

Machine Learning 1

Introduction, nearest neighbor classifier

Master 2 Computer Science

Aurélien Garivier

2019-2020



Table of contents

1. Before we start
2. What is Machine Learning?
3. The Learning Models
4. Machine Learning Methodology
5. Supervised Classification
6. Nearest-Neighbor Classification

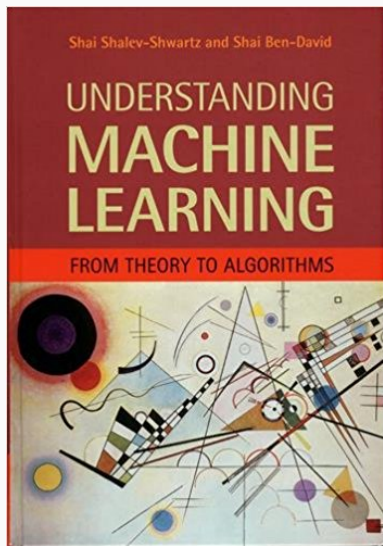
Before we start

Outline (1/2)

- 1. 09.09 Introduction, nearest-neighbor classification
- 2. 09.16 ML methodology, k-nearest neighbors, decision trees
- 3. 09.23 PAC Learning Theory, no-free-lunch theorem
- 4. 09.30 Dimensionality Reduction: PCA, random projections
- 5. 10.07 VC dimension, empirical risk minimization
- 6. 10.14 Linear separators, Support Vector Machines
- 7. 10.21 Kernels, regularization
 - 10.28 holidays
- 8. 11.4 Boosting, Bagging, Random Forests
 - 11.11 bank holiday

Outline (2/2)

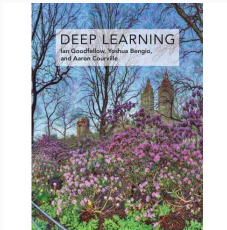
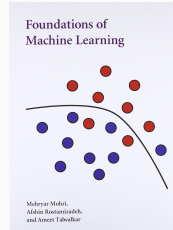
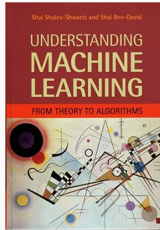
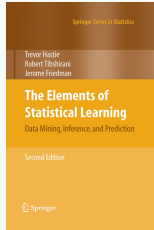
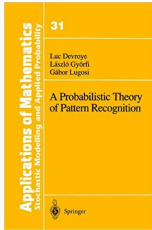
- 9. 11.18 Neural networks and stochastic gradient descent
- 10. 11.25 Regression, model selection
- 11. 12.02 Clustering
- 12. 12.09 Online Learning
- 13. 12.16 Reinforcement Learning
- 12.23 holidays
- 12.30 holidays
- 14. 01.06 Questions / Exercises
- 15. 01.13 Final Exam



General introduction to Machine Learning theory, by two leading researchers of the field.

Covers a good part of the content of this course (other references will be provided for specific topics).

Additional References



Homework and in-class exercises

and

- analysis and review of a research article
 - report + oral presentation
 - articles will be proposed along the lectures
- or participation in a ML student challenge:
 - topic: anomaly detection
 - data: Airbus sensors
 - teams: 4 participants
 - start: October 10th
 - see <https://defi-ia.insa-toulouse.fr/>

(you choose)

What is Machine Learning?

Why Machine Learning?

Actualité

Yann LeCun, Geoffrey Hinton et Yoshua Bengio reçoivent le prix Turing

Par [Stephane Nachez](#) - 27 mars 2019



LE MACHINE LEARNING PROVOQUE UNE CRISE DANS LE DOMAINE DE LA SCIENCE

[Bastien L.](#) · 19 février 2019 · [Analytics, Data Analytics, Intelligence artificielle](#) · 1 commentaire

Le Machine Learning est en train de provoquer une grave crise de reproductibilité dans le domaine de la science. C'est ce qu'affirme la statisticienne Genevera Allen de la Rice University dans le cadre de la conférence AAAS Annual Meeting.

De plus en plus de chercheurs utilisent le **Machine Learning** pour analyser des données et y détecter des tendances. Cependant, dans le cadre de la conférence scientifique AAAS Annual Meeting, la statisticienne Genevera Allen de la Rice University a tenu à tirer la sonnette d'alarme. Selon elle, le Machine Learning est en passe de provoquer **une crise de reproductibilité dans le domaine de la science.**

SHARE SPECIAL VIEWPOINTS



Machine Learning for Science: State of the Art and Future Prospects

Eric Mjølhus, Dennis DeCoste
* See all authors and affiliations

Science 14 Sep 2019
Vol. 365, Issue 6402, pp. 2051-2055
DOI: 10.1126/science.1263.0037.2051

Article Figures & Data Info & Metrics eLetters PDF

Abstract

Recent advances in machine learning methods, along with successful applications across a wide variety of fields such as planetary science and bioinformatics, promise powerful new tools for practicing scientists. This viewpoint highlights some useful characteristics of modern machine learning methods and their relevance to scientific applications. We conclude with some speculations on near-term progress and promising directions.

PUBLIC RELEASE: 15-FEB-2019

Can we trust scientific discoveries made using machine learning?

