Lect6: Advanced methods

Tokyo Tech.
Intro. to Comp. & Data
Lecture week6

Intro. to some advanced machine learning algorithms and methods.

1. SVM: support vector machine
   (slides borrowed from some lectures)

2. Boosting

3. NCD, normalized compression distance:
   an eccentric (but sometimes useful) method
   for measuring "distance" among strings

4. On homework #6

1. Support Vector Machine
1.1. Basic concept

Support Vector Machine (in short, SVM) is a method for achieving a classification task. [Vapnik etal. 1963, 1992, 1995]
It has the following features:

(1) "maximum margin" separator,

(2) defined by "support vectors" (= boundary instances),

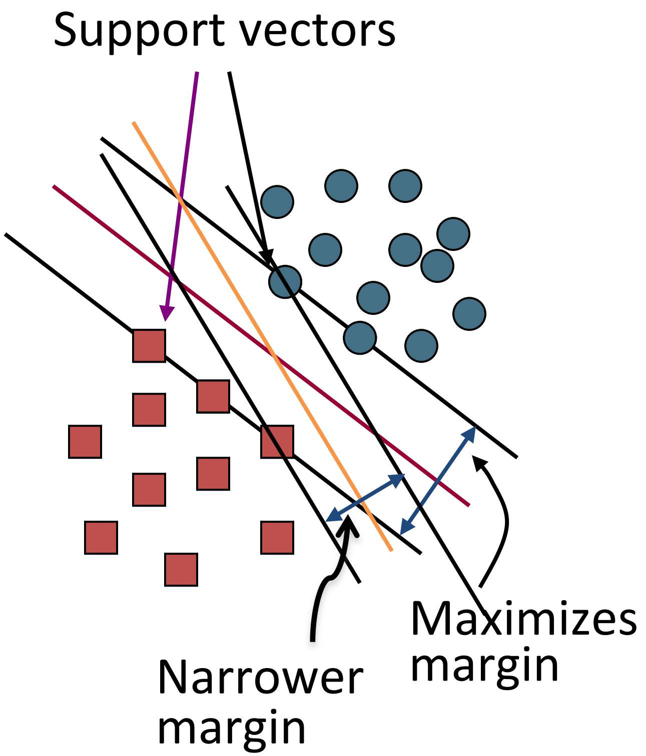
(3) can be extended to "nonlinear separators".

Let us first see features (1) and (2) by considering the linear SVM.

A part of the following slides are from the slides of Christopher Manning and Pandu Nayak (in which they ack. to Rey Mooney for borrowing his slides):
https://web.stanford.edu/class/cs276/handouts/lecture14-SVMs.ppt

Linear SVM: For the binary classification, consider the case where two classes (+1, -1) are linearly separable. (Attributes are all numerical.)

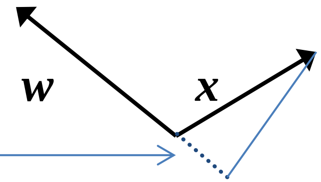
- SVMs maximize the margin around
  ||
  "distance" the hyperplane × 2
  the separating hyperplane.
  - A.k.a. large margin classifiers
- The decision function is fully specified by a subset of training samples, the support vectors.



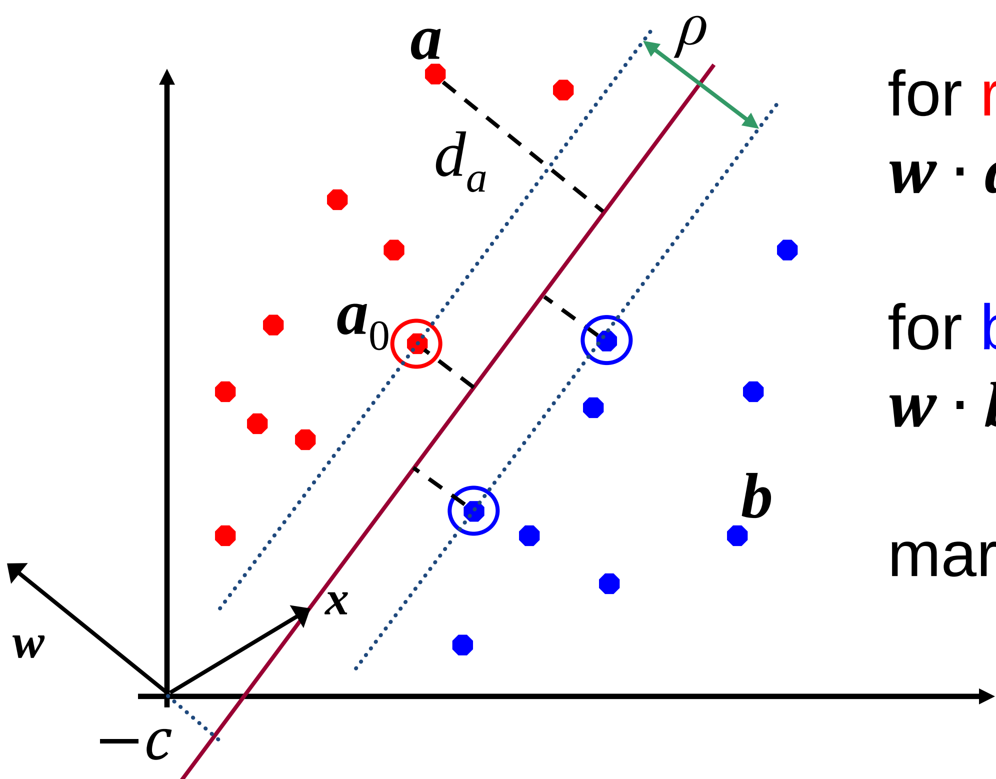
a machinery that defines the classifier

support vectors = instances (of the training set) closest to the separating hyperplane.

