Use & Value of Unusual Data

Arthur Charpentier

UQAM

Actuarial Summer School 2019

The Team for the Week

Jean-Philippe Boucher UQAM (Montréal, Canada)
Professor, holder of The Co-operators Chair in Actuarial Risk Analysis, author of several research articles in actuarial science.

@J_P_Boucher jpboucher.uqam.ca

Arthur Charpentier UQAM (Montréal, Canada)
Professor, editor of *Computational Actuarial Science with R*. Former director of the Data Science for Actuaries program of the French Institute of Actuaries

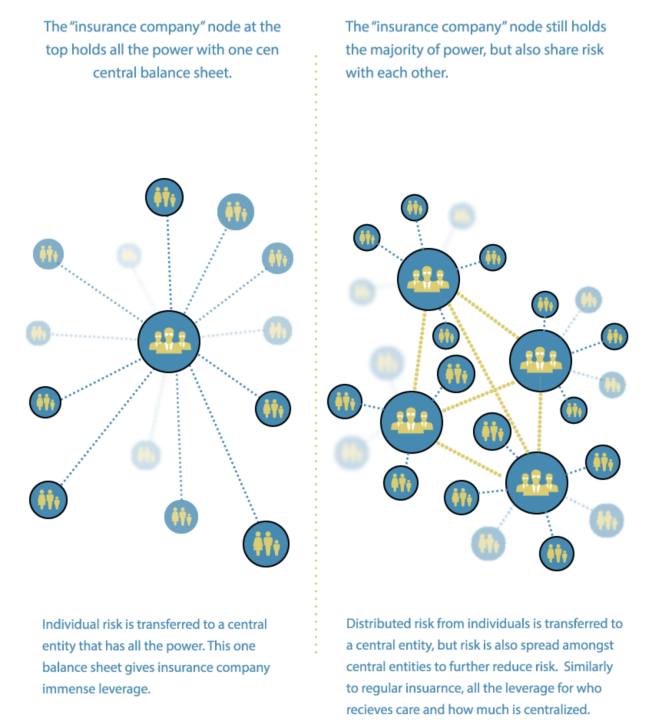
@freakonometrics freakonometrics freakonometrics.hypotheses.org

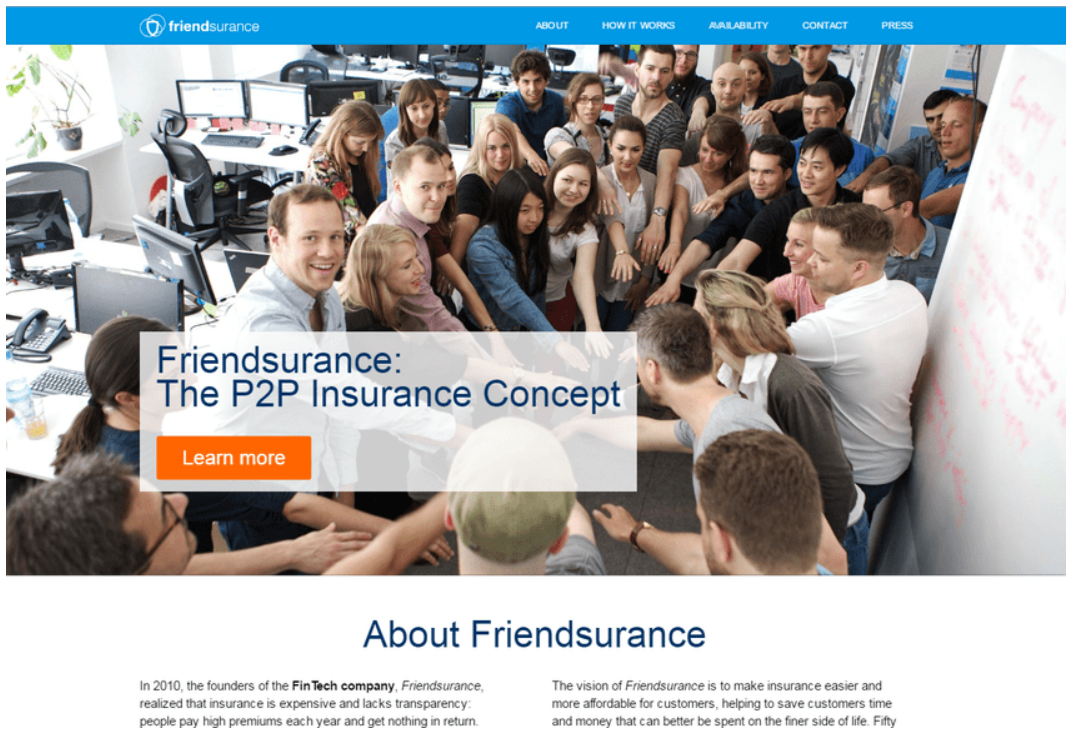
Ewen Gallic AMSE (Aix-Marseille, France)
Assistant professor, teaching computational econometrics, data science and machine learning.

@3wen 3wen egallic.fr

Practical Issues

General Context (in Insurance)





Networks and P2P Insurance (via https://sharable.life and source)

General Context (in Insurance)

GOING BACK TO BASICS

Innovators have set their sights on simplifying insurance by adopting new peer-to-peer business models. Who are the key players and does the Australian industry need to pay attention?


Janis Meyer-Plath, Sebastian Herfurth and Tim Kunde of Friendsurance

DISCUSSION OF the sharing economy has taken up considerable column space in recent times.

PricewaterhouseCoopers estimates the five main sharing sectors (peer-to-peer finance, online staffing, peer-to-peer accommodation, car sharing and music video streaming) will potentially generate global revenues of US$335bn (A$439bn) by 2025. Today, PwC says revenue generated is around US$15bn (A$19.67bn).

One sharing economy sector still very much in its infancy is peer-to-peer insurance. But Amy Gibbs, digital communications and content strategy manager at ANZIIF, says popularity of the concept is increasing.

"When it takes off, it will likely happen much quicker than we expect," she says.

"P2P insurance is not about a new technology threatening an industry but about people demanding an industry that gives them what they believe they deserve. Consumers expect different things in 2016, and technology now allows them the power to get what they want or go elsewhere."

The current crop in P2P

On 24 March, Germany's high-profile P2P player, Friendsurance, announced it had collected US$15.3m (A$20.26m) from investors in its latest round of financing.

Tim Kunde, Friendsurance's co-founder and managing director, says the organisation intends to use that fresh capital to grow further in Germany and expand internationally. He says the first expansion target for 2016 is Australia, and expansion opportunities for other markets are currently being considered.

Friendsurance is one of the players Gibbs has been keeping a close eye on.

"Friendsurance and Guevara are ones that I watch keenly," she says.

"They have interesting models and appear to have put a lot of thought into them, plus they get the marketing/consumer angle, which is crucial."

Friendsurance, founded in Germany in 2010, operates as an independent insurance broker. It describes its mission as to make insurance easier and more affordable for customers, and to reduce the number of fraudulent claims.

"Our idea is inspired by insurance in its original form, when people got together in small groups ... and supported each other in case of damage," Tim Kunde, Friendsurance's co-founder and managing director, tells *Insurance Business*.

"This was easy and efficient but also limited in the extent of coverage. Today, big insurance companies can carry claims of any amount, but marketing, administration and fraud cause remarkable costs.

"P2P insurance is not about a new technology threatening an industry but about people demanding an industry that gives them what they believe they deserve"

Amy Gibbs, ANZIIF

"Against this background, we developed an insurance concept that again [creates] smaller groups within bigger insurance societies and rewards remaining claimless within [these groups] with annual cashbacks."

Friendsurance customers with the same insurance type form small groups online. Part of their insurance premium is paid into a cashback pool, and part of it is provided to the group's insurer (or reinsurer).

When small claims are made, customers are reimbursed from the general fund. Larger claims still go through the insurer. Groups that have no claims during a year receive a cashback bonus the following January. Friendsurance says its claims-free bonus is available for private liability, home contents and legal expenses insurance.

"So far, more than 80% of users received some of their insurance fees back," says Kunde.

"In the property insurance line, the average cashback was 33% of the paid insurance fees."

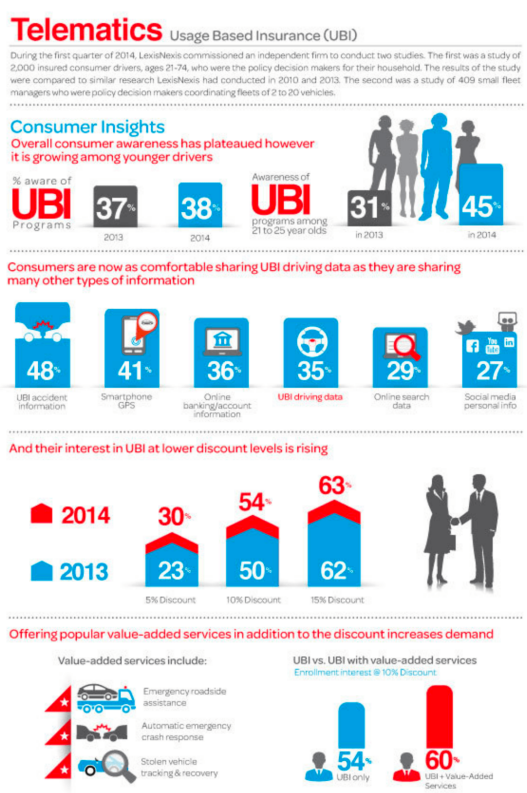
Kunde reports that, in 2015, Friendsurance engaged 75,000 new customers.

"Today, we have a six-digit number of customers, 70 insurance partners, 15 copycats and 80 employees."

Over in the UK, start-up Guevara has received

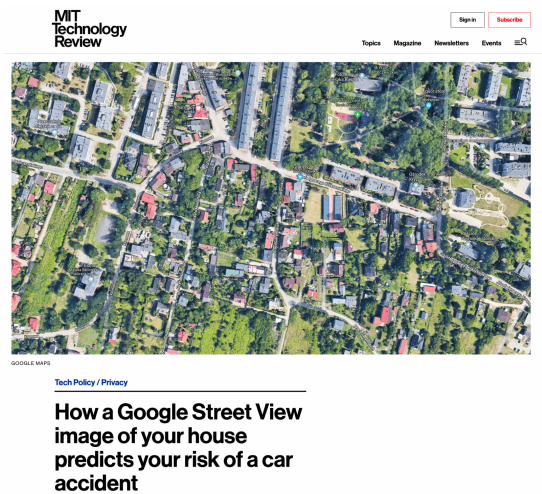
Networks and P2P Insurance, via Insurance Business (2016)

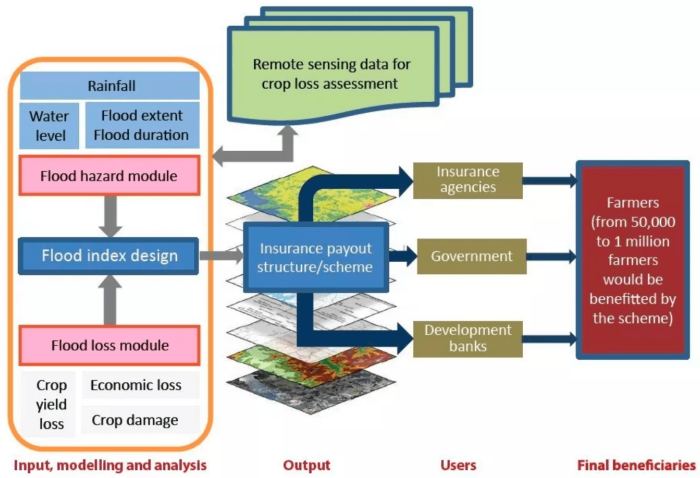
General Context (in Insurance)



Telematics and Usage-Based-Insurance (source, source, The Actuary)

