Model selection and assessment

Gilles Gasso

INSA Rouen - ITI Department
Laboratory LITIS

December 15, 2025

Plan

The goal

Goal

- $\mathcal{D} = \{(\boldsymbol{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}_{i=1}^n$: set of labeled data
- $(\boldsymbol{x}, y) \sim \mathrm{p}(X, Y)$ with $\mathrm{p}(X, Y)$ the joint distribution generally unknown
- Goal : learn from $\mathcal{D}$ a function

$$\begin{aligned} f : \quad \mathcal{X} &\longrightarrow \mathcal{Y} \\ x &\longmapsto \hat{y} = f(\boldsymbol{x}) \end{aligned}$$

that predicts the output $\hat{y}$ associated to each point $\boldsymbol{x} \in \mathcal{X}$

Properties of the learning

- $\forall\ (\boldsymbol{x}_i, y_i) \in \mathcal{D}$, we want f to predict the correct label y_i
- f should correctly predict the labels of unseen sample $\boldsymbol{x}_j$

Example

Example : image classification





Classification methods

- K-NN
- Logistic Regression
- SVM (linear or non-linear)
- ...

$\Longrightarrow$ Which model to select ? How to asess its ability to generalize to unseen data ?

Loss function

Loss function $\ell(Y, f(X))$

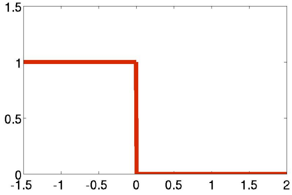
- evaluates how "close" is the prediction $f(\boldsymbol{x})$ to the true label y
- it penalizes errors: $\ell(y, f(\boldsymbol{x})) = \begin{cases} 0 & \text{if} \quad y = f(\boldsymbol{x}) \\ \geq 0 & \text{if} \quad y \neq f(\boldsymbol{x}) \end{cases}$

For binary classification

- We suppose $\mathcal{Y} = \{-1, 1\}$
- 0 - 1 cost

$$\ell(y, f(\boldsymbol{x})) = \mathbb{I}_{yf(\boldsymbol{x}) \leq 0} = \begin{cases} 0 & \text{if} \quad yf(\boldsymbol{x}) > 0 \\ 1 & \text{if} \quad yf(\boldsymbol{x}) \leq 0 \end{cases}$$

measures the number of classification errors



Risk function and learning

Risk function

Assesses the expected error (generalization ability) of f

$$R(f) = \mathbb{E}_{X,Y} \ell(Y, f(X))$$
$$R(f) = \int_{\mathcal{X},\mathcal{Y}} \ell(y, f(\boldsymbol{x}))\mathsf{p}(\boldsymbol{x}, y) d\boldsymbol{x} dy$$

Statistical learning problem

Find the function f^* that minimises $R(f)$

$$f^* = \text{argmin}_f \mathbb{E}_{X,Y} \ell(Y, f(X))$$

However

f^* is not attainable as $\mathsf{p}(X, Y)$ is unknown

Empirical risk

We only have access to a finite set of samples $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$.

Define the empirical risk

$$R_\mathsf{n}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(\boldsymbol{x}_i))$$

Empirical risk minimization

- We are looking for a decision function

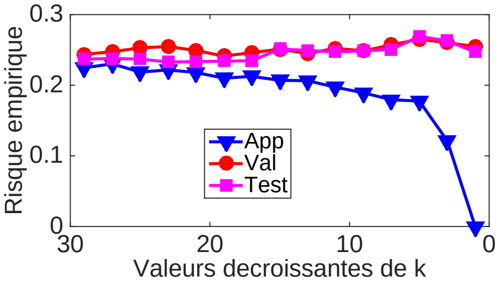
$$f_n \quad = \quad \text{argmin}_f R_\mathsf{n}(f)$$

- $R_\mathsf{n}(f_n)$ is the empirical risk corresponding to f_n. It is an approximation of the real risk $R(f_n) = \mathbb{E}_{X,Y} \ell(Y, f_n(X))$

