Statistical Machine Learning

https://cvml.ist.ac.at/courses/SML_W20

Christoph Lampert

Institute of Science and Technology

Fall Semester 2020/2021

Lecture 6

Overview (tentative)

Date		no.	Topic
Oct 05	Mon	1	A Hands-On Introduction
Oct 07	Wed	2	Bayesian Decision Theory, Generative Probabilistic Models
Oct 12	Mon	3	Discriminative Probabilistic Models
Oct 14	Wed	4	Maximum Margin Classifiers, Generalized Linear Models
Oct 19	Mon	5	Estimators; Overfitting/Underfitting, Regularization, Model Selection
Oct 21	Wed	6	Bias/Fairness, Domain Adaptation
Oct 26	Mon	-	no lecture (public holiday)
Oct 28	Wed	7	Learning Theory I
Nov 02	Mon	8	Learning Theory II
Nov 04	Wed	9	Deep Learning I
Nov 09	Mon	10	Deep Learning II
Nov 11	Wed	11	Unsupervised Learning
Nov 16	Mon	12	project presentations
Nov 18	Wed	13	buffer

Bias and Fairness

Machine Learning has started to influence our everyday lives

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.

translate.google.com/translate?hl=&sl=de&tl=en&u...

https://amsalgorithmus.at/

Translate From: German To: English View: Translation Original

Stops the AMS algorithm

Computers are not allowed to make decisions about people!

So far 3403 of 5000 signatures.

Support demands now!

Explanatory video | requirements | Contact the minister | Request AMS data | Spread the word

Machine Learning has started to influence our everyday lives

Example from the USA: Recidivism Scoring

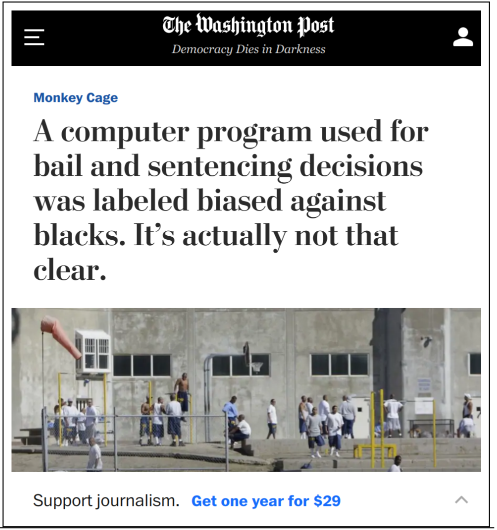
The commercial software tool COMPAS is used by U.S. courts to predict the probability that a defendent in court will will commit a new crime at a later time.

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.

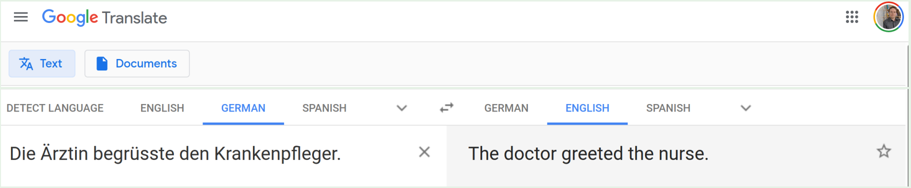
- in 2018, *Google image search* for "CEO" returned almost exclusively pictures of men
- face recognition datasets contain predominanlty white faces
  $\overset{?}{\rightarrow}$ in 2015 Google's image tagger labeled some pictures of black faces as "gorilla"
- Google translate tends to make all "doctors"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

Bias and Fairness in Machine Learning

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

Data sources should not have a bias.

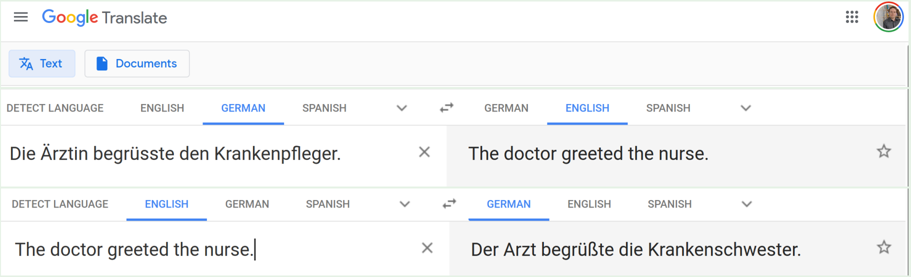
- in 2018, *Google image search* for "CEO" returned almost exclusively pictures of men
- face recognition datasets contain predominanlty white faces
  $\overset{?}{\longrightarrow}$ in 2015 Google's image tagger labeled some pictures of black faces as "gorilla"
- Google translate tends to make all "doctors"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



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.

Imagene 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 or not 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: Gradschool Recruiting

Current process at IST Austria:

1. candidates prepare their applications and upload them
2. references prepare their recommendation letters and upload them
3. volunteers (typically postdocs) pre-filter applications, removing ≈50% "hopeless cases"
4. faculty members read remaining applications and assign quality scores
5. faculty members discuss and decide whom they want to interview
6. faculty members interview invited candidates
7. faculty members score candidates as: definite offer, possible offer, reject
8. faculty members discuss each applicant and decide on offer or rejection
9. accepted candidates decide to accept or reject

Every single step is influenced by (explicit or subconcious) bias.

