Machine Learning

Lecture 13: Advanced Topics

Prof. Dr. Stephan Günnemann
Aleksandar Bojchevski

Data Analytics and Machine Learning
Technical University of Munich

Winter term 2020/2021

Roadmap

Introduction – Beyond Accuracy

Differential Privacy

Algorithmic Fairness

Overview

- In the previous lectures we were focusing on models and algorithms and we were optimizing for simple metrics (e.g. misclassification rate, reconstruction error, etc.)
- As ML/AI is becoming more widespread and is used in critical applications (e.g. algorithmic decision-making involving humans) we have to consider societal impacts
- As ML models get deployed in the real-world they create feedback loops which can have potentially unintended consequences
- We should always ask: "Are we optimizing for the right thing?"

Beyond accuracy we have to consider

- **Privacy**: how to avoid revealing (sensitive) information about the individuals in the training set (e.g. medical diagnosis)
- **Security**: what if an attacker can "fool" the system with malicious input, poison the training data, "steal" the model, etc.?
- **Fairness**: how can we ensure that the system does not disadvantage particular individuals or (marginalized) groups
- **Explainability**: people should be able to understand why and how a decision was made about them (GDPR)
- **Accountability**: an outside auditor should be able to verify that the system is functioning as intended

Impact of AI/ML more broadly

- How should self-driving cars trade off the safety of passengers, pedestrians, etc.? (Trolley problems)
- Face recognition and other surveillance-enabling technologies
- Autonomous weapons
- Risk of international AI arms races
- Long-term risks of superintelligent AI
- Unemployment due to automation
- Bad side effects of optimizing for click-through

Disclaimer

- These concepts sound "vague" and properly formalizing them is half the challenge [1]
- Any "solution" can only be partly technical and tackling these issues must always involve social/legal/political aspects and must be an interdisciplinary effort
- Given the above we will focus on three topics: privacy and fairness since they have well-established technical principles and techniques that address part of the problem

[1] Most of these topics in this lecture are active areas of research with serious effort starting only around 5 years ago.

Section 2

Differential Privacy

Anonymization is hard

- US government releases a dataset of medical visits (Sweeney, '13)
  - Identifying info (names, addresses and SSNs) was removed
  - Data on zip code, birth date, and gender was left
  - Around 87% of Americans are uniquely identifiable from this triplet
- Netflix Challenge: competition to improve movie recommendations
  - Dataset of 100 million movie ratings with anonymized user ID
  - 99% of users who rated at least 6 movies could be identified by cross-referencing with IMDB reviews (associated with real names) (Narayanan & Smahtikov '08)
- Re-Identifying >40% of anonymous volunteers in DNA Study (Sweeney, '13)
- "A Face is Exposed for AOL Searcher No. 4417749" (Barrabo, '06)
- ... and many others

Anonymization is hard

It is not sufficient to prevent unique identification of individuals

Name	Age	Gender	Zip	Smoker	Diagnosis
*	60-70	Male	191**	Y	Heart disease
*	60-70	Female	191**	N	Arthritis
*	60-70	Male	191**	Y	Lung Cancer
*	60-70	Female	191**	N	Crohn's disease
*	60-70	Male	191**	Y	Lung Cancer
*	50-60	**Female**	191**	N	HIV
*	50-60	Male	191**	Y	Lyme disease
*	50-60	Male	191**	Y	Seasonal allergies
*	50-60	**Female**	191**	N	Ulcerative Colitis

Table: Kearns & Roth, The Ethical Algorithm

If we know Rebecca is 55 years old and in this (fictional) hospital database, then we know she has 1 of 2 diseases.

Sensitive information in the model

- Even if you don't release the raw data, the weights of a trained network might reveal sensitive information
- **Model inversion** attacks recover information about the training data from the trained model

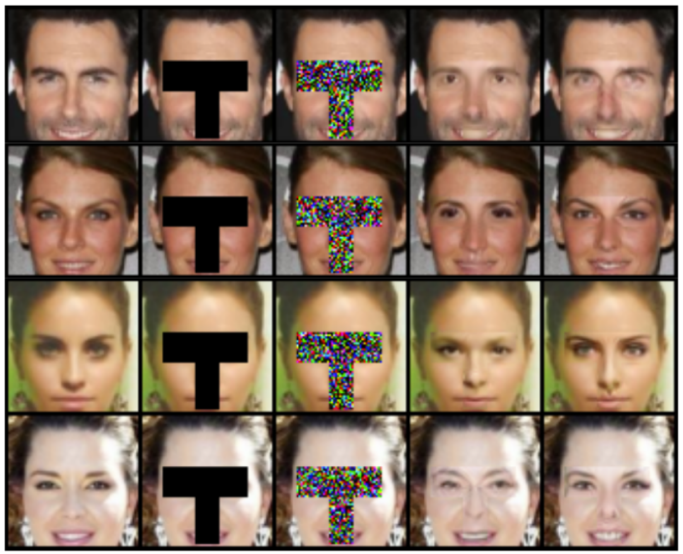


Figure: Reconstructing faces given a classifier trained on private data and a generative model trained on public data, Zhang et al., 2019

- Email provider uses language models for email autocompletion, the model can remember (and spit out) sensitive info from past emails

How can we compute (statistical) queries and train ML models without leaking (too much) (sensitive) information about any individual?

