

# Introduction générale au machine learning

L'IA, un outil de compétitivité pour les entreprises

---

Formation DIRECCTE

Aurélien Garivier

# Contenu de la séance

Nous allons voir

- en quoi consiste l'approche " machine learning" pour la résolution de problèmes
- quels sont les grands types de problème résolubles par apprentissage
- quel type de données les algorithmes d'apprentissage savent traiter, et comment parfois s'y ramener
- quels sont les grandes familles d'algorithmes d'apprentissage et quels outils permettent de les mettre en œuvre

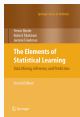
Nous n'allons pas voir :

- l'ensemble des techniques de ML
- la théorie du ML et les problèmes de recherche les plus récents
- comment utiliser des GPU et autres architectures massivement parallèles

# Bibliographie - Ressources



Pattern Classification (2001) - Wiley Interscience, *R. Duda, P. Hart, D. Stork*



The Elements of Statistical Learning (2001) - Springer, *T. Hastie, R. Tibshirani, J. Friedman*

Disponible en ligne : <http://www-stat.stanford.edu/~tibs/ElemStatLearn/>



Data Mining - Technip, *S. Tufféry*



Cours en ligne de Andrew Ng (Stanford):  
<https://www.coursera.org/course/ml>



<http://wikistat.fr/>

dont sont issus certains de ces slides !



<http://scikit-learn.org>



Base de données de benchmarking:

<http://archive.ics.uci.edu/ml/>

Machine Learning: when Artificial Intelligence meets Big Data

The Learning Models

Machine Learning Data and Methodology

Machine Learning Algorithms: a flavor

- Unsupervised learning

- Supervised learning

# Artificial Intelligence (AI): Definition

## Intelligence exhibited by machines

- emulate cognitive capabilities of humans  
(big data: humans learn from abundant and diverse sources of data).
- a machine mimics "cognitive" functions that humans associate with other human minds, such as "learning" and "problem solving".

## **Ideal "intelligent" machine =**

flexible rational agent that perceives its environment and takes actions that maximize its chance of success at some goal.

## **Founded on the claim that human intelligence**

"can be so precisely described that a machine can be made to simulate it."

## Operational goals

- Autonomous robots for not-too-specialized tasks
- In particular, vision + understand and produce language

## Tension between operational and philosophical goals

- As machines become increasingly capable, facilities once thought to require intelligence are removed from the definition. For example, optical character recognition is no longer perceived as an exemplar of "artificial intelligence"; having become a routine technology.
- Capabilities still classified as AI include advanced Chess and Go systems and self-driving cars.

# Machine Learning (ML): Definition

## **Arthur Samuel (1959)**

Field of study that gives computers the ability to learn without being explicitly programmed

## **Tom M. Mitchell (1997)**

A computer program is said to learn from experience  $E$  with respect to some class of tasks  $T$  and performance measure  $P$  if its performance at tasks in  $T$ , as measured by  $P$ , improves with experience  $E$ .

# ML: Learn from and make predictions on data

- Algorithms operate by building a model from **example** inputs in order to make data-driven **predictions or decisions**...
- ...rather than following strictly static program instructions: useful when designing and programming explicit algorithms is unfeasible or poorly efficient.

## Within Data Analytics

- Machine Learning used to devise complex models and algorithms that lend themselves to **prediction** - in commercial use, this is known as *predictive analytics*.
- [www.sas.com](http://www.sas.com): "Produce reliable, repeatable decisions and results" and uncover "hidden insights" through learning from historical **relationships and trends** in the data.
- evolved from the study of pattern recognition and computational learning theory in artificial intelligence.



# Machine Learning: Typical Problems

- spam filtering, text classification
- optical character recognition (OCR)
- search engines
- recommendation platforms
- speech recognition software
- computer vision
- bio-informatics, DNA analysis, medicine
- etc.

For each of this task, it is possible but very inefficient to write an explicit program reaching the prescribed goal.

It proves much more succesful to have a machine infer what the good decision rules are.

## Related Fields

- **Computational Statistics**: focuses in prediction-making through the use of computers together with statistical models (ex: Bayesian methods).
- **Statistical Learning**: ML by statistical methods, with statistical point of view (probabilistic guarantees: consistency, oracle inequalities, minimax)  
→ more focused on *correlation*, less on *causality*
- **Data Mining** (unsupervised learning) focuses more on exploratory data analysis: discovery of (previously) unknown properties in the data. This is the analysis step of Knowledge Discovery in Databases.
- Importance of **probability**- and **statistics**-based methods → **Data Science** (Michael Jordan)
- Strong ties to **Mathematical Optimization**, which delivers methods, theory and application domains to the field

# What is Data?

Data (here): digital content used for learning.

- images (objects, satellite, hand-written text, hyperspectral, etc.)
- times series (finance, earthquakes activity, ECG, etc.)
- internet traces (marketing, social network)
- texts
- audio (signal)
- behaviour: game, robot activity
- ...

# Qu'est-ce qu'une (très grande) masse de données ?



VLDB  
 XLDB  
 Massive Data  
 Data Masses  
 Data Deluge  
 Very Big Data  
 Big Data

