Machine Learning

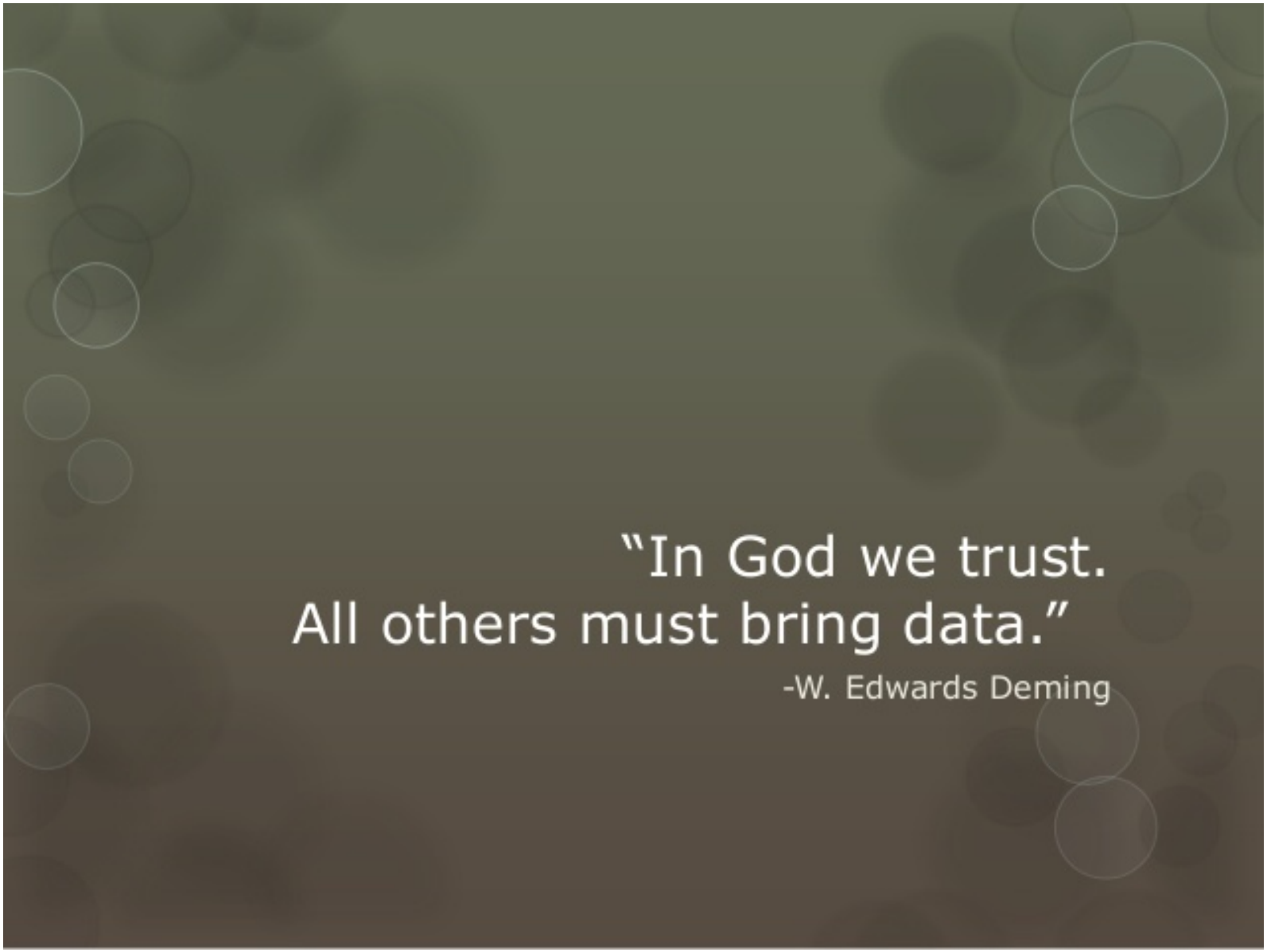
Statistical learning theory and algorithms

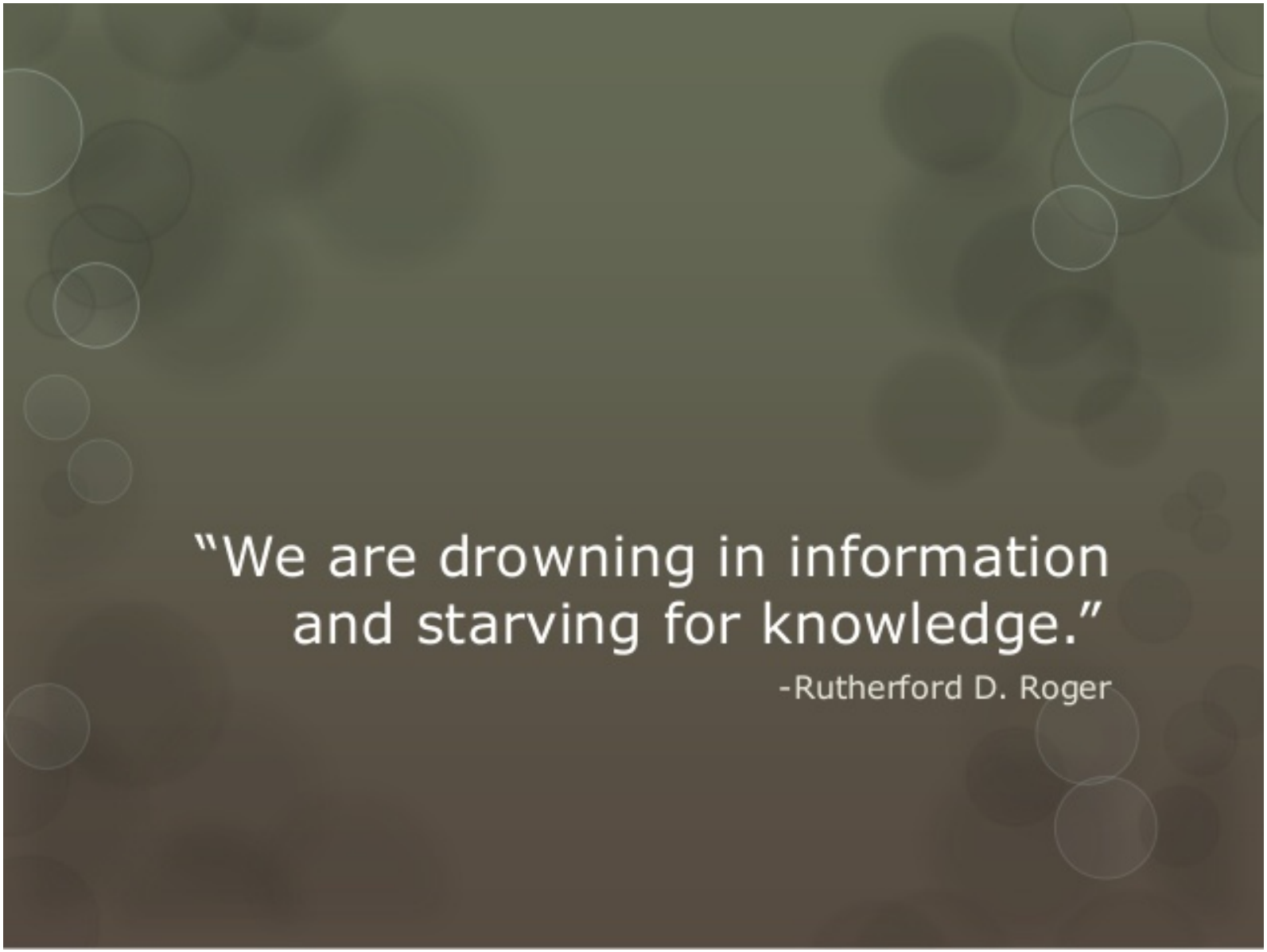
Benjamin Guedj, Ph.D.

https://bguedj.github.io
Inria Lille - Nord Europe

The rising of Artificial intelligence (AI)

Introduction

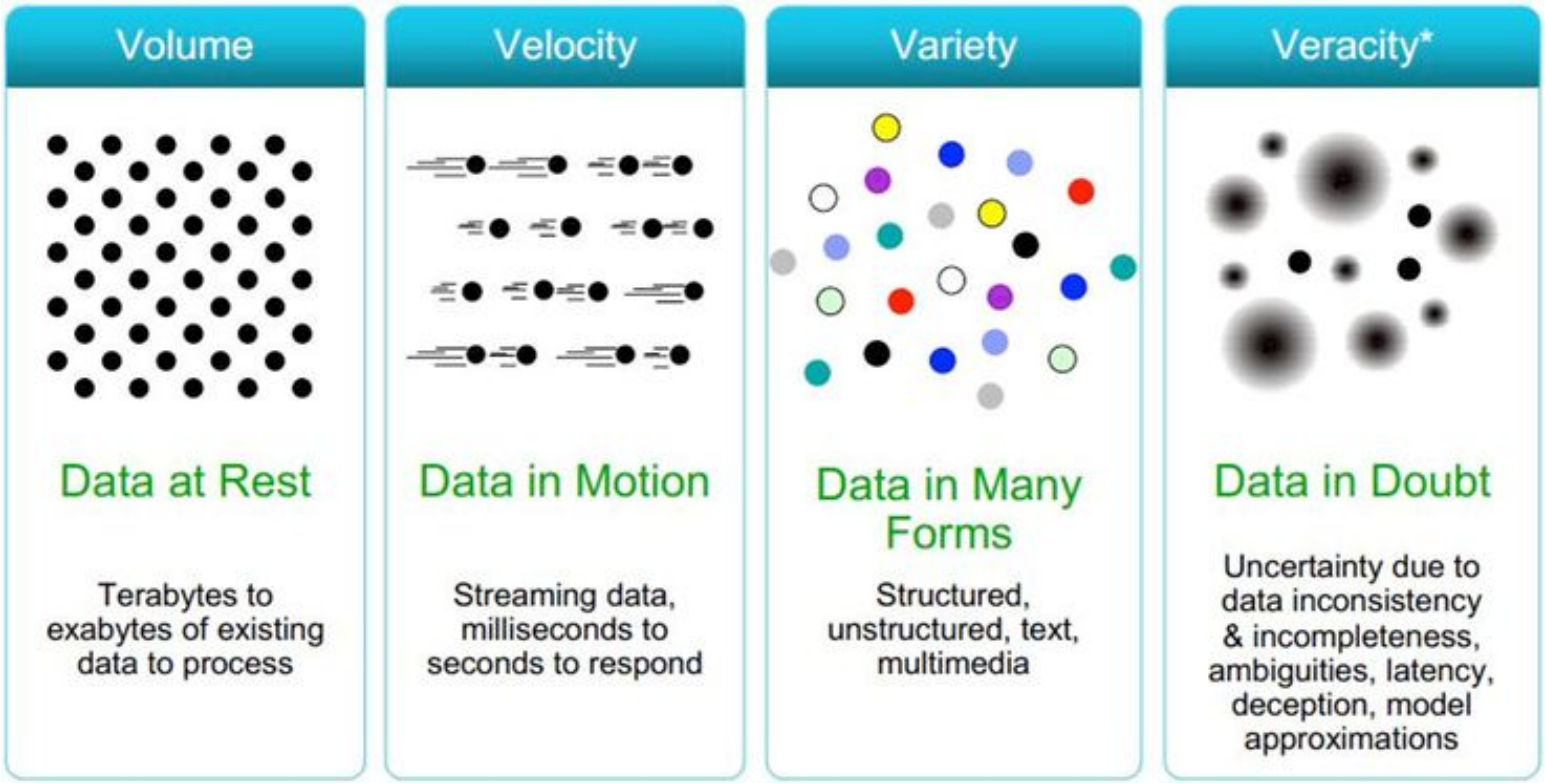

"In God we trust.
All others must bring data."
-W. Edwards Deming


"We are drowning in information
and starving for knowledge."
-Rutherford D. Roger

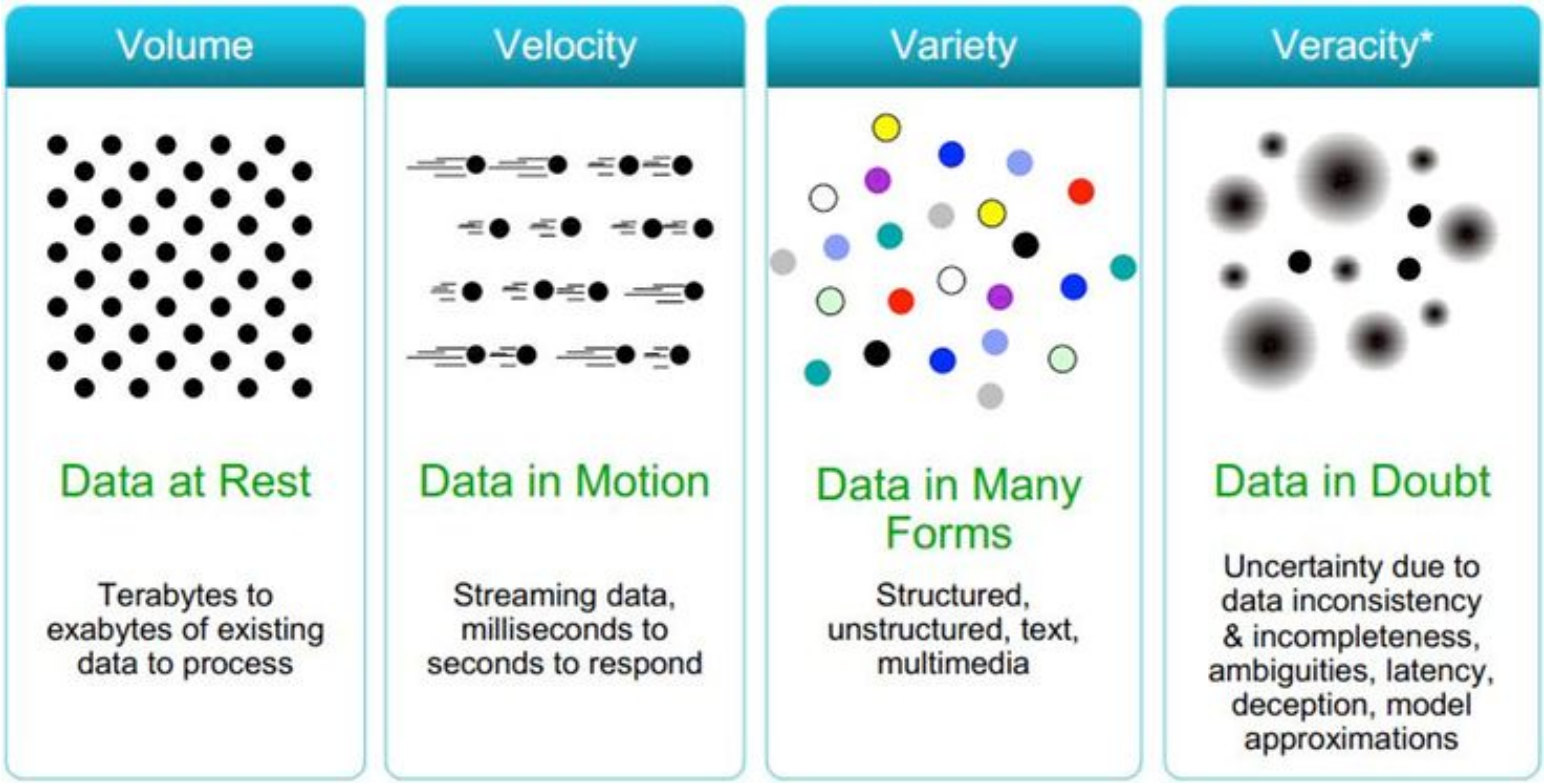
It is vital to remember that information - in the sense of raw data - is not knowledge, that knowledge is not wisdom, and that wisdom is not foresight. But information is the first essential step to all of these.

Arthur C Clarke

Big Data 4 V's


Volume
Data at Rest
Terabytes to exabytes of existing data to process
Velocity
Data in Motion
Streaming data, milliseconds to seconds to respond
Variety
Data in Many Forms
Structured, unstructured, text, multimedia
Veracity*
Data in Doubt
Uncertainty due to data inconsistency & incompleteness, ambiguities, latency, deception, model approximations

Big Data 4 V's



$\longrightarrow$ Value ($)

Data Scientists: 100,000 jobs by 2020. Demand is expected to exceed supply by 50 to 60% (McKinsey, 2015)

Volume / Velocity

i	3	6	9	12	15	18	21	24	
10^i	kilo	mega	giga	tera	peta	exa	zeta	yotta	bytes

- Google: 24 petabytes/day
- Facebook: 10 terabytes/day
- Twitter: 7 terabytes/day
- Large Hadron Collider: 40 terabytes/second
- Google Street View: 20 petabytes
- AT&T network: 30 petabytes/day
- Human brain may store about 2.5 petabytes of binary data
- ...

Variety / Veracity

categorical
Marital status
maried

integer
Children
3

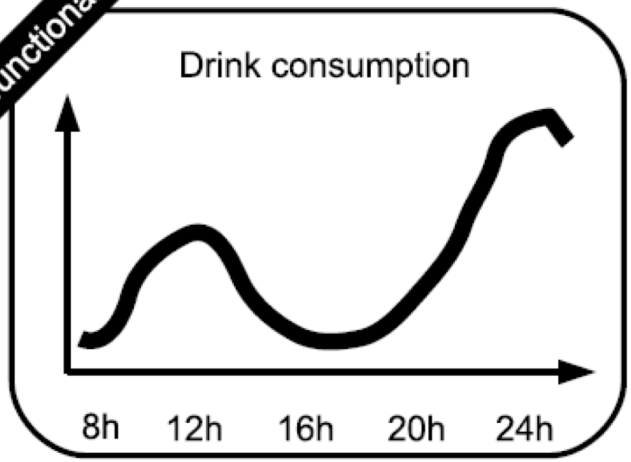
missing
Size (m)
?

rank
Drink preference
beer > soda > water

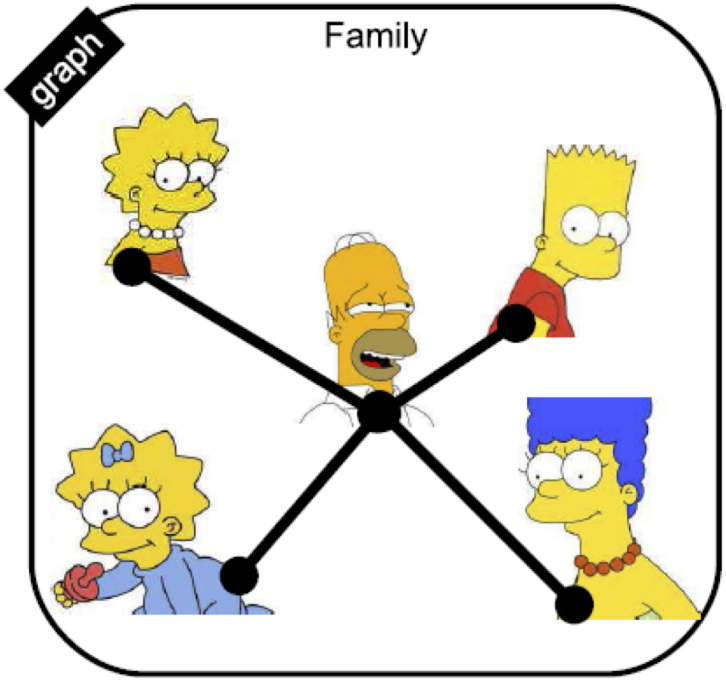
ordinal
Intelligence
low

continuous
Weight (kg)
119.5

functional

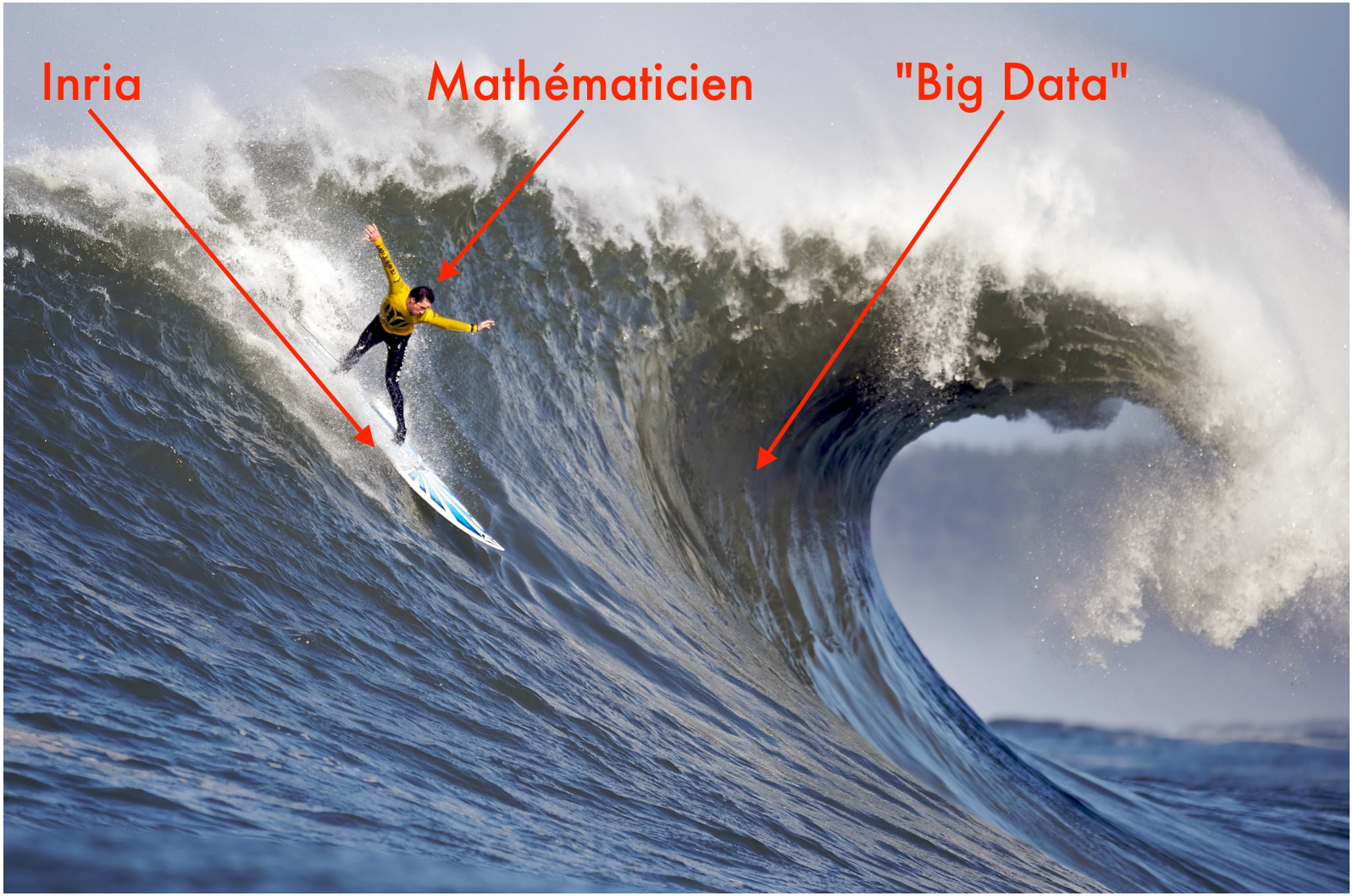


graph



And so on...