Warmup: Randomized Response

- Goal: Conduct a survey on a potentially incriminating or sensitive question with a binary (yes/no) answer
- Examples:
  - Have you ever committed tax fraud?
  - Does anyone in your family suffer from HIV?

- We would like to motivate the participants to answer truthfully despite the sensitive nature of the question
- Idea: introduce randomization to provide **plausible deniability**

Warmup: Randomized Response

- Let each of the n participants follow the procedure (Warner, 1965):
  - Flip a coin
  - If it lands tails answer truthfully
  - Else flip another coin. If it lands tails answer Yes, else answer No
- What is the fraction of participants that answer Yes truthfully?

- We can accurately estimate the population mean μ from the randomized responses by $\mu = \frac{1}{4}(1 - \hat{\mu}) + \frac{3}{4}\hat{\mu}$, where $\hat{\mu}$ is the MLE
  - $P(\text{response} = \text{Yes} \mid \text{truth} = \text{Yes}) = \frac{3}{4}$
  - $P(\text{response} = \text{Yes} \mid \text{truth} = \text{No}) = \frac{1}{4}$
- μ is an unbiased estimator of the non-randomized mean, the variance decays as $\frac{1}{n}$ but it is $4\times$ larger because of the randomization

Beyond Randomized Response

- With Randomized Response we could compute useful queries (e.g. the fraction) in aggregate without learning the truthful answer for any individual
- In general, randomness is a useful technique for preventing information leakage

- Q: How to answer more complex (general) queries (e.g. computing arbitrary functions over data) with mathematical privacy guarantees?
- A: Differential Privacy

Differential Privacy

- A (trusted) curator is given access to some input data $X \in \mathcal{X}$
- The curator computes some function $Y = f(X) \in \mathcal{Y}$
- The curator wants to release the output Y to the public without leaking (too much) information about the data

Example:

- Let $\mathcal{X} = \{0,1\}^n \cup \{0,1\}^{n+1}$ be the set of binary vectors, e.g. containing the answers to a survey question for n or $(n+1)$ users
- Let f be the mean, thus $\mathcal{Y} \in [0,1]$
- Let $X, X' \in \mathcal{X}$ be two "neighboring" input vectors, s.t. X' is obtained by appending the answer for a new participant to X
  - Or alternatively they differ in the answer of a single participant
- Informally, DP enforces that $f(X)$ and $f(X')$ do not differ significantly preventing to leak the answer of the new participant

Differential Privacy

General Setup:

- Input space $\mathcal{X}$ with symmetric neighboring relation $\simeq$
- Output space $\mathcal{Y}$ (with σ-algebra of measurable events)
- Function of interest f, and Privacy parameter $\epsilon \geq 0$

Definition

A randomized mechanism $\mathcal{M}_f : \mathcal{X} \to \mathcal{Y}$ [2] is ϵ-differentially private if **for all** neighboring inputs $X \simeq X'$ and **for all** sets of outputs $Y \subseteq \mathcal{Y}$ we have:[3]

$$\mathbb{P}[\mathcal{M}_f(X) \in Y] \leq e^{\epsilon}[\mathcal{M}_f(X') \in Y]$$

For any possible set of outputs we have

$$e^{-\epsilon} \leq \frac{\mathbb{P}[\mathcal{M}_f(X) \in Y]}{\mathbb{P}[\mathcal{M}_f(X') \in Y]} \leq e^{\epsilon}$$

[2] Note, $\mathcal{M}_f$ includes the function f we want to compute. It is not useful to just output random numbers.

[3] There is a more general (ϵ, δ) definition which we won't cover in the lecture

Differential Privacy Intuition

The outcome of the statistical analysis should not change by much if we include/exclude or modify the information of a single instance

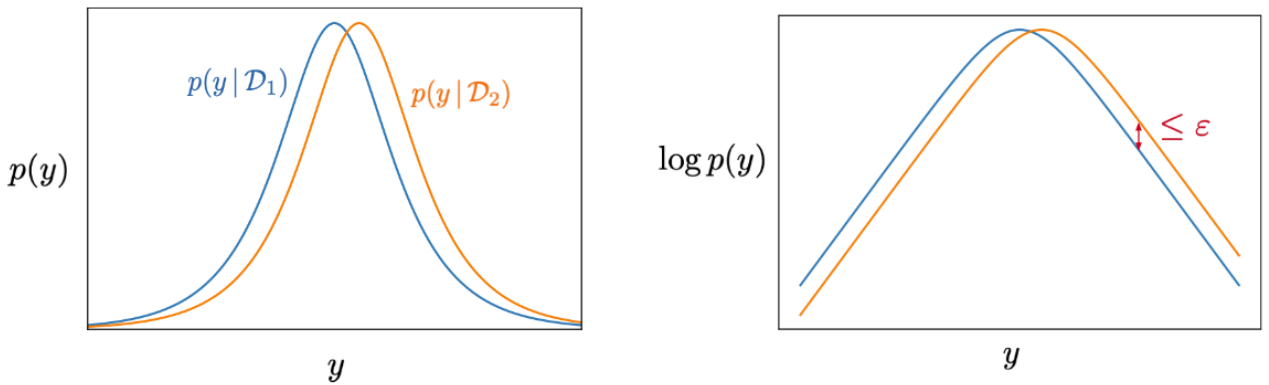


Figure: Difference between two neighboring datasets (Grosse, CSC 2515)