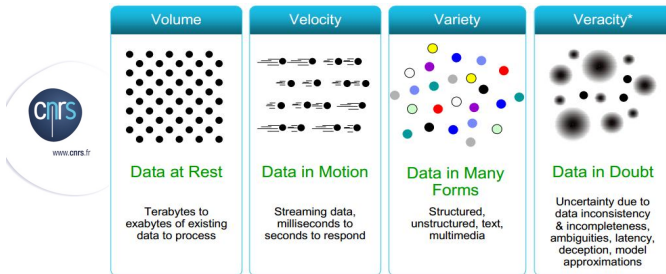
| Data inflation |                            |   |
|----------------|----------------------------|---|
| Unit           | Size                       | What it means   |
| Bit (b)        | 1 or 0                     | Short for "binary digit", after the binary code (1 or 0) computers use to store and process data                              |
| Byte (B)       | 8 bits                     | Enough information to create an English letter or number in computer code. It is the basic unit of computing                  |
| Kilobyte (KB)  | 1,000, or $2^{10}$ , bytes | From "thousand" in Greek. One page of typed text is 2KB   |
| Megabyte (MB)  | 1,000KB; $2^{20}$ bytes    | From "large" in Greek. The complete works of Shakespeare total 5MB. A typical pop song is about 4MB                           |
| Gigabyte (GB)  | 1,000MB; $2^{30}$ bytes    | From "giant" in Greek. A two-hour film can be compressed into 1-2GB   |
| Terabyte (TB)  | 1,000GB; $2^{40}$ bytes    | From "monster" in Greek. All the catalogued books in America's Library of Congress total 15TB                                 |
| Petabyte (PB)  | 1,000TB; $2^{50}$ bytes    | All letters delivered by America's postal service this year will amount to around 5PB. Google processes around 1PB every hour |
| Exabyte (EB)   | 1,000PB; $2^{60}$ bytes    | Equivalent to 10 billion copies of <i>The Economist</i>   |
| Zettabyte (ZB) | 1,000EB; $2^{70}$ bytes    | The total amount of information in existence this year is forecast to be around 1.2ZB   |
| Yottabyte (YB) | 1,000ZB; $2^{80}$ bytes    | Currently too big to imagine  |

The prefixes are set by an international group, the International Bureau of Weights and Measures. Yotta and Zetta were added in 1991; terms for larger amounts have yet to be established.

Source: The Economist

Grandes Conf du domaine: VLDB, XLDB, ICDE, EDBT, ...

# Complexité multidimensionnelle des Big Data



• Nouvelles archi. de stockage

• Nouvelles archi. d'interopérabilité

• Défi pour les réseaux de communication

• Nouveaux modèles de calcul sur des flux

• Nettoyage et transformation

• Fusion de données

Nouveaux modèles de qualité (données & processus de traitement)

<http://www.datasciencecentral.com/profiles/blogs/data-verity>

# Défis accompagnant les chgts

## DÉFIS TRANSVERSES

- Passage à l'échelle
- Rapidité traitements
- Protection, sécurité
- Interaction

Acquisition

Extraction, nettoyage

Intégration

Analyse

Interpretation

Stockage

Accès/  
Requêtage,  
Raisonnement

Valeur

Véracité

Velocité

Variété

Volume

*repenser les outils algorithmiques et mathématiques*

*inspired by "Big Data and Its Technical Challenges, Communications of the ACM, July 2014, vol 57, n°7", © H.V. Jagadish et al.*

4

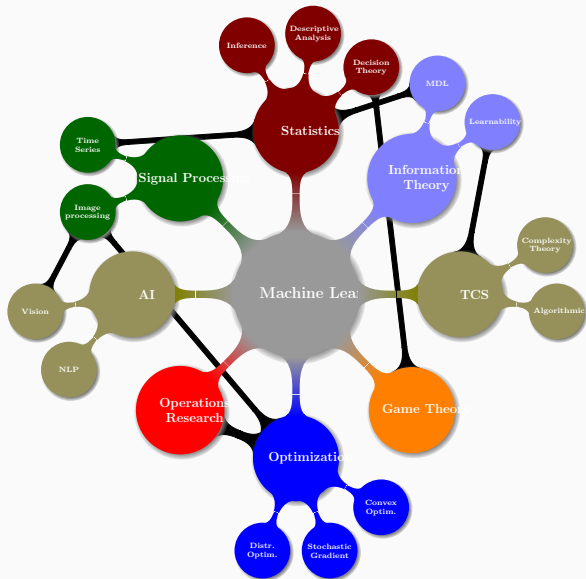
- Data analysis (inference, description) is the goal of statistics for long.
- Machine Learning has more **operational** goals (ex: consistency is important the statistics literature, but often makes little sense in ML).

Models (if any) are *instrumental*

Ex: linear model (nice mathematical theory) vs Random Forests.

- Machine Learning/big data: no separation between statistical modelling and optimization (in contrast to the statistics tradition).
- In ML, data is often here before (unfortunately)
- No clear separation (statistics evolves as well).

# ML and its neighbors





Machine Learning: when Artificial Intelligence meets Big Data

The Learning Models

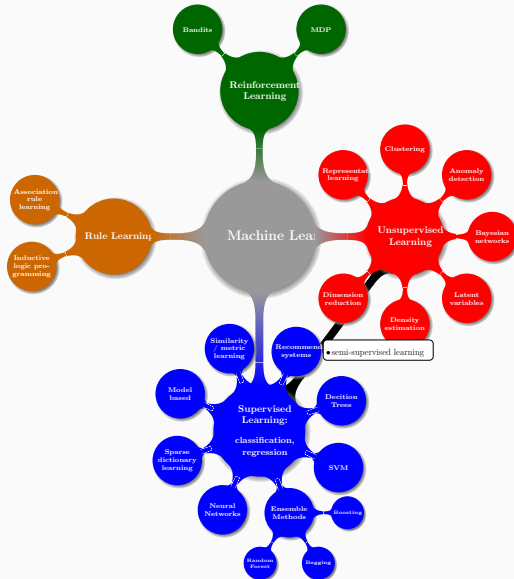
Machine Learning Data and Methodology

Machine Learning Algorithms: a flavor

- Unsupervised learning

- Supervised learning

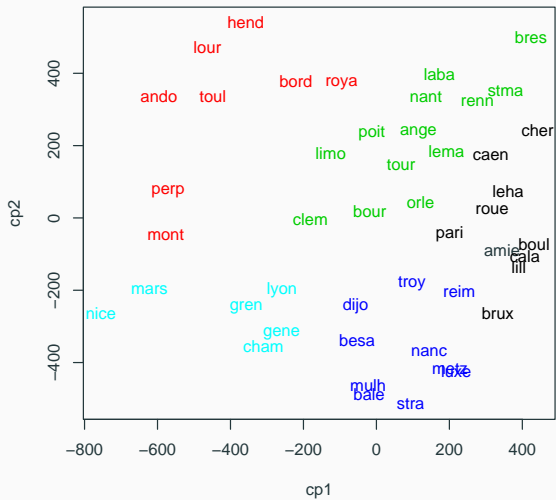
# What ML is composed of



# Unsupervised Learning

- (many) observations on (many) individuals
- need to have a simplified, structured overview of the data
- *taxonomy*: untargeted search for *homogeneous clusters* emerging from the data
- Examples:
  - customer segmentation
  - image analysis (recognizing different zones)
  - exploration of data