Rice U. expert: Key is creating ML systems that question their own predictions

RICE UNIVERSITY

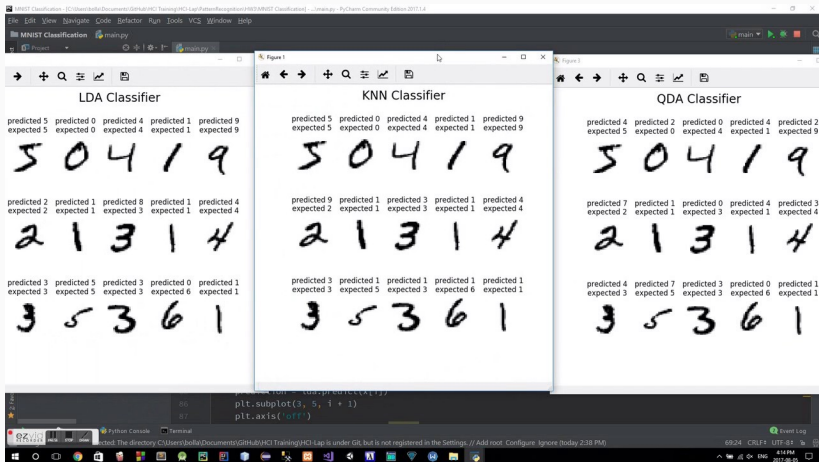
What is Machine Learning?

- 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 Artificial Intelligence

- evolved from the study of pattern recognition and computational learning theory in artificial intelligence.
- AI: 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".

Example: MNIST dataset



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 .

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.

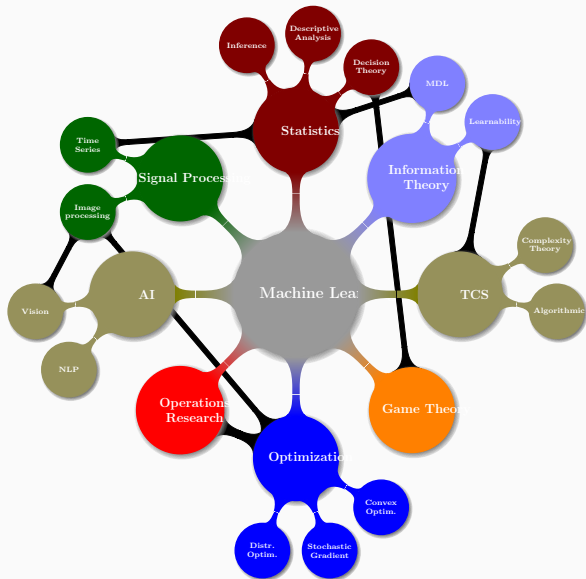
For each of this task, it is possible but very inefficient to write an explicit program reaching the prescribed goal.

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

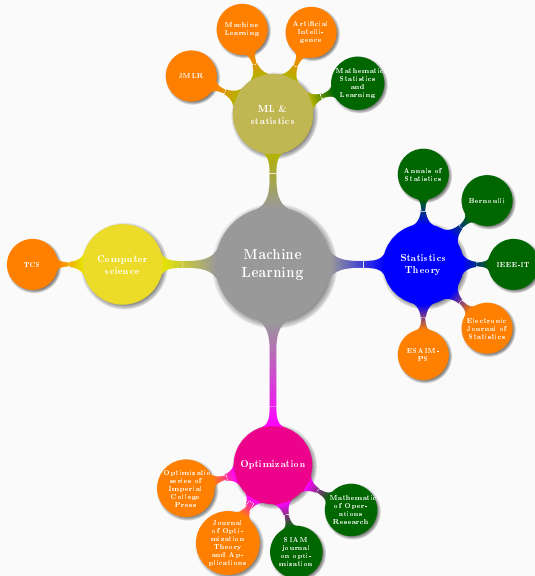
What is Statistical Learning?

- = Machine Learning using statistics-inspired tools and guarantees
- Importance of **probability**- and **statistics**-based methods
→ **Data Science** (Michael Jordan)
 - **Computational Statistics**: focuses in prediction-making through the use of computers together with statistical models (ex: Bayesian methods).
 - **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.
 - Machine Learning has more **operational** goals
Ex: ~~consistency~~ → oracle inequalities
Models (if any) are *instrumental*.
ML more focused on *correlation*, less on *causality* (now changing).
 - Strong ties to **Mathematical Optimization**, which furnishes methods, theory and application domains to the field