My job (allegory)



Data sources

https://www.kaggle.com
https://archive.ics.uci.edu/ml/datasets.html
https://www.data.gov
https://data.gov.uk
https://www.data.gouv.fr/fr/
http://www.census.gov/data.html
http://data.europa.eu/euodp/en/data/
http://www.healthdata.gov
https://aws.amazon.com/fr/datasets/
https://www.gapminder.org/data/
https://www.google.com/trends/explore
https://www.google.com/finance
...

Boston Housing

Attribute Information:

1. CRIM: per capita crime rate by town
2. ZN: proportion of residential land zoned for lots over 25,000 sq.ft.
3. INDUS: proportion of non-retail business acres per town
4. CHAS: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
5. NOX: nitric oxides concentration (parts per 10 million)
6. RM: average number of rooms per dwelling
7. AGE: proportion of owner-occupied units built prior to 1940
8. DIS: weighted distances to five Boston employment centres
9. RAD: index of accessibility to radial highways
10. TAX: full-value property-tax rate per $10,000
11. PTRATIO: pupil-teacher ratio by town
12. B: 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
13. LSTAT: % lower status of the population
14. MEDV: Median value of owner-occupied homes in $1000's

Wine Quality

Attribute Information:

For more information, read [Cortez et al., 2009].
Input variables (based on physicochemical tests):
1 - fixed acidity
2 - volatile acidity
3 - citric acid
4 - residual sugar
5 - chlorides
6 - free sulfur dioxide
7 - total sulfur dioxide
8 - density
9 - pH
10 - sulphates
11 - alcohol
Output variable (based on sensory data):
12 - quality (score between 0 and 10)

Email Spam Detection

Used Cars' Prices

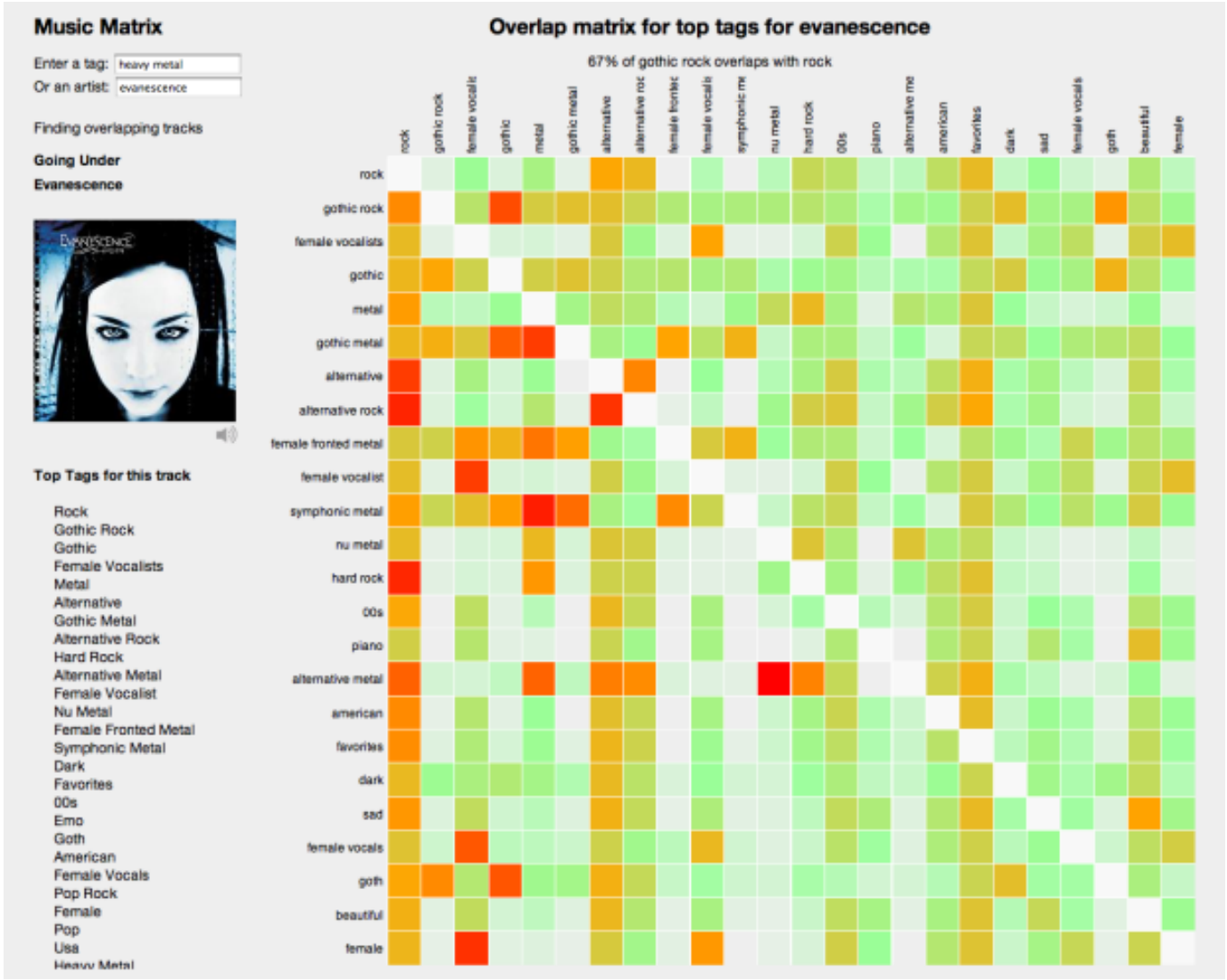
Attribute Information:

Attribute: Attribute Range

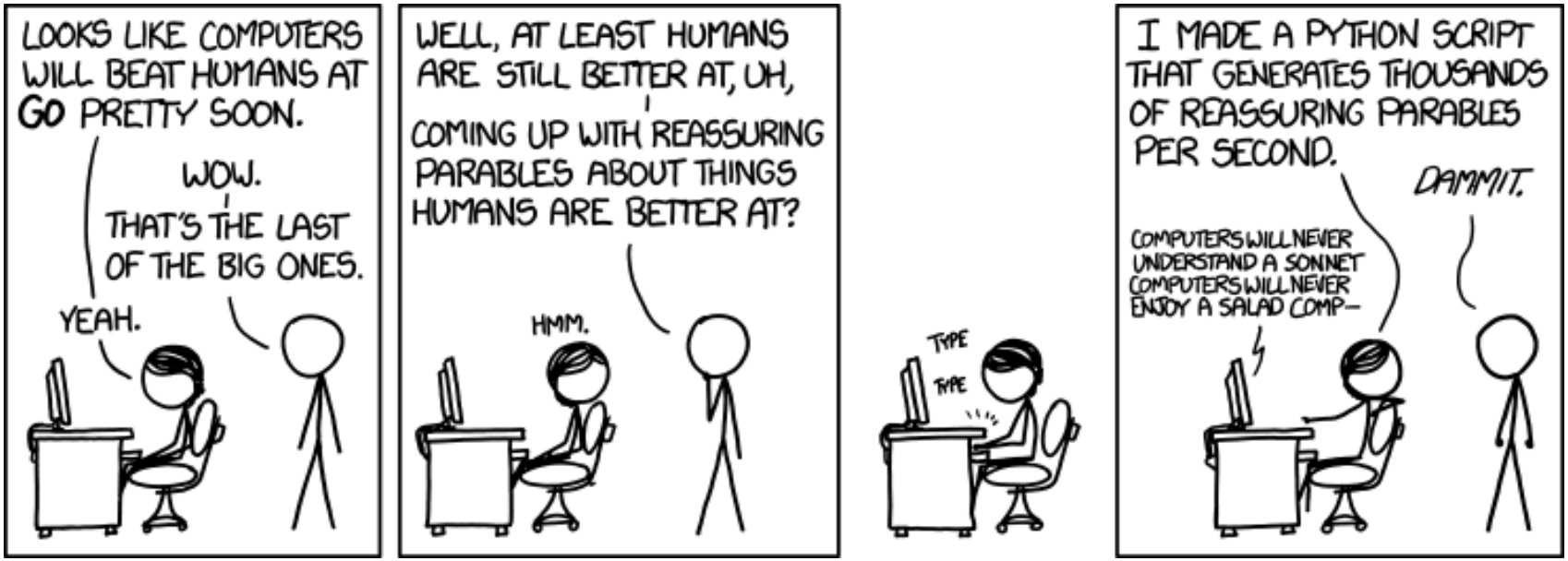
1. symboling: -3, -2, -1, 0, 1, 2, 3.
2. normalized-losses: continuous from 65 to 256.
3. make:
alfa-romero, audi, bmw, chevrolet, dodge, honda,
isuzu, jaguar, mazda, mercedes-benz, mercury,
mitsubishi, nissan, peugot, plymouth, porsche,
renault, saab, subaru, toyota, volkswagen, volvo

4. fuel-type: diesel, gas.
5. aspiration: std, turbo.
6. num-of-doors: four, two.
7. body-style: hardtop, wagon, sedan, hatchback, convertible.
8. drive-wheels: 4wd, fwd, rwd.
9. engine-location: front, rear.
10. wheel-base: continuous from 86.6 120.9.
11. length: continuous from 141.1 to 208.1.
12. width: continuous from 60.3 to 72.3.
13. height: continuous from 47.8 to 59.8.
14. curb-weight: continuous from 1488 to 4066.
15. engine-type: dohc, dohcv, l, ohc, ohcf, ohcv, rotor.
16. num-of-cylinders: eight, five, four, six, three, twelve, two.
17. engine-size: continuous from 61 to 326.
18. fuel-system: 1bbl, 2bbl, 4bbl, idi, mfi, mpfi, spdi, spfi.
19. bore: continuous from 2.54 to 3.94.
20. stroke: continuous from 2.07 to 4.17.
21. compression-ratio: continuous from 7 to 23.
22. horsepower: continuous from 48 to 288.
23. peak-rpm: continuous from 4150 to 6600.
24. city-mpg: continuous from 13 to 49.
25. highway-mpg: continuous from 16 to 54.
26. price: continuous from 5118 to 45400.

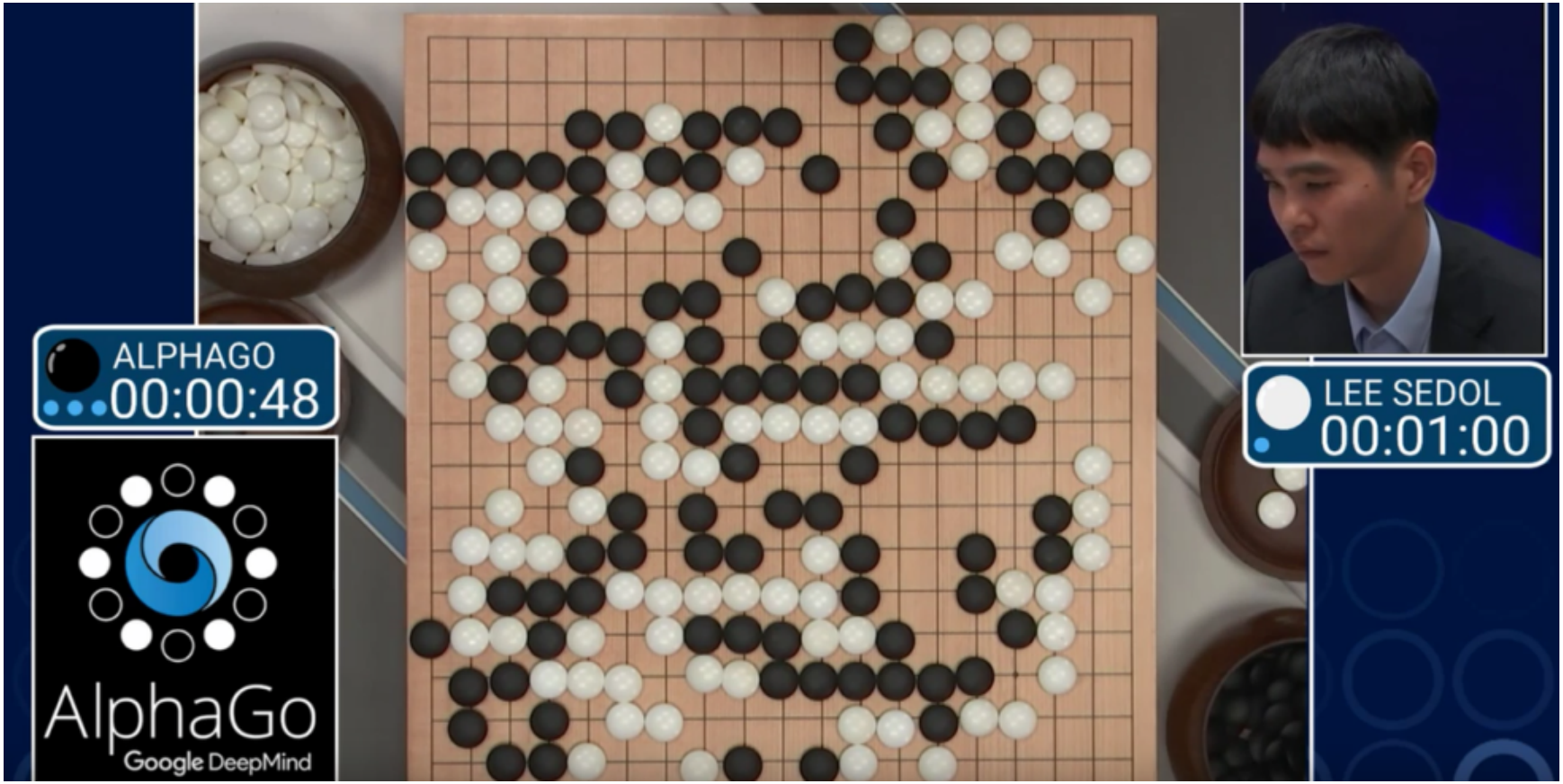
Million Song Dataset



Chess (1997)

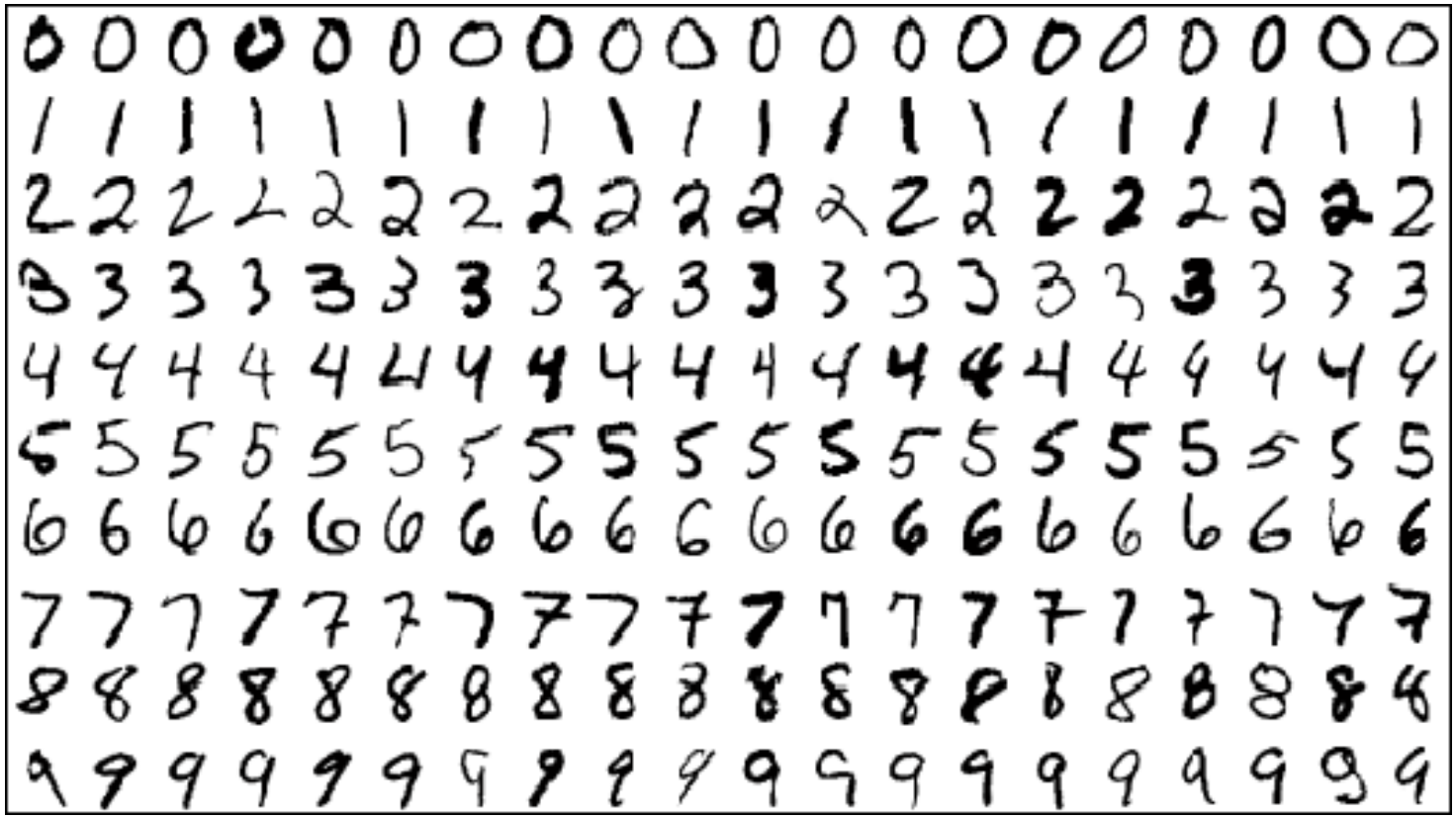

LOOKS LIKE COMPUTERS WILL BEAT HUMANS AT GO PRETTY SOON.
YEAH.
WOW. THAT'S THE LAST OF THE BIG ONES.
WELL, AT LEAST HUMANS ARE STILL BETTER AT, UH, COMING UP WITH REASSURING PARABLES ABOUT THINGS HUMANS ARE BETTER AT?
HMM.
TYPE
TYPE
I MADE A PYTHON SCRIPT THAT GENERATES THOUSANDS OF REASSURING PARABLES PER SECOND.
COMPUTERS WILL NEVER UNDERSTAND A SONNET COMPUTERS WILL NEVER ENJOY A SALAD COMP—
DAMMIT.

