Understanding the Efficiency of Machine Learning: Progress and Challenges

Réseau Numérique en Terre Solide (NuTS)

Aurélien Garivier

May 30th, 2023

UNIVERSITÉ DE LYON
ENS DE LYON

Table of contents

Supervised Learning

What we want to do: prediction

Phenomenon: observations $(x, y) \in \mathcal{X} \times \mathcal{Y}$ in a product of measurables spaces $\mathcal{X} \subset \mathbb{R}^p$ and $\mathcal{Y} \subset \mathbb{R}^q$.

Goal: predict y from x. Prediction error measure by *loss* $\ell(\hat{y}, y) = \|\hat{y} - y\|^2/2$ typically.

Statistical hypothesis: there exists $F : \mathcal{X} \times \Omega \to \mathcal{Y}$ such that the observations are distributed as (X, Y) where X has distribution $\mathbb{P}_X$ and $Y = F(X, \omega)$. Typically, $Y = f(X) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

Examples:

- classification (OCR, image recognition, text classification, etc.)
- regression (response to a drug, weather or stock price forecast, etc.)

Target = best possible guess of Y given X: $f(X) = \mathbb{E}[Y|X]$

Supervised Learning Framework

Mechanism of f is complex or hidden. Access to f only thru **examples** i.e. a sample $S_n = \big((X_1, Y_1), \ldots, (X_n, Y_n)\big)$ of random pairs

Learning algorithm $\mathcal{A}_n : S_n \mapsto \hat{f}_n$ where $\hat{f}_n \in \mathcal{F} \subset \mathcal{Y}^{\mathcal{X}} \subset (\mathbb{R}^q)^{\mathbb{R}^p}$

$\mathcal{F}$ = **hypothesis class** = model. Example: linear regression

$$\mathcal{F} = \left\{ f_\theta : x \mapsto \left(\theta_{i,0} + \sum_{j=1}^{p} \theta_{i,j} x_j \right)_{1 \le i \le q} : \theta \in \mathcal{M}_{q,1+p}(\mathbb{R}) \right\}$$

Quality of prediction $\hat{y}$: **loss function** $\ell : \mathbb{R}^q \times \mathbb{R}^q \to \mathbb{R}_+$ e.g. $\ell(\hat{y}, y) = \frac{(\hat{y} - y)^2}{2}$

Quality of hypothesis $f \in \mathcal{F}$: **generalization error** = average loss

$$L(f) = \mathbb{E}\big[\ell(f(X), Y)\big]$$ expectation is on new observation (X,Y)

Quality of the learning algorithm $\mathcal{A}$: **risk** = average average loss

$$R_n(\mathcal{A}_n) = \mathbb{E}\left[L(\hat{f}_n)\right]$$ expectation is on sample S_n

Empirical Risk Minimization

Learning = how to find the best possible $f \in \mathcal{F}$?

→ Minimize the **empirical loss = training error**

$$L_n(f) = \frac{1}{n}\sum_{k=1}^{n} \ell\big(f(X_k), Y_k\big) \qquad \text{average loss on the sample}$$

= unbiased estimator of the generalization error $L(f)$

Empirical Risk Minimizer: $\hat{f}_n \in \arg\min_{f\in\mathcal{F}} L_n(f)$

Example: linear regression with quadratic loss (dates back at least to Gauss) $\hat{f}_n = f_{\hat{\theta}_n}$ where $\hat{\theta}_n^T = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$, with

$$\mathbf{X} = \begin{pmatrix} 1 & X_1^1 & \dots & X_1^p \\ & \dots & & \\ 1 & X_n^1 & \dots & X_n^p \end{pmatrix} \text{ and } \mathbf{Y} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}$$

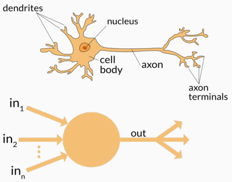
Regression by polynomials of degrees $1, 2, \dots, n-1$ → more parameters is not necessarily better, bias / variance tradeoff, Structural Risk Minimization (penalize empirical risk by model complexity)

Feedforward Neural Networks: Mimicking Brains?

Neuron: $x \mapsto \sigma\big(\langle w, x\rangle + b\big)$ with

- parameter $w \in \mathbb{R}^p, b \in R$
- (non-linear) activation function $\sigma : \mathbb{R} \to \mathbb{R}$

  typically $\sigma(x) = \frac{1}{1+\exp(-x)}$ or $\sigma(x) = \max(x, 0)$ called ReLU



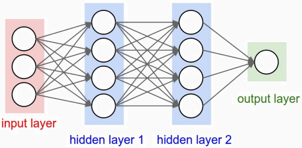
Layer: $x \mapsto \boldsymbol{\sigma}\big(Mx + \mathbf{b}\big)$ with

- parameter $M \in M_{q,p}(\mathbb{R}), \mathbf{b} \in \mathbb{R}^q$
- component-wise activation function $\boldsymbol{\sigma} = \sigma^{\otimes q}$

Network: composition of layers $f_\theta = \boldsymbol{\sigma}_D \circ T_D \circ \cdots \circ \boldsymbol{\sigma}_1 \circ T_1$ with

- architecture $A = \big(D, (p_1, \ldots, p_{D-1})\big)$
- $x_0 = x, \quad x_d = \boldsymbol{\sigma}_d\big(T_d x_{d-1}\big) \in \mathbb{R}^{p_d}$
- $T_d x = M_d x + \mathbf{b}_d$
- parameter $\theta = (M_1, \mathbf{b}_1, \ldots, \ldots, M_D, \mathbf{b}_D)$

  $\theta \in \Theta_A = \prod_{d=1}^{D} \mathcal{M}_{p_{d-1},p_d}(\mathbb{R}) \times \mathbb{R}^{p_d}$
- *depth* D (⚠st. nb layers), *width* $\max_{1\leq d\leq D} p_d$



Deep Neural Networks in the last Decade

Several other important ideas:

- not fully connected layers
- convolution layers
- max-pooling
- dropout
- physics-informed loss functions
- etc...

Some were considered to be central and are then left apart... Even, without those complications, understanding the success of neural nets remains a challenge

How to learn with feedforward neural networks?

1. Choose architecture $A = \left[D, (p_1, \ldots, p_{D-1})\right]$
    - depth D?
    - what architectures are good if f has some with given properties?
    - activation function? sigmoid $\sigma(x) = \frac{1}{1+\exp(-x)}$ or ReLU $\sigma(x) = \max(x, 0)$

    → approximation theory?
2. Learn = find the good coefficients using S_n
    - Empirical Risk Minimization: $\hat{f}_n$ solution of

$$\min_{\substack{T_k \in \mathcal{M}_{p_d, 1+p_{d-1}}(\mathbb{R}) \\ 1 \leq d \leq D}} \quad \frac{1}{n} \sum_{k=1}^{n} \ell\big(\boldsymbol{\sigma}_D \circ T_D \circ \cdots \circ \boldsymbol{\sigma}_1 \circ T_1(X_k), Y_k\big)$$

    - non convex, high-dimensional optimization problem
    - but gradient can be computed by **back-propagation**

    → does gradient descent work?
3. Apply $\hat{f}_n$ to new data (X, Y)
    - how to bound the generalization error $L(\hat{f}_n)$?
    - should we regularize = penalize large coefficients?

    → no overfitting?

→ How to explain the huge empirical success of deep learning?

Outline

Depth-2 Networks Are Universal

Cybenko ['89] Approximation by superposition of sigmoidal functions

Theorem

Let σ be any bounded, measurable (or continuous) function such that $\sigma(t) \to 0$ as $t \to -\infty$ and $\sigma(t) \to 1$ as $t \to \infty$. Then for every continuous function f on $[0,1]^p$ there exists a width p_1 and a depth-2 neural network with activation functions $\sigma_1 = \sigma$ and $\sigma_2 = id$

$$f_\theta(x) = \sum_{j=1}^{p_1} \alpha_j \sigma\big(\langle w_j, x\rangle + b_j\big)$$

such that $\|f_\theta - f\|_\infty$.