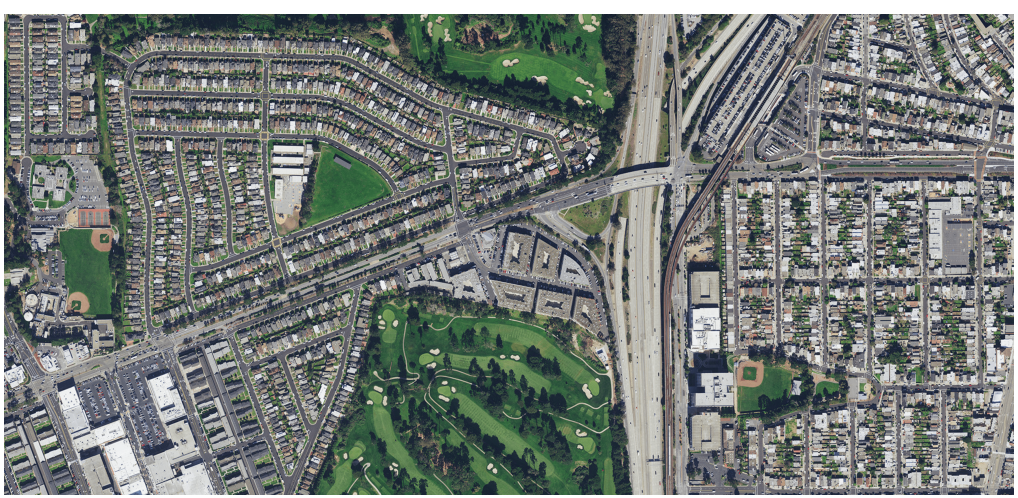
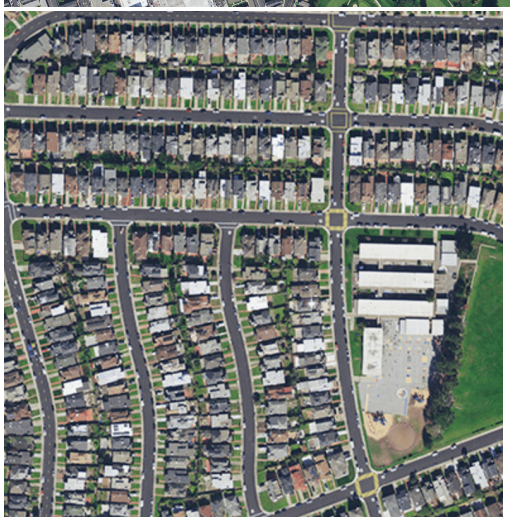
General Context (in Insurance)





Satellite pictures (source and Index-based Flood Insurance (IBFI), http://ibfi.iwmi.org/)

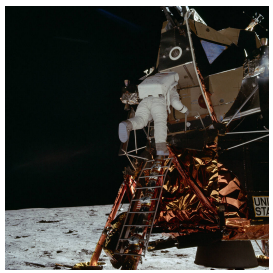
General Context (in Insurance)

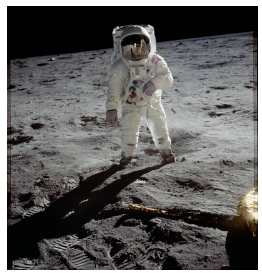
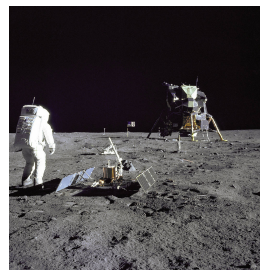
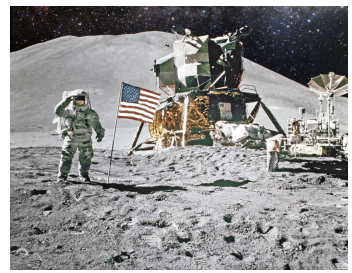




Very small granularity (1 pixel is $15 \times 15\text{cm}^2$, see detail)

More and More (exponentially) Data

1969, (legend says) 5 photographs were taken by Neil Armstrong



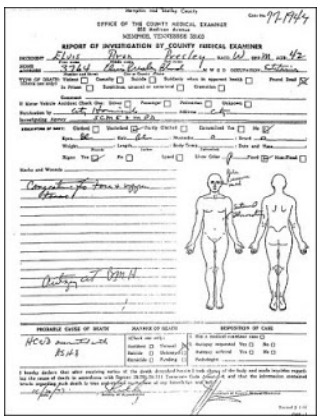




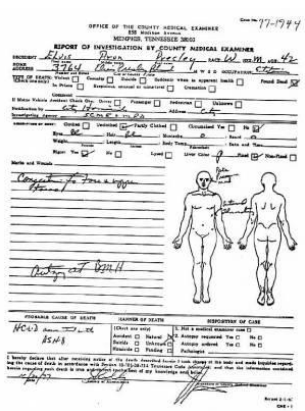
More than 800,000,000 pictures are uploaded on every day, on Facebook (2 billion photos per day with instagram, Messenger and Whatsapp)

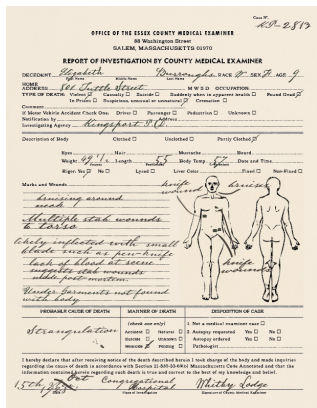


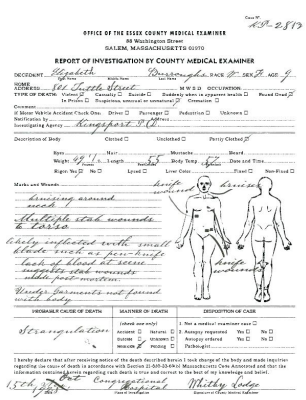

Data Complexity

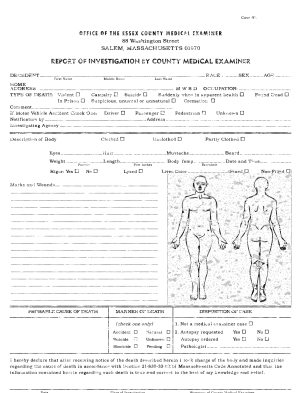
Pictures, drawings, videos, handwriting text, ECG, medical imagery, satellite pictures, tweets, phone calls, etc.





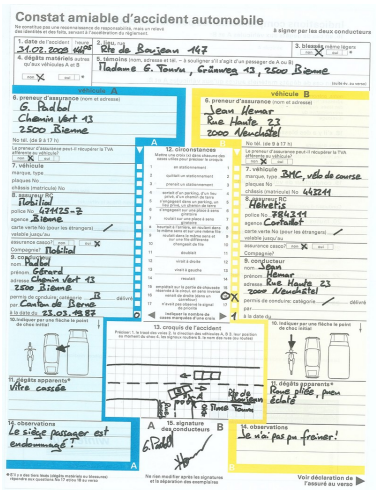


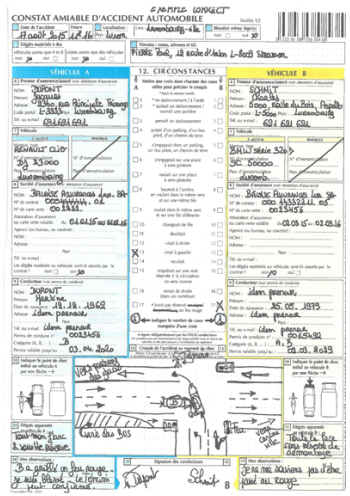


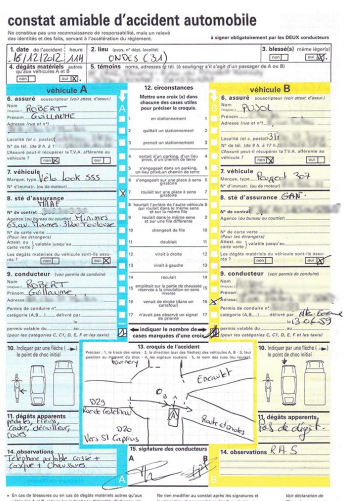


Hard to distinguish structured and un-structured data...

Data Complexity

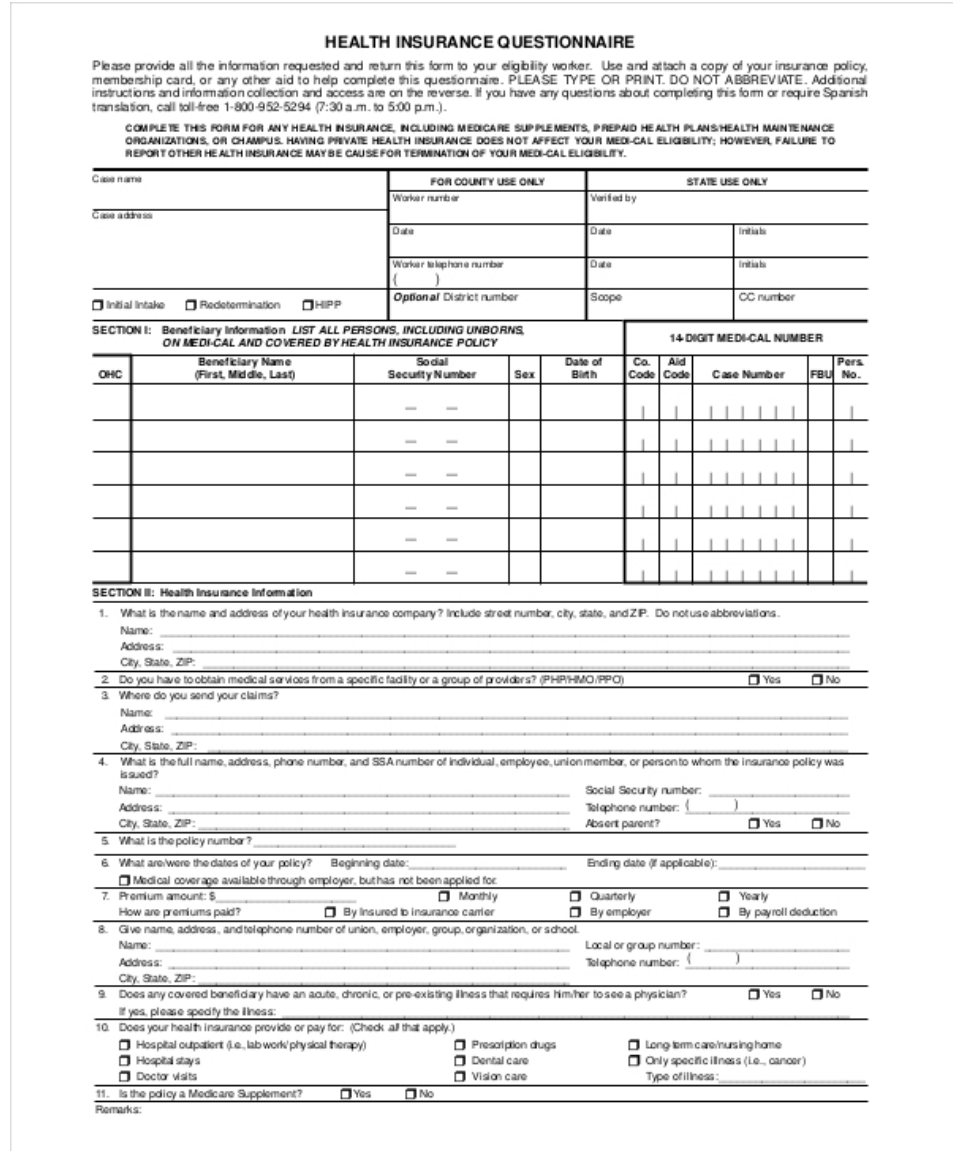






Insurance Data

From questionnaires
to data frames $x_{i,j}$
row i : insured, policy, claim
column j : variables (features)



HEALTH INSURANCE QUESTIONNAIRE

Please provide all the information requested and return this form to your eligibility worker. Use and attach a copy of your insurance policy, membership card, or any other aid to help complete this questionnaire. PLEASE TYPE OR PRINT. DO NOT ABBREVIATE. Additional instructions and information collection and access are on the reverse. If you have any questions about completing this form or require Spanish translation, call toll-free 1-800-952-5294 (7:30 a.m. to 5:00 p.m.).

COMPLETE THIS FORM FOR ANY HEALTH INSURANCE, INCLUDING MEDICARE SUPPLEMENTS, PREPAID HEALTH PLANS/HEALTH MAINTENANCE ORGANIZATIONS, OR CHAMPUS. HAVING PRIVATE HEALTH INSURANCE DOES NOT AFFECT YOUR MEDI-CAL ELIGIBILITY; HOWEVER, FAILURE TO REPORT OTHER HEALTH INSURANCE MAY BE CAUSE FOR TERMINATION OF YOUR MEDI-CAL ELIGIBILITY.

