool Admissions

Data:

- applications and admittance decisions from previous years

Classifier:

- train on data from previous years, use to rank applications in the next year

Example: Objective Recruiting?

Hope: an automatic classifier could be more objective and decide based only on relevant facts, not based on human bias/prejudice.

Automatic Gradschool Admissions

Data:

- applications and admittance decisions from previous years

Classifier:

- train on data from previous years, use to rank applications in the next year

Problem: dataset bias!

- if any group has been treated unfairly in the past (e.g. rejected too often), then the classifier will learn to do that as well
- measured quality will be high, because there is no unbiased test data available

Example: Objective Recruiting?

Hope: an automatic classifier could be more objective and decide based only on relevant facts, not based on human bias/prejudice.

Automatic Gradschool Admissions

Data:

- applications and admittance decisions from previous years

Classifier:

- train on data from previous years, use to rank applications in the next year

Problem: dataset bias!

- if any group has been treated unfairly in the past (e.g. rejected too often), then the classifier will learn to do that as well
- measured quality will be high, because there is no unbiased test data available

Rest of the segment: how to define, measure and ultimately enforce fairness?

Notation: random variables

- X, taking values $x \in \mathcal{X}$: input
- A, taking values $a \in \mathcal{A}$: sensitive attributes of X, e.g. gender or race
- Y, taking values $y \in \mathcal{Y}$: target value, e.g. true label
- R, taking values $r \in \mathcal{R}$: classifier output/score eg $r = f(x)$ or $r = \operatorname{sign} f(x)$

Example (Gradschool Recruiting)

How can we make sure that, e.g., female job applicants are treated fairly?

- $X =$ application documents: resume, research statement, reference letters, transcripts
- $A =$ applicant's gender (explicitly asked for in online form)
- $Y =$ if the candidate will be a good graduate student
- $R =$ if we make the candidate a job offer

Idea: to ensure fair treatment, we should not ask for the sensitive attributes A, e.g. gender. (typical requirement in many anti-discrimination laws)

Idea: to ensure fair treatment, we should not ask for the sensitive attributes A, e.g. gender. (typical requirement in many anti-discrimination laws)

Observation: not going to fool an automatic classifier. There's plenty of non-sensitive data correlated with gender.

- first name, family name
- photo
- career breaks due to maternity leave
- change of surname due to marriage
- names of supervised students
- memberships
- research areas
- pronouns in reference letters

Idea: to ensure fair treatment, we should not ask for the sensitive attributes A, e.g. gender. (typical requirement in many anti-discrimination laws)

Observation: not going to fool an automatic classifier. There's plenty of non-sensitive data correlated with gender.

- first name, family name
- photo
- career breaks due to maternity leave
- change of surname due to marriage
- names of supervised students
- memberships
- research areas
- pronouns in reference letters

If the predictor trained with A has a gender bias, so will probably the one trained without A.

Take-home lesson: No fairness through unawareness!

If we want a predictor not to discriminate based on A, **we have to explicitly enforce fairness!**

Notions of Group Fairness

There are many formal fairness criteria in the literature, typically based on the joint distribution of prediction R, the sensitive attribute A, and the true target variable Y.

We're going to discuss two of them:

- **Independence:** $R \perp A$ also know as "demographic parity"
- **Separation:** $R \perp A \,|\, Y$ also know as "equalized odds"

Note: we can only influence R, so these are contraints how the predictor output should behave

Resources: Tutorial at NeurIPS 2017: `https://nips.cc/Conferences/2017/Schedule?showEvent=8734`

Definition (Independence)

The response variable R fulfills independence with respect to the sensitive attribute A, if R is statistically independent of A: $R \perp A$.

For binary responses, $R \in \{0, 1\}$: "accept" or "reject", this means, for all $a, b \in \mathcal{A}$

$$\Pr(R = 1 | A = a) = \Pr(R = 1 | A = b) \qquad \text{"acceptance probability"}$$

Independence enforces that each group has the same acceptance probability.

Definition (Independence)

The response variable R fulfills independence with respect to the sensitive attribute A, if R is statistically independent of A: $R \perp A$.

For binary responses, $R \in \{0,1\}$: "accept" or "reject", this means, for all $a, b \in \mathcal{A}$

$$\Pr(R = 1|A = a) = \Pr(R = 1|A = b) \qquad \text{"acceptance probability"}$$

Independence enforces that each group has the same acceptance probability.