basics of linear algebra



- inner product $\boldsymbol{w}\cdot\boldsymbol{x}$ is
- a hyperplane is defined by its normal vector $\boldsymbol{w}$ as a set of points $\boldsymbol{x}$ such that $\boldsymbol{w}\cdot\boldsymbol{x}-(-c)=0$ (assume that $\|\boldsymbol{w}\|=1$)



for red instances $\boldsymbol{a}$, we have

$$\boldsymbol{w}\cdot\boldsymbol{a}-(-c)=\boldsymbol{w}\cdot\boldsymbol{a}+c=d_a$$

for blue instances $\boldsymbol{b}$, we have

$$\boldsymbol{w}\cdot\boldsymbol{b}-(-c)=\boldsymbol{w}\cdot\boldsymbol{b}+c=-d_b$$

margin ρ is $\rho=2(\boldsymbol{w}\cdot\boldsymbol{a}_0+c)$

How to compute support vectors?

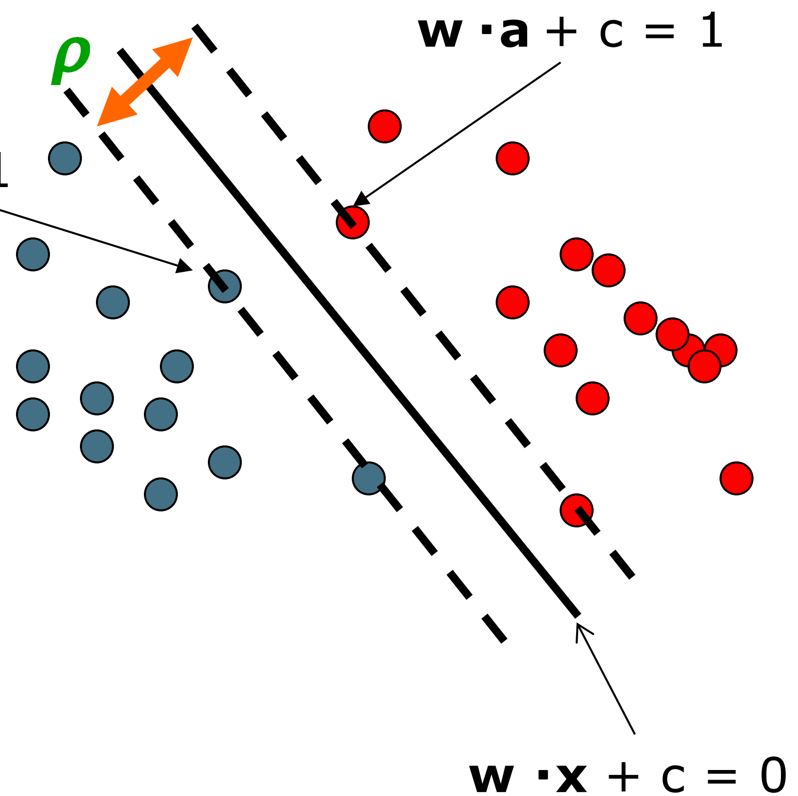
- **Hyperplane**

  $\mathbf{w} \cdot \mathbf{x} + c = 0$

  so that

  $\mathbf{w} \cdot \mathbf{x}_i + b > 0$ for positive instances,

  and < 0 for negative instances.



- Find **w** so that

  $\min_{i=1,\ldots,n} |\mathbf{w} \cdot \mathbf{x}_i + b|$

  becomes the smallest.

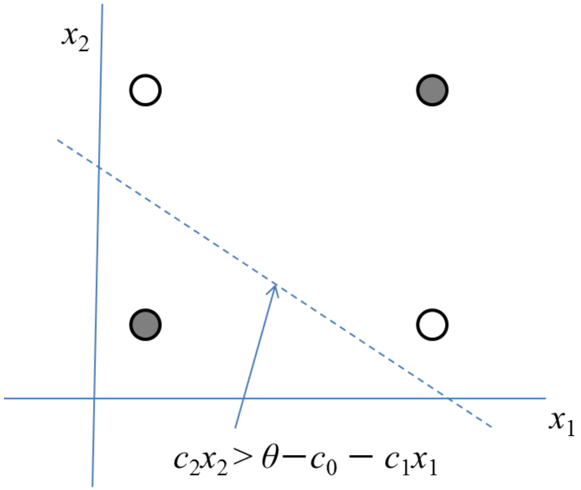
  (Here $x_1, \ldots, x_n$ are instances of the test set.)