- $\simeq$ captures what is protected, e.g. two different vectors X and X' that differ in a single coordinate, or two different datasets $\mathcal{D}$ and $\mathcal{D}'$ that differ in single instance
- If the mechanism $\mathcal{M}_f$ behaves nearly identically for X and X', an attacker can't tell whether X or X' was used and thus can't learn much about the individual

Laplace Mechanism (Output Perturbation)

- Define the global sensitivity of a function $f : \mathcal{X} \to \mathbb{R}^d$ as $\Delta_p = \sup_{X \simeq X'} ||f(X) - f(X')||_p$
- Δ_p measures the magnitude by which a single instance can change the output of the function in the worst case

- Output perturbation with Laplace Noise:
  - A curator holds data $X = (x_1, \ldots, x_n) \in \mathcal{X}$ about n individuals
  - The curator computes the function $f(X)$
  - They sample i.i.d. Laplace noise $Z \sim \text{Lap}(0, \frac{\Delta_1}{\epsilon})^d$
  - They reveal the noisy value $f(X) + Z$

  We can prove that: $\mathcal{M}_{f,\text{Lap}}$ is ϵ-DP [4]

[4] Adding Gaussian noise $Z \sim \mathcal{N}(0, \sigma^2)$ to the output of f with $\sigma = \frac{\Delta_2 \sqrt{C \log(1/\delta)}}{\epsilon}$ yields a (ϵ, δ)-DP private mechanism $\mathcal{M}_{f,\mathcal{N}}$, where δ accounts for "bad" events

Example: Laplace Mechanism for the Mean

- Computing the mean $\mu = f(X) = f(x_1, \ldots, x_n) = \frac{1}{n} \sum_{i=1}^{n} x_i$ where $x_i \in \{0, 1\}$ is binary
- The global l_1-sensitivity of the mean is $\Delta_1 = \frac{1}{n}$
  - Changing/excluding the value of a single instance/individual can change the output by at most $\frac{1}{n}$
- Sample noisy $Z \sim \text{Lap}(0, \frac{1}{\epsilon n})$
- Reveal the noisy mean $\tilde{\mu} = \mu + Z$

- In this case we can also say something about the **utility** of this mechanism
  - $|\mu - \tilde{\mu}| = \text{Exponential}(\epsilon n)$, which has a mean of $(\frac{1}{\epsilon n})$
  - The true mean is not going to differ by much from the randomized mean and this difference decreases with the size of the data
- In general, computing the sensitivity of a function is challenging, and showing something about the utility is even more challenging

DP for Machine Learning

Goal:

- The curator would like to learn an ML model on a dataset $\mathcal{D}$, i.e. find the optimal weights θ^* of some parametrized model
- And release the weights θ^* in public, e.g. by publishing the trained classifier

Setup:

- Let $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ be a dataset of instances, and $\theta \in \Theta$ be the parameters of a model (e.g. logistic regression weights)
- Let $f(\mathcal{D}) = \arg\min_{\theta \in \Theta} \mathcal{L}(\mathcal{D}, \theta)$ be the trained weights
  - e.g. $\mathcal{L}(\mathcal{D}, \theta) = \frac{1}{n} \sum_{i=1}^{n} l(x_i, y_i, \theta)$, where l is e.g. cross-entropy
- Define for two datasets $\mathcal{D} \simeq \mathcal{D}'$ if they differ in one instance (x_j, y_j)
  - Compared to before where we had vectors X and X', now we have entire datasets $\mathcal{D}$ and $\mathcal{D}'$

Differential Privacy techniques for ML models

- **Perturb input**: perturb $\mathcal{D}$ directly and rely on the post-processing property (next slide)
- **Perturb weights**: Compute the optimal weights $\theta^* = \arg\min \mathcal{L}_{\theta \in \Theta}(\mathcal{D}, \theta)$ and perturb them with Laplace noise
  - Need to calculate the global sensitivity of the optimization procedure which is difficult for complex models
- **Perturb objective**: Optimize $\mathcal{L}(\mathcal{D}, \theta) + \theta^T Z$ where Z is some noise
- **Perturb gradients**: Perturb and release the gradient of $\mathcal{L}$ w.r.t. a mini-batch, useful in Federated Learning (next slides)

Fundamental properties of DP

- **Robustness to post-processing**: if $\mathcal{M}$ is ϵ-DP then $g \circ \mathcal{M}$ is ϵ-DP
  - You can apply any function g on an output from a DP mechanism and the new output remains DP (as long as you don't touch again the data)
- **Composition**: if $\mathcal{M}_j, j = 1, \ldots, k$ are ϵ_j-DP then $X \to (\mathcal{M}_1(X), \ldots, \mathcal{M}_k(X))$ is $(\sum_{j=1}^{k} \epsilon_j)$-DP[5]
- **Group privacy**: if $\mathcal{M}$ is ϵ-DP with respect to $X \simeq X'$ then $\mathcal{M}$ is $(t\epsilon)$-DP with respect to changes of t instances/individuals

- DP generalizes to arbitrary spaces beyond Euclidean space using the Exponential noise mechanism