Case name
Case address

FOR COUNTY USE ONLY		STATE USE ONLY	
Worker number		Verified by	
Date		Date	Initials
Worker telephone number ( )		Date	Initials
Optional District number		Scope	CC number

☐ Initial Intake ☐ Redetermination ☐ HIPP

SECTION I: Beneficiary Information *LIST ALL PERSONS, INCLUDING UNBORNS, ON MEDI-CAL AND COVERED BY HEALTH INSURANCE POLICY*

OHC	Beneficiary Name (First, Middle, Last)	Social Security Number	Sex	Date of Birth	14-DIGIT MEDI-CAL NUMBER: Co. Code	Aid Code	Case Number	FBU	Pers. No.
		– –							
		– –							
		– –							
		– –							
		– –							
		– –							

SECTION II: Health Insurance Information

1. What is the name and address of your health insurance company? Include street number, city, state, and ZIP. Do not use abbreviations.
   Name:
   Address:
   City, State, ZIP:
2. Do you have to obtain medical services from a specific facility or a group of providers? (PHP/HMO/PPO) ☐ Yes ☐ No
3. Where do you send your claims?
   Name:
   Address:
   City, State, ZIP:
4. What is the full name, address, phone number, and SSA number of individual, employee, union member, or person to whom the insurance policy was issued?
   Name: Social Security number:
   Address: Telephone number: ( )
   City, State, ZIP: Absent parent? ☐ Yes ☐ No
5. What is the policy number?
6. What are/were the dates of your policy? Beginning date: Ending date (if applicable):
   ☐ Medical coverage available through employer, but has not been applied for.
7. Premium amount: $ ☐ Monthly ☐ Quarterly ☐ Yearly
   How are premiums paid? ☐ By insured to insurance carrier ☐ By employer ☐ By payroll deduction
8. Give name, address, and telephone number of union, employer, group, organization, or school.
   Name: Local or group number:
   Address: Telephone number: ( )
   City, State, ZIP:
9. Does any covered beneficiary have an acute, chronic, or pre-existing illness that requires him/her to see a physician? ☐ Yes ☐ No
   If yes, please specify the illness:
10. Does your health insurance provide or pay for: (Check all that apply.)
    ☐ Hospital outpatient (i.e., lab work/physical therapy) ☐ Prescription drugs ☐ Long-term care/nursing home
    ☐ Hospital stays ☐ Dental care ☐ Only specific illness (i.e., cancer)
    ☐ Doctor visits ☐ Vision care Type of illness:
11. Is the policy a Medicare Supplement? ☐ Yes ☐ No

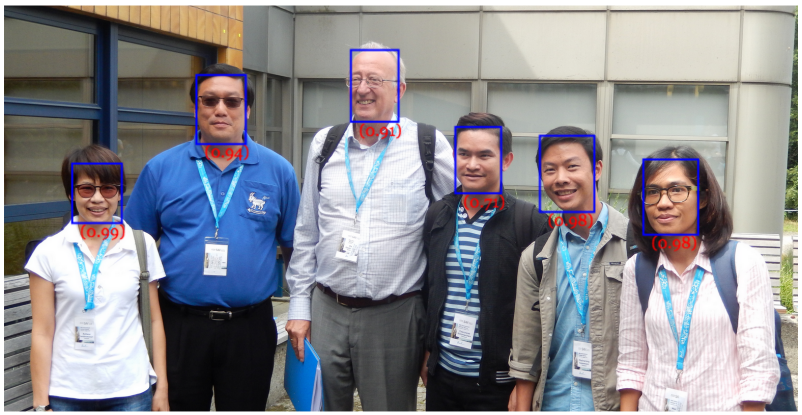
Remarks:

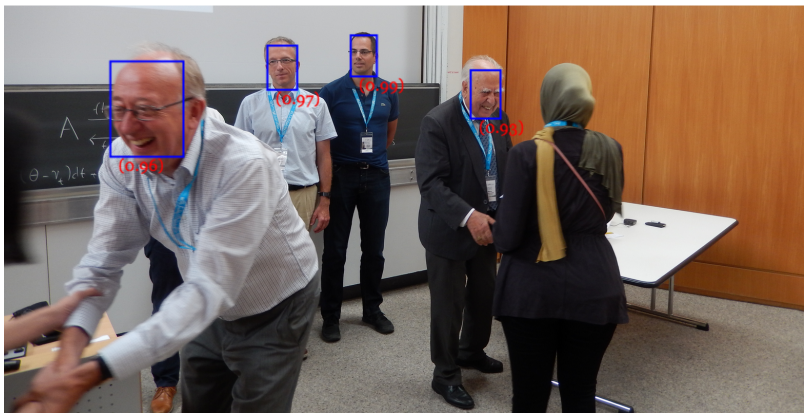
From Pictures to Data

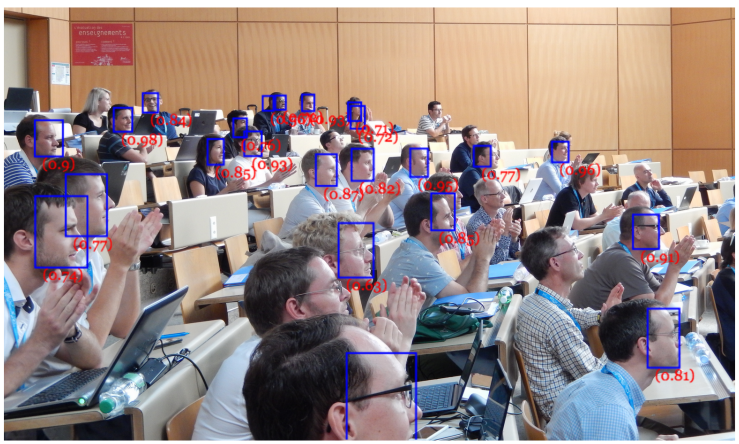
Consider pictures saa-iss.ch/testalbum/nggallery/all-old-galleries/

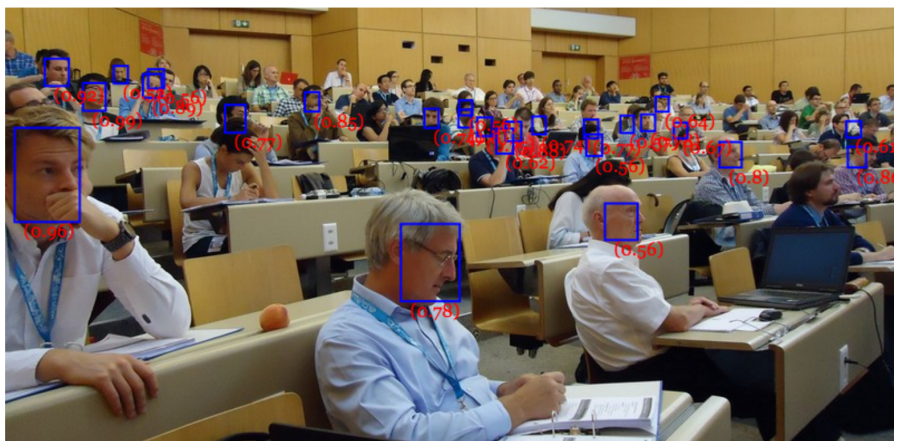
From Pictures to Data

Face detection face-api.js (justadudewhohacks.github.io)



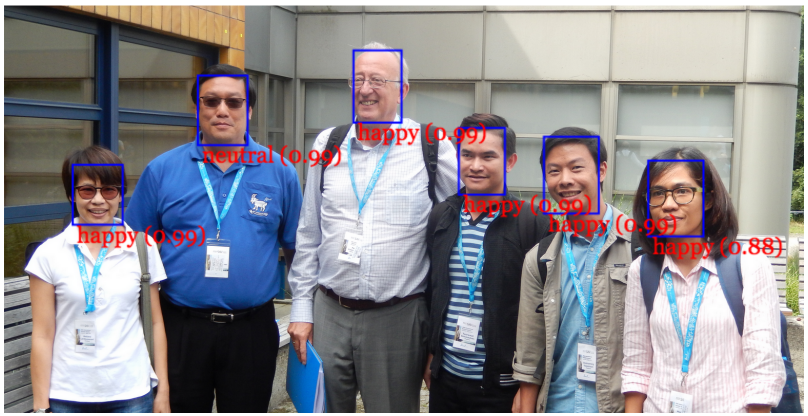


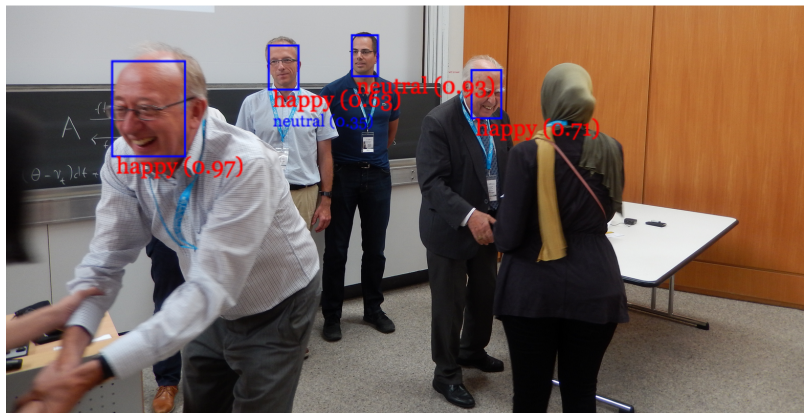


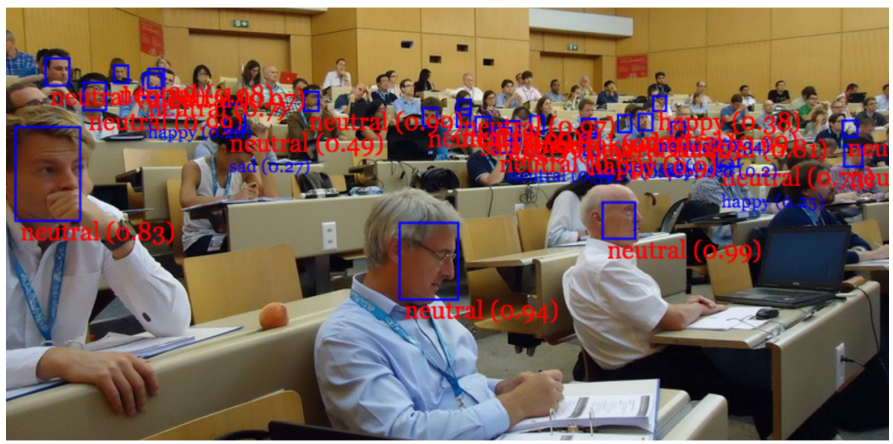


From Pictures to Data

Expression recognition face-api.js (justadudewhohacks.github.io)

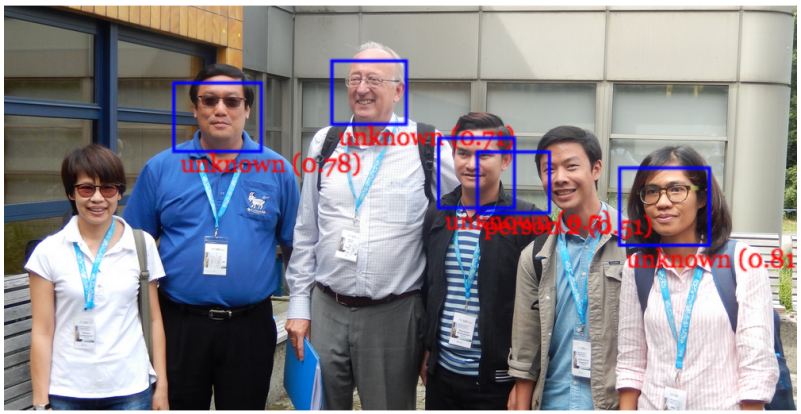


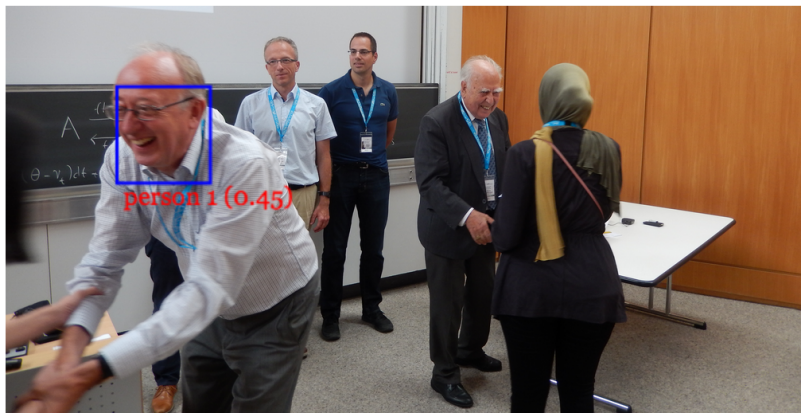




From Pictures to Data

Face recognition face-api.js (justadudewhohacks.github.io)