Empirical risk and over-fitting

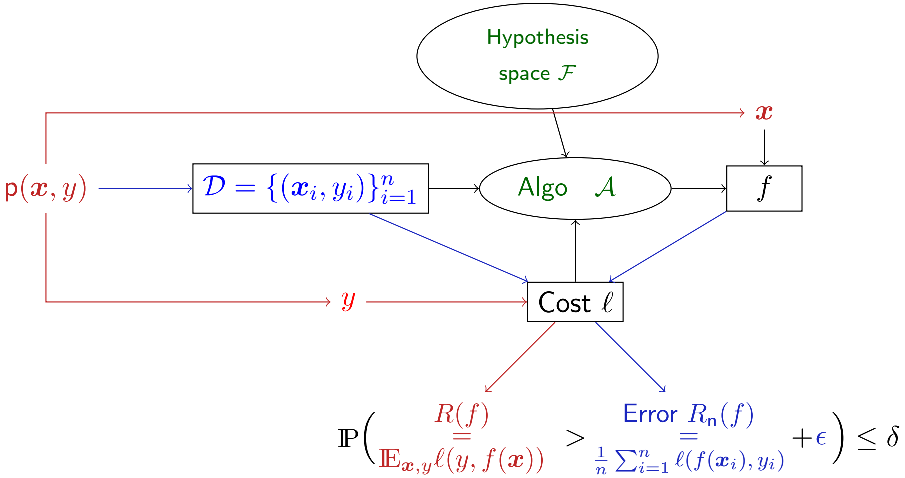
- Should we choose f based on $R_{\mathsf{n}}(f_n)$? NO !
- as we can design a sufficiently complex function f_n such that $R_{\mathsf{n}}(f_n) \to 0$ but with high risk $R(f_n)$

K-NN classification function



$\Longrightarrow$ Control the complexity of the function f

The paradigm of statistical learning



With given $\mathcal{D}$, find a model f in a family $\mathcal{F}$ (linear, kernel SVM ...) with good generalization properties

Why the learning is possible

Supremum on generalization error

Let's $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ the dataset. Let $\mathcal{F}$ be a space of functions. For each $f \in \mathcal{F}$, with probability $1 - \delta$ we have

$$R(f) \leq R_{\mathsf{n}}(f) + \mathcal{O}\left(\sqrt{\frac{h}{n}\log\frac{2en}{h} + \frac{\log 2/\delta}{n}}\right)$$

$h > 0$ measures the "complexity" of the functions class $\mathcal{F}$

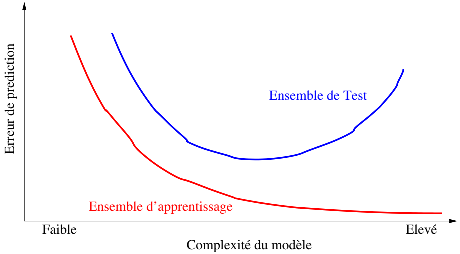
- Generalization occurs whenever $h < \infty$
- Bigger is n better it is ($n >> h$: the number of data increases with model complexity)
- Linear model $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$ with $w \in \mathbb{R}^d$, $h = d + 1$

Illustration

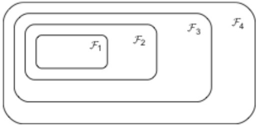
Generalization / over-fitting

$$R(f) \leq \frac{1}{n}\sum_{i=1}^{n} \ell(f(\boldsymbol{x}_i), y_i) + \mathsf{term}(n, h(\mathcal{F}))$$

- $R_{\mathsf{n}}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(\boldsymbol{x}_i), y_i)$ is not a good estimator of generalization ability
- Over-fitting appears with the increasing complexity of f



Complexity control: regularisation



Let $k_1 < k_2 < k_3 < \cdots$

We define $\mathcal{F}_j = \{f : \Omega(f) \leq k_j\}$

$\Omega(f)$: regularisation function

Example : $\Omega(f) = \|f\|^2$

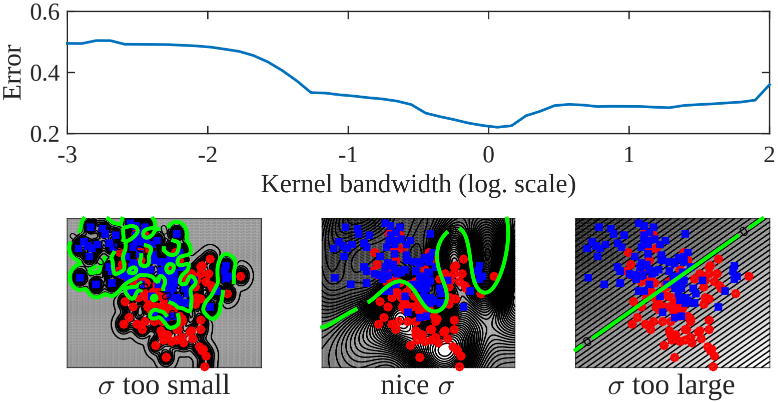
Minimization of the regularized empiric risk

$$\min_f \frac{1}{n} \sum_{i=1}^{n} \ell(f(\boldsymbol{x}_i), y_i) + \lambda\, \Omega(f)$$