[5] There is an advanced composition theorem with better guarantees

Federated Learning

- Differential Privacy assumes there's a curator who we trust with access to all the raw data
  - Compare this to the Randomized Response procedure where the user randomizes their answer before sending it the curator
  - Allows for better privacy guarantees but not always possible
- **Federated learning**: learning a model without any centralized entity having access to all the data
  - Send the current weights of the model to the user
  - Perform few steps of SGD locally
  - Send the updated weights to the central entity for aggregation
- Does not satisfy DP automatically but can be made private by randomizing the local updates

Differential Privacy Summary

- A lot of ML models are trained on datasets containing sensitive information about individuals, and database reconstruction attacks can be surprisingly effective
- Differential privacy gives a way of provably preventing (much) information about individuals from leaking
- Building blocks of differential privacy
  - Laplace mechanism adds noise to the output
  - Composition rules combine multiple private queries
- Sometimes differentially private algorithms can accurately answer queries for large populations
  - The 2020 US Census used differential privacy

Section 3

Algorithmic Fairness

Motivation: Algorithms influence our lives in many ways

- Machine Learning based systems have been used (to automate complex decision) for:
  - Selecting job applicants
  - Recidivism prediction and predictive policing
  - Credit scoring and loans
  - Facial recognition
  - Search and recommendations
  - Machine Translation
  - ... and many other critical applications (involving humans)
- Unfortunately it has been repeatedly shown that these systems are (often significantly) biased

Biased algorithms influence our lives in many ways

- Selecting job applicants
  - XING ranks less qualified male candidates higher than more qualified female candidates (Lahoti et al. 2018)
- Recidivism prediction and predictive policing
  - COMPAS: high-risk FP: 23.5% for white vs. 44.9% for black; low-risk FP: 47.7% for white vs. 28.0% for black (ProPublica article)
- Facial recognition
  - Commercial software has much lower accuracy on females with darker color (Buolamwini and Gebru, 2018)
- Search and recommendations
  - Search queries for African-American names more likely to return ads suggestive of an arrest (Sweeney, 2019)
- Bias found in word embeddings
  - man-woman=surgeon-nurse (Bolukbasi et al. 2016)
- ...

What causes the bias?

- **Tainted training data**: Any ML system maintains (and amplifies) the existing bias in the data caused by human bias, e.g. hiring decisions made by a (biased) *manager* used as labels, historic and systematic biases in the data collection process, etc.
- **Skewed sample**: Initial predictions influence future observations, e.g. regions with initial high crime rate get more police attention (and thus higher recorded crime in the future), Selection bias
- **Proxies**: Even if we exclude legally protected features (e.g. race, gender, sexuality) other features may be highly correlated with these
- **Sample size disparity**: Models will tend to fit the larger groups first (possibly) trading off accuracy for minority groups
- **Limited features**: Features may be less informative or reliably collected for minority groups

Why Fairness is Hard

- How to define fairness?
  - How can we formulate it so it can be considered in ML systems?
- Two distinct notions from the law (Barocas and Selbst, 2016):
  - **Disparate treatment**: decisions are (partly) based on the subject's sensitive attribute
  - **Disparate impact**: disproportionately hurt (or benefit) people with certain sensitive attribute values
- Currently, no consensus on the mathematical formulations of fairness

An illustrating example

- We are a bank trying to fairly decide who should get a loan
  - i.e. predict which people will likely pay us back?
- We have two groups: Blue and Orange (the sensitive attribute)
  - This is where discrimination could occur

Figure: Simulating loan thresholds, research.google.com/bigpicture

Definitions of Fairness

- How can we test if our (loan repay) classifier is fair?

- The notions of **Group** fairness aim to treat all groups equally
  - e.g. We can require that the same percentage of Blue and Orange receive loans
  - or Require equal false positive/negative rates, e.g.

$$P(\text{no loan} \mid \text{would repay}, \text{Blue}) = P(\text{no loan} \mid \text{would repay}, \text{Orange})$$

- **Individual** notions of fairness (treat similar examples similarly) also exist but won't be covered in this lecture
- **Counterfactual** fairness uses tools from causal inference
  - Same decision in the actual world and a counterfactual world where the individual belonges to a different group

Setup – Group Fairness

Consider binary classification with single sensitive attribute for simplicity:

- $X \in \mathbb{R}^d$: features of an individual (e.g. credit history)
- $A \in \{a, b, \dots\}$: sensitive feature (gender, race, etc.)
- $R = r(X, A) \in \{0, 1\}$: binary predictor (e.g. whether to grant a loan or not) which makes a decision
  - thresholding a score $R = r(X, A) \in [0, 1]$, e.g. a NN classifier
- $Y \in \{0, 1\}$: the target variable representing the ground truth

- Assume $(X, A, Y) \sim \mathcal{D}$ are generated from an underlying distribution
- X, A, Y and R are thus random variables
- Notation: $P_a\{R\} = P\{R \mid A = a\}$

Naive Approach: Fairness through Unawareness

- We should not include the sensitive attribute as a feature in the training data
- $R = r(X)$ instead of $R = r(X, A)$

Pros/Cons:

- Intuitive, easy to use and implement
- Consistent with disparate treatment which has legal support (e.g. the "General Equal Treatment Act" in Germany)
- However, there can be many highly correlated features (e.g. neighborhood) that are proxies of the sensitive attribute (e.g. race)

First Criterion: Independence

- Require: R independent of A, denoted $R \perp\!\!\!\perp A$
  - Also called Demographic Parity, Statistical Parity, Group Fairness, Darlington criterion (4)
- In case of binary classification for all groups a, b:

$$P_a\{R = 1\} = P_b\{R = 1\}$$

- In our example, this means that the **acceptance rates** of the applicants from the two groups must be equal, i.e. same percentage of applications receive loans
- Approximate versions:

$$\frac{P_a\{R = 1\}}{P_b\{R = 1\}} \geq 1 - \epsilon \qquad |P_a\{R = 1\} - P_b\{R = 1\}| \leq \epsilon$$

How to achieve Independence?