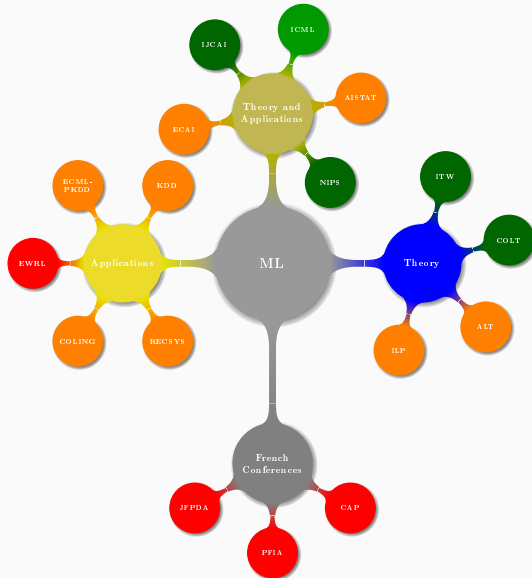
ML and its neighbors



ML journals

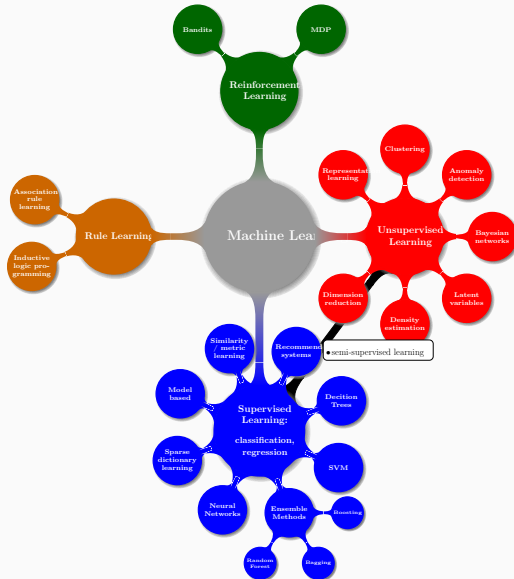


ML conferences



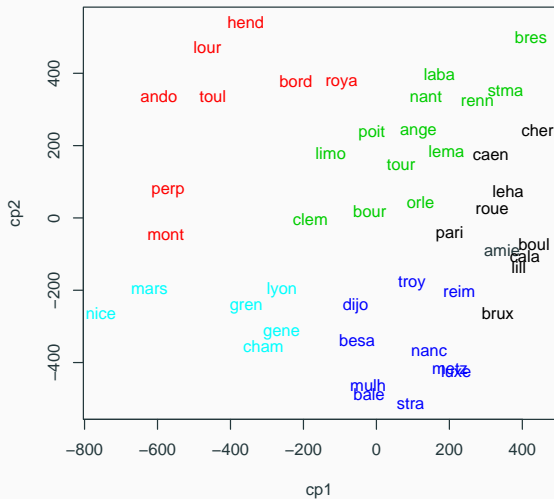
The Learning Models

What ML is composed of



- (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: representing the climate of cities



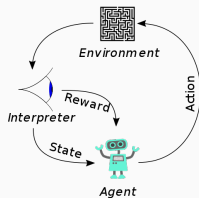
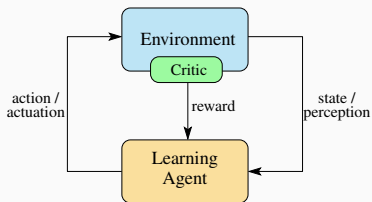
Supervised Learning

- 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)

Examples:

- Spam filtering / text categorization
- Image recognition
- Credit risk ranking

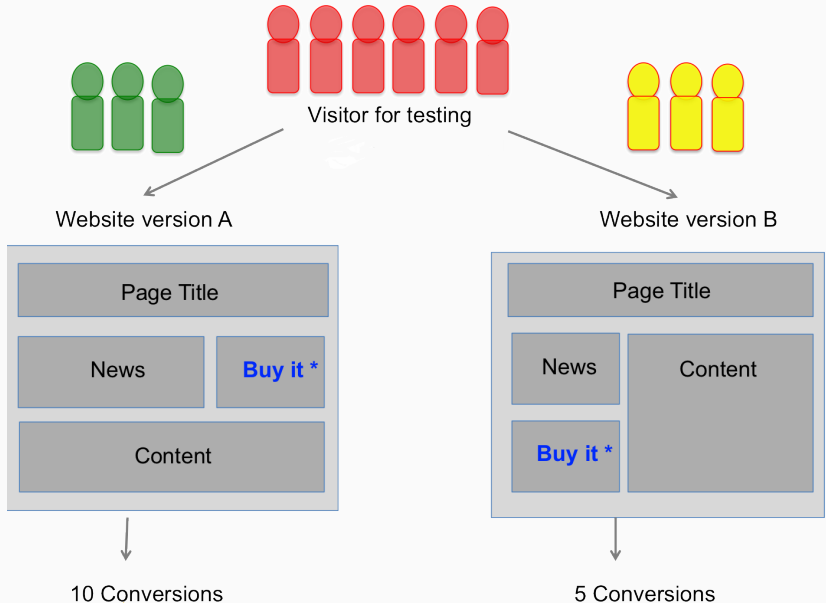
Reinforcement Learning



[Src: 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: A/B testing



Machine Learning Methodology

m -by- p matrix X

- m 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 (m, 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

 - learning sample
 - validation sample
 - test sample
3. For each algorithm: parameter estimation using training and validation samples
4. Choice of final algorithm using testing sample, risk estimation