- $\lambda > 0$: regularization hyper-parameter
- $\lambda >> 1 \rightarrow$ we encourage f to be of low complexity

Example : SVM $\min_f \frac{1}{n} \sum_{i=1}^{n} \ell(f(\boldsymbol{x}_i), y_i) + \lambda \|f\|^2$ with cost function $\ell(y, f(\boldsymbol{x})) = \max(0, 1 - yf(\boldsymbol{x}))$ and $\lambda = 1/C$

Illustration: influence of model's hyper-parameters

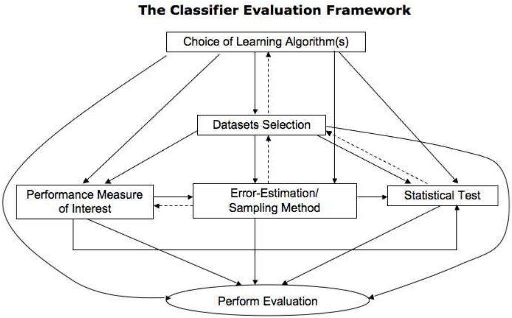


- The choice of the hyper-parameter's value (hence of the model) impacts the quality of the prediction

Model selection and evaluation

Raised issues

- Model evaluation : what measure(s) of performance?
- Estimation of the generalisation capacity of the model
- Practical model selection procedures



Plan

Assessing the quality of a model

The confusion matrix

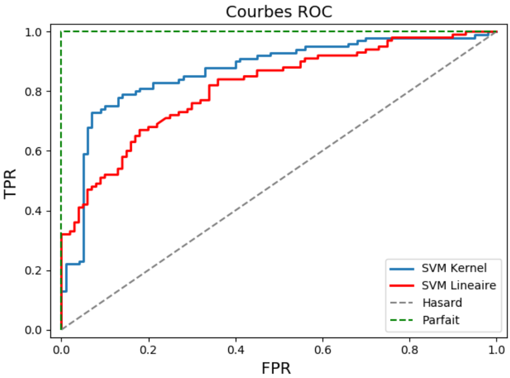
A matrix showing the predicted and actual classifications. A confusion matrix is of size $p \times p$, where p is the number of classes.

Predicted / Actual	**Positive**	**Negative**
Positive	TP	FP
Negative	FN	TN
	P = TP + FN	N = FP + TN

- Error rate $=$ (FP + FN)/(P + N) ($\searrow\searrow$)
- Accuracy $=$ 1 - Error rate $=$ (TP + TN)/(P + N) ($\nearrow\nearrow$)
- Precision $=$ TP/(TP + FP)
- Recall, Sensitivity $=$ TP/P
- Specificity $=$ FP/N
- F-Measure $= 2\dfrac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$ ($\nearrow\nearrow$)

ROC Curve

- It's the curve $\text{TPR} = \text{fonction}(\text{FPR})$
- Allows graphical comparison of different models



Measure of performances

Area Under the ROC Curve (AUC)

- Let $\mathcal{D} = \{(\boldsymbol{x}_i, y_i = 1)\}_{i=1}^{P} \cup \{(\boldsymbol{x}_j, y_j = -1)\}_{i=1}^{N}$ and f be the decision function. The AUC is defined by

$$\text{AUC} = \sum_{i=1}^{P}\sum_{j=1}^{N} \frac{\mathbb{I}\left[f(\boldsymbol{x}_i) > f(\boldsymbol{x}_j)\right] + 0.5\,\mathbb{I}\left[f(\boldsymbol{x}_i) = f(\boldsymbol{x}_j)\right]}{P \times N}$$

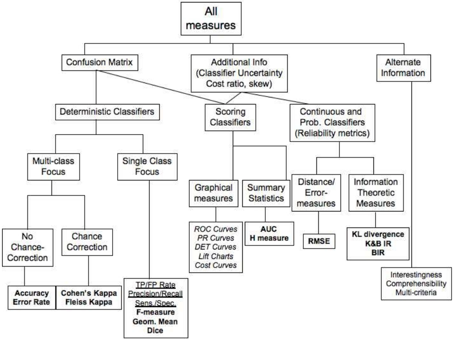
  with $\mathbb{I}$ the indicator function