$\Rightarrow$ an equivalent but a simpler goal

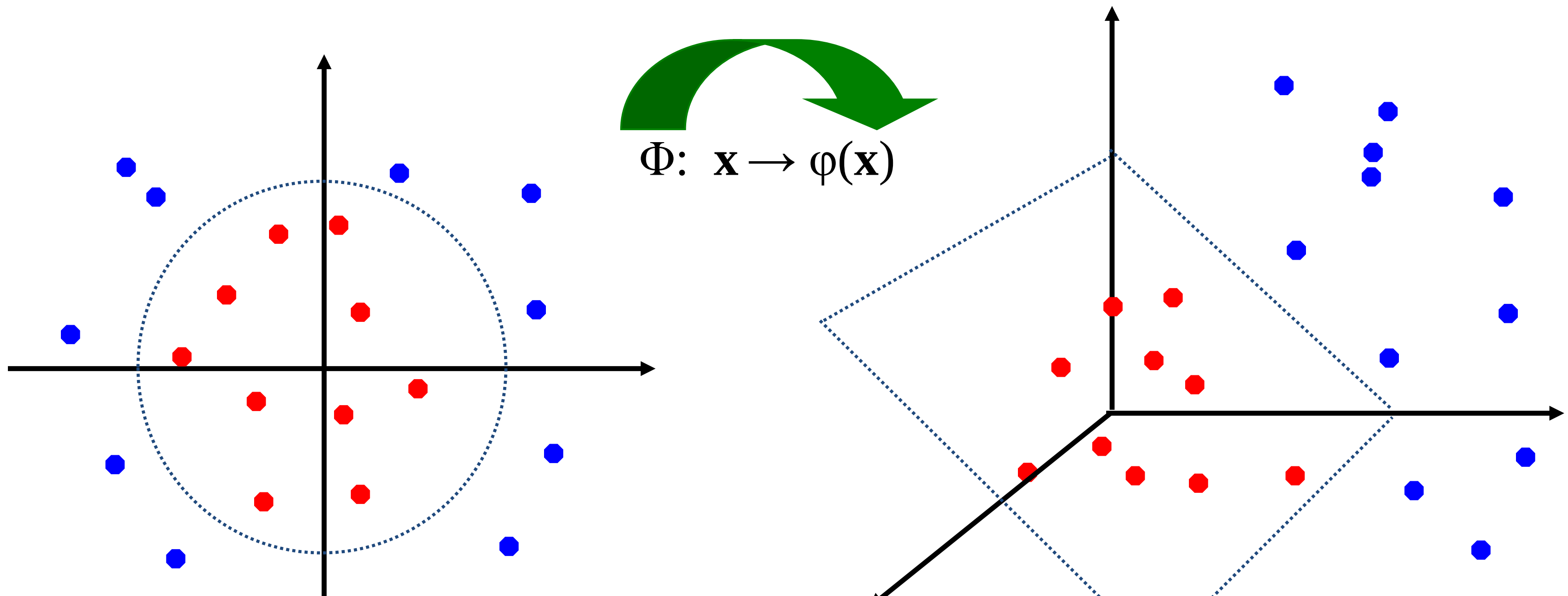
> Find **w** and *c* such that
> $\Phi(\boldsymbol{w}) = \frac{1}{2}\, \boldsymbol{w} \cdot \boldsymbol{w}$ is minimized;
> and for all $\{(\boldsymbol{x}_i, y_i)\}$: $y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + c) \geq 1$ (where y_i is the class value of x_i)

1. Support Vector Machine
1.2. Nonlinear SVM (standard SVM)

Linear separators have some limit.



- General idea: map the original space to some higher-dimensional space where the training set is separable:



2. Boosting

Boosting is one type of ensemble learning

||

combining predictions for classification

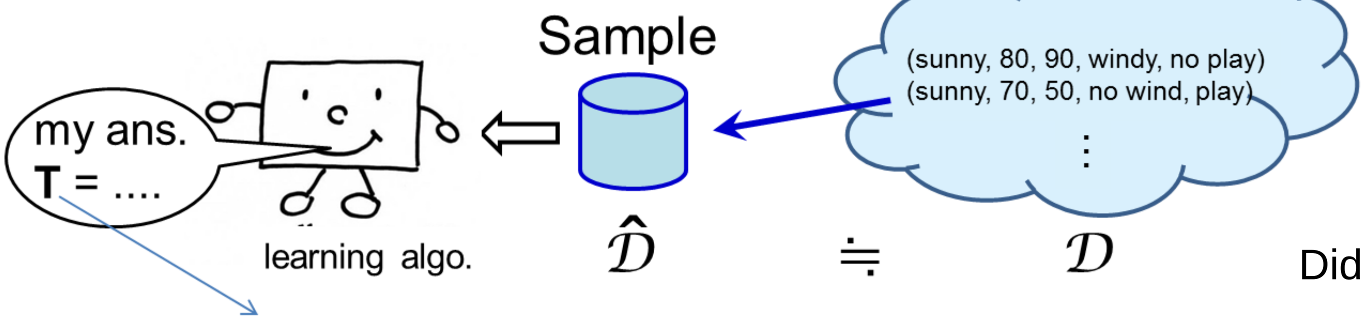
- Basic idea:
  build different "experts", let them vote
- Advantage:
  - often improves predictive performance
- Disadvantage:
  - usually produces output that is very hard to analyze
  - but: there are approaches that aim to produce a single comprehensible structure

This is guaranteed to work for the boosting under a certain assumption

2. Boosting
2.1. Basic concept

Recall ...