Proof:

- these functions σ are such that if for a measure μ on $[0,1]^p$

$$\int_{[0,1]^p} \sigma\big(\langle w, x\rangle + b\big) d\mu(x) = 0$$

for all $w \in \mathbb{R}^p$ and $b \in \mathbb{R}$, then $\mu = 0$.

- Hahn-Banach + Riesz representation: the closure of $\bigcup_p \left\{ f_\theta : \theta \in \mathcal{M}_{p_1,p+1}(\mathbb{R}) \times \mathbb{R}^{p_1} \right\}$ has empty complement

An Quantitative bounds for ReLU depth-2 networks

Lemma [e.g. Eldan&Shamir'16]

Let $g : \mathbb{R} \to \mathbb{R}$ be constant outside of an interval $[-R, R]$ and L-Lipschitz. There exists a depth-2 ReLU network f with linear output of width at most $8RL/\epsilon$ and weights at most $\max\left(2L, \|g\|_\infty\right)$ such that $\|f - g\|_\infty \leq \epsilon$.

Proof. If $2RL \leq \epsilon$, take f to be constantly equal to $g(-R)$.
Otherwise, take $m = \lceil RL/\epsilon \rceil \leq 2RL/\epsilon$, and let f be the piecewise linear function coinciding with g at points $x_i = i\epsilon/L$, $i \in \{-m, \ldots, m\}$, linear between x_i and x_{i+1}, and constant outside of $[-x_{-m}, x_m]$. Since g is L-Lipschitz, $\|f - g\|_\infty \leq \epsilon$. But f can be written as a depth-2 ReLU network with $2m + 2 \leq 8RL/\epsilon$ neurons:

$$f(x) = f(x_{-m}) + \sum_{i=-m}^{m} \left[f'(x_i+) - f'(x_i-)\right] r(x - x_i)$$

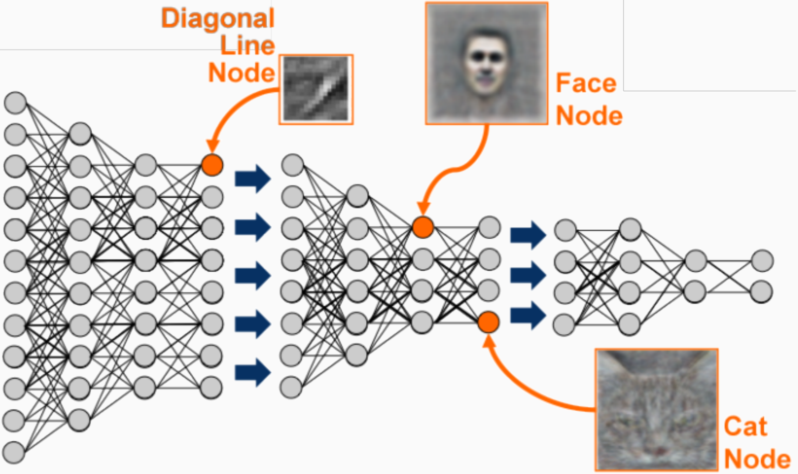
where $f'(x_i+) = g(x_{i+1}) - g(x_i)$ and $f'(x_i-) = g(x_i) - g(x_{i-1})$ for all $-m < i < m$. Except maybe for the constant $f(x_{-m}) = g(-R)$, the coefficients are bounded by $\left|g(x_{i+1}) - g(x_i) - g(x_i) + g(x_{i-1})\right| \leq 2L$.

Example: radial function

Corollary [Daniely'17, Cor. 6]

Let $g : [-1, 1] \to [-1, 1]$ be L-Lipschitz function and let $\epsilon > 0$. For a positive integer d, let $G : \mathbb{S}^{d-1} \times \mathbb{S}^{d-1} \to [-1, 1]$ be defined by $G(\mathbf{x}, \mathbf{x}') = g\big(\langle \mathbf{x}, \mathbf{x}' \rangle\big)$.
There exists a depth-3 ReLU network f of width at most $\frac{16d^2L}{\epsilon}$ and weights bounded by $\max(4, 2L)$ such that $\|f - G\|_\infty \leq \epsilon$.

Why deep learning, then? The dream

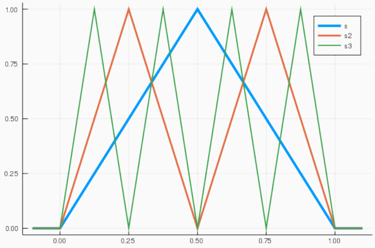


Example: sawteeth function

Let $s(x) = \begin{cases} 2x & \text{if } 0 \leq x \leq \frac{1}{2} \\ 2 - 2x & \text{if } \frac{1}{2} \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases}$

$$= 2r(x) - 4r\left(x - \frac{1}{2}\right) + 2r(x-1)$$

and for all $m \geq 1$ let $s_m = \underbrace{s \circ \cdots \circ s}_{m \text{ times}}$



Lemma

For all $m \geq 1$, all $k \in \{0, \ldots, 2^{m-1} - 1\}$ and all $t \in [0, 1]$,

$$s_m\left(\frac{k+t}{2^{m-1}}\right) = \begin{cases} 2t & \text{if } t \leq \frac{1}{2} \\ 2 - 2t & \text{if } t \geq \frac{1}{2} \end{cases}$$

Example: square function

Let $g(x) = x^2$, and for $m \geq 0$ let $g_m(x)$ be such that $\forall k \in \{0, \ldots, 2^m\}$:

- $g_m\left(\frac{k}{2^m}\right) = g\left(\frac{k}{2^m}\right)$
- g_m is linear on $\left[\frac{k}{2^m}, \frac{k+1}{2^m}\right]$

Lemma

For all $k \in \{0, \ldots, 2^m - 1\}$ and all $t \in [0, 1]$,

$$g_m\left(\frac{k+t}{2^m}\right) - g\left(\frac{k+t}{2^m}\right) = \frac{t(1-t)}{4^m}$$

In particular, $\|g - g_m\|_\infty = \frac{1}{4^{m+1}}$ and for all $m \geq 1$,

$$g_m = g_{m-1} - \frac{1}{4^m}\, s_m = id - \sum_{j=1}^{m} \frac{1}{4^j}\, s_j$$

Corollary

For every $\epsilon > 0$, there exists a neural network f of depth $\lceil \log_4(1/\epsilon) \rceil$, width 3 and coefficients in $[-4, 2]$ such that $\|f - g\|_\infty \leq \epsilon$ on $[0, 1]$

Example: square function

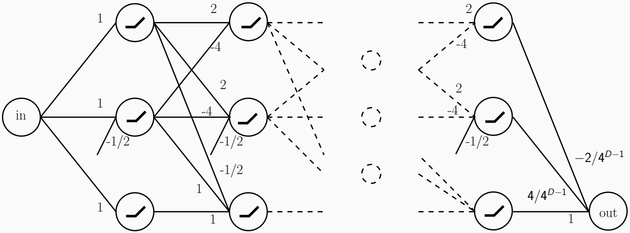
Lemma

$\|g - g_m\|_\infty = \frac{1}{4^{m+1}}$ and for all $m \geq 1$,

$$g_m = g_{m-1} - \frac{1}{4^m}\, s_m = id - \sum_{j=1}^{m} \frac{1}{4^j}\, s_j$$