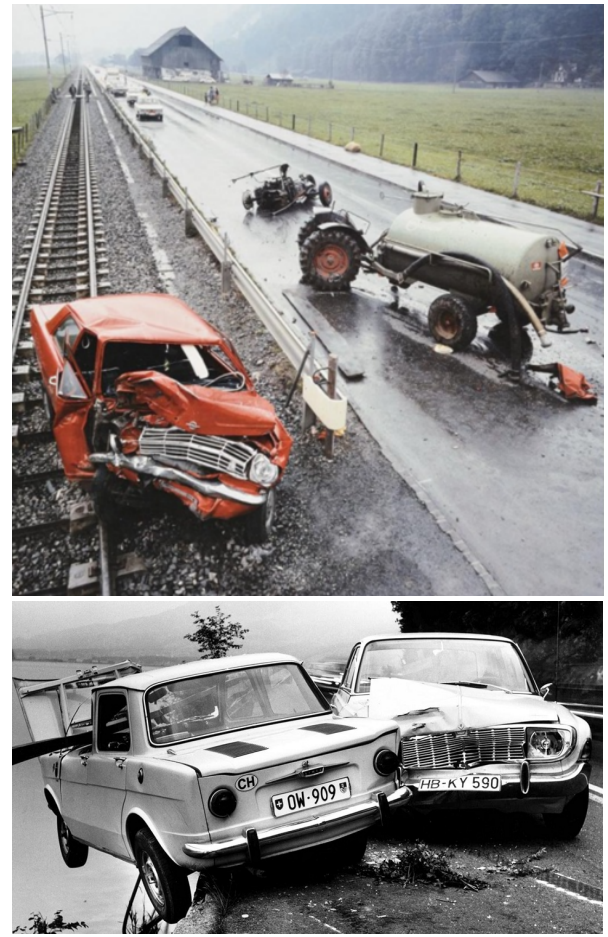




Pictures

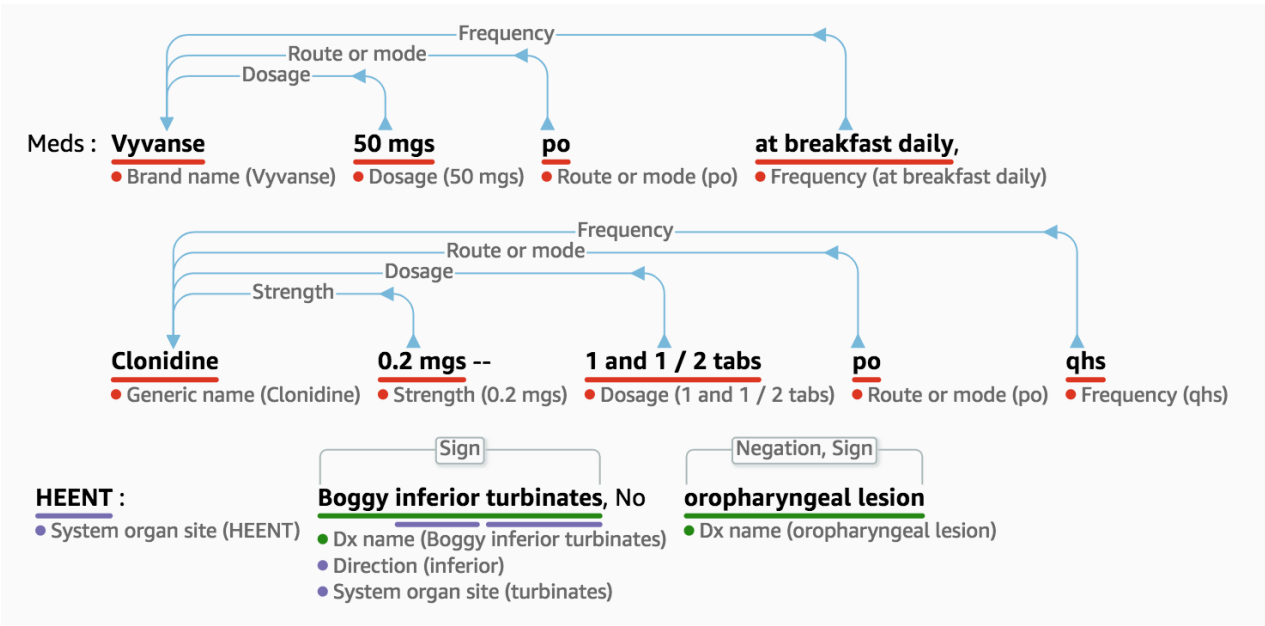
Extracting information from a picture
What is on the picture ?
What is the cost of such a claim ?
High dimensional data :
$h \times d$ color picture, e.g. 850×1280
$\boldsymbol{x}_i \in \mathbb{R}^d$ with $d > 3$ million
Need new statistical tools
(deep) neural networks perform well
or reduce dimension

Pictures Odermatts (2003, Karambolage)



Text

Kaufman *et al.* (2016, Natural Language Processing-Enabled and Conventional Data Capture Methods for Input to Electronic Health Records), Chen *et al.* (2018, A Natural Language Processing System That Links Medical Terms in Electronic Health Record Notes to Lay Definitions) or Simon (2018, Natural Language Processing for Healthcare Customers)

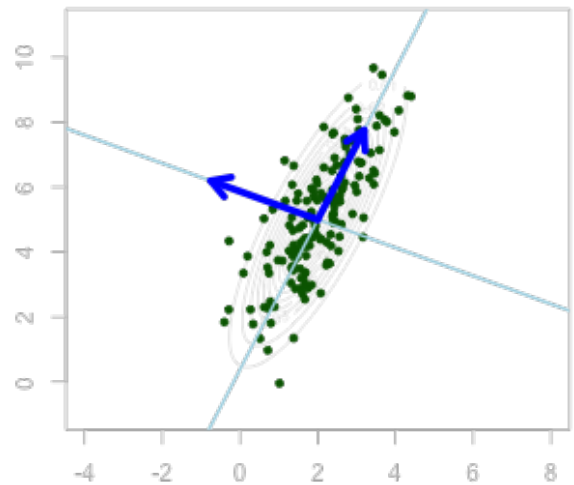


Statistics & ML Toolbox : PCA

Projection method, used to reduce dimensionality

Principal Component Analysis (PCA)

find a small number of directions in input space that explain variation in input data represent data by projecting along those directions



n observations in dimension d, stored in matrix $\boldsymbol{X} \in \mathcal{M}_{n \times d}$
Natural ideal: linearly project (multiply by a projection matrix) to much lower dimensional space $k < d$
Search for orthogonal directions in space with highest variance
Information is stored in the covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{X}^\top \boldsymbol{X}$ when variables are centered

Statistics & ML Toolbox : PCA

Find the k largest eigenvectors of $\boldsymbol{\Sigma}$: principal components

Assemble these eigenvectors into a $d \times k$ matrix $\tilde{\boldsymbol{\Sigma}}$
Express d-dimensional vectors $\boldsymbol{x}$ by projecting them to m-dimensional $\tilde{\boldsymbol{x}}$, $\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{\Sigma}}^\top \boldsymbol{x}$
We have data $\boldsymbol{X} \in \mathcal{M}_{n \times d}$, i.e. n vectors $\boldsymbol{x}_j \in \mathbb{R}^d$,
Let $\boldsymbol{\omega}_1 \in \mathbb{R}^d$ denote weights.
We want to maximize the variance of $\boldsymbol{z} = \boldsymbol{\omega}_1^\top \boldsymbol{x}$

$$\max \left\{ \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{\omega}_1^\top \boldsymbol{x}_i - \boldsymbol{\omega}_1^\top \overline{x})^2 \right\} = \max \left\{ \boldsymbol{\omega}_1^\top \boldsymbol{\Sigma} \boldsymbol{\omega}_1 \right\} \text{ s.t. } \|\boldsymbol{\omega}_1\| = 1$$

Statistics & ML Toolbox : PCA

To solve

$$\max\left\{\frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{\omega}_1^\top \boldsymbol{x}_i - \boldsymbol{\omega}_1^\top \overline{x})^2\right\} = \max\left\{\boldsymbol{\omega}_1^\top \boldsymbol{\Sigma}\boldsymbol{\omega}_1\right\} \text{ s.t. } \|\boldsymbol{\omega}_1\| = 1$$

we can use Lagrange multiplier to solve this constraint optimization problem

$$\max_{\boldsymbol{\omega}_1,\lambda_1}\left\{\boldsymbol{\omega}_1^\top \boldsymbol{\Sigma}\boldsymbol{\omega}_1 + \lambda(1-\boldsymbol{\omega}_1^\top \boldsymbol{\omega}_1)\right\}$$

If we differentiate $\boldsymbol{\Sigma}\boldsymbol{\omega}_1 - \lambda_1\boldsymbol{\omega}_1 = \mathbf{0}$, i.e. $\boldsymbol{\Sigma}\boldsymbol{\omega}_1 = \lambda_1\boldsymbol{\omega}_1$ i.e. $\boldsymbol{\omega}_1$ is an eigenvector of $\boldsymbol{\Sigma}$, with eigenvalue the Lagrange multiplier. It should be the one with the largest eigenvalue.

Statistics & ML Toolbox : PCA

$$\max \left\{ \boldsymbol{\omega}_2^\top \boldsymbol{\Sigma} \boldsymbol{\omega}_2 \right\}$$
$$\text{subject to } \|\boldsymbol{\omega}_2\| = 1$$
$$\boldsymbol{\omega}_2^\top \boldsymbol{\omega}_1 = 0$$

The Lagragian is $\boldsymbol{\omega}_2^\top \boldsymbol{\Sigma} \boldsymbol{\omega}_2 + \lambda_2(1 - \boldsymbol{\omega}_2^\top \boldsymbol{\omega}_2) - \lambda_1 \boldsymbol{\omega}_2^\top \boldsymbol{\omega}_1$
when differentiating, we get $\boldsymbol{\Sigma} \boldsymbol{\omega}_2 = \lambda_2 \boldsymbol{\omega}_2$
λ_2 is the second largest eigenvalue
Set $\boldsymbol{z} = W\boldsymbol{x}$ and $\tilde{\boldsymbol{x}} = M\boldsymbol{z}$. We want to solve (where ℓ is the standard ℓ_2 norm)

$$\min_{W,M} \left\{ \sum_{i=1}^{n} \ell(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_i) \right\}$$

i.e.

$$\min_{W,M} \left\{ \sum_{i=1}^{n} \ell(x_i, MWx_i) \right\}$$

Statistics & ML Toolbox : PCA

$$\min_{W,M} \left\{ \sum_{i=1}^{n} \ell(x_i, MWx_i) \right\}$$

for some $n \times k$ (for W) and $k \times n$ (for M) matrices, with $k < n$, see Plaut (2018, From Principal Subspaces to Principal Components with Linear Autoencoders).
More generally,

$$\min_{W,M} \left\{ \sum_{i=1}^{n} \ell(x_i, \phi \circ \psi x_i) \right\}$$

for some nonlinear functions ϕ and ψ.
This is nonlinear PCA (see Monahan (2016, An Introduction to Nonlinear Principal Component Analysis)
See also autoencoder (neural nets).

Statistics & ML Toolbox : GLM

Generalized Linear Model

Flexible generalization of linear regression that allows for y to have distribution other than a normal distribution

The *classical* regression is (matrix form)

$$\underset{n\times 1}{\boldsymbol{y}} = \underset{n\times k}{\boldsymbol{X}}\underset{k\times 1}{\beta} + \underset{n\times 1}{\varepsilon} \text{ or } y_i = \underbrace{\boldsymbol{x}_i^\top \beta}_{\langle \boldsymbol{x}_i, \beta\rangle} + \varepsilon_i$$

$$Y_i \sim \mathcal{N}(\mu_i, \sigma^2) \quad , \text{ with } \mu_i = \mathbb{E}[Y_i|\boldsymbol{x}_i] = \boldsymbol{x}_i^\top \beta$$

It is possible to consider a more general regression.
For a classification $\boldsymbol{y} \in \{0,1\}^n$,

$$Y_i \sim \mathcal{B}(p_i) \quad , \text{ with } p_i = \mathbb{E}[Y_i|\boldsymbol{x}_i] = \frac{e^{\boldsymbol{x}_i^\top \beta}}{1 + e^{\boldsymbol{x}_i^\top \beta}}$$

Recall that we assumed that $(Y_i, \boldsymbol{X}_i)$ were i.i.d. with unknown distribution $\mathbb{P}$.