We assume a certain probability distribution $\mathcal{D}$ on instances (i.e., tuples of attribute values) of an assumed domain, and our data set (or, sample) is a typical example of $\mathcal{D}$. And we want our learning algorithm finds a "good rule" for $\mathcal{D}$ (or more precisely, for the classification on $\mathcal{D}$)



Did we study this?

sorry!

Want: $\Pr[\ \mathrm{T}(x)$ is correct $\mid x \leftarrow \mathcal{D}\ ] \fallingdotseq 1$

Boosting assumption

For some $\gamma > 0$, the learning algo. achieves

$$\Pr[\ \mathrm{T}(x) \text{ is correct} \mid x \leftarrow \mathcal{D}\ ] = \frac{1}{2} + \gamma$$

w.r.t. any distribution $\mathcal{D}$.

Boosting assumption

For some $\gamma > 0$, the learning algo. achieves

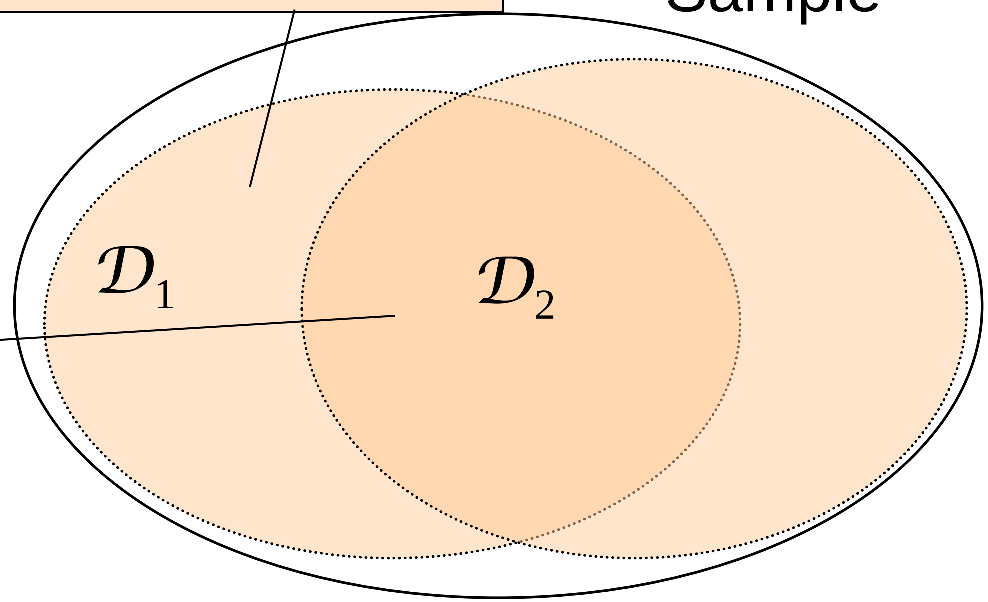
$$\Pr[\ \mathrm{T}(x) \text{ is correct} \mid x \leftarrow \mathcal{D}\ ] = \frac{1}{2} + \gamma$$

w.r.t. any distribution $\mathcal{D}$.

Initial dist. $\mathcal{D}_0 \rightarrow$ T1
Initial dist. $\mathcal{D}_1 \rightarrow$ T2
Initial dist. $\mathcal{D}_2 \rightarrow$ T3
⋮

Set of instances that T1 answers wrongly.

Sample



Set of instances that T2 answers wrongly.

Final model = *weighted majority vote* of T1, T2, T3, ...

An example for intuition

Min. Enclosing Disk Problem

Input: n points in some 2D area

output: the smallest disk covering points

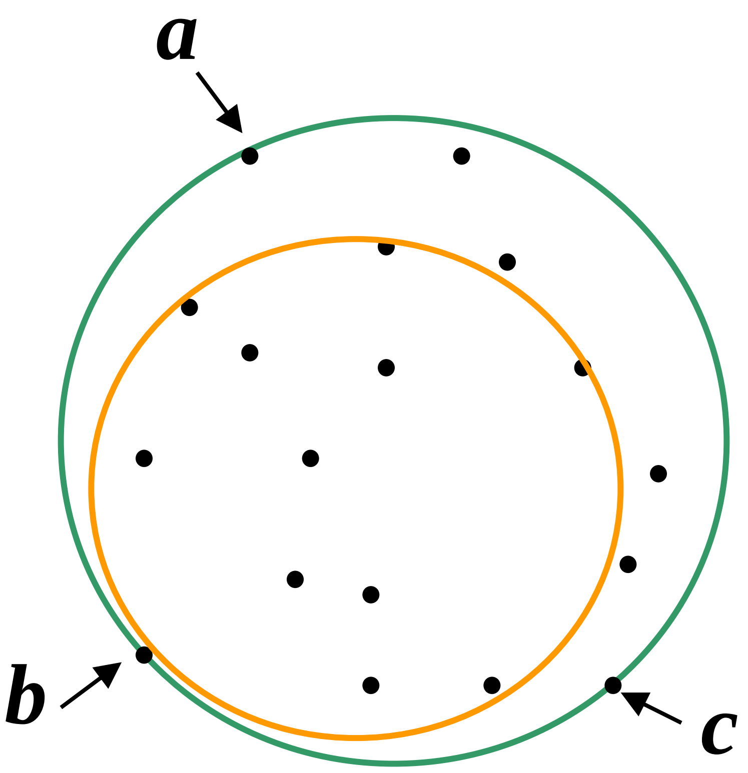
Standard

$O(n^3)$

Randomized

$O(n \log n)$

[Clarkson]



Reference: Emo Welzl, Smallest enclosing disks (balls and ellipsoids). In: Maurer H. (eds) New Results and New Trends in Computer Science. Lecture Notes in Computer Science, vol 555. Springer, 1991.