# Example

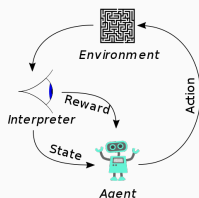
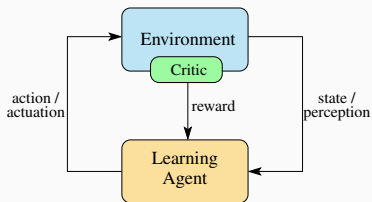


- observations = pairs  $(X_i, Y_i)$
- goal = learn to *predict*  $Y_i$  given  $X_i$
- regression (when  $Y$  is continuous)
- classification (when  $Y$  is discrete)
- statistical technique: linear models

## Example: Character Recognition

|   |                       |
|---|-----------------------|
| Input space $\mathcal{X}$   | $64 \times 64$ images |
| Output space $\mathcal{Y}$  | $\{0, 1, \dots, 9\}$  |
| Joint distribution $P(x, y)$  | ?                     |
| Prediction function $h \in \mathcal{H}$   |                       |
| Risk $R(h) = P(h(X) \neq Y)$  |                       |
| Sample $\{(x_i, y_i)\}_{i=1}^n$   | MNIST dataset         |
| Empirical risk<br>$\hat{R}_n(h) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{h(x_i) \neq y_i\}$                                   |                       |
| Learning algorithm<br>$\phi_n : (\mathcal{X} \times \mathcal{Y})^n \rightarrow \mathcal{H}$                                 | NN, boosting...       |
| Expected risk $R_n(\phi) = \mathbb{E}_n[R(\phi_n)]$   |                       |
| Empirical risk minimizer<br>$\hat{h}_n = \operatorname{argmin}_{h \in \mathcal{H}} \hat{R}_n(h)$                            |                       |
| Regularized empirical risk minimizer<br>$\hat{h}_n = \operatorname{argmin}_{h \in \mathcal{H}} \hat{R}_n(h) + \lambda C(h)$ |                       |

# Reinforcement Learning



[Src: [https://en.wikipedia.org/wiki/Reinforcement\\_learning](https://en.wikipedia.org/wiki/Reinforcement_learning)]

- area of machine learning inspired by behaviourist psychology
- how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward.
- Model: random system (typically : Markov Decision Process)
  - agent
  - state
  - actions
  - rewards
- sometimes called approximate dynamic programming, or neuro-dynamic programming

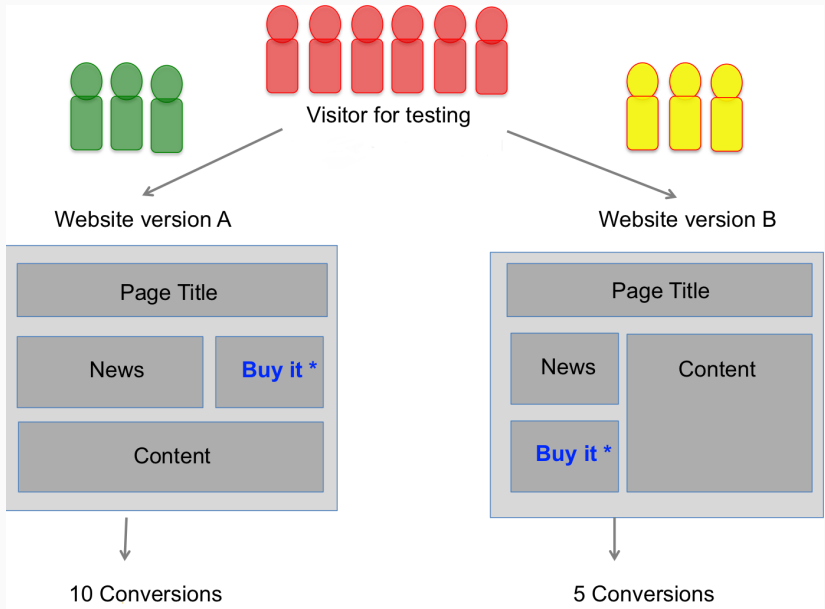
## Example: the Retail Store Management Problem

At each month  $t$ , a store contains  $x_t$  items of a specific goods and the demand for that goods is  $D_t$ . At the end of each month the manager of the store can order  $a_t$  more items from his supplier. Furthermore we know that:

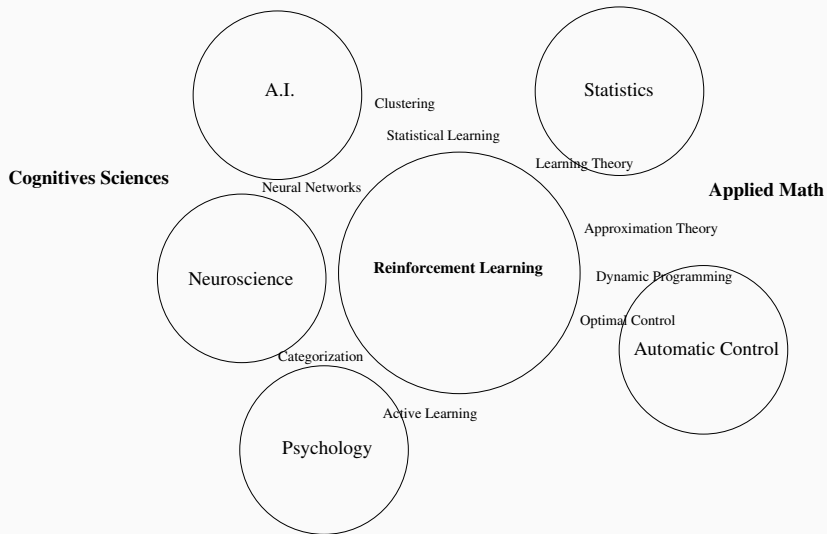
- The **cost** of maintaining an inventory of  $x$  is  $h(x)$ .
- The **cost** to order  $a$  items is  $C(a)$ .
- The **income** for selling  $q$  items is  $f(q)$ .
- If the demand  $D$  is bigger than the available inventory  $x$ , customers that cannot be served leave.
- The **value of the remaining inventory** at the end of the year is  $g(x)$ .
- **Constraint**: the store has a maximum capacity  $M$ .