Go (2016)



Digits recognition

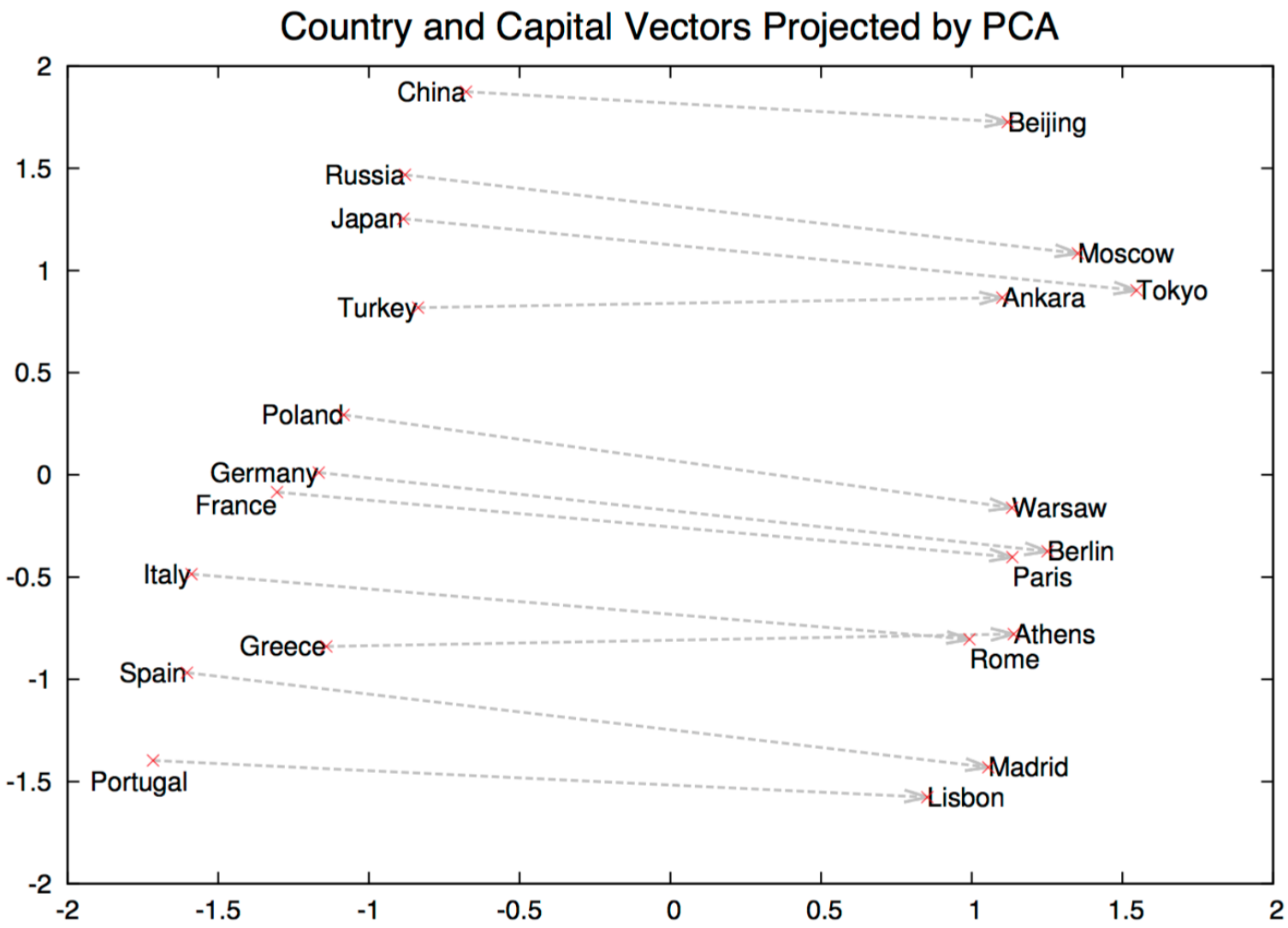
[Demo]



Captcha

Characters under typical distortions	Recognition rate
	~100%
	96+%
	100%
	98%
	~100%
	95+%

Learning words representations



Learning words representations

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

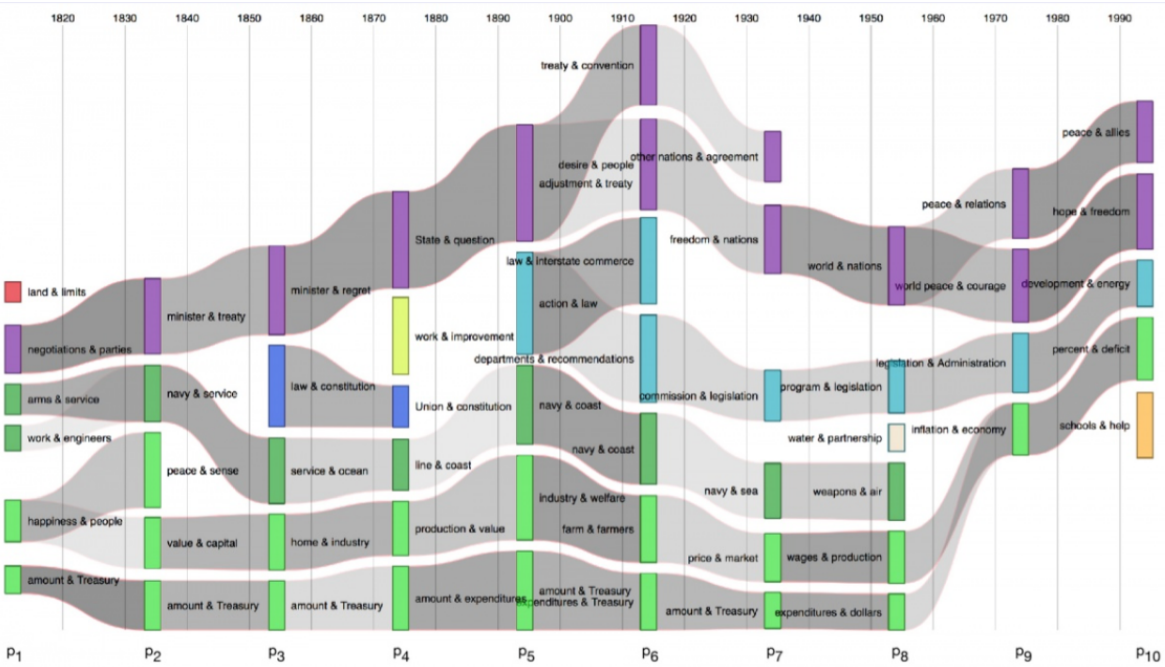
Paris - France + Italy = Rome

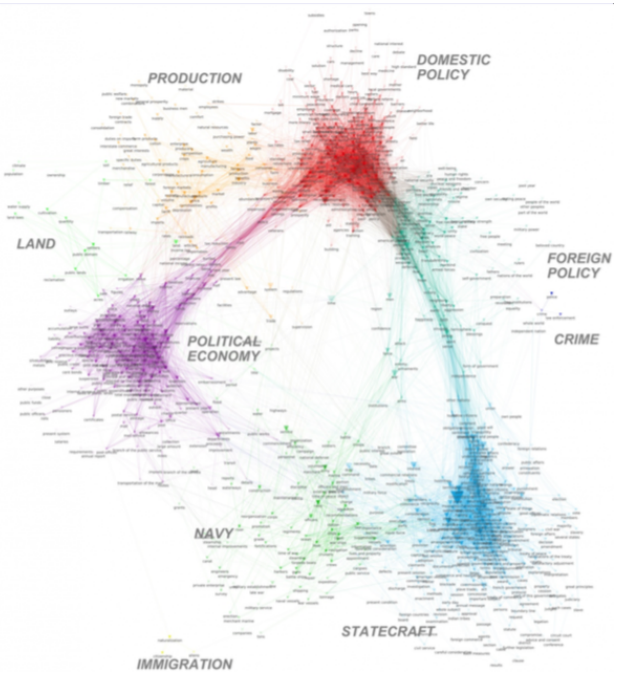
[Demo]
[Demo 2]


A person riding a motorcycle on a dirt road.
Two dogs play in the grass.
A skateboarder does a trick on a ramp.
A dog is jumping to catch a frisbee.
A group of young people playing a game of frisbee.
Two hockey players are fighting over the puck.
A little girl in a pink hat is blowing bubbles.
A refrigerator filled with lots of food and drinks.
A herd of elephants walking across a dry grass field.
A close up of a cat laying on a couch.
A red motorcycle parked on the side of the road.
A yellow school bus parked in a parking lot.
Describes without errors
Describes with minor errors
Somewhat related to the image
Unrelated to the image

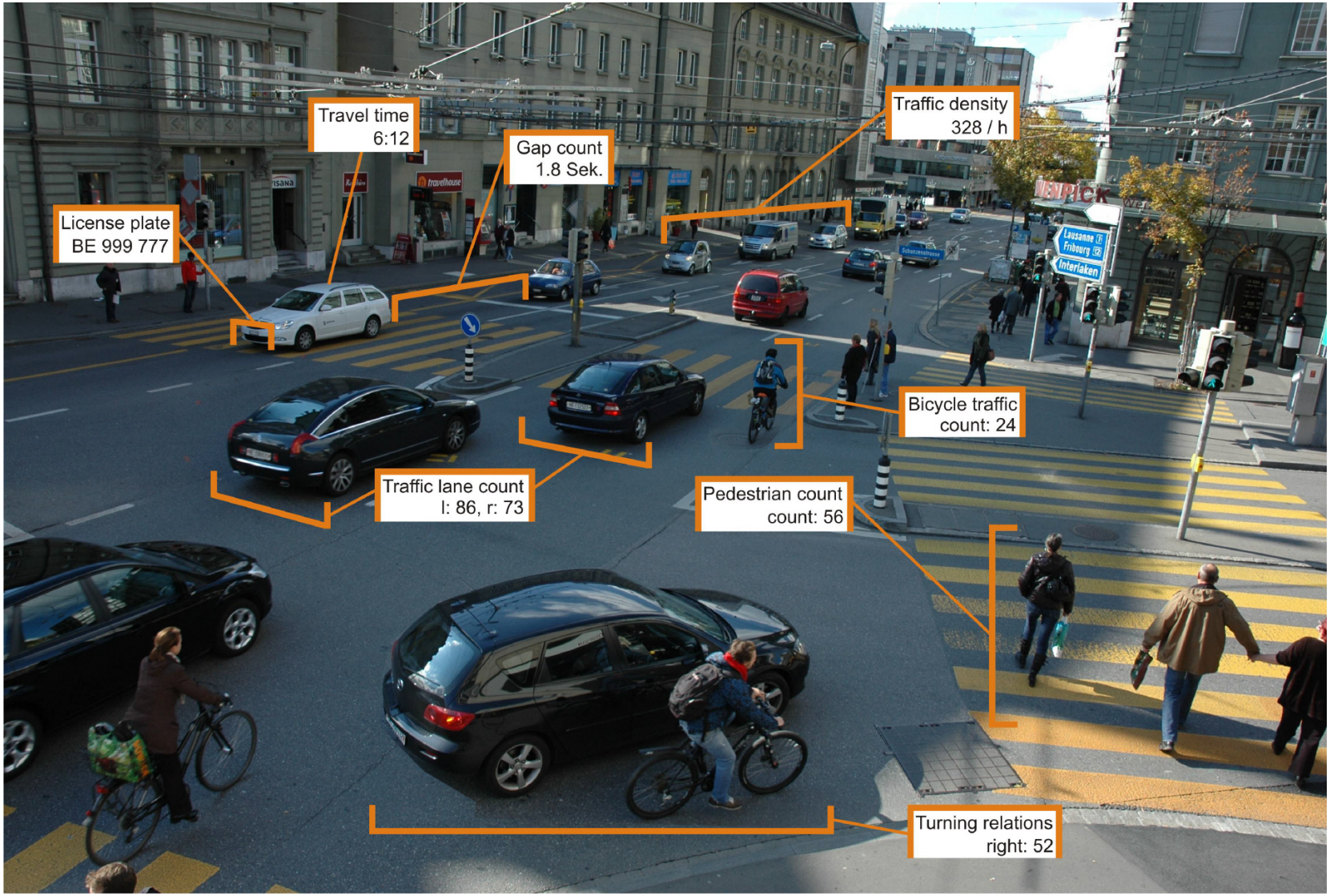
Text

- Translation $\longrightarrow$ [Video]
- Plagiarism detection, automatic summary
- Topics detection
- Sentiment analysis $\longrightarrow$ [Demo]

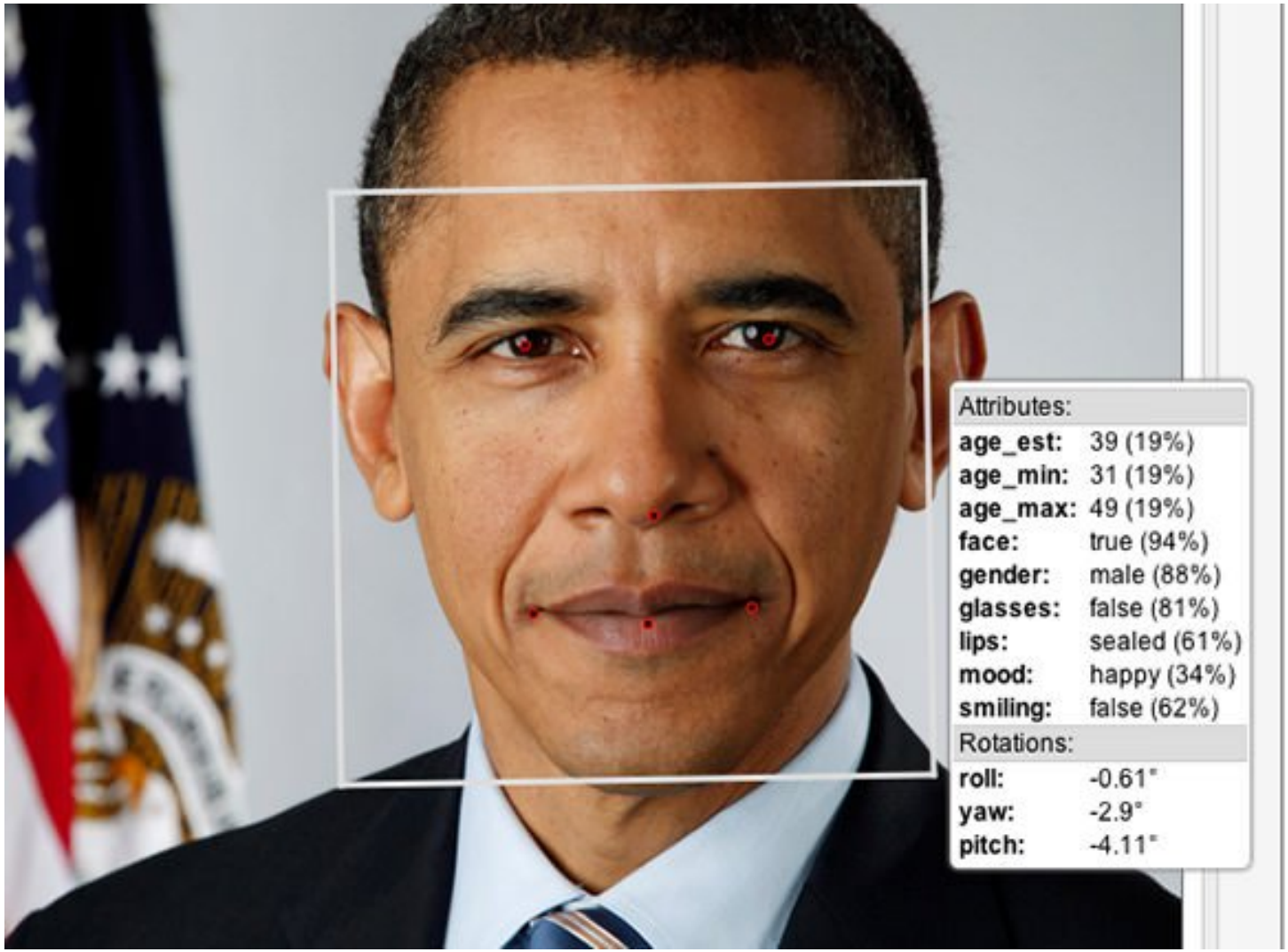




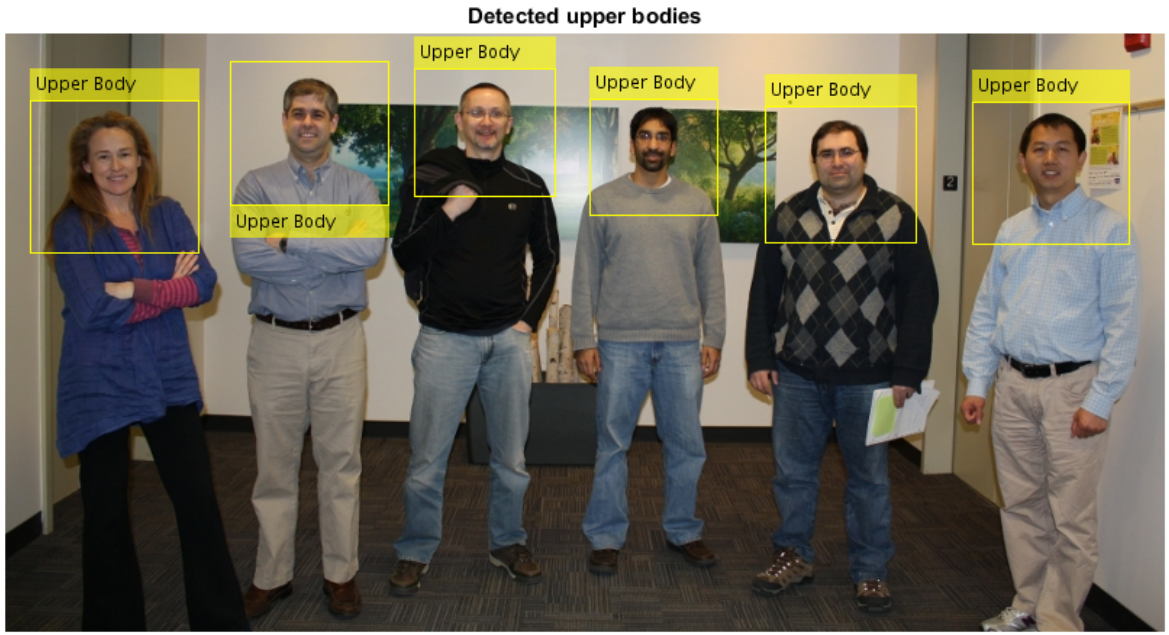
[Video]


Travel time
6:12
Gap count
1.8 Sek.
Traffic density
328 / h
License plate
BE 999 777
Bicycle traffic
count: 24
Traffic lane count
l: 86, r: 73
Pedestrian count
count: 56
Turning relations
right: 52

[Demo]
[Demo 2]

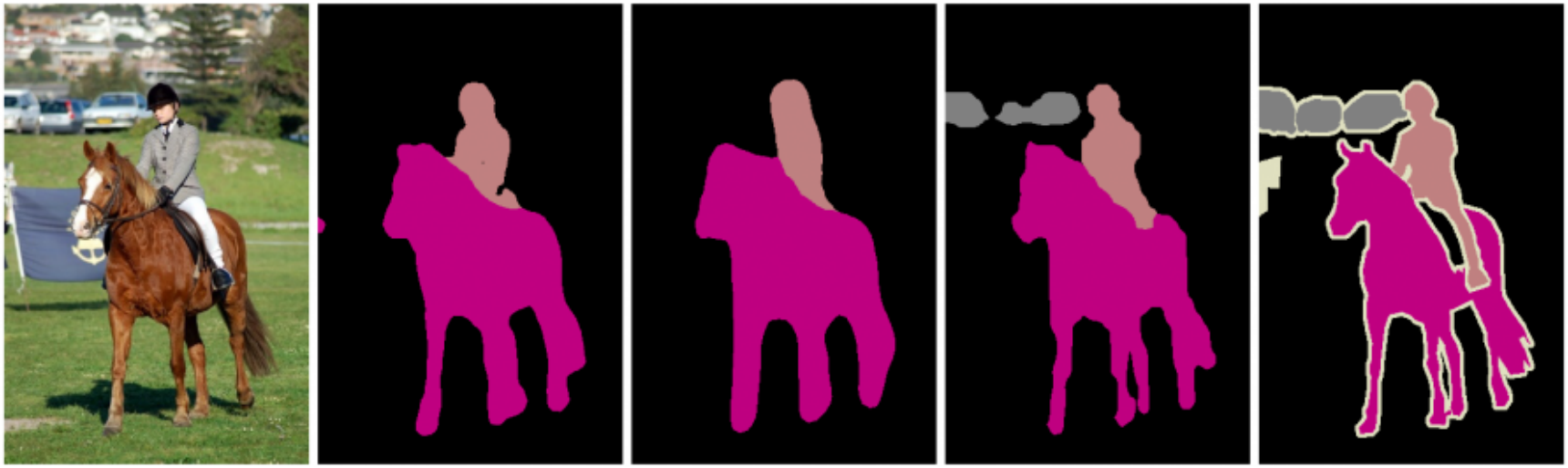

Attributes:
age_est: 39 (19%)
age_min: 31 (19%)
age_max: 49 (19%)
face: true (94%)
gender: male (88%)
glasses: false (81%)
lips: sealed (61%)
mood: happy (34%)
smiling: false (62%)
Rotations:
roll: -0.61°
yaw: -2.9°
pitch: -4.11°