2. Boosting
2.2. AdaBoost

Depending on a way to define new distributions, there are various boosting methods.

Boosting has been invented for answering an open problem asked by L. Valiant, a founder of the PAC learning framework.

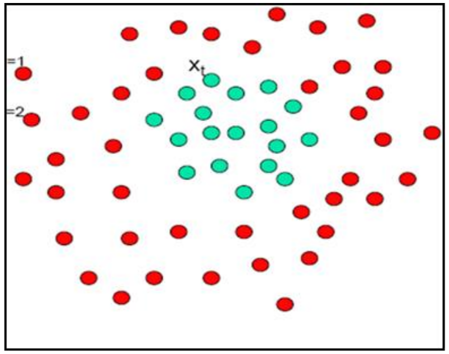
[Valiant1989]

The first boosting method [Shapire 1990] was not so practical. Boosting became a popular data mining method when *AdaBoost* has been invented.

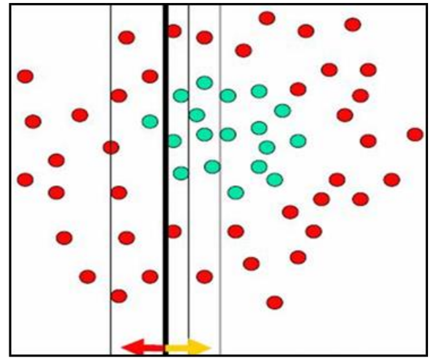
[Freund and Shapire 1995]

I also proposed MadaBoost (a similar to LogitBoost invented later).

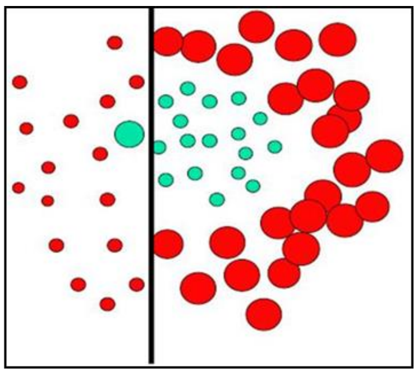
Example



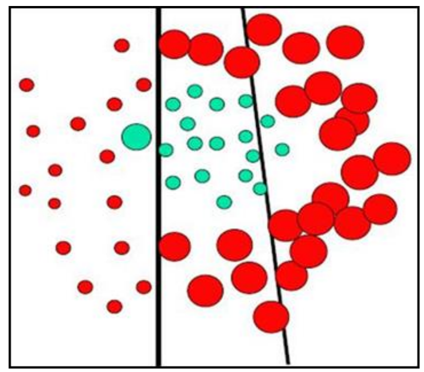
original (1st distribution)



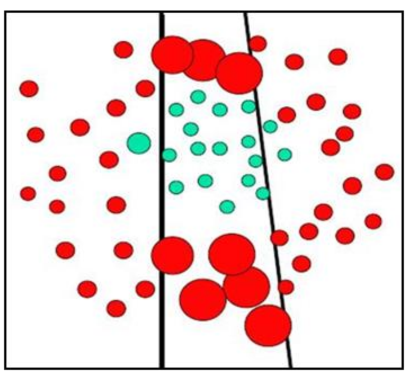
1st "weak rule"



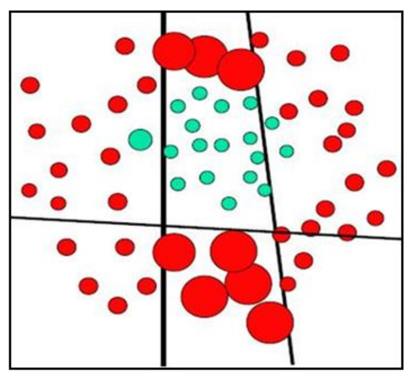
2nd distribution



2nd "weak rule"