# Example: A/B testing



# Reinforcement Learning and the others



Machine Learning: when Artificial Intelligence meets Big Data

The Learning Models

Machine Learning Data and Methodology

Machine Learning Algorithms: a flavor

- Unsupervised learning

- Supervised learning

$n$ -by- $p$  matrix  $X$

- $n$  examples = points of observations
- $p$  features = characteristics measured for each example

Questions to consider:

- Are the features centered?
- Are the features normalized? bounded?

In `scikitlearn`, all methods expect a 2D array of shape  $(n, p)$  often called

`X (n_samples, n_features)`

- Inside R: package datasets
- Inside scikitlearn: package sklearn.datasets
- UCI Machine Learning Repository
- Challenges: Kaggle, etc.



# The big steps of data analysis

1. Extracting the data to expected format
2. Exploring the data
  - detection of outliers, of inconsistencies
  - descriptive exploration of the distributions, of correlations
  - data transformations
3. Random partitioning of the data: (see also: cross-validation)
  - learning sample
  - validation sample
  - test sample
4. For each algorithm: parameter estimation using training and validation samples
5. Choice of final algorithm using testing sample, risk estimation

# Machine Learning tools: R

The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains R code for a regression analysis. The code defines a model with various predictors (TT, RR, F, lines, slopes) and uses the `lm()` function to fit the model. It also includes a loop to calculate the R-squared value for different strategies.
- Environment/History:** Shows the objects created in the environment, including `d`, `d$UCB`, `d$CTC`, `d$CTUG`, `Fb`, `dayData`, and `UCB`.
- Console:** Displays the output of the R code, showing the results of the `lm()` function and the R-squared values for different strategies.
- Plots:** A line plot showing the R-squared values for different strategies across a range of values (50 to 5000). The plot includes a legend with the following entries:
  - FB-ETC: 3.65
  - BAI-ETC: 2.98
  - UCB: 1.69
  - SPRT-ETC: 1.03
  - D-UCB: 0.77

# Machine Learning tools: python

The image shows a Spyder Python IDE window with a script for survival analysis. The script defines a Kaplan-Meier estimator and a function to estimate the NA (Nelson Aalen) estimate. A plot titled 'Figure 2' shows the NA estimate over time (smetime). The console output shows the execution of the script, including the definition of the Kaplan-Meier estimator and the calculation of the NA estimate.

```
1 # -*- coding: utf-8 -*-
2 Created on Tue Aug 16 09:05:03 2016
3
4 @author: agarvis
5
6
7
8
9
10
11
12
13
14 def kaplan_meier(tz, Delta, censorDate):
15     k = 1/(Delta)
16     atrisk = np.zeros(N) # atrisk before Delta[]
17     survived = np.zeros(N) # dmed between Delta[] and Delta[]+1
18     for i in k:
19         i = 0
20         while j < N and x[i] + Delta[] < censorDate and x[i] + Delta[] < x[i+1]:
21             atrisk[j] += 1
22             if x[i] + Delta[] < x[i+1]:
23                 survived[j] += 1
24             j = j + 1
25     S = np.concatenate([[], np.cumprod(survived/atrisk)])
26     return(S)
27
28 # atrisk: index shift of 1 - see kaplan_meier_2 which conforms to package
29
30 def plotKaplanMeier():
31     plotKaplanMeier()
32
33 N = 10000
34 x = [p.censval[1][k], 1][k] for k in range(N)
35 now = 2
36 x = [[], x][1][k] + now * (2 - x[k]) for k in x
37
38 N = 40
39 Delta = np.array([float(i)/N for i in range(N+1)])
40 print(S)
41
42 # plt.step(Delta, S)
43 # plt.hold('on')
44 # plt.plot(Delta, 1 - Delta)
45 # plt.hold('off')
46
47 # plt.axis([0, 0.5, 1, 0.1])
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Figure 2: A plot showing the NA estimate (Nelson Aalen estimate) over time (smetime). The x-axis ranges from 0.0 to 0.6, and the y-axis ranges from 0.0 to 1.4. The plot shows a blue line representing the NA estimate, which increases over time and levels off around 1.0. The plot is titled 'Figure 2' and has a legend indicating 'NA estimate'.

Console Output:

```
Python 2.7.10 [Anaconda 3.3.0 (64-bit)] (default, Sep 15 2015, 14:50:01)
Type "copyright", "credits" or "license()" for more information.

Python 4.0.0 -- an enhanced Interactive Python.
> Introduction and overview of IPython's features.
?> quickref -> quick reference.
help -> Python's own help system.
object? -> details about "object", use "object???" for extra details.
kgfref -> a brief reference about the graphical user interface.

