An introduction to Machine Learning

Pierre Geurts
p.geurts@ulg.ac.be

Last update: 3/10/2017

Institut Montefiore
University of Liège

Outline

- Introduction
- Supervised Learning
- Other learning protocols/frameworks

Machine Learning: definition

Two definitions:

- Machine Learning is concerned with the design, the analysis, and the application of algorithms that allow computers to learn
  - A computer learns if it improves its performance at some task with experience (i.e., by collecting **data)**

- Machine Learning is concerned with the design, the analysis, and the application of algorithms to extract a model of a system from the sole observation (or the simulation) of this system in some situations (i.e., by collecting **data**).
  - A model can be any relationship between the variables used to describe the system.
  - Two main goals: make predictions and better understand the system

Machine learning: when ?

- Learning is useful when:
  - Human expertise does not exist

    navigating on Mars
  - Humans are unable to explain their expertise

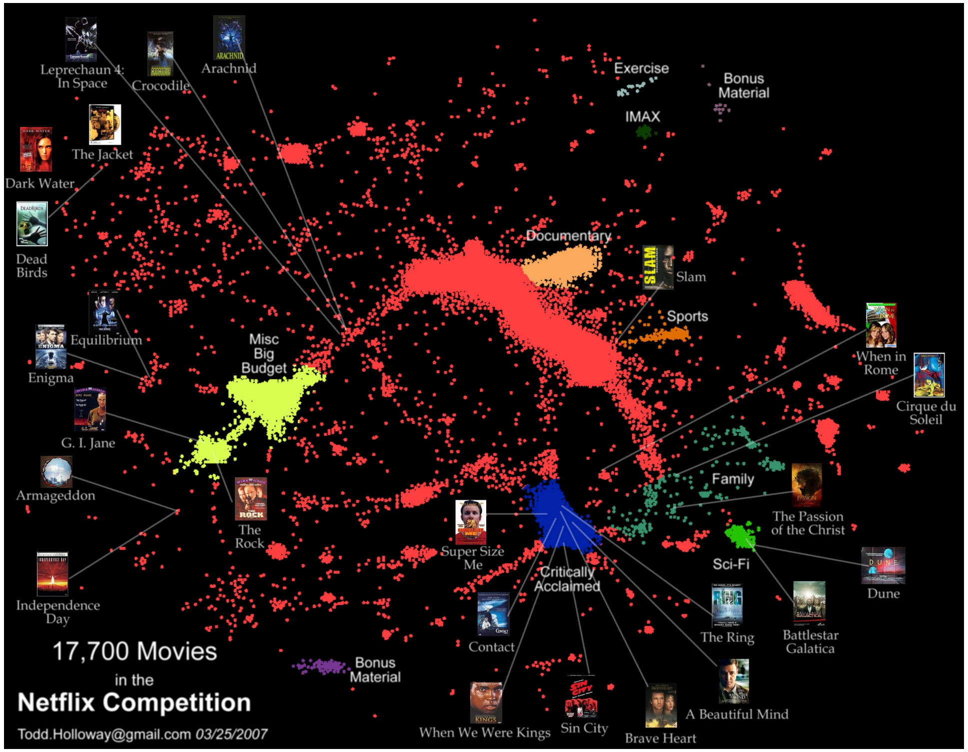
    speech recognition
  - Solution changes in time

    routing on a computer network
  - Solution needs to be adapted to particular cases

    user biometrics
- Example:
  - It is easier to write a program that learns to play checkers or backgammon well by self-play rather than converting the expertise of a master player to a program.

Applications: recommendation system

- Netflix prize: predict how much someone is going to love a movie based on his movies preferences
- Data: over 100 million ratings that over 480,000 users gave to nearly 18,000 movies
- Reward: $1,000,000 dollars if 10% improvement with respect to Netflix's system in 2006 (two teams succeeded in 2009)

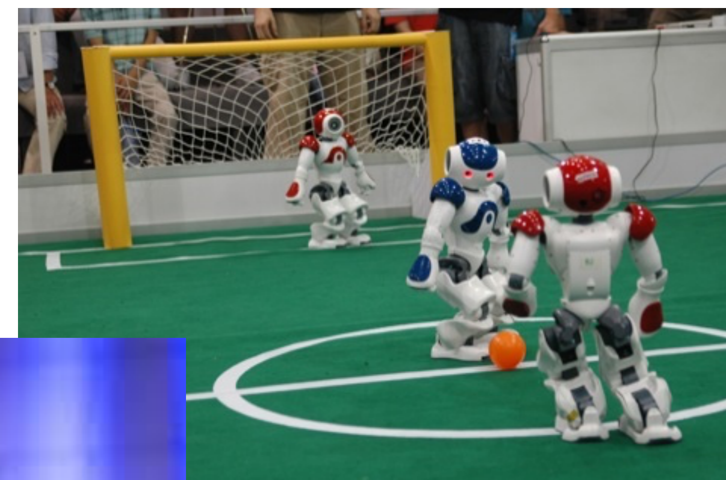


http://www.netflixprize.com

Applications: robotics

Machine learning is a core technology within robotics:

- To better sense the environment (computer vision, sound recognition…)
- To decide on which sequence of actions to take to perform a given task (reinforcement learning, imitation learning…)



Applications: credit risk analysis

- Data:

Customer103: (time=t0)	*Customer103:* (time=t1)	...	*Customer103:* (time=tn)
Years of credit: 9	Years of credit: 9		Years of credit: 9
Loan balance: $2,400	Loan balance: $3,250		Loan balance: $4,500
Income: $52k	Income: ?		Income: ?
Own House: Yes	Own House: Yes		Own House: Yes
Other delinquent accts: 2	Other delinquent accts: 2		Other delinquent accts: 3
Max billing cycles late: 3	Max billing cycles late: 4		Max billing cycles late: 6
Profitable customer?: ?	Profitable customer?: ?		**Profitable customer?: No**
...	...		...

- Logical rules automatically learned from data:

```
If   Other-Delinquent-Accounts > 2, and
     Number-Delinquent-Billing-Cycles > 1
Then Profitable-Customer? = No
     [Deny Credit Card application]
If   Other-Delinquent-Accounts = 0, and
     (Income > $30k)  OR  (Years-of-Credit > 3)
Then Profitable-Customer? = Yes
     [Accept Credit Card application]
```

Some applications at ULg

Wide area control of power systems (ULg, PEPITe, Hydro-Québec)

Problem

- Improve emergency control scheme
  - Churchill-Falls power plant
- Reduce probability of blackout

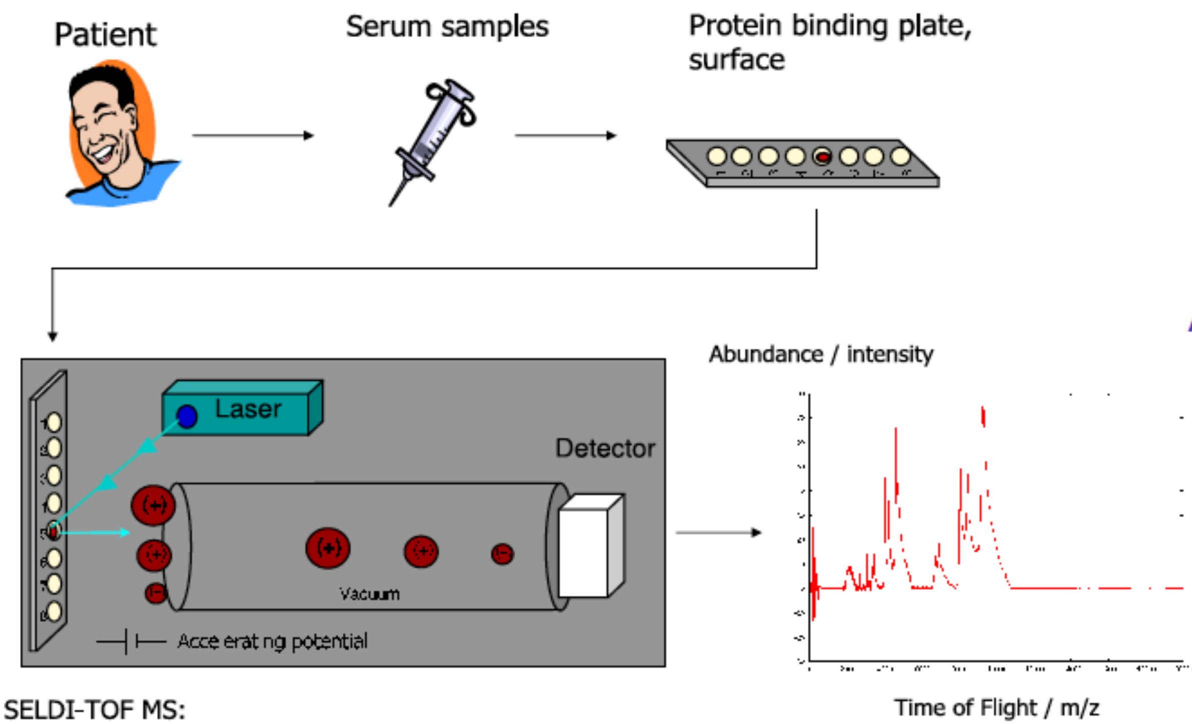
Approach

- 10,000 real-time snapshots sampled (several years)
- Massive time-domain simulations
- Automatically learn decision rules to determine optimal amount of generation and load to trip
- Implement rules in real-time
- New rules enhance security

Some applications at ULg

Medical diagnosis

(CBIG/GIGA collaboration)



SELDI-TOF MS:

Surface Enhanced Laser Desorption/ Ionisation Time of Flight Mass Spectrometry

Problem

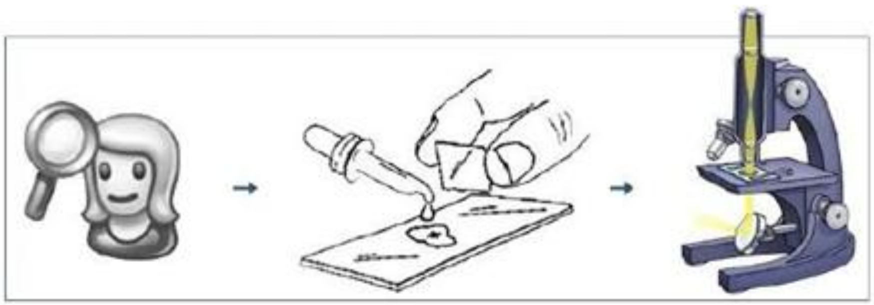
- Diagnosis of Rhumatoid Arthritis and other inflammatory diseases

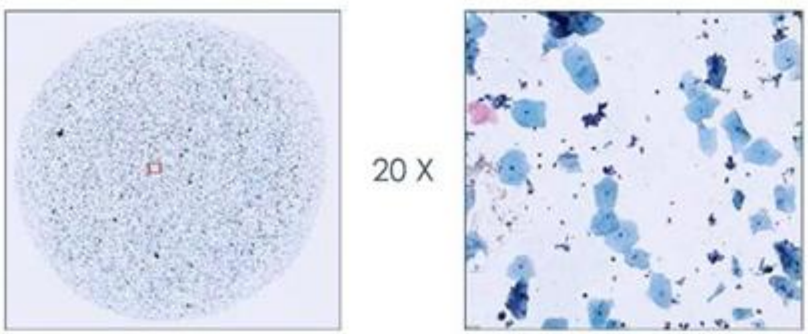
Approach [GFd+04]

- Proteomic analysis of serum samples
- Automatic learning to
  - identify biomarkers (protein fragments) specific of disease
  - derive classifier for medical diagnosis

Some applications at ULg

Computer-aided cytology





Problem

- Early diagnosis of disease based on cytological tests
- Improve detection of rare abnormal cells among tens of thousands of normal cells

Approach

- Pathologists collect a database of normal and abnormal cells
- Automatically learn a cell classification model
- Scan whole-slide digital images (>= 40000 x 40000 pixels) and rank most suspicious ones for expert review

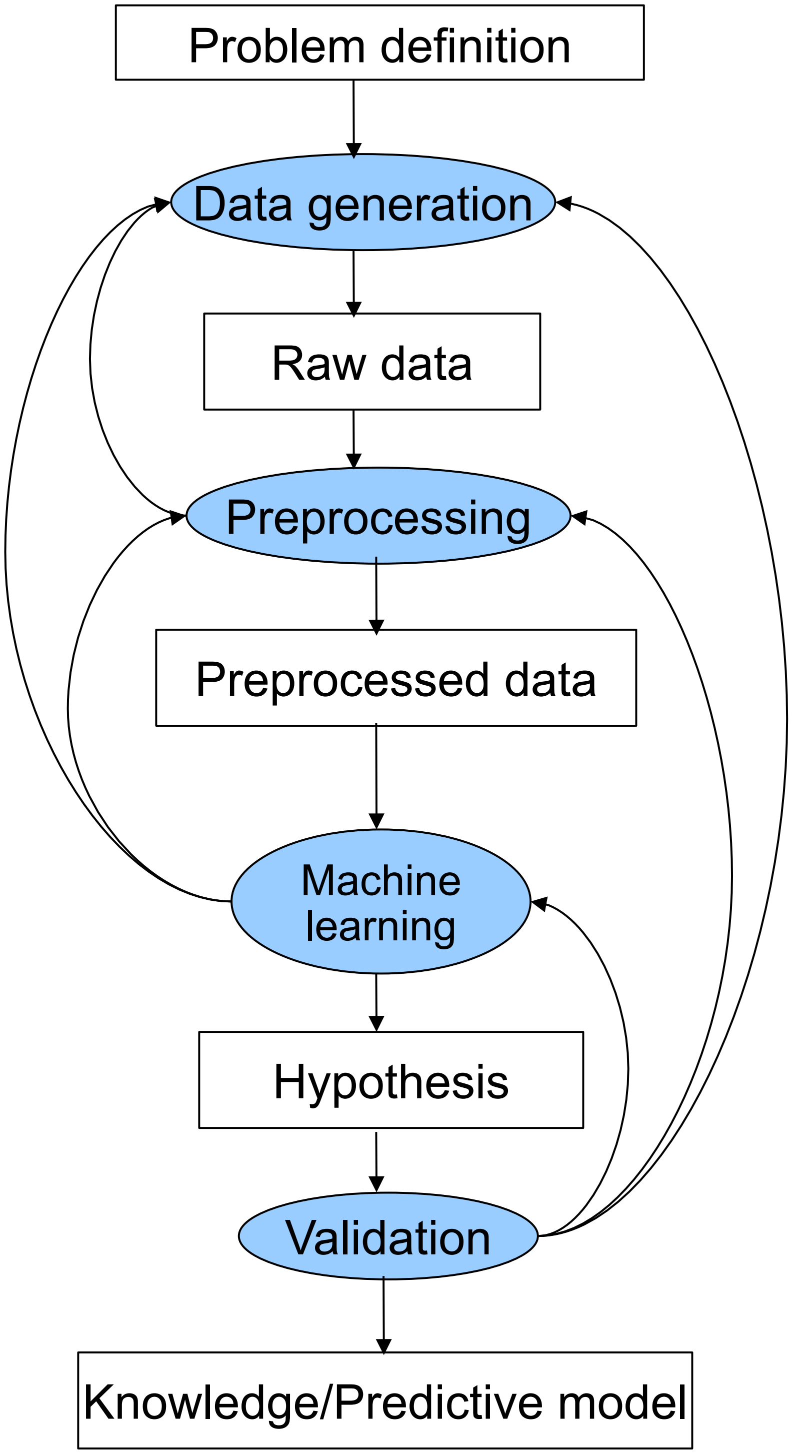
(Collaboration with US company CellSolutions and PEPITe)

Other applications

- Machine learning has a wide spectrum of applications including:
  - Retail: Market basket analysis, Customer relationship management (CRM)
  - Finance: Credit scoring, fraud detection
  - Manufacturing: Optimization, troubleshooting
  - Power systems: monitoring, control
  - Medicine: Medical diagnosis
  - Telecommunications: Quality of service optimization, routing
  - Bioinformatics: Motifs, alignment, network inference
  - Web mining: Search engines
  - ...

Related fields

- Artificial Intelligence: smart algorithms
- Statistics: inference from a sample
- Computer Science: efficient algorithms and complex models
- Systems and control: analysis, modeling, and control of dynamical systems
- Data Mining/data science: searching through large volumes of data



One part of the data mining process

- Each step generates many questions:
  - Data generation: data types, sample size, online/offline...
  - Preprocessing: normalization, missing values, feature selection/extraction...
  - Machine learning: hypothesis, choice of learning paradigm/algorithm...
  - Hypothesis validation: cross-validation, model deployment...

Glossary

- Data=a table (dataset, database, sample)

Variables (attributes, features) =
measurements made on objects

	VAR 1	VAR 2	VAR 3	VAR 4	VAR 5	VAR 6	VAR 7	VAR 8	VAR 9	VAR 10	VAR 11	...
Object 1	0	1	2	0	1	1	2	1	0	2	0	...
Object 2	2	1	2	0	1	1	0	2	1	0	2	...
Object 3	0	0	1	0	1	1	2	0	2	1	2	...
Object 4	1	1	2	2	0	0	0	1	2	1	1	...
Object 5	0	1	0	2	1	0	2	1	1	0	1	...
Object 6	0	1	2	1	1	1	1	1	1	1	1	...
Object 7	2	1	0	1	1	2	2	2	1	1	1	...
Object 8	2	2	1	0	0	0	1	1	1	1	2	...
Object 9	1	1	0	1	0	0	0	0	1	2	1	...
Object 10	1	2	2	0	1	0	1	2	1	0	1	...
...	...	...	...	...	...	...	...	...	...	...	...	...

Objects (samples, observations,
individuals, examples, patterns)

Dimension=number of variables
Size=number of objects

- Objects: samples, patients, documents, images...
- Variables: genes, proteins, words, pixels...

Outline

- Introduction
- Supervised Learning
  - Introduction
  - Model selection, cross-validation, overfitting
  - Some supervised learning algorithms
  - Beyond classification and regression
- Other learning protocols/frameworks

Supervised learning

Inputs				Output
X1	X2	X3	X4	Y
-0.61	-0.43	Y	0.51	Healthy
-2.3	-1.2	N	-0.21	Disease
0.33	-0.16	N	0.3	Healthy
0.23	-0.87	Y	0.09	Disease
-0.69	0.65	N	0.58	Healthy
0.61	0.92	Y	0.02	Disease

Learning sample

Supervised learning $\longrightarrow$

$$\hat{Y} = f(X_1, X_2, X_3, X_4)$$

model, hypothesis

- Goal: from the database (learning sample), find a function *f* of the inputs that approximates ***at best*** the output
- Formally:

  From a learning sample $\{(x_i, y_i) | i = 1, \ldots, N\}$ with $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$, find a function $f : \mathcal{X} \to \mathcal{Y}$ that minimizes the expectation of some loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to I\!R$ over the joint distribution of input/output pairs:

$$E_{x,y}\{\ell(f(x), y)\}$$

- Symbolic output $\Rightarrow$ *classification*, Numerical output $\Rightarrow$ *regression*

Two main goals

- Predictive:

Make predictions for a ***new*** object described by its attributes

X1	X2	X3	X4	Y
-0.71	-0.27	T	-0.72	Healthy
-2.3	-1.2	F	-0.92	Disease
0.42	0.26	F	-0.06	Healthy
0.84	-0.78	T	-0.3	Disease
-0.55	-0.63	F	-0.02	Healthy
0.07	0.24	T	0.4	Disease
0.75	0.49	F	-0.88	?