3rd distribution

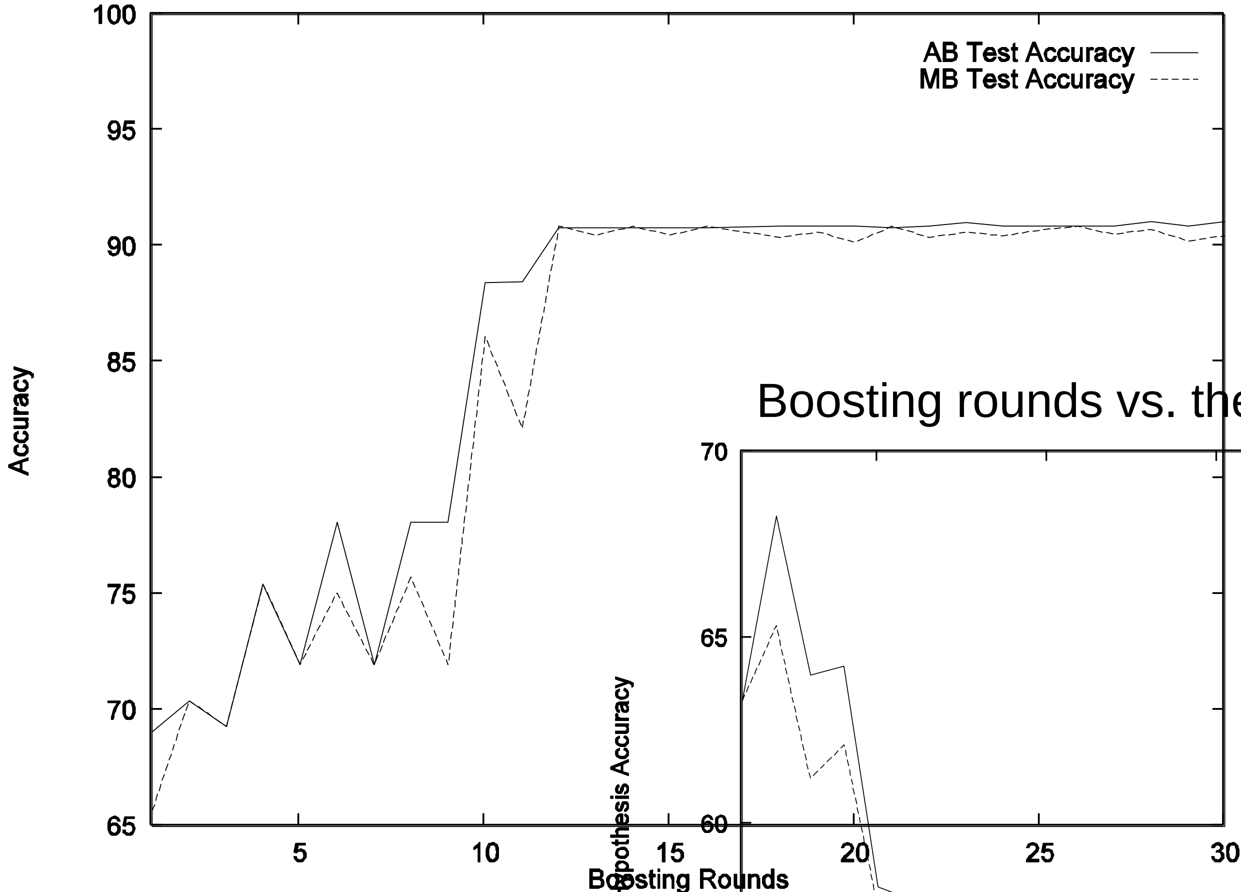


3rd "weak rule"

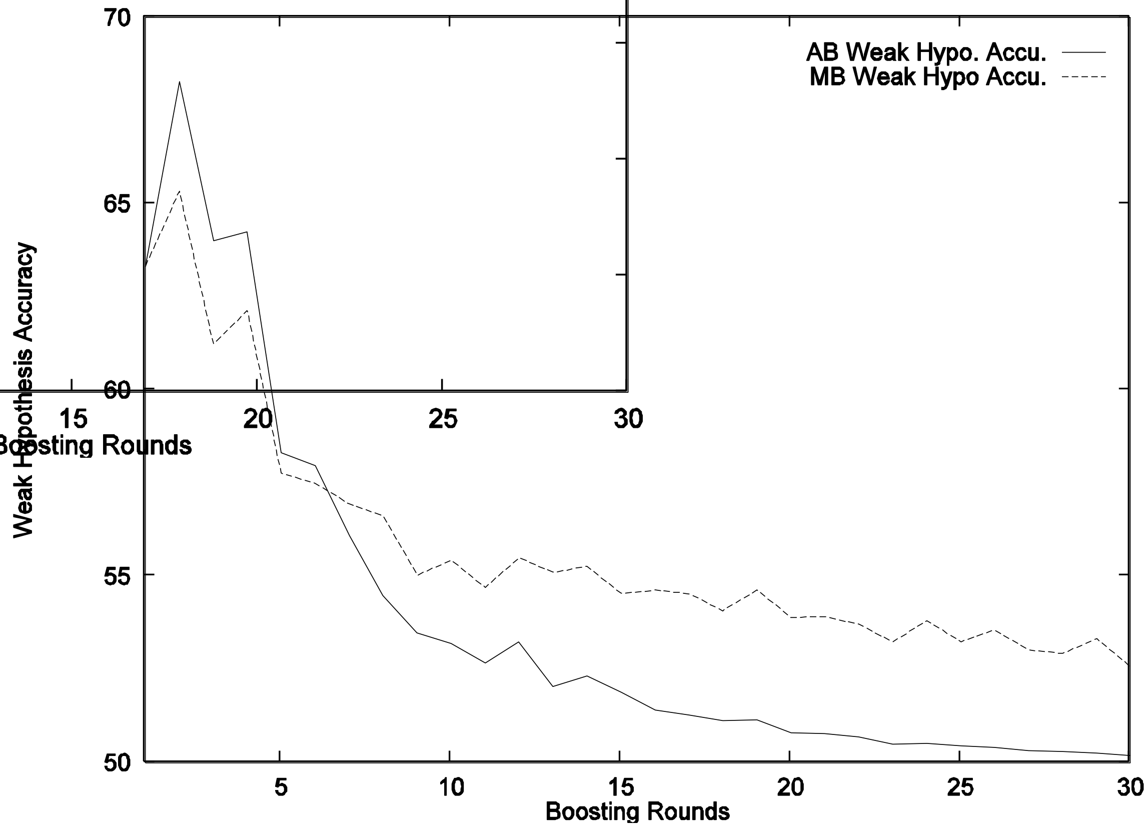
From slides of
B. Erika and K. Zsolt

http://www.cs.ubbcluj.ro/~csatol/mach_learn/bemutato/BenkKelemen_Boosting.pdf

Boosting rounds vs. the accuracy on a test set



Boosting rounds vs. the accuracy of "weak rules"



3. Normalized compression distance

A universal method for measuring similarity (mainly) between strings.