- Post-processing
  - Adjust a learned classifier so as to be uncorrelated with the sensitive attribute
- Training time constraint
  - Include the exact/approximate constraints in the optimization
- Pre-processing: e.g. via representation learning (next slide)

Representation learning approach

- Map (X, A) to a representation Z (e.g. dimensionality reduction)
- Train the predictor on the representation: $R = r(Z)$
- How to learn a fair representation Z?
  - e.g. optimize for $\max I(X; Z)$ and $\min I(A; Z)$, where I is some measure of information (e.g. mutual information)
  - e.g. Fair PCA, Fair VAE

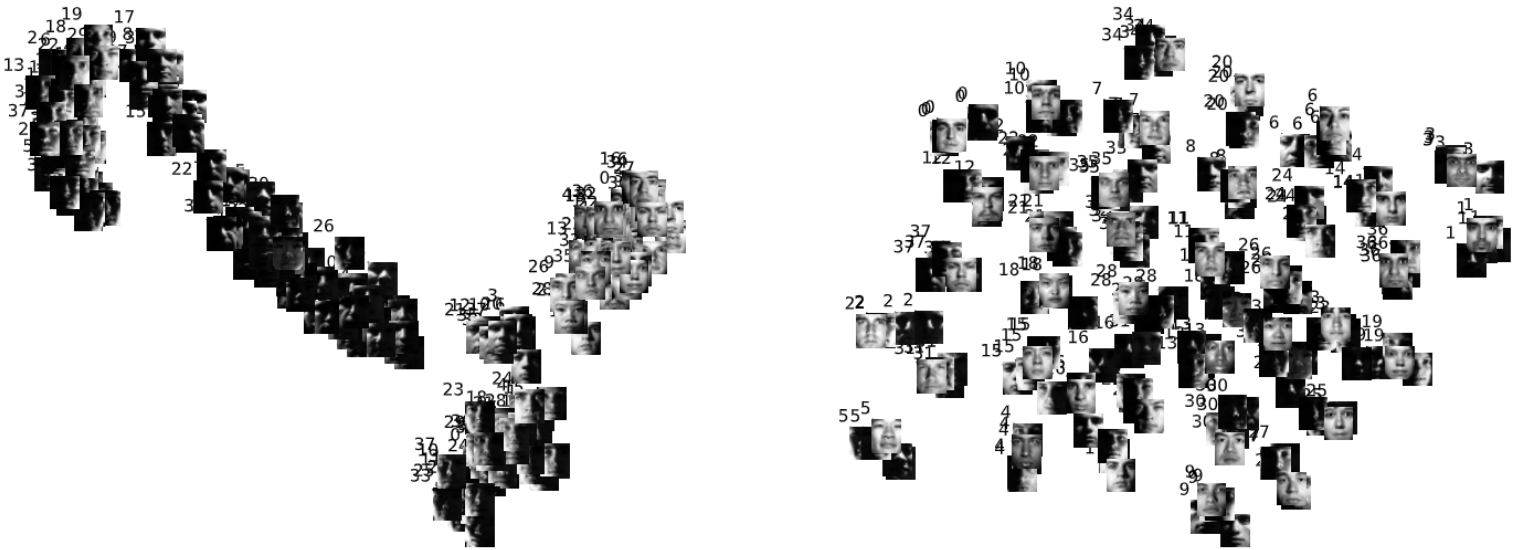


Figure: The Variational Fair Autoencoder (Louizos et al., 2016)

Pros/Cons of Independence

- Legal support: "**four-fifth rule**" prescribes that a selection rate for any disadvantaged group that is less than four-fifths of that for the group with the highest rate must be justified

- What if 83% of Blue is likely to repay, but only 43% of Orange is?
  - Then Independence is too strong
  - Rules out perfect predictor $R = Y$ when base rates are different

- Laziness: We can trivially satisfy the criterion if we give loan to qualified people from one group and random people from the other
  - Can even establish a negative track record for the second group

Second Criterion: Separation

- Require: The prediction R and A to be independent *conditional* on the target Y, denoted $R \perp\!\!\!\perp A \mid Y$
  - Also called Equalized Odds, Conditional procedure Accuracy, Avoiding disparate mistreatment,
- In case of binary classification for all groups a, b:

$$P_a(R = 1 \mid Y = 1) = P_b(R = 1 \mid Y = 1) \qquad \text{true positive (TP)}$$
$$P_a(R = 1 \mid Y = 0) = P_b(R = 1 \mid Y = 0) \qquad \text{false positive (FP)}$$