How can we ensure a (more) fair process?

Example: Student 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 some years, evaluate accuracy or ROC-curve on other years

Example: Student 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 some years, evaluate accuracy or ROC-curve on other years

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 the test data is as biased as the training data

Example: Student 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 some years, evaluate accuracy or ROC-curve on other years

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 the test data is as biased as the training data

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

(Group) Fairness in the Language of Probability

Notation: random variables

- X, taking values $x \in \mathcal{X}$: input
- A, taking values $a \in \mathcal{A}$: sensitive attributes of X
- 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 candidates 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

Fairness Through Unawareness

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

Fairness Through Unawareness

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

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

- first 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

Fairness Through Unawareness

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

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

- first 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.

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 contraits how the predictor output should behave

---

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

Formal Fairness Criteria: Independence

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.

Formal Fairness Criteria: Independence

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? At least three options:

- Pre-processing: extract features in which no information about A remains
  - \+ broadly applicable: needs only the raw data, afterwards any classifier can be trained by anyone
  - − classifier quality might suffer, more information than necessary is discarded

  $\rightarrow$ special case of second part of today's lecture

- 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
  - \+ applicable for black-box/pretrained classifiers, efficient
  - − classifier quality might suffer, more information than necessary can get lost

Example 1: training with *independence* constraints

Regularized Risk Minimization with Fairness Constraints:

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

Example 1: training with *independence* constraints

Regularized Risk Minimization with Fairness Constraints:

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

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

$$F(\theta) = \sum_{a,b \in \mathcal{A}} \Big(\frac{1}{|\mathcal{D}_a|} \sum_{(x,y) \in \mathcal{D}_a} f(x) \; - \; \frac{1}{|\mathcal{D}_b|} \sum_{(x,y) \in \mathcal{D}_b} f(x) \Big)^2$$

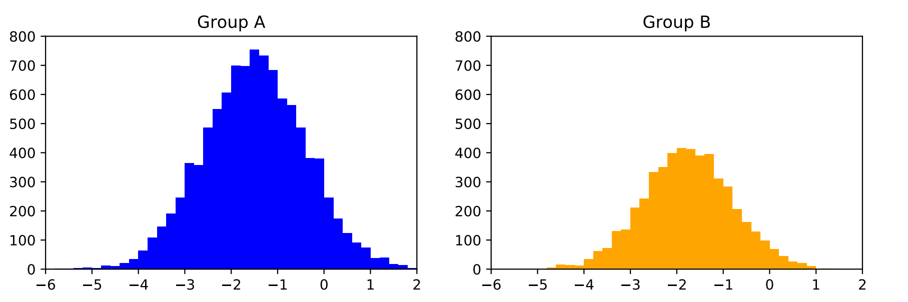
where $\mathcal{D}_a = \{(x, y) \in \mathcal{D} : x_A = a\}$ for any $a \in \mathcal{A}$.

Note: we can do this on the level of decisions, $f(x) \in \{0, 1\}$, or confidences, $f(x) \in \mathbb{R}$.