Corollary

For every $\epsilon > 0$, there exists a neural network f of depth $\lceil \log_4(1/\epsilon) \rceil$, width 3 and coefficients in $[-4, 2]$ such that $\|f - g\|_\infty \leq \epsilon$ on $[0, 1]$



$x_0 = x$ $x_1 = x$ $x_2 = x - \frac{s(x)}{4}$ $x_D = x - \frac{s(x)}{4} - \cdots - \frac{s_{D-1}(x)}{4^{D-1}}$

Examples

Square on $[-1,1]$: $|x| = r(x) + r(-x)$ → one additionnal width-2 layer is sufficient

Product: $\forall x, y \in \mathbb{R}$, $xy = [(x+y)^2 - (x-y)^2]/4$ → same depth, width 5

Polynomials: approximated by products

Continuous functions on $[0,1]$: use uniform approximation of Lagrange interpolation at Chebishev's points [Liang & Srikant '19]

See [M. Telgarsky '16-'19. Benefits of depth in neural networks]

See work and presentation by Rémi Gribonval

Exponential separation result: [Daniely '17. Depth Separation for Neural Networks]

Outline

Gradient Descent on the empirical loss

Let $r(\theta) = L_n(f_\theta) = \frac{1}{n}\sum_{k=1}^{n} \ell\big(f_\theta(X_k), Y_k\big)$

- The weights are initialized at random, e.g. $\theta_0^d(i,j) \sim \mathcal{N}(0,1)$
- Then, they are updated by gradient descent: $\theta_t = \theta_{t-1} - \eta_t \nabla r$
- Possibility to penalize the empirical loss with $\|\theta\|^2 \rightarrow$ adds a tampering term in gradient descent
- Possibly Stochastic Gradient Descent: pick a point (or a batch) at random (or turn on the data in epochs)
- convergence to a local minimum (and how to choose η_t)?
- to a global minimum? especially when over-parameterized?
  See [Mei, Montanari, Nguyen '18-'19. A Mean Field View of the Landscape of Two-Layers Neural Networks]

Computing the Gradient by Backpropagation

For every layer $d \in \{1, \ldots, D\}$, we define the vector $\delta_d \in \mathbb{R}^{p_d}$ by $\delta^d(i) = \frac{\partial r}{\partial x_d(i)} \sigma'_d\big(\tilde{x}_d(i)\big)$

Recursive Equations of Backpropagation

For the squared loss $\ell(\hat{y}, y) = \frac{\|\hat{y}-y\|^2}{2}$,

$$\delta_D = \frac{1}{n}\sum_{k=1}^{n}(\hat{f}_n(X_k) - Y_k) \,.*\, \sigma'_d(\tilde{x}_D(k))$$

$$\delta_{d-1} = M_d^T \delta^d \,.*\, \sigma'_{d-1}(\tilde{x}_{d-1})$$

$$\nabla_{M_d} r = \delta_d \, x_{d-1}^T$$

Cf. Automatic Differentiation.

Outline

Overfitting: the Double Descent Phenomenon

Src: `https://openai.com`

Classical statistics suggest that there are too many parameters wrt. the number of observations, BUT this is not what is empirically observed!

Deep neural nets overfit, but (contrary to polynomials) they seem to generalize well (especially in high dimension)

→ how to explain that?

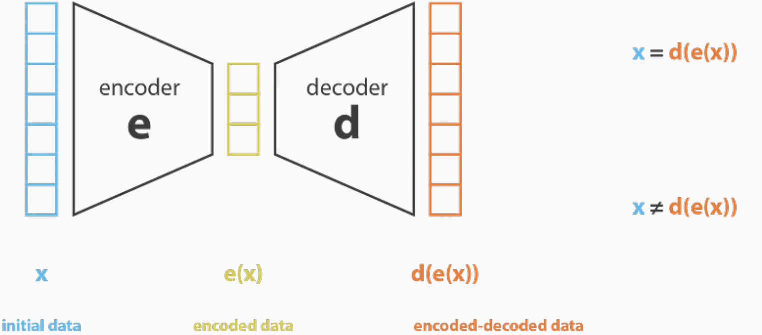
Beginning of answer: Benign Overfitting in Linear Regression Bartlett, by Long et al., 2019

Dimensionality Reduction and Generative Models

Outline

Dimensionality reduction

- Data: $X = \begin{pmatrix} x_1^T \\ \vdots \\ x_n^T \end{pmatrix} \in \mathcal{M}_{n,p}(\mathbb{R})$, $p \gg 1$
- Dimensionality reduction: replace x_i with $y_i = \mathrm{enc}(x_i)$, where $\mathrm{enc} : \mathbb{R}^p \to \mathbb{R}_d$, $d \ll p$
- Hopefully, we do not loose too much by replacing x_i by y_i: there exists a recovering mapping $\mathrm{dec} : \mathbb{R}^d \to \mathbb{R}^p$ such that for all $i \in \{1, \dots, n\}$, $\mathrm{dec}(\mathrm{enc}(x_i)) \approx x_i$

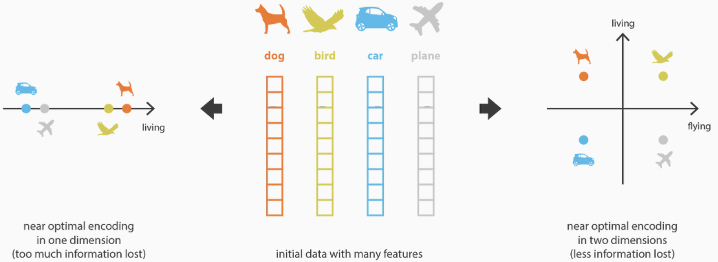


PCA = optimal linear dimensionality reduction

PCA aims at finding the compression matrix W $(= \mathrm{enc})$ and the recovering matrix U $(= \mathrm{dec})$ such that the total squared distance between the original and the recovered vectors is minimal:

$$\underset{W \in \mathcal{M}_{d,p}(\mathbb{R}), U \in \mathcal{M}_{p,d}(\mathbb{R})}{\arg\min} \sum_{i=1}^{n} \left\| x_i - UWx_i \right\|^2$$

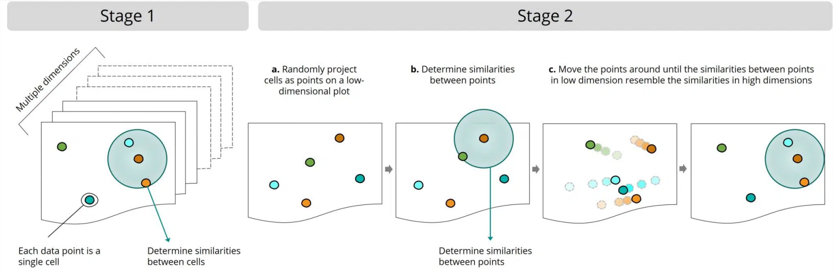
Thm: The solution is given by choosing $U =$ the eigenvectors corresponding to the highest eigenvalues of $\sum_{i=1}^{n} x_i x_i^T$, and $W = U^T$



Src: `https://towardsdatascience.com/`

t-SNE, UMAP, etc.