In [1]: matplotlib._home/Agarvis/ownCloud/prog/python/HC/payment_renewal_study_ayasmoothazard.py,
wdir=/home/Agarvis/ownCloud/prog/python/HC
Traceback (most recent call last):
  File "c:\python\ipynb-1-863c8d4892b", line 1, in <module>
    matplotlib._home/Agarvis/ownCloud/prog/python/HC/payment_renewal_study_ayasmoothazard.py,
wdir=/home/Agarvis/ownCloud/prog/python/HC
  File "/home/Agarvis/anaconda/lib/python2.7/site-
packages/numpy/lib/_idutils_extmodule1/xisctomsize.py", line 685, in nutils
    _execfile(filenam, namespace)
  File "/home/Agarvis/anaconda/lib/python2.7/site-
packages/numpy/lib/_idutils_extmodule1/xisctomsize.py", line 78, in _execfile
    builtin._execfile(filenam, namespace)
  File "/home/Agarvis/ownCloud/prog/python/HC/payment_renewal_study_ayasmoothazard.py", line 68,
in <module>
    (t, S) = hist(S)
NameError: name 'ka' is not defined

In [2]: matplotlib._home/Agarvis/ownCloud/prog/python/HC/deprecated/kaplan_meier.py,
wdir=/home/Agarvis/ownCloud/prog/python/HC/deprecated
Loaded module: survival
0.8536348 0.82865707 0.8699236 0.78493786 0.737225 0.7216366
0.7087517 0.6632844 0.651111 0.6480397 0.6124787 0.59106173
0.56849225 0.5494156 0.5132682 0.4871414 0.4642252 0.4380984
nan nan nan nan nan nan
nan nan nan nan nan nan