Detection and tracking



[Video]
[Video 2]
[Video 3]

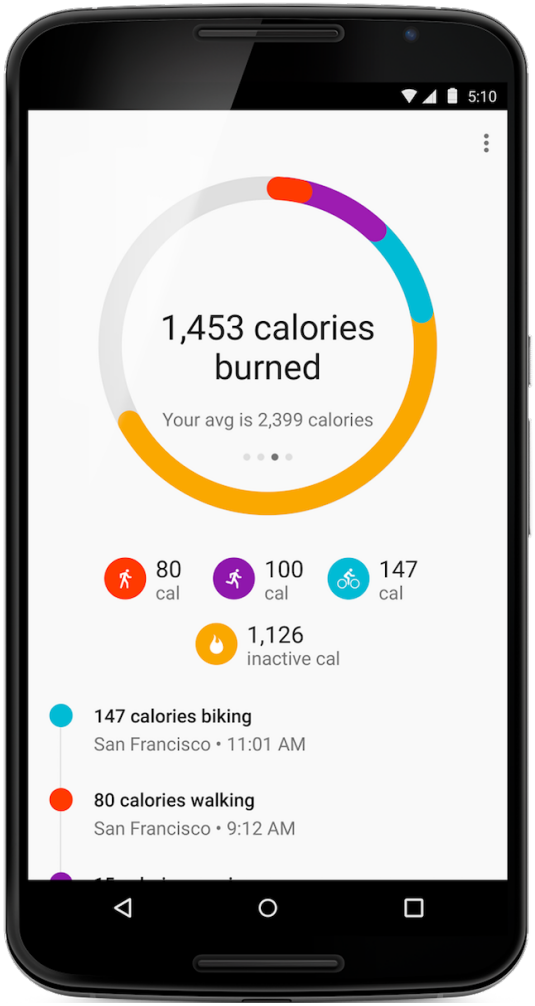
Image segmentation



[Video]
[Video 2]

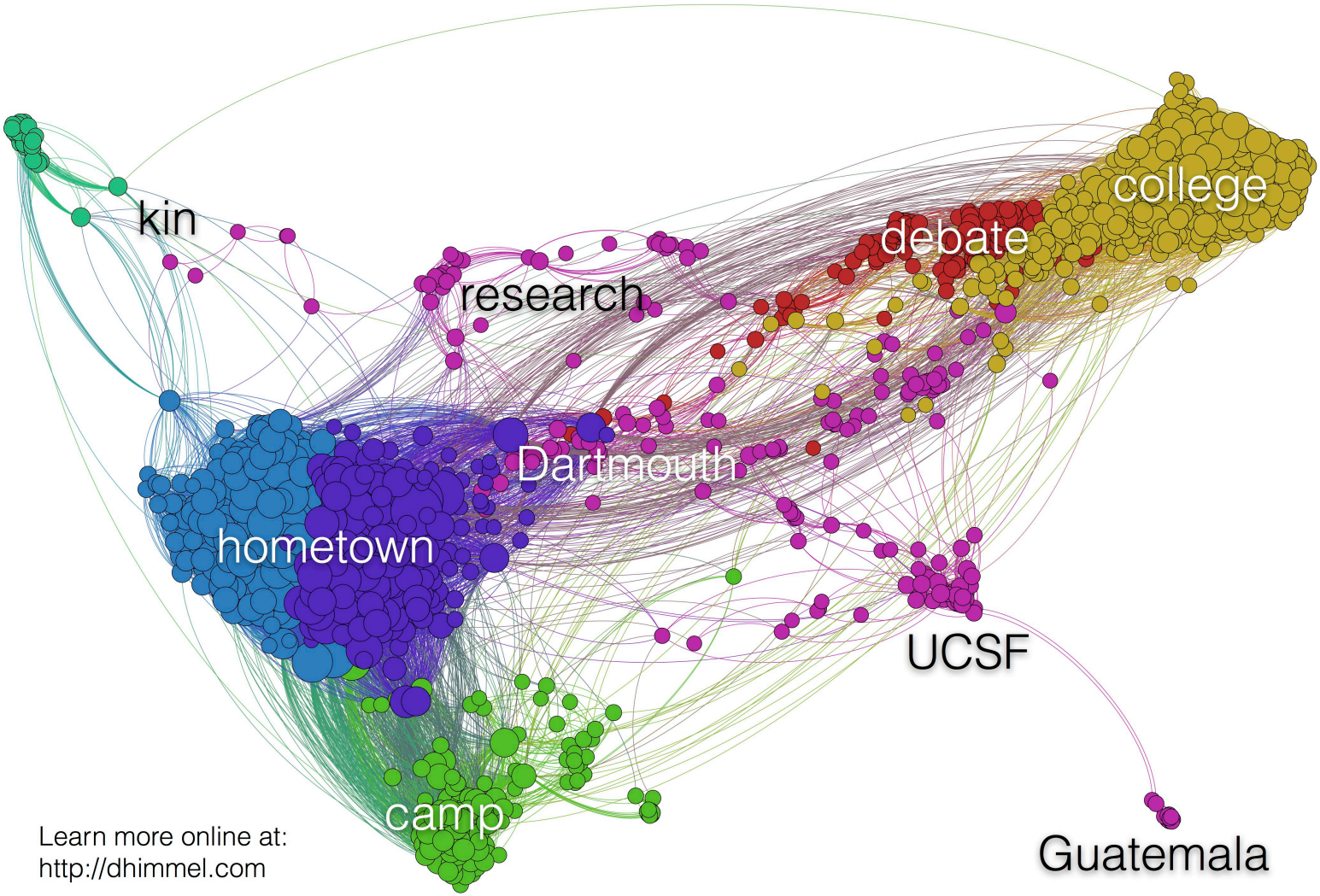
Sound

- Source separation $\longrightarrow$ [Video] [Video 2]
- Denoising/restoration
- Speaker recognition
- Music classification
- ...

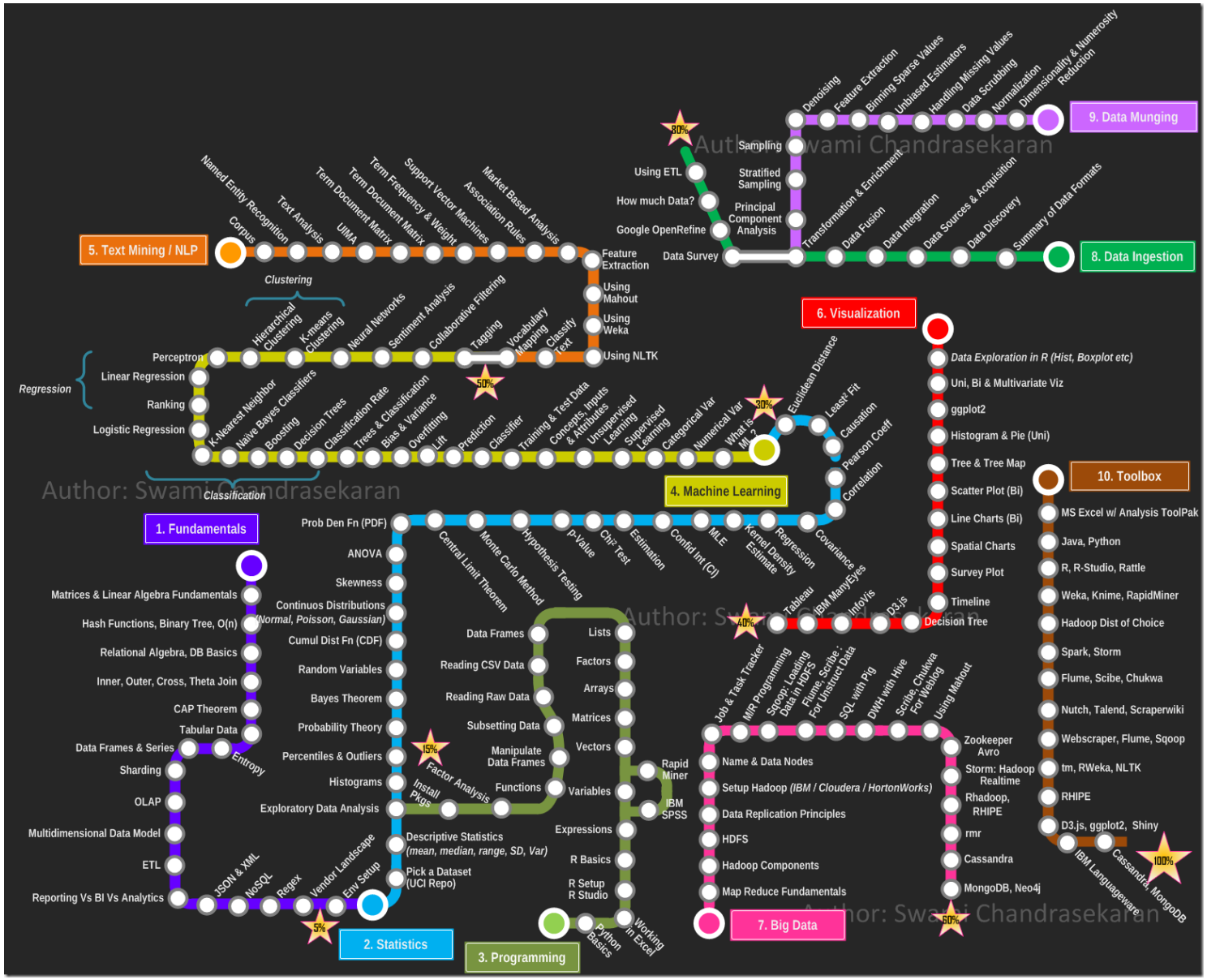

5:10
1,453 calories burned
Your avg is 2,399 calories
80 cal
100 cal
147 cal
1,126 inactive cal
147 calories biking
San Francisco • 11:01 AM
80 calories walking
San Francisco • 9:12 AM

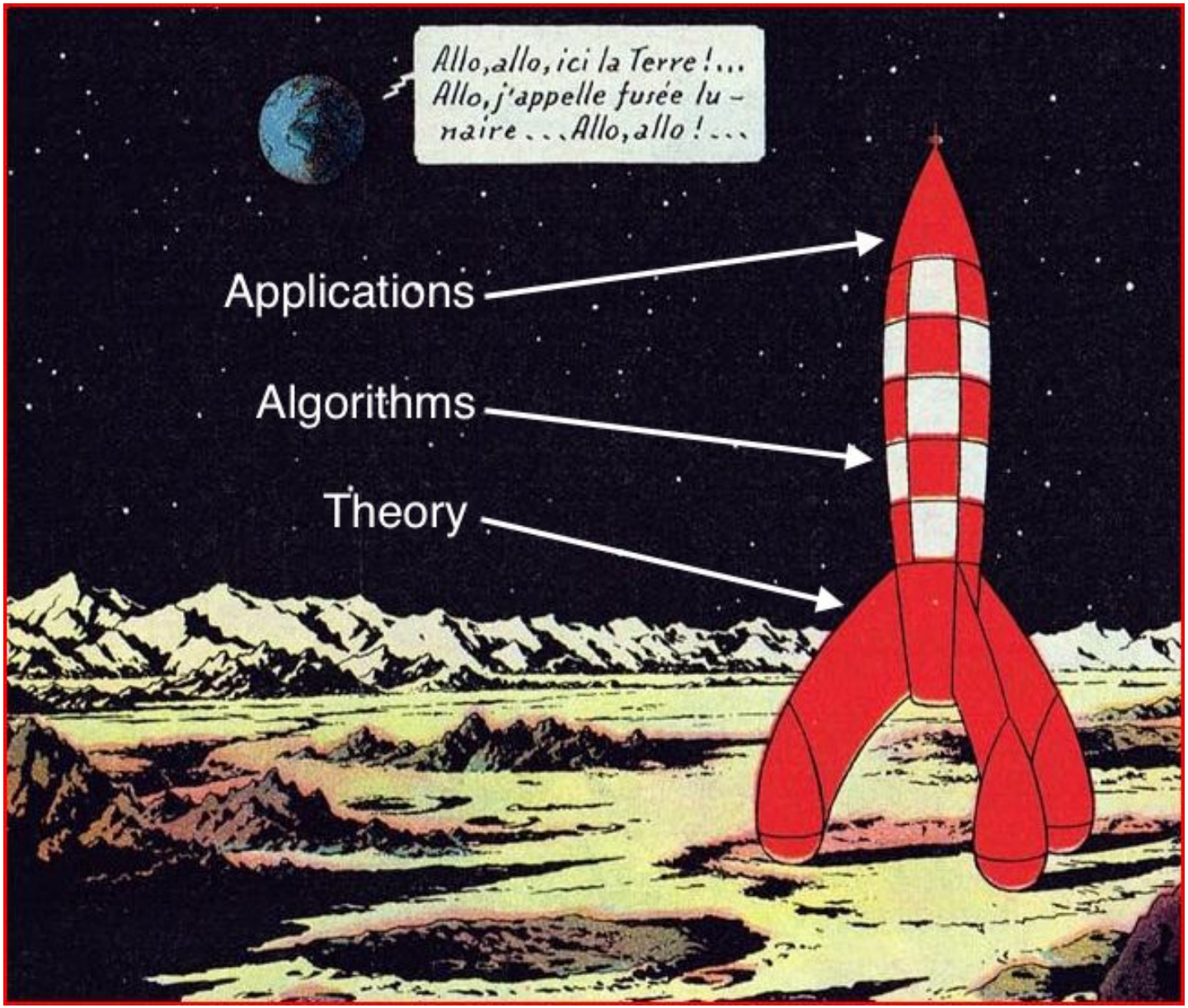

9:41 AM
100%
Dashboard
Day
Week
Month
Year
Steps
Daily Average: 8,757
4,152 steps
Today, 9:41 AM
13,648
4,152
Aug 12
19
26
Sep 2
9
Weight
Daily Average: 165.5
165lbs
Today, 9:41 AM
168
163
Heart Rate
Min: 41 Max: 138
88bpm
Today, 9:41 AM
105h
Dashboard
Health Data
Sources
Medical ID

The Friendship Network of Daniel Himmelstein




tinder
badoo
adopteunmec.com
meetic
NETFLIX
Spotify
DEEZER
YouTube
amazon
Google


1. Fundamentals
Matrices & Linear Algebra Fundamentals
Hash Functions, Binary Tree, O(n)
Relational Algebra, DB Basics
Inner, Outer, Cross, Theta Join
CAP Theorem
Tabular Data
Data Frames & Series
Entropy
Sharding
OLAP
Multidimensional Data Model
ETL
Reporting Vs BI Vs Analytics
JSON & XML
NoSQL
Regex
Vendor Landscape
Env Setup
5%
2. Statistics
Prob Den Fn (PDF)
ANOVA
Skewness
Continuos Distributions (Normal, Poisson, Gaussian)
Cumul Dist Fn (CDF)
Random Variables
Bayes Theorem
Probability Theory
Percentiles & Outliers
Histograms
Exploratory Data Analysis
Descriptive Statistics (mean, median, range, SD, Var)
Pick a Dataset (UCI Repo)
Central Limit Theorem
Monte Carlo Method
Hypothesis Testing
p-Value
Chi² Test
Estimation
Confid Int (CI)
MLE
Kernel Density Estimate
Regression
Covariance
15%
Factor Analysis
Install Pkgs
3. Programming
Data Frames
Reading CSV Data
Reading Raw Data
Subsetting Data
Manipulate Data Frames
Functions
Lists
Factors
Arrays
Matrices
Vectors
Variables
Expressions
R Basics
R Setup
R Studio
Python Basics
Working in Excel
Rapid Miner
IBM SPSS
4. Machine Learning
Regression
Perceptron
Linear Regression
Ranking
Logistic Regression
Classification
K-Nearest Neighbor
Naive Bayes Classifiers
Boosting
Decision Trees
Classification Rate
Trees & Classification
Bias & Variance
Overfitting
Lift
Prediction
Classifier
Training & Test Data
Concepts, Inputs & Attributes
Unsupervised Learning
Supervised Learning
Categorical Var
Numerical Var
50%
Clustering
Hierarchical Clustering
K-means Clustering
Neural Networks
Sentiment Analysis
Collaborative Filtering
Tagging
Vocabulary Mapping
Classify Text
Using NLTK
Using Mahout
Using Weka
Feature Extraction
Named Entity Recognition
Corpus
Text Analysis
UIMA
Term Document Matrix
Term Document Matrix
Term Frequency & Weight
Support Vector Machines
Association Rules
Market Based Analysis
5. Text Mining / NLP
6. Visualization
What is ML?
30%
Euclidean Distance
Least² Fit
Causation
Pearson Coeff
Correlation
Data Exploration in R (Hist, Boxplot etc)
Uni, Bi & Multivariate Viz
ggplot2
Histogram & Pie (Uni)
Tree & Tree Map
Scatter Plot (Bi)
Line Charts (Bi)
Spatial Charts
Survey Plot
Timeline
Decision Tree
40%
Tableau
IBM ManyEyes
InfoVis
D3.js
7. Big Data
Job & Task Tracker
MR Programming
Sqoop: Loading Data in HDFS
Flume, Scribe: For Unstruct Data
SQL with Pig
DWH with Hive
Scribe, Chukwa For Weblog
Using Mahout
Name & Data Nodes
Setup Hadoop (IBM / Cloudera / HortonWorks)
Data Replication Principles
HDFS
Hadoop Components
Map Reduce Fundamentals
Zookeeper Avro
Storm: Hadoop Realtime
Rhadoop, RHIPE
rmr
Cassandra
MongoDB, Neo4j
60%
8. Data Ingestion
Data Survey
Google OpenRefine
How much Data?
Using ETL
80%
Sampling
Stratified Sampling
Principal Component Analysis
Transformation & Enrichment
Data Fusion
Data Integration
Data Sources & Acquisition
Data Discovery
Summary of Data Formats
9. Data Munging
Denoising
Feature Extraction
Binning Sparse Values
Unbiased Estimators
Handling Missing Values
Data Scrubbing
Normalization
Dimensionality & Numerosity Reduction
10. Toolbox
MS Excel w/ Analysis ToolPak
Java, Python
R, R-Studio, Rattle
Weka, Knime, RapidMiner
Hadoop Dist of Choice
Spark, Storm
Flume, Scibe, Chukwa
Nutch, Talend, Scraperwiki
Webscraper, Flume, Sqoop
tm, RWeka, NLTK
RHIPE
D3.js, ggplot2, Shiny
IBM Languageware
Cassandra, MongoDB
100%
Author: Swami Chandrasekaran


Allo, allo, ici la Terre !...
Allo, j'appelle fusée lu-
naire ...Allo, allo !...
Applications
Algorithms
Theory

1 A Theory
of Statistical Learning

{Statistical,Machine} Learning: building automatic procedures to infer general rules from examples.

{Statistical,Machine} Learning: building automatic procedures to infer general rules from examples.

In the (~~rather~~ not so?) long term: mimic the inductive functioning of the humain brain to develop an artificial intelligence.

{Statistical,Machine} Learning: building automatic procedures to infer general rules from examples.

In the (~~rather~~ not so?) long term: mimic the inductive functioning of the humain brain to develop an artificial intelligence.

In the Big Data Era, very dynamic field at the crossroads of Computer Science and Statistics. Strategic focus at Inria!