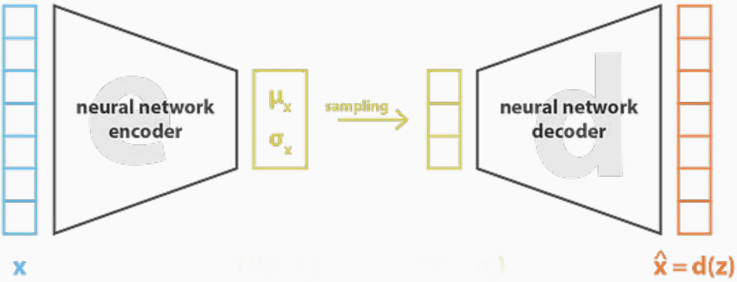
t-distributed stochastic neighbor embedding



Src: `https://www.scdiscoveries.com/`

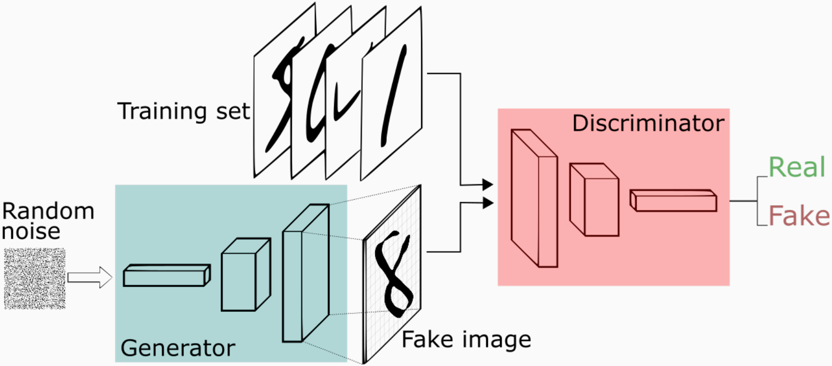
Still to be better understood and interpreted – see [A Probabilistic Graph Coupling View of Dimension Reduction, *van Assel et al.*]

Auto-encoders



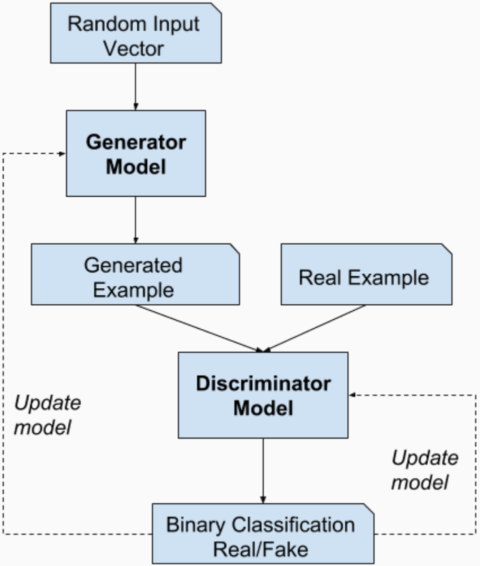
Src: https://towardsdatascience.com/

Outline



Src: https://sthalles.github.io/

Generator / Discriminator

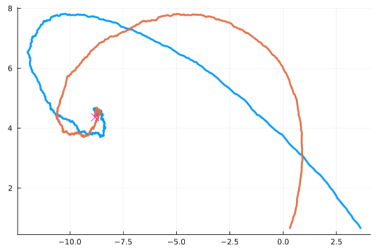


Src: `https://machinelearningmastery.com`

Convergence of a GAN

Target distribution: $Z \sim \mathcal{N}(\theta^\star, I_d)$

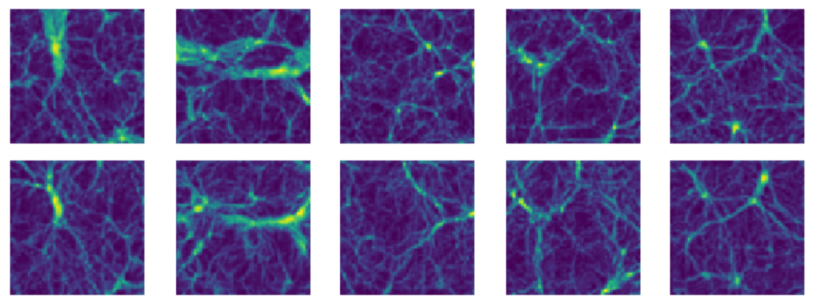
- $U \sim \mathcal{N}(0, I_d)$
- $X = U + \theta$
- fake data: $L_\psi(X, -1) = \|X + \psi\|^2$
- True data: $L_\psi(Z, -1) = \|Z - \psi\|^2$



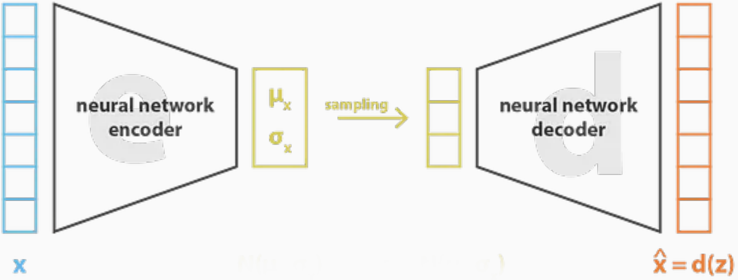
$\Longrightarrow$ in general, the convergence of a GAN is a hard problem!

Example

[Encoding large scale cosmological structure with Generative Adversarial Networks, *Marion Ullmo, Aurélien Decelle and Nabila Aghanim*, Astronomy & Astrophysics]

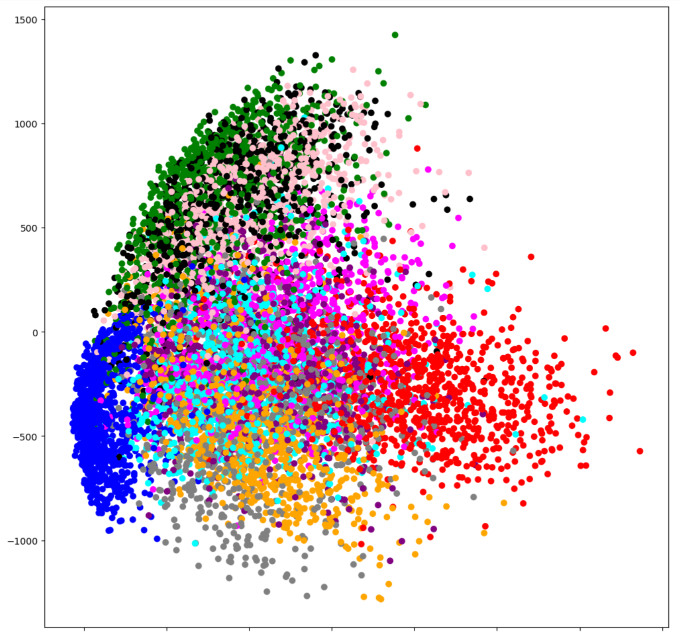


Using auto-encoders for data generation?

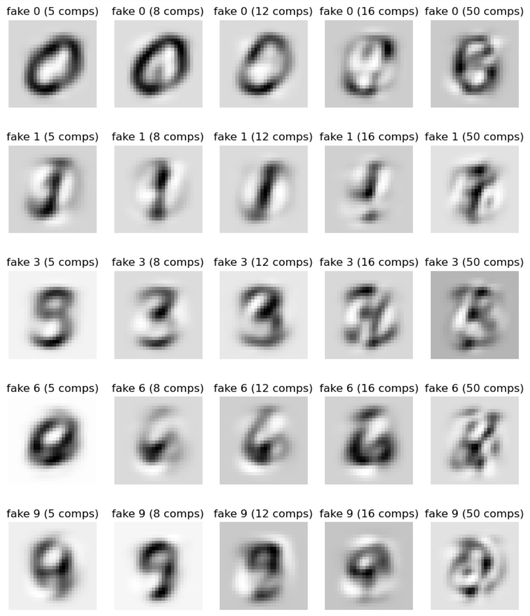


Src: https://towardsdatascience.com/

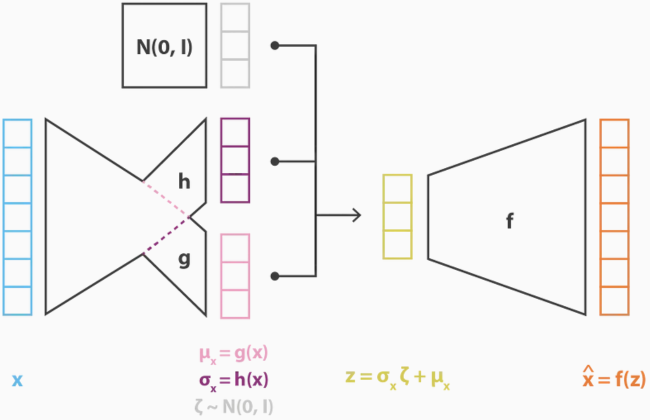
PCA for data generation? No...