Statistics & ML Toolbox : GLM

In the (standard) linear model, we assume that $(Y|\boldsymbol{X} = \boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{x}^\top \beta, \sigma^2)$.
The maximum-likelihood estimator of β is then

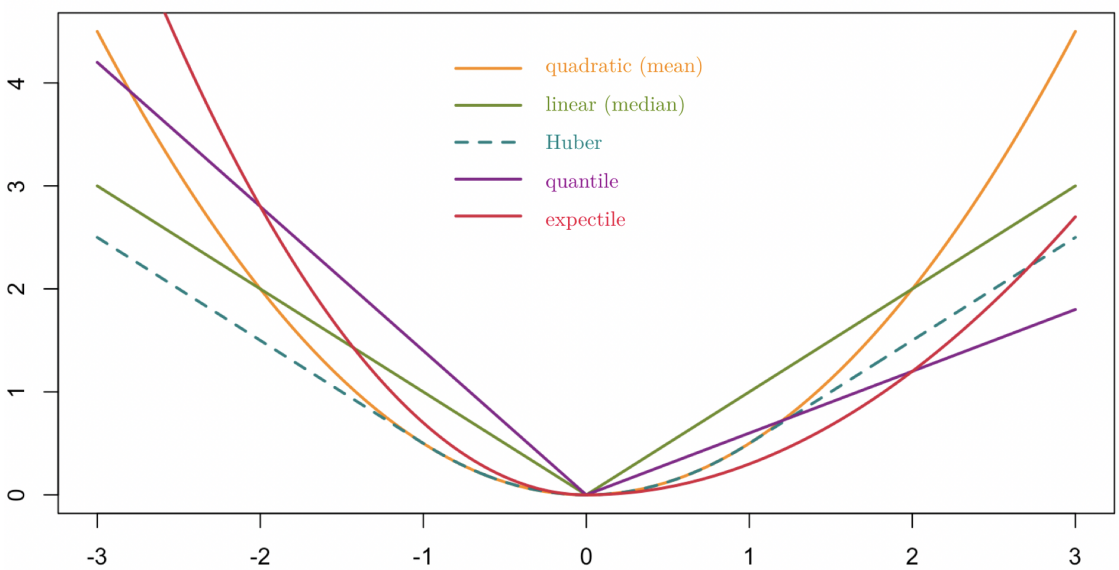
$$\widehat{\beta} = \underset{\beta \in \mathbb{R}^{p+1}}{\text{argmax}} \left\{ -\sum_{i=1}^{n} (y_i - \boldsymbol{x}_i^\top \beta)^2 \right\}$$

For the logistic regression, $(Y|\boldsymbol{X} = \boldsymbol{x}) \sim \mathcal{B}(\text{logit}^{-1}(\boldsymbol{x}^\top \beta))$.
The maximum-likelihood estimator of β is then

$$\underset{\beta \in \mathbb{R}^{p+1}}{\text{argmin}} \left\{ \sum_{i=1}^{n} \left(y_i \log(\text{logit}^{-1}(\boldsymbol{x}_i^\top \beta)) + (1 - y_i) \log(1 - \text{logit}^{-1}(\boldsymbol{x}_i^\top \beta)) \right) \right\}$$

Statistics & ML Toolbox : loss and risk

In a general setting, consider some loss function $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$, e.g. $\ell(y, m) = (y - m)^2$ (quadratic ℓ_2 norm)



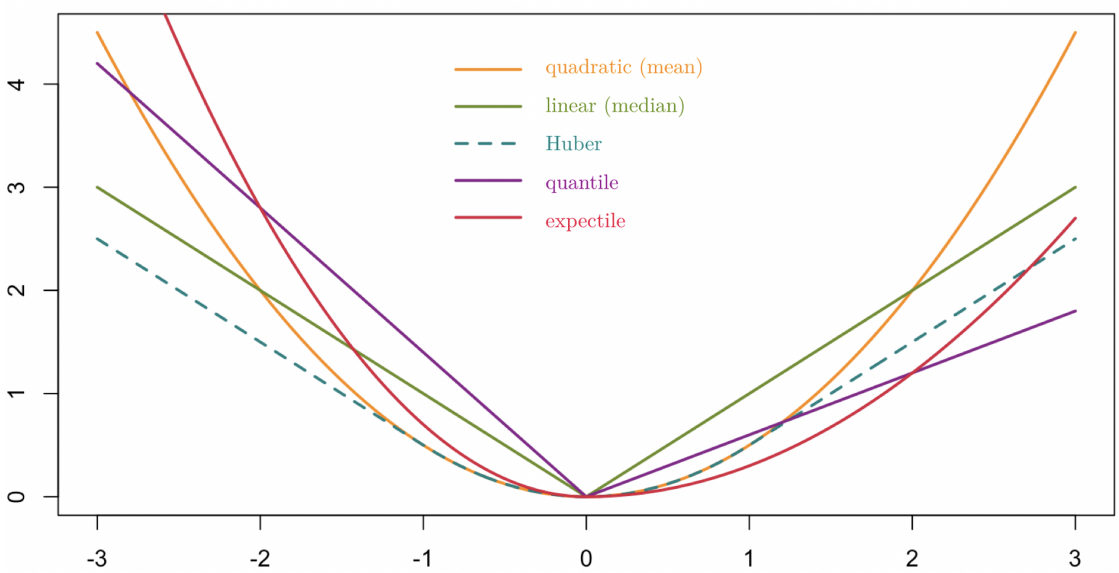
Average Risk

The average risk associated with $\widehat{m}_n$ is

$$\mathcal{R}_{\mathbb{P}}(\widehat{m}_n) = \mathbb{E}_{\mathbb{P}}\big[\ell(Y, \widehat{m}_n(\boldsymbol{X}))\big]$$

Statistics & ML Toolbox : loss and risk

In a general setting, consider some loss function $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$, e.g. $\ell(y, m) = (y - m)^2$ (quadratic ℓ_2 norm)



Empirical Risk

The empirical risk associated with $\widehat{m}_n$ is

$$\widehat{\mathcal{R}}_n(\widehat{m}_n) = \frac{1}{n} \sum_{i=1}^{n} \ell\big(y_i, \widehat{m}_n(\boldsymbol{x}_i)\big).$$

Statistics & ML Toolbox : loss and risk

If we minimize the average risk, we overfit...

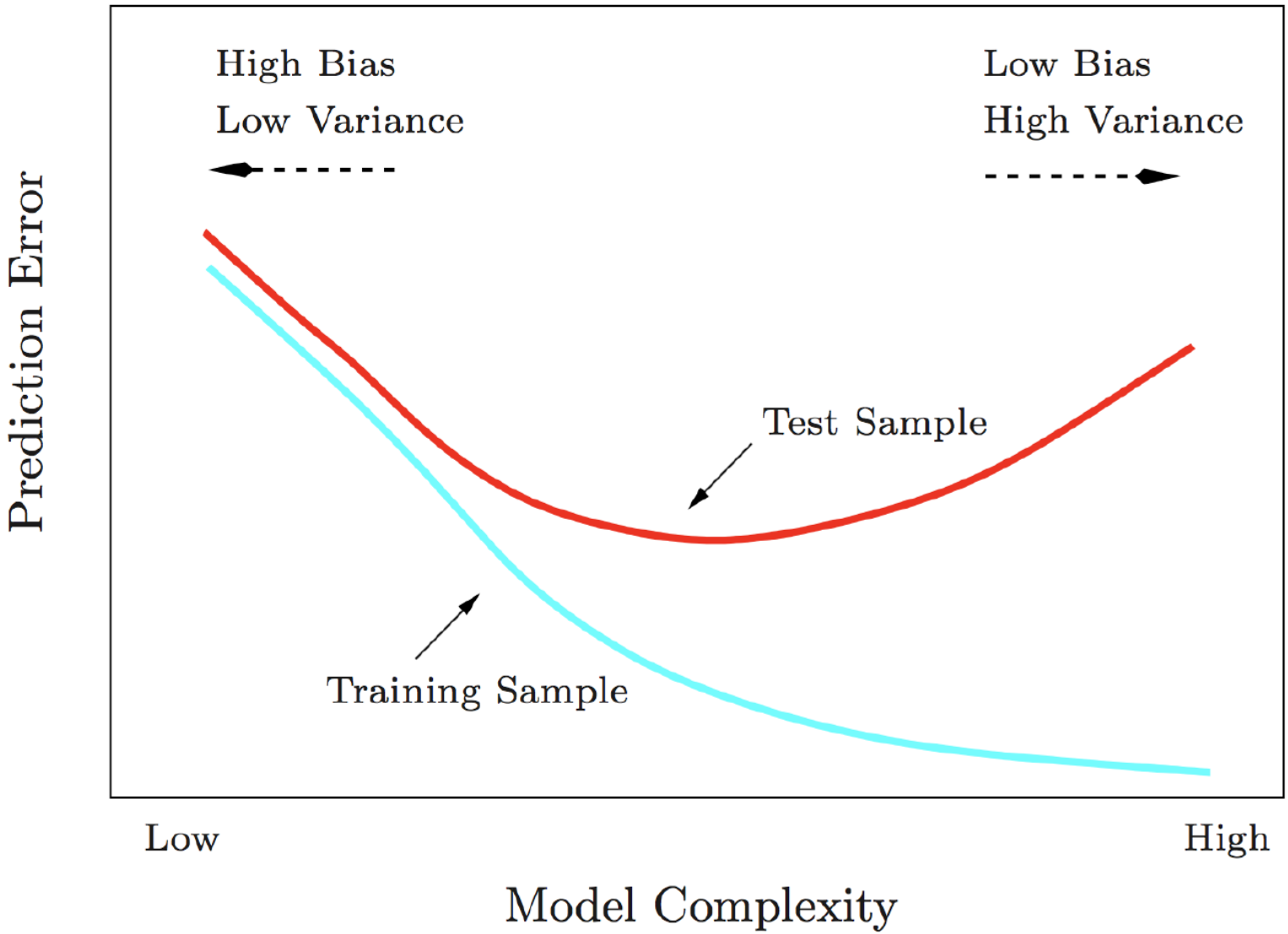
Hold-Out Cross Validation

1. Split $\{1, 2, \cdots, n\}$ in T (training) and V (validation)
2. Estimate $\widehat{m}$ on sample $(y_i, \boldsymbol{x}_i)$, $i \in T$:

$$\widehat{m}_T = \underset{m \in \mathcal{M}}{\operatorname{argmin}} \frac{1}{|T|} \sum_{i \in V} \ell(y_i, m(x_{1,i}, \cdots, x_{p,i}))$$

3. Compute $\dfrac{1}{|V|} \displaystyle\sum_{i \in V} \ell(y_i', \widehat{m}_T(x_{1,i''}, \cdots, x_{p,i'}))$

Statistics & ML Toolbox : loss and risk



source: Hastie *et al.* (2009, The Elements of Statistical Learning)

Statistics & ML Toolbox : loss and risk

Leave-one-Out Cross Validation

1. Estimate n models : estimate $\widehat{m}_{-j}$ on sample $(y_i, \boldsymbol{x}_i)$, $i \in \{1, \cdots, n\} \backslash \{j\}$

$$\widehat{m}_{-j} = \underset{m \in \mathcal{M}}{\text{argmin}} \frac{1}{n-1} \sum_{i \neq j} \ell(y_i, m(x_{1,i}, \cdots, x_{p,i}))$$

2. Compute $\dfrac{1}{n} \displaystyle\sum_{i=1}^{n} \ell(y_i, \widehat{m}_{-i}(\boldsymbol{x}_i))$

Statistics & ML Toolbox : loss and risk

K-Fold Cross Validation

1. Split $\{1, 2, \cdots, n\}$ in K groups $V_1, \cdots, V_K$
2. Estimate K models : estinate $\widehat{m}_k$ on sample $(y_i, \boldsymbol{x}_i)$, $i \in \{1, \cdots, n\} \backslash V_k$
3. Compute $$\frac{1}{K} \sum_{k=1}^{K} \frac{1}{|V_k|} \sum_{i \in V_k} \ell(y_i, \widehat{m}_k(x_{1,i}, \cdots, X_{p,i}))$$

Statistics & ML Toolbox : loss and risk

Bootstrap Cross Validation

1. Generate B boostrap samples from $\{1, 2, \cdots, n\}$, $I_1, \cdots, I_B$
2. Estimate B models : $\widehat{m}_b$ on sample $(y_i, \boldsymbol{x}_i)$, $i \in I_b$
3. Compute $$\frac{1}{B}\sum_{b=1}^{B}\frac{1}{n-|I_b|}\sum_{i\notin I_b}\ell(y_i, \widehat{m}_b(x_{1,i}, \cdots, X_{p,i}))$$

The probability that $i \notin I_b$ is $\left(1-\frac{1}{n}\right)^n \sim e^{-1}(= 36.78\%)$

At stage b, we validate on $\sim 36.78\%$ of the dataset.

Statistics & ML Toolbox : loss and risk

$$\boldsymbol{y} \in \mathbb{R}^d, \overline{y} = \underset{m \in \mathbb{R}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \frac{1}{n} \underbrace{[y_i - m]}_{\varepsilon_i}^2 \right\} \text{ is the empirical version of}$$

$$\mathbb{E}[Y] = \underset{m \in \mathbb{R}}{\operatorname{argmin}} \left\{ \int \underbrace{[y - m]}_{\varepsilon}^2 dF(y) \right\} = \underset{m \in \mathbb{R}}{\operatorname{argmin}} \left\{ \mathbb{E}[\| \underbrace{Y - m}_{\varepsilon} \|_{\ell_2}] \right\}$$

where Y is a random variable.