- **Equality of Opportunity** is a commonly used relaxation
  - Only match the TP rate: $P_a(R = 1 \mid Y = 1) = P_b(R = 1 \mid Y = 1)$
- In our example, this means we should give loan to equal proportion of individuals who would in reality repay

Achieving Separation

- Area under the ROC (Receiver Operating Characteristic) curve
- Each point on the solid curve(s) is realized by thresholding the predicted score at some value, i.e. predict $\mathbb{I}(r(X, A) > t)$
- Pick a classifier that minimizes the given cost (e.g. maximizes profit)

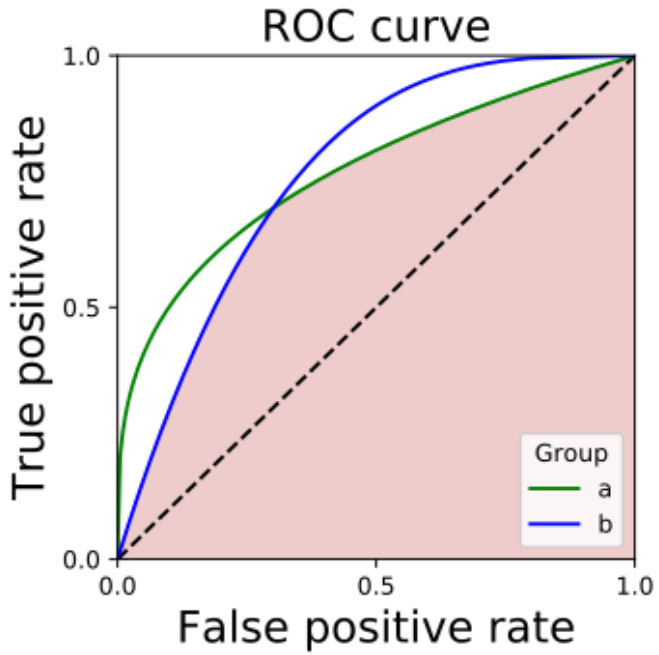


Figure: Intersection of area under the curves (`https://fairmlbook.org/`)

Pros/Cons of Separation

- Optimal predictor not ruled out: $R = Y$ is allowed
- Penalizes laziness: it provides incentive to reduce errors uniformly in all groups
- It may not help closing the gap between two groups
  - Granting more loans to the group that is more likely to repay **now** makes the groups more likely to have better living conditions and thus even more likely to repay in the **future**, thus widening the gap

Third Criterion: Sufficiency

- Require the target Y and A to be independent *conditional* on the prediction (or score) R, denoted $Y \perp\!\!\!\perp A \mid R$
  - Also called Cleary model, Conditional use accuracy, Calibration within groups
- In case of binary classification for all groups a, b and all $r \in [0, 1]$:

$$P_a(Y = 1 \mid R = r) = P_b(Y = 1 \mid R = r)$$

- In our example, the score used to determine if a candidate would repay should reflect the candidate's real/actual capability of repaying

Achieving Sufficiency

- In general a classifier R is calibrated if for all $r \in [0, 1]$:

$$P(Y = 1 \mid R = r) = r$$

- Of all instances assigned a score value r an r fraction of them should be positive

- Calibration for each group a implies sufficiency:

$$P_a(Y = 1 \mid R = r) = r$$

- Apply standard calibration techniques to each group (if necessary)

- Platt Scalling: given an uncalibrated score treat it as a single feature and fit a one variable regression model against Y

Pros/Cons of Sufficiency

- Satisfied by the Bayes optimal classifier

$$r(X, A) = \mathbb{E}[Y \mid X = x, A = a]$$

- For predicting Y do not need to see A when we have R
- Equal chance of success ($Y = 1$) given acceptance ($R = 1$)
- Similar to before it may not help closing the gap between the groups