PCA for data generation? No...



Variational Auto-Encoders

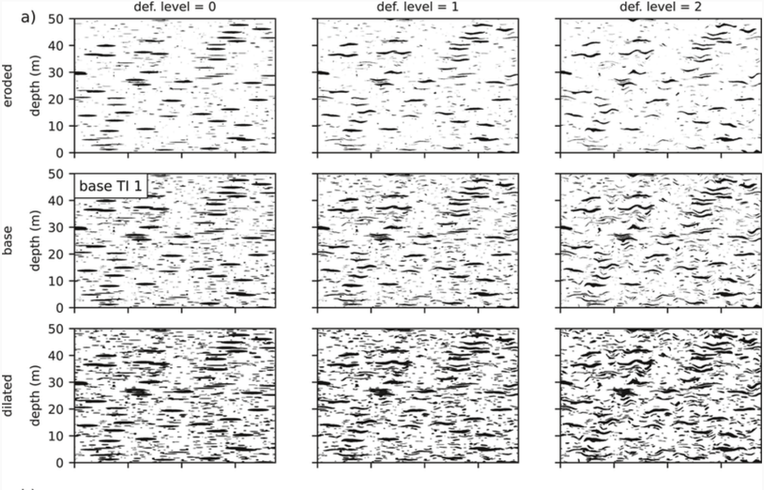


$$\text{loss} = C\,\|x - \hat{x}\|^2 + \mathrm{KL}[\,N(\mu_x, \sigma_x), N(0, I)\,] = C\,\|x - f(z)\|^2 + \mathrm{KL}[\,N(g(x), h(x)), N(0, I)\,]$$

Src: `https://towardsdatascience.com/`

Example

[Geophysical Inversion Using a Variational Autoencoder to Model an Assembled Spatial Prior Uncertainty, *Jorge Lopez-Alvis, Frederic Nguyen, M. C. Looms, Thomas Hermans*, Journal of Geophysical Research: Solid Earth]



Privacy, Fairness, Interpretability, etc.

Outline

Differential Privacy

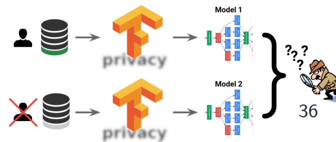
Differentially private algorithms make assurance that attackers can learn virtually nothing more about an individual than they would understand if that individual's record were absent from the dataset.

Smoker example

if an individual is openly "smoking" but wants privacy on her medical status,

- a medical study will prove the risk associated with smoking (whether she participates or not)
- a *DP* study will make it impossible to know if she indeed participated or not, even to someone who would have all the remaining information

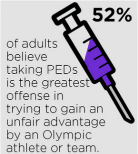
Fundamental Law of Information Recovery:
Need to *randomize* the output.



Src: `https://blog.tensorflow.org`

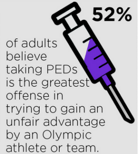
Survey on triathletes: "do you use doping?"

Triathletes doping status $X_i \stackrel{iid}{\sim} \mathcal{B}(p)$
but they may lie: answer $Y_i \in \{0, 1\}$



Survey on triathletes: "do you use doping?"

Triathletes doping status $X_i \stackrel{iid}{\sim} \mathcal{B}(p)$
but they may lie: answer $Y_i \in \{0, 1\}$



RANDOMIZED RESPONSE: A SURVEY TECHNIQUE FOR ELIMINATING EVASIVE ANSWER BIAS

STANLEY L. WARNER
Claremont Graduate School

For various reasons individuals in a sample survey may prefer not to confide to the interviewer the correct answers to certain questions. In such cases the individuals may elect not to reply at all or to reply with incorrect answers. The resulting evasive answer bias is ordinarily difficult to assess. In this paper it is argued that such bias is potentially removable through allowing the interviewee to maintain privacy through the device of randomizing his response. A randomized response method for estimating a population proportion is presented as an example. Unbiased maximum likelihood estimates are obtained and their mean square errors are compared with the mean square errors of conventional estimates under various assumptions about the underlying population.

1. INTRODUCTION

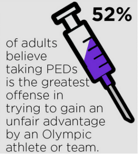
For reasons of modesty, fear of being thought bigoted, or merely a reluctance to confide secrets to strangers, many individuals attempt to evade certain questions put to them by interviewers. In survey vernacular, these people become the "non-cooperative" group [5, pp. 235–72], either refusing outright to be surveyed, or consenting to be surveyed but purposely providing wrong answers to the questions. In the one case there is the problem of refusal bias [1, pp. 355–61], [2, pp. 33–6], [5, pp. 261–9]; in the other case there is the problem of response bias [3, p. 89], [4, pp. 280–325].

Journal of the American Statistical Association, Mar. 1965, Vol.60, No.309, pp. 63-69

See also Chong, Chun Yin Andy & Chu, Amanda & So, Mike & Chung, Ray. (2019). *Asking Sensitive Questions Using the Randomized Response Approach in Public Health Research: An Empirical Study on the Factors of Illegal Waste Disposal*. International Journal of Environmental Research and Public Health.

Survey on triathletes: "do you use doping?"

Triathletes doping status $X_i \overset{iid}{\sim} \mathcal{B}(p)$
but they may lie: answer $Y_i \in \{0, 1\}$



Randomized Response [Warner'65]

Flip a coin, then:

$\rightarrow$ if tails, answer according to another coin flip

$\rightarrow$ if heads, give the right answer

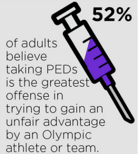
$$\mathbb{P}(Y = 1 | X = x) = 1/4 + x/2 \qquad \frac{\mathbb{P}(Y = 1 | X = 1)}{\mathbb{P}(Y = 1 | X = 0)} = 3$$

- No triathlete can be prosecuted one cannot condemn 1/4th of the innocent triathletes!
- But still permits to estimate the proportion of dopers by $2\bar{Y}_n - 1$.

Cost: for the same precision, requires $\approx 4x$ more data or even more if $x(1-x) \ll 1$

Survey on triathletes: "do you use doping?"

Triathletes doping status $X_i \overset{iid}{\sim} \mathcal{B}(p)$
but they may lie: answer $Y_i \in \{0, 1\}$



Randomized Response [Warner'65]

Flip a coin, then:

→ if tails, answer according to another coin flip

→ if heads, give the right answer

$$\mathbb{P}(Y = 1 | X = x) = 1/4 + x/2 \qquad \frac{\mathbb{P}(Y = 1 | X = 1)}{\mathbb{P}(Y = 1 | X = 0)} = 3$$

- No triathlete can be prosecuted one cannot condemn 1/4th of the innocent triathletes!
- But still permits to estimate the proportion of dopers by $2\bar{Y}_n - 1$.