Probabilistic framework: n-sample $\mathcal{D}_n = (\mathbf{X}_i, \mathbf{Y}_i)_{i=1}^n$ of i.i.d. replications of some random variable

$$(\mathbf{X}, \mathbf{Y}) \in \mathcal{X} \times \mathcal{Y},$$

$$\dim(\mathcal{X}) = d, \quad \dim(\mathcal{Y}) = m.$$

Probabilistic framework: n-sample $\mathcal{D}_n = (\mathbf{X}_i, \mathbf{Y}_i)_{i=1}^n$ of i.i.d. replications of some random variable

$$(\mathbf{X}, \mathbf{Y}) \in \mathcal{X} \times \mathcal{Y},$$

$$\dim(\mathcal{X}) = d, \quad \dim(\mathcal{Y}) = m.$$

We want to infer the link between the explanatory variable $\mathbf{X}$ and the response variable $\mathbf{Y}$, *i.e.*, use $\mathcal{D}_n$ to build up $\hat{\phi}$ such that $\hat{\phi}(\mathbf{X})$ is a "good" approximation of $\mathbf{Y}$.

Probabilistic framework: n-sample $\mathcal{D}_n = (\mathbf{X}_i, \mathbf{Y}_i)_{i=1}^n$ of i.i.d. replications of some random variable

$$(\mathbf{X}, \mathbf{Y}) \in \mathcal{X} \times \mathcal{Y},$$

$$\dim(\mathcal{X}) = d, \quad \dim(\mathcal{Y}) = m.$$

We want to infer the link between the explanatory variable $\mathbf{X}$ and the response variable $\mathbf{Y}$, *i.e.*, use $\mathcal{D}_n$ to build up $\hat{\phi}$ such that $\hat{\phi}(\mathbf{X})$ is a "good" approximation of $\mathbf{Y}$.

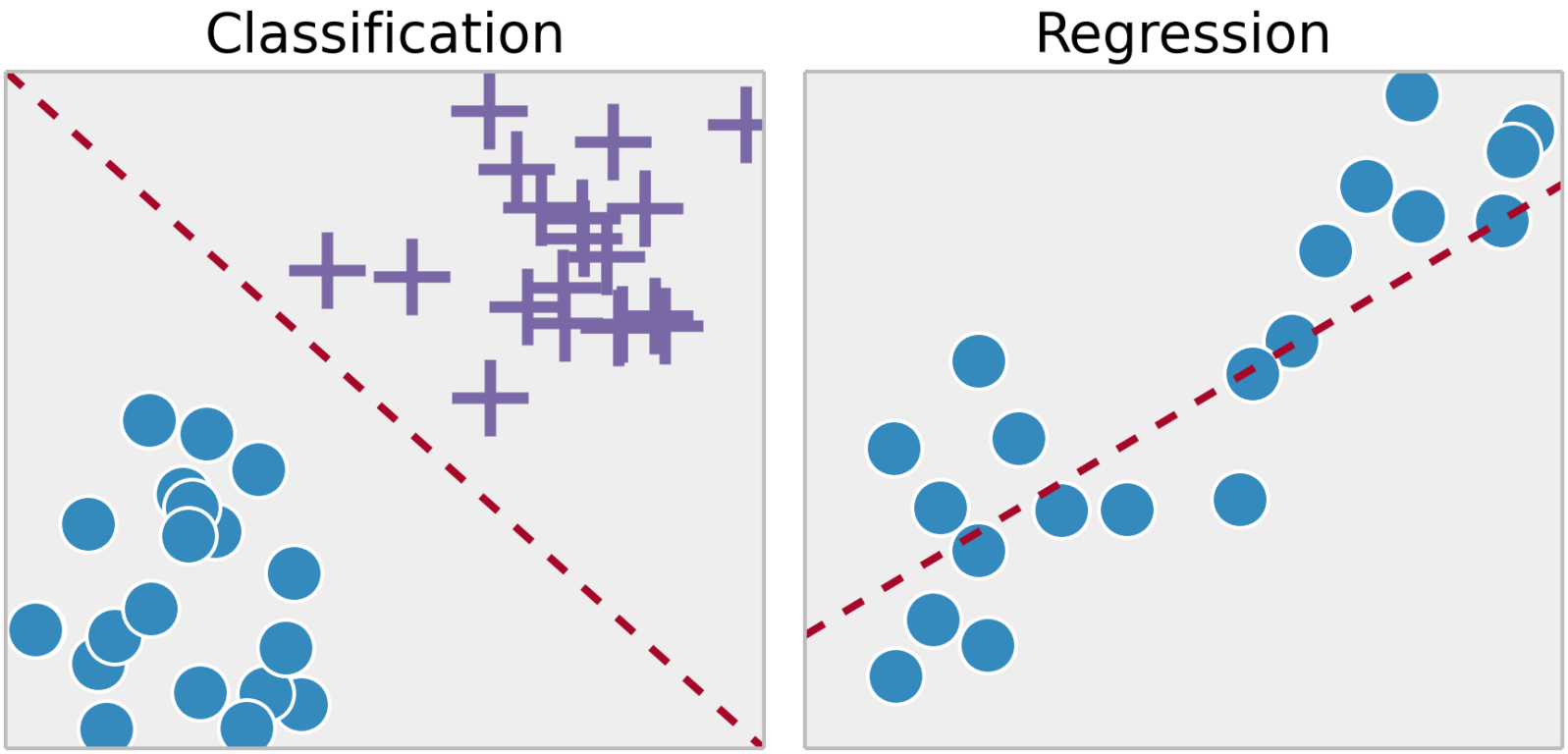
- Classification: $\mathcal{Y}$ is discrete.
- Regression: $\mathcal{Y}$ is a continuum.

- Online learning: observations are revealed over time.
- Batch learning: all observations are revealed at once.

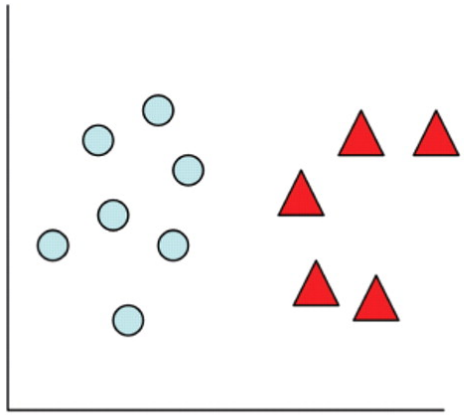
- Online learning: observations are revealed over time.
- Batch learning: all observations are revealed at once.

Big Data Era: easy/cheap to collect massive amounts of data, hence typically $\mathcal{X} = \mathbb{R}^d$ where d may be (extremely) large.
In the sequel, $\dim(\mathcal{Y}) = 1$ and $\mathbf{X} = (X_1, \ldots, X_d)$.

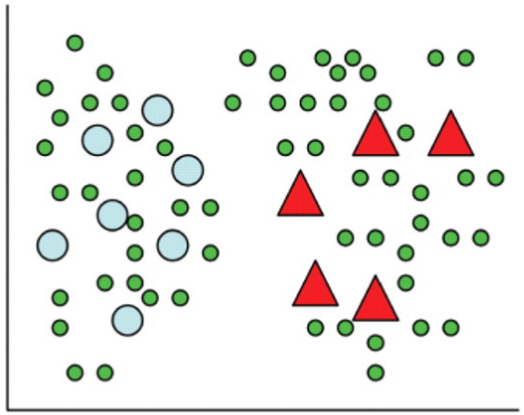
- Supervised learning: all of the $\mathbf{Y}_i$s are observed.



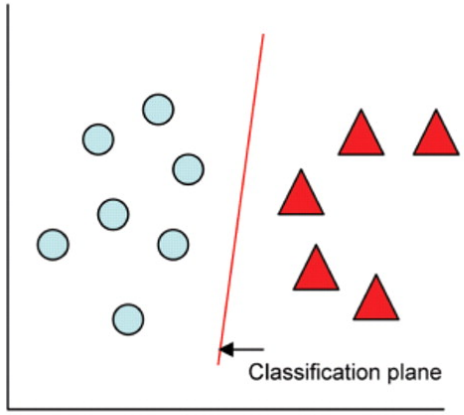
- Semi-supervised learning: some of the $\mathbf{Y}_i$s are observed (labeling is expensive or difficult).



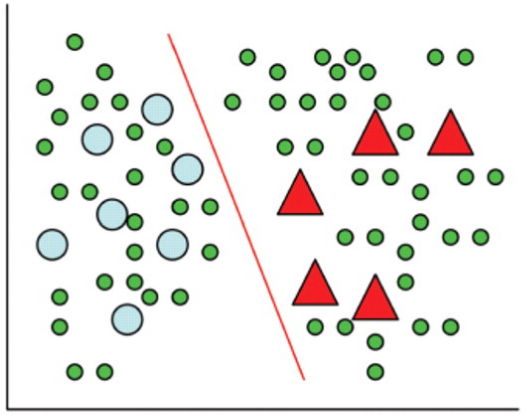
Labeled Data
(a)



Labeled and Unlabeled Data
(b)

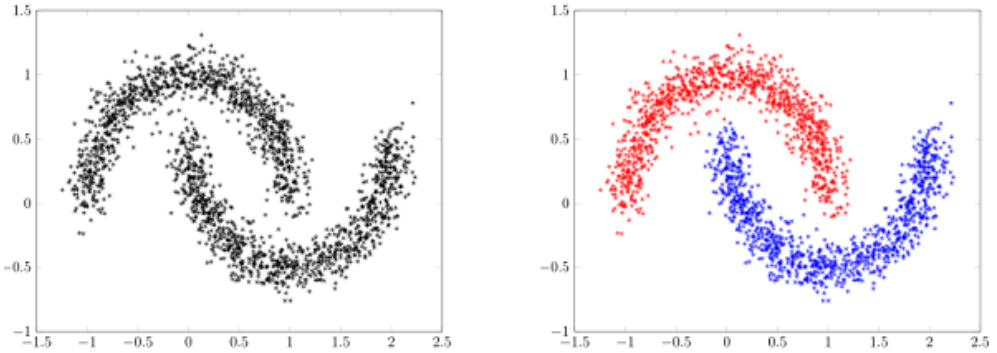


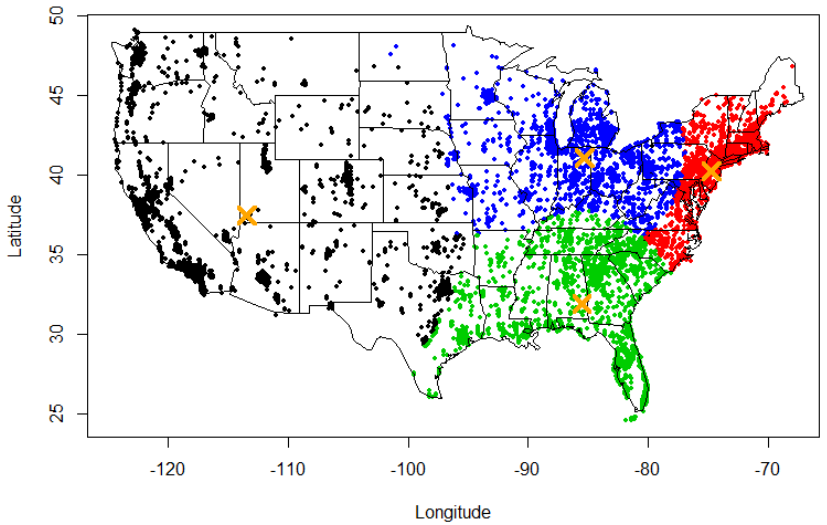
Supervised Learning
(c)



Semi-Supervised Learning
(d)

- Unsupervised learning: none of the $\mathbf{Y}_i$s are observed (detect patterns).





- Unsupervised learning: none of the $\mathbf{Y}_i$s are observed (detect patterns).

- Reinforcement learning: possible feedback from the environment (robotics, adversarial environments, training...).

Loss function:

$$\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+.$$

Loss function:

$$\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+.$$

$\ell(\hat{\phi}(\mathbf{X}), Y)$ (random) quantifies how a predictor $\hat{\phi}(\mathbf{X})$ is a "good" approximation of $\mathbf{Y}$.

Loss function:

$$\ell \colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+.$$

$\ell(\hat{\phi}(\mathbf{X}), Y)$ (random) quantifies how a predictor $\hat{\phi}(\mathbf{X})$ is a "good" approximation of $\mathbf{Y}$.

A predictor is any mapping

$$\hat{\phi} \colon (\mathcal{X} \times \mathcal{Y})^n \to \mathcal{Y}^{\mathcal{X}}.$$

Loss function:

$$\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+.$$

$\ell(\hat{\phi}(\mathbf{X}), Y)$ (random) quantifies how a predictor $\hat{\phi}(\mathbf{X})$ is a "good" approximation of $\mathbf{Y}$.

A predictor is any mapping

$$\hat{\phi} : (\mathcal{X} \times \mathcal{Y})^n \to \mathcal{Y}^{\mathcal{X}}.$$

Risk:

$$R(\hat{\phi}) = \mathbb{E}\left[\ell\left(\hat{\phi}(\mathbf{X}), Y\right)\right].$$

- Quadratic loss (regression): $\ell(a, b) = (a - b)^2$.

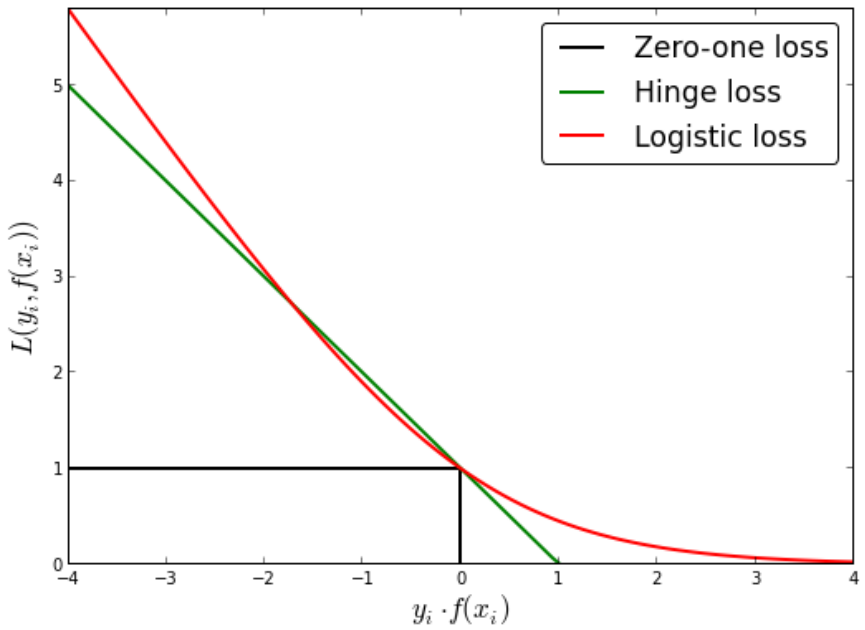
- Quadratic loss (regression): $\ell(a, b) = (a - b)^2$.
- Absolute loss (regression): $\ell(a, b) = |a - b|$.

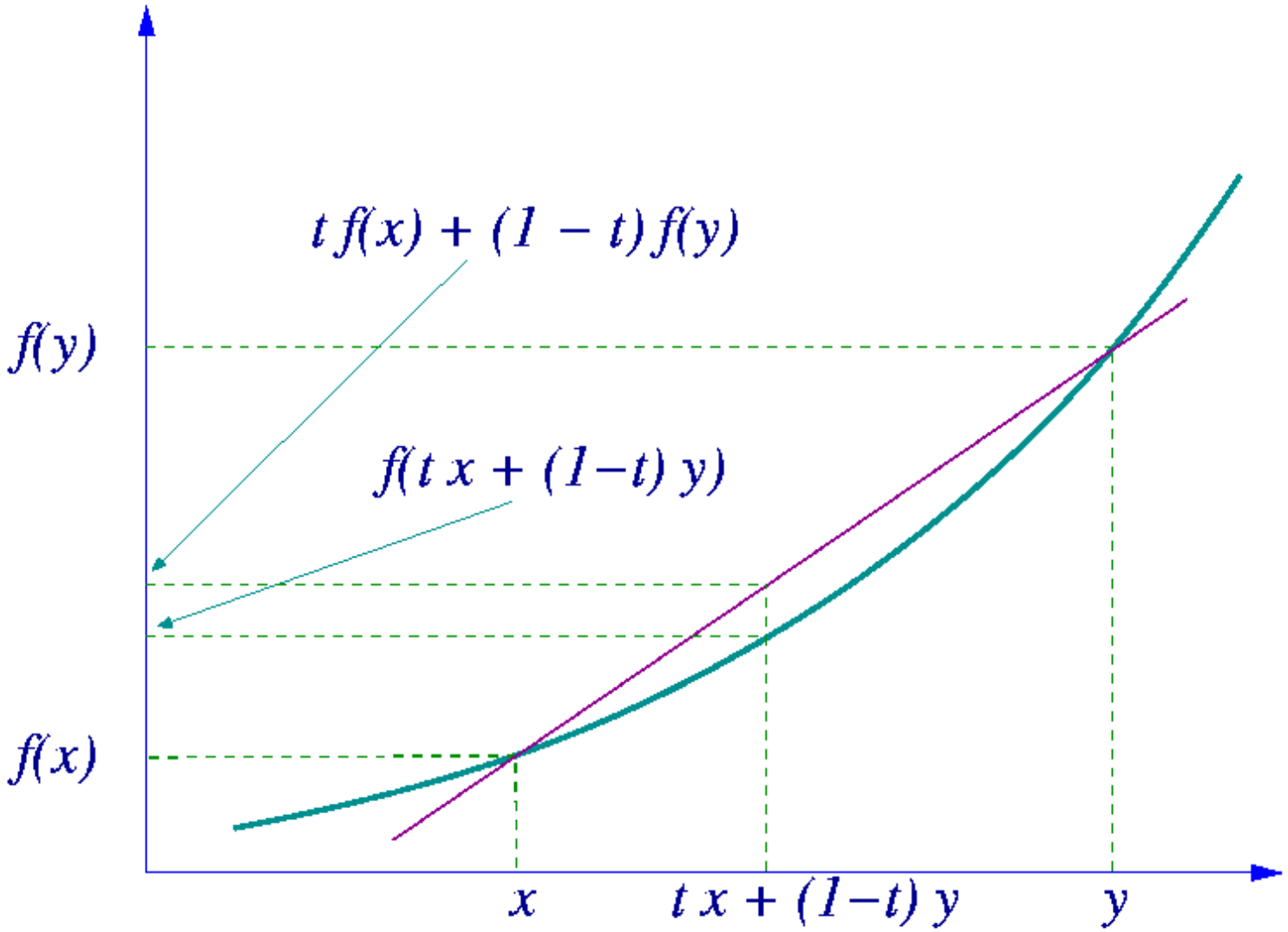
- Quadratic loss (regression): $\ell(a, b) = (a - b)^2$.
- Absolute loss (regression): $\ell(a, b) = |a - b|$.
- 0-1 loss (classification): $\ell(a, b) = \mathbb{1}_{\{a \neq b\}}$.

- Quadratic loss (regression): $\ell(a, b) = (a - b)^2$.
- Absolute loss (regression): $\ell(a, b) = |a - b|$.
- 0-1 loss (classification): $\ell(a, b) = \mathbb{1}_{\{a \neq b\}}$.
- Hinge loss (classification): $\ell(a, b) = \max(0, 1 - ab)$.

- Quadratic loss (regression): $\ell(a, b) = (a - b)^2$.
- Absolute loss (regression): $\ell(a, b) = |a - b|$.
- 0-1 loss (classification): $\ell(a, b) = \mathbb{1}_{\{a \neq b\}}$.
- Hinge loss (classification): $\ell(a, b) = \max(0, 1 - ab)$.
- Logistic loss (classification): $\ell(a, b) = \log[1 + \exp(-ab)]$.