Example 2: *independence* by postprocessing

Group-specific threshold selection

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

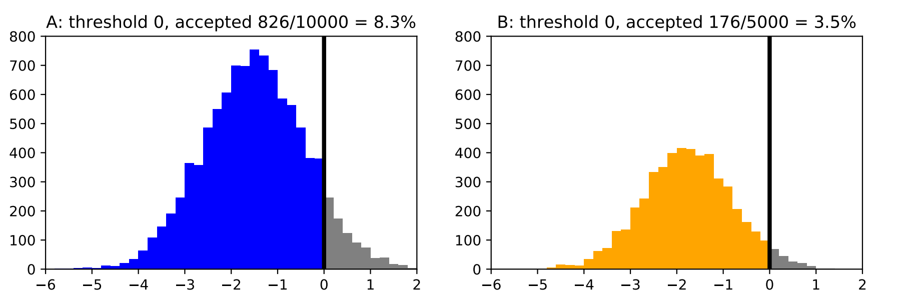


original confidence scores per group

Example 2: *independence* by postprocessing

Group-specific threshold selection

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

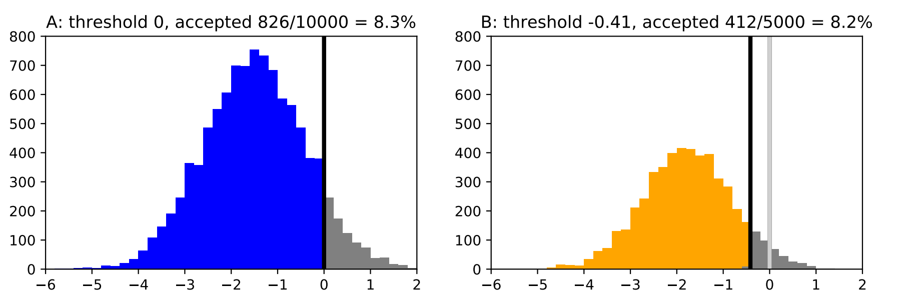


with equal thresholds, independence is violated

Example 2: *independence* by postprocessing

Group-specific threshold selection

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

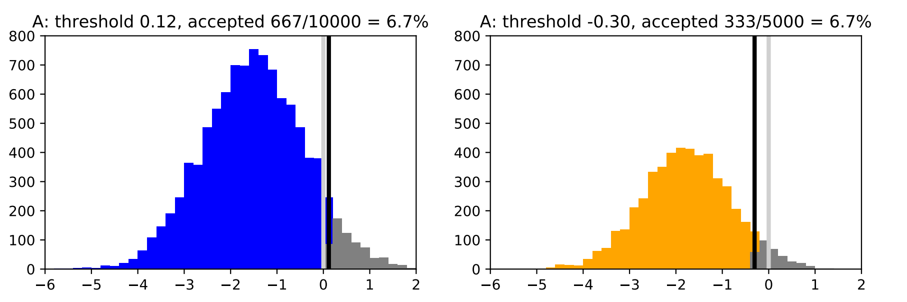


lower threshold for group B achieves independence, but overall acceptance rate now too high

Example 2: *independence* by postprocessing

Group-specific threshold selection

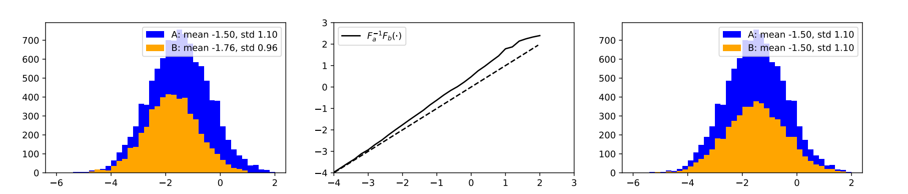
Adjust the acceptance threshold for each group to achieve equal acceptance rate:



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!

Group-specific score transformations



Apply group-specific post-processing operation to scores, e.g.

- denote by $F_g(\cdot)$ the cumulative distribution function of scores for group $A = g$
- for all examples with $A = b$ apply the score transformation

$$\phi_{b \to a}(\cdot) := F_a^{-1}(F_b(\cdot))$$

- afterwards, both groups will have (approximately) the same score distribution
  $\rightarrow$ the same thresholds can be used for both groups

Formal Fairness Criteria: Independence – Problems

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 in judging the cases)

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

Even worse: in the long run, accepted students with $A = a$ will probably do better on average than those with $A = b$, potentially reinforcing stereotypes and providing arguments to opponents of "fair recruiting".

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 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 $\to$ 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 $\to$ 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 $\to$ 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 $a, b \in \mathcal{A}$.

$$\Pr\{R = 1 \mid Y = 1, A = a\} = \Pr\{R = 1 \mid Y = 1, A = b\}$$
$$\Pr\{R = 1 \mid Y = 0, A = a\} = \Pr\{R = 1 \mid Y = 0, A = b\}$$

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.

Formal Fairness Criteria: Separation – Properties

Property 1) Separation allows making perfect decisions.

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

Formal Fairness Criteria: Separation – Properties

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) Is 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 papers from each continent (TPR=p)
  - reject all others (FPR=0)
- quality determines the chance of acceptance, not the author origin

Formal Fairness Criteria: Independence – Problems

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

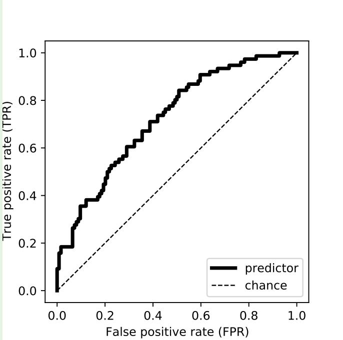
- identify all suitable candidates $(Y = 1)$:
  - BSc in engineering, physics, computer science, or math
  - at least 3 years professional flight test experience or 1000 hours as aircraft pilot
  - 20/20 vision, blood pressure not exceeding 140/90
  - between 157cm and 190cm tall

  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

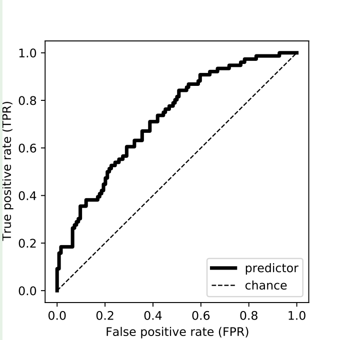
Can we achieve separation by post-processing the scores without retraining?



Separation by group-specific thresholds

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

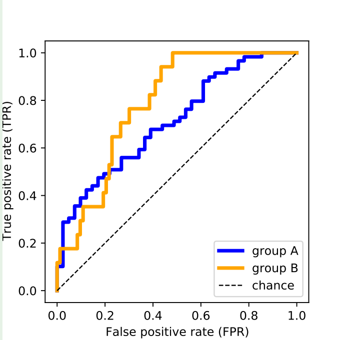
- ROC curve: FPR/TPR for all possible thresholds



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



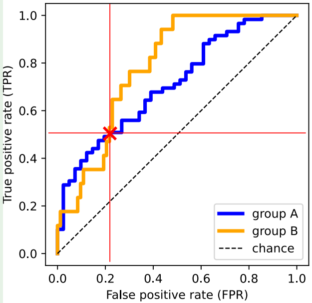
Separation by group-specific thresholds

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?