- Informative:

Help to understand the relationship between the inputs and the output

$$\hat{Y} = \text{disease if } X_3 = F \text{ and } X_2 < 0.3$$

Find the most relevant inputs

Example of applications

- Biomedical domain: medical diagnosis, differentiation of diseases, prediction of the response to a treatment...

Gene expression, Metabolite concentrations...

Patients

X1	X2	...	X4	Y
-0.1	0.02	...	0.01	Healthy
-2.3	-1.2	...	0.88	Disease
0	0.65	...	-0.69	Healthy
0.71	0.85	...	-0.03	Disease
-0.18	0.14	...	0.84	Healthy
-0.64	0.15	...	0.03	Disease

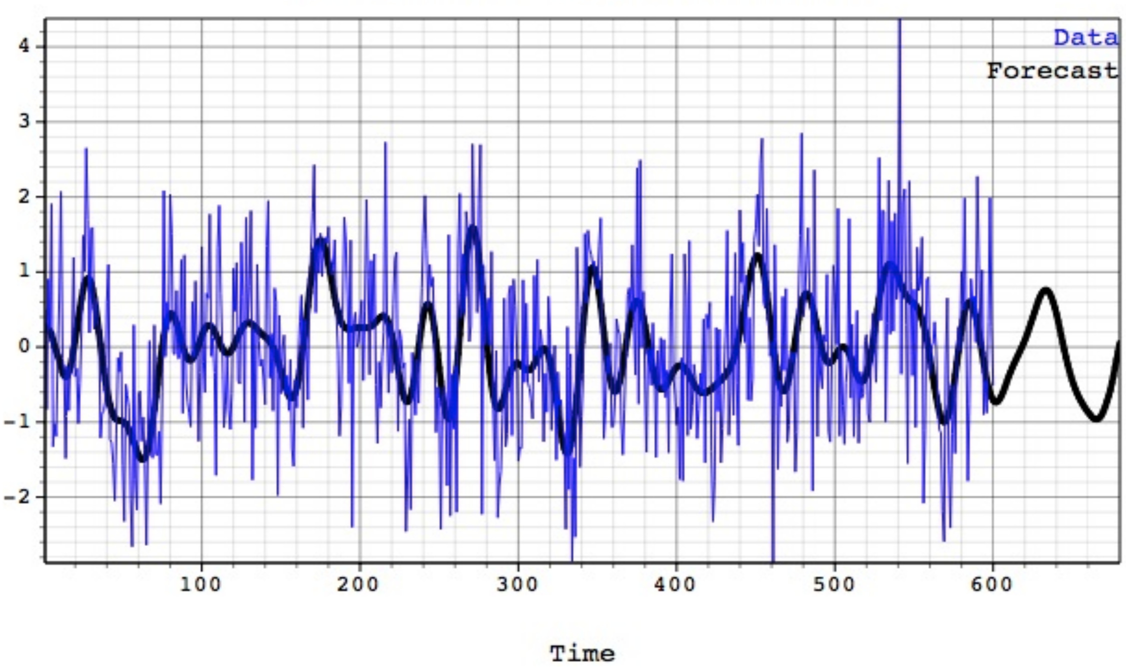
Example of applications

- Perceptual tasks: handwritten character recognition, speech recognition...

- Inputs:
  - a grey intensity [0,255] for each pixel
  - each image is represented by a vector of pixel intensities
  - eg.: 32x32=1024 dimensions
- Output:
  - 9 discrete values
  - Y={0,1,2,...,9}

Example of applications

- Time series prediction: predicting electricity load, network usage, stock market prices...

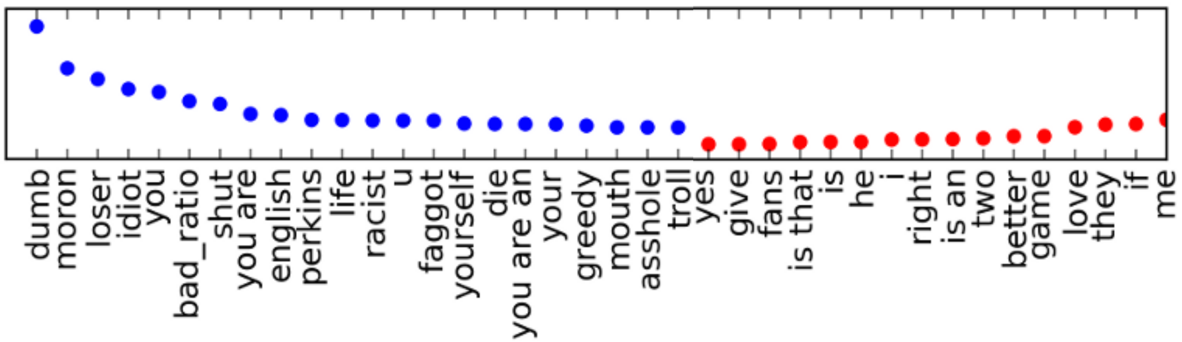


Past course projects

- Develop a system to detect North Atlantic right whales calls from audio recordings (to prevent collisions with shipping traffic)

- Develop a system to recognize (German) traffic signs (for autonomous driving)

- Detect insulting messages in social commentary



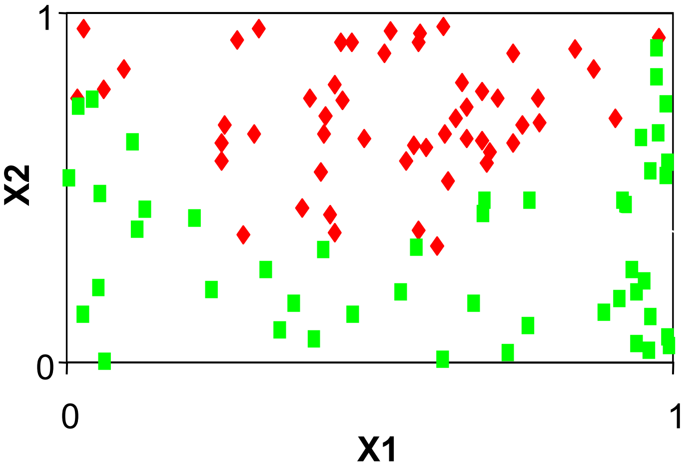
Outline

- Introduction
- Supervised Learning
  - Introduction
  - Model selection, cross-validation, overfitting
  - Some supervised learning algorithms
  - Beyond classification and regression
- Other learning protocols/frameworks

Illustrative problem

- Medical diagnosis from two measurements (eg., weights and temperature)

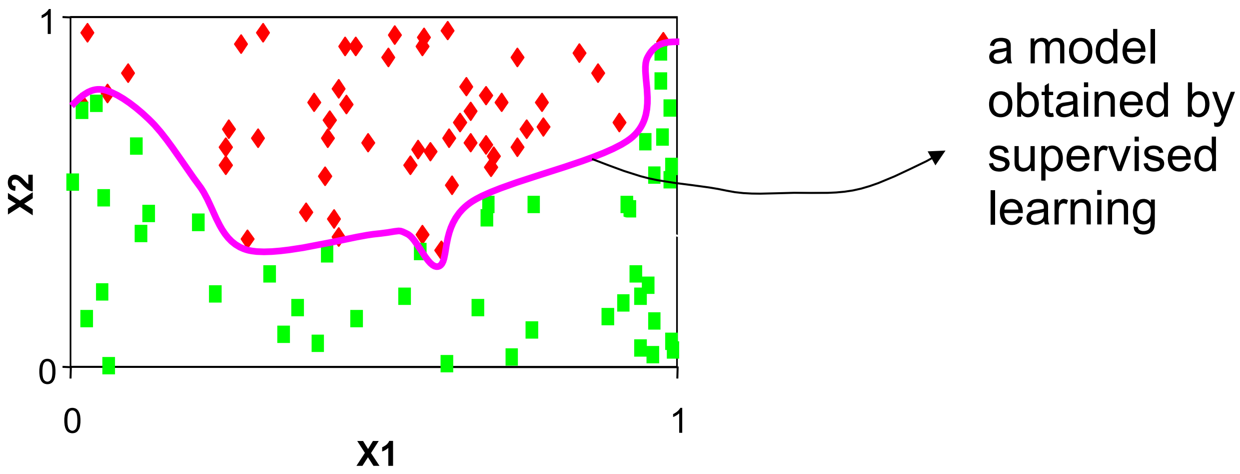
X1	X2	Y
0.93	0.9	Healthy
0.44	0.85	Disease
0.53	0.31	Healthy
0.19	0.28	Disease
...	...	...
0.57	0.09	Disease
0.12	0.47	Healthy



- Goal: find a model that classifies at best new cases for which X1 and X2 are known

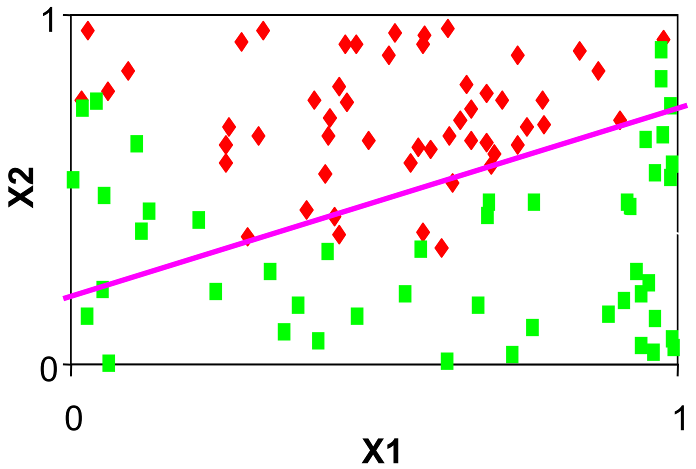
Learning algorithm

- A learning algorithm is defined by:
  - a family of candidate models (=hypothesis space *H*)
  - a quality measure for a model
  - an optimization strategy
- It takes as input a learning sample and outputs a function *h* in *H* of maximum quality



Linear model

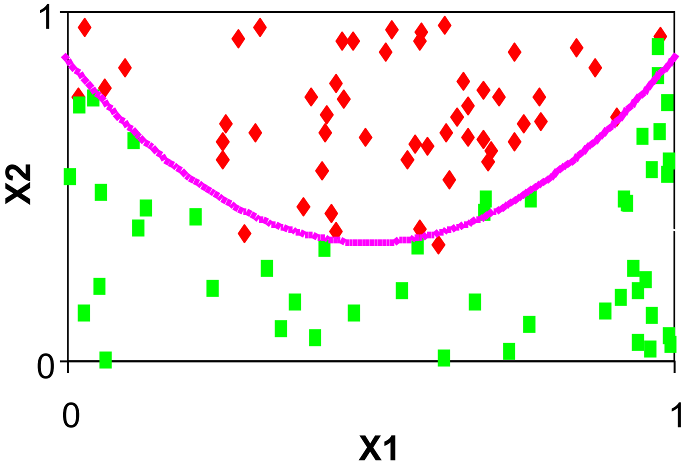
$$h(X1,X2)=\begin{cases} \text{Disease} & \text{if } w0+w1*X1+w2*X2>0 \\ \text{Normal} & \text{otherwise} \end{cases}$$



- Learning phase: from the learning sample, find the best values for w0, w1 and w2
- Many alternatives even for this simple model (LDA, Perceptron, SVM...)

Quadratic model

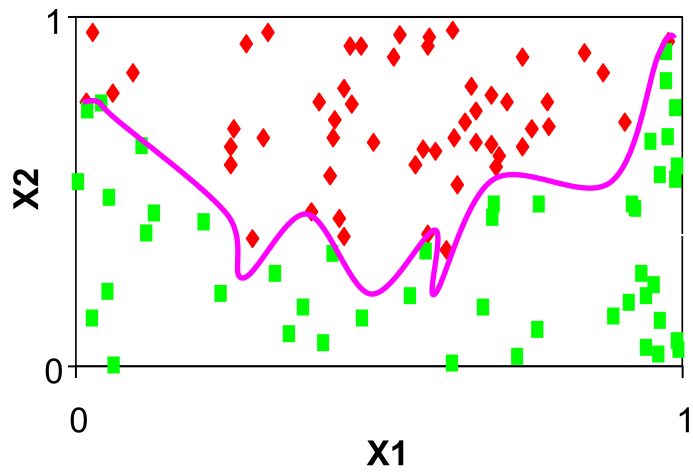
$$h(X1,X2)= \begin{cases} \text{Disease} & \text{if } w0+w1*X1+w2*X2+w3*X1^2+w4*X2^2>0 \\ \text{Normal} & \text{otherwise} \end{cases}$$



- Learning phase: from the learning sample, find the best values for w0, w1,w2, w3 and w4
- Many alternatives even for this simple model (LDA, Perceptron, SVM...)

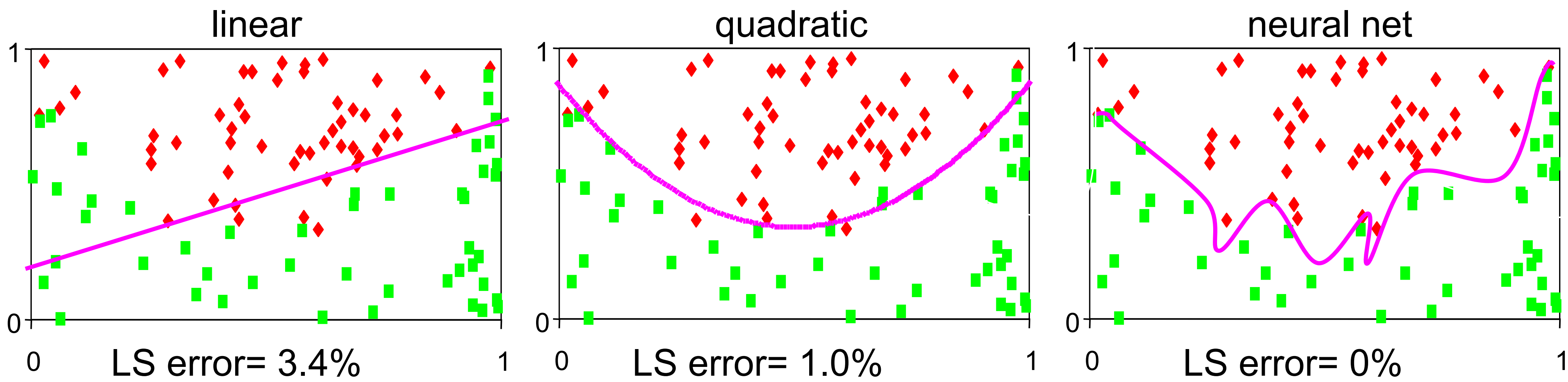
Artificial neural network

h(X1,X2)= { Disease if *some very complex function of X1,X2*>0
Normal otherwise

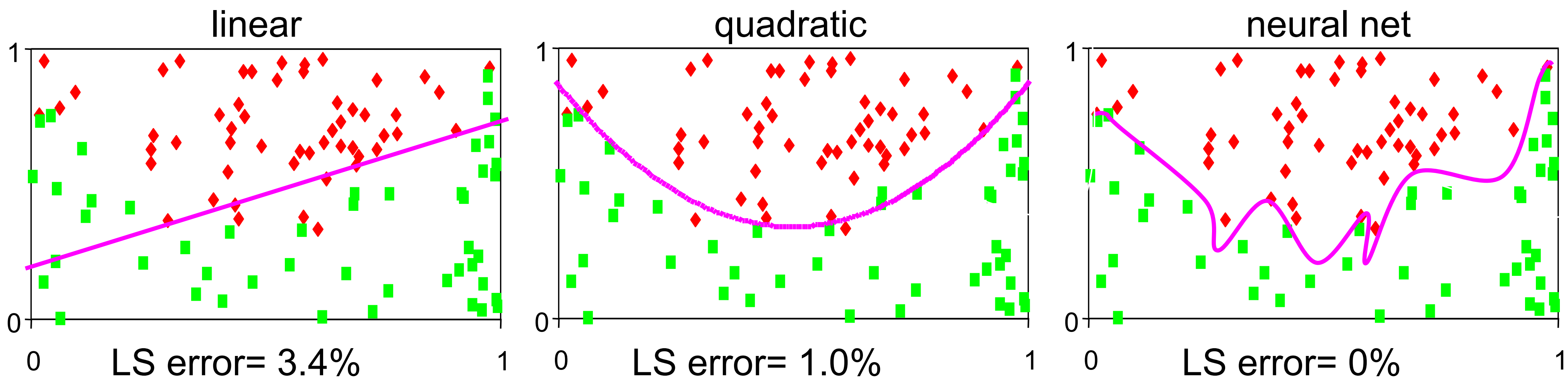


- Learning phase: from the learning sample, find the numerous parameters of the very complex function

Which model is the best?

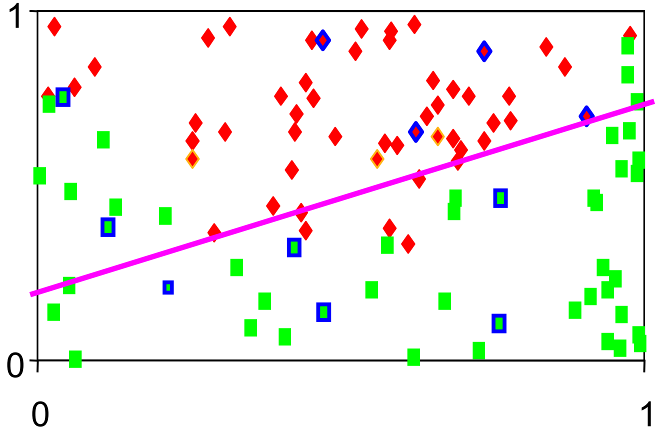


Which model is the best?