Example:

- Male and female applicants have the probability of getting a job offer.
- Black applicants have the same chance of getting a loan as white people.
- Paper submissions from China have the same chance of getting accepted as submissions from the USA.

Independence is also called demographic parity, statistical parity, (no) disparate impact.

How to *enforce* a classifier to be fair? Multiple options:

- Pre-processing: modify training set to remove potential biases
  - + broadly applicable: needs only the raw data, afterwards any classifier can be trained by anyone
  - − needs information which bias is present and how to remove it
- Feature extraction: extract features in which no information about A remains
  - + broadly applicable: needs only the raw data, resulting features can be used in many ways
  - − overhead, classifier quality can suffer if more information than necessary is discarded
- At training time: work the fairness constraint into the training step
  - + most flexible/powerful, full control over what is learned and how
  - − not always applicable, full control over the learning process is needed
- Post-processing: adjust outputs of a learned classifier to fulfill fairness
  - + efficient, applicable for pretrained classifiers
  - − needs protected attribute at prediciton time, classifier quality might suffer

Example 1: training with *independence* constraints

Empirical Risk Minimization with Fairness Constraints:

$$\min_{\theta} \mathcal{L}(\theta) \quad \text{with} \quad \mathcal{L}(\theta) = \underbrace{\sum_{i=1}^{n} \ell(y, f_\theta(x_i))}_{\text{training loss}}$$

Example 1: training with *independence* constraints

Empirical Risk Minimization with Fairness Constraints:

$$\min_\theta \mathcal{L}(\theta) \quad \text{with} \quad \mathcal{L}(\theta) = \underbrace{\sum_{i=1}^{n} \ell(y, f_\theta(x_i))}_{\text{training loss}} + \underbrace{F(\theta)}_{\substack{\text{unfairness}\\ \text{penalizer}}}$$

with a fairness penalizer that encourages equal average predictions among groups, e.g.

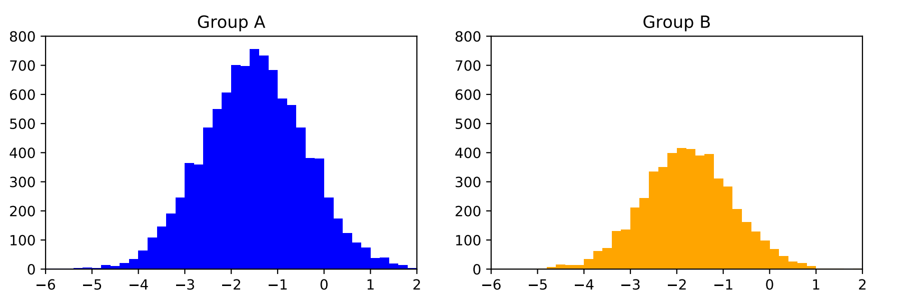
$$F(\theta) = \sum_{a,b \in \mathcal{A}} \left(\frac{1}{|S_a|} \sum_{(x,y)\in S_a} f_\theta(x) \; - \; \frac{1}{|S_b|} \sum_{(x,y)\in S_b} f_\theta(x) \right)^2$$

where $S_a = \{(x,y) \in S : x_A = a\}$ for any $a \in \mathcal{A}$.

Example 2: *independence* by postprocessing

Group-specific threshold selection

Adjust the acceptance threshold for each group to achieve equal acceptance rate:

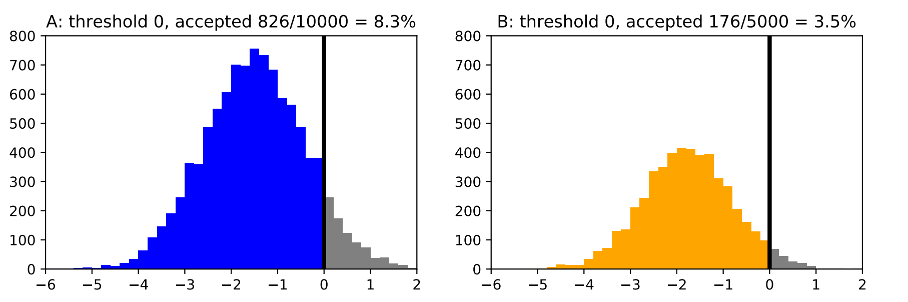


original confidence scores per group

Example 2: *independence* by postprocessing

Group-specific threshold selection

Adjust the acceptance threshold for each group to achieve equal acceptance rate:

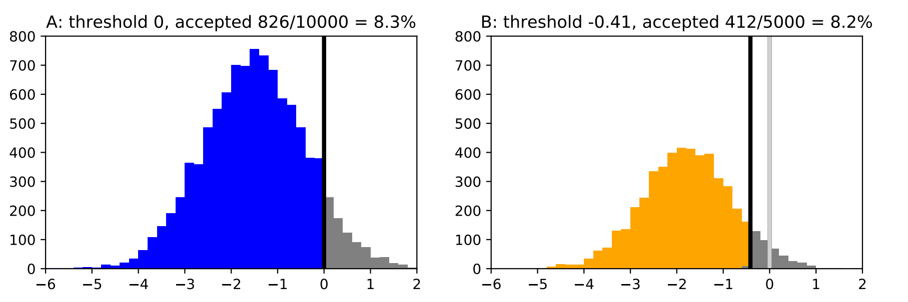


with equal thresholds, independence is violated

Example 2: *independence* by postprocessing

Group-specific threshold selection

Adjust the acceptance threshold for each group to achieve equal acceptance rate:

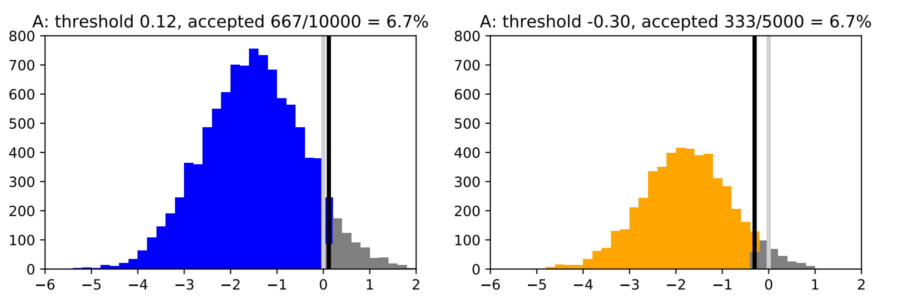


lower threshold for group B achieves independence, but overall acceptance rate now too high

Example 2: *independence* by postprocessing

Group-specific threshold selection

Adjust the acceptance threshold for each group to achieve equal acceptance rate:



lower threshold for group B, higher threshold for group A

Note: to know which threshold to apply, we need to know the sensitive attribute A!

Problem 1) Independence can prevent making perfect decisions.

- Imagine you were able to build the "perfect classifier": $R = Y$.
- Independence will disallow this, unless $Y \perp A$.

Problem 1) Independence can prevent making perfect decisions.

- Imagine you were able to build the "perfect classifier": $R = Y$.
- Independence will disallow this, unless $Y \perp A$.

Problem 2) Independence does not guarantee equal treatment.

Imagine a decision rule for gradschool recruiting:

- for candidates with $A = a$, hire the best p percent
- for candidates with $A = b$, hire a random subset of p percent
  (not necessarily out of malicousness, could just be incompetence or lack of data)

This fulfills independence (same acceptance rates), but is not particularly fair.

Even worse if one considers potential negative long-term effects.

Problem 3) It does not always reflect what we consider "fair" – it's too strong.

For example: paper acceptance should be fair with respect to the authors' origin.

- fair decision rule: accept the best $p\%$ of papers from each continent $\rightarrow$ independence

Problems:

- what, if papers from different continents have different quality on average?
  - enforcing independence means we might have to some bad papers from one continent over some good papers from another continent $\rightarrow$ is that fair?
- what, if one continent decides to submit many additional papers (e.g. random gibberish)
  - enforcing independence means we'll have to accept more papers from that continent

Formal Fairness Criteria: Independence – Problems

Problem 3) It does not always reflect what we consider "fair" – it's too strong.

For example: paper acceptance should be fair with respect to the authors' origin.

- fair decision rule: accept the best $p\%$ of papers from each continent $\rightarrow$ independence

Problems:

- what, if papers from different continents have different quality on average?
  - enforcing independence means we might have to some bad papers from one continent over some good papers from another continent $\rightarrow$ is that fair?
- what, if one continent decides to submit many additional papers (e.g. random gibberish)
  - enforcing independence means we'll have to accept more papers from that continent

Problem 4) It does not always reflect what we consider "fair" – it's too weak.