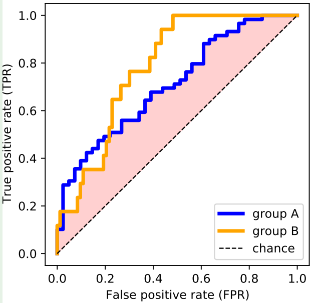
- ROC curve: FPR/TPR for all possible thresholds
- per-groups thresholds → FPR/TPR adjustable per group

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:

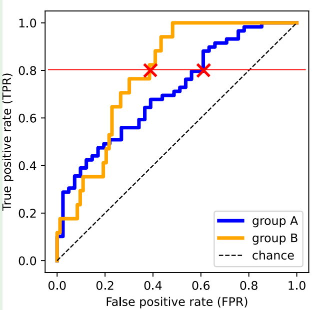
- equal FPR and TPR between groups only where curves intersect → might no *nowhere*
- typically not the desired operating points

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, ...

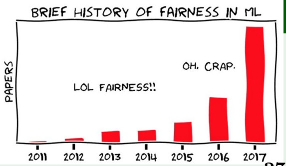
- Psychology etc.: what do people consider fair in which situation?
- ML/Stats: many different (usually mutually exclusive) formal definition of fairness

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



Domain Adaptation

Beyond i.i.d. data

The main assumption underlying machine learning is that

- at prediction time, data comes distributed according to (some unknown) $p(x, y)$
- the training set contains i.i.d. samples from $d(x, y)$

In practice, this is very often not true:

- class bias: some classes are over-/underrepresented
- domain shift: classes have different distribution at training vs prediction time
- label noise: some labels in the training data are (randomly?) flipped
- dependent data: training data is not independent, e.g. a time series
- ...

There's many possible reasons for that:

- data collection: data is collected by people, who are human
- annotation issues: labels are provided by people, who are human
- real-world vs simulation: simulated data is much easier to obtain than real world data
- ...

Notation:

- situation at training time: "source", abbreviated S
- situation at prediction time: "target", abbreviated T
- training data: $\mathcal{D}_{\mathsf{S}} \overset{i.i.d.}{\sim} p_{\mathsf{S}}(x, y)$
- goal: find a predictor $f : \mathcal{X} \to \mathcal{Y}$ with small target risk, $\mathcal{R}(f) = \mathbb{E}_{(x,y)\sim p_{\mathsf{T}}} \ell(y, f(x))$

Domain Adaptation

Domain adaptation research studies the question if and how learning is possible when

$$p_{\mathsf{S}}(x, y) \neq p_{\mathsf{T}}(x, y).$$

Domain Adaptation Scenarios

There's at least three different scenarios for $p_\mathsf{S}(x,y) \neq p_\mathsf{T}(x,y)$ that allow different treatment:

- **prior shift**: write $p_\mathsf{S}(x,y) = p_\mathsf{S}(x|y)p_\mathsf{S}(y)$ and $p_\mathsf{T}(x,y) = p_\mathsf{T}(x|y)p_\mathsf{T}(y)$:

$$p_\mathsf{S}(y) \neq p_\mathsf{T}(y) \qquad \text{and} \qquad p_\mathsf{S}(x|y) = p_\mathsf{T}(x|y)$$

- **covariate shift**: write $p_\mathsf{S}(x,y) = p_\mathsf{S}(y|x)p_\mathsf{S}(x)$ and $p_\mathsf{T}(x,y) = p_\mathsf{T}(y|x)p_\mathsf{T}(x)$:

$$p_\mathsf{S}(x) \neq p_\mathsf{T}(x) \qquad \text{and} \qquad p_\mathsf{S}(y|x) = p_\mathsf{T}(y|x)$$

- **arbitrary shift**: anything else

Tackling class prior shift

Can we derive an estimator of the target risk if only source data is available?

$$
\begin{aligned}
\mathcal{R}_{\mathsf{T}}(f) = \mathop{\mathbb{E}}_{(x,y)\sim p_{\mathsf{T}}} \ell(y, f(x)) &= \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p_{\mathsf{T}}(x,y)\ell(y,f(x)) \\
&= \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} \frac{p_{\mathsf{T}}(x,y)}{p_{\mathsf{S}}(x,y)} p_{\mathsf{S}}(x,y)\ell(y,f(x)) \\
&= \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} \frac{\cancel{p_{\mathsf{T}}(x|y)}p_{\mathsf{T}}(y)}{\cancel{p_{\mathsf{S}}(x|y)}p_{\mathsf{S}}(y)} p_{\mathsf{S}}(x,y)\ell(y,f(x)) \\
&= \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} \underbrace{\frac{p_{\mathsf{T}}(y)}{p_{\mathsf{S}}(y)}}_{=:w(y)} p_{\mathsf{S}}(x,y)\ell(y,f(x)) \\
&= \mathop{\mathbb{E}}_{(x,y)\sim p_{\mathsf{S}}} \Big[w(y)\ell(y,f(x)) \Big]
\end{aligned}
$$

$$
\rightarrow \quad \hat{\mathcal{R}}_{\mathsf{T}}(f) = \frac{1}{|\mathcal{D}_{\mathsf{S}}|} \sum_{(x,y)\in\mathcal{D}_{\mathsf{S}}} w(y)\ell(y,f(x))
$$