3.1. Background: Kolmogorov complexity

What is Randomness?

E.g. , Which is random?

1 0 1 0 0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0 0 1

1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0

No answer from the Probability Theory.
A. Kolmogorov suggested an answer.

Andrei N. Kolmogorov
1903 - 1987

http://www-history.mcs.st-ndrews.ac.uk

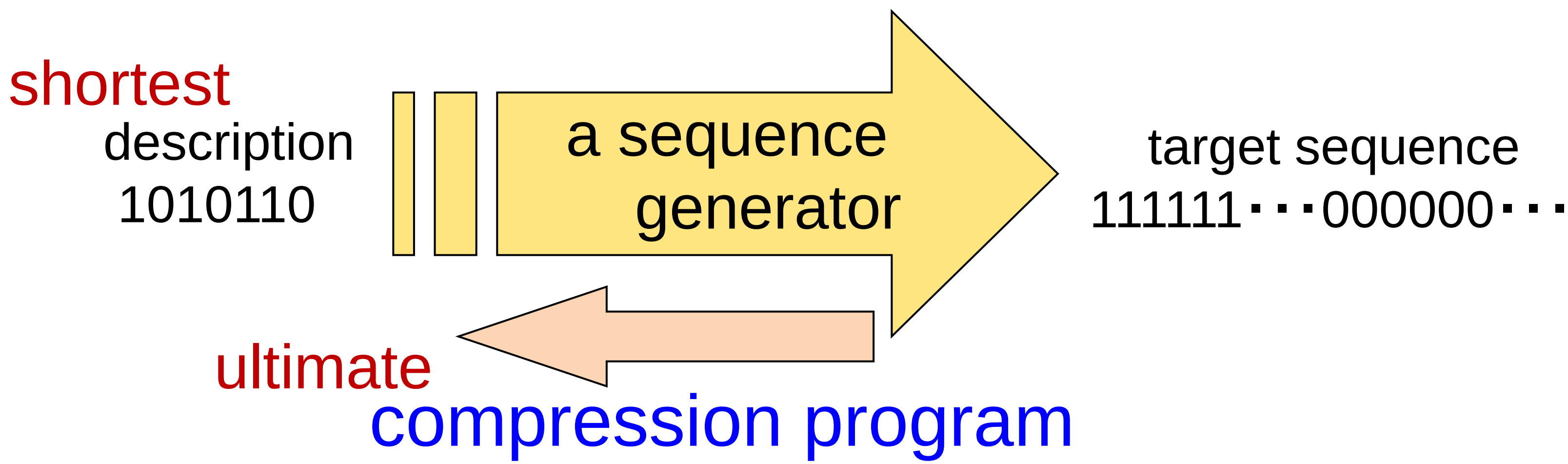
2000 bits strings

A. 1 0 1 0 0 1 0 1 1 0 1 0 0 1 0 1 .... ← 2000 bits

B. 1 1 1 1 1 1 ...... 1 1 0 0 0 0 0 0 ..... 0 0 ← 2000 bits

A. Kolmogorov ⇒ R. Solmonov, J. Chaitin

random sequence = seq. with no short description



K(x) = length of the shortest description of x

Kolmogorov complexity

relative randomness

K(x | y) = length of the shortest description of x when y is given

randomness of x relative to y

unsimilarity of x to y

Cf.

K(x) = length of the shortest description of x

randomness of x

Example:

K(x00 | x) = constant, i.e., const. bits

K(0011000011 | 01001) = const. bits

relative randomness

$K(x \mid y)$ = length of the shortest description of x when y is given

$\boxed{\text{randomness of } x \text{ relative to } y}$

unsimilarity of x to y

Let's use this !

M. Li & P. Vitanyi

but how!?

Thm. [Levin]

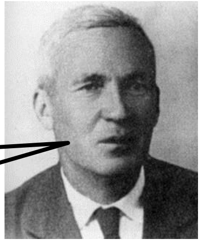
$$K(x \mid y) \fallingdotseq K(xy) - K(y)$$

Use available one!
gzip, bzip, ...

$K(x)$ = length of ultimately compressed code for x

$\fallingdotseq$ length of reasonably compressed code for x

oh!