Thus, $\underset{m:\mathbb{R}^k \to \mathbb{R}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \frac{1}{n} \underbrace{[y_i - m(\boldsymbol{x}_i)]}_{\varepsilon_i}^2 \right\}$ is the empirical version of $\mathbb{E}[Y|\boldsymbol{X} = \boldsymbol{x}]$.

See Legendre (1805, Nouvelles méthodes pour la détermination des orbites des comètes) and Gauβ (1809, Theoria motus corporum coelestium in sectionibus conicis solem ambientium).

Statistics & ML Toolbox : loss and risk

$$\text{med}[\boldsymbol{y}] \in \underset{m\in\mathbb{R}}{\text{argmin}} \left\{ \sum_{i=1}^{n} \frac{1}{n} |\underbrace{y_i - m}_{\varepsilon_i}| \right\} \text{ is the empirical version of}$$

$$\text{med}[Y] \in \underset{m\in\mathbb{R}}{\text{argmin}} \left\{ \int |\underbrace{y - m}_{\varepsilon}| dF(y) \right\} = \underset{m\in\mathbb{R}}{\text{argmin}} \left\{ \mathbb{E}[\|\underbrace{Y - m}_{\varepsilon}\|_{\ell_1}] \right\}$$

where $\mathbb{P}[Y \leq \text{med}[Y]] \geq \frac{1}{2}$ and $\mathbb{P}[Y \geq \text{med}[Y]] \geq \frac{1}{2}$.

$$\underset{m:\mathbb{R}^k\to\mathbb{R}}{\text{argmin}} \left\{ \sum_{i=1}^{n} \frac{1}{n} |\underbrace{y_i - m(\boldsymbol{x}_i)}_{\varepsilon_i}| \right\} \text{ is the empirical version of}$$

$\text{med}[Y|\boldsymbol{X} = \boldsymbol{x}]$.

See Boscovich (1757, De Litteraria expeditione per pontificiam ditionem ad dimetiendos duos meridiani) and Laplace (1793, Sur quelques points du système du monde).

Statistics & ML Toolbox : loss and risk

For least-squares (ℓ_2 norm), we have a strictly convex problem... classical optimization routines can be used (e.g. gradient descient)
More complicated with the ℓ_1 norm
Consider a sample $\{y_1, \cdots, y_n\}$.

To compute the median, solve $\min_{\mu} \left\{ \sum_{i=1}^{n} |y_i - \mu| \right\}$ which can be solved using linear programming techniques., see Dantzig (1963, Linear programming).

More precisely, this problem is equivalent to $\min_{\mu, \boldsymbol{a}, \boldsymbol{b}} \left\{ \sum_{i=1}^{n} a_i + b_i \right\}$

with $a_i, b_i \geq 0$ and $y_i - \mu = a_i - b_i$, $\forall i = 1, \cdots, n$.

Statistics & ML Toolbox : loss and risk

For a quantile, the linear program is $\min_{\mu,\boldsymbol{a},\boldsymbol{b}} \left\{ \sum_{i=1}^{n} \tau a_i + (1-\tau) b_i \right\}$
with $a_i, b_i \geq 0$ and $y_i - \mu = a_i - b_i$, $\forall i = 1, \cdots, n$.
This is extended to the quantile regression,

$$\min_{\beta^\tau,\boldsymbol{a},\boldsymbol{b}} \left\{ \sum_{i=1}^{n} \tau a_i + (1-\tau) b_i \right\}$$

with $a_i, b_i \geq 0$ and $y_i - \boldsymbol{x}_i^\top \beta^\tau = a_i - b_i$, $\forall i = 1, \cdots, n$.

But is the following problem the one we should really care about ?

$$m^\star = \underset{m \in \mathcal{M}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \ell(y_i, m(\boldsymbol{x}_i)) \right\}$$

Statistics & ML Toolbox : Penalization

Usually, there are tuning parameters p. To avoid overfit,

1. use cross-validation

$$m^\star_{p^\star} = \underset{p}{\text{argmin}} \left\{ \sum_{i \in \text{validation}} \ell(y_i, m^\star_p(\boldsymbol{x}_i)) \right\}$$

$$\text{where } m^\star_p = \underset{m \in \mathcal{M}_p}{\text{argmin}} \left\{ \sum_{i \in \text{training}} \ell(y_i, m(\boldsymbol{x}_i)) \right\}$$

2. use penalization

$$m^\star = \underset{m \in \mathcal{M}}{\text{argmin}} \left\{ \sum_{i=1}^{n} \ell(y_i, m(\boldsymbol{x}_i)) + \text{penalization}(m) \right\}$$

Statistics & ML Toolbox : Penalization

Penalization often yield bias... but it might interesting if the risk is the mean squared error...

Consider a parametric model, with true (unknown) parameter θ, then

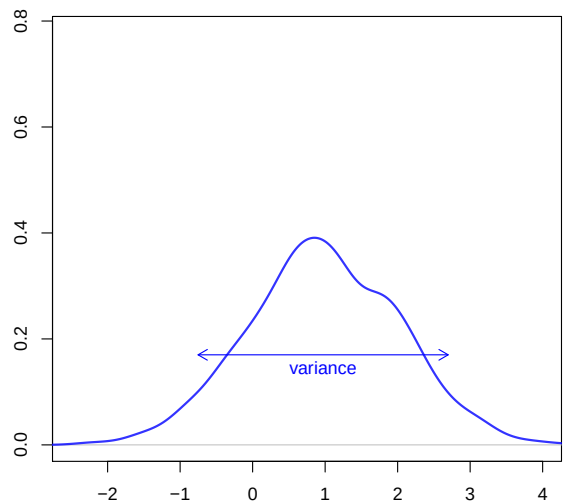
$$\text{mse}(\hat{\theta}) = \mathbb{E}\left[(\hat{\theta} - \theta)^2\right] = \underbrace{\mathbb{E}\left[(\hat{\theta} - \mathbb{E}[\hat{\theta}])^2\right]}_{\text{variance}} + \underbrace{\mathbb{E}\left[(\mathbb{E}[\hat{\theta}] - \theta)^2\right]}_{\text{bias}^2}$$

One can think of a shrinkage of an unbiased estimator $\widetilde{\theta}$, $\widehat{\theta} = \alpha \cdot \widetilde{\theta}$

Let $\widetilde{\theta}$ denote an unbiased estimator of θ,

$$\hat{\theta} = \frac{\theta^2}{\theta^2 + \text{mse}(\widetilde{\theta})} \cdot \widetilde{\theta} \leq \widetilde{\theta}$$

satisfies $\text{mse}(\hat{\theta}) \leq \text{mse}(\widetilde{\theta})$.



Statistics & ML Toolbox : Ridge

as in Wieringen (2018 Lecture notes on ridge regression)

Ridge Estimator (OLS)

$$\widehat{\boldsymbol{\beta}}_{\lambda}^{\text{ridge}} = \text{argmin} \left\{ \sum_{i=1}^{n} (y_i - \boldsymbol{x}_i^{\top} \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}$$

$$\widehat{\boldsymbol{\beta}}_{\lambda}^{\text{ridge}} = (\boldsymbol{X}^{\mathsf{T}} \boldsymbol{X} + \lambda \mathbb{I})^{-1} \boldsymbol{X}^{\mathsf{T}} \boldsymbol{y}$$

Ridge Estimator (GLM)

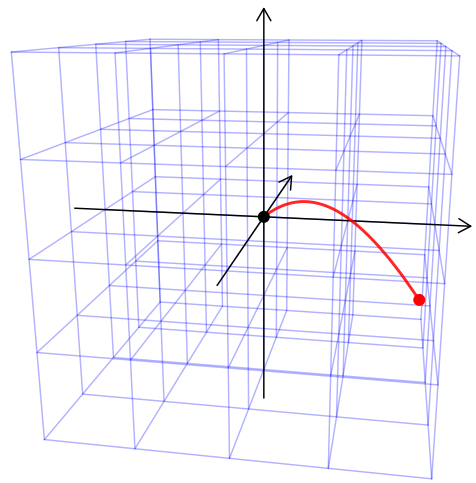
$$\widehat{\boldsymbol{\beta}}_{\lambda}^{\text{ridge}} = \text{argmin} \left\{ - \sum_{i=1}^{n} \log f(y_i | \mu_i = g^{-1}(\boldsymbol{x}_i^{\top} \boldsymbol{\beta})) + \frac{\lambda}{2} \sum_{j=1}^{p} \beta_j^2 \right\}$$

Statistics & ML Toolbox : Ridge

Observe that if $\boldsymbol{x}_{j_1} \perp \boldsymbol{x}_{j_2}$, then

$$\widehat{\beta}_\lambda^{\text{ridge}} = [1+\lambda]^{-1}\widehat{\beta}^{\text{ols}}$$

which explain relationship with shrinkage.
But not in the general case...



* chicago.txt dataset, with three covariates

Smaller mse

There exists λ such that $\text{mse}[\widehat{\beta}_\lambda^{\text{ridge}}] \leq \text{mse}[\widehat{\beta}^{\text{ols}}]$

It is important to normalize variables. We assume here that we did.

Statistics & ML Toolbox : Ridge

From a Bayesian perspective,

$$\underbrace{\mathbb{P}[\boldsymbol{\theta}|\boldsymbol{y}]}_{\text{posterior}} \propto \underbrace{\mathbb{P}[\boldsymbol{y}|\boldsymbol{\theta}]}_{\text{likelihood}} \cdot \underbrace{\mathbb{P}[\boldsymbol{\theta}]}_{\text{prior}} \quad \text{i.e.} \quad \log \mathbb{P}[\boldsymbol{\theta}|\boldsymbol{y}] = \underbrace{\log \mathbb{P}[\boldsymbol{y}|\boldsymbol{\theta}]}_{\text{log likelihood}} + \underbrace{\log \mathbb{P}[\boldsymbol{\theta}]}_{\text{penalty}}$$

If β has a prior $\mathcal{N}(\mathbf{0}, \tau^2\mathbb{I})$ distribution, then its posterior distribution has mean

$$\mathbb{E}[\beta|\boldsymbol{y}, \boldsymbol{X}] = \left(\boldsymbol{X}^\mathsf{T}\boldsymbol{X} + \frac{\sigma^2}{\tau^2}\mathbb{I}\right)^{-1} \boldsymbol{X}^\mathsf{T}\boldsymbol{y}.$$

Statistics & ML Toolbox : LASSO

LASSO Estimator (OLS)

$$\widehat{\beta}_\lambda^{\text{lasso}} = \text{argmin} \left\{ \sum_{i=1}^{n} (y_i - \boldsymbol{x}_i^\top \beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

LASSO Estimator (GLM)

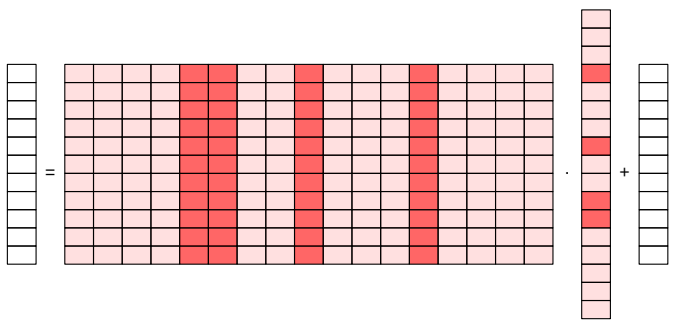
$$\widehat{\beta}_\lambda^{\text{lasso}} = \text{argmin} \left\{ -\sum_{i=1}^{n} \log f(y_i | \mu_i = g^{-1}(\boldsymbol{x}_i^\top \beta)) + \frac{\lambda}{2} \sum_{j=1}^{p} |\beta_j| \right\}$$

* least absolute shrinkage and selection operator
(pun with *garrote*, Breiman (1995, Better Subset Regression Using the Nonnegative Garrote)
see Bayesian regression when β has a Laplace prior.

Statistics & ML Toolbox : sparcity

In several applications, k can be (very) large, but a lot of features are just noise: $\beta_j = 0$ for many j's. Let s denote the number of relevant features, with $s << k$, cf Hastie, Tibshirani & Wainwright (2015, Statistical Learning with Sparsity),

$$s = \text{card}\{\mathcal{S}\} \text{ where } \mathcal{S} = \{j; \beta_j \neq 0\}$$



The model is now $y = \boldsymbol{X}_{\mathcal{S}}^{\mathsf{T}}\beta_{\mathcal{S}} + \varepsilon$, where $\boldsymbol{X}_{\mathcal{S}}^{\mathsf{T}}\boldsymbol{X}_{\mathcal{S}}$ is a full rank matrix.