- Quadratic loss (regression): $\ell(a, b) = (a - b)^2$.
- Absolute loss (regression): $\ell(a, b) = |a - b|$.
- 0-1 loss (classification): $\ell(a, b) = \mathbb{1}_{\{a \neq b\}}$.
- Hinge loss (classification): $\ell(a, b) = \max(0, 1 - ab)$.
- Logistic loss (classification): $\ell(a, b) = \log[1 + \exp(-ab)]$.




t f(x) + (1 − t) f(y)
f(y)
f(t x + (1−t) y)
f(x)
x
t x + (1−t) y
y

Statistical Learning vs. Machine Learning

Different approaches:

- In machine learning, given some dataset $(\mathbf{x}_i, y_i)$, solve

$$\widehat{\phi}(\cdot) = \arg\min_m \left\{ \sum_{i=1}^{n} \ell(y_i, m(\mathbf{x}_i) \right\}.$$

- In statistical modeling, assume that the Y_is are realisations of some random variable Y (given $\mathbf{X}$) with distribution P. Solve

$$\widehat{\phi}(\cdot) = \arg\max_m \left\{ \sum_{i=1}^{n} \log \mathrm{d}P(y_i, m(\mathbf{x}_i)) \right\}.$$

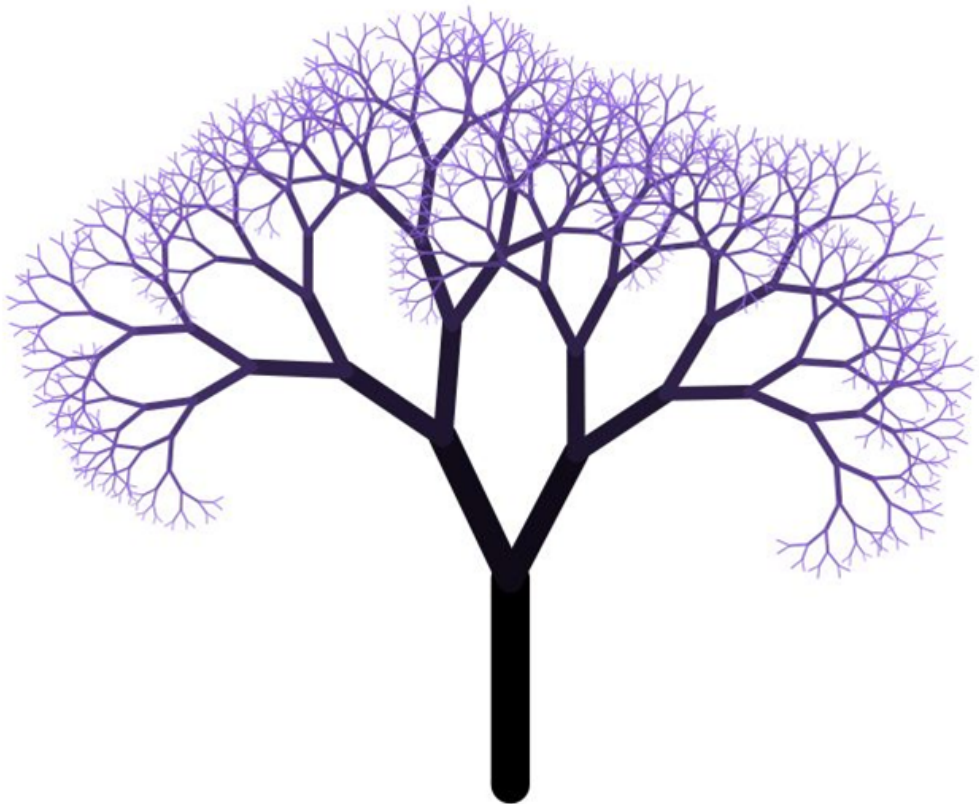
2 Algorithms

Decision trees and random forests

Leo Breiman (1928–2005)

Binary Tree

- Build recursively a binary decision tree until a stop criterion is met.
- No model! This is a purely data-driven method.



In a nutshell

- Principle: Build a partition of $\mathbb{R}^d$ using (simple) hyperplans. The cells of the partition are the leaves of the tree.

In a nutshell

- Principle: Build a partition of $\mathbb{R}^d$ using (simple) hyperplans. The cells of the partition are the leaves of the tree.
- How the tree is grown: At each node, select a covariate X_j ($j \in \{1, \ldots, d\}$) and a cut $\{X_j \geq k\} \cup \{X_j < k\}$ for some $k \in \mathbb{R}$, using some criterion.

In a nutshell

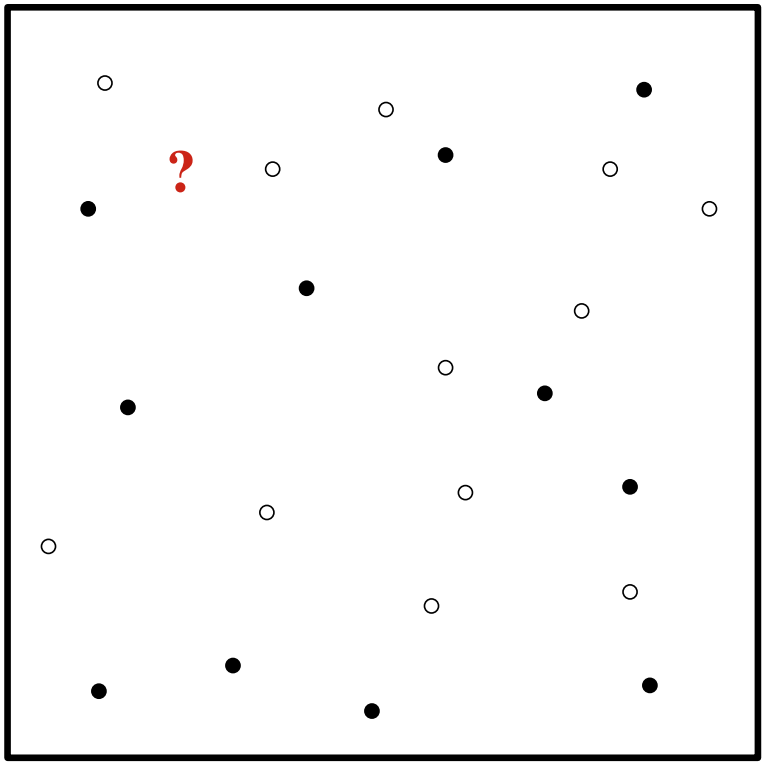
- Principle: Build a partition of $\mathbb{R}^d$ using (simple) hyperplans. The cells of the partition are the leaves of the tree.
- How the tree is grown: At each node, select a covariate X_j ($j \in \{1, \ldots, d\}$) and a cut $\{X_j \geq k\} \cup \{X_j < k\}$ for some $k \in \mathbb{R}$, using some criterion.

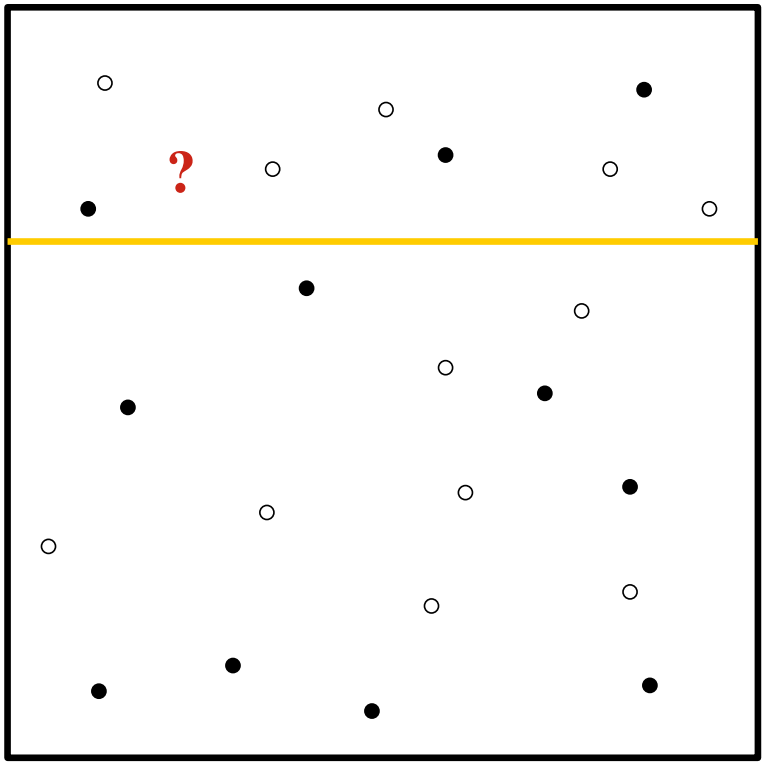
- Classification: The label of a new data point is obtained through a majority vote within its cell / leaf.

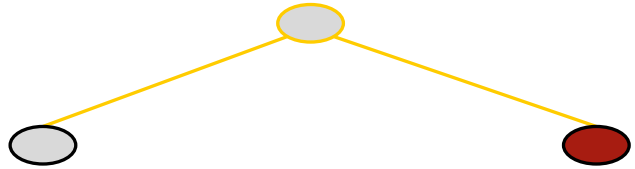
In a nutshell

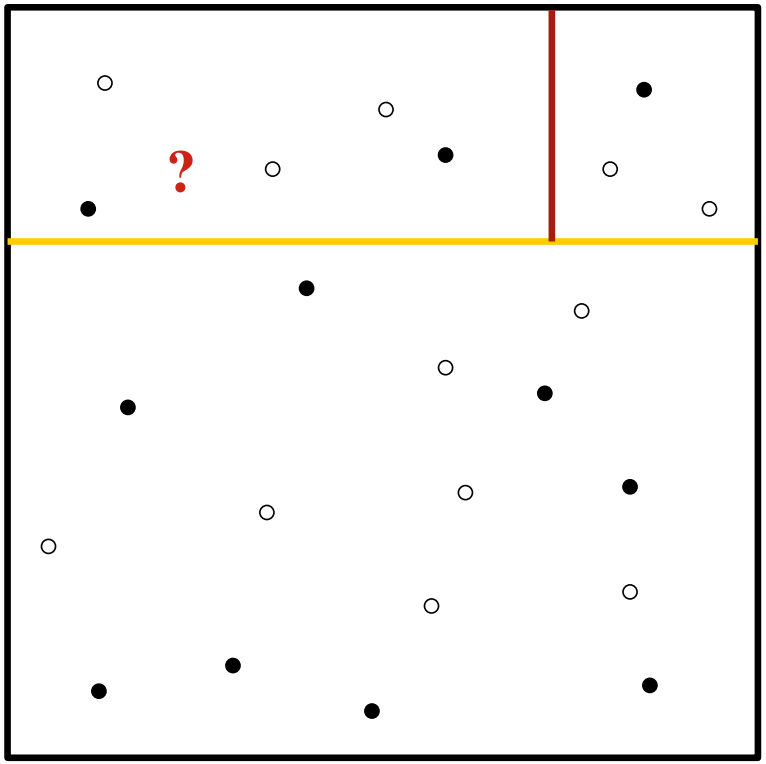
- Principle: Build a partition of $\mathbb{R}^d$ using (simple) hyperplans. The cells of the partition are the leaves of the tree.
- How the tree is grown: At each node, select a covariate X_j ($j \in \{1, \ldots, d\}$) and a cut $\{X_j \geq k\} \cup \{X_j < k\}$ for some $k \in \mathbb{R}$, using some criterion.

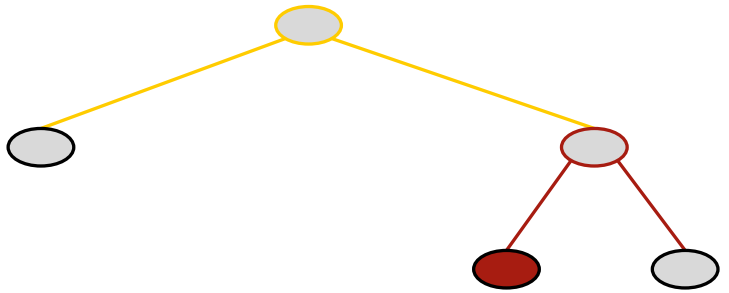
- Classification: The label of a new data point is obtained through a majority vote within its cell / leaf.
- Regression: The response of a new data point is the mean of its cell / leaf.

