

Università di Verona

A.Y. 2021-22

Machine Learning & Artificial Intelligence

Introduction to

Pattern Recognition & Machine Learning & AI

Vittorio Murino

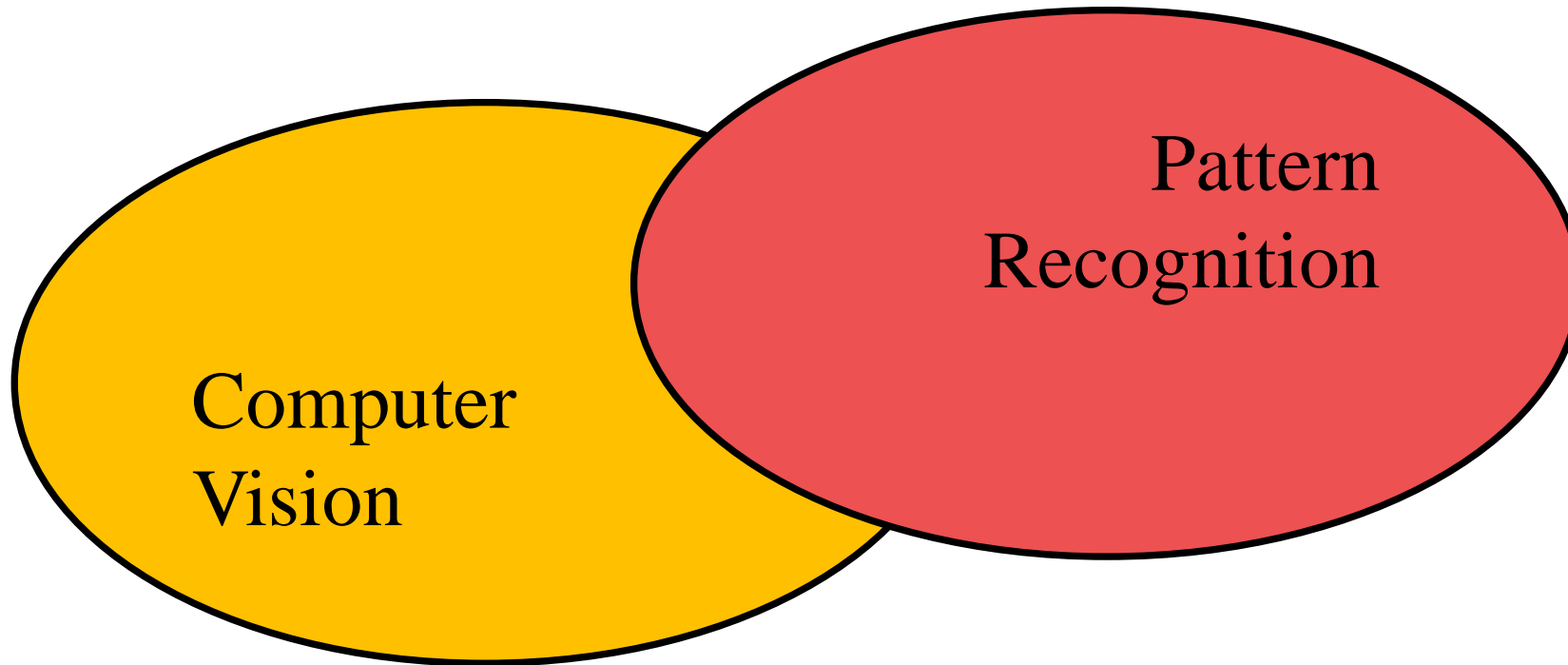
Overview

- Pattern Recognition systems in humans:
 - recognize the face of a known person, even if this changes hairstyle, has sunglasses, ...
 - understand what a person is saying, even if the tone of his voice varies;
 - recognize text in a handwritten letter;
 - ...
- Activities that humans solve in a very natural way, for a computer they have remarkable complexity

Some possible definitions

- *Pattern recognition*
 - study of problems related to the use of computers for the automatic recognition of data, otherwise called *patterns*.
- Study of how machines can observe the environment, learn to distinguish patterns of interest from background information and make decisions related to the category of patterns.
- *Pattern Recognition system*: the process that takes *raw* data input and performs an action based on the data "category".