Statistics & ML Toolbox : sparcity

ℓ_0-regularization Estimator (OLS)

$$\widehat{\beta}_\lambda^{\ell_0} = \text{argmin} \left\{ \sum_{i=1}^{n} (y_i - \boldsymbol{x}_i^\top \beta)^2 + \lambda \|\beta\|_{\ell_0} \right\}$$

where $\|\boldsymbol{a}\|_{\ell_0} = \sum \mathbf{1}(|a_j| > 0) = \dim(\beta)$.

More generally

ℓ_p-regularization Estimator (OLS)

$$\widehat{\beta}_\lambda^{\ell_p} = \text{argmin} \left\{ \sum_{i=1}^{n} (y_i - \boldsymbol{x}_i^\top \beta)^2 + \lambda \|\beta\|_{\ell_p} \right\}$$

- sparsity is obtained when $p \leq 1$
- convexity is obtained when $p \geq 1$

Statistics & ML Toolbox : sparcity

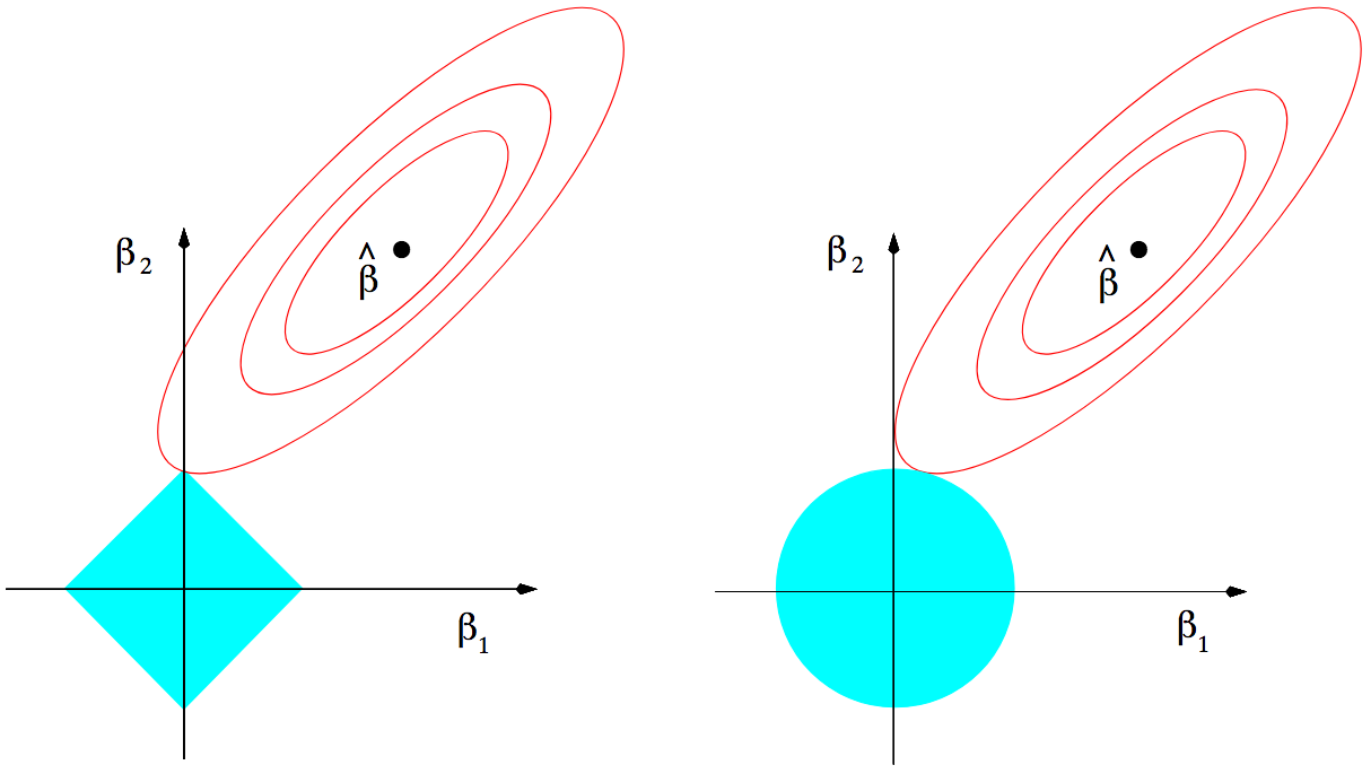


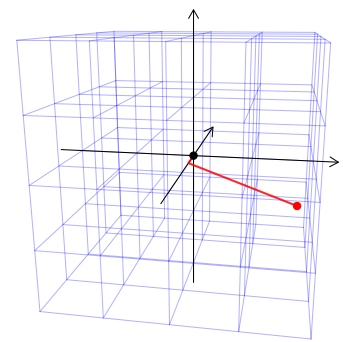
FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*

source: Hastie *et al.* (2009, The Elements of Statistical Learning)

Statistics & ML Toolbox : LASSO

Penalized regression is seen here as contrained optimization (objective function is a Lagrangian - see Kuhn-Tucker Theorem)
In the orthogonal case, $\boldsymbol{X}^{\mathsf{T}}\boldsymbol{X} = \mathbb{I}$,

$$\widehat{\beta}_{k,\lambda}^{\text{lasso}} = \text{sign}(\widehat{\beta}_k^{\text{ols}}) \left(|\widehat{\beta}_k^{\text{ols}}| - \frac{\lambda}{2} \right)_+$$



i.e. the LASSO estimate is related to the soft threshold function

$$S_h(x) = \text{sign}(x)(|x| - h)$$

* chicago.txt dataset, with three covariates

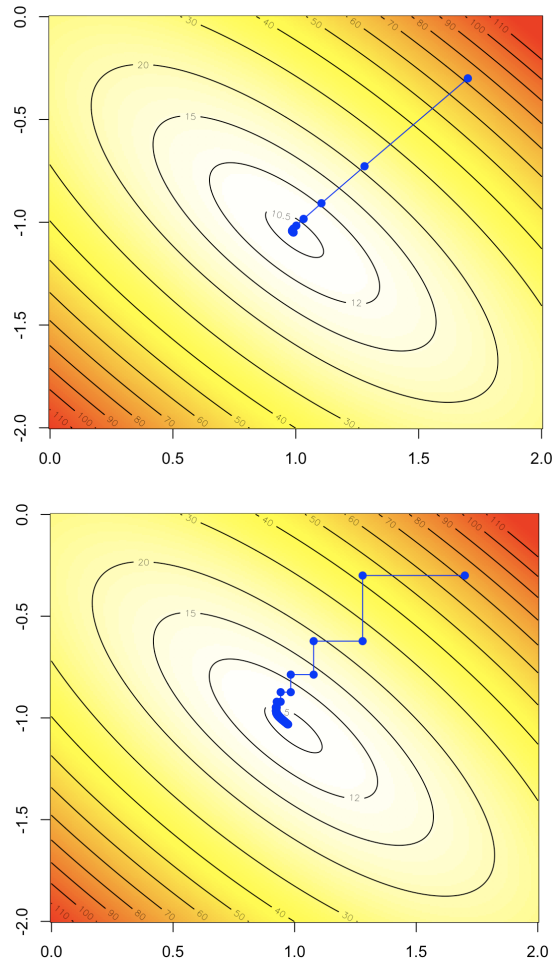
See https://freakonometrics.hypotheses.org/52894 for codes.

Numerical Toolbox

LASSO Coordinate Descent Algorithm

1. Set $\beta_0 = \widehat{\beta}$
2. For $k = 1, \cdots$
   for $j = 1, \cdots, p$
   (i) compute $R_j = \boldsymbol{x}_j^\top \left(\boldsymbol{y} - \boldsymbol{X}_{-j}\beta_{k-1(-j)}\right)$
   (ii) set $\beta_{k,j} = R_j \cdot \left(1 - \frac{\lambda}{2|R_j|}\right)_+$
3. The final estimate β_κ is $\widehat{\beta}_\lambda$

The softmax function $\log(1 + e^{x-s})$ can be used as a smooth version of $(x - s)_+$



Statistics & ML Toolbox : trees

Use recursive binary splitting to grow a tree

Trees (CART)

Each interior node corresponds to one of the input variables; there are edges to children for each of the possible values of that input variable. Each leaf represents a value of the target variable given the values of the input variables represented by the path from the root to the leaf.

based on some impurity index, with, for leaf ℓ, and $f(p) = p(1-p)$

$$I(\ell) = \sum_{y \in \{0,1\}} \frac{n_{y,\ell}}{n_\ell} \cdot f\left(\frac{n_{y,\ell}}{n_\ell}\right) \text{ with no split}$$

$$I(\ell_L, \ell_R) = \sum_{x \in \{L,R\}} \sum_{y \in \{0,1\}} \frac{n_{y,\ell_x}}{n_{\ell x}} \cdot f\left(\frac{n_{y,\ell_x}}{n_{\ell x}}\right) \text{ with split}$$

Statistics & ML Toolbox : forests

Bagging (Bootstrap + Aggregation)

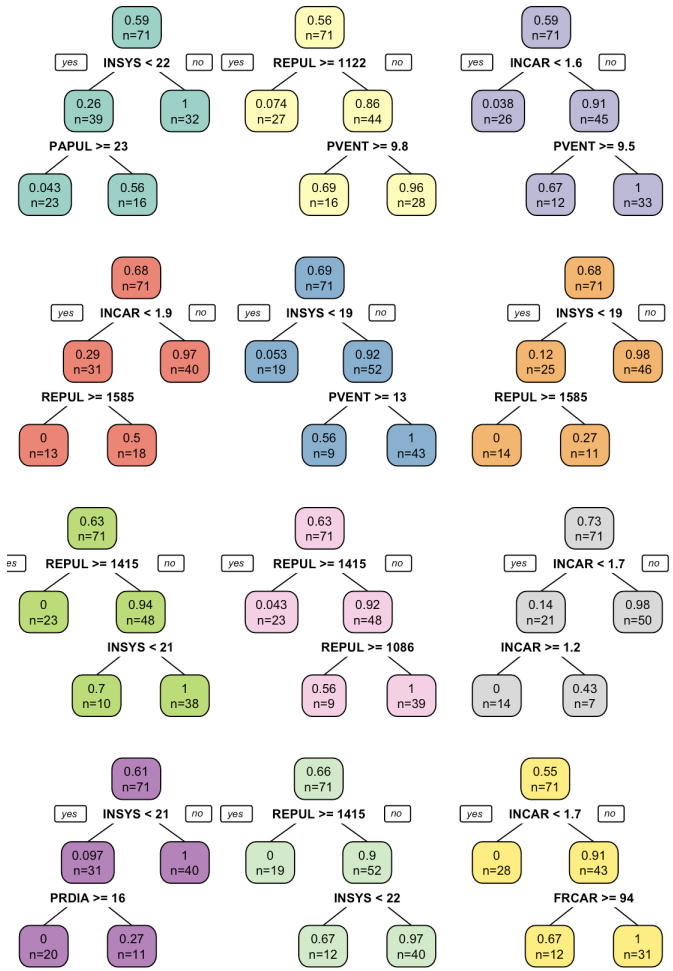
1. For $k = 1, \cdots$

   (i) draw a bootstrap sample from $(y_i, \boldsymbol{x}_i)$'s

   (ii) estimate a model $\widehat{m}_k$ on that sample

2. The final model is

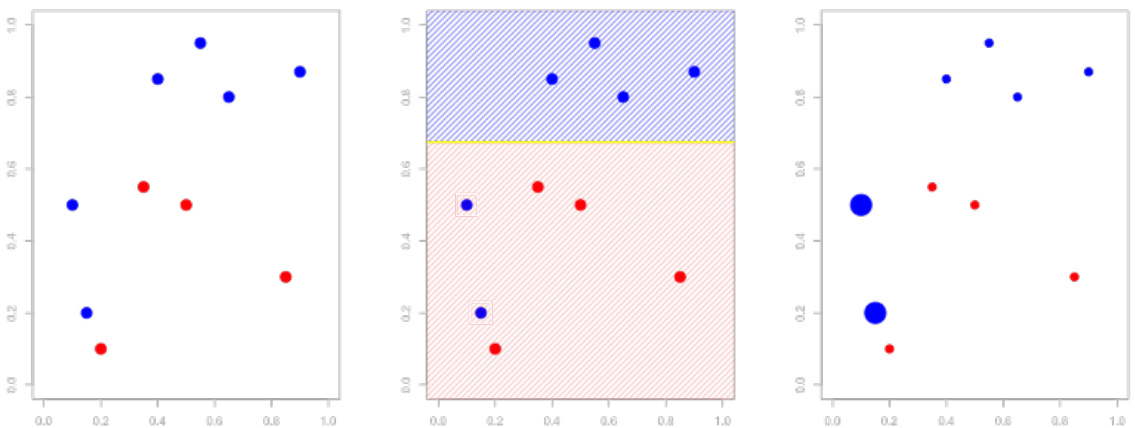
$$m^\star(\cdot) = \frac{1}{\kappa} \sum_{k=1}^{\kappa} \widehat{m}_k(\cdot)$$



Statistics & ML Toolbox : boosting

Bosting & Sequential Learning

$$m_k(\cdot) = m_{k-1}(\cdot) + \underset{h \in \mathcal{H}}{\text{argmin}} \left\{ \sum_{i=1}^{n} \ell(\underbrace{y_i - m_{k-1}(\boldsymbol{x}_i)}_{\varepsilon_i}, h(\boldsymbol{x}_i)) \right\}$$



Statistics & ML : Variable Importance

Parallel boostrap + trees = random forests (bagging)
Sequential learning + stumps = boosted trees

see Elith *et al.* (2000, A working guide to boosted regression trees)

Variable Importance

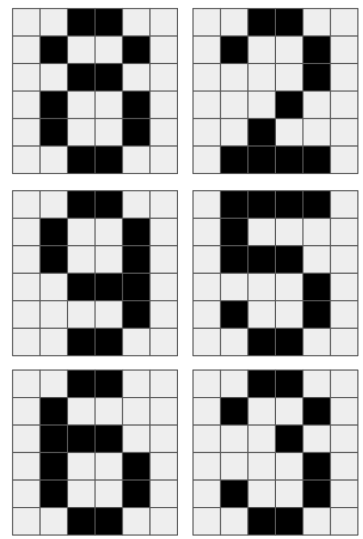
Importance of feature k calculates each feature importance as the sum over the number of splits (across all tress) that include the feature k, proportionally to the number of samples it splits.