Machine Learning tools: R

The screenshot displays the RStudio interface with a script editor on the left containing R code for a regression analysis. The code defines a function `createData` that generates data for different strategies (FB-ETC, BAI-ETC, UCB, SPRT-ETC, D-UCB) across various time points. The main script then calls `createData` and uses `lme4` to fit mixed-effects models for each strategy. The results are stored in a list `names` and printed to the console.

```
## Source on Save
264 plot(TT, RR[i,], type="l", lwd=4, lty=lty[i], log="x", xlim=c(min(TT), max(TT)), ylim = c(0, max(RR[
265 #arrow(TT, RR[i,]*2^SD[i,]/sqrt(Nmc), TT, RR[i,]*2^SD[i,]/sqrt(Nmc), length=0.1, lwd=4, col=3, log=
266 for (j in 1:nStrategies){
267   lines(TT, RR[i,], lwd=4, lty=lty[j])
268   f <- lme4::MCMCglmm(TT=10000) ~ log(TT*(1-10000))
269   lines(TT[TT<500], rCoefficients[1] + rCoefficients[2]*log(TT[TT<500]), col="red", lty=lty[j])
270   slopes[j] <- rCoefficients[2]^4
271   #arrow(TT, RR[i,]*2^SD[i,]/sqrt(Nmc), TT, RR[i,]*2^SD[i,]/sqrt(Nmc), length=0.1, lwd=4, col=2, l
272 }
273 #for (h in c(0.5, 1, 2, 4)){ lines(TT, h*log(TT^h/2^4)/d, col="red", lwd=2) }
274
275 #plot(TT, #RR[i,]/log(TT), type="l", lwd=4, lty=lty[i], log="x", xlim=c(min(TT), max(TT)), ylim = c
276 #arrow(TT, RR[i,]*2^SD[i,]/sqrt(Nmc), TT, RR[i,]*2^SD[i,]/sqrt(Nmc), length=0.1, lwd=4, col=3, log=
277 #for (j in 2:nStrategies){
278   # lines(TT, #RR[i,]/log(TT), lwd=4, lty=lty[j])
279   #arrow(TT, RR[i,]*2^SD[i,]/sqrt(Nmc), TT, RR[i,]*2^SD[i,]/sqrt(Nmc), length=0.1, lwd=4, col=3, l
280 #}
281
282 names <- c("FB-ETC", "SPRT-ETC", "BAI-ETC", "D-UCB", "UCB") #???-UCB" "UCB")
283 order <- c(1,3,5,2,4)
284 legend(50, 80, slope$order, function(k) paste(names[k], "\n", round(slopes[k], 2))), lty = lty$orde
285
286
287 # T <- c(2.5, 10, 20, 30, 50, 100, 150, 200, 300, 400, 500, 600, 800, 1000, 1500, 2000, 3000, 5000, 8000, 10000)/d^2; Nmc
288 # T <- c(2.5, 10, 20, 30, 50, 100, 150, 200, 300, 400, 500, 600, 800, 1000)/d^2; Nmc <- 10000; createData(T, Nmc);
289 # T <- c(10, 20, 20^2); Nmc <- 1000; createData(T, Nmc);
290 # T <- c(400, 800, 1500)/d^2; Nmc <- 10000; createData(T, Nmc);
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
```

The console output shows the following results:

```
[[1] 0.9119 0.00303 1.00000 0.00039 0.00618
[[1] 10000
[[1] 72.72462 34.65975 53.25412 27.32834 28.02640
[[1] 0.00027 0.50208 1.00000 0.00023 0.00425
[[1] 12500
[[1] 77.44936 36.14412 56.56396 28.23064 49.41559
[[1] 0.00719 0.00177 1.00000 0.00021 0.00361
[[1] 15000
[[1] 80.50309 36.92527 59.43862 29.01284 41.89409
[[1] 0.00548 0.00148 1.00000 0.00013 0.00272
[[1] 20000
[[1] 85.12948 38.00491 63.51994 29.99955 44.16748
[[1] 0.00435 0.00107 1.00000 0.00016 0.00210
[[1] 25000
[[1] 89.93376 39.54227 66.87458 30.88489 45.81386
[[1] 0.00370 0.00083 1.00000 0.00012 0.00166
[[1] 37500
[[1] 96.90037 42.01518 72.95729 32.54084 49.32925
[[1] 0.00236 0.00050 1.00000 0.00008 0.00127
[[1] 50000
[[1] 102.77196 42.75523 77.29644 33.54500 51.36300
[[1] 0.00177 0.00030 1.00000 0.00008 0.00080
```

The plot shows the results of the regression analysis. The x-axis represents time (T) on a logarithmic scale from 50 to 50000. The y-axis represents the value of the function from 0 to 100. Five lines are plotted, representing different strategies: FB-ETC (3.65), BAI-ETC (2.98), UCB (1.69), SPRT-ETC (1.03), and D-UCB (0.77). The lines show that the value of the function increases over time, and the rate of increase is highest for FB-ETC and lowest for D-UCB.

Machine Learning tools: python

The image shows a Python IDE (Spyder) with a script for Kaplan-Meier survival analysis. The script defines a function `kaplan_meier` that takes `Delta` and `censorDate` as input. It calculates the survival function `S` using the Kaplan-Meier estimator. The script also includes a plot of the NA estimate.