Pattern Recognition and Computational (Computer) Vision



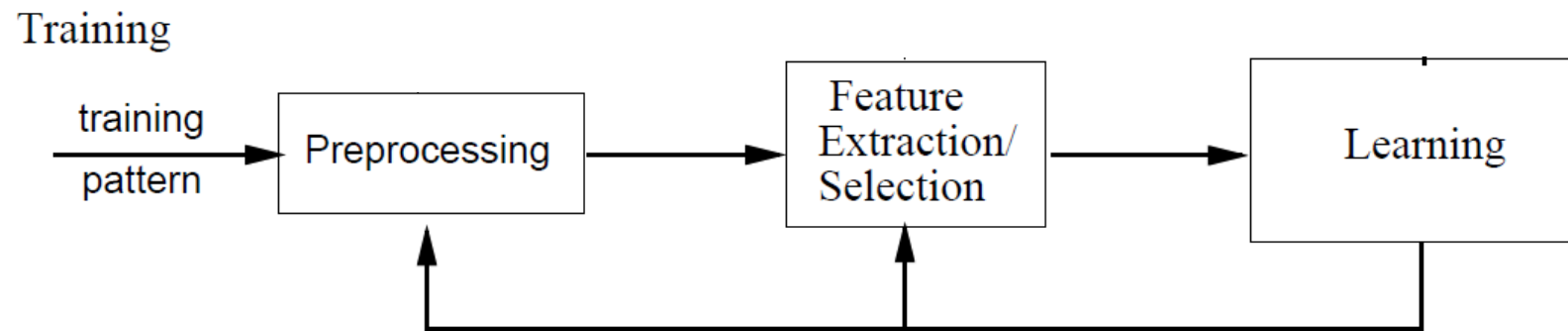
Recognition can be seen as a classification problem in which classes are known or not (and are estimated from the data)

Table 1: Applications

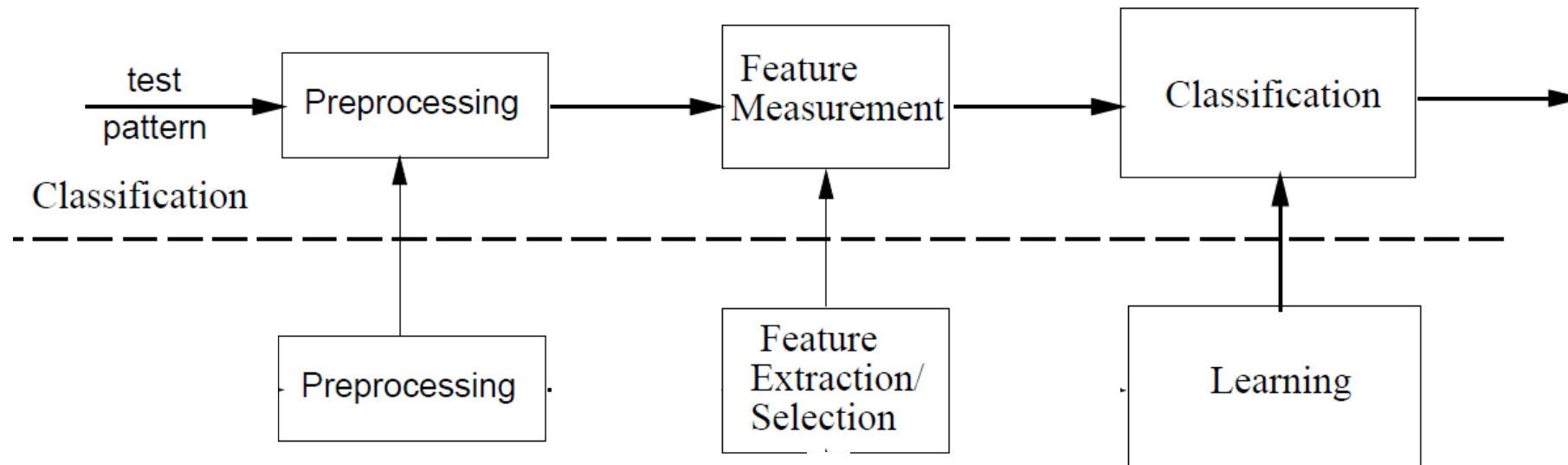
Problem Domain	Application	Input Pattern	Pattern Classes
Bioinformatics	Sequence Analysis	DNA/Protein sequence	Known types of genes/ patterns
Data mining	Searching for meaningful patterns	Points in multi- dimensional space	Compact and well- separated clusters
Document classification	Internet search	Text document	Semantic categories (e.g., business, sports, etc.)
Document image analysis	Reading machine for the blind	Document image	Alphanumeric characters, words
Industrial automation	Printed circuit board inspection	Intensity or range image	Defective / non-defective nature of product
Multimedia database retrieval	Internet search	Video clip	Video genres (e.g., action, dialogue, etc.)
Biometric recognition	Personal identification	Face, iris, fingerprint	Authorized users for access control
Remote sensing	Forecasting crop yield	Multispectral image	Land use categories, growth pattern of crops
Speech recognition NLP, NLU	Telephone directory enquiry without operator assistance	Speech waveform	Spoken words