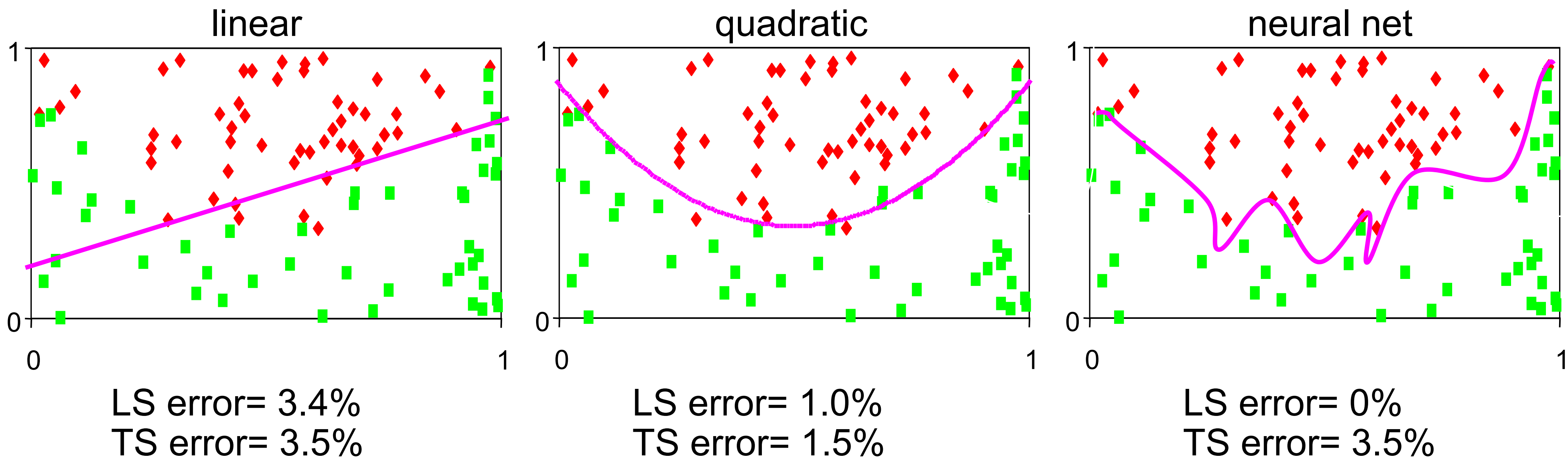
- Why not choose the model that minimises the error rate on the learning sample? (also called *re-substitution error*)
- How well are you going to predict future data drawn from the same distribution? (*generalisation error*)

The test set method



1. Randomly choose 30% of the data to be in a test sample
2. The remainder is a learning sample
3. Learn the model from the learning sample
4. Estimate its future performance on the test sample

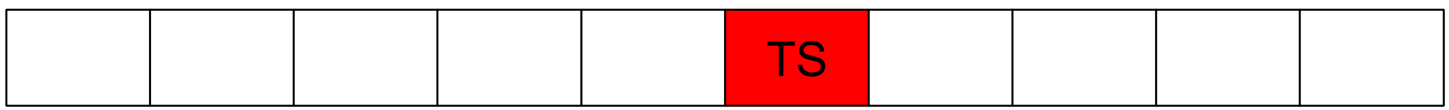
Which model is the best?



- We say that the neural network overfits the data
- Overfitting occurs when the learning algorithm starts fitting noise.
- (by opposition, the linear model underfits the data)

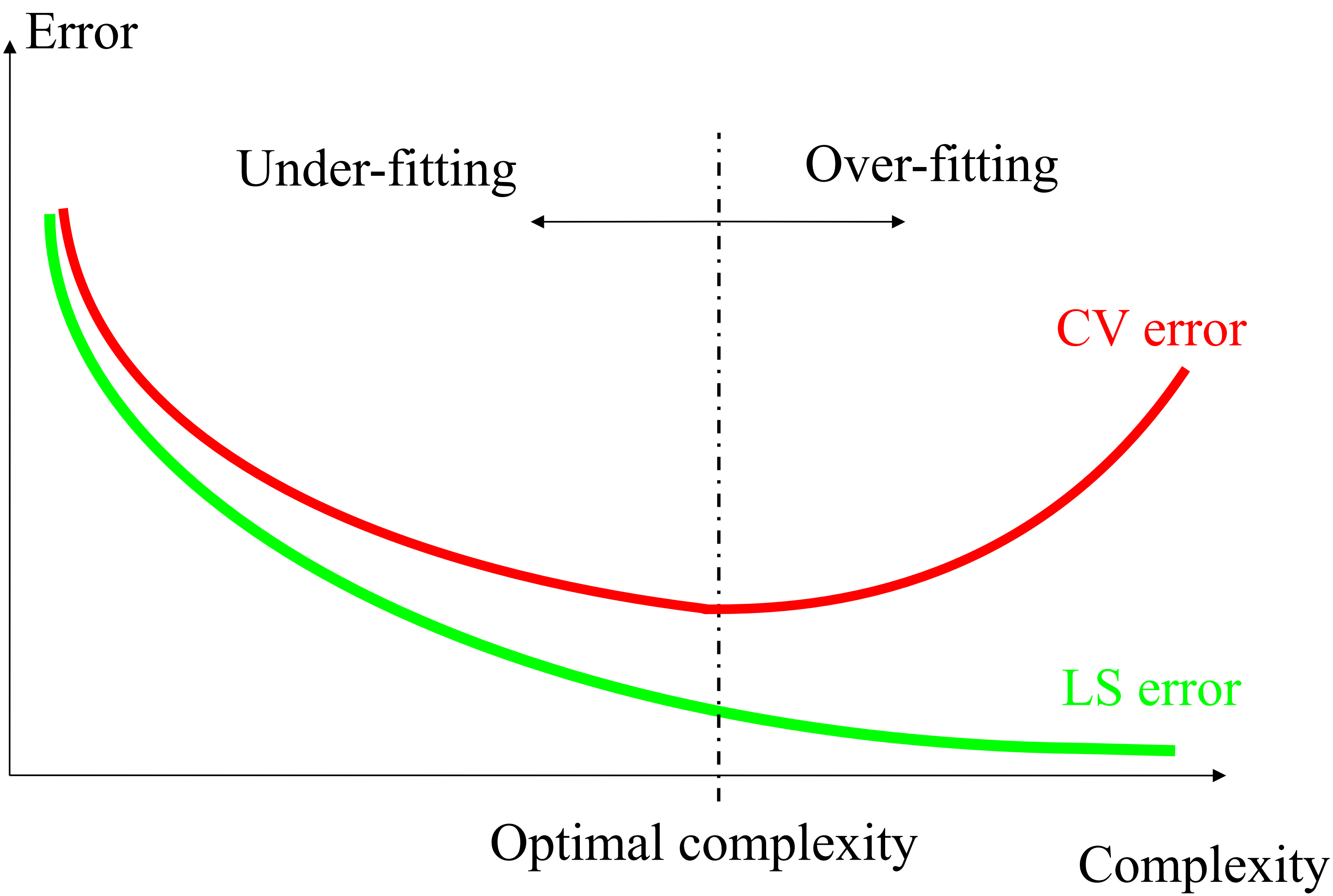
k-fold Cross Validation

- Randomly partition the dataset into *k* subsets (for example 10)



- For each subset:
  - learn the model on the objects that are not in the subset
  - compute the error rate on the points in the subset
- Report the mean error rate over the *k* subsets
- When *k*=the number of objects $\Rightarrow$ leave-one-out cross validation

CV-based complexity control



Complexity

- Controlling complexity is called regularization or smoothing
- Complexity can be controlled in several ways
  - The size of the hypothesis space: number of candidate models, range of the parameters...
  - The performance criterion: learning set performance versus parameter range, eg. minimizes

$$Err(LS)+\lambda\, C(model)$$

  - The optimization algorithms: number of iterations, nature of the optimization problem (one global optimum versus several local optima)...

Outline

- Introduction
- Model selection, cross-validation, overfitting
- Some supervised learning algorithms
  - k-NN
  - Linear methods
  - Artificial neural networks
  - Support vector machines
  - Decision trees
  - Ensemble methods
- Beyond classification and regression

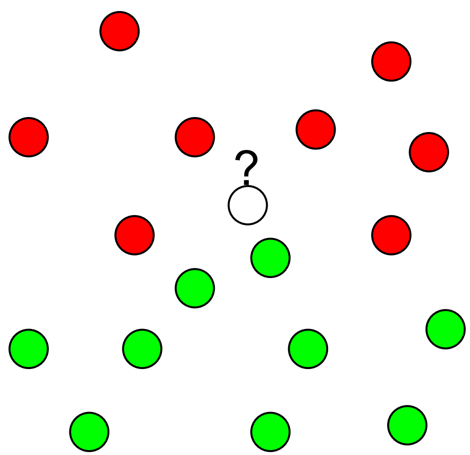
Comparison of learning algorithms

- Three main criteria:
  - Accuracy:
    - Measured by the generalization error (estimated by CV)
  - Efficiency:
    - Computing times and scalability for learning and testing
  - Interpretability:
    - Comprehension brought by the model about the input-output relationship
- Unfortunately, there is usually a trade-off between these criteria

1-Nearest Neighbor (1-NN)

(prototype based method, instance based learning, non-parametric method)

- One of the simplest learning algorithm:
  - outputs as a prediction the output associated to the sample which is the closest to the test object



	M1	M2	Y
1	0.32	0.81	Healthy
2	0.15	0.38	Disease
3	0.39	0.34	Healthy
4	0.62	0.11	Disease
5	0.92	0.43	?

$$d(5,1)=\sqrt{(0.32-0.92)^2+(0.81-0.43)^2}=0.71$$

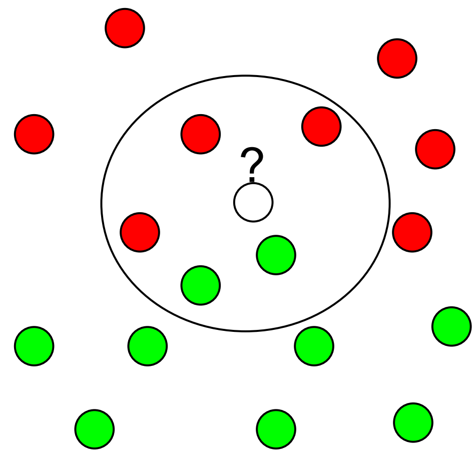
$$d(5,2)=\sqrt{(0.15-0.92)^2+(0.38-0.43)^2}=0.77$$

$$d(5,3)=\sqrt{(0.39-0.92)^2+(0.34-0.43)^2}=0.71$$

$$d(5,4)=\sqrt{(0.62-0.92)^2+(0.43-0.43)^2}=0.44$$

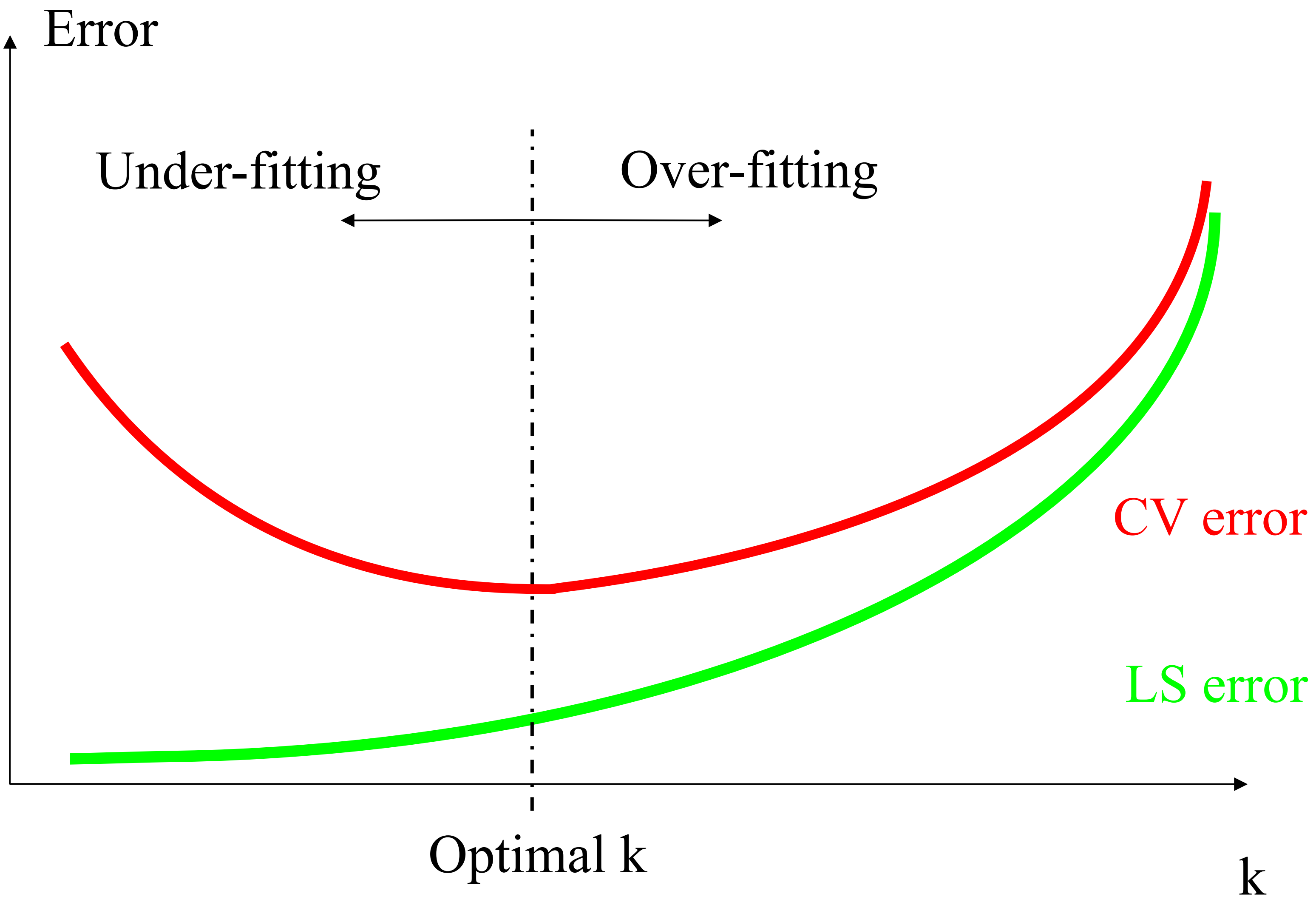
  - closest=usually of minimal Euclidian distance

Obvious extension: k-NN



- Find the k nearest neighbors (instead of only the first one) with respect to Euclidian distance
- Output the most frequent class (classification) or the average outputs (regression) among the k neighbors.

Effect of k on the error

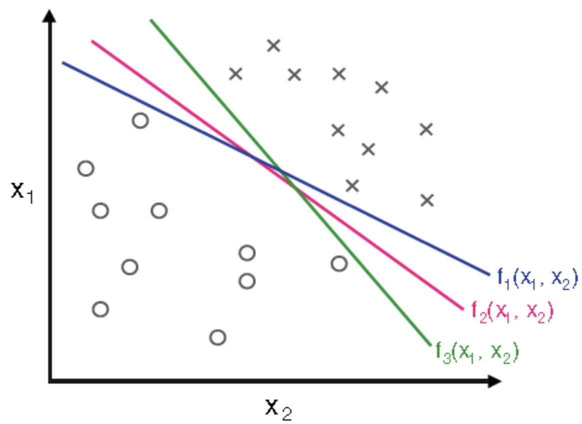


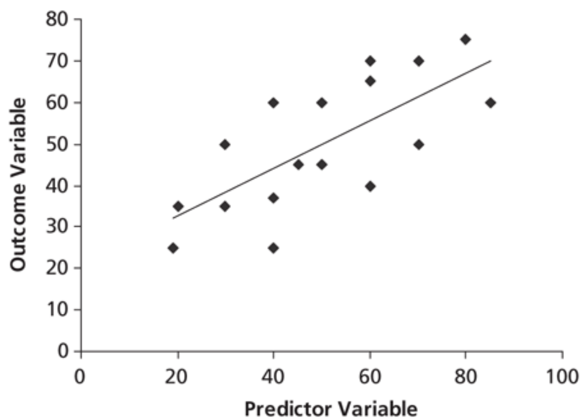
k-NN

- Advantages:
  - very simple
  - can be adapted to any data type by changing the distance measure
- Drawbacks:
  - choosing a good distance measure is a hard problem
  - very sensitive to the presence of noisy variables
  - slow for testing

Linear methods

- Find a model which is a linear combination of the inputs
  - Regression: $y = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_n w_n$
  - Classification: $y = c_1$ if $w_0 + w_1 x_1 + \ldots + w_n x_n > 0$, c_2 otherwise



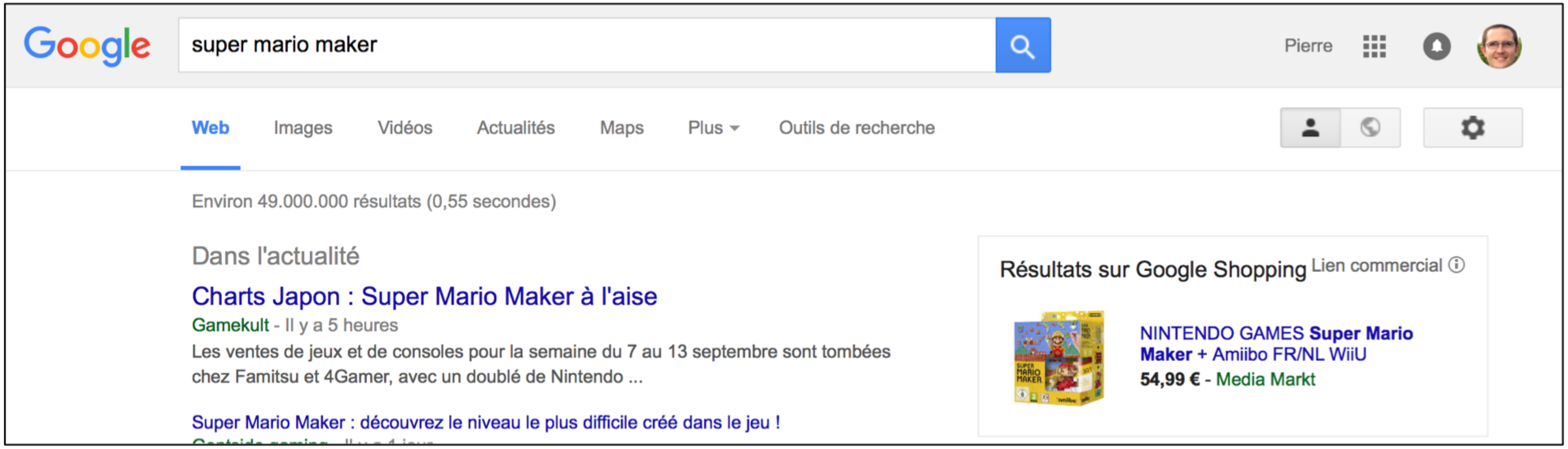


- Several methods exist to find coefficients $w_0, w_1\ldots$ corresponding to different objective functions, optimization algorithms, eg.:
  - Regression: least-square regression, ridge regression, partial least square, support vector regression, LASSO...
  - Classification: linear discriminant analysis, logistic regression, PLS-discriminant analysis, support vector machines...