How to get the class weights?

$$w(y) = \frac{p_{\mathsf{T}}(y)}{p_{\mathsf{S}}(y)}$$

Observation:

- $w \in \mathbb{R}^{|\mathcal{Y}|}$, vector of ratio of probabilities
- need either prior knowledge, or data from $p_{\mathsf{T}}(y)$

Method 1:

- estimate $\hat{p}_{\mathsf{S}}(y)$ and $\hat{p}_{\mathsf{T}}(y)$ from data
  - see lecture about generative models: empirical frequencies, Laplace smoothing, etc.
- set $\hat{w}(y) = \frac{\hat{p}_{\mathsf{T}}(y)}{\hat{p}_{\mathsf{S}}(y)}$ "plug-in estimator"

Note:

- $\hat{w}(y)$ is *not* an unbiased estimator of $w(y)$ (because $\mathbb{E}\,\frac{A}{B} \neq \frac{\mathbb{E}\,A}{\mathbb{E}\,B}$)
- the bias is of order $\frac{1}{n}$, so it vanishes for $n \to \infty$

$$\begin{aligned}\mathcal{R}_{\mathsf{T}}(f) = \mathop{\mathbb{E}}_{(x,y)\sim p_{\mathsf{T}}} \ell(y, f(x)) &= \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p_{\mathsf{T}}(x,y)\ell(y,f(x)) \\ &= \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} \frac{p_{\mathsf{T}}(x,y)}{p_{\mathsf{S}}(x,y)} p_{\mathsf{S}}(x,y)\ell(y,f(x)) \\ &= \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} \frac{\cancel{p_{\mathsf{T}}(y|x)}p_{\mathsf{T}}(x)}{\cancel{p_{\mathsf{S}}(y|x)}p_{\mathsf{S}}(x)} p_{\mathsf{S}}(x,y)\ell(y,f(x)) \\ &= \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} \underbrace{\frac{p_{\mathsf{T}}(x)}{p_{\mathsf{S}}(x)}}_{=:w(x)} p_{\mathsf{S}}(x,y)\ell(y,f(x)) \\ &= \mathop{\mathbb{E}}_{(x,y)\sim p_{\mathsf{S}}} \Big[w(x)\ell(y,f(x))\Big]\end{aligned}$$

$$\rightarrow \quad \hat{\mathcal{R}}_{\mathsf{T}}(f) = \frac{1}{|\mathcal{D}_{\mathsf{S}}|} \sum_{(x,y)\in\mathcal{D}_{\mathsf{S}}} w(x)\ell(y,f(x))$$

How to get the data weights?

$$w(x) = \frac{p_\mathsf{T}(x)}{p_\mathsf{S}(x)}$$

Observation:

- weights are a function, $w : \mathcal{X} \to \mathbb{R}_+$, but values at $w(x_1), \dots, w(x_n)$ for $x_i \in \mathcal{D}_\mathsf{S}$ suffice
- need either prior knowledge, or data from $p_\mathsf{T}(x)$ $\leftarrow$ unlabeled data suffices!

Method 1:

- estimate $\hat{p}_\mathsf{S}(x)$ and $\hat{p}_\mathsf{T}(x)$ from data
  - see lecture about generative models: Parzen window, Gaussian Mixture Model, etc.
  - needs a lot of samples if data is high-dimensional
- set $\hat{w}(x) = \frac{\hat{p}_\mathsf{T}(x)}{\hat{p}_\mathsf{S}(x)}$ "plug-in estimator", not unbiased

Method 2:

- estimate directly...

Density Ratio Estimation by Logistic Regression

Available data $\mathcal{D}_\mathsf{S} = \{x_1, \dots, x_n\}$, $\mathcal{D}_\mathsf{T} = \{x'_1, \dots, x'_m\}$. Define auxiliary distribution q:

- introduce indicator variable $z = \{\mathsf{src}, \mathsf{tgt}\}$ with $q(z = \mathsf{src}) = q(z = \mathsf{tgt}) = \frac{1}{2}$.
- set $q(x|z = \mathsf{src}) = p_\mathsf{S}(x)$ and $q(x|z = \mathsf{tgt}) = p_\mathsf{T}(x)$

$$w(x) = \frac{p_\mathsf{T}(x)}{p_\mathsf{S}(x)} = \frac{q(x|z = \mathsf{tgt})}{q(x|z = \mathsf{src})} \overset{\text{Bayes rule}}{=} \frac{q(z = \mathsf{tgt}|x)q(x)q(z = \mathsf{src})}{q(z = \mathsf{src}|x)q(x)q(z = \mathsf{tgt})} = \frac{q(z = \mathsf{tgt}|x)}{q(z = \mathsf{src}|x)}$$