A common point of such applications is that the available features (in the order of thousands) are not suggested by experts but must be extracted and optimized by data-driven procedures

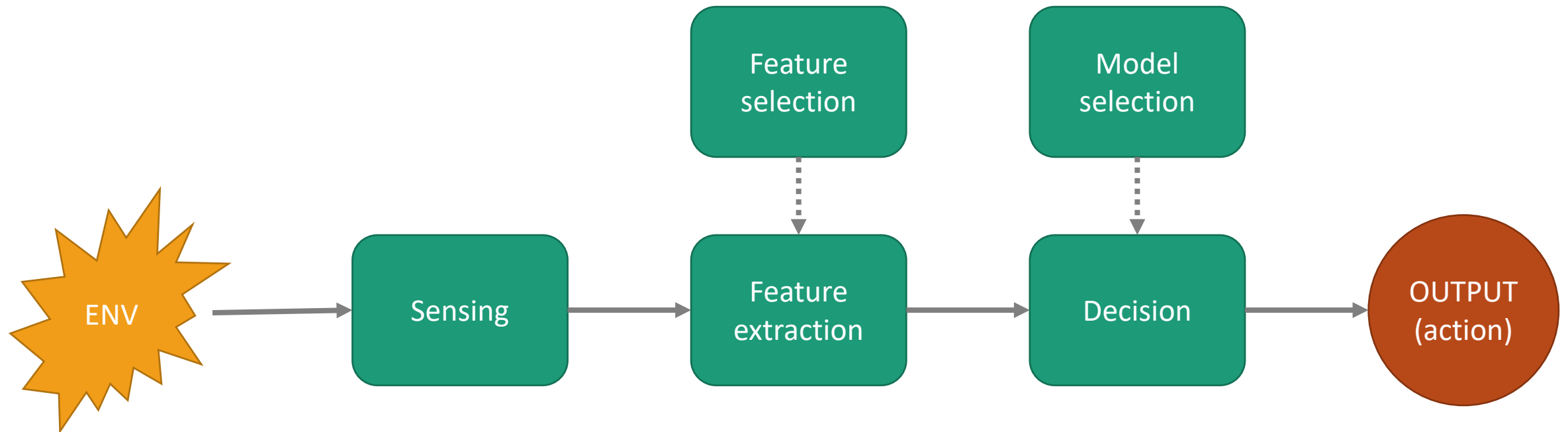
- Model of Statistical PR



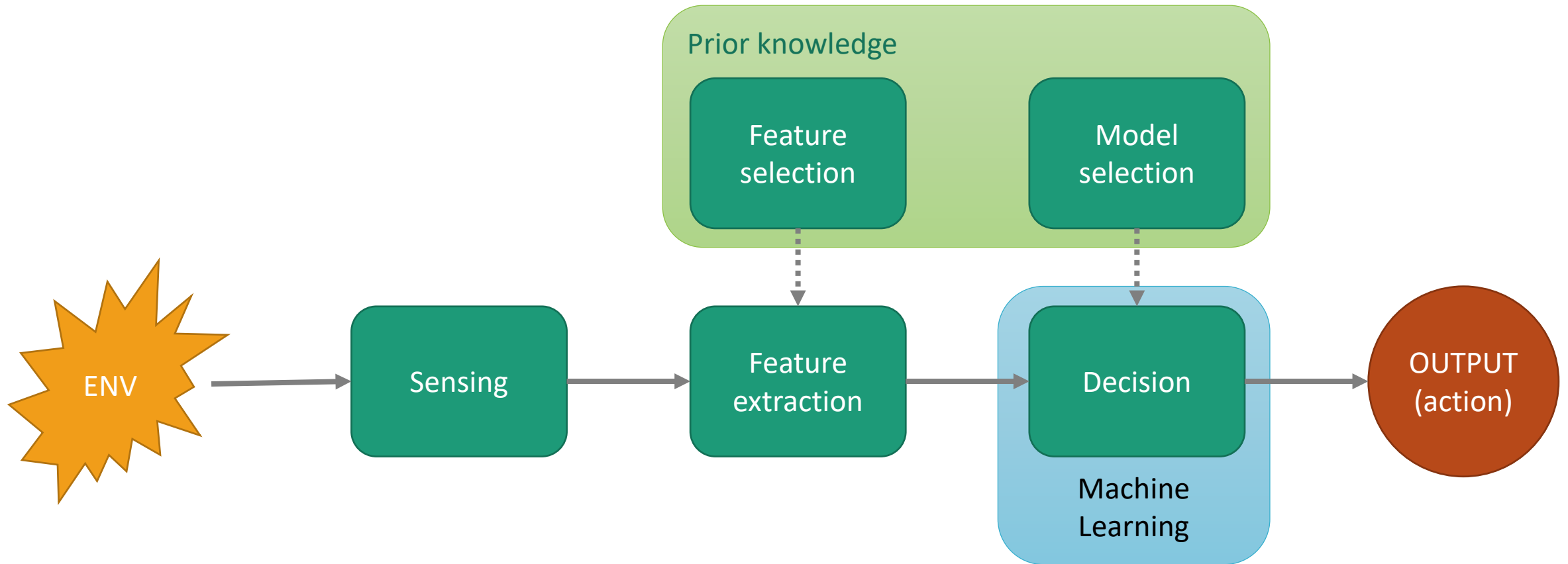
- Model of Statistical PR



Pattern recognition system