Linear methods

- Advantages:
  - simple
  - there exist fast and scalable variants
  - provide interpretable models through variable weights (magnitude and sign)
- Drawbacks:
  - often not as accurate as other (non-linear) methods

Application: Click-through rate prediction



- **Goal**: given an ad and a query, predict the probability that the user will click on the ad
- Crucial to decide which ads to show, in which order and at what price
- One of the most profitable applications of ML methods
- Most used method in this domain is ***logistic regression,*** a linear classification method that is well adapted to:
  - Massive datasets: billions of samples and billions of features
  - Very sparse features: very few non-zero features per sample

Non-linear extensions

- Generalization of linear methods:

$$y = w_0 + w_1 \phi_1(x) + w_2 \phi_2(x_2) + \ldots + w_n \phi_n(x)$$

  - Any linear method can be applied (but regularization becomes more important)

- Artificial neural networks (with a single hidden layer):

$$y = g\left(\sum_j W_j\, g\left(\sum_i w_{i,j} x_i\right)\right)$$

  where g is a non linear function (eg. sigmoid)

  - (a non linear function of a linear combination of non linear functions of linear combinations of inputs)

- Kernel methods:

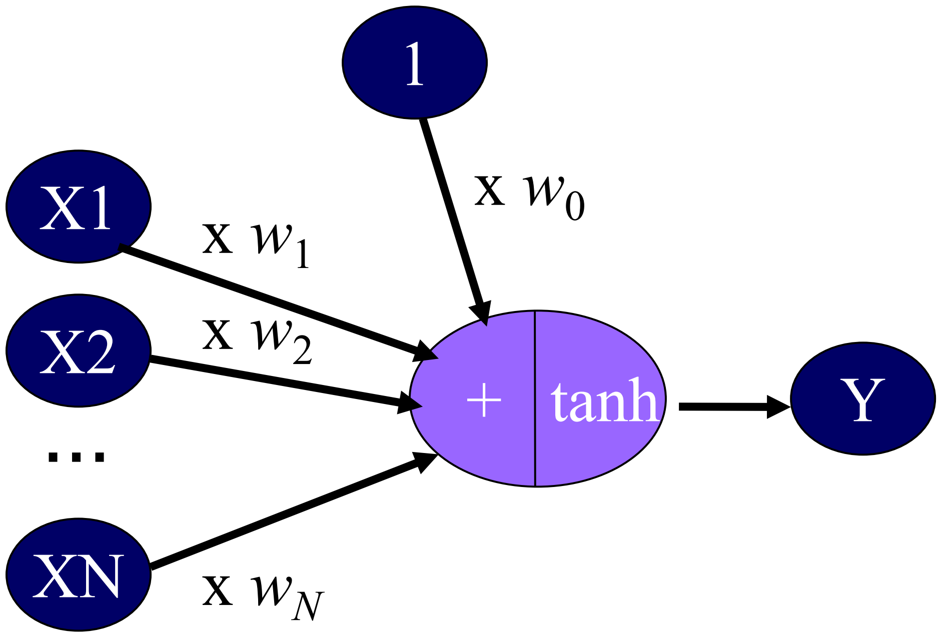
$$y = \sum_i w_i \phi_i(x) \quad \Leftrightarrow \quad y = \sum_j \alpha_j k(x_j, x)$$

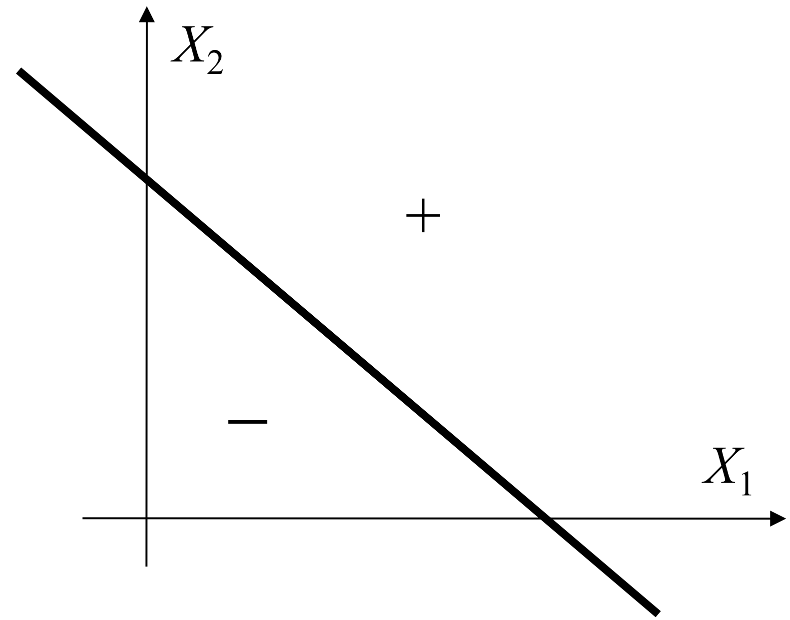
  where $k(x, x') = \langle \phi(x), \phi(x') \rangle$ is the dot-product in the feature space and *j* indexes training examples

Artificial neural networks

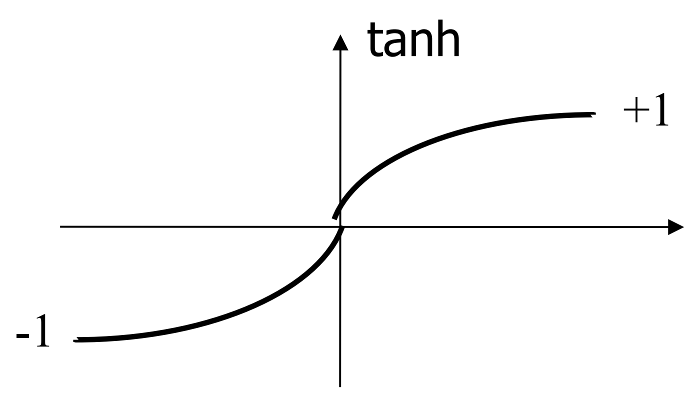
- Supervised learning method initially inspired by the behavior of the human brain
- Consists of the inter-connection of several small units
- Essentially numerical but can handle classification and discrete inputs with appropriate coding
- Introduced in the late 50s, very popular in the 90s, much less popular in the 2000s, recent come-back (new branding: *deep learning*)

Hypothesis space: a single neuron



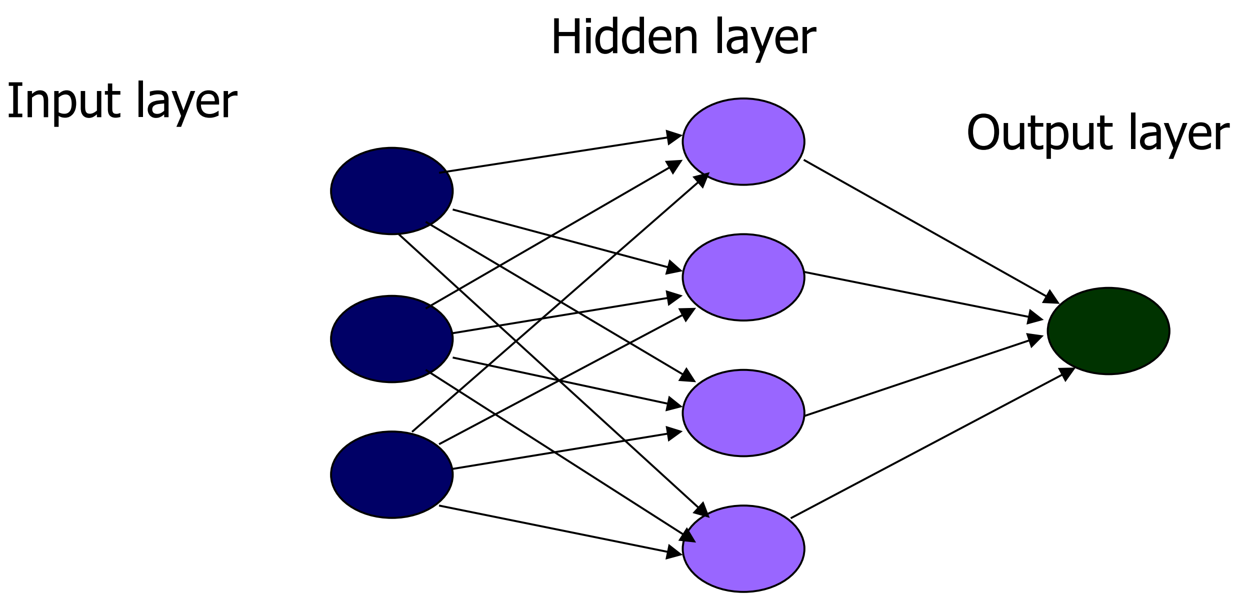


$Y=\tanh(w_1 * X_1 + w_2 * X_2 + \ldots + w_N * X_N + w_0)$



Hypothesis space: Multi-layers Perceptron

- Inter-connection of several neurons (just like in the human brain)



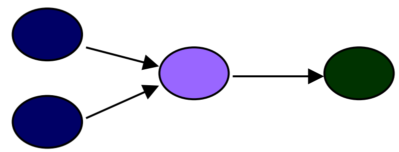
- With a sufficient number of neurons and a sufficient number of layers, a neural network can model any function of the inputs.

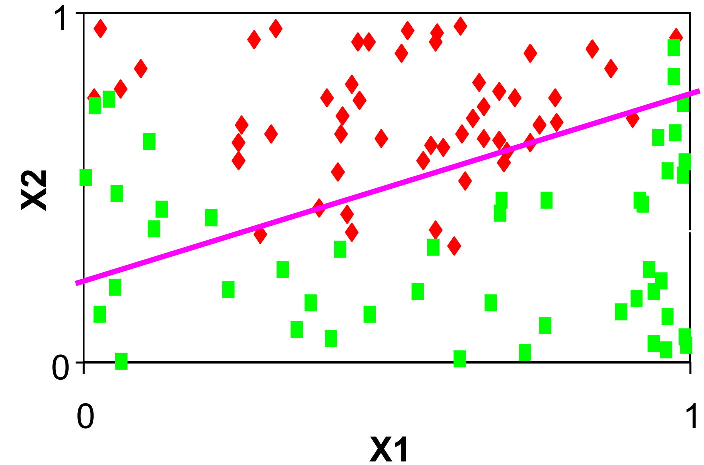
Learning

- Choose a structure
- Tune the value of the parameters (connections between neurons) so as to minimize the learning sample error.
  - Non-linear optimization by the back-propagation algorithm. In practice, quite slow.
- Repeat for different structures
- Select the structure that minimizes CV error

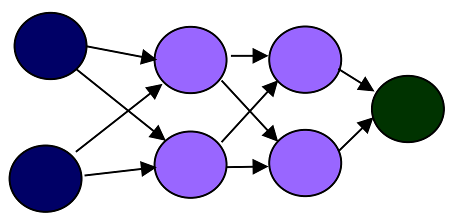
Illustrative example

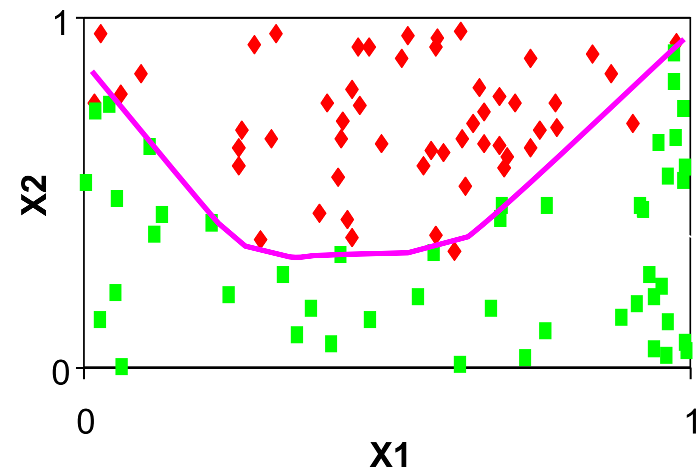
1 neuron



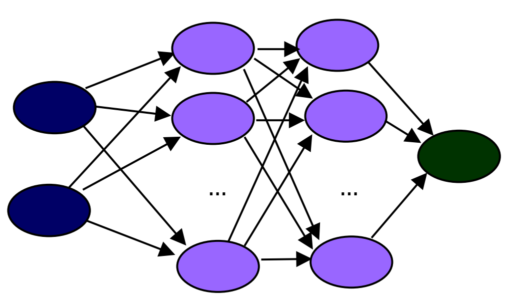


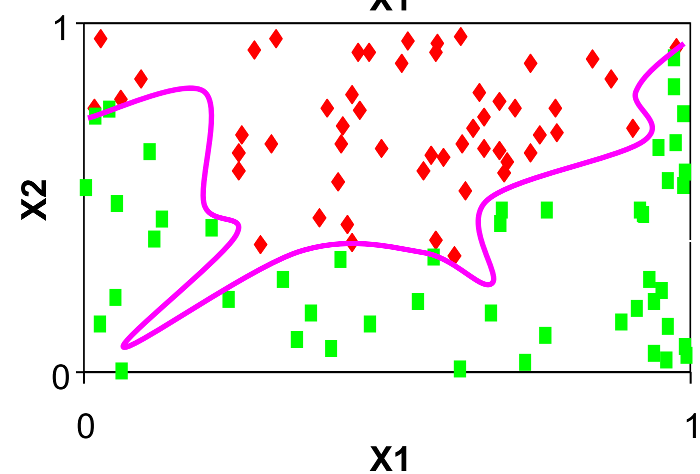
2 +2 neurons





10 +10 neurons



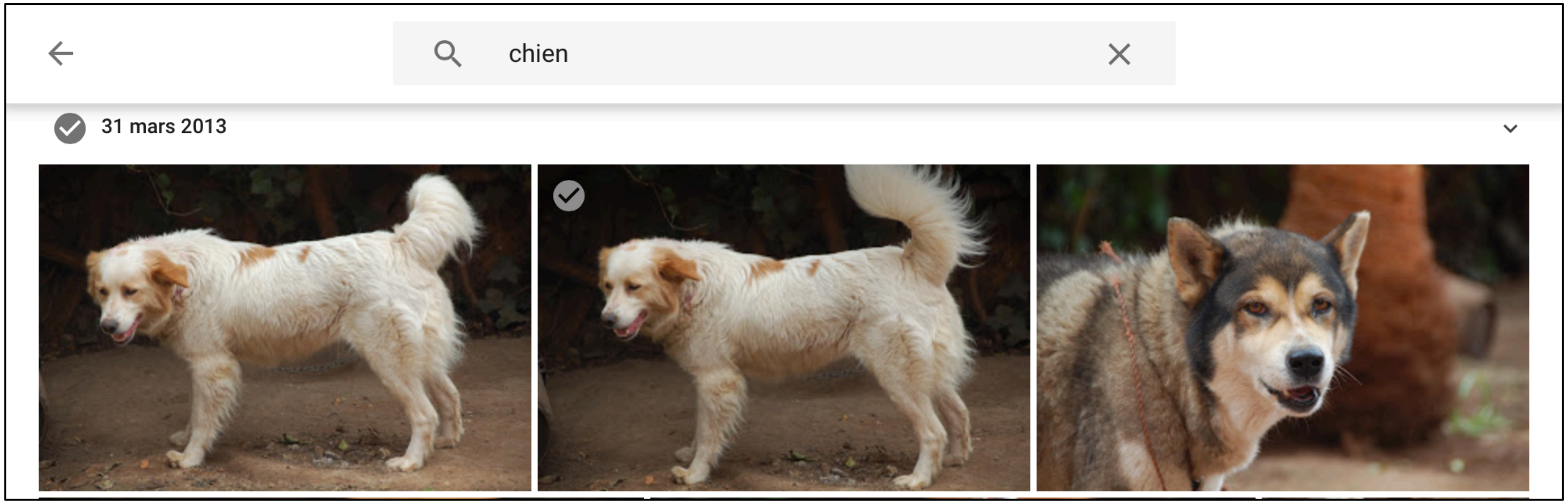


Artificial neural networks

- Advantages:
  - Universal approximators
  - May be very accurate (if the method is well used)
- Drawbacks:
  - The learning phase may be very slow
  - Black-box models, very difficult to interpret
  - Scalability

Application: image labeling

- Goal: Label images with info about their content
- Eg.: Google photos' search engine:



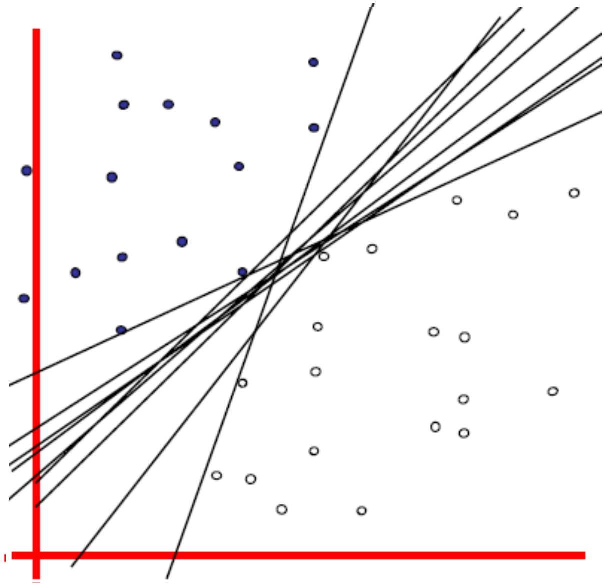
- Most successful methods in this domain are based on huge (*deep*) neural networks. E.g., GoogLeNet:
  - 100 layers (22 with tuned parameters), more than 4M parameters
  - Trained from 1.2M images, with 1000 image categories
  - 6.67% top-5 error rate http://arxiv.org/pdf/1409.4842.pdf

Support vector machines

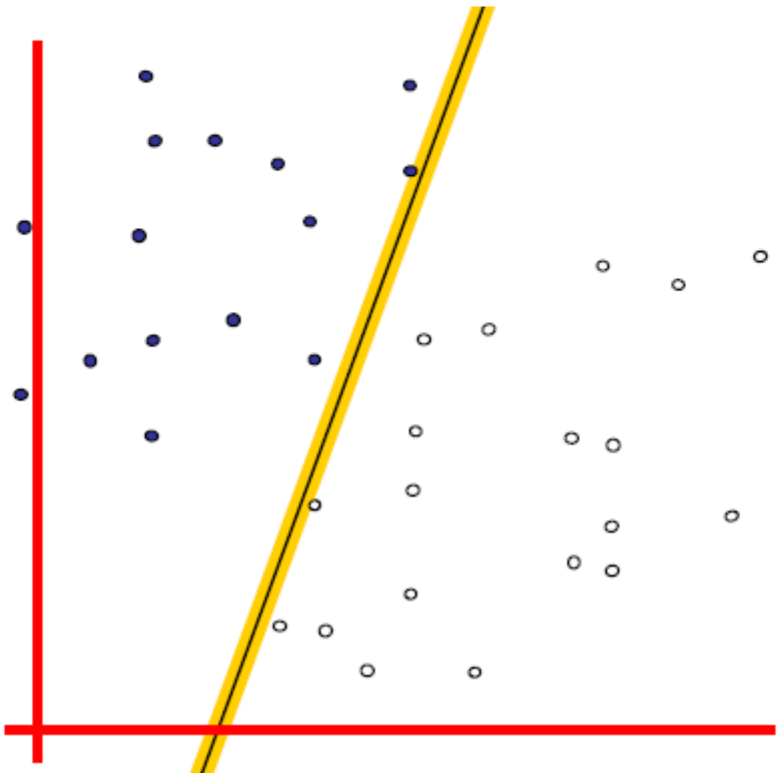
- Recent (mid-90's) and very successful method
- Based on two smart ideas:
  - large margin classifier
  - kernelized input space

Linear classifier

- Where would you place a linear classifier?