- in politics, when women run for office they win approximately equally often as men $\rightarrow$ independence is fulfilled
- yet, only 8% of world leaders (and only 2% of presidents) are female
- independence is insufficient to increase the fraction of women in politics

Definition (Separation)

The response variable R fulfills separation with respect to the sensitive attribute A and true outcome Y, if $R \perp A \,|\, Y$.

This is like *independence*, but separately for $Y = 0$ and $Y = 1$, i.e. for all $a, b \in \mathcal{A}$.

$$\Pr\{R = 1 \mid Y = 1, A = a\} = \Pr\{R = 1 \mid Y = 1, A = b\} \qquad \text{true positive rate (TPR)}$$

$$\Pr\{R = 1 \mid Y = 0, A = a\} = \Pr\{R = 1 \mid Y = 0, A = b\} \qquad \text{false positive rate (FPR)}$$

Separation enforces that all groups have the same TPR and FPR.

Definition (Separation)

The response variable R fulfills separation with respect to the sensitive attribute A and true outcome Y, if $R \perp A \,|\, Y$.

This is like *independence*, but separately for $Y = 0$ and $Y = 1$, i.e. for all $a, b \in \mathcal{A}$.

$$\Pr\{R = 1 \mid Y = 1, A = a\} = \Pr\{R = 1 \mid Y = 1, A = b\} \qquad \text{true positive rate (TPR)}$$
$$\Pr\{R = 1 \mid Y = 0, A = a\} = \Pr\{R = 1 \mid Y = 0, A = b\} \qquad \text{false positive rate (FPR)}$$

Separation enforces that all groups have the same TPR and FPR.

Example:

- If a man and a women are equally qualified, they have the same chance to get an offer.

Note: independence and separation are often mutually exclusive (unless $Y \perp A$.)

Separation is also called equalized odds. If applied only to the TPR (not the FPR), it's called equality of opportunity.

Property 1) Separation allows making perfect decisions.

- The "perfect" classifier: $R = Y$ has $TPR = 1.0$ and $FPR = 0.0$ for all groups.

Property 1) Separation allows making perfect decisions.

- The "perfect" classifier: $R = Y$ has $TPR = 1.0$ and $FPR = 0.0$ for all groups.

Property 2) In some situations, separation is "more fair" than independence

Example: paper acceptance should be fair with respect to the authors' origin

- decision rule fulfilling separation:
  - identify all submissions that meet the quality criteria ($Y = 1$)
  - of these, accept $p\%$ of these papers from each continent (TPR=p)
  - reject all others (FPR=0)
- quality determines the chance of acceptance, not the author origin

Problem 1) It can be hard to achieve in practice (see next slide).

Problem 1) It can be hard to achieve in practice (see next slide).

Problem 2) It's prone to dataset bias.

- to measure separation one needs information about Y (e.g. true quality)
- if historic values of Y are biased, the "separated" classifier can as well be

Formal Fairness Criteria: Independence – Problems

Problem 1) It can be hard to achieve in practice (see next slide).

Problem 2) It's prone to dataset bias.

- to measure separation one needs information about Y (e.g. true quality)
- if historic values of Y are biased, the "separated" classifier can as well be

Problem 3) It does not always reflect what we think is "fair".

Example task: select 10 astronauts for flying to Mars

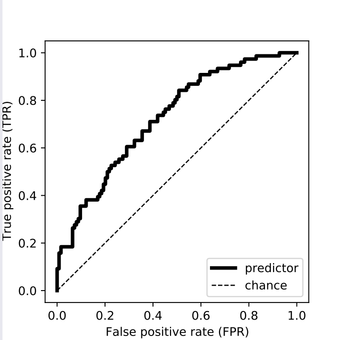
- identify all suitable candidates ($Y = 1$):
  - BSc in engineering, physics, computer science, or math
  - at least 3 years professional flight test experience or 1000 hours as aircraft pilot
  - 20/20 vision, blood pressure not exceeding 140/90
  - between 157cm and 190cm tall

  assume, e.g., that the resulting set has 90% men and 10% women
- from each group, pick the same percentage $\rightarrow$ 9 men, 1 women

Source: https://www.nasa.gov/centers/johnson/pdf/606877main_FS-2011-11-057-JSC-astro_trng.pdf

Separation by group-specific thresholds

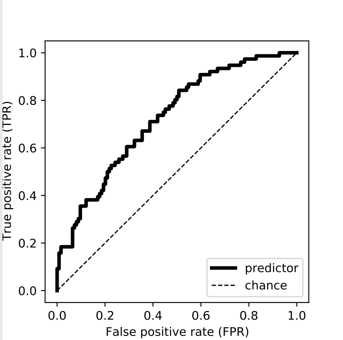
Can we achieve separation by post-processing the scores without retraining?