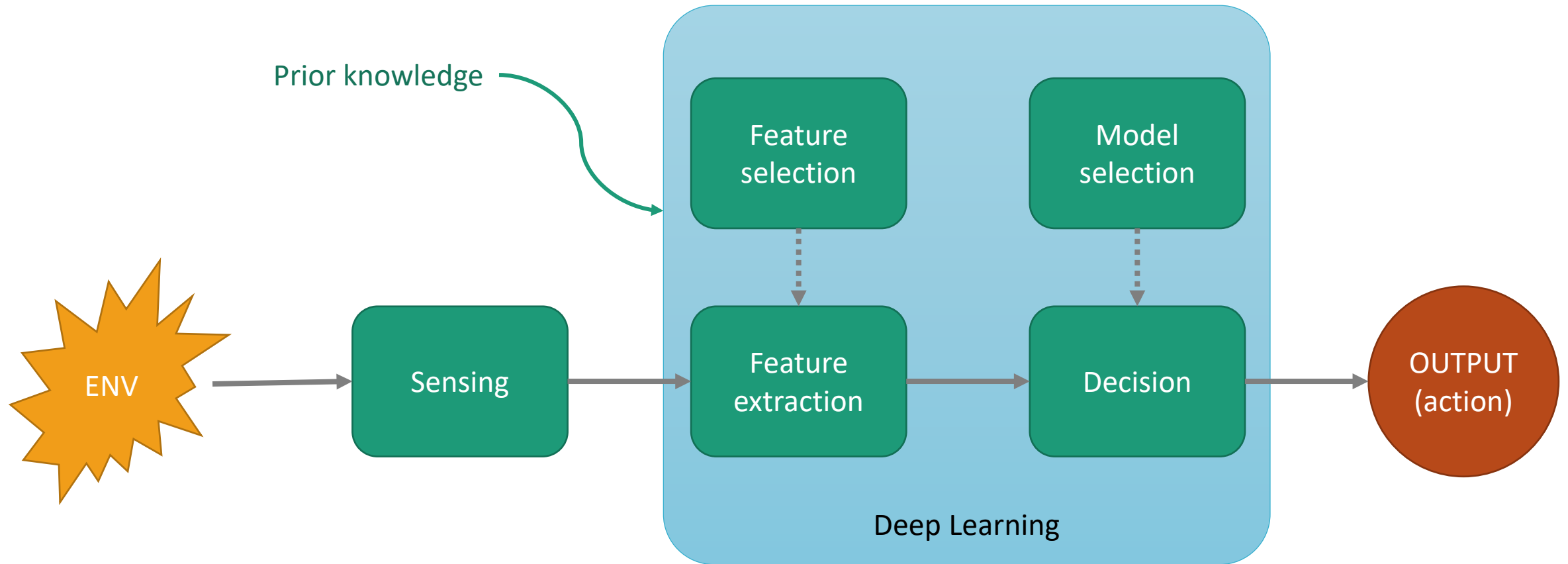


Pattern recognition systems: Traditional approach

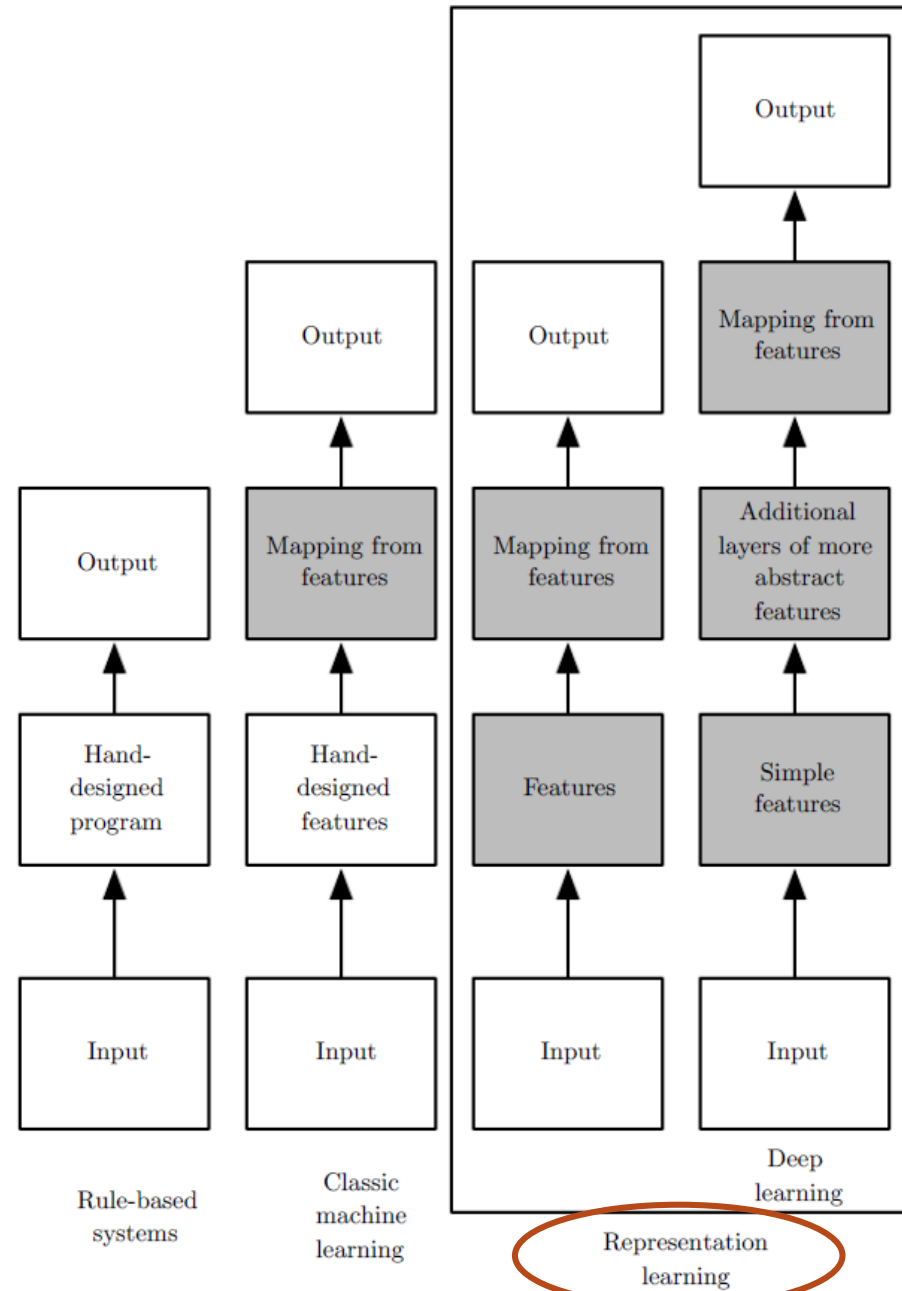


Pattern recognition systems:

Deep learning

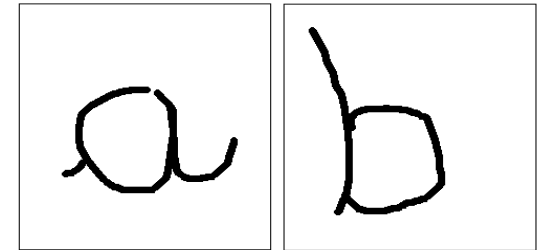


Deep Learning

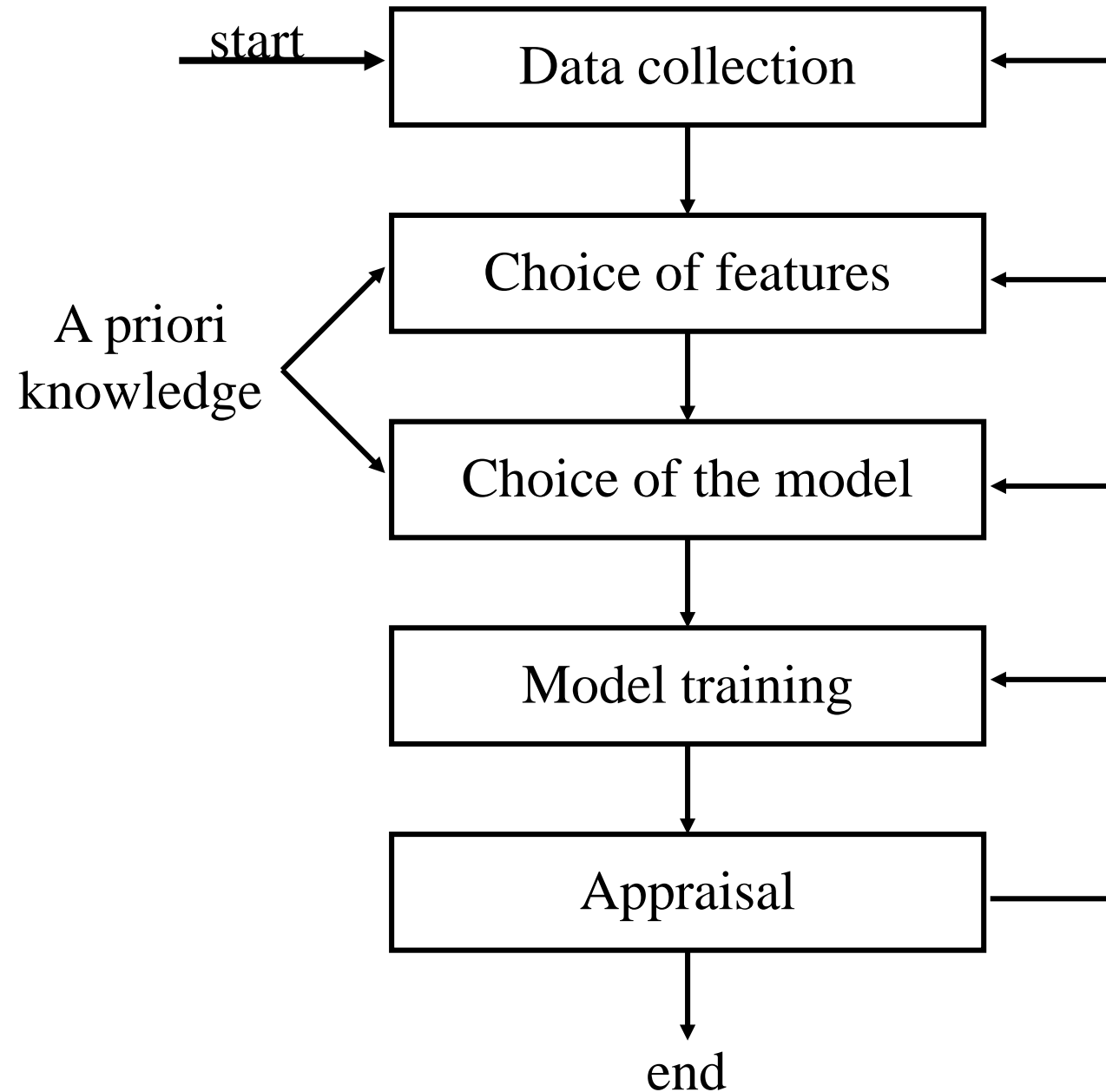


Pattern Recognition System

- Data collection
- Choice of the *features*
- Choice of the model
- Model training
- Appraisal



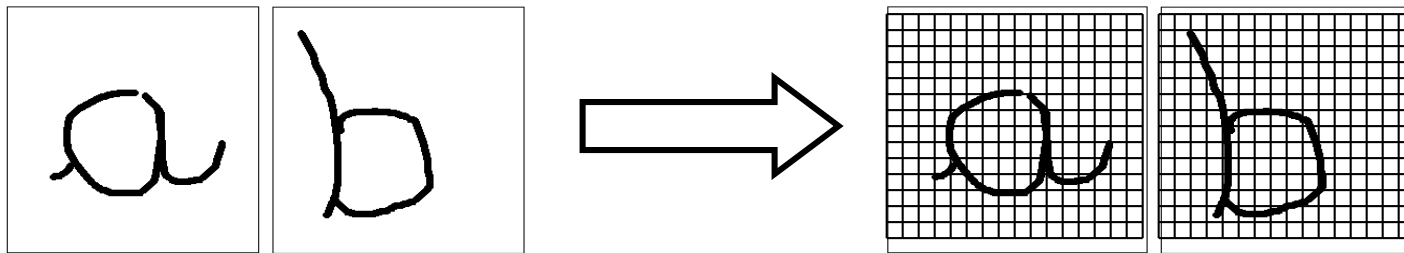
- *Example guide:* system that distinguishes between the hand-written letters "a" and "b".



Data collection

- Collection of a "sufficient" and "representative" set of examples from the problem under consideration.
 - "sufficient"?
 - "representative"?
- Sensor problems (resolution, bandwidth, ...)

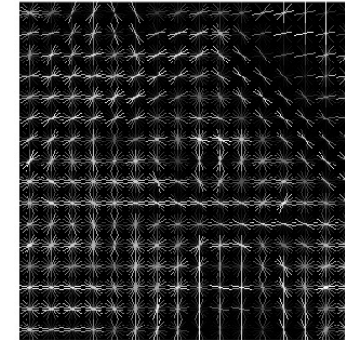
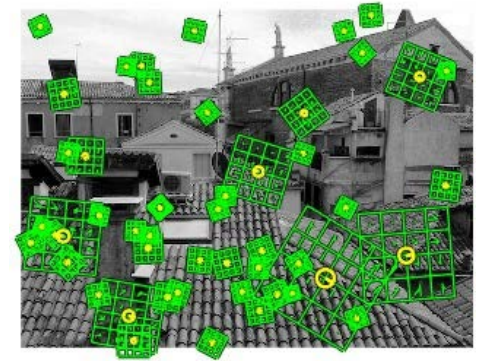
- *Example:* a set of images containing the letters "a" and "b" is captured by a camera, and stored in the computer



The image is represented by an array of pixels, each pixel takes value between 0 (completely white) and 1 (completely black)

Choice of features

- Data cannot be used as it is (256x256 image is 65536 pixels)
- Feature: measurable characteristics of the phenomenon under consideration (*pattern* = vector of features):
 - simple to calculate;
 - invariant to irrelevant transformations;
 - reliable;
 - independent;
 - discriminating;
 - few (*curse of dimensionality* problem);
- At this stage it is very useful to use a priori knowledge about the problem



Example:

- a feature could be the total number of