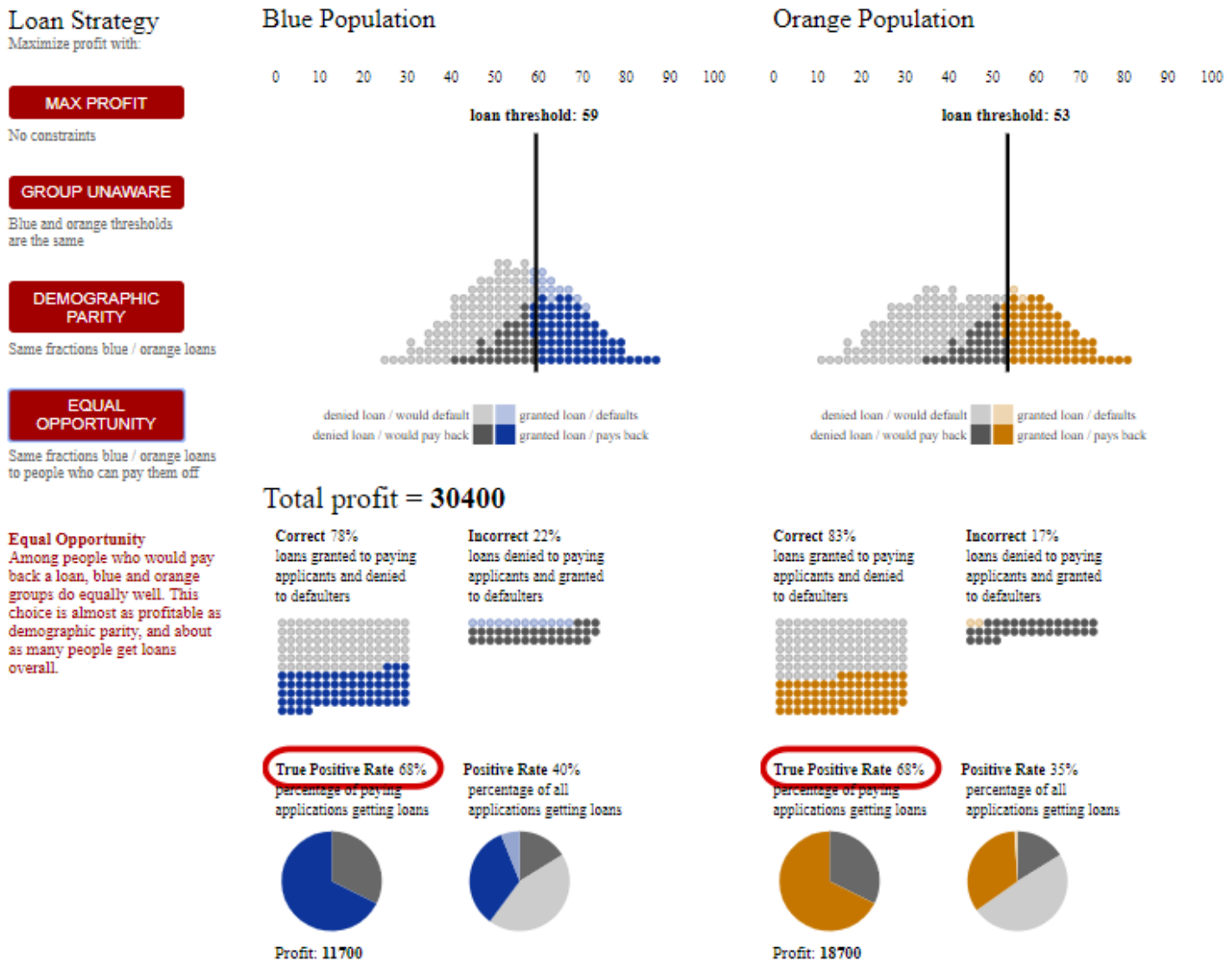
Fairness Summary: A growing list of fairness criteria

General theme: Require some invariance w.r.t. the sensitive attribute

- Independence: $R \perp\!\!\!\perp A$
- Separation: $R \perp\!\!\!\perp A \mid Y$
- Equality of Opportunity: $R \perp\!\!\!\perp A \mid Y = 1$
- Sufficiency: $Y \perp\!\!\!\perp A \mid R$
- Conditional statistical parity
- Predictive equality
- Predictive parity
- ... and many many more

Many of these definitions are (provably) incompatible, i.e. they are mutually exclusive except in degenerate cases

Visualizing the trade-offs: research.google.com/bigpicture



Comparing different criteria

- Profit for a TP and cost for a FP
  - The cost of FP is typically much greater than the profit for TP

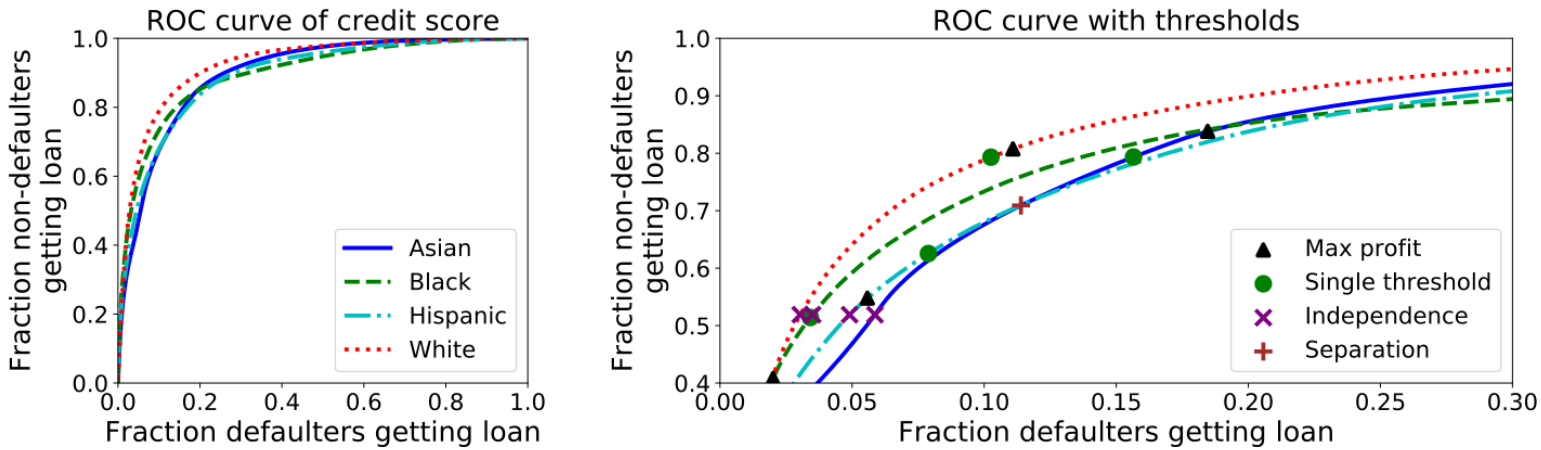


Figure: Different thresholds induced by different criteria (Hardt et al., 2016)