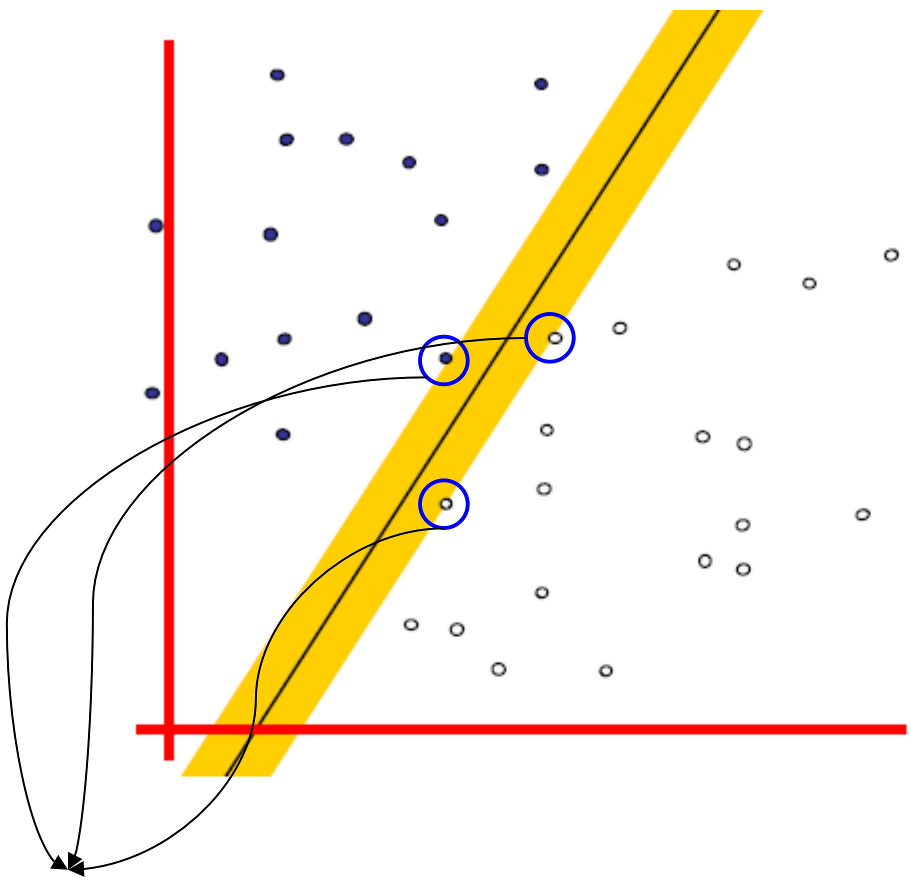


Margin of a linear classifier



- The margin = the width that the boundary could be increased by before hitting a datapoint.

Maximum-margin linear classifier

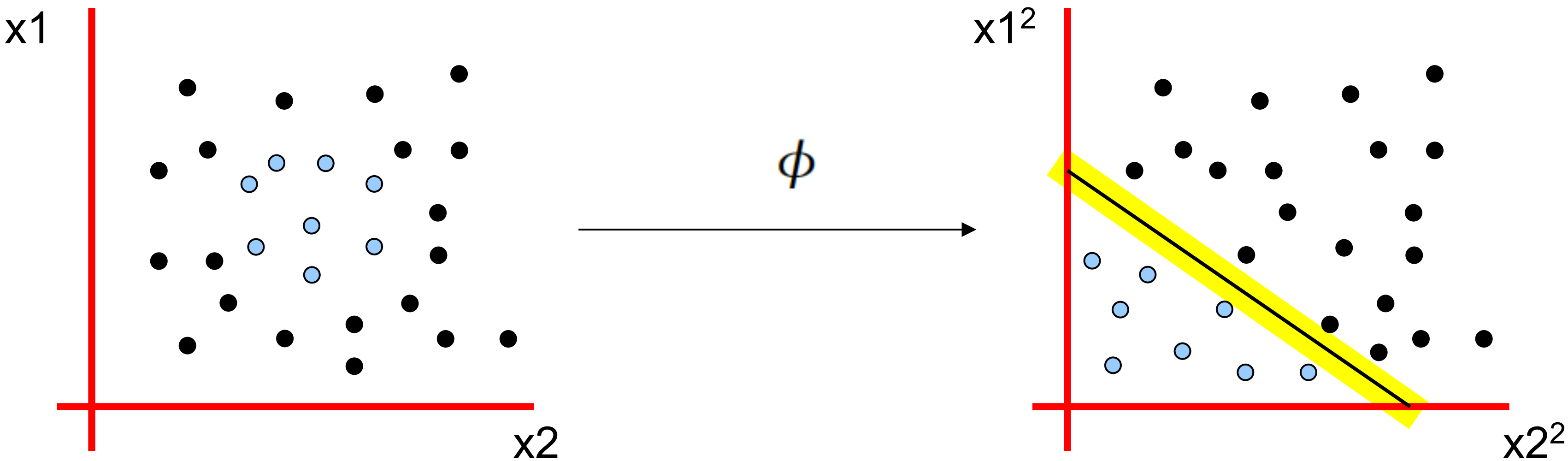


Support vectors: the samples the closest to the hyperplane

- The linear classifier with the maximum margin (= Linear SVM)
- Why ?
  - Intuitively, safest
  - Works very well
  - Theoretical bounds: E(TS)<O(1/margin)
  - Kernel trick

Non-linear boundary

– What about this problem?



- Solution:
  - map the data into a new feature space where the boundary is linear
  - Find the maximum margin model in this new space

The kernel trick

- Intuitively:
  - You don't need to compute explicitly the mapping φ
  - All you need is a (special) similarity measure between objects (like for the kNN)
  - This similarity measure is called a kernel
- Mathematically:
  - The maximum-margin classifier in some feature space can be written only in terms of dot-products in that feature space:

$$k(x,x') = \langle \varphi(x), \varphi(x') \rangle$$

Support vector machines

	X1	X2	Y
1	0.49	0.94	C1
2	0.86	0.59	C2
3	0.6	0.79	C2
4	0.83	0.66	C1
5	0.63	0.27	C1
6	-0.76	0.47	C2

	X1	Y
1	ACGCTCTATAG	C1
2	ACTCGCTTAGA	C2
3	GTCTCTGAGAG	C2
4	CGCTAGCGTCG	C1
5	CGATCAGCAGC	C1
6	GCTCGCGCTCG	C2

kernel matrix

	1	2	3	4	5	6
1	1	0.14	0.96	0.17	0.01	0.24
2	0.14	1	0.02	0.17	0.22	0.67
3	0.96	0.02	1	0.15	0.27	0.07
4	0.17	0.7	0.15	1	0.37	0.55
5	0.01	0.22	0.27	0.37	1	-0.25
6	0.24	0.67	0.07	0.55	-0.25	1

Class labels

	Y
1	C1
2	C2
3	C2
4	C1
5	C1
6	C2

SVM algorithm

Classification model

Support vector machines

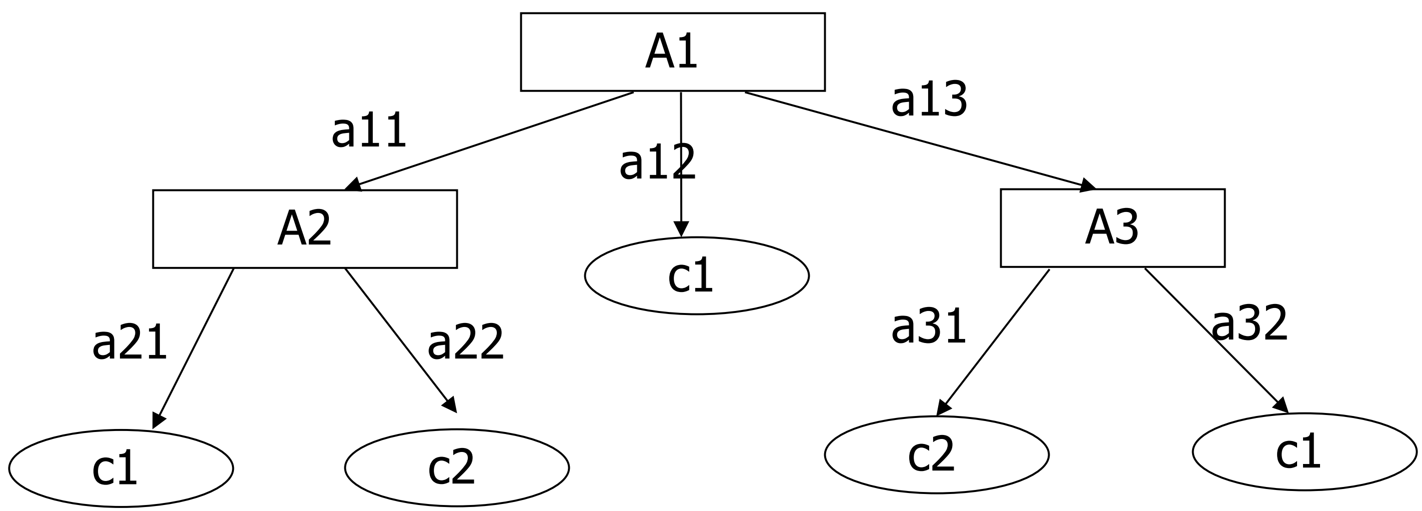
- Advantages:
  - State-of-the-art accuracy on many problems
  - Can handle any data types by changing the kernel (many applications on sequences, texts, graphs...)
- Drawbacks:
  - Tuning the method parameter is very crucial to get good results and somewhat tricky
  - Black-box models, not easy to interpret

Decision (classification) trees

- A learning algorithm that can handle:
  - Classification problems (binary or multi-valued)
  - Attributes may be discrete (binary or multi-valued) or continuous.
- Classification trees were invented at least twice (in mid-80's):
  - By statisticians: CART (Breiman et al.)
  - By the AI community: ID3, C4.5 (Quinlan et al.)

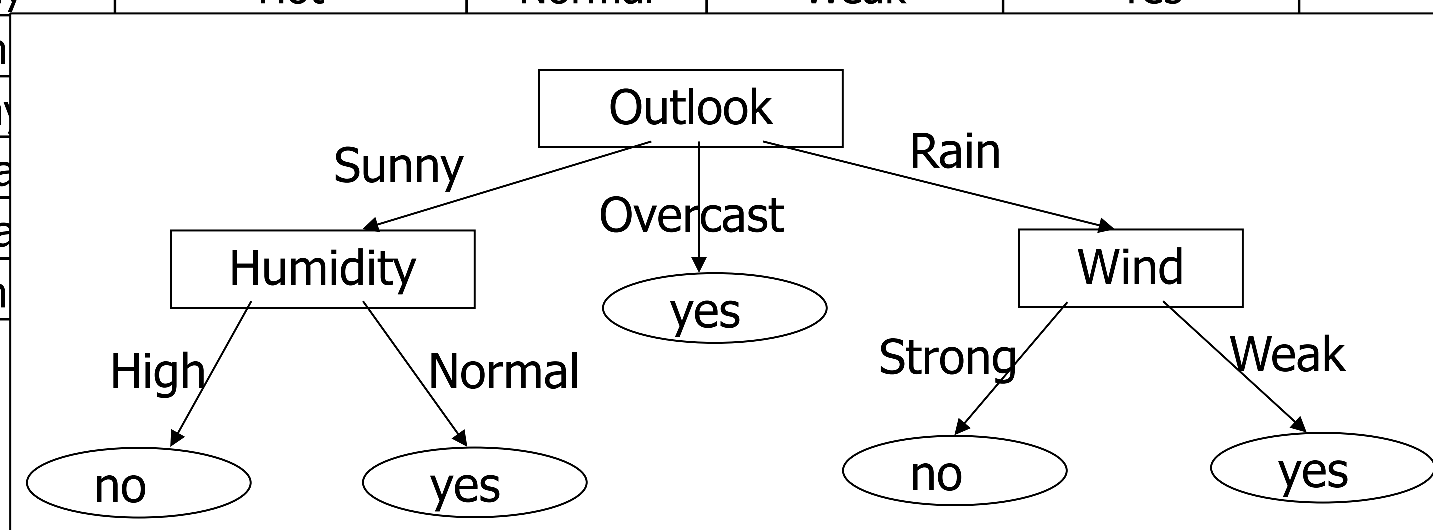
Decision trees

- A decision tree is a tree where:
  - Each interior node tests an attribute
  - Each branch corresponds to an attribute value
  - Each leaf node is labeled with a class



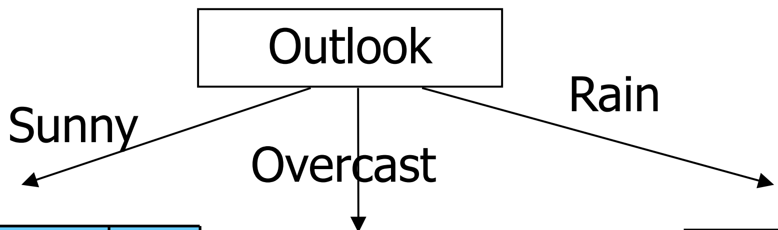
A simple database: playtennis

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	Normal	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	High	Strong	Yes
D8	Sunny	Mild	Normal	Weak	No
D9	Sunny	Hot	Normal	Weak	Yes
D10	Rain				
D11	Sunn				
D12	Overca				
D13	Overca				
D14	Rain				



Top-down induction of DTs

- Choose « best » attribute
- Split the learning sample
- Proceed recursively until each object is correctly classified

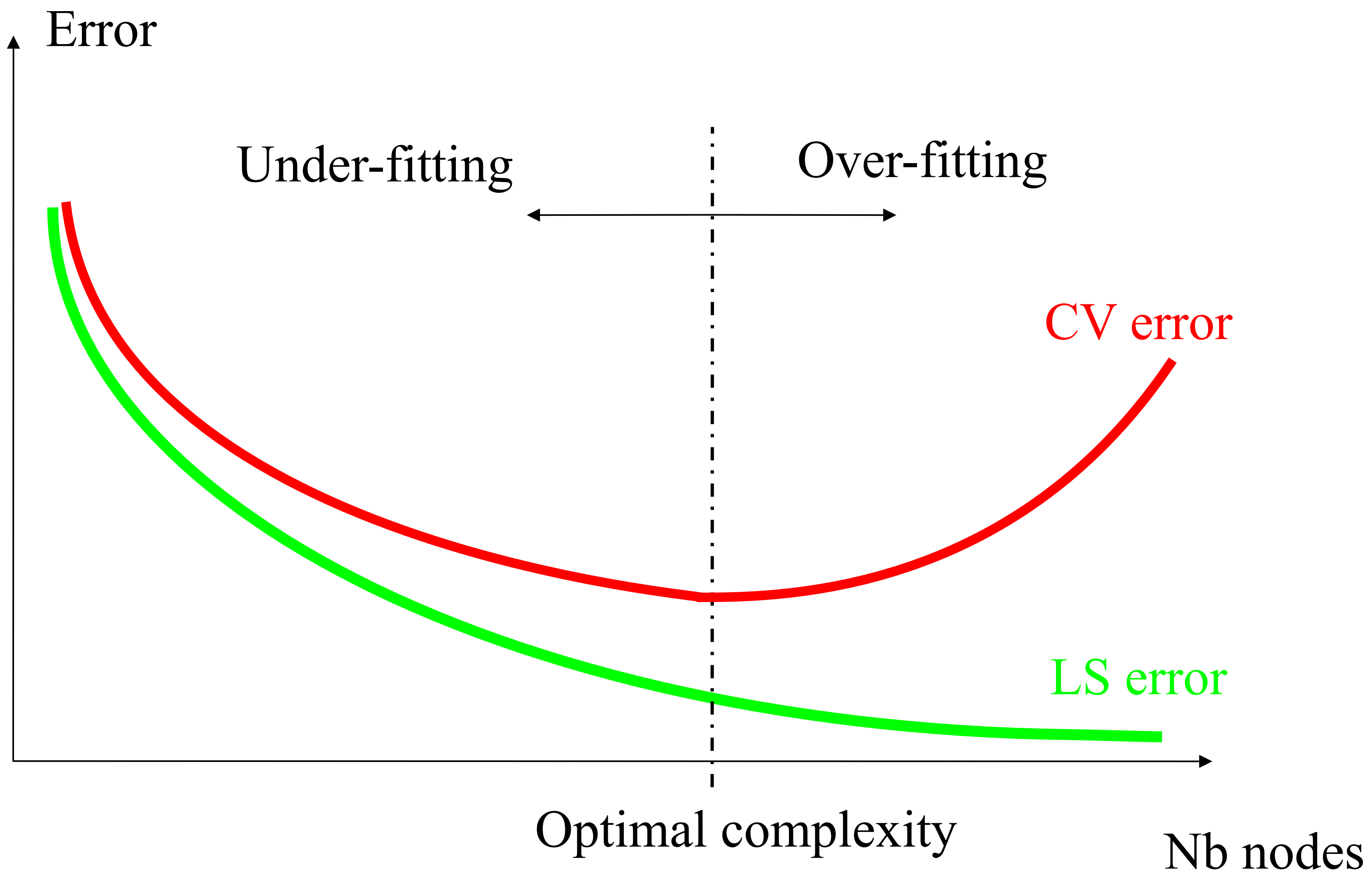


Day	Outlook	Temp.	Humidity	Wind	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Hot	Normal	Wea	
D11	Sunny	Cool	Normal	Stro	

Day	Outlook	Temp.	Humidity	Wind	Play
D3	Overcast	Hot	High	Weak	Yes
D7	Overcast	Cool	High	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes

Day	Outlook	Temp.	Humidity	Wind	Play
D4	Rain	Mild	Normal	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
	Rain	Mild	Normal	Strong	Yes
	Rain	Mild	High	Strong	No

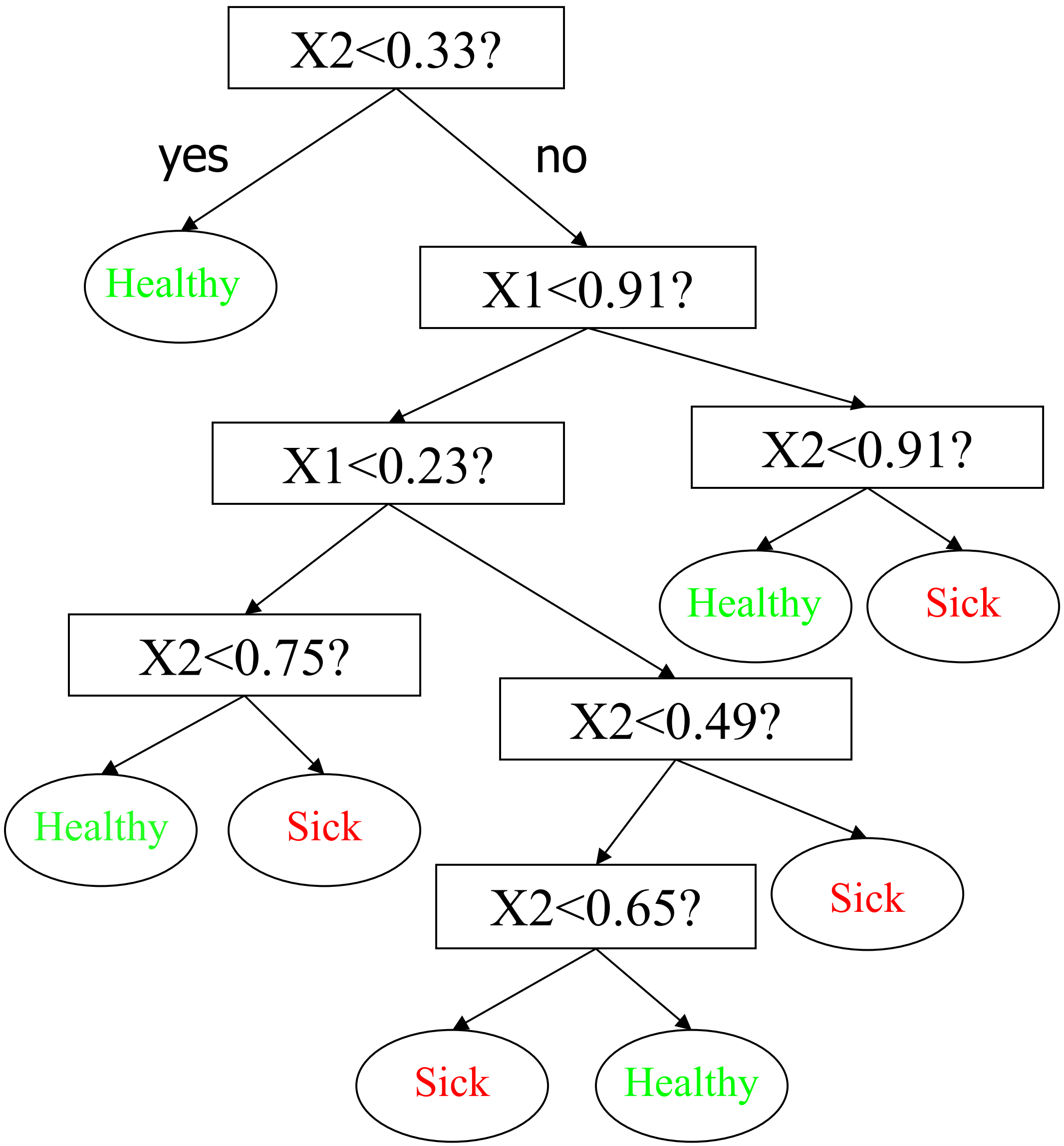
Effect of number of nodes on error

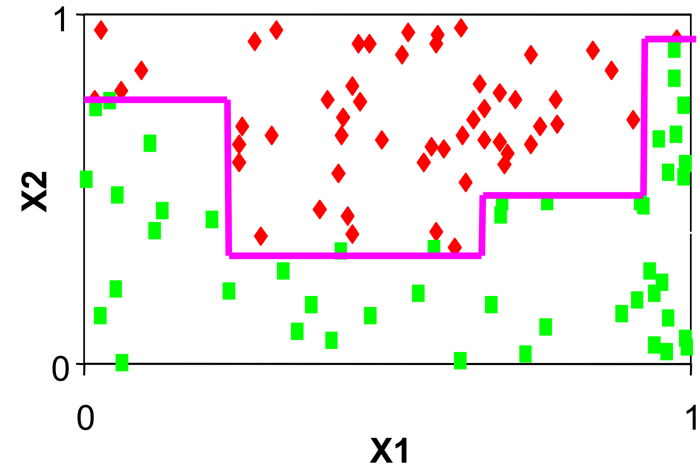


How can we avoid overfitting?