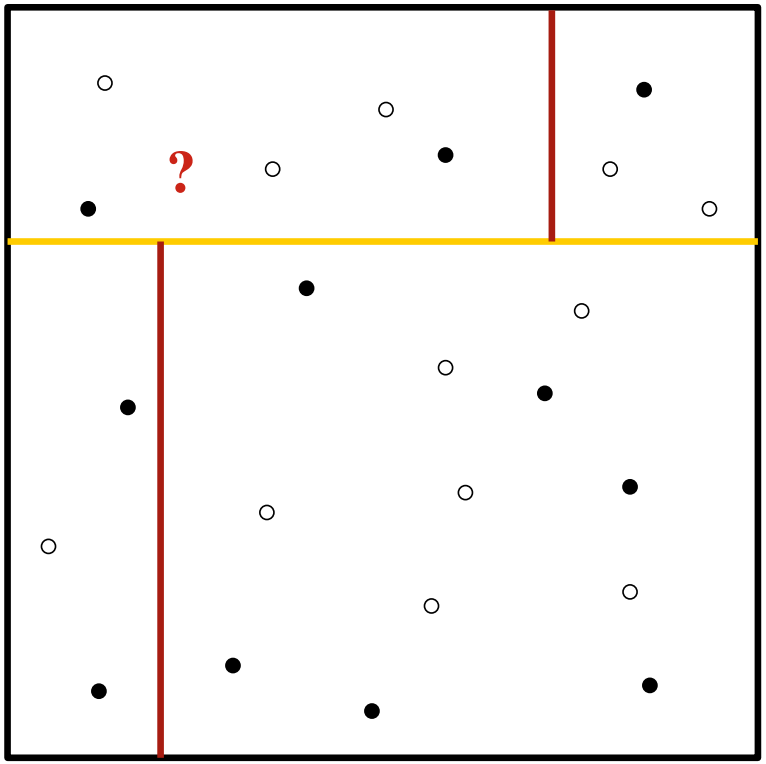


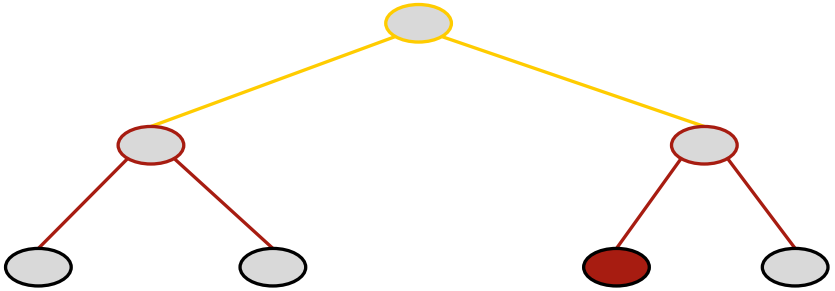


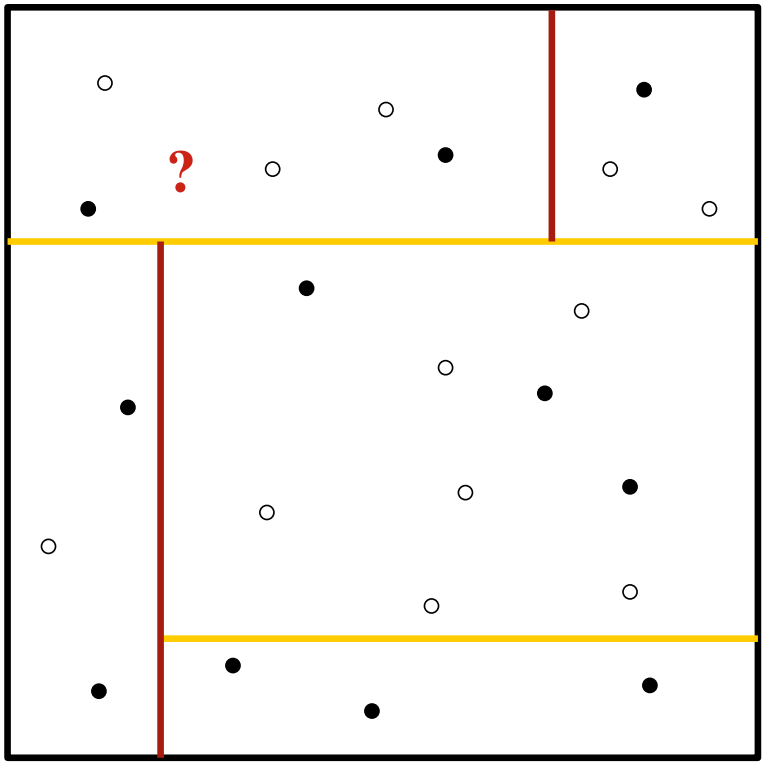


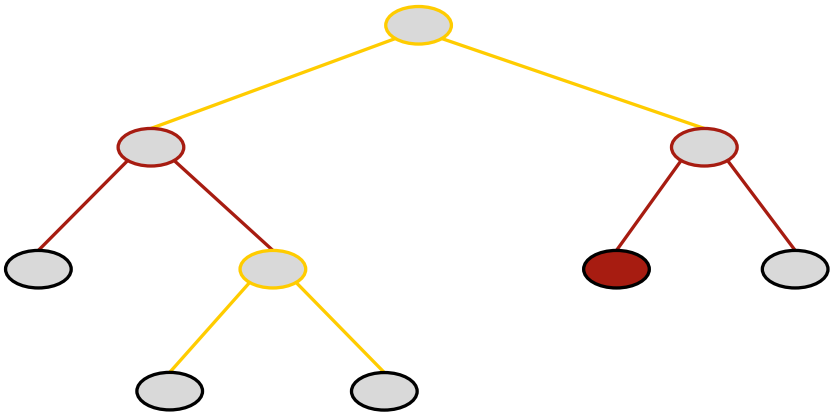


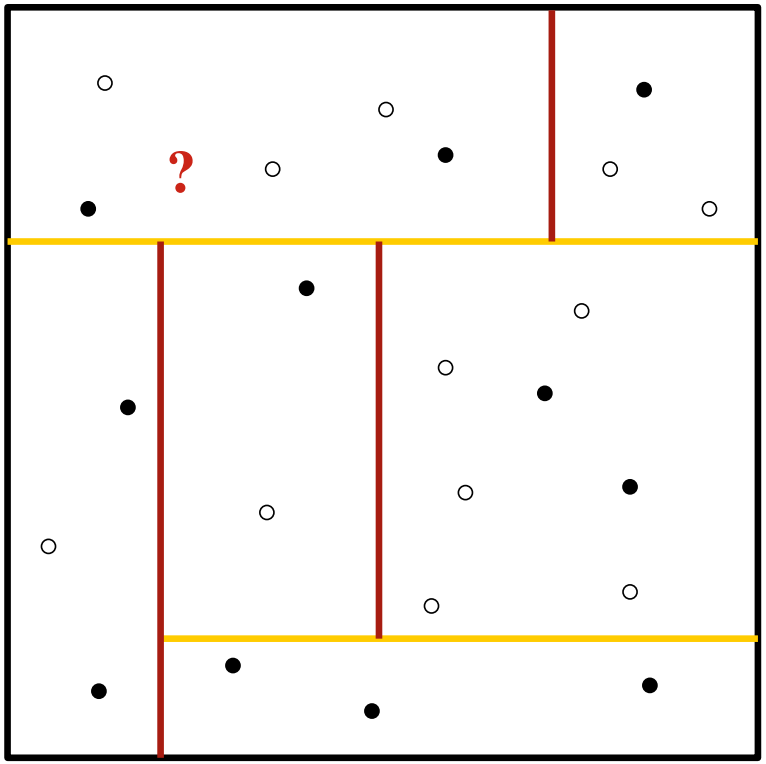


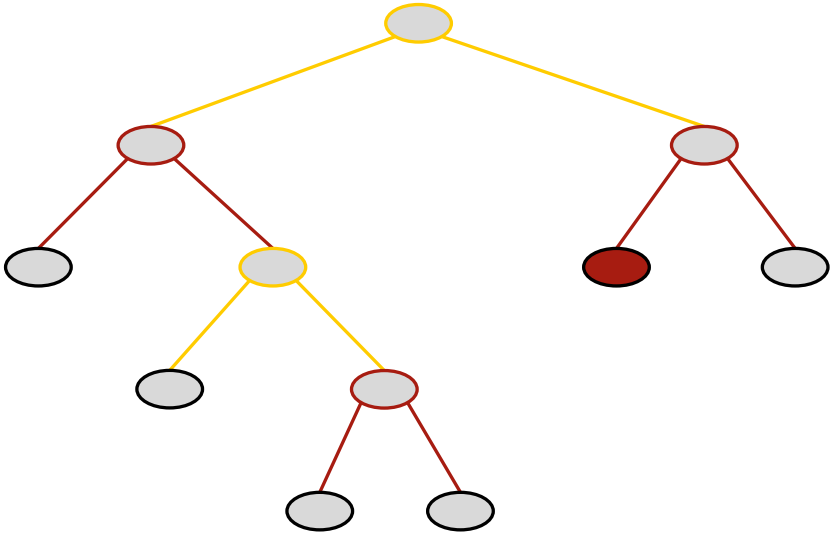


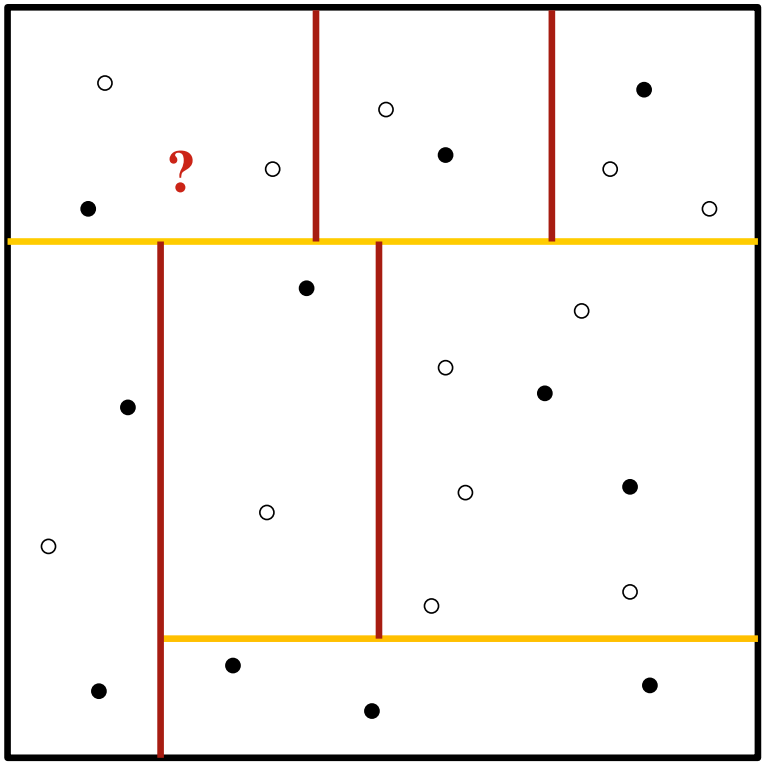


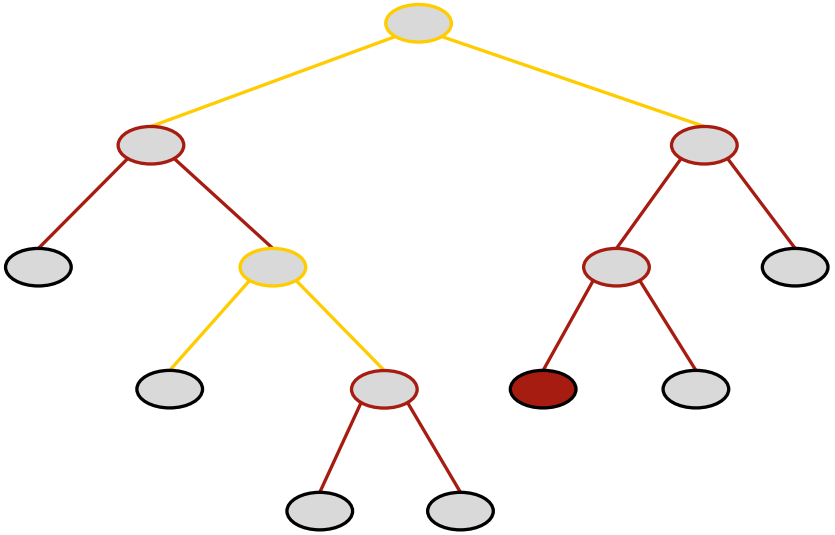












Pseudocode

Initialization. TREE = root
Expansion. For each node N of TREE
If N does not meet the stop criterion
Create subnodes
TREE = TREE + (subnodes)
End If
Pruning. For each node N of TREE
If N meets the pruning condition
TREE = TREE - (node N and its subnodes)
End If

Cutting

- Classification: Our goal is to obtain homogeneous groups. Achieved by minimizing the Gini Index $\mathcal{I}_G$ at each node f:

$$\mathcal{I}_G(f) = 1 - \sum_{i=1}^{2} f_i^2,$$

where f_i is the fraction of observations with label i in the node f.

- Regression: Our goal is to have the smaller possible variance within each node. Achieved by minimizing the empirical variance

$$\frac{1}{n_f} \sum_{i \in f} (Y_i - \bar{Y}_f)^2$$

at each node f.

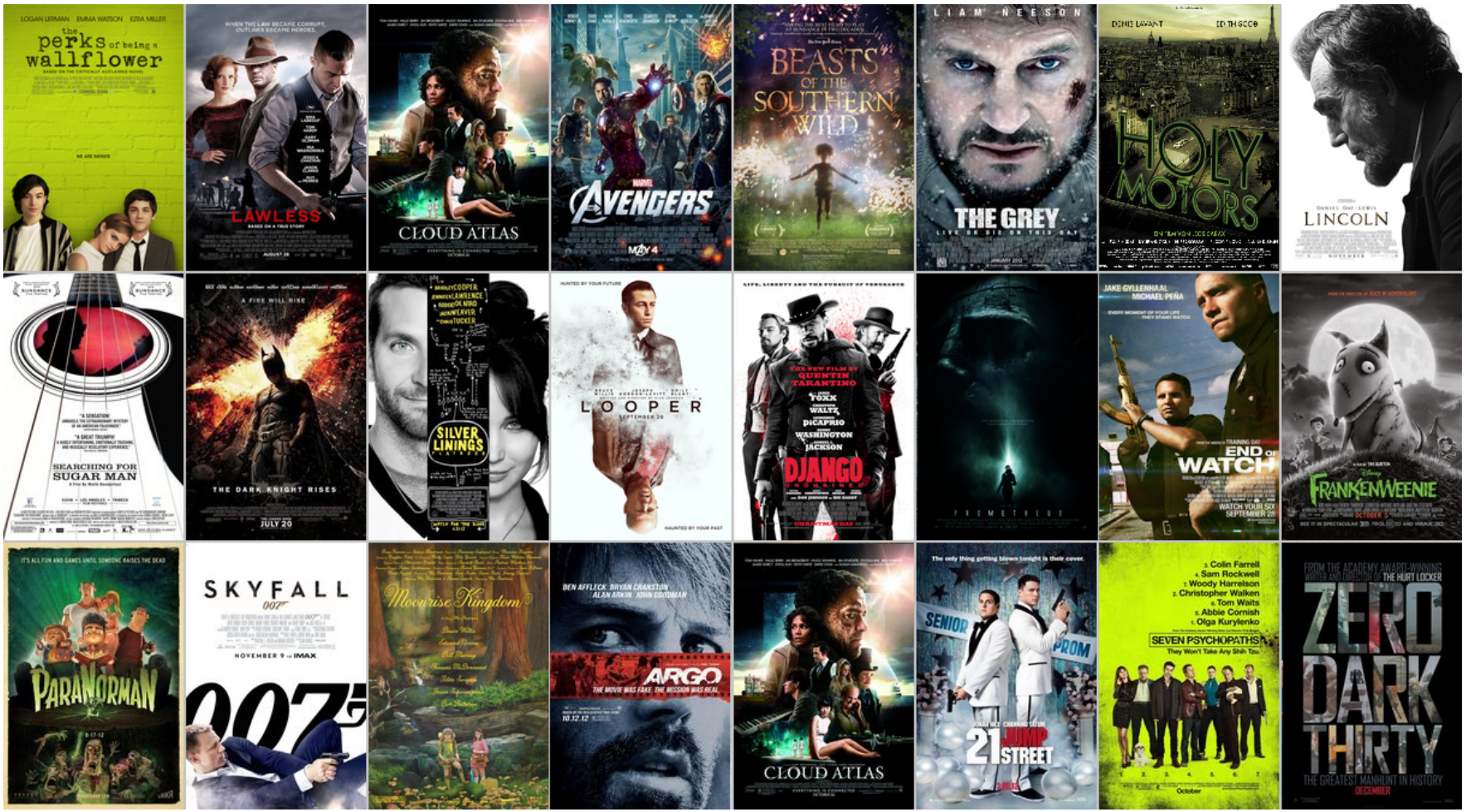
Stop criterion, pruning & importance

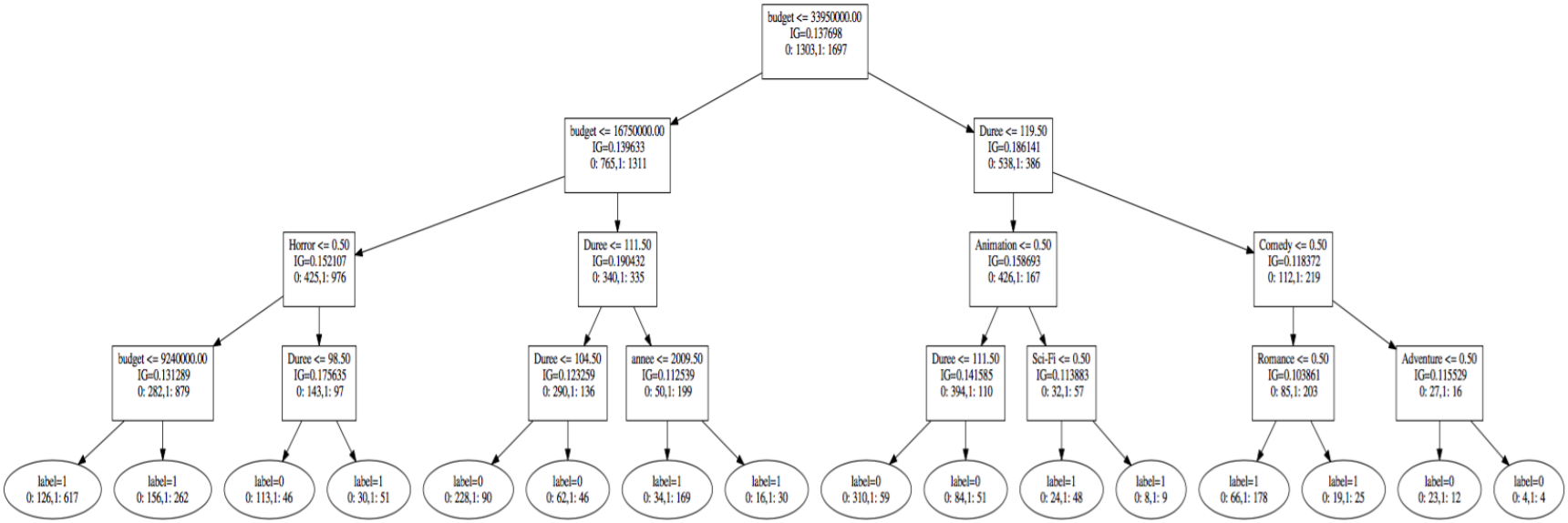
- Typically, the algorithm stops when the tree reaches a fixed depth, when the number of leaves exceeds a fixed bound, or when the number of individuals / points in each nodes falls below a fixed threshold.

- The maximal tree might not be the best choice: it has a large variance and is computationally more demanding. Hence the idea of pruning (finding the "best" sub-tree in the maximal tree). We may remove a branch if it does not degrade "too much" the predictive performance $\Rightarrow$ compromise between the performance and the number of leaves.

- Variable selection: notion of importance.

NETFLIX


the perks of being a wallflower
LAWLESS
CLOUD ATLAS
AVENGERS
BEASTS OF THE SOUTHERN WILD
THE GREY
HOLY MOTORS
LINCOLN
SEARCHING FOR SUGAR MAN
THE DARK KNIGHT RISES
SILVER LININGS PLAYBOOK
LOOPER
DJANGO UNCHAINED
PROMETHEUS
END OF WATCH
FRANKENWEENIE
PARANORMAN
SKYFALL
007
Moonrise Kingdom
ARGO
CLOUD ATLAS
21 JUMP STREET
SEVEN PSYCHOPATHS
ZERO DARK THIRTY


budget <= 33950000.00
IG=0.137698
0: 1303,1: 1697
budget <= 16750000.00
IG=0.139633
0: 765,1: 1311
Duree <= 119.50
IG=0.186141
0: 538,1: 386
Horror <= 0.50
IG=0.152107
0: 425,1: 976
Duree <= 111.50
IG=0.190432
0: 340,1: 335
Animation <= 0.50
IG=0.158693
0: 426,1: 167
Comedy <= 0.50
IG=0.118372
0: 112,1: 219
budget <= 9240000.00
IG=0.131289
0: 282,1: 879
Duree <= 98.50
IG=0.175635
0: 143,1: 97
Duree <= 104.50
IG=0.123259
0: 290,1: 136
annee <= 2009.50
IG=0.112539
0: 50,1: 199
Duree <= 111.50
IG=0.141585
0: 394,1: 110
Sci-Fi <= 0.50
IG=0.113883
0: 32,1: 57
Romance <= 0.50
IG=0.103861
0: 85,1: 203
Adventure <= 0.50
IG=0.115529
0: 27,1: 16
label=1
0: 126,1: 617
label=1
0: 156,1: 262
label=0
0: 113,1: 46
label=1
0: 30,1: 51
label=0
0: 228,1: 90
label=0
0: 62,1: 46
label=1
0: 34,1: 169
label=1
0: 16,1: 30
label=0
0: 310,1: 59
label=0
0: 84,1: 51
label=1
0: 24,1: 48
label=1
0: 8,1: 9
label=1
0: 66,1: 178
label=1
0: 19,1: 25
label=0
0: 23,1: 12
label=0
0: 4,1: 4

About the unstability and lack of robustness: averaging M trees still delivers a predictor $\mathcal{X} \to \mathcal{Y}$, and massively improves its stability. We call a set of $M \geq 2$ trees a... forest.

Simple (unoptimized) and randomized (different) trees:

- No pruning
- At each node, select only m_{try} variables among $\{1, \ldots, d\}$, and propose a cut with the CART criterion, or even a random cut

A cut may be represented by a vector $\theta = (\theta_1, \ldots, \theta_N)$ where N is the number of nodes (same for all trees). Let m_θ denote the tree formed with the cut θ.

Denote $(\theta^1, \ldots, \theta^M)$ a sequence of M cuts, a random forest for regression is

$$m \colon \mathbf{x} \mapsto \frac{1}{M} \sum_{j=1}^{M} m_{\theta^j}(\mathbf{x}).$$

R package `randomForest`.

Trees and forests

- A great tool!
  - Nonparametric & no model specification
  - Performs classification & regression
  - Qualitative and quantitative input variables
  - Easy interpretation (trees)
  - Deals with complex and massive data

- Sadly not perfect...
  - Very few theoretical results (no oracle inequalities)
  - Unstable: performance highly depends on the cut (trees)
  - No robustness to outliers (trees)
  - No interpretation (forests)

2 Algorithms

Regularized regression

Simple linear regression model

$$Y_i = \mathbf{X}_i \beta^\star + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, \sigma_i^2), \quad i = 1, \ldots, n.$$

In the Gauss-Markov model (the errors are homoscedastic, *i.e.*, $\sigma_i^2 = \sigma^2 \ \forall i$, and the $(X_{ij})_j$ are independent), least-squares fit

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} = \underset{\beta \in \mathbb{R}^d}{\arg\min} \ ||\mathbf{Y} - \mathbf{X}\beta||^2.$$

Curse of dimensionality

Assuming that $\mathbf{X}'\mathbf{X}$ is invertible (restrictive), its computational cost is of magnitude $\mathcal{O}(d^3)$.

Curse of dimensionality

Assuming that $\mathbf{X}'\mathbf{X}$ is invertible (restrictive), its computational cost is of magnitude $\mathcal{O}(d^3)$.

Solution: assume a sparse representation of $\beta^\star$, *i.e.*, with at most $d_0 \ll \min(n, d)$ nonzero components.

Curse of dimensionality

Assuming that $\mathbf{X}'\mathbf{X}$ is invertible (restrictive), its computational cost is of magnitude $\mathcal{O}(d^3)$.

Solution: assume a sparse representation of $\beta^\star$, *i.e.*, with at most $d_0 \ll \min(n, d)$ nonzero components.

If those components were known, we could build the least squares estimator $\hat{\beta}_0$ and obtain oracle inequalities of the flavor

$$\mathbb{E}[R(\hat{\beta}_0) - R(\beta^\star)] \leq \text{cste} \times \frac{\sigma^2 d_0}{n}.$$

Curse of dimensionality

Assuming that $\mathbf{X}'\mathbf{X}$ is invertible (restrictive), its computational cost is of magnitude $\mathcal{O}(d^3)$.

Solution: assume a sparse representation of $\beta^\star$, *i.e.*, with at most $d_0 \ll \min(n, d)$ nonzero components.

If those components were known, we could build the least squares estimator $\hat{\beta}_0$ and obtain oracle inequalities of the flavor

$$\mathbb{E}[R(\hat{\beta}_0) - R(\beta^\star)] \leq \text{cste} \times \frac{\sigma^2 d_0}{n}.$$

Obviously, neither those components nor d_0 are known!

ℓ^0 penalty

- Natural idea: extend least squares minimization by penalizing the number of nonzero coordinates: for some $\lambda > 0$,

$$\hat{\beta}_\lambda \in \underset{\beta \in \mathbb{R}^d}{\arg\min} \{||\mathbf{Y} - \mathbf{X}\beta||^2 + \lambda ||\beta||_0\}.$$

ℓ^0 penalty

- Natural idea: extend least squares minimization by penalizing the number of nonzero coordinates: for some $\lambda > 0$,

$$\hat{\beta}_\lambda \in \arg\min_{\beta \in \mathbb{R}^d} \{||\mathbf{Y} - \mathbf{X}\beta||^2 + \lambda ||\beta||_0\}.$$

- This approach—ℓ^0-penalization— is known to deliver solid theoretical guarantee: for $\lambda \asymp \log(d)/n$,

$$\mathbb{E}[R(\hat{\beta}_\lambda) - R(\beta^\star)] \leq \text{cst} \times \frac{\sigma^2 d_0 \log(d)}{n}.$$

ℓ^0 penalty

- Natural idea: extend least squares minimization by penalizing the number of nonzero coordinates: for some $\lambda > 0$,

$$\hat{\beta}_\lambda \in \arg\min_{\beta \in \mathbb{R}^d} \{||\mathbf{Y} - \mathbf{X}\beta||^2 + \lambda ||\beta||_0\}.$$

- This approach—ℓ^0-penalization— is known to deliver solid theoretical guarantee: for $\lambda \asymp \log(d)/n$,

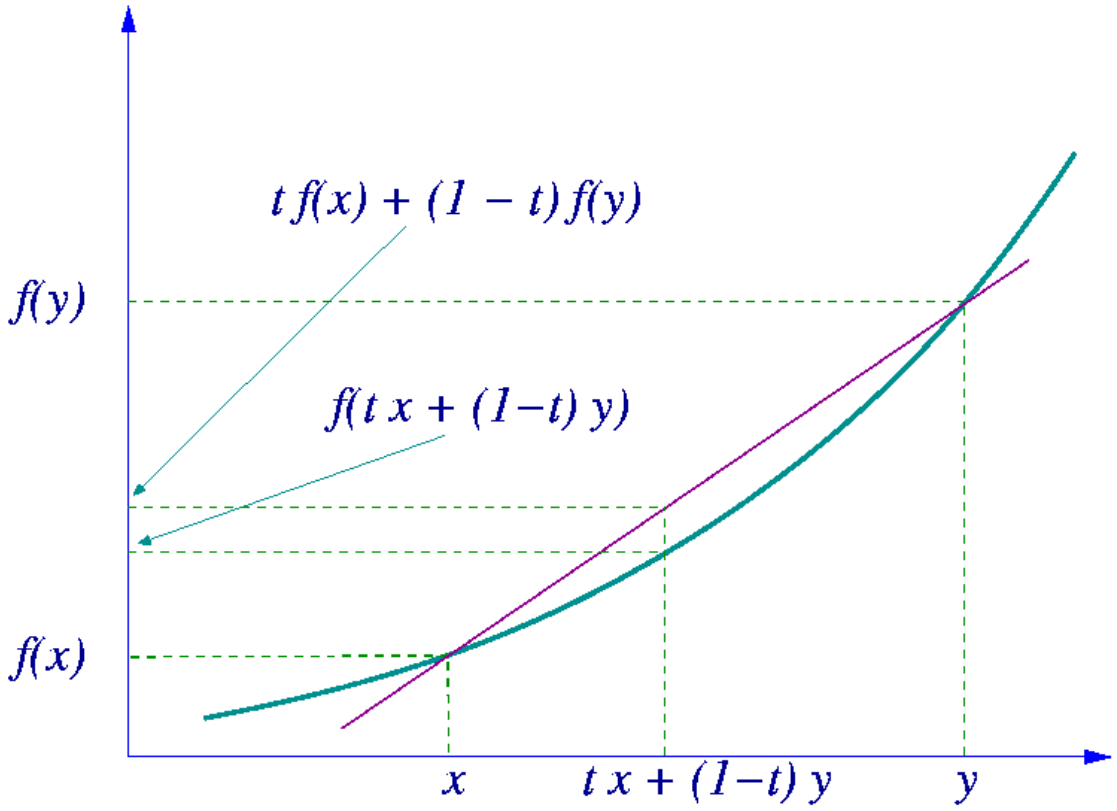
$$\mathbb{E}[R(\hat{\beta}_\lambda) - R(\beta^\star)] \leq \mathrm{cst} \times \frac{\sigma^2 d_0 \log(d)}{n}.$$

- Unfortunately, computing such estimators is out of reach as soon as d is a few tens.