Cost: for the same precision, requires ≈ $4x$ more data or even more if $x(1 - x) \ll 1$

"smoker example": if $\hat{p} = 98\%$,

Formal Definition

Randomized algorithm $\mathcal{A}(x)$ = random variable on $\mathcal{T}$

Def: Neighboring databases $x \sim x'$ if $\exists i \in \{1, \dots, n\}, \forall j \neq i, x_{i,\cdot} = x'_{j,\cdot}$

Differential Privacy

[Calibrating Noise to Sensitivity, TCC'2006, C.Dwork, F. McSherry, K. Nissim et A. Smith

$\implies$ Gödel Prize 2017]

$\mathcal{A}$ is ϵ-DP if for all $x \sim x'$ and all $\mathcal{S} \subset \mathcal{T}$

$$\mathbb{P}\Big(\mathcal{A}(x) \in S\Big) \leq e^{\epsilon}\, \mathbb{P}\Big(\mathcal{A}(x') \in S\Big)$$

Equivalently,

- if $\mathcal{A}(x)$ is discrete, $\quad -\epsilon \leq \ln \frac{\mathbb{P}\left(\mathcal{A}(x)=t\right)}{\mathbb{P}\left(\mathcal{A}(x')=t\right)} \leq \epsilon \quad$ for all $t \in \mathcal{T}$
- if $\mathcal{A}(x)$ has density $f(\cdot|x)$, $\quad -\epsilon \leq \ln \frac{f(t|x)}{f(t|x')} \leq \epsilon \quad$ for all $t \in \mathcal{T}$

Formal Definition

Randomized algorithm $\mathcal{A}(x)$ = random variable on $\mathcal{T}$

Def: Neighboring databases $x \sim x'$ if $\exists i \in \{1, \ldots, n\}, \forall j \neq i, x_{i,\cdot} = x'_{j,\cdot}$

Differential Privacy

[Calibrating Noise to Sensitivity, TCC'2006, C.Dwork, F. McSherry, K. Nissim et A. Smith

$\Longrightarrow$ Gödel Prize 2017]

$\mathcal{A}$ is ϵ-DP if for all $x \sim x'$ and all $\mathcal{S} \subset \mathcal{T}$

$$\mathbb{P}\big(\mathcal{A}(x) \in S\big) \leq e^{\epsilon}\, \mathbb{P}\big(\mathcal{A}(x') \in S\big)$$

In the previous example on the DP survey, algorithm $\mathcal{A}(x) = (Y_1, \ldots, Y_n)$ is ln(3)-DP.

Note that it outputs an entire (differentially private), which is unusual: more often, we just want the answer to a query.

Formal Definition

Randomized algorithm $\mathcal{A}(x)$ = random variable on $\mathcal{T}$

Def: Neighboring databases $x \sim x'$ if $\exists i \in \{1, \ldots, n\}, \forall j \neq i, x_{i,\cdot} = x'_{j,\cdot}$

Differential Privacy

[Calibrating Noise to Sensitivity, TCC'2006, C.Dwork, F. McSherry, K. Nissim et A. Smith

$\Longrightarrow$ Gödel Prize 2017]

$\mathcal{A}$ is ϵ-DP if for all $x \sim x'$ and all $\mathcal{S} \subset \mathcal{T}$

$$\mathbb{P}\big(\mathcal{A}(x) \in S\big) \leq e^{\epsilon}\, \mathbb{P}\big(\mathcal{A}(x') \in S\big)$$

A person's privacy cannot be compromised by a statistical release if their data are not in the database. Therefore, with differential privacy, the goal is to give each individual roughly the same privacy that would result from having their data removed. That is, the statistical functions run on the database should not overly depend on the data of any one individual.

Formal Definition

Randomized algorithm $\mathcal{A}(x)$ = random variable on $\mathcal{T}$

Def: Neighboring databases $x \sim x'$ if $\exists i \in \{1, \ldots, n\}, \forall j \neq i, x_{i,\cdot} = x'_{j,\cdot}$

Differential Privacy

[Calibrating Noise to Sensitivity, TCC'2006, C.Dwork, F. McSherry, K. Nissim et A. Smith

$\Longrightarrow$ Gödel Prize 2017]

$\mathcal{A}$ is ϵ-DP if for all $x \sim x'$ and all $\mathcal{S} \subset \mathcal{T}$

$$\mathbb{P}\big(\mathcal{A}(x) \in S\big) \leq e^{\epsilon}\, \mathbb{P}\big(\mathcal{A}(x') \in S\big)$$

An algorithm is said to be differentially private if by looking at the output, one cannot tell whether any individual's data was included in the original dataset or not.

Cryptographic origins (and vocabulary).

Formal Definition

Randomized algorithm $\mathcal{A}(x)$ = random variable on $\mathcal{T}$

Def: Neighboring databases $x \sim x'$ if $\exists i \in \{1, \ldots, n\}, \forall j \neq i, x_{i,\cdot} = x'_{j,\cdot}$

Differential Privacy

[Calibrating Noise to Sensitivity, TCC'2006, C.Dwork, F. McSherry, K. Nissim et A. Smith

$\implies$ Gödel Prize 2017]

$\mathcal{A}$ is ϵ-DP if for all $x \sim x'$ and all $\mathcal{S} \subset \mathcal{T}$

$$\mathbb{P}\big(\mathcal{A}(x) \in S\big) \leq e^{\epsilon}\, \mathbb{P}\big(\mathcal{A}(x') \in S\big)$$

Differential privacy mathematically guarantees that anyone seeing the result of a differentially private analysis will essentially make the same inference about any individual's private information, whether or not that individual's private information is included in the input to the analysis.

Private Estimation and Learning

How to estimate privately? How to fit a model privately?

- Privacy Budget Management
- Laplace and Gaussian Mechanisms
- Exponential Mechanism
- DPSGD

How does privacy affect accuracy?

- Minimax rates
- Cramer-Rao bounds
- "free privacy"

See [On the Statistical Complexity of Estimation and Testing under Privacy Constraints, *Lalanne, Garivier, Gribonval*, Transactions on Machine Learning Research]

Outline

Bias in the Data

PIXELS | CHRONIQUES DES (R)ÉVOLUTIONS NUMÉRIQUES | VIE EN LIGNE

Une étude démontre les biais de la reconnaissance faciale, plus efficace sur les hommes blancs

Lorsqu'il s'agit de reconnaître le genre d'un homme blanc, des logiciels affichent un taux de réussite de 99 %. La tâche se complique lorsque la peau d'une personne est plus foncée, ou s'il s'agit d'une femme.

http://www.lemonde.fr/pixels/

Joy Buolamwini (MIT) has studied three face recognition software (by IBM, MICROSOFT and FACE++) on 1 270 official portraits of policitians from Rwanda, Senegal, South Africa, Finland and Sweden, asking to **predict their gender**.

Buolamwini Study

Average results are good: 93,7% success rate for MICROSOFT, 90% for FACE++, and 87,9% pour IBM.

BUT

- Less successful for women than for men: for example, FACE++ classifies correctly 99,3% of the men but only 78,7% of the women.
- Less successful for dark skins than for pale skins: for the IBM softwares, success rates are 77;6% versus 95%.
- 93,6% of the mistakes of the Microsoft software were on dark skins, and 95,9% of the mistakes of Face ++ were on women!

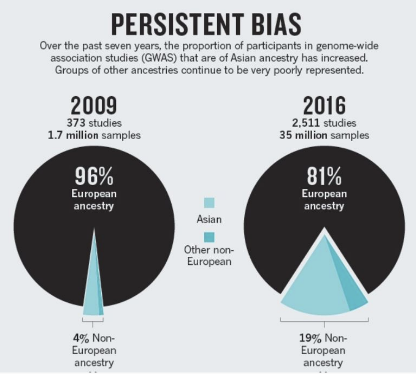
Why? Bias in the data!