Density Ratio Estimation by Logistic Regression

Available data $\mathcal{D}_\mathsf{S} = \{x_1, \dots, x_n\}$, $\mathcal{D}_\mathsf{T} = \{x'_1, \dots, x'_m\}$. Define auxiliary distribution q:

- introduce indicator variable $z = \{\mathsf{src}, \mathsf{tgt}\}$ with $q(z = \mathsf{src}) = q(z = \mathsf{tgt}) = \frac{1}{2}$.
- set $q(x|z = \mathsf{src}) = p_\mathsf{S}(x)$ and $q(x|z = \mathsf{tgt}) = p_\mathsf{T}(x)$

$$w(x) = \frac{p_\mathsf{T}(x)}{p_\mathsf{S}(x)} = \frac{q(x|z = \mathsf{tgt})}{q(x|z = \mathsf{src})} \overset{\text{Bayes rule}}{=} \frac{q(z = \mathsf{tgt}|x)q(x)q(z = \mathsf{src})}{q(z = \mathsf{src}|x)q(x)q(z = \mathsf{tgt})} = \frac{q(z = \mathsf{tgt}|x)}{q(z = \mathsf{src}|x)}$$

Idea: train a discriminative probabilistic model, e.g. Logistic Regression:

$$\hat{q}(z = \mathsf{tgt}|x) = \frac{\exp\langle\theta, \phi(x)\rangle}{1 + \exp\langle\theta, \phi(x)\rangle} \qquad \hat{q}(z = \mathsf{src}|x) = \frac{1}{1 + \exp\langle\theta, \phi(x)\rangle}$$

to distinguish between classes tgt and src, i.e. between $\mathcal{D}_\mathsf{T}$ and $\mathcal{D}_\mathsf{S}$.

$$\hat{w}(x) = \frac{\exp\langle\theta, \phi(x)\rangle}{\frac{1}{n}\sum_{x\in\mathcal{D}_\mathsf{S}} \exp\langle\theta, \phi(x)\rangle} \qquad \text{where numerator ensures } \frac{1}{n}\sum_{x\in\mathcal{D}_\mathsf{S}} \hat{w}(x) = 1$$

Kullback-Leibler Importance Estimation Procedure (KLIEP)

Available data $\mathcal{D}_{\mathsf{S}} = \{x_1, \dots, x_n\}$, $\mathcal{D}_{\mathsf{T}} = \{x'_1, \dots, x'_m\}$.

Parameterize $\hat{w}$ as a generalized log-linear model with suitable normalization as

$$\hat{w}(x) = \frac{\exp(\langle \theta, \phi(x) \rangle)}{\frac{1}{n}\sum_{x \in \mathcal{D}_{\mathsf{S}}} \exp(\langle \theta, \phi(x_i) \rangle)}$$

Idea: Find $\hat{w}(x)$ by minimizing KL-Divergence between $p_{\mathsf{T}}(x)$ and $\tilde{p}_{\mathsf{T}}(x) = \hat{w}(x) p_{\mathsf{S}}(x)$.

$$\mathrm{KL}(p_{\mathsf{T}} | \tilde{p}_{\mathsf{T}}) = \mathop{\mathbb{E}}_{x \sim p_{\mathsf{T}}(x)} \log \frac{p_{\mathsf{T}}(x)}{\tilde{p}_{\mathsf{T}}(x)} = \underbrace{\mathop{\mathbb{E}}_{x \sim p_{\mathsf{T}}(x)} \log \frac{p_{\mathsf{T}}(x)}{p_{\mathsf{S}}(x)}}_{\text{independent of } \hat{w}} - \mathop{\mathbb{E}}_{x \sim p_{\mathsf{T}}(x)} \log \hat{w}(x)$$

Minimizing KL w.r.t. $\hat{w}$ is equivalent to maximizing $\mathop{\mathbb{E}}_{x \sim p_{\mathsf{T}}(x)} \log \hat{w}(x) \approx \frac{1}{m} \sum_{x \in \mathcal{D}_{\mathsf{T}}} \log \hat{w}(x)$

The resulting optimization problem is convex and unconstrained $\rightarrow$ solve via gradient descent.

$$\begin{aligned}
\mathcal{R}_{\mathsf{T}}(f) = \mathop{\mathbb{E}}_{(x,y)\sim p_{\mathsf{T}}} \ell(y, f(x)) &= \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p_{\mathsf{T}}(x,y)\ell(y, f(x)) \\
&= \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} \frac{p_{\mathsf{T}}(x,y)}{p_{\mathsf{S}}(x,y)} p_{\mathsf{S}}(x,y)\ell(y, f(x)) \\
&= \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} \underbrace{\frac{p_{\mathsf{T}}(x,y)}{p_{\mathsf{S}}(x,y)}}_{=:w(x,y)} p_{\mathsf{S}}(x,y)\ell(y, f(x)) \\
&= \mathop{\mathbb{E}}_{(x,y)\sim p_{\mathsf{S}}} \Big[w(x,y)\ell(y, f(x)) \Big]
\end{aligned}$$

$$\rightarrow \quad \hat{\mathcal{R}}_{\mathsf{T}}(f) = \frac{1}{|\mathcal{D}_{\mathsf{S}}|} \sum_{(x,y)\in\mathcal{D}_{\mathsf{S}}} w(x,y)\ell(y, f(x))$$

Estimating the weights

Method 1:

- estimate $\hat{p}_{\mathsf{S}}(x,y)$ and $\hat{p}_{\mathsf{T}}(x,y)$ from data
- set $\hat{w}(x,y) = \frac{\hat{p}_{\mathsf{S}}(x,y)}{\hat{p}_{\mathsf{T}}(x,y)}$ "plug-in estimator", not unbiased
- note: a good estimate of $\hat{p}_{\mathsf{T}}(x,y)$ we need a lot of data. If we have that, why not simply use it as a training set?