- Pre-pruning: stop growing the tree earlier, before it reaches the point where it perfectly classifies the learning sample
- Post-pruning: allow the tree to overfit and then post-prune the tree
- Ensemble methods (later)

Illustrative example

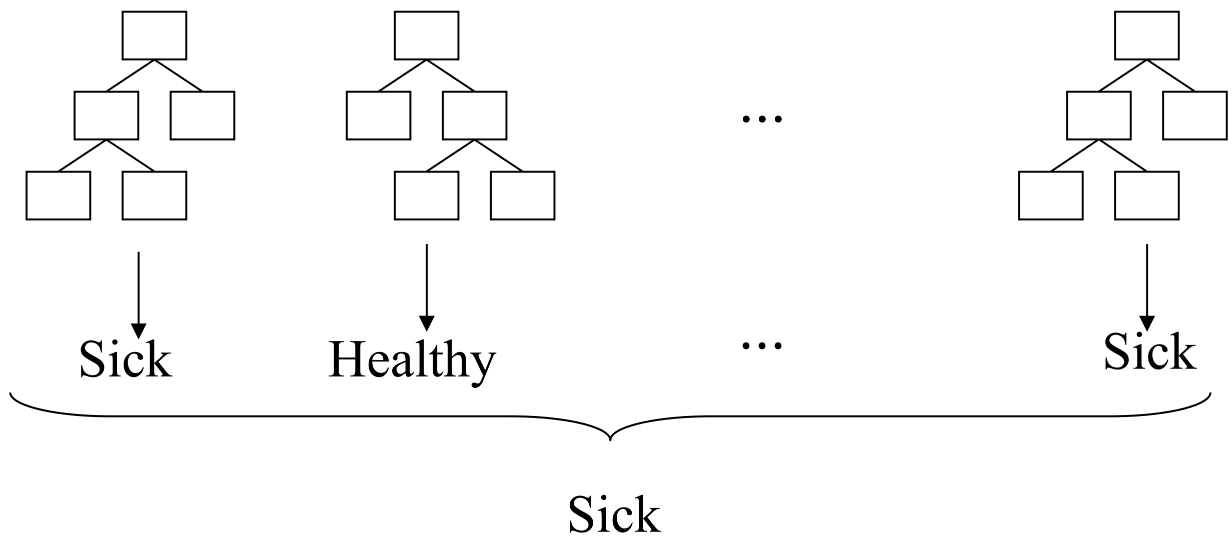




Decision and regression trees

- Advantages:
  - very fast and scalable method (able to handle a very large number of inputs and objects)
  - provide directly interpretable models and give an idea of the relevance of attributes
- Drawbacks:
  - high variance (more on this later)
  - often not as accurate as other methods

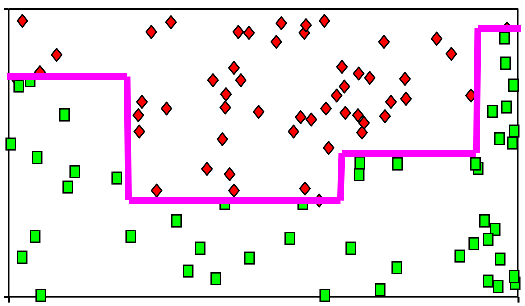
Ensemble methods

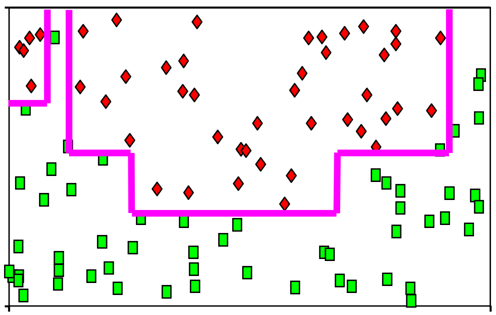


- Combine the predictions of several models built with a learning algorithm. Often improve very much accuracy.
- Often used in combination with decision trees for efficiency reasons
- Examples of algorithms: Bagging, Random Forests, Boosting...

Bagging: motivation

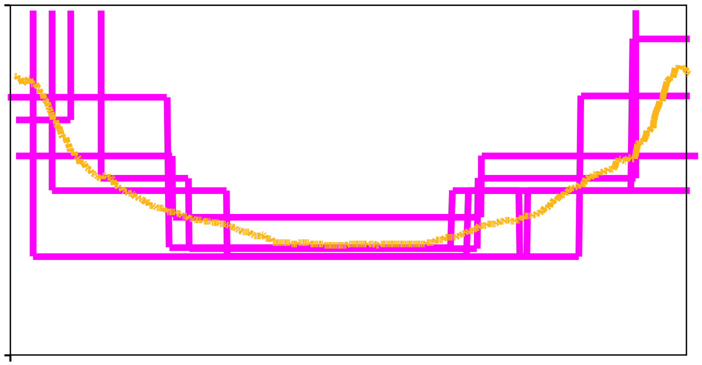
- Different learning samples yield different models, especially when the learning algorithm overfits the data





As there is only one optimal model, this *variance* is source of error

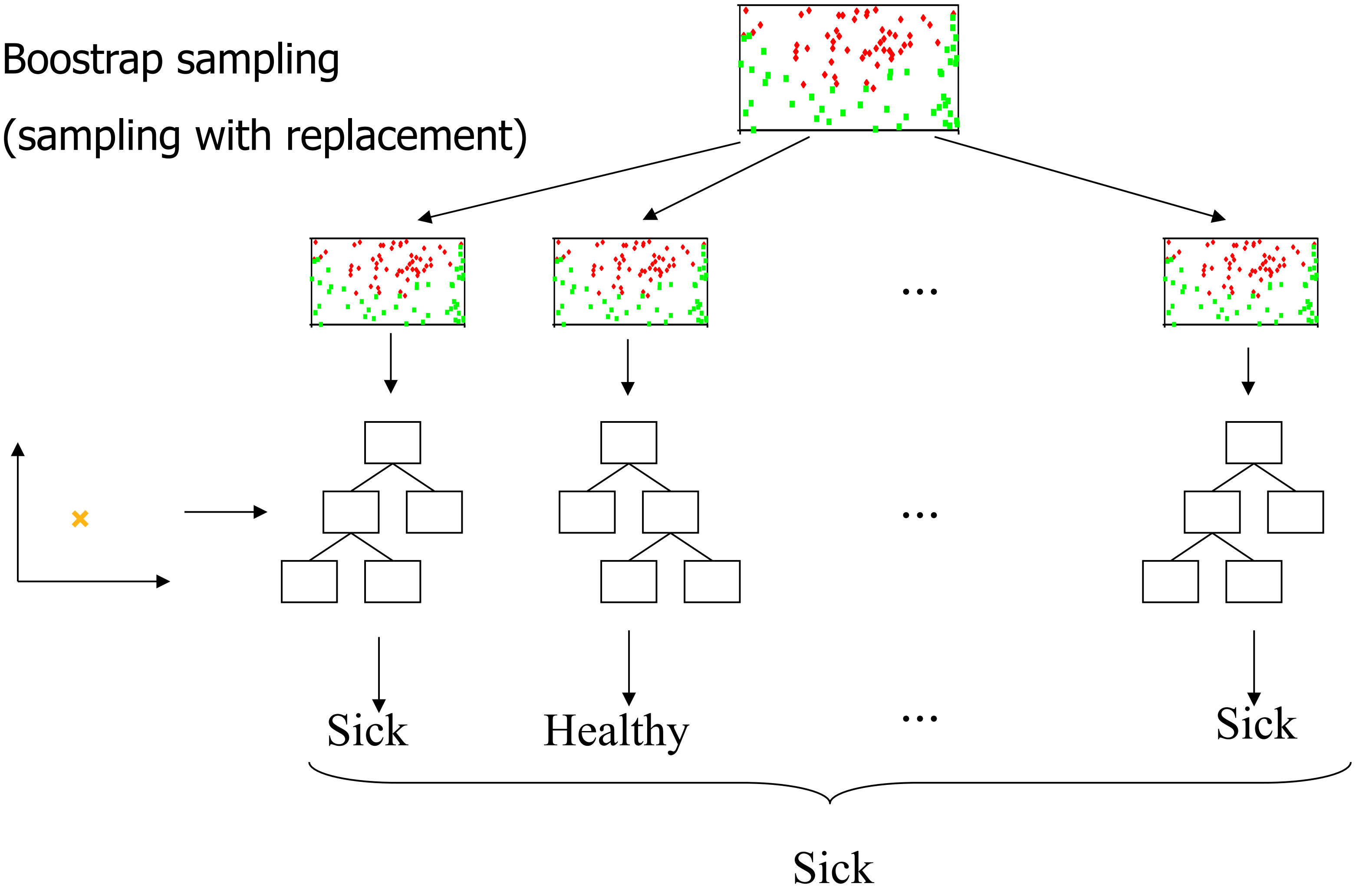
- Solution: aggregate several models to obtain a more stable one



Bagging: bootstrap aggregating

Boostrap sampling

(sampling with replacement)



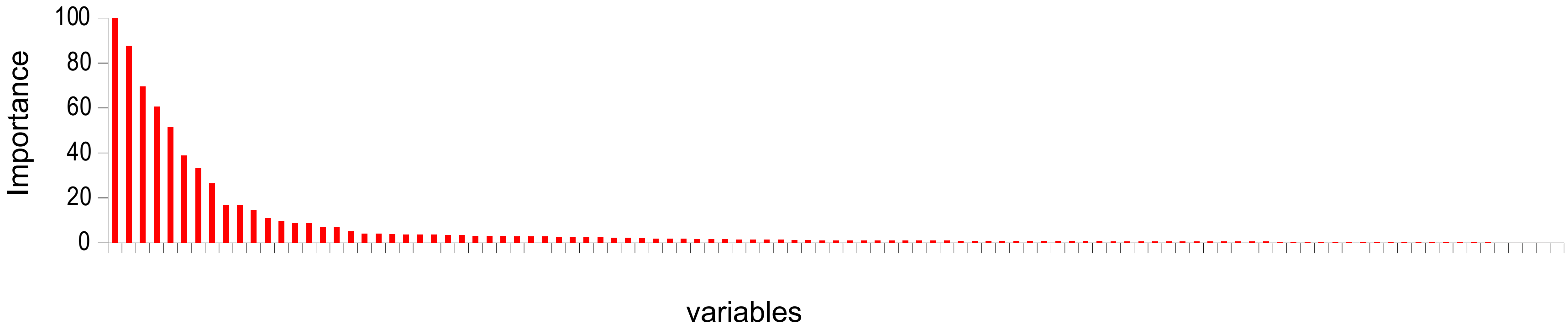
Note: the more models, the better.

Example on microarray data

- 72 patients, 7129 gene expressions, 2 classes of Leukemia (ALL and AML) (Golub et al., Science, 1999)
- Leave-one-out error with several variants

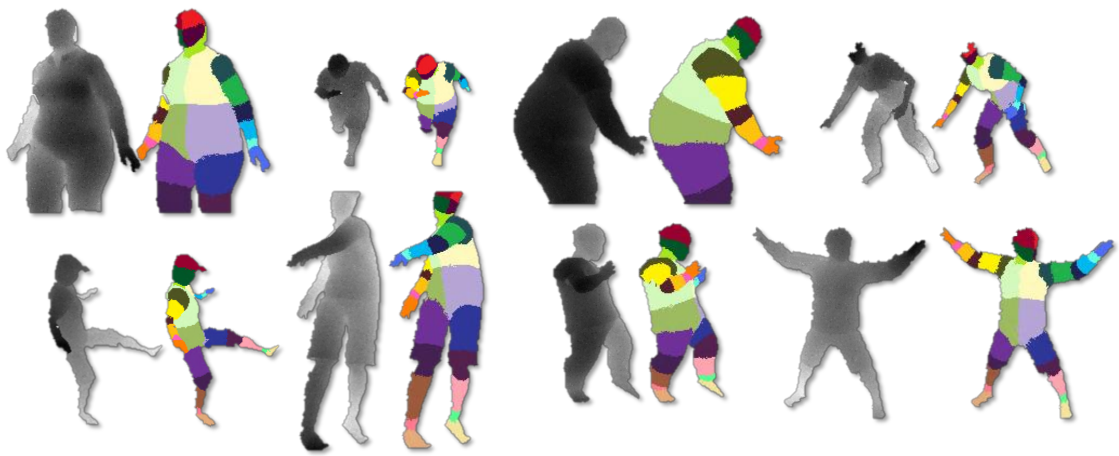
Method	Error
1 decision tree	22.2% (16/72)
Random forests (k=85,T=500)	9.7% (7/72)
Extra-trees (s_{th}=0.5, T=500)	5.5% (4/72)
Adaboost (1 test node, T=500)	1.4% (1/72)

- Variable importance with boosting



Application: Kinect

- Ensemble of randomized decision trees are used in Microsoft's Xbox Kinect for body part labeling:



- Each sample corresponds to a single pixel and is described by depth differences between neighbor pixels
- Final model is implemented on GPU to get very fast predictions (200 frames per second)

http://research.microsoft.com/pubs/145347/BodyPartRecognition.pdf

Method comparison

Method	Accuracy	Efficiency	Interpretability	Ease of use
kNN	++	+	++	++
DT	+	+++	+++	+++
Linear	++	++++	++	+++
Ensemble	+++	+++	++	+++
ANN	++++	++	+	++
SVM	+++	+	+	+

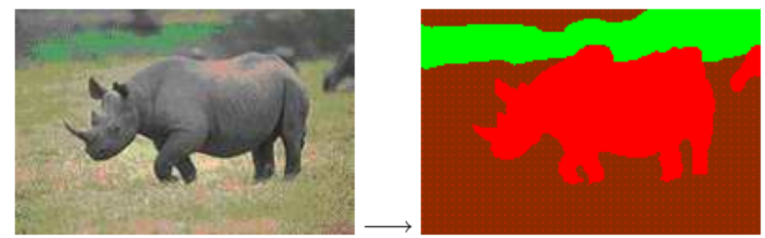
- Note:
  - The relative importance of the criteria depends on the specific application
  - These are only general trends. Eg., in terms of accuracy, no algorithm is always better or worse than all others.