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue Aug 16 09:05:03 2016
4
5 @author: agarvis
6 """
7
8 import numpy as np
9 from random import random
10 import matplotlib as mpl
11 import matplotlib.pyplot as plt
12
13
14 def kaplan_meier(Delta, censorDate):
15     k = len(Delta)
16     atrisk = np.zeros(k) # atrisk before Delta[i]
17     survived = np.zeros(k) # dicit between Delta[i] and Delta[i+1]
18     for i in k:
19         s = 0
20         while j < k and x[i] + Delta[i] < censorDate and x[i] + Delta[i] < x[i+1]:
21             atrisk[j] += 1
22             if x[i] + Delta[i] < x[i+1]:
23                 survived[j] += 1
24             j = j + 1
25     S = np.concatenate([1], np.cumprod(survived/atrisk))
26     return(S)
27
28 # atrisk: index where of 3? see kaplan_meier_2 which conforms to package
29
30 # plot: class "all"
31 # plot: class "all"
32
33 N = 10000
34 x = [p.causal[random(), random()] for k in range(N)]
35 now = 2
36 x = [True, x[i]+(2-True)] for z 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.subplot(Delta, 0)
43 # plt.hold('on')
44 # plt.plot(Delta, 1-Delta)
45 # plt.hold('off')
46
47 # plt.subplot(0.05, 1, 8, 1)
48 # plt.plot(S)
49
50
51 def kaplan(T, E):
52     I = np.argsort(T)
53     T = [T[i] for k in I]
54     e = [E[i] for k in I]
55     n = len(I)
56     S = np.ones(n)
57     H = np.zeros(n)
58     for k in range(n-1):
59         S[k+1] = S[k] * (n-k-e[k])/(n-k)
60         H[k+1] = H[k] + e[k]/(n-k+1)
```

The plot, titled "Figure 2", shows the NA estimate (Survival Function) on the y-axis (ranging from 0.0 to 1.4) against `smetime` on the x-axis (ranging from 0.0 to 0.6). The plot displays a blue line representing the survival function, which starts at (0, 1) and decreases as `smetime` increases, showing a step-like pattern characteristic of the Kaplan-Meier estimator.

The console output shows the following data:

```
[ 0]  nantile /home/agarvis/ownCloud/prog/python/HC/deprecated/kaplan_meier.py,
1 loaded module: survive
2 0.8536348 0.82865707 0.8069256 0.7849376 0.772275 0.7216366
3 0.7087517 0.6632844 0.6511511 0.6480397 0.6124787 0.59106173
4 0.5684225 0.5494156 0.5132682 0.4871614 0.462252 0.4380984
5 nan nan nan nan nan
6 0.41398202 0.39895707 0.36949202 0.3493209 nan nan
7 nan nan nan nan nan]
```



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. The 'Classify' tab is active, showing the 'Classifier' dropdown set to 'J48 -C 0.25 -M 2'. Under 'Test options', 'Use training set' is selected. The 'Classifier output' pane 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
```