Method 2:

- direct estimation as for $w(x)$, still needs sufficiently much labeled data from p_{T}

Method 3:

- use a "supervised domain adaptation" method instead of the weighted estimator

Supervised Domain Adaptation by Feature Augmentation

Available data: $\mathcal{D}_{\mathsf{S}} = \{(x_1, y_1), \dots, (x_n, y_n)\}$, $\mathcal{D}_{\mathsf{T}} = \{(x'_1, y'_1) \dots, (x'_m, y'_m)\}$, with $n \gg m$.

Idea: learn a predictor for from a newly constructed training set with transformed features:

$$\tilde{\mathcal{D}} = \{(\tilde{x}_1, y_1), \dots, (\tilde{x}_n, y_n), (\tilde{x}'_1, y'_1), \dots, (\tilde{x}'_m, y'_m)\}$$

where

$$\tilde{x}_i = \Big(\underbrace{x_i}_{\in\mathbb{R}^d}, \underbrace{x_i}_{\in\mathbb{R}^d}, \underbrace{0, \cdots, 0}_{d \text{ times}} \Big) \qquad \tilde{x}'_i = \Big(\underbrace{x_i}_{\in\mathbb{R}^d}, \underbrace{0, \cdots, 0}_{d \text{ times}}, \underbrace{x_i}_{\in\mathbb{R}^d} \Big)$$

- any original feature is made available twice: once as part of a shared feature space and once as part of a domain specific feature space.
- for data with consistent labeling between src and tgt, the classifier can use the shared part of the feature space
- for data with inconsistent labeling between src and tgt, the classifier can use the domain-specific part of the feature space

[Hal Daumé III, "Frustratingly Easy Domain Adaptation", ACL 2007]

Supervised Domain Adaptation by Feature Extraction or Fine-Tuning

Available data: $\mathcal{D}_{\mathsf{S}} = \{(x_1, y_1), \dots, (x_n, y_n)\}$, $\mathcal{D}_{\mathsf{T}} = \{(x'_1, y'_1) \dots, (x'_m, y'_m)\}$, with $n \gg m$.

Particularly popular with deep neural networks, which have the form $f(x) = \langle w, \phi(x) \rangle$, where the feature function $\phi : \mathcal{X} \to \mathbb{R}^D$ and the weight vector $w \in \mathbb{R}^D$ are both learned.

Idea:

- use $\mathcal{D}_{\mathsf{S}}$ a source predictor $f_{\phi,w}$; use $\mathcal{D}_{\mathsf{T}}$ to learn a target predictor, but stay close to $f_{\phi,w}$

Example: deep features

- learn target weight vector w_{T} but keep the feature mapping fixed, $\phi_{\mathsf{T}} = \phi_{\mathsf{S}}$,
  $\rightarrow$ much fewer parameters than for learning both, less data suffices

Example: finetuning

- learn target feature mapping ϕ_{T} and target weight vector w_{T}, but initialize learning at source values and regularize to stay close (e.g. early stopping): $w_{\mathsf{T}} \approx w_{\mathsf{S}}$ and $\phi_{\mathsf{T}} \approx \phi_{\mathsf{S}}$
  $\rightarrow$ strong regularization prevents overfitting, less data required

If no labeled target data is available, one has to use **unsupervised domain adaptation**.

Idea:

- find transformations $\phi_\mathsf{S} : p_\mathsf{S} \to \tilde{p}_\mathsf{S}$ and $\phi_\mathsf{T} : p_\mathsf{T} \to \tilde{p}_\mathsf{T}$, such that $\tilde{p}_\mathsf{S} \approx \tilde{p}_\mathsf{T}$
- apply ϕ_S to $\mathcal{D}_\mathsf{S}$ to get a labeled dataset $\tilde{\mathcal{D}}_\mathsf{S} \sim \tilde{p}_\mathsf{S}$
- learn a predictor, f, from $\tilde{\mathcal{D}}_\mathsf{S}$
- use $f \circ \phi_\mathsf{T}$ as predictor on new data $x \sim p_\mathsf{T}$, where $(f \circ \phi_\mathsf{T})(x) := f(\tilde{x})$ for $\tilde{x} = \phi_\mathsf{T}(x)$
- since $\tilde{x} = \phi_\mathsf{T}(x) \sim \tilde{p}_\mathsf{T} \approx \tilde{p}_\mathsf{S}$, the predictor f can be expected to work well

Example (exercise sheet): src $=$ {left eyes}, tgt $=$ {right eyes}.
Flipping images horizontally turns one into the other.

Generally, we don't have such prior knowledge and want to learn the transformations.