Kolmogorov

3. Normalized compression distance
3.2. Definition and application examples

$$\text{Ideal metric} = \frac{\min(\, K(x \mid y),\, K(y \mid x)\,)}{\max(\, K(x),\, K(y)\,)}$$

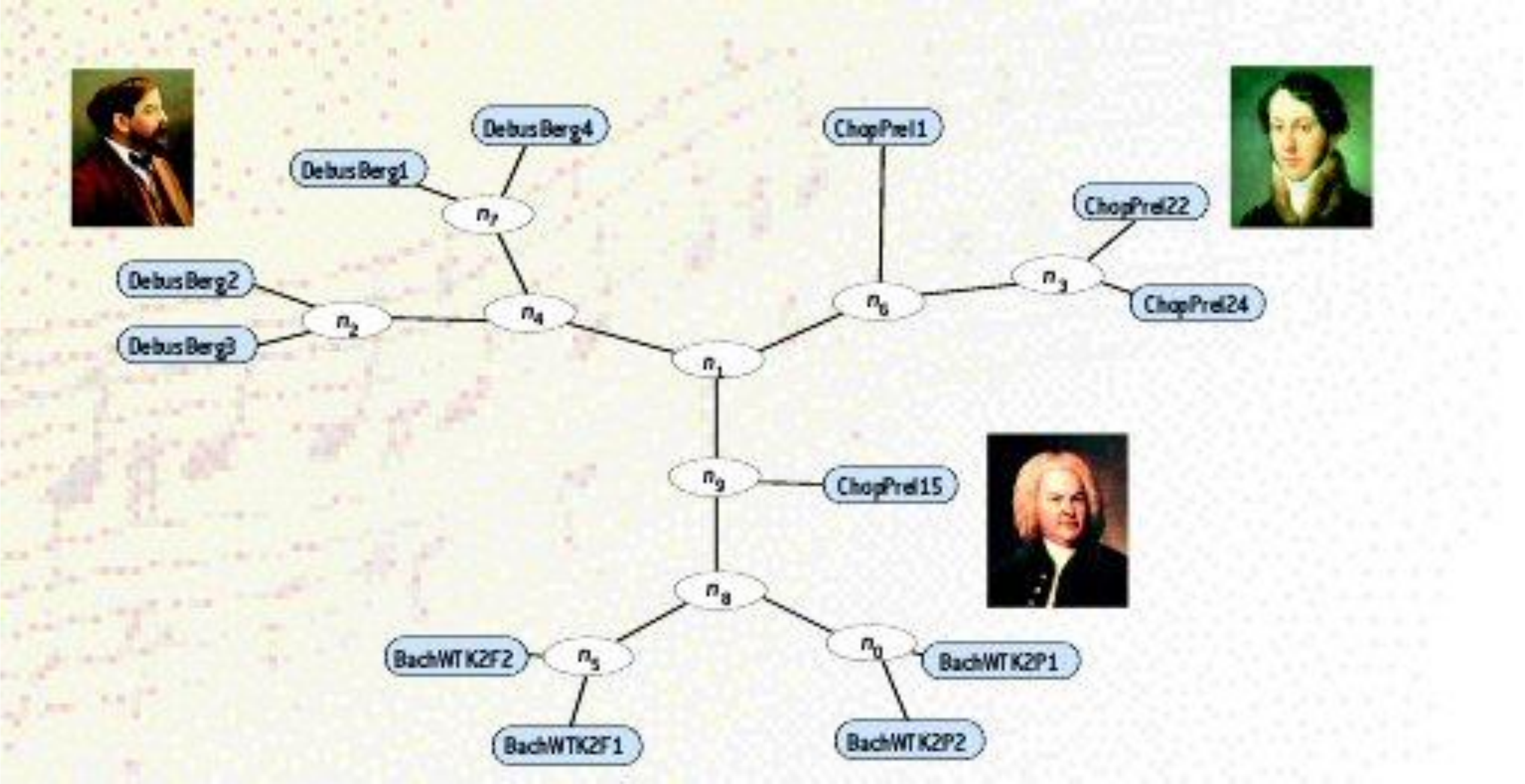
$$\fallingdotseq \frac{K(xy) - \max(\, K(x),\, K(y)\,)}{\max(\, K(x),\, K(y)\,)}$$

(Recall $K(x \mid y) \fallingdotseq K(xy) - K(y)$)

$$NCD_Z(x\,,y) = \frac{Z(xy) - \max(\, Z(x),\, Z(y)\,)}{\max(\, Z(x),\, Z(y)\,)}$$

where $Z(x)$ = the length of the compressed string x computed by compression algo. Z.

Ex1: Music piece (MIDI) similarity



R.Cilibrasi, P.Vitanyi, and R.deWolf,
Algorithmic Clustering of Music, 2003

Ex2: Russian novel similarity

5. Distances des textes d'auteurs russes par la méthode de compression : un seul texte de Tolsto est mal classé.

Ex0: Chain letter analysis [Ming Li, et al.]

Ex4: Analysis of SARS virus varieties

Ex5: Language similarity !?

NCD(English, French) > NCD(English, Spanish)

Q. What did they compare?

Advantage of NCD:
・ universal

Warning:
・ No reasoning for using a particular compression algo.
(In fact, it is said that bzip is better than gzip, but why??)

⇒ Could be used if there is no other way.

4. Homework of this week

Choose one of the following learning algorithms (or, precisely speaking, heuristics) and explain its outline and its key technical point.

Please try to write it within 5 pages by A4 size paper.
Using examples/figures is recommended.
(You may write a report in Japanese.)

1. C4.5 (a basis of J4.8)
2. Perceptron
3. Apriori algorithm (an improved version)
4. EM algorithm for clustering
5. AdaBoost