In [3]:
```





The screenshot shows the scikit-learn website homepage. At the top, there is a navigation bar with links for Home, Installation, Documentation, and Examples. A search bar is also present. The main header features the scikit-learn logo and the tagline "Machine Learning in Python". Below this, a grid of small images illustrates various machine learning concepts. A list of key features is provided: simple and efficient tools for data mining and data analysis, accessibility to everybody, built on NumPy, SciPy, and matplotlib, and being open source under a BSD license. The page is organized into several sections: Classification, Regression, Clustering, Dimensionality reduction, Model selection, and Preprocessing. Each section includes a brief description, applications, and algorithms. At the bottom, there are sections for News, Community, and Who uses scikit-learn?, along with the AWeber logo.

scikit-learn  
Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

### Classification

Identifying to which category an object belongs to.

**Applications:** Spam detection, Image recognition.

**Algorithms:** SVM, nearest neighbors, random forest, ... — Examples

### Regression

Predicting a continuous-valued attribute associated with an object.

**Applications:** Drug response, Stock prices.

**Algorithms:** SVR, ridge regression, Lasso, ... — Examples

### Clustering

Automatic grouping of similar objects into sets.

**Applications:** Customer segmentation, Grouping experiment outcomes

**Algorithms:** k-Means, spectral clustering, mean-shift, ... — Examples

### Dimensionality reduction

Reducing the number of random variables to consider.

**Applications:** Visualization, Increased efficiency

**Algorithms:** PCA, feature selection, non-negative matrix factorization, ... — Examples

### Model selection

Comparing, validating and choosing parameters and models.

**Goal:** Improved accuracy via parameter tuning

**Modules:** grid search, cross validation, metrics, ... — Examples

### Preprocessing

Feature extraction and normalization.

**Application:** Transforming input data such as text for use with machine learning algorithms.

**Modules:** preprocessing, feature extraction, ... — Examples

### News

**On-going development:** What's new (Changelog)

### Community

**About us** See authors and contributing  
**More Machine Learning** Find related

### Who uses scikit-learn?

AWeber COMMUNICATIONS

# Knime, Weka and co: integrated environments

The screenshot displays the Weka Explorer application window, which is divided into several functional areas:

- Classifier:** A dropdown menu is set to "J48 -C 0.25 -M 2".
- Test options:** Includes radio buttons for "Use training set" (selected), "Supplied test set", "Cross-validation", and "Percentage split". A "More options..." button is also present.
- Classifier output:** A text area showing the results of a stratified cross-validation. The output includes a summary table:

| === Stratified cross-validation === |       |      |
|-------------------------------------|-------|------|
| === Summary ===                     |       |      |
| Correctly Classified Instances      | 144   | 96 % |
| Incorrectly Classified Instances    | 6     | 4 %  |
| Kappa statistic                     | 0.94  |      |
| Mean absolute error                 | 0.035 |      |
| Root mean square                    |       |      |
- Tree View:** A separate window displays a decision tree structure. The root node is "petalwidth".
  - Left branch ( $\leq 0.6$ ): "Iris-setosa (50.0)".
  - Right branch ( $> 0.6$ ): "petalwidth".
    - Left sub-branch ( $\leq 1.7$ ): "petalength".
      - Left sub-sub-branch ( $\leq 4.9$ ): "Iris-versicolor (48.0/1.0)".
      - Right sub-sub-branch ( $> 4.9$ ): "petalwidth".
        - Left sub-sub-sub-branch ( $\leq 1.5$ ): "Iris-virginica (3.0)".
        - Right sub-sub-sub-branch ( $> 1.5$ ): "Iris-versicolor (3.0/1.0)".
    - Right sub-branch ( $> 1.7$ ): "Iris-virginica (46.0/1.0)".

- Visualization:** A scatter plot on the right shows data points colored by class. A vertical line is drawn at  $x = 6.9$ . Below the plot, the text "versicolor" and "Iris-virginica" is visible.

Machine Learning: when Artificial Intelligence meets Big Data

The Learning Models

Machine Learning Data and Methodology

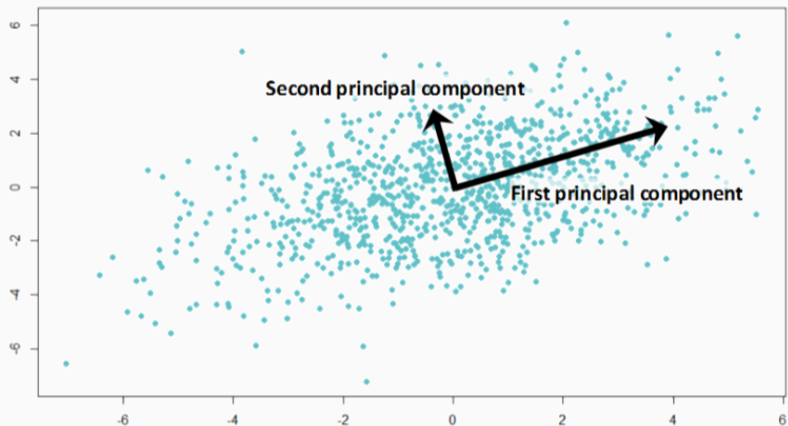
**Machine Learning Algorithms: a flavor**

Unsupervised learning

Supervised learning

- PCA
- K-means
- Hierarchical Methods
- Others:
  - Spectral clustering
  - t-SNE
  - ...

# PCA: visualization



Src: [<https://techannouncer.com/global-pca-unit-market-2017-adelte-airmak-industries-amss-ltd-cavotec-airport-division-ciat-effeti/>]

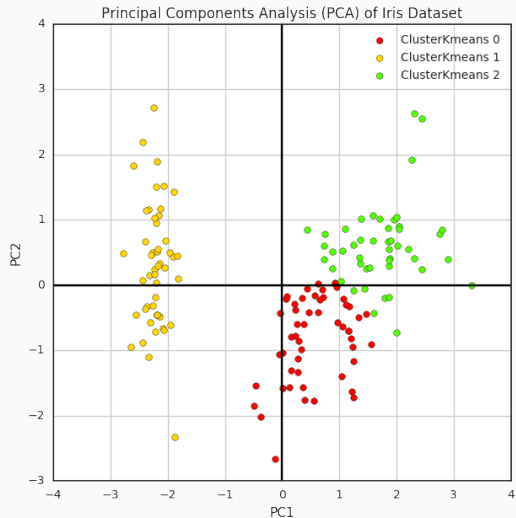
## PCA

- Center all variables
- Compute the  $p \times p$  empirical covariance matrix  $X^T X$ .