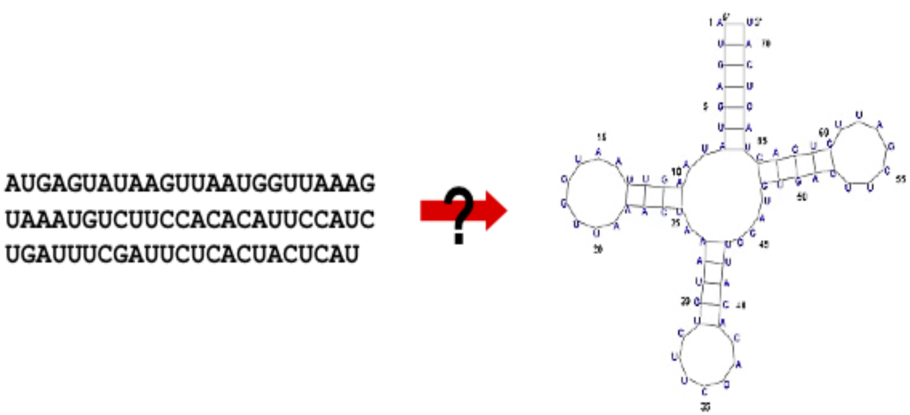
Outline

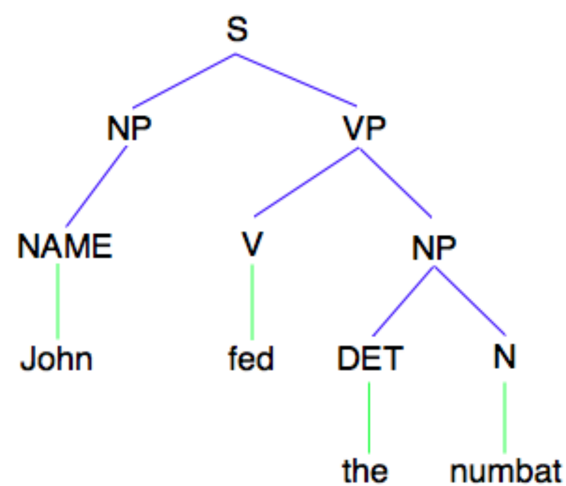
- Introduction
- Supervised Learning
  - Introduction
  - Model selection, cross-validation, overfitting
  - Some supervised learning algorithms
  - Beyond classification and regression
- Other learning protocols/frameworks

Beyond classification and regression

- All supervised learning problems can not be turned into standard classification or regression problems
- Examples:
  - Graph predictions
  - Sequence labeling
  - image segmentation







Structured output approaches

- Decomposition:
  - Reduce the problem to several simpler classification or regression problems by decomposing the output
  - Not always possible and does not take into account dependencies between sub-outputs
- Multiple/kernel output methods:
  - Extend regression methods to handle an output space endowed with a kernel
  - This can be done with regression trees or ridge regression for example
- Large margin methods
  - Use SVM-based approaches to learn a model that scores directly input-output pairs:

$$y = arg\,max_{y'} \sum_i w_i \phi_i(x, y')$$

Outline

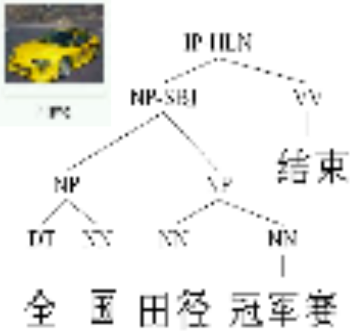
- Introduction
- Supervised learning
- Other learning protocols/frameworks
  - On-line versus batch learning
  - Semi-supervised learning
  - Transductive learning
  - Active learning
  - Reinforcement learning
  - Unsupervised learning

On-line versus batch learning

- **Batch learning**: all learning examples are supposed to be accessible (in memory) for training the model
- **On-line learning**: examples are treated one by one (mini-batch learning: examples are treated in blocs).
- Applications:
  - Big data: training set is too large to be stored in memory
  - Concept drift: allows to continuously adapt the model to dynamical change of the system
  - Learning with streaming data
- There exist online implementations of most machine learning methods

Labeled versus unlabeled data

- Unlabeled data=input-output pairs without output value
- In many settings, unlabeled data is cheap but labeled data can be hard to get
  - labels may require human experts
  - human annotation is expensive, slow, unreliable
  - labels may require special devices
- Examples:
  - Biomedical domain
  - Speech analysis
  - Natural language parsing
  - Image categorization/segmentation
  - Network measurement

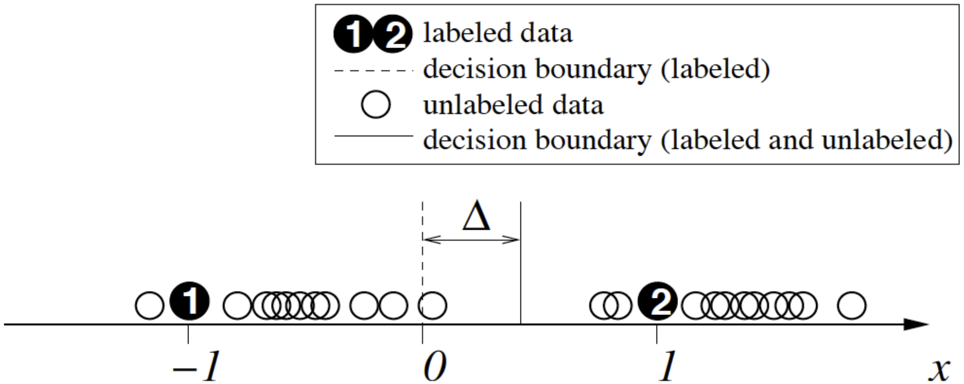


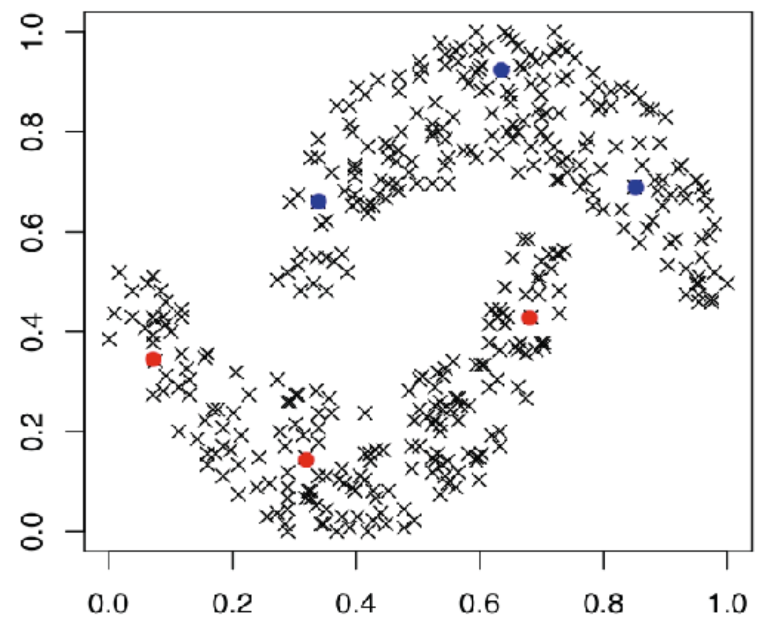
Semi-supervised learning

- Goal: exploit both labeled and unlabeled data to build better models than using each one alone

	A1	A2	A3	A4	Y
labeled data	0.01	0.37	T	0.54	Healthy
	-2.3	-1.2	F	0.37	Disease
	0.69	-0.78	F	0.63	Healthy
unlabeled data	-0.56	-0.89	T	-0.42	
	-0.85	0.62	F	-0.05	
	-0.17	0.09	T	0.29	
test data	-0.09	0.3	F	0.17	?

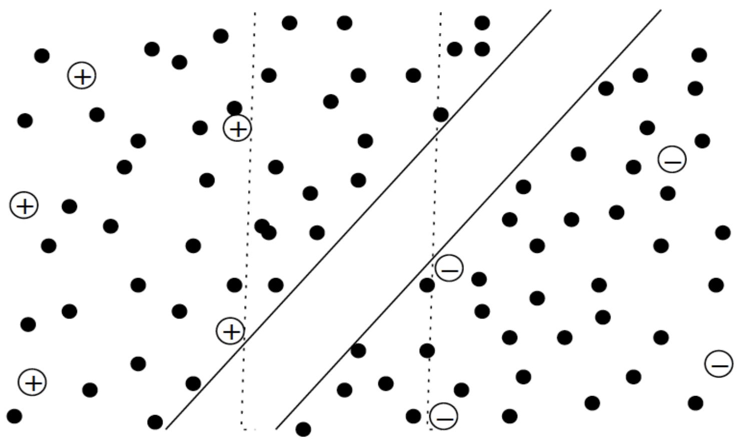
- Why would it improve?





Some approaches

- Self-training
  - Iteratively label some unlabeled examples with a model learned from the previously labeled examples
- Semi-supervised SVM (S3VM)
  - Enumerate all possible labeling of the unlabeled examples
  - Learn an SVM for each labeling
  - Pick the one with the largest margin

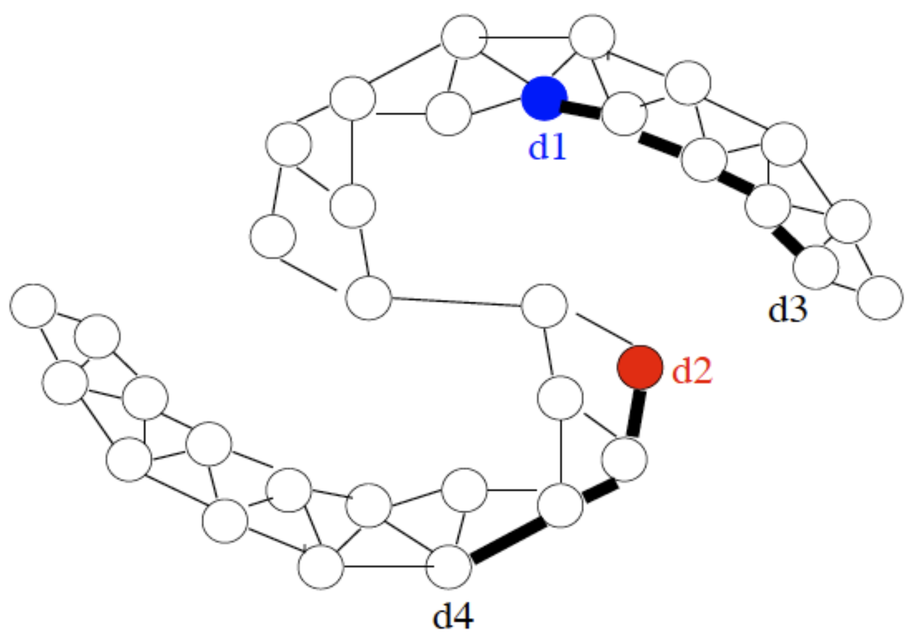


Transductive learning

- Like supervised learning but we have access to the test data from the beginning and we want to exploit it
- We don't want a model, only compute predictions for the unlabeled data
- Simple solution:
  - Apply semi-supervised learning techniques using the test data as unlabeled data to get a model
  - Use the resulting model to make predictions on the test data
- There exist also specific algorithms that avoid building a model

Some approaches

- Graph-based algorithms
  - Build a graph over the (labeled and unlabeled) examples (from the inputs)
  - Learn a model that predicts well labeled examples and is smooth over the graph

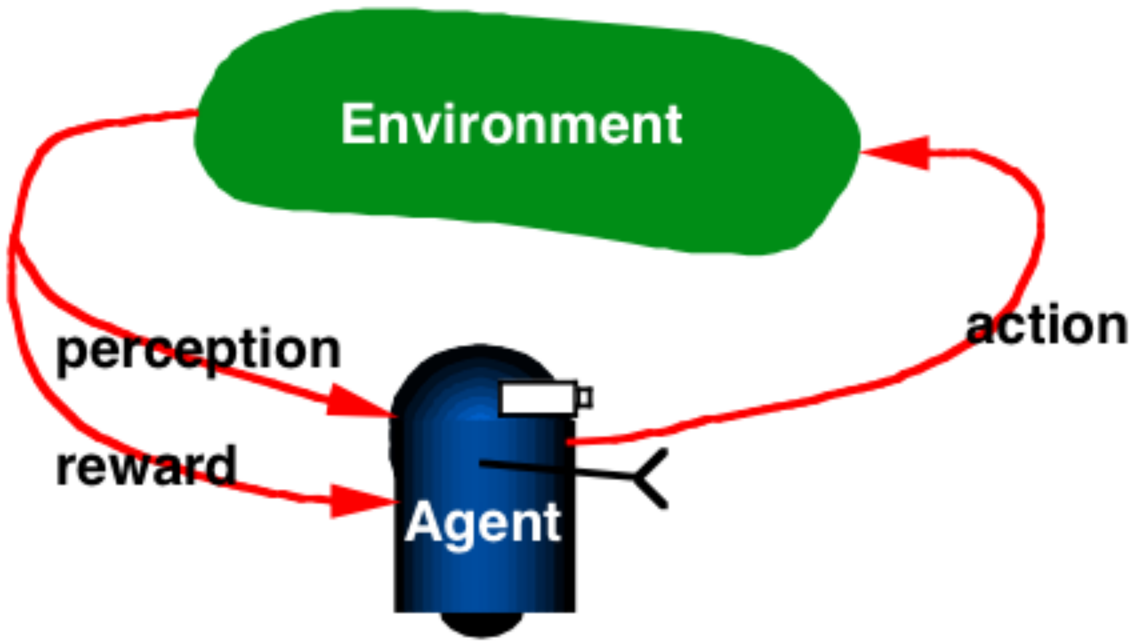


Active learning

- Goal:
  - Given unlabeled data, find (adaptively) the examples to label in order to learn an accurate model
  - The hope is to reduce the number of labeled instances with respect to the standard batch SL
- Usually, in an online setting:
  - Choose the k "*best*" unlabeled examples
  - Determine their labels
  - Update the model and iterate
- Algorithms differ in the way the "*best*" unlabeled examples are selected
  - Example: choose the k examples for which the model predictions are the most uncertain

Reinforcement learning (see INFO8003)

Learning from interactions



$$s_0 \xrightarrow[r_0]{a_0} s_1 \xrightarrow[r_1]{a_1} s_2 \xrightarrow[r_2]{a_2} \ldots$$

Goal: learn to choose sequence of actions (= policy) that maximizes $r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots$, where $0 \leqslant \gamma < 1$

RL approaches

- System is usually modeled by
  - state transition probabilities $P(s_{t+1}|s_t, a_t)$
  - reward probabilities $P(r_{t+1}|s_t, a_t)$

(= Markov Decision Process)

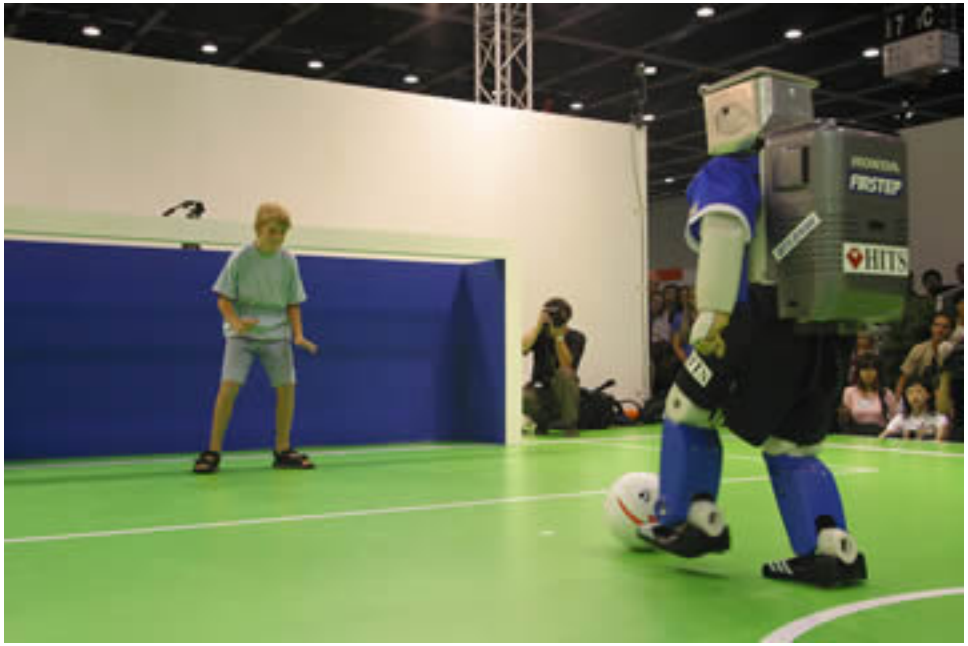
- Model of the dynamics and reward is known $\Rightarrow$ try to compute optimal policy by dynamic programming
- Model is unknown
  - Model-based approaches $\Rightarrow$ first learn a model of the dynamics and then derive an optimal policy from it (DP)
  - Model-free approaches $\Rightarrow$ learn directly a policy from the observed system trajectories

Examples of applications

- Robocup Soccer Teams (Stone & Veloso, Riedmiller et al.)
- Inventory Management (Van Roy, Bertsekas, Lee &Tsitsiklis)
- Dynamic Channel Assignment, Routing (Singh & Bertsekas, Nie & Haykin, Boyan & Littman)
- Elevator Control (Crites & Barto)
- Many Robots: navigation, bi-pedal walking, grasping, switching between skills...
- Games: TD-Gammon and Jellyfish (Tesauro, Dahl), GO, video games...

Robocup

- Goal: by the year 2050, develop a team of fully autonomous humanoid robots that can win against the human world soccer champion team.



http://www.robocup.org
http://www.youtube.com/watch?v=xqWELifR2P0

Autonomous helicopter

http://heli.stanford.edu/

AlphaGo (Google DeepMind)

- Supervised learning from strong amateur games using (deep) neural networks (to predict next move and game winner)
- Improvement using reinforcement learning by playing against versions of itself

https://storage.googleapis.com/deepmind-media/alphago/AlphaGoNaturePaper.pdf

Unsupervised learning

- Unsupervised learning tries to find any regularities in the data without guidance about inputs and outputs

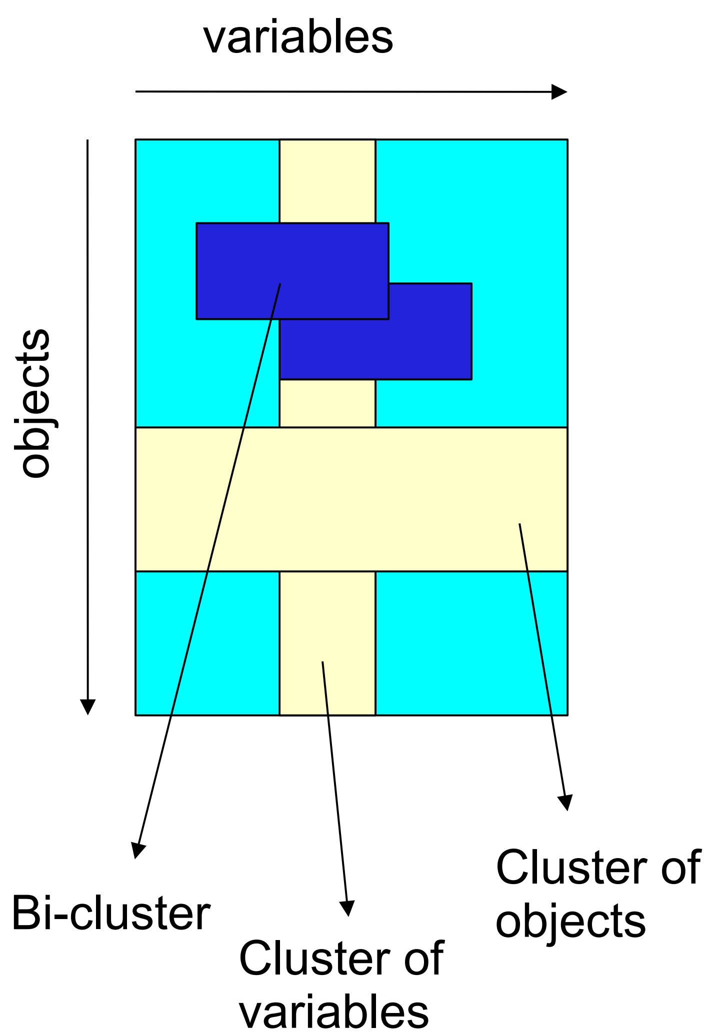
A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19
-0.27	-0.15	-0.14	0.91	-0.17	0.26	-0.48	-0.1	-0.53	-0.65	0.23	0.22	0.98	0.57	0.02	-0.55	-0.32	0.28	-0.33
-2.3	-1.2	-4.5	-0.01	-0.83	0.66	0.55	0.27	-0.65	0.39	-1.3	-0.2	-3.5	0.4	0.21	-0.87	0.64	0.6	-0.29
0.41	0.77	-0.44	0	0.03	-0.82	0.17	0.54	-0.04	0.6	0.41	0.66	-0.27	-0.86	-0.92	0	0.48	0.74	0.49
0.28	-0.71	-0.82	0.27	-0.21	-0.9	0.61	-0.57	0.44	0.21	0.97	-0.27	0.74	0.2	-0.16	0.7	0.79	0.59	-0.33
-0.28	0.48	0.79	-0.14	0.8	0.28	0.75	0.26	0.3	-0.78	-0.72	0.94	-0.78	0.48	0.26	0.83	-0.88	-0.59	0.71
0.01	0.36	0.03	0.03	0.59	-0.5	0.4	-0.88	-0.53	0.95	0.15	0.31	0.06	0.37	0.66	-0.34	0.79	-0.12	0.49
-0.53	-0.8	-0.64	-0.93	-0.51	0.28	0.25	0.01	-0.94	0.96	0.25	-0.12	0.27	-0.72	-0.77	-0.31	0.44	0.58	-0.86
0.04	0.94	-0.92	-0.38	-0.07	0.98	0.1	0.19	-0.57	-0.69	-0.23	0.05	0.13	-0.28	0.98	-0.08	-0.3	-0.84	0.47
-0.88	-0.73	-0.4	0.58	0.24	0.08	-0.2	0.42	-0.61	-0.13	-0.47	-0.36	-0.37	0.95	-0.31	0.25	0.55	0.52	-0.66
-0.56	0.97	-0.93	0.91	0.36	-0.14	-0.9	0.65	0.41	-0.12	0.35	0.21	0.22	0.73	0.68	-0.65	-0.4	0.91	-0.64

- Are there interesting groups of variables or samples? outliers? What are the dependencies between variables?