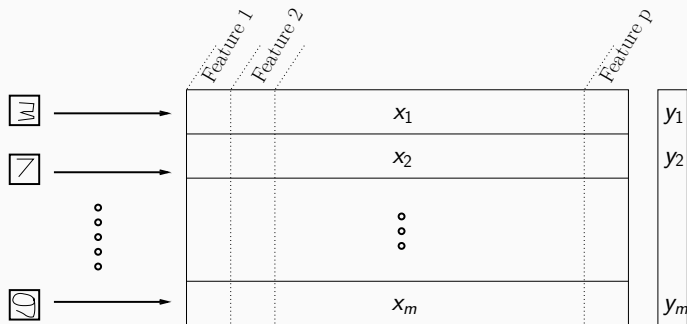
The 'Tree View' window shows a decision tree structure:

```
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 main interface also includes a 'Result list' on the left, a 'Start' button, and a 'Visualize' section on the right with a scatter plot and a 'Jitter' slider.

Supervised Classification

What is a classifier?



$$X \in \mathcal{M}_{m,p}(\mathbb{R})$$

$$Y \in \mathcal{Y}^m$$

Data: m -by- p matrix X

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

Classifier \mathcal{A}_m

$$\begin{array}{c} \downarrow \\ h_m : \mathcal{X} \rightarrow \mathcal{Y} \\ \boxed{6} \mapsto 6 \end{array}$$

Assumption

- The examples $(X_i, Y_i)_{1 \leq i \leq m}$ are iid samples of an unknown joint distribution \mathcal{D} ;
- The points to classify later are also independent draws of the *same* distribution \mathcal{D} .

Hence, for every *decision rule* $h : \mathcal{X} \rightarrow \mathcal{Y}$ we can define the *risk*

$$L_{\mathcal{D}}(h) = \mathbb{P}_{(X,Y) \sim \mathcal{D}}(h(X) \neq Y) = \mathcal{D}\left(\{(x,y) : h(x) \neq y\}\right).$$

The goal of the learning algorithm is to *minimize the expected risk*:

$$R_m(\mathcal{A}_m) = \mathbb{E}_{\mathcal{D}^{\otimes m}} \left[L_{\mathcal{D}} \left(\underbrace{\mathcal{A}_m((X_1, Y_1), \dots, (X_m, Y_m))}_{\hat{h}_m} \right) \right]$$

for every distribution \mathcal{D} , using only the examples.

Realizable case vs agnostic learning

One usually distinguishes

- the *realizable case*: there exists $h : \mathcal{X} \rightarrow \mathcal{Y}$ such that $\mathbb{P}_{(X,Y) \sim \mathcal{D}}(h(X) = Y) = 1$,
- and the *agnostic case* otherwise (x does not permit to predict y with certainty).

Examples:

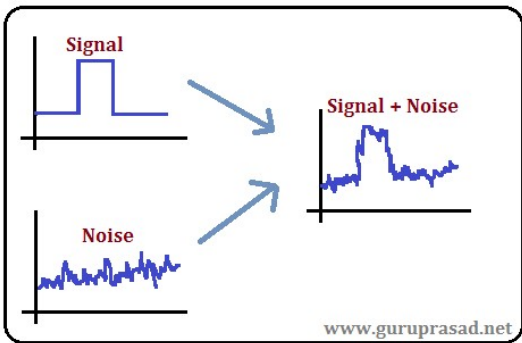
- spam filtering, character recognition
- credit risk, heart disease prediction

We generally focus on the agnostic case.

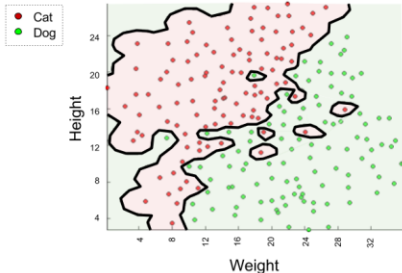
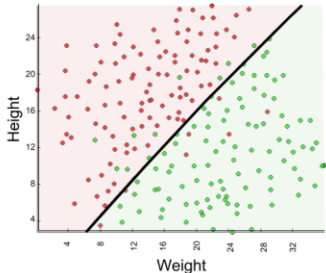
Signal and Noise

new york times bestseller
noise and the noi
the signal and th
and the noise and
the noise and th
why so many noi
predictions fail—
but some don't th
and the noise and
nate silver the nc

"Could turn out to be one of the more momentous books of the decade." —*The New York Times Book Review*



www.guruprasad.net



- Domain set \mathcal{X}
- Label set \mathcal{Y}
- Statistical Model: $\{D \text{ probability over } \mathcal{X} \times \mathcal{Y}\}$
- Training data: pairs $(X_i, Y_i) \in \mathcal{X} \times \mathcal{Y}$, $1 \leq i \leq m$
m = sample size
- Learner's output: $\hat{h}_m : \mathcal{X} \rightarrow \mathcal{Y}$. Possibly $\hat{h}_m \in \mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$.
- Measures of success: risk measure of hypothesis $h \in \mathcal{H}$

$$L_{\mathcal{D}}(h) = \mathbb{P}_{(X, Y) \sim \mathcal{D}}(h(X) \neq Y) = D\left(\{(x, y) : h(x) \neq y\}\right).$$

Example: Character Recognition

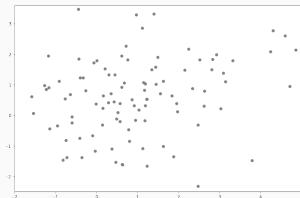
Domain set \mathcal{X} Label set \mathcal{Y} Joint distribution \mathcal{D}	64×64 images $\{0, 1, \dots, 9\}$?
Prediction function $h \in \mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ Risk $R(h) = P_{\mathcal{X}, \mathcal{Y}}(h(X) \neq Y)$	