- Compute the components  $W_d =$  the  $d$  first eigenvectors of  $X^T X$  in decreasing order of the eigenvalues
- Return the projection of  $X$  onto the  $d$  first components  $T_d = X W_d$ .

Then:

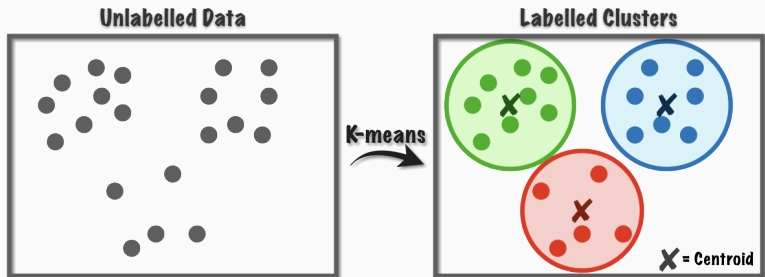
- either visualize clusters (2d or 3d plots)
- or use another clustering algorithm on the lower-dimensionnal data  $T_d$  (*dimension reduction*)

# Example: IRIS



[Src: <https://www.kaggle.com/bburns/iris-exploration-pca-k-means-and-gmm-clustering>]

# Model-free clustering: K-means



Src: [towardsdatascience.com](https://towardsdatascience.com)



## K-means target

- Observations  $X_1, \dots, X_n$  in  $\mathbb{R}^p$ ;
- Objective function: for candidate cluster centers  $\mu = (\mu_1, \dots, \mu_K)$  and cluster assignments  $z = (z_1, \dots, z_n)$ :

$$L(\mu, z) = \sum_{k=1}^K \sum_{i: z_i=k} \|X_i - \mu_k\|^2 = \sum_{i=1}^n \sum_{k=1}^K \mathbb{1}\{z_i = k\} \|X_i - \mu_k\|^2$$

- If  $S_k = \{i : z_i = k\}$ ,

$$L(\mu, z) = \sum_{k=1}^K |S_k| \text{Var}[S_k]$$

- Minimizing  $L$  is equivalent to minimizing pairwise deviations in the clusters:

$$\operatorname{argmin}_{\mu, z} L(\mu, z) = \operatorname{argmin}_{\mu, z} \sum_{k=1}^K \frac{1}{|S_k|} \sum_{i, j \in S_k} \|X_i - X_j\|^2$$

# Lloyd's algorithm

- For a fixed  $\mu$ , optimizing in  $z$  is easy: choose  $z_i = \operatorname{argmin}_k \|X_i - \mu_k\|$
- BUT optimizing in  $\mu$  is NP-hard!

## k-means

- randomly initialize  $\theta_0$
- compute Lloyd's iterations until convergence:
  - *membership variables*  $z_i^j = \operatorname{argmin}_k \|X_i - \mu_k^j\|$
  - *updated cluster weights*  $N_k^j = \sum_{i=1}^n \mathbb{1}\{z_i^j = k\}$
  - *updated cluster means*  $\mu_k^{j+1} = \frac{\sum_{i: z_i^j = k} X_i}{N_k^j}$
- start again (a few times) to look for a better local optimum

# (Agglomerative) Hierarchical Cluster Analysis

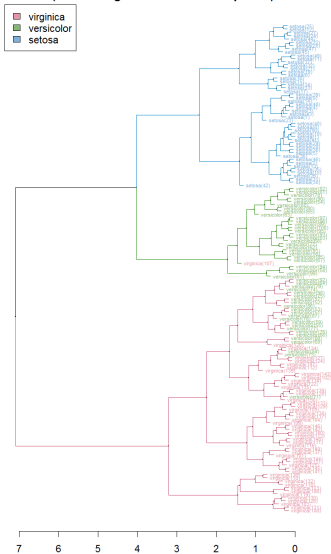
- greedy bottom-up algorithm
- requires a distance (dissimilarity) between observations  $\|x - x'\|$
- choice of *distance between clusters*:
  - complete linkage:  $d(A, B) = \max \{ \|x - x'\| : x \in A, x' \in B \}$
  - single linkage:  $d(A, B) = \min \{ \|x - x'\| : x \in A, x' \in B \}$
  - average linkage distance:  $d(A, B) = \frac{1}{|A||B|} \sum_{x \in A} \sum_{x' \in B} \|x - x'\|$
  - Ward distance for Euclidian mean:  $d(A, B) = \frac{|A||B|}{n(|A| + |B|)} \|\bar{A} - \bar{B}\|$
  - sum of intra-cluster variance
  - etc.

## HCA

- Initialization: all observations are clusters  $\{X_1\}, \dots, \{X_n\}$
- As long as there are at least two clusters:
  - add a link between two clusters with smallest distance
  - merge them for the next iterations
- Return the *dendrogram* = hierarchy of clusters

# HCA: Dendrogram

Clustered Iris data set  
(the labels give the true flower species)



Author: Talgalili

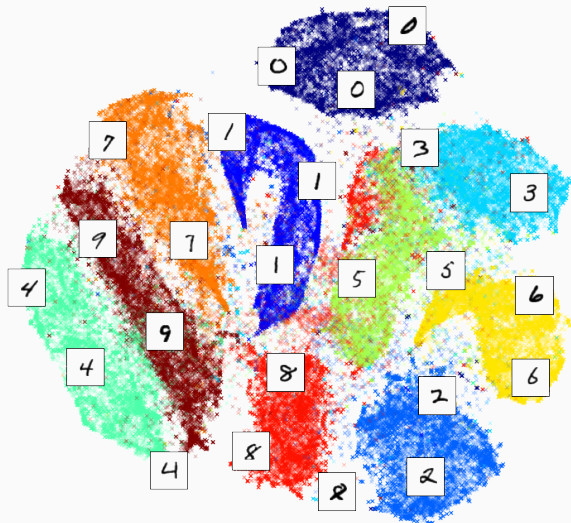
<https://commons.wikimedia.org/wiki/File:>

Iris\_dendrogram.png

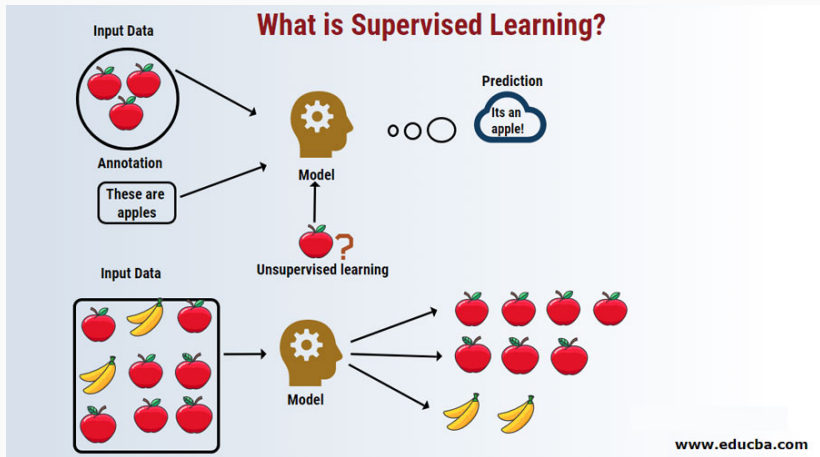
# t-SNE, and the others...

## MNIST dataset

Two-dimensional embedding of 70,000 handwritten digits with t-SNE



# Supervised learning



# Qualities of a supervised learning algorithm

- Efficiency (consistency / minimax)
- Tuning: parameter-free
- Result-critic
- Interpretability / Explicability
- Complexity
- Sequentiality
- Privacy



# La méthodes des $k$ plus proches voisins

Règle des  $k$  plus proches voisins : pour tout  $x \in \mathcal{X}$ , trouver ses plus proches voisins  $x_{(1)}, \dots, x_{(k)}$



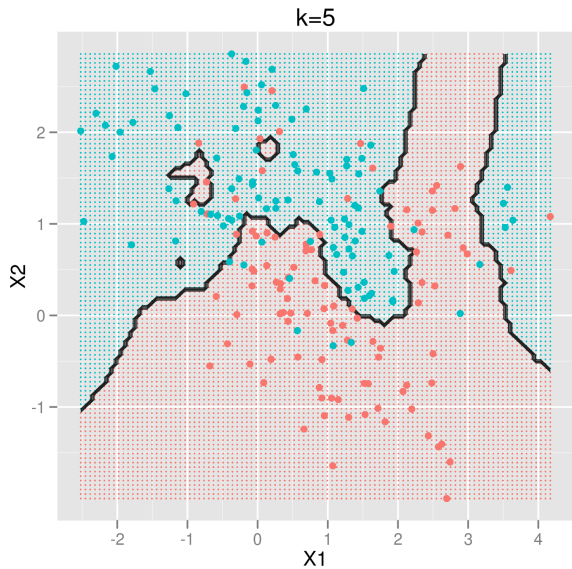
- classification:

$$f_n(x) = \operatorname{argmax}_{y \in \mathcal{Y}} \sum_{j=1}^k \mathbb{1}\{y_{(j)} = y\}$$

- régression:

$$f_n(x) = \frac{1}{k} \sum_{j=1}^k y_{(j)}$$

# Visualisation d'une règle k-NN



Qualités :

- simplicité
- interprétabilité (?)
- pas de consistance (quel que soit  $k$ )
- MAIS asymptotiquement erreur au plus 2x supérieure à la règle de Bayes.
- possibilité de faire croître  $k$  avec  $n$ , consistance (théorique) par exemple pour  $k = \log(n)$

Paramétrage:

- quelle métrique sur  $\mathcal{X}$  ?
- ⇒ au minimum, *normaliser* les variables (pb pour les qualitatives)
- Quelle valeur de  $k$  choisir ?

**Interprétabilité:** OUI et NON

**Critique:** OUI mais pas très fiable

**Consistance:** NON mais possible si  $k = \log(n)$  (par exemple)

**Minimax:** NON

**Parameter-free:** NON

**Vitesse:** OUI et NON, implémentation possible en  $O(n \log n)$

**Online:** OUI

# Arbres de décision

The screenshot displays the Weka Explorer interface. The 'Classify' tab is active, showing the 'Classifier' set to 'J48 -C 0.25 -M 2'. The 'Test options' section includes 'Use training set' selected, 'Cross-validation' with 10 folds, and 'Percentage split' at 66%. The 'Classifier output' window shows the following summary:

```
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances      144      96 %
Incorrectly Classified Instances    6         4 %
Kappa statistic                    0.94
Mean absolute error                 0.035
Root mean square error              0.035
```

The 'Tree View' window displays a decision tree structure:

```
graph TD
    Root((petalwidth)) -- "<= 0.6" --> Node1[Iris-setosa (50.0)]
    Root -- "> 0.6" --> Node2((petalwidth))
    Node2 -- "<= 1.7" --> Node3((petalength))
    Node2 -- "> 1.7" --> Node4[Iris-virginica (46.0/1.0)]
    Node3 -- "<= 4.9" --> Node5[Iris-versicolor (48.0/1.0)]
    Node3 -- "> 4.9" --> Node6((petalwidth))
    Node6 -- "<= 1.5" --> Node7[Iris-virginica (3.0)]
    Node6 -- "> 1.5" --> Node8[Iris-versicolor (3.0/1.0)]
```

The 'Visualize' window shows a scatter plot of the data points, with 'Y: petalwidth (Num)' selected. The plot includes a vertical line at 6.9 and a legend for 'Iris-versicolor' (red) and 'Iris-virginica' (green).

La segmentation par arbre est une approche non-paramétrique de l'analyse discriminante.

**But** : expliquer une variable réponse (qualitative ou quantitative) à l'aide d'autres variables.

**Principe** : construire un arbre à l'aide de divisions successives des individus d'un ensemble  $E$  en deux segments (appelés aussi noeuds) homogènes par rapport à une variable  $Y$  (binaire, nominale, ordinale ou quantitative) en utilisant l'information de  $p$  variables  $X^1, \dots, X^p$  (binaires, nominales, ordinales ou quantitatives).

L'arbre obtenu est sous forme d'un arbre inversé comportant à la racine l'échantillon total  $E$  à segmenter et les autres segments sont

- soit des segments intermédiaires (encore divisibles),
- soit des segments terminaux.

L'ensemble des segments terminaux constitue une partition de l'ensemble  $E$  en classes homogènes et distinctes, relativement à la variable  $Y$ .

Il s'agit d'**arbre de classement** si  $Y$  est qualitative et d'**arbre de régression** si  $Y$  est quantitative.

## Avantages / Inconvénients

- La méthode CART (**Classification And Regression Tree**) fournit des solutions sous formes graphiques simples à interpréter.
- Elle est complémentaire des méthodes statistiques classiques, très calculatoire et efficace à condition d'avoir de grandes tailles d'échantillon.
- Elle est capable de gérer à la fois les variables quantitatives ET qualitatives simultanément.
- Peu d'hypothèses requises !
- Algorithme étant basé sur une stratégie pas à pas hiérarchisée, il peut passer à côté d'un optimum global.



Soient  $p$  variables quantitatives ou qualitatives explicatives  $X^j$  et une variable à expliquer  $Y$  qualitative à  $m$  modalités  $\{\tau_l, l = 1, \dots, m\}$  ou quantitative réelle, observée sur un échantillon de  $n$  individus.

La construction d'un arbre de discrimination binaire consiste à déterminer une séquence de **noeuds**.

- Un noeud est défini par le choix conjoint d'une variable parmi les explicatives et d'une **division** qui induit une partition en deux classes.
- Une division est elle-même définie par une valeur seuil de la variable quantitative sélectionnée ou un partage en deux groupes des modalités si la variable est qualitative.
- À la racine ou au noeud initial correspond l'ensemble de l'échantillon. La procédure est ensuite itérée sur chacun des sous-ensembles.

## L'algorithme considéré nécessite

- 1 la définition d'un critère permettant de sélectionner la "meilleure" division parmi toutes celles **admissibles** pour les différentes variables ;
- 2 une règle permettant de décider qu'un noeud est terminal : il devient alors **feuille** ;
- 3 l'affectation de chaque feuille à l'une des classes ou à une valeur de la variable à expliquer.

Le point 2. correspond encore à la recherche d'un modèle parcimonieux. Un arbre trop détaillé, associé à une sur-paramétrisation, est instable et donc probablement plus défaillant pour la prévision d'autres observations.

Breiman et al. ont mise en place une stratégie de recherche de l'arbre optimal.

- 1 Construire l'arbre maximal  $A_{max}$ .
- 2 Ordonner les sous-arbres selon une séquence emboîtée suivant la décroissance d'un critère pénalisé de déviance ou de taux de mal-classés.
- 3 Sélectionner le sous-arbre optimal : c'est la procédure d'**élagage**.

**Objectif** : Rechercher le meilleur compromis entre

- un arbre tr s d taill , fortement d pendant des observations qui ont permis son estimation, qui fournira un mod le de pr vision tr s instable
- un arbre trop robuste mais grossier qui donne des pr dictions trop approximatives.

**Principe**

- Construire une suite embo t e de sous-arbres de l'arbre maximum par  lagage successif.
- Choisir, parmi cette suite, l'arbre optimal au sens d'un crit re.

La solution obtenue par algorithme pas   pas n'est pas n cessairement, globalement optimale mais l'efficacit  et la fiabilit  sont pr f r es   l'optimalit .

**Interprétabilité:** OUI !

**Critique:** OUI mais pas très précis

**Consistance:** OUI (sous certaines réserves) MAIS instable!

**Minimax:** NON !

**Parameter-free:** NON

**Vitesse:** OUI

**Online:** NON