Statistics & ML Toolbox : Neural Nets

Very popular for pictures...
Picture $\boldsymbol{x}_i$ is

- a $n \times n$ matrix in $\{0,1\}^{n^2}$ for black & white
- a $n \times n$ matrix in $[0,1]^{n^2}$ for grey-scale
- a $3 \times n \times n$ array in $([0,1]^3)^{n^2}$ for color
- a $T \times 3 \times n \times n$ tensor in $(([0,1]^3)^T)^{n^2}$ for video

y here is the label ("8", "9", "6", etc)
Suppose we want to recognize a "6" on a picture



$$m(\boldsymbol{x}) = \begin{cases} +1 \text{ if } \boldsymbol{x} \text{ is a "6"} \\ -1 \text{ otherwise} \end{cases}$$

Statistics & ML Toolbox : Neural Nets

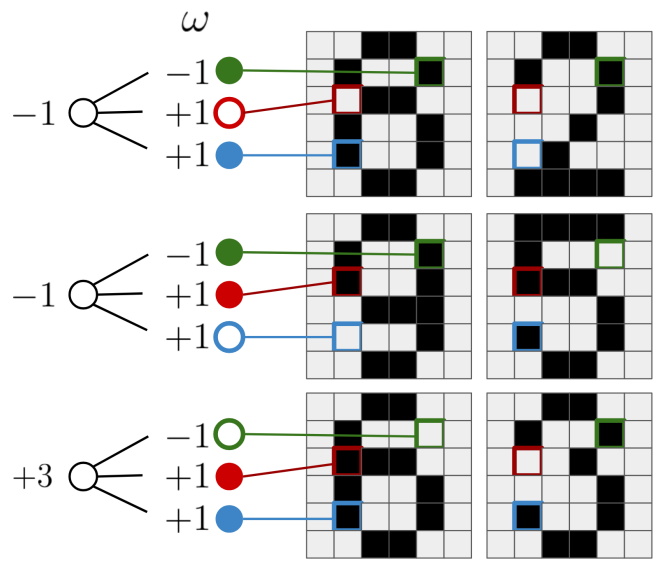
Consider some specifics pixels, and associate weights ω such that

$$\widehat{m}(\boldsymbol{x}) = \text{sign}\left(\sum_{i,j} \omega_{i,j} x_{i,j}\right)$$

where

$$x_{i,j} = \begin{cases} +1 \text{ if pixel } x_{i,j} \text{ is black} & \blacksquare \\ -1 \text{ if pixel } x_{i,j} \text{ is white} & \square \end{cases}$$

for some weights $\omega_{i,j}$ (that can be negative...)



Statistics & ML Toolbox : Neural Nets

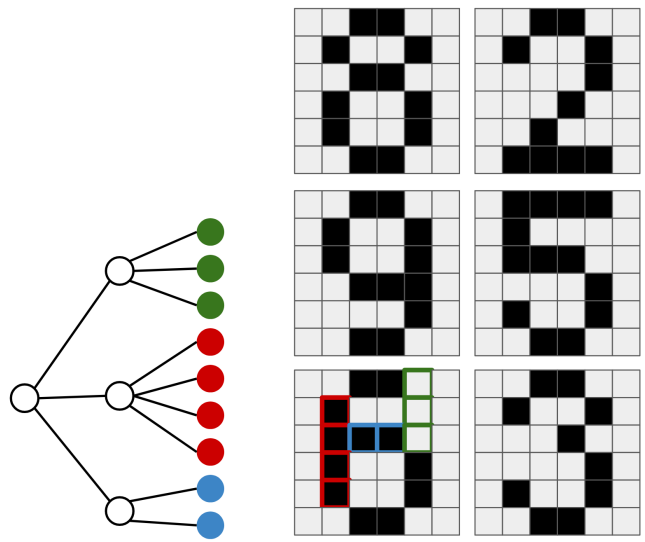
A deep network is a network with a lot of layers

$$\widehat{m}(\boldsymbol{x}) = \text{sign}\left(\sum_i \omega_i \widehat{m}_i(\boldsymbol{x})\right)$$

where $\widehat{m}_i$'s are outputs of previous neural nets

Those layers can capture shapes in some areas

nonlinearities, cross-dependence, etc



Binary Threshold Neuron & Perceptron

If $\boldsymbol{x} \in \{0,1\}^p$, McCulloch & Pitts (1943, A logical calculus of the ideas immanent in nervous activity) suggested a simple model, with threshold b

$$y_i = f\left(\sum_{j=1}^{p} x_{j,i}\right) \text{ where } f(x) = \mathbf{1}(x \geq b)$$

where $\mathbf{1}(x \geq b) = +1$ if $x \geq b$ and 0 otherwise, or (equivalently)

$$y_i = f\left(\omega + \sum_{j=1}^{p} x_{j,i}\right) \text{ where } f(x) = \mathbf{1}(x \geq 0)$$

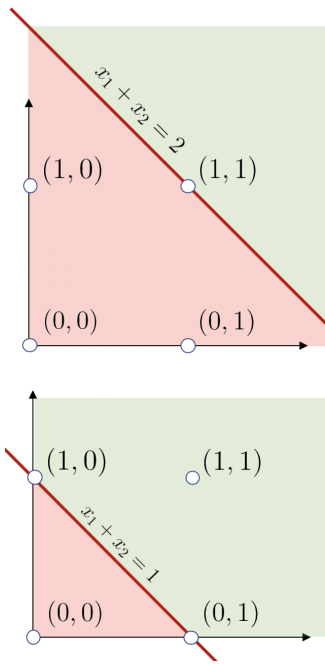
with weight $\omega = -b$. The trick of adding 1 as an input was very important !
$\omega = -1$ is the or logical operator : $y_i = 1$ if $\exists j$ such that $x_{j,i} = 1$
$\omega = -p$, is the and logical operator : $y_i = 1$ if $\forall j$, $x_{j,i} = 1$

Binary Threshold Neuron & Perceptron

but not possible for the xor logical operator : $y_i = 1$ if $x_{1,i} \neq x_{2,i}$

Rosenblatt (1961, Principles of Neurodynamics: Perceptron & Theory of Brain Mechanism) considered the extension where x's are real-valued, with weight $\boldsymbol{\omega} \in \mathbb{R}^p$

$$y_i = f\left(\sum_{j=1}^{p} \omega_j x_{j,i}\right) \text{ where } f(x) = \mathbf{1}(x \geq b)$$



where $\mathbf{1}(x \geq b) = +1$ if $x \geq b$ and 0 otherwise, or (equivalently)

$$y_i = f\left(\omega_0 + \sum_{j=1}^{p} \omega_j x_{j,i}\right) \text{ where } f(x) = \mathbf{1}(x \geq 0)$$

with weights $\boldsymbol{\omega} \in \mathbb{R}^{p+1}$

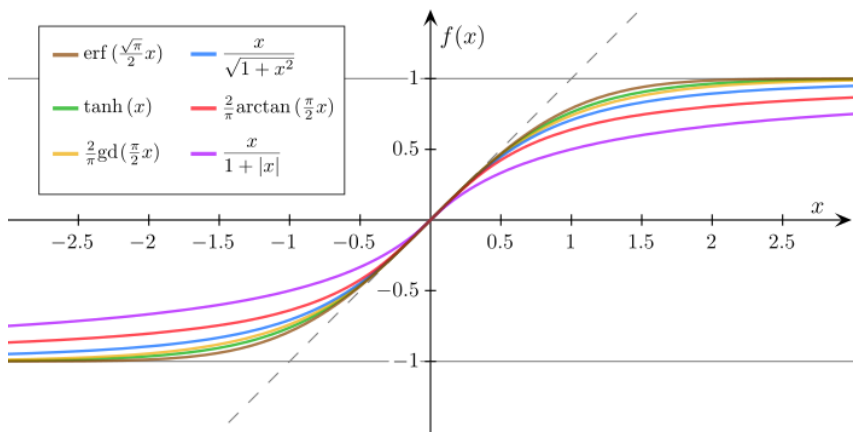
Binary Threshold Neuron & Perceptron

Minsky & Papert (1969, Perceptrons: an Introduction to Computational Geometry) proved that perceptron were a linear separator, not very powerful

Define the sigmoid function $f(x) = \dfrac{1}{1+e^{-x}} = \dfrac{e^x}{e^x+1}$ (= logistic function).

This function f is called the activation function.

If $y \in \{-1, +1\}$, one can consider the hyperbolic tangent $f(x) = \dfrac{(e^x - e^{-x})}{(e^x + e^{-x})}$ or the inverse tangent function (see wikipedia).



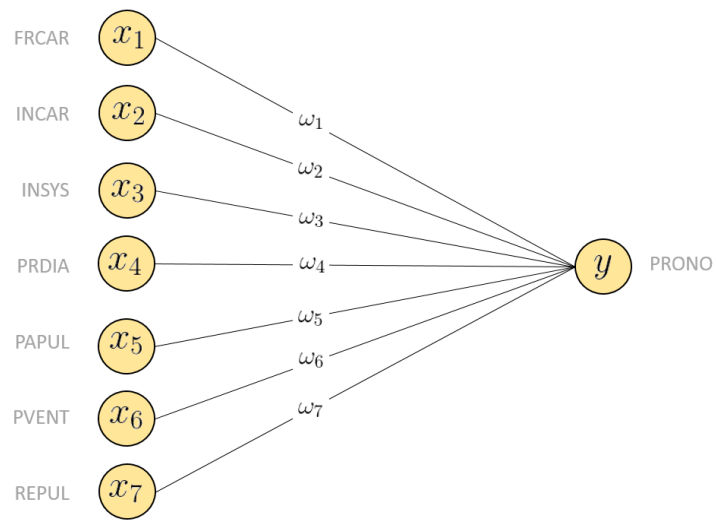
Statistics & ML Toolbox : Neural Nets

So here, for a classication problem,

$$y_i = f\left(\omega_0 + \sum_{j=1}^{p} \omega_j x_{j,i}\right) = m(\boldsymbol{x}_i)$$

The error of that model can be computed used the quadratic loss function,

$$\sum_{i=1}^{n} (y_i - m(\boldsymbol{x}_i))^2$$



or cross-entropy

$$\sum_{i=1}^{n} (y_i \log m(\boldsymbol{x}_i) + [1 - y_i] \log[1 - m(\boldsymbol{x}_i)])$$

Statistics & ML Toolbox : Neural Nets

Consider a single hidden layer, with 3 different neurons, so that

$$m(\boldsymbol{x}) = f\left(\omega_0 + \sum_{h=1}^{3} \omega_h f_h\left(\omega_{h,0} + \sum_{j=1}^{p} \omega_{h,j} x_j\right)\right)$$

or

$$m(\boldsymbol{x}) = f\left(\omega_0 + \sum_{h=1}^{3} \omega_h f_h\left(\omega_{h,0} + \boldsymbol{x}^\top \boldsymbol{\omega}_h\right)\right)$$