Caveat: make sure to avoid trivial solutions, such as $\phi_\mathsf{S}(x) = 0$, $\phi_\mathsf{T}(x) = 0$.

Available data: $\mathcal{D}_{\mathsf{S}} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ (labeled), $\mathcal{D}_{\mathsf{T}} = \{x'_1, \ldots, x'_m\}$ (unlabeled).

For any $\phi_{\mathsf{T}} : \mathcal{X} \to \mathbb{R}^D$, $\phi_{\mathsf{S}} : \mathcal{X} \to \mathbb{R}^D$, define transformed datasets

$$\tilde{\mathcal{D}}_{\mathsf{S}}^{\phi_{\mathsf{S}}} = \{\tilde{x}_1, \ldots, \tilde{x}_n\} \text{ with } \tilde{x}_j = \phi_{\mathsf{S}}(x_j) \quad \text{and} \quad \tilde{\mathcal{D}}_{\mathsf{T}}^{\phi_{\mathsf{T}}} = \{\tilde{x}'_1 \ldots, \tilde{x}'_m\} \text{ with } \tilde{x}'_j = \phi_{\mathsf{T}}(x'_j)$$

How to check if $\tilde{\mathcal{D}}_{\mathsf{S}}$ and $\tilde{\mathcal{D}}_{\mathsf{T}}$ have the same distribution?

Unsupervised domain adaptation

Available data: $\mathcal{D}_\mathsf{S} = \{(x_1, y_1), \dots, (x_n, y_n)\}$ (labeled), $\mathcal{D}_\mathsf{T} = \{x'_1, \dots, x'_m\}$ (unlabeled).

For any $\phi_\mathsf{T} : \mathcal{X} \to \mathbb{R}^D$, $\phi_\mathsf{S} : \mathcal{X} \to \mathbb{R}^D$, define transformed datasets

$$\tilde{\mathcal{D}}_\mathsf{S}^{\phi_\mathsf{S}} = \{\tilde{x}_1, \dots, \tilde{x}_n\} \text{ with } \tilde{x}_j = \phi_\mathsf{S}(x_j) \quad \text{and} \quad \tilde{\mathcal{D}}_\mathsf{T}^{\phi_\mathsf{T}} = \{\tilde{x}'_1 \dots, \tilde{x}'_m\} \text{ with } \tilde{x}'_j = \phi_\mathsf{T}(x'_j)$$

How to check if $\tilde{\mathcal{D}}_\mathsf{S}$ and $\tilde{\mathcal{D}}_\mathsf{T}$ have the same distribution?

Excurse: similarity measures between sample sets

Desirable properties of a similarity measure $d(S, S')$ for $S \overset{i.i.d.}{\sim} p$ and $S' \overset{i.i.d.}{\sim} p'$:

1) $p \approx p' \quad \Rightarrow \quad d(S, S')$ should be small (at least, if enough samples are available)

2) $d(S, S')$ is small $\quad \Rightarrow \quad$ learning on S and learning on S' should yield similar predictors

Observation:

- most candidate distances do not fulfill both conditions simultaneously:
  - geometric: average Euclidean distance, Chamfer distance, Haussdorf distance, ...
  - probabilistic: moments, Wasserstein distance, total variation, KL-divergence, ...
- discrepancy distance does fulfill both conditions!

Definition (Discrepancy Distance [Kifer *et al.*, 2004])

For binary classifiers $f \in \mathcal{H}$:

$$\mathrm{disc}(S, S') := 2 \sup_{h \in \mathcal{H}} \Big(\frac{1}{|S|} \sum_{x \in S} [\![h(x) = 1]\!] - \frac{1}{|S'|} \sum_{x' \in S'} [\![h(x') = 1]\!] \Big)$$

Properties:

- $\mathrm{disc}(S, S') = 2\big(1 - \inf_{h \in \mathcal{H}} \alpha(h)\big)$ for $\alpha(h) = \frac{1}{|S|} \sum_{x \in S} [\![h(x) \neq 1]\!] + \frac{1}{|S'|} \sum_{x' \in S'} [\![h(x') \neq 0]\!]$
- α is the (class-balanced) loss of a h as a classifier distinguishing between S and S'
- $\inf_h \alpha(h)$, and therefore disc, can be computed by training a classifier on a dataset obtained by merging S and S' with different labels assigned to them.

Final task to solve for unsupervised domain adaptation with discrepancy distance:

$$\min_{\phi_{\mathsf{S}}, \phi_{\mathsf{T}}, f} \quad \hat{\mathcal{R}}_{\mathsf{S}}(f \circ \phi_{\mathsf{S}}) + \mathrm{disc}(\tilde{\mathcal{D}}_{\mathsf{S}}^{\phi_{\mathsf{S}}}, \tilde{\mathcal{D}}_{\mathsf{T}}^{\phi_{\mathsf{T}}})$$

Resulting classifier $f(\phi_{\mathsf{T}}(x))$ should have low target risk $\mathcal{R}_{\mathsf{T}}$. (theoretical guarantees exist)

[D. Kifer, S. Ben-David, J. Gehrke: *"Detecting change in data streams"*, VLDB 2004]