"Men with white skin are over-represented, and in fact white skins in general are." http://www.lemonde.fr/pixels/article/2018/02/12/une-etude-demontre-les-biais-de-la-reconnaissance-faciale-plus-efficace-sur-les-hommes-blancs_5255663_4408996.html#EZuQdOCJvJ3kYTiL.99

This is not only about face recognition

- ...but also insurance, employment, credit risk assessment...

- ... personalized medicine: most study of pangenomic association were conducted on white/European population.
  $\Longrightarrow$ The estimated risk factors will possibly be different for patients with African or Asian origins!



Popejoy A., Fullerton S. (2016).
Genomics is failing on diversity, Nature 538

Detecting a bias

Detecting an individual discrimination: **Testing**

- Idea: modify just one protected feature of the individual and check if decision in changed
- Recognized by justice
- Discrimination for house rental, employment, entry in shops, insurance, etc.

Detecting a group discrimination: Discrimination Impact Assessment. Three measures:

- Disparate Impact (Civil Right Act 1971): $DI = \frac{\mathbb{P}(\hat{h}_n(X) = 1 | S = 0)}{\mathbb{P}(\hat{h}_n(X) = 1 | S = 1)}$
- Cond. Error Rates: $\mathbb{P}(\hat{h}_n(X) \neq Y | S = 1) = \mathbb{P}(\hat{h}_n(X) \neq Y | S = 0)$
- Equality of odds: $\mathbb{P}(\hat{h}_n(X) = 1 | S = 1)$ vs $\mathbb{P}(\hat{h}_n(X) = 1 | S = 0)$

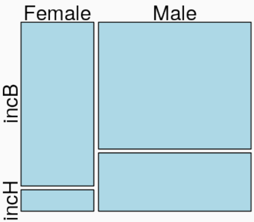
An Example in more Detail

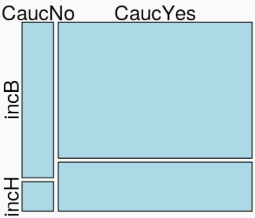
The following example is based on a Jupyter Notebook by **Philippe Besse** (INSA Toulouse) freely available (in R and python) on `https://github.com/wikistat`

Adult Census Dataset of UCI

- 48842 US citizens (1994)
- 14 features:
  - Y= income threshold ($50k)
  - **age**: continuous.
  - **workclass**: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
  - **fnlwgt**: continuous.
  - **education**: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
  - **education-num**: continuous.
  - **marital-status**: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
  - **occupation**: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
  - **relationship**: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.

Obvious Social Bias






```
Confidence interval for the DI
(by delta method)

round(dispImp(datBas[,"sex"],
datBas[,"income"]),3)

0.349 0.367 0.384
```

```
Confidence interval for the
(delta method)

round(dispImp(datBas$origEt
datBas$income),3)

0.566 0.601 0.637
```

Logistic Regression augments the bias!

```
log.lm=glm(income~.,data=datApp,family=binomial)

# significativity of the parameters
anova(log.lm,test="Chisq")

Df          Deviance          Resid. Df       Resid. Dev      Pr(>Chi)
NULL        NA      NA        35771     40371,72          NA
age         1       1927,29010        35770   38444,43        0,000000e+00
educNum 1           4289,41877        35769   34155,01        0,000000e+00
mariStat            3       6318,12804        35766   27836,88        0,000000e+00
occup       6       812,50516         35760   27024,38        3,058070e-172
origEthn            1       17,04639          35759   27007,33        3,647759e-05
sex         1       50,49872          35758   26956,83        1,192428e-12
hoursWeek           1       402,82271         35757   26554,01        1,338050e-89
LcapitalGain        1       1252,69526        35756   25301,31        2,154522e-274
LcapitalLoss        1       310,38258         35755   24990,93        1,802529e-69
child       1       87,72437          35754   24903,21        7,524154e-21

# Prevision
pred.log=predict(log.lm,newdata=daTest,type="response")
# Confusion matrix
confMat=table(pred.log>0.5,daTest$income)

incB    incH
FALSE   6190    899
TRUE    556     1298

tauxErr(confMat): 16,27

round(dispImp(daTest[,"sex"],Yhat),3) : 0.212 0.248 0.283

# Overall Accuracy Equality?
apply(table(pred.log<0.5,daTest$income,daTest$sex),3,tauxErr)

Female 91.81      Male 79.7
```

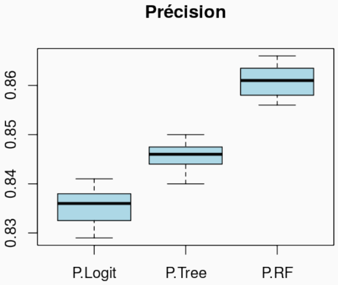
What about Random Forest?

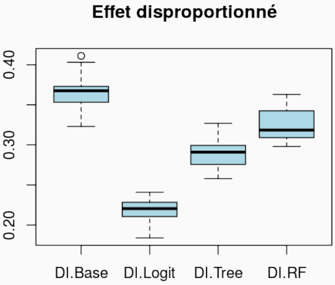
Random Forest improves significantly the predicition quality...

```
rf.mod=randomForest(income~.,data=datApp)
pred.rf=predict(rf.mod,newdata=daTest,type="response")
confMat=table(pred.rf,daTest$income)
confMat
tauxErr(confMat)


pred.rf  incB    incH
incB     6301     795
incH     445     1402

13,87


round(dispImp(daTest[,"sex"],pred.rf),3)
0.329  0.375  0.42
```

... without augmenting the bias (here).

Summary of the results by algorithm





$\implies$ Random Forest is here both more performant and less discriminative (BUT not interpretable)

$\implies$ This is not a general rule! It depends on the dataset

$\implies$ A serious learning should consider the different algorithms, and include a discussion on the discriminative effects

Individual Biases: Testing

Are the predictions changed if the value of variable "sex" is switched?

```
daTest2=daTest
# Changement de genre
daTest2$sex=as.factor(ifelse(daTest$sex=="Male","Fe
# Prevision du "nouvel" echantillon test
pred2.log=predict(log.lm,daTest2,type="response")
table(pred.log<0.5,pred2.log<0.5,daTest$sex)

Female
FALSE   TRUE
FALSE   195    0
TRUE    23     2679

Male
FALSE   TRUE
FALSE   1489   155
TRUE    0      4402
```

➔ 178 have a different prediction, in the expected direction.

Outline

Avoid Issues with Testing

Easy: use maximal prediction of all modalilities of the protected variable

```
fairPredictGenre=ifelse(pred.log<pred2.log,pred2.log
confMat=table(fairPredictGenre>0.5,daTest$income)
confMat;tauxErr(confMat)

incB     incH
FALSE    6145     936
TRUE     535      1327

16.45


round(dispImp(daTest$sex,as.factor(fairPredictGenre
0.24  0.277  0.314

# recall:
round(dispImp(daTest$sex,as.factor(pred.log>0.5)),3
0.212  0.248  0.283
```

➔ No influence on the prediction quality

➔ Small bias reduction, but does not remove group over-discrimination!

Naive approach: suppress the protected variable

```
# estimation without the variable "sex"
log_g.lm=glm(income~.,data=datApp[,-6],family=binomial)

# Prevision
pred_g.log=predict(log_g.lm,newdata=daTest[,-8],type="response")
# Confusion Matrix
confMat=table(pred_g.log>0.5,daTest$income)
confMat


      incB incH
FALSE 6157  953
TRUE   523 1310

tauxErr(confMat)

16.5



Yhat_g=as.factor(pred_g.log>0.5)
round(dispImp(daTest[,"sex"],Yhat_g),3)

0.232 0.269 0.305
```

$\Longrightarrow$ the quality of prediction is not deteriorated, but the bias augmentation remains the same!