- AUC is between 0 and 1 ($\nearrow\nearrow$)
- Favours the decision function such that $f(\boldsymbol{x}_i) > f(\boldsymbol{x}_j)$ $\forall\ (y_i = 1, y_j = -1)$

Other performance measures

- Many performance measures exist
- Each classifier may be the best one according to a specific measure
- Keep in mind that your model may fail according to another measure

→ Choose wisely according to your problematic



N. Japkowicz & M. Shah, "Evaluating Learning Algorithms: A Classification Perspective", Cambridge University Press, 2011

The model' generalization

- Let f be a decision-making function developed using the data $\mathcal{D}_n = \{(\boldsymbol{x}_i, y_i)\}_{i=1\cdots n}$
- We are looking at $R(\mathcal{D}_\infty, f)$ the theoretical performance of f on all possible future data

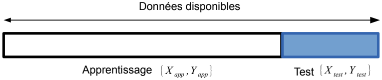
Generalisation Capacity

Capacity of f to perform well (measured with one of the previous metrics) when tested on data other than those used for training

How to estimate $R(\mathcal{D}_\infty, f)$ in practice ?

Paradigm test set/training set

Randomly split $\mathcal{D}_n$ into two disjoints sets $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$



- $\mathcal{D}_{train} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_{train}}$: data used for training f
- $\mathcal{D}_{test} = \{(\boldsymbol{x}_i, y_i)\}_{i=}^{n_{test}}$: data used to evaluate the generalization capacity of f

Remark

- Bigger n_{train} is, better the training
- Bigger n_{test} is, better the estimation of performance is f
- $\mathcal{D}_{test}$ is used only once !

Error bars on Bernoulli trials

Hypothesis

My new method classifies well 90 (n_S) examples over 100 (n). 10 (n_F) examples are mis-classified. What is my level of confidence?

Level of confidence α

success probability : $\widehat{p} = 0.9$

$$\hat{p}_\alpha = \widehat{p} \pm z\sqrt{\frac{\widehat{p}\,(1-\widehat{p})}{n}} = \frac{n_S}{n} \pm \frac{z}{n}\sqrt{\frac{n_S n_F}{n}}$$

with z is the $1-\frac{\alpha}{2}$ quantile of a standard normal distribution.

- Consider $\alpha = 0.95$,
- z = scipy.stats.norm.ppf(0.975)*np.sqrt(0.9*(1−0.9)/100)
  $\widehat{p}_\alpha = 0.9 \pm 0.059$
- ie. 95% of time: $0.84 < \widehat{p} < 0.96$

http://en.wikipedia.org/wiki/Binomial_proportion_confidence_interval

To improve the estimate

Dataset size

- If you increase the number of runs, your confidence increases.
- Check the confidence interval

Increase n

- Random Subsampling (The repeated holdout method)
- K-Fold Cross-Validation ($K = 10, 5, 2, \ldots$)
- Leave-one-out Cross-Validation ($K = n$)
- Bootstrap (each sample can be in differents subsets)

Conclusion

Best practices

- Simulate real conditions
- Avoid test set bias by adding it within learning procedure
- Look for stability rather than performance

Plan

Model Selection: the principle

Problem

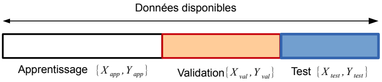
- Given a set of models $\mathcal{F} = \{f_1, f_2, \cdots\}$, choose the decision function giving the best performances on future data

Examples of function choice by classification type

- K-NN :choice of K
- Sparse Logistic Regression : number of selected variables
- SVM : choice of the hyper-parameter C, kernel tuning
- $\cdots$

Validation set

How to choose the "best" model without testing on $\mathcal{D}_{test}$?



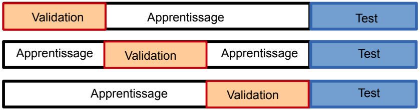
1. Randomly split $\mathcal{D}_n = \mathcal{D}_{train} \cup \mathcal{D}_{val} \cup \mathcal{D}_{test}$
2. Train each possible model on $\mathcal{D}_{train}$
3. Evaluate the performance on $\mathcal{D}_{val}$
4. Select the model with the best performance on $\mathcal{D}_{val}$
5. Test the selected model on $\mathcal{D}_{test}$

Remark

- $\mathcal{D}_{test}$ is used only one time !