Adversarial Examples

Adversarial Examples are deliberate perturbations of the data designed to achieve a specific malicious goal (e.g. cause a misclassification)

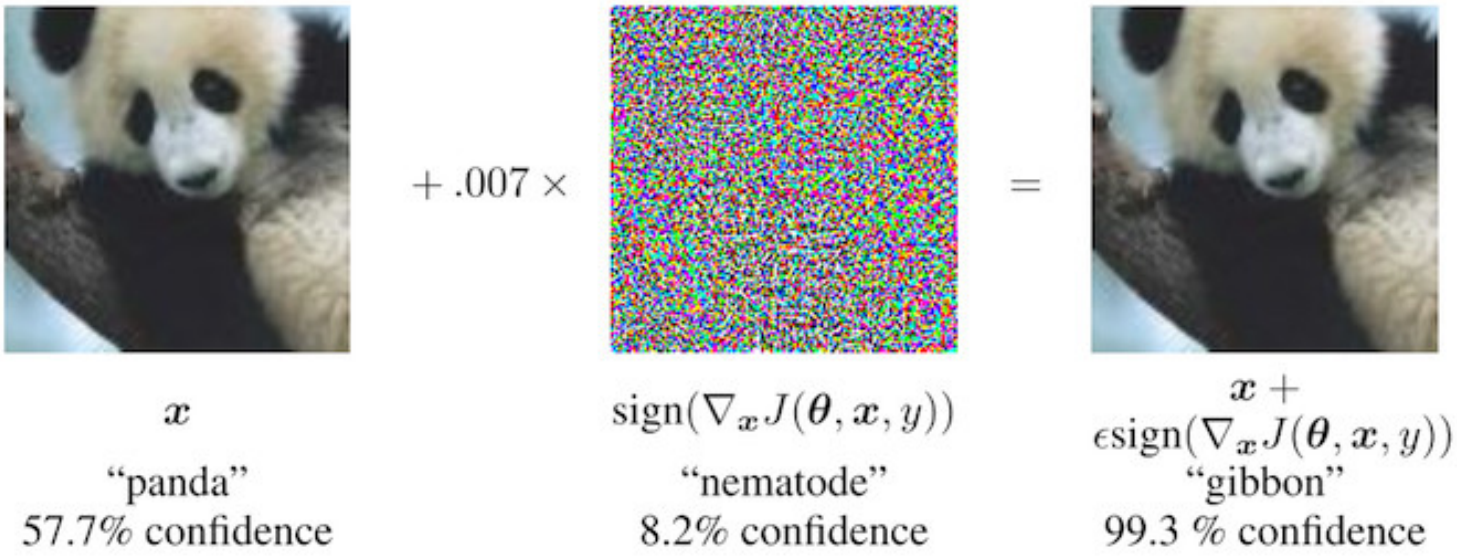


Figure: The panda is classified as a gibbon by the NN, Goodfellow et. al, 2014

Adversarial Examples

Figure: Adversarial glasses fool Facial Recognition systems into classifying the wearer as someone else, Sharif et al., 2016

Figure: ML systems classify the adversarially modified Stop sign as a Speed Limit sign, Eykholt et al., 2018

Adversarial Examples

- Many other recent studies show that most ML models are vulnerable to adversarial examples
  - On a high level this is because ML model do not really generalize
  - If the distribution of the test data is even slightly different from the distribution of the training data they fail miserably

- How can we create, detect and defend against Adversarial Examples?
- Especially important to quantify this risk if we are in a safety-critical application, e.g. self-driving cars
- **Certifiable robustness** provides mathematical guarantees

- Nature as an adversary: Even if there is no adversary in our use-case, we should quantify **robustness to worst-case noise**

Summary

- Decision based on data are not always accurate, reliable, or fair
- **Differential Privacy** allows us to compute arbitrary queries on (sensitive) data with provable guarantees on information leakage
  - There are no absolute privacy guarantees, your neighbor's habits are correlated with your habits
- **Algorithmic Fairness** criterions require (and enforce) some invariances w.r.t. sensitive attributes
  - Algorithmic Fairness $\neq$ Actual Fairness, social/legal/political effort also needed
  - Without a model of long-term impact it is difficult to foresee the effect of a fairness criterion implemented as a constraint
- Accuracy, Fairness, Privacy, Adversarial Robustness, Explainability and other aspects are non-trivially related
- Algorithmic solutions are only (small) part of the puzzle

Reading material

Main reading

- "The Algorithmic Foundations of Differential Privacy" by Dwork and Roth
  [ch. 2, 3.1-3.5],
  `https://www.cis.upenn.edu/~aaroth/privacybook.html`
- "Fairness and Machine Learning" by Barocas, Hardt, and Narayanan
  [ch. 1, 2], `https://fairmlbook.org/`[6]

[6] Part of the slides adapted from the CSC 2515 lecture by Roger Grosse and the Differential Privacy Tutorial by Borja Balle