Sample $S = \{(x_i, y_i)\}_{i=1}^m$ Empirical risk $L_S(h) = \frac{1}{m} \sum_{i=1}^m \mathbb{1}\{h(x_i) \neq y_i\}$	MNIST dataset
Learning algorithm $\mathcal{A} = (\mathcal{A}_m)_m, \mathcal{A}_m : (\mathcal{X} \times \mathcal{Y})^m \rightarrow \mathcal{H}$ Expected risk $R_m(\mathcal{A}) = \mathbb{E}_m[L(\mathcal{A}_m(S_m))]$	neural nets, boosting...
Empirical risk minimizer $\hat{h}_m = \arg \min_{h \in \mathcal{H}} L_S(h)$ Regularized empirical risk minimizer $\hat{h}_m = \arg \min_{h \in \mathcal{H}} L_S(h) + \lambda C(h)$	

Statistical Learning

One can have 2 visions of D :

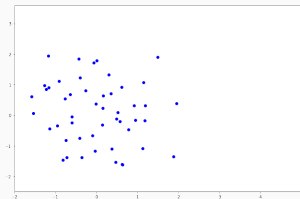
As a pair (D_x, k) , where

- for $A \subset \mathcal{X}$, $D_x(A) = D(A \times \mathcal{Y})$ is the marginal distribution of X ,
- and for $x \in \mathcal{X}$ and $B \subset \mathcal{Y}$,
 $k(B|x) = \mathbb{P}(Y \in B|X = x)$ is (a version of) the conditional distribution of Y given X .



As a pair $(D_y, (D(\cdot|y))_y)$, where

- for $y \in \mathcal{Y}$, $D_y(y) = D(\mathcal{X} \times y)$ is the marginal distribution of Y ,
- and for $A \subset \mathcal{X}$ and $y \in \mathcal{Y}$,
 $D(A|y) = \mathbb{P}(X \in A|Y = y)$ is the conditional distribution of X given $Y = y$.



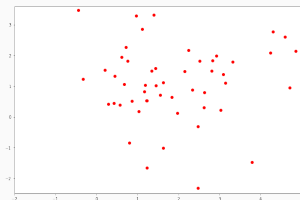
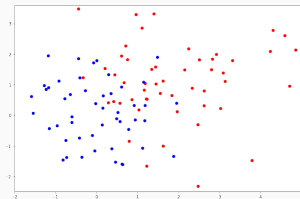
One can have 2 visions of D :

As a pair (D_x, k) , where

- for $A \subset \mathcal{X}$, $D_x(A) = D(A \times \mathcal{Y})$ is the marginal distribution of X ,
- and for $x \in \mathcal{X}$ and $B \subset \mathcal{Y}$,
 $k(B|x) = \mathbb{P}(Y \in B|X = x)$ is (a version of) the conditional distribution of Y given X .

As a pair $(D_y, (D(\cdot|y))_y)$, where

- for $y \in \mathcal{Y}$, $D_y(y) = D(\mathcal{X} \times y)$ is the marginal distribution of Y ,
- and for $A \subset \mathcal{X}$ and $y \in \mathcal{Y}$,
 $D(A|y) = \mathbb{P}(X \in A|Y = y)$ is the conditional distribution of X given $Y = y$.



Bayes Classifier

Consider binary classification $\mathcal{Y} = \{0, 1\}$, let $\eta(x) = \mathbb{P}(Y = 1|X = x)$.

Theorem

The Bayes classifier is defined by

$$h^*(x) = \mathbb{1}\{\eta(x) \geq 1/2\} = \mathbb{1}\{\eta(x) \geq 1 - \eta(x)\} = \mathbb{1}\{2\eta(x) - 1 \geq 0\}.$$

For every classifier $h : \mathcal{X} \rightarrow \mathcal{Y} = \{0, 1\}$,

$$L_{\mathcal{D}}(h) \geq L_{\mathcal{D}}(h^*) = \mathbb{E}\left[\min(\eta(X), 1 - \eta(X))\right].$$

The Bayes risk $L_{\mathcal{D}}^ = L_{\mathcal{D}}(h^*)$ is called the **noise** of the problem.*

More precisely,

$$L_{\mathcal{D}}(h) - L_{\mathcal{D}}(h^*) = \mathbb{E}\left[|2\eta(X) - 1| \mathbb{1}\{h(X) \neq h^*(X)\}\right].$$

Extends to $|\mathcal{Y}| > 2$.

$$\begin{aligned}
L_D(h) - L_D(h^*) &= \mathbb{E} \left[\mathbb{1}\{h(X) \neq h^*(X)\} \left(\right. \right. \\
&\quad \mathbb{1}\{Y = 1\} \left(\mathbb{1}\{h^*(X) = 1\} - \mathbb{1}\{h^*(X) = 0\} \right) \\
&\quad \left. \left. + \mathbb{1}\{Y = 0\} \left(\mathbb{1}\{h^*(X) = 0\} - \mathbb{1}\{h^*(X) = 1\} \right) \right) \right] \\
&= \mathbb{E} \left[\mathbb{1}\{h(X) \neq h^*(X)\} \left(2\mathbb{1}\{Y = 1\} - 1 \right) \left(2\mathbb{1}\{h^*(X) = 1\} - 1 \right) \right] \\