K-fold validation

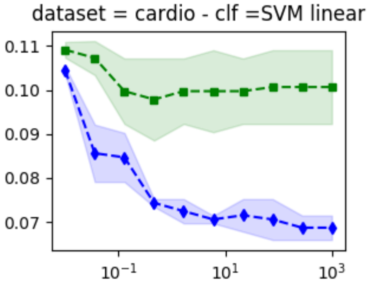
What if the size of $\mathcal{D}_n$ is small ?



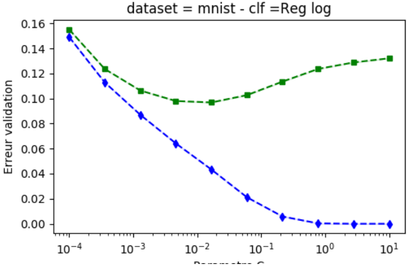
1. Randomly split $\mathcal{D}_n = \mathcal{D}_{train} \cup \mathcal{D}_{test}$
2. Then split randomly $\mathcal{D}_{train} = \mathcal{D}_1 \cup \cdots \cup \mathcal{D}_K$ in K sets
3. For $k = 1$ to K
    1. Put aside $\mathcal{D}_k$
    2. Train the model f on the $K - 1$ remaining sets
    3. Evaluate its performance R_k on generalizing to $\mathcal{D}_k$
4. Average the K measures of performance R_k

Illustration

K-Fold Cross-Validation



Cross-Validation



Plan

Fairness in Machine Learning

- Fairness refers to the absence of unjustified discrimination in algorithmic decision-making.
- A machine learning system is unfair if it systematically disadvantages individuals or groups based on sensitive attributes.
- Sensitive attributes may include:
  - Race, gender, age
  - Disability status
  - Socioeconomic background

Fairness: COMPAS example[1]

White defendants

Outcome	Prediction: Low Risk	Prediction: High Risk
No Recidivism	1139 (TN)	349 (FP)
Recidivated	461 (FN)	505 (TP)

Error Rate $\approx 33\%$
False Positive Rate $\approx 23.5\%$
False Negative Rate $\approx 47.7\%$

Black defendants

Outcome	Prediction: Low Risk	Prediction: High Risk
No Recidivism	990 (TN)	805 (FP)
Recidivated	532 (FN)	1369 (TP)

Error Rate $\approx 36.2\%$
False Positive Rate $\approx 44.9\%$
False Negative Rate $\approx 28.0\%$

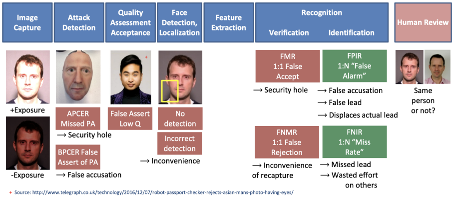
Findings

- Similar overall error rates between white and black defendants but...
- ...very different outcomes for white and black defendants
  - Black defendants have 1.9x higher False Positive Rate
  - White defendants have 1.7x higher False Negative Rate

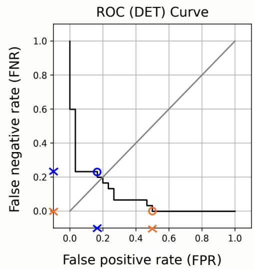
[1] https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm

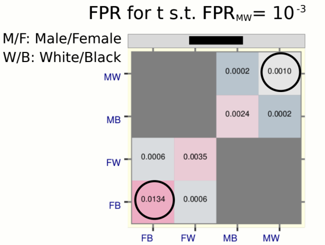
Fairness: Facial Recognition (FR)

FR involves ML or AI algorithms at different stages of the processing pipeline



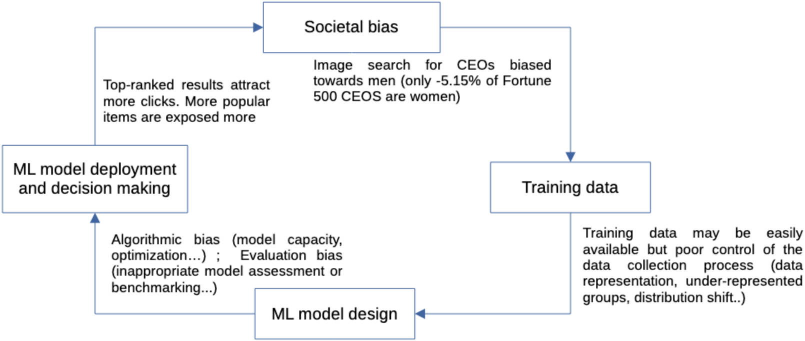
NIST reports show discrepancies in error rates between social groups for FR