Convexity (the return)

Convex function f: for any $t \in (0,1)$,

$$f\left(tx + (1-t)y\right) \leq tf(x) + (1-t)f(y).$$



Convex surrogates for the ℓ^0 norm

$||\theta||_0 = \#\{i|\theta_i \neq 0\}$. ℓ^p norm for $p \in \{1, 2, \ldots\}$:

$$||\theta||_p = \left(\sum_{j=1}^{d} |\theta_j|^p \right)^{\frac{1}{p}}.$$

Convex surrogates for the ℓ^0 norm

$||\theta||_0 = \#\{i|\theta_i \neq 0\}$. ℓ^p norm for $p \in \{1, 2, \ldots\}$:

$$||\theta||_p = \left(\sum_{j=1}^{d} |\theta_j|^p \right)^{\frac{1}{p}}.$$

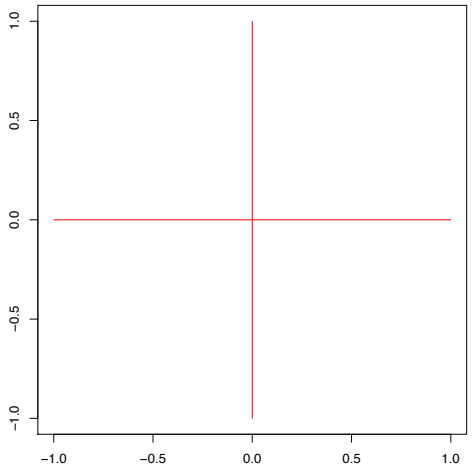
Unit ball in ℓ^0, ℓ^1 and ℓ^2 norms.

Convex surrogates for the ℓ^0 norm

$||\theta||_0 = \#\{i|\theta_i \neq 0\}$. ℓ^p norm for $p \in \{1, 2, \ldots\}$:

$$||\theta||_p = \left(\sum_{j=1}^{d} |\theta_j|^p \right)^{\frac{1}{p}}.$$

Unit ball in ℓ^0, ℓ^1 and ℓ^2 norms.

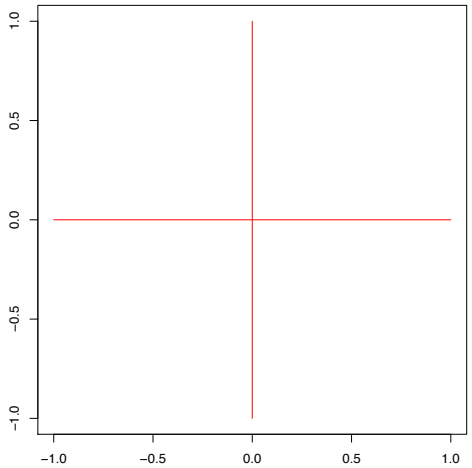


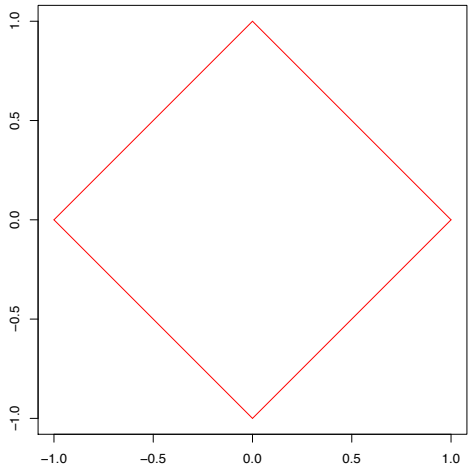
Convex surrogates for the ℓ^0 norm

$||\theta||_0 = \#\{i|\theta_i \neq 0\}$. ℓ^p norm for $p \in \{1, 2, \ldots\}$:

$$||\theta||_p = \left(\sum_{j=1}^{d} |\theta_j|^p \right)^{\frac{1}{p}}.$$

Unit ball in ℓ^0, ℓ^1 and ℓ^2 norms.



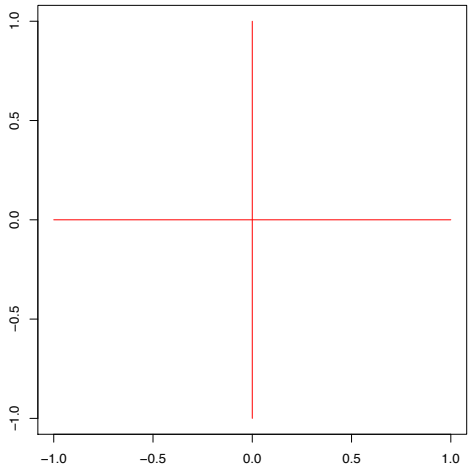


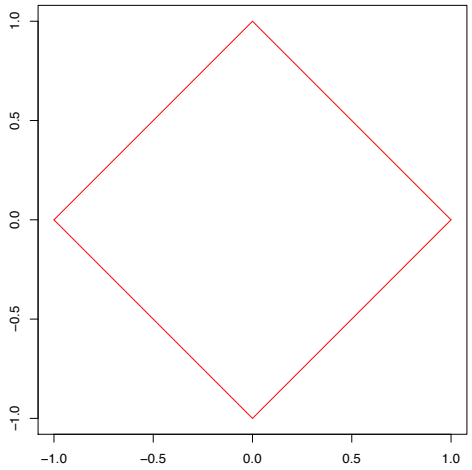
Convex surrogates for the ℓ^0 norm

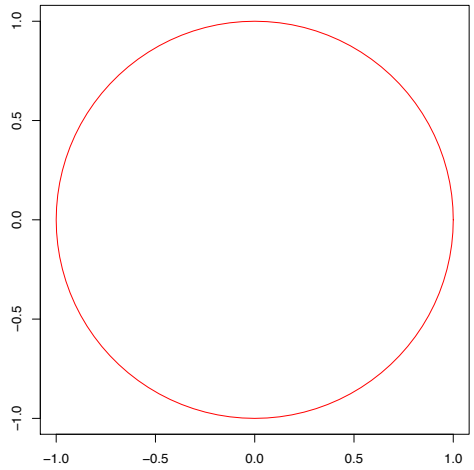
$||\theta||_0 = \#\{i|\theta_i \neq 0\}$. ℓ^p norm for $p \in \{1, 2, \ldots\}$:

$$||\theta||_p = \left(\sum_{j=1}^{d} |\theta_j|^p \right)^{\frac{1}{p}}.$$

Unit ball in ℓ^0, ℓ^1 and ℓ^2 norms.







ℓ^1 penalty

Goal: Circumvent the computational dead-end of the ℓ^0 penalty while preserving its good statistical properties.

ℓ^1 penalty

Goal: Circumvent the computational dead-end of the ℓ^0 penalty while preserving its good statistical properties.

Convexified problem

$$\hat{\beta}_\lambda \in \underset{\beta \in \mathbb{R}^d}{\arg\min} \{||\mathbf{Y} - \mathbf{X}\beta||^2 + \lambda||\beta||_1\}.$$

Oracle inequality

For $\lambda \asymp \log(d)/n$,

$$\mathbb{E}||\hat{\beta}_\lambda - \beta^\star||^2 \leq \mathrm{cst} \times d_0 \sqrt{\frac{\log(d)}{n}}.$$

Gradient Descent

Goal: minimize a differentiable function f (compute $\arg\min_x f(x)$).

Gradient Descent

Goal: minimize a differentiable function f (compute $\arg\min_x f(x)$).

Input: tolerance ϵ, initialization x_0, step size α.

Gradient Descent

Goal: minimize a differentiable function f (compute $\arg\min_x f(x)$).

Input: tolerance ϵ, initialization x_0, step size α.

While $f'(x_k) \geq \epsilon$ $x_{k+1} = x_k - \alpha f'(x_k)$

A Variety of Penalties

Ridge regression (a.k.a. Tikhonov regularization):

$$\hat{\beta}_\lambda \in \underset{\beta \in \mathbb{R}^d}{\arg\min} \left\{ ||\mathbf{Y} - \mathbf{X}\beta||^2 + \lambda ||\beta||_2^2 \right\}.$$

A Variety of Penalties

Ridge regression (a.k.a. Tikhonov regularization):

$$\hat{\beta}_\lambda \in \underset{\beta \in \mathbb{R}^d}{\arg\min} \left\{ ||\mathbf{Y} - \mathbf{X}\beta||^2 + \lambda ||\beta||_2^2 \right\}.$$

Fused Lasso:

$$\hat{\beta}_\lambda \in \underset{\beta \in \mathbb{R}^d}{\arg\min} \left\{ ||\mathbf{Y} - \mathbf{X}\beta||^2 + \lambda ||\beta||_1 + \lambda \sum_{j=2}^{d} |\beta_j - \beta_{j-1}| \right\}.$$

A Variety of Penalties

Smoothed Lasso:

$$\hat{\beta}_\lambda \in \underset{\beta \in \mathbb{R}^d}{\arg\min} \left\{ ||\mathbf{Y} - \mathbf{X}\beta||^2 + \lambda ||\beta||_1 + \lambda \sum_{j=2}^{d} (\beta_j - \beta_{j-1})^2 \right\}.$$

A Variety of Penalties

Smoothed Lasso:

$$\hat{\beta}_\lambda \in \underset{\beta \in \mathbb{R}^d}{\arg\min} \left\{ ||\mathbf{Y} - \mathbf{X}\beta||^2 + \lambda ||\beta||_1 + \lambda \sum_{j=2}^{d} (\beta_j - \beta_{j-1})^2 \right\}.$$

Elastic Net:

$$\hat{\beta}_\lambda \in \underset{\beta \in \mathbb{R}^d}{\arg\min} \left\{ ||\mathbf{Y} - \mathbf{X}\beta||^2 + \lambda ||\beta||_1 + \lambda ||\beta||_2^2 \right\}.$$

2 Algorithms

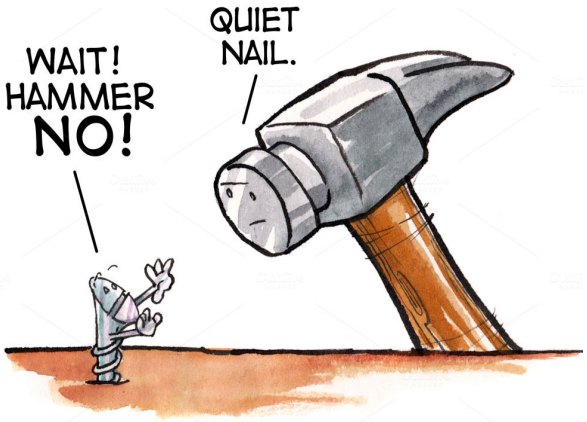
Aggregation

All models are wrong
but some are useful

George E.P. Box

If the only tool you have is a hammer, you tend to see every problem as a nail.

(Abraham Maslow)



Principle

Assume that over some preliminary sample $\mathcal{D}_k = (\mathbf{X}_i, Y_i)_{i=1}^k$, you have generated M different predictors $\hat{r}_{k,1}, \ldots, \hat{r}_{k,M}$, *i.e.*,

$$\hat{r}_{k,j} : \mathcal{X} \to \mathcal{Y}, \qquad \forall j = 1, \ldots, M.$$

Principle

Assume that over some preliminary sample $\mathcal{D}_k = (\mathbf{X}_i, Y_i)_{i=1}^k$, you have generated M different predictors $\hat{r}_{k,1}, \ldots, \hat{r}_{k,M}$, *i.e.*,

$$\hat{r}_{k,j} : \mathcal{X} \to \mathcal{Y}, \qquad \forall j = 1, \ldots, M.$$

Let

$$\mathbb{D} = \left\{ \hat{r}_{k,1}, \ldots, \hat{r}_{k,M} \right\}.$$

Principle

Assume that over some preliminary sample $\mathcal{D}_k = (\mathbf{X}_i, Y_i)_{i=1}^k$, you have generated M different predictors $\hat{r}_{k,1}, \ldots, \hat{r}_{k,M}$, *i.e.*,

$$\hat{r}_{k,j} : \mathcal{X} \to \mathcal{Y}, \qquad \forall j = 1, \ldots, M.$$

Let

$$\mathbb{D} = \{\hat{r}_{k,1}, \ldots, \hat{r}_{k,M}\}.$$

Let $\mathcal{D}_\ell = (\mathbf{X}_i, Y_i)_{i=1}^\ell$ be another sample.

Principle

Assume that over some preliminary sample $\mathcal{D}_k = (\mathbf{X}_i, Y_i)_{i=1}^k$, you have generated M different predictors $\hat{r}_{k,1}, \ldots, \hat{r}_{k,M}$, *i.e.*,

$$\hat{r}_{k,j} : \mathcal{X} \to \mathcal{Y}, \qquad \forall j = 1, \ldots, M.$$

Let

$$\mathbb{D} = \{\hat{r}_{k,1}, \ldots, \hat{r}_{k,M}\}.$$

Let $\mathcal{D}_\ell = (\mathbf{X}_i, Y_i)_{i=1}^\ell$ be another sample.

An *aggregation procedure* is a functional Γ_ℓ of the elements of the dictionary $\mathbb{D}$ such that Γ_ℓ is still a predictor.

Selection

$$\Gamma_\ell(\hat{r}_{k,1}, \ldots, \hat{r}_{k,M}) = \hat{r}_{k,j} \quad \text{for some} \quad j \in \{1, \ldots, M\}.$$

Selection

$$\Gamma_\ell(\hat{r}_{k,1},\ldots,\hat{r}_{k,M}) = \hat{r}_{k,j} \quad \text{for some} \quad j \in \{1,\ldots,M\}.$$

Example: for some $\lambda > 0$,

$$\Gamma_\ell(\hat{r}_{k,1},\ldots,\hat{r}_{k,M}) = \arg\inf_{f\in\mathbb{D}} \{R_\ell(f) + \mathrm{pen}_\lambda(f)\}.$$

Convex aggregation

$$\Gamma_\ell(\hat{r}_{k,1},\ldots,\hat{r}_{k,M}) = \sum_{j=1}^{M} \omega_{\ell,j}\hat{r}_{k,j},$$

such that $\omega_{\ell,1},\ldots,\omega_{\ell,M} \geq 0$ and $\sum_{j=1}^{M} \omega_{\ell,j} = 1$.

Convex aggregation

$$\Gamma_\ell(\hat{r}_{k,1}, \ldots, \hat{r}_{k,M}) = \sum_{j=1}^{M} \omega_{\ell,j} \hat{r}_{k,j},$$

such that $\omega_{\ell,1}, \ldots, \omega_{\ell,M} \geq 0$ and $\sum_{j=1}^{M} \omega_{\ell,j} = 1$.

Examples: Uniform weights ($\omega_{\ell,j} = 1/M$), Exponentially Weighted Aggregation (EWA).

EWA

For some (inverse temperature parameter) $\lambda > 0$,

$$\Gamma_{\ell,\lambda}^{\mathrm{ewa}} = \sum_{j=1}^{M} \frac{\exp[-\lambda R_\ell(\hat{r}_{k,j})]}{\sum_{i=1}^{M} \exp[-\lambda R_\ell(\hat{r}_{k,i})]} \hat{r}_{k,j}.$$

EWA

For some (inverse temperature parameter) $\lambda > 0$,

$$\Gamma_{\ell,\lambda}^{\mathrm{ewa}} = \sum_{j=1}^{M} \frac{\exp[-\lambda R_\ell(\hat{r}_{k,j})]}{\sum_{i=1}^{M} \exp[-\lambda R_\ell(\hat{r}_{k,i})]} \hat{r}_{k,j}.$$

$\lambda \to 0$:

EWA

For some (inverse temperature parameter) $\lambda > 0$,

$$\Gamma_{\ell,\lambda}^{\mathrm{ewa}} = \sum_{j=1}^{M} \frac{\exp[-\lambda R_\ell(\hat{r}_{k,j})]}{\sum_{i=1}^{M} \exp[-\lambda R_\ell(\hat{r}_{k,i})]} \hat{r}_{k,j}.$$

$\lambda \to 0$: uniform weights.

EWA

For some (inverse temperature parameter) $\lambda > 0$,

$$\Gamma_{\ell,\lambda}^{\mathrm{ewa}} = \sum_{j=1}^{M} \frac{\exp[-\lambda R_\ell(\hat{r}_{k,j})]}{\sum_{i=1}^{M} \exp[-\lambda R_\ell(\hat{r}_{k,i})]} \hat{r}_{k,j}.$$

$\lambda \to 0$: uniform weights.
$\lambda \to \infty$:

EWA

For some (inverse temperature parameter) $\lambda > 0$,

$$\Gamma_{\ell,\lambda}^{\mathrm{ewa}} = \sum_{j=1}^{M} \frac{\exp[-\lambda R_\ell(\hat{r}_{k,j})]}{\sum_{i=1}^{M} \exp[-\lambda R_\ell(\hat{r}_{k,i})]} \hat{r}_{k,j}.$$

$\lambda \to 0$: uniform weights.
$\lambda \to \infty$: ERM.

EWA is supported by sharp oracle inequalities of the flavor

$$\mathbb{E}[\Gamma_\ell^{\mathrm{ewa}}(\mathbf{X}) - Y]^2 \leq \min_{j=1,\ldots,M} \mathbb{E}[\hat{r}_{k,j}(\mathbf{X}) - Y]^2 + \frac{\log(M)}{\lambda n}.$$

Linear aggregation

$$\Gamma_\ell(\hat{r}_{k,1},\ldots,\hat{r}_{k,M}) = \sum_{j=1}^{M} \omega_{\ell,j}\hat{r}_{k,j}.$$

Linear aggregation

$$\Gamma_\ell(\hat{r}_{k,1}, \ldots, \hat{r}_{k,M}) = \sum_{j=1}^{M} \omega_{\ell,j} \hat{r}_{k,j}.$$

Example: PAC-Bayesian Aggregation.

Conclusion

Take-home messages

- Sound mathematical foundations for statistical learning

Take-home messages

- Sound mathematical foundations for statistical learning
- Abundance of methods to deal with prediction: parametric, semiparametric, nonparametric...

Take-home messages

- Sound mathematical foundations for statistical learning
- Abundance of methods to deal with prediction: parametric, semiparametric, nonparametric... and meta-methods: aggregation
- Theoretical goal: tight risk upper bounds (oracle inequalities)

Take-home messages

- Sound mathematical foundations for statistical learning
- Abundance of methods to deal with prediction: parametric, semiparametric, nonparametric... and meta-methods: aggregation
- Theoretical goal: tight risk upper bounds (oracle inequalities)
- Algorithmical goal: tractable methods which scale up to modern (massive and complex) data

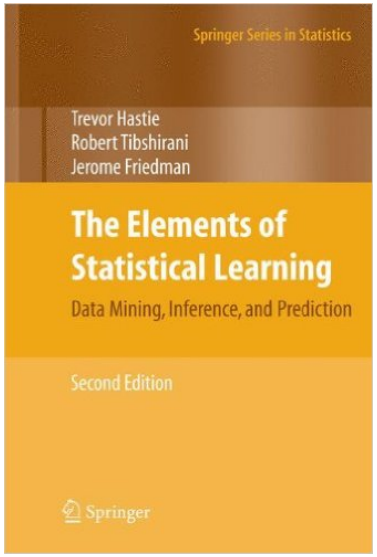
Take-home messages

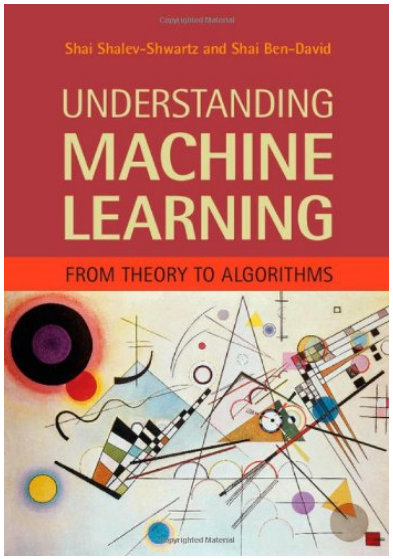
- Sound mathematical foundations for statistical learning
- Abundance of methods to deal with prediction: parametric, semiparametric, nonparametric... and meta-methods: aggregation
- Theoretical goal: tight risk upper bounds (oracle inequalities)
- Algorithmical goal: tractable methods which scale up to modern (massive and complex) data
- A very exciting field to work in!

References

T. Hastie, R. Tibshirani and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, Springer, 2009.

S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*, Cambridge University Press, 2014.





Benjamin Guedj, Ph.D.

https://bguedj.github.io
Inria Lille - Nord Europe