&= \mathbb{E} \left[\mathbb{1}\{h(X) \neq h^*(X)\} \left(2\mathbb{1}\{Y = 1\} - 1 \right) \left(2\mathbb{1}\{\eta(X) \geq \frac{1}{2}\} - 1 \right) \right] \\
&= \mathbb{E} \left[\mathbb{1}\{h(X) \neq h^*(X)\} \left(2\mathbb{1}\{\eta(X) \geq \frac{1}{2}\} - 1 \right) \mathbb{E} \left[2\mathbb{1}\{Y = 1\} - 1 \mid X \right] \right] \\
&= \mathbb{E} \left[\mathbb{1}\{h(X) \neq h^*(X)\} \left(2\mathbb{1}\{\eta(X) \geq \frac{1}{2}\} - 1 \right) \left(2\mathbb{E}[\mathbb{1}\{Y = 1\} \mid X] - 1 \right) \right] \\
&= \mathbb{E} \left[\mathbb{1}\{h(X) \neq h^*(X)\} \operatorname{sign} \left(\eta(X) - \frac{1}{2} \right) (2\eta(X) - 1) \right] \\
&= \mathbb{E} \left[\mathbb{1}\{h(X) \neq h^*(X)\} |2\eta(X) - 1| \right]
\end{aligned}$$

Nearest-Neighbor Classification

The Nearest-Neighbor Classifier

We assume that \mathcal{X} is a metric space with distance d .

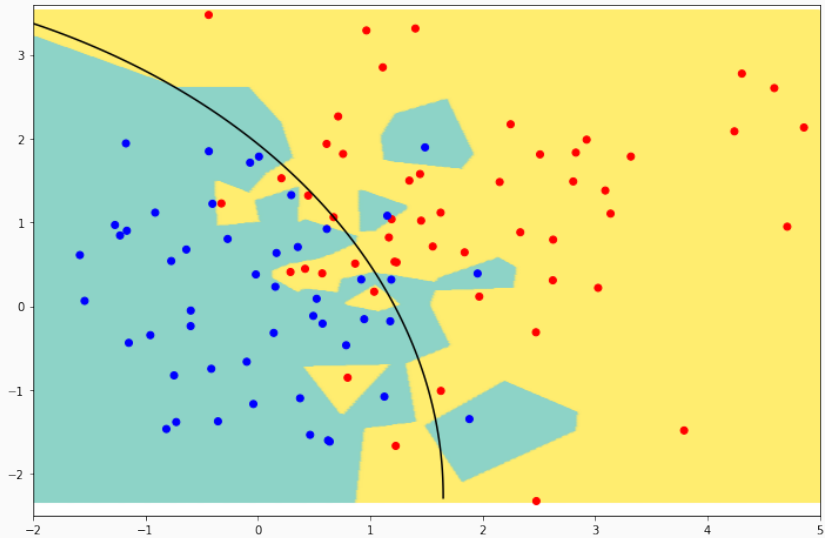
The nearest-neighbor classifier $\hat{h}_m^{NN} : \mathcal{X} \rightarrow \mathcal{Y}$ is defined as

$$\hat{h}_m^{NN}(x) = Y_I \text{ where } I \in \arg \min_{1 \leq i \leq m} d(x - X_i).$$

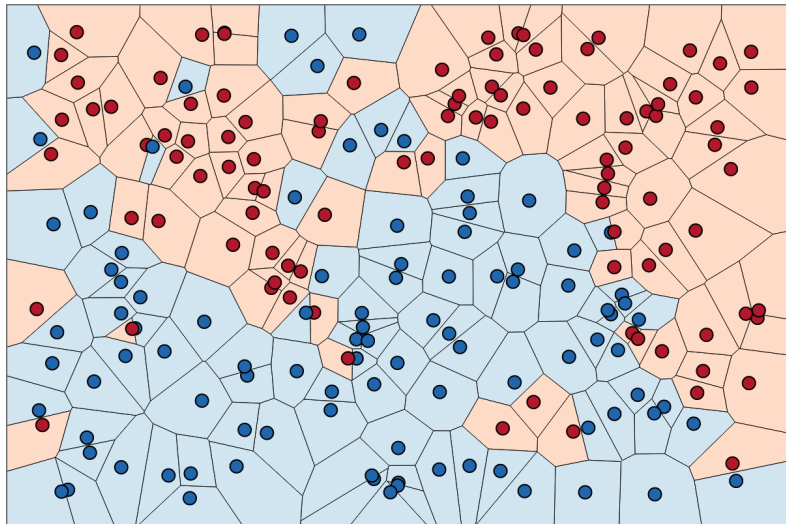
Typical distance: L^2 norm on \mathbb{R}^d $\|x - x'\| = \sqrt{\sum_{j=1}^d (x_j - x'_j)^2}$.

Buts many other possibilities: Hamming distance on $\{0, 1\}^d$, etc.

Numerically



Numerically



- A1.** $\mathcal{Y} = \{0, 1\}$.
- A2.** $\mathcal{X} = [0, 1]^d$.
- A3.** η is c -Lipschitz continuous:

$$\forall x, x' \in \mathcal{X}, |\eta(x) - \eta(x')| \leq c \|x - x'\| .$$

Theorem

Under the previous assumptions, for all distributions D and all $m \geq 1$

$$R_m(\hat{h}_m^{NN}) \leq 2L_D^* + \frac{3c\sqrt{d}}{m^{1/(d+1)}}$$

Proof Outline

- Conditioning: as $I(x) = \arg \min_{1 \leq i \leq m} \|x - X_i\|$,

$$R_m(\hat{h}_m^{NN}) = \mathbb{E} \left[\mathbb{E} \left[\mathbb{1}\{Y \neq Y_{I(X)}\} \mid X, X_1, \dots, X_m \right] \right].$$

- $Y \sim \mathcal{B}(p)$, $Y' \sim \mathcal{B}(q) \implies \mathbb{P}(Y \neq Y') \leq 2 \min(p, 1-p) + |p - q|$,

$$\mathbb{E} \left[\mathbb{1}\{Y \neq Y_{I(X)}\} \mid X, X_1, \dots, X_m \right] \leq 2 \min(\eta(X), 1 - \eta(X)) + c \|X - X_{I(X)}\|.$$

- Partition \mathcal{X} into $|\mathcal{C}| = T^d$ cells of diameter \sqrt{d}/T :

$$\mathcal{C} = \left\{ \left[\frac{j_1 - 1}{T}, \frac{j_1}{T} \right] \times \dots \times \left[\frac{j_d - 1}{T}, \frac{j_d}{T} \right], \quad 1 \leq j_1, \dots, j_d \leq T \right\}.$$

- 2 cases: either the cell of X is occupied by a sample point, or not:

$$\|X - X_{I(X)}\| \leq \sum_{c \in \mathcal{C}} \mathbb{1}\{X \in c\} \left(\frac{\sqrt{d}}{T} \mathbb{1} \bigcup_{i=1}^m \{X_i \in c\} + \sqrt{d} \mathbb{1} \bigcap_{i=1}^m \{X_i \notin c\} \right).$$

- $\implies \mathbb{E}[\|X - X_{I(X)}\|] \leq \frac{\sqrt{d}}{T} + \frac{\sqrt{d}T^d}{em}$ and choose $T = \lfloor m^{\frac{1}{d+1}} \rfloor$.

What does the analysis say?

- Is it loose? (sanity check: uniform \mathcal{D}_X)
- Non-asymptotic (finite sample bound)
- The second term $\frac{3c\sqrt{d}}{m^{1/(d+1)}}$ is distribution independent
- Does not give the trajectorial decrease of risk
- Exponential bound d (cannot be avoided...)
 \implies *curse of dimensionality*

- How to improve the classifier?