(Grother and Ngan, 2019)

Source of bias



Formalizing fairness

Individual Fairness

- Principle: *Similar individuals* (differing only on sensitive attributes) should receive *similar outcomes* $\|\boldsymbol{x} - \boldsymbol{x}'\| \leq \varepsilon \Rightarrow \|f(\boldsymbol{x}) - (\boldsymbol{x}')\| \leq \varepsilon'$
- Requires to define the task-specific similarity metric
- Scale poorly to large scale data.

Group fairness

- Ensures statistical parity across predefined groups

$$\text{Fairness metric} = R(R(f; \mathcal{D}_1), \cdots, R(f; \mathcal{D}_K)) \quad \text{for } K \text{ subgroups}$$

- Groups are defined by sensitive attributes S
- Easier to measure and commonly used in practice

Different strategies to ensure Fairness

- Pre-processing: produce discrimination-free training data
  - Reweighting samples
  - Removing sensitive features
  - Learning fair representations
- In-processing: fairness-aware model training
- Post-processing: correcting biased predictors
  - output correction
  - input correction
  - classifier correction

In-processing: example

- Minimize classification error with fairness constraints over subgroup defined by the attribute S

$$\begin{aligned} \min_f \quad & R(f) \\ \text{s.t.} \quad & \mathbb{P}(f(X,S) > 0 | Y = 1, S = A) = \mathbb{P}(f(X,S) > 0 | Y = 1, S = B) \end{aligned}$$

- Empirical minimization

$$\begin{aligned} \min_f \quad & R_\mathsf{n}(f) \\ \text{s.t.} \quad & |R_\mathsf{n}^A(f) - R_\mathsf{n}^B(f)| \leq \varepsilon \end{aligned}$$

with $R_\mathsf{n}^A(f) = \hat{\mathbb{P}}\left(f(X,S) > 0 | Y = 1, S = A\right)$ the empirical probability

In-processing: example for kernel SVM [2]

Let $\mathcal{H}$ a Hilbert space induced by kernel k such that the feature map is defined by $\boldsymbol{x} \mapsto \phi(\boldsymbol{x})$ and $f(\boldsymbol{x}) = \langle w, \phi(\boldsymbol{x}) \rangle$

Optimization problem

$$\begin{aligned} \min_{w \in \mathcal{H}} \quad & \tfrac{1}{n} \textstyle\sum_{i=1}^{n} \ell(f(\boldsymbol{x}_i), y_i) + \lambda \|w\|^2 \\ \text{s.t.} \quad & |\langle w, u \rangle|_{\mathcal{H}} \leq \varepsilon \end{aligned}$$

Relaxation of the fairness constraint

$$u = u_A - u_B \quad \text{with} \quad u_A = \frac{1}{n_A} \sum_{i=1, S_i = A} \phi(\boldsymbol{x}_i)$$

[2] Empirical Risk Minimization under Fairness Constraints

Post-processing: example

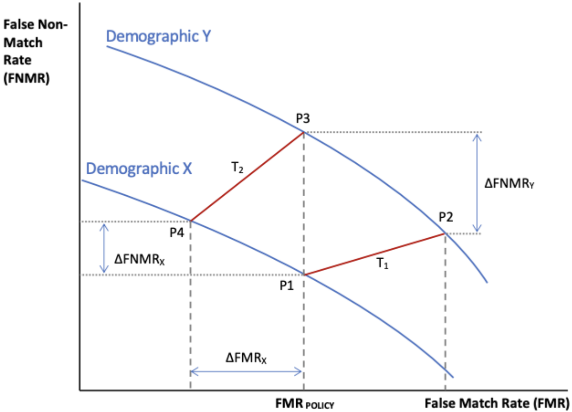


Figure 28: The figure shows the increases in FNMR implied by increasing the operating threshold to achieve the target FMR on the high-FMR demographic, Y.

https://nvlpubs.nist.gov/nistpubs/ir/2019/nist.ir.8280.pdf