Adapting the threshold to each class

```
Yhat_cs=as.factor(ifelse(daTest$sex=="Female",pred.log>0.4,pred.log>0.5))
round(dispImp(daTest[,"sex"],Yhat_cs),3)
tauxErr(table(Yhat_cs,daTest$income))

0.293 0.334 0.375

16.55

# Stronger correction forcing  the DI to be at least 0.8:

Yhat_cs=as.factor(ifelse(daTest$sex=="Female",pred.log>0.15,pred.log>0.5))
round(dispImp(daTest[,"sex"],Yhat_cs),3)
tauxErr(table(Yhat_cs,daTest$income))

0.796 0.863 0.93

18.57
```

$\Longrightarrow$ the prediction performance is significantly deteriorated

$\Longrightarrow$ this kind of affirmative action is a questionable choice

Building one classifier per class

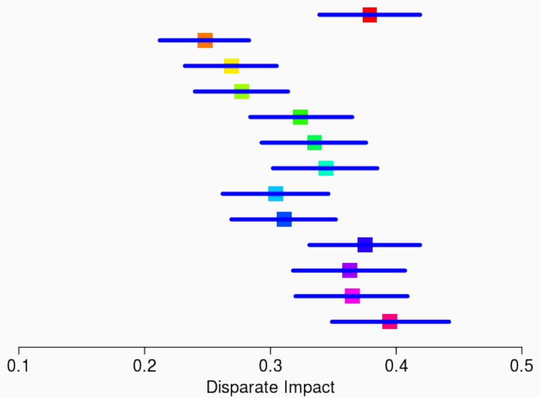
Logistic regression → consider the interactions of the protected variable with the others

```
yHat=predict(reg.log, newdata=daTest, type="response")
yHatF=predict(reg.logF, newdata=daTestF, type="response")
yHatM=predict(reg.logM, newdata=daTestM, type="response")

yHatFM=c(yHatF, yHatM);  daTestFM=rbind(daTestF, daTestM)

# Cumulated errors
table(yHatFM>0.5, daTestFM$income)
incB     incH
FALSE    6150     935
TRUE     530      1328

table(yHat>0.5, daTest$income)
incB     incH
FALSE    6154     950
TRUE     526      1313

tauxErr(table(yHatFM>0.5, daTestFM$income))
16.38

tauxErr(table(yHat>0.5, daTest$income))
16.5

# Bias with an without class separation
round(dispImp(daTestFM[,"sex"], as.factor(yHatFM>0.5)),3)
0.284  0.324  0.365

round(dispImp(daTest[,"sex"], as.factor(yHat>0.5)),3)
0.212  0.248  0.283
```

⟹ it reduces the bias

Comparison of several classifiers

Model	Accuracy
dataBaseBias	100
linLogit	83.5
linLogit_w_S	83.5
linLogit-testing	83.55
condLinLogit	83.62
quadLogit	83.54
condQuadLogit	83.32
binaryTree	85.45
wBinaryTree	85.31
condBinTree	85.16
randomForest	85.98
wRandomForest	85.91
condRandForest	85.74



Summary

- Automatic classification can *augment* the social bias
- All algorithms are not equivalent
- Linear classifiers should be particularly watched
- Random Forest can (at least sometimes) be less discriminative
- The bias augmentation diminishes with the consideration of variable interactions
- Removing the protected variable from the analysis is not sufficient
- Fitting different models on the different classes is in general a quick and simple way to avoid bias augmentation...
- ... if the protected variable is observed!

See [L'IA du Quotidien peut elle être Éthique ? : Loyauté des Algorithmes d'Apprentissage Automatique, *Besse, Castets-Renard, Garivier, Loubes*, Statistique et Société]

Outline

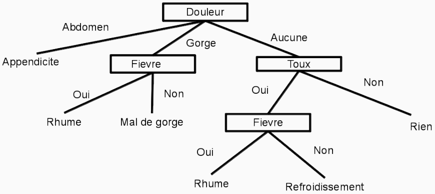
Explainability vs Interpretability

Two distinct notions (but the vocabulary is misleading: we flllow here

https://www2.eecs.berkeley.edu/Pubs/TechRpts/2017/EECS-2017-159.pdf).

A decision rule is said to be:

interpretable if we understand how a prediction is associated to an observation; typical example: decision tree



http://www.up2.fr/

explainable if we understand what feature values led to the prediction, possibly by a counterfactual analysis; for example: "if variable X_3 had taken that other value, then the prediction would have been different".

Explainability vs Interpretability

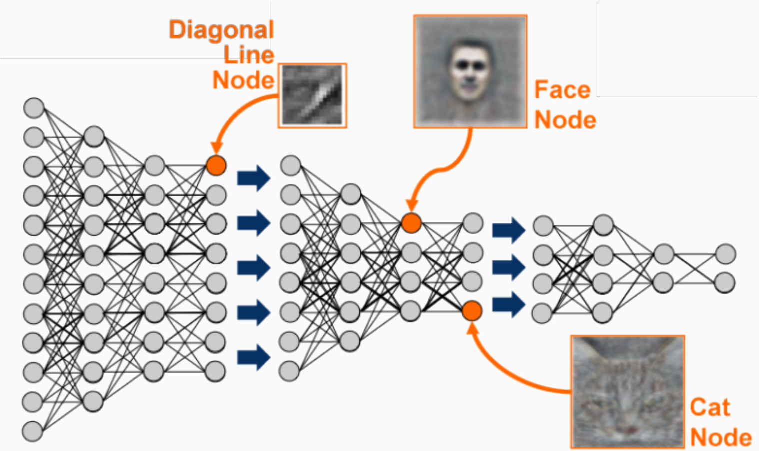
Two distinct notions (but the vocabulary is misleading: we flllow here
https://www2.eecs.berkeley.edu/Pubs/TechRpts/2017/EECS-2017-159.pdf).

A decision rule is said to be:

interpretable if we understand how a prediction is associated to an observation; typical example: decision tree

explainable if we understand what feature values led to the prediction, possibly by a counterfactual analysis; for example: "if variable X_3 had taken that other value, then the prediction would have been different".

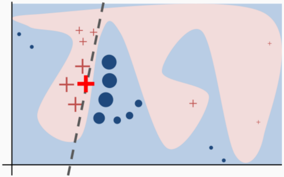
Expainability relates to the statistical notions of *causal inference* and *sensibility analysis*



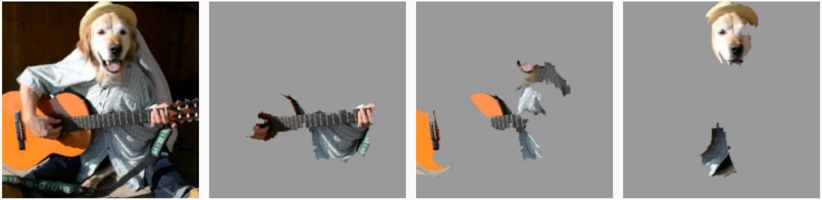
http://aiehive.com

An audacious scientific bet...

Local Interpretable Model-Agnostic Explanations: LIME



Linear model with feature selection on local subset of data



(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Src: "Why Should I Trust You?" Explaining the Predictions of Any Classifier, by Marco Tulio Ribeiro, Sameer Singh and Carlos Guestrin.

Local Interpretable Model-Agnostic Explanations: LIME

(a) Husky classified as wolf (b) Explanation

Figure 11: Raw data and explanation of a bad model's prediction in the "Husky vs Wolf" task.

Src: "Why Should I Trust You?" Explaining the Predictions of Any Classifier, by Marco Tulio Ribeiro, Sameer Singh and Carlos Guestrin.

Conclusion

- Huge need for more research and good practice
- Not only average performance matters
- Fairness should be included in data analysis with human impact
- Important issues that everyone should be aware of
- Interesting experiments to run at every level