Unsupervised learning methods

- Many families of tasks/problems exist, among which:
  - Clustering: try to find natural groups of samples/variables
    - eg: k-means, hierarchical clustering
  - Dimensionality reduction: project the data from a high-dimensional space down to a small number of dimensions
    - eg: principal/independent component analysis, MDS
  - Density estimation: determine the distribution of data within the input space
    - eg: bayesian networks, mixture models.

Clustering



- Clustering rows

grouping similar objects

- Clustering columns

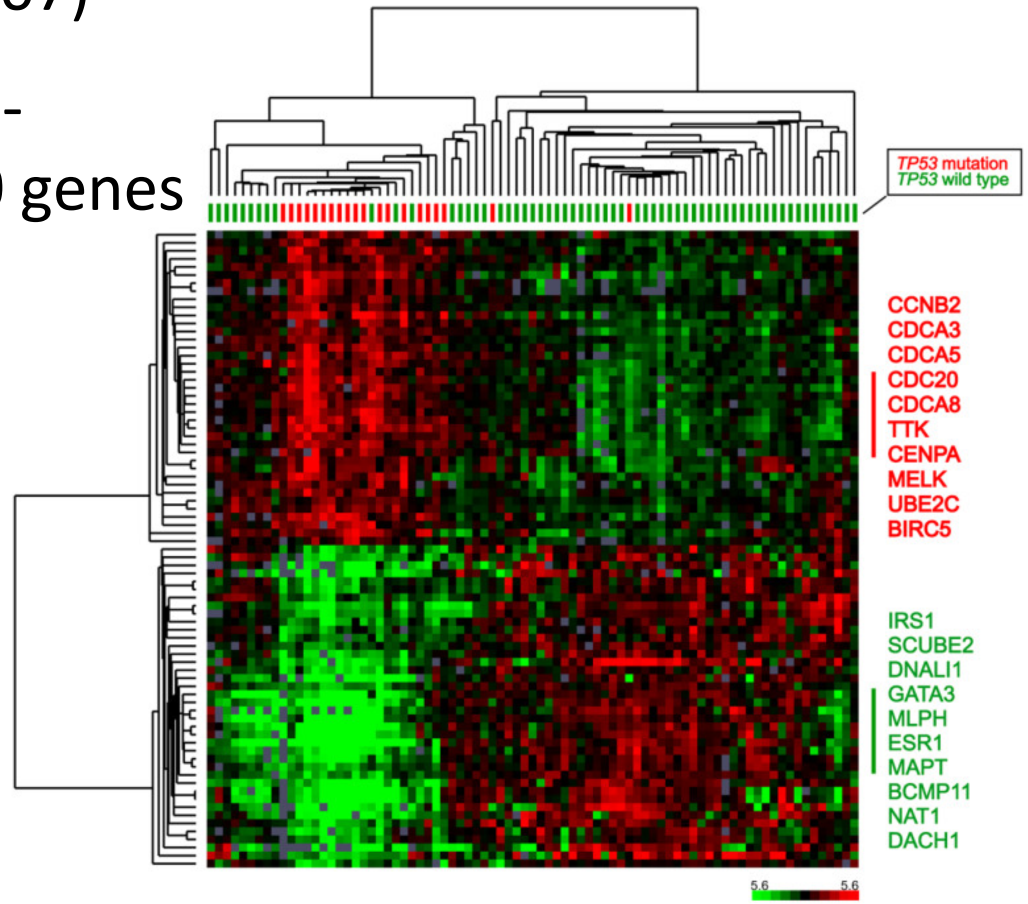
grouping similar variables across samples

- Bi-Clustering/Two-way clustering

grouping objects that are similar across a subset of variables

Illustrations (1)

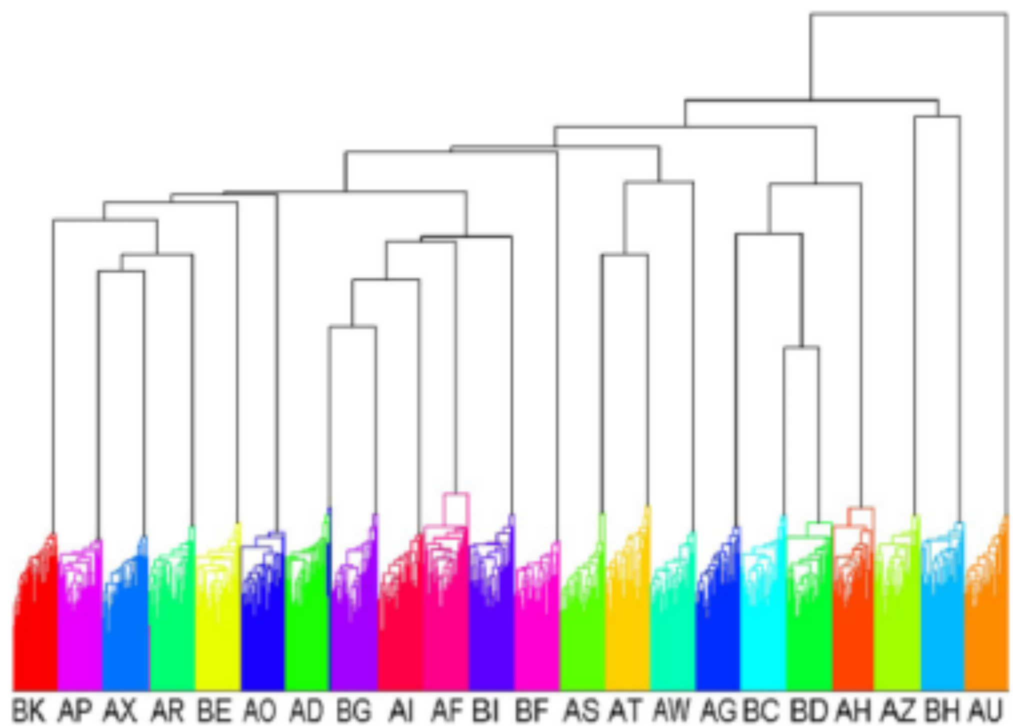
- Breast cancer data (Langerød et al., Breast cancer, 2007)
- 80 tumor samples (wild-type,TP53 mutated), 80 genes



Illustrations (2)

Assfalg et al., *PNAS*, Jan 2008

- Evidence of different metabolic phenotypes in humans
- Urine samples of 22 volunteers over 3 months, NMR spectra analysed by HCA



Application: vector quantization

FIGURE 14.9. *Sir Ronald A. Fisher (1890-1962) was one of the founders of modern day statistics, to whom we owe maximum-likelihood, sufficiency, and many other fundamental concepts. The image on the left is a* 1024×1024 *grayscale image at* 8 *bits per pixel. The center image is the result of* 2×2 *block VQ, using* 200 *code vectors, with a compression rate of* 1.9 *bits/pixel. The right image uses only four code vectors, with a compression rate of* 0.50 *bits/pixel*

Principal Component Analysis

- An exploratory technique used to reduce the dimensionality of the data set to a smaller space (2D, 3D)

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
0.25	0.93	0.04	-0.78	-0.53	0.57	0.19	0.29	0.37	-0.22
-2.3	-1.2	-4.5	-0.51	-0.76	0.07	0.81	0.95	0.99	0.26
-0.29	-1	0.73	-0.33	0.52	0.13	0.13	0.53	-0.5	-0.48
-0.16	-0.17	-0.26	0.32	-0.08	-0.38	-0.48	0.99	-0.95	0.34
0.07	-0.87	0.39	0.5	-0.63	-0.53	0.79	0.88	0.74	-0.14
0.61	0.15	0.68	-0.94	0.5	0.06	-0.56	0.49	0	-0.77

→

PC1	PC2
0.36	0.1
-2.3	-1.2
0.27	-0.89
-0.19	0.7
-0.77	-0.7
-0.65	-0.99

- Transform some large number of variables into a smaller number of uncorrelated variables called principal components (PCs)

Objectives of PCA

- Reduce dimensionality (pre-processing for other methods)
- Choose the most useful (informative) variables
- Compress the data
- Visualize multidimensional data
  - to identify groups of objects
  - to identify outliers

Illustration (1)

Holmes et al., *Nature, Vol. 453, No. 15, May 2008*

- Investigation of metabolic phenotype variation across and within four human populations (17 cities from 4 countries: China, Japan, UK, USA)
- ^{1}H NMR spectra of urine specimens from 4630 participants
- PCA plots of median spectra per population (city) and gender

Holmes et al., Nature, *Vol. 453, No. 15, May 2008*

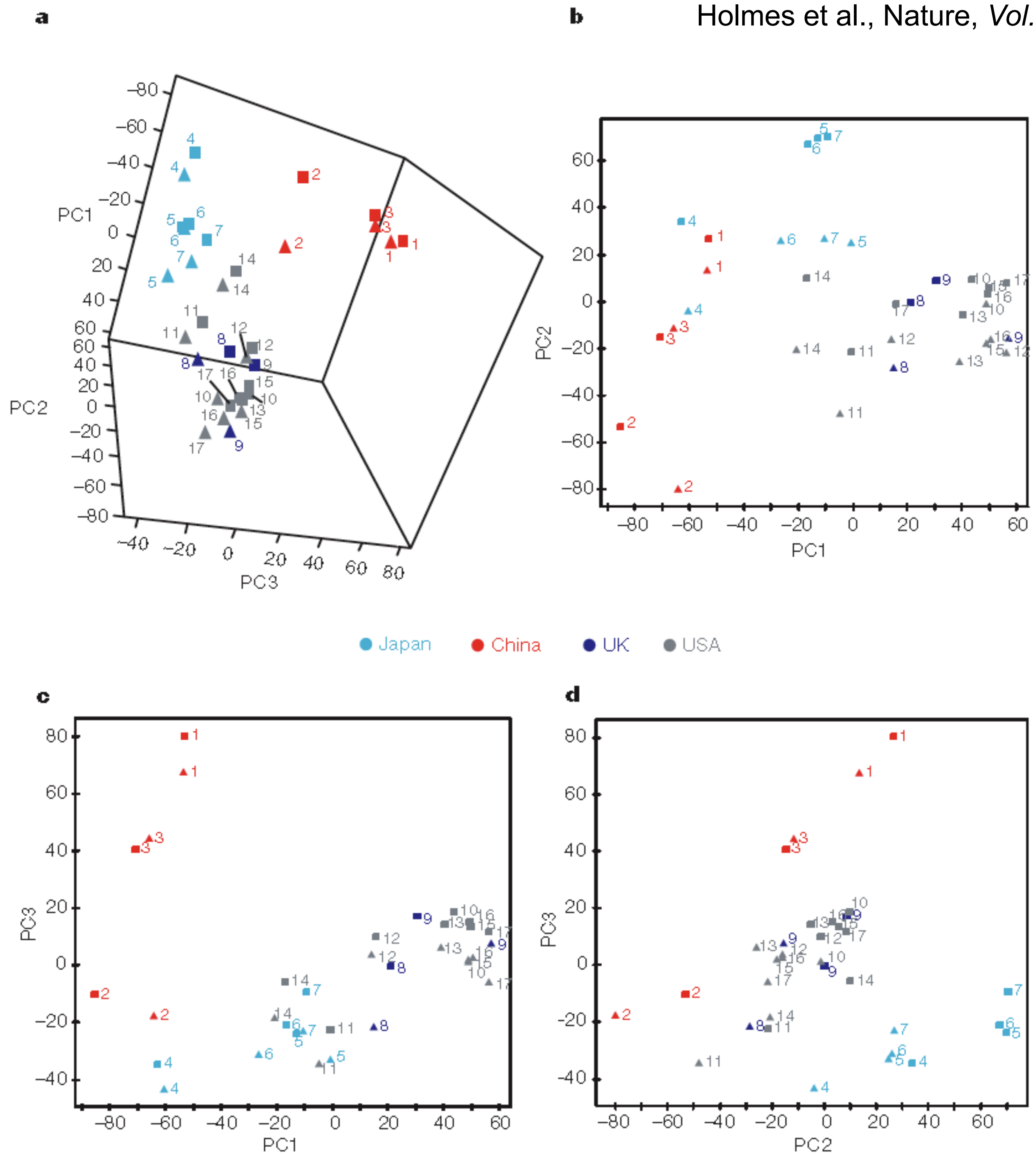
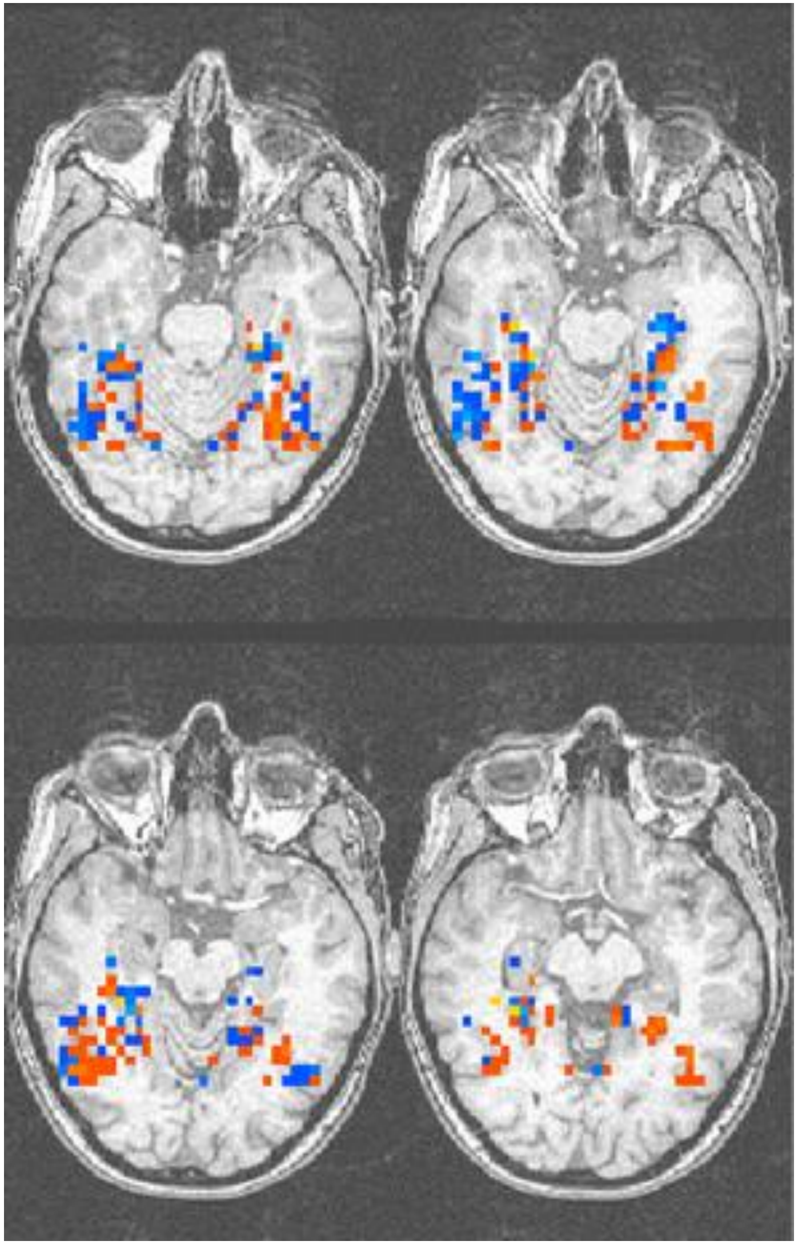


Illustration (2)
Neuroimaging

L voxels (brain regions)

A1	A2	A3	A4	A5	...	A7	A8
-0.91	0.74	0.74	0.97	-0.06	...	-0.04	-0.73
-2.3	-1.2	-4.5	0.47	0.13	...	0.16	0.26
-0.98	-0.46	0.98	0.77	-0.14	...	0.44	-0.12
0.97	-0.64	-0.3	-0.14	-0.29	...	-0.43	0.27
-0.64	-0.34	0.21	-0.57	-0.39	...	0.02	-0.61
0.41	-0.95	0.21	-0.17	-0.68	...	0.11	0.49

N patients/brain maps



Books

- Reference book for the course:
  - *The elements of statistical learning: data mining, inference, and prediction*. T. Hastie et al, Springer, 2001 (second edition in 2009)
  - Freely downloadable here:
    - http://statweb.stanford.edu/~tibs/ElemStatLearn/
- Other textbooks
  - *Machine Learning*. Tom Mitchell, McGraw Hill, 1997.
  - *Pattern classification* (2nd edition). R.Duda, P.Hart, D.Stork, Wiley Interscience, 2000
  - *Pattern Recognition and Machine Learning (Information Science and Statistics)*. C.M.Bishop, Springer, 2004
  - *Introduction to Machine Learning*. Ethan Alpaydin, MIT Press, 2004.
  - *Machine Learning: The Art and Science of Algorithms that Make Sense of Data.* Peter Flach. Cambridge University Press, 2012.
  - *Machine Learning: a probabilistic perspective.* Kevin P. Murphy. MIT Press, 2012.

Books

- More advanced/specific topics
  - *kernel methods for pattern analysis.* J. Shawe-Taylor and N. Cristianini. Cambridge University Press, 2004
  - *Reinforcement Learning: An Introduction.* R.S. Sutton and A.G. Barto. MIT Press, 1998
  - *Neuro-Dynamic Programming.* D.P Bertsekas and J.N. Tsitsiklis. Athena Scientific, 1996
  - *Semi-supervised learning.* Chapelle et al., MIT Press, 2006
  - *Predicting structured data.* G. Bakir et al., MIT Press, 2007
  - *Deep learning*. Goodfellow, Bengio, Courville, MIT Press, 2016

Generic ML toolboxes

- scikit-learn
  - scikit-learn.org
  - Open source machine learning toolbox (in Python)
- WEKA
  - http://www.cs.waikato.ac.nz/ml/weka/
  - Open source machine learning toolbox (in Java)
- Many R and Matlab packages
  - http://www.kyb.mpg.de/bs/people/spider/
  - http://www.cs.ubc.ca/~murphyk/Software/BNT/bnt.html
  - …

Journals

- Journal of Machine Learning Research
- Machine Learning
- IEEE Transactions on Pattern Analysis and Machine Intelligence
- Journal of Artificial Intelligence Research
- Neural computation
- Annals of Statistics
- IEEE Transactions on Neural Networks
- Data Mining and Knowledge Discovery
- ...

Conferences

- International Conference on Machine Learning (ICML)
- European Conference on Machine Learning (ECML)
- Neural Information Processing Systems (NIPS)
- Uncertainty in Artificial Intelligence (UAI)
- International Joint Conference on Artificial Intelligence (IJCAI)
- International Conference on Artificial Neural Networks (ICANN)
- Computational Learning Theory (COLT)
- Knowledge Discovery and Data mining (KDD)
- ...