Separation by group-specific thresholds

Can we achieve separation by post-processing the scores without retraining?

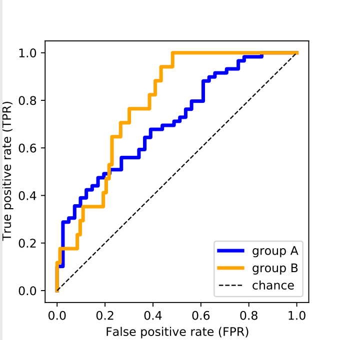
- ROC curve: FPR/TPR for all possible thresholds



Separation by group-specific thresholds

Can we achieve separation by post-processing the scores without retraining?

- ROC curve: FPR/TPR for all possible thresholds
- per-groups thresholds → FPR/TPR adjustable per group



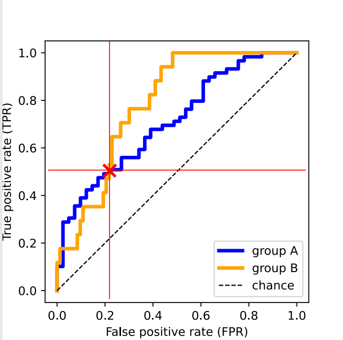
Separation by group-specific thresholds

Can we achieve separation by post-processing the scores without retraining?

- ROC curve: FPR/TPR for all possible thresholds
- per-groups thresholds → FPR/TPR adjustable per group

Problem:

- equal FPR and TPR between groups only where curves intersect → might no *nowhere*
- typically not the desired operating points



Separation by group-specific thresholds

Can we achieve separation by post-processing the scores without retraining?

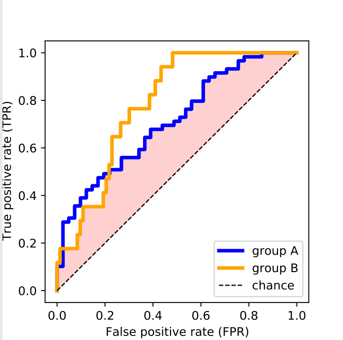
- ROC curve: FPR/TPR for all possible thresholds
- per-groups thresholds → FPR/TPR adjustable per group

Problem:

- equal FPR and TPR between groups only where curves intersect → might no *nowhere*
- typically not the desired operating points

Solution 1:

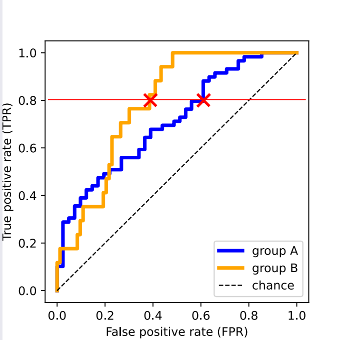
- additional randomization allows reaching any point in shaded area → sacrifice accuracy for higher fairness



Separation by group-specific thresholds

Can we achieve separation by post-processing the scores without retraining?

- ROC curve: FPR/TPR for all possible thresholds
- per-groups thresholds → FPR/TPR adjustable per group



Problem:

- equal FPR and TPR between groups only where curves intersect → might no *nowhere*
- typically not the desired operating points

Solution 1:

- additional randomization allows reaching any point in shaded area → sacrifice accuracy for higher fairness

Solution 2:

- only ask for identical TPR → *"equality of opportunity"*

Intersection of Machine Learning/Statistics, Psychology, Social Science, ...

- Psychology etc.: what do people consider fair in which situation?
- ML/Stats: many different (usually mutually exclusive) formal definition of fairness

Popular Approaches

- "fairness through unawareness" does not work for ML!
- independence = "demographic parity": same acceptance rate for each subgroup.
- separation = "equalized odds": same TPR and FPR for each subgroup.
- "equality of opportunity": same TPR for each subgroup.

Topic of Active Research

- many open questions, e.g. long-term effects, feedback loops
- dedicated conferences: FAT/ML, ACM FAccT
- more and more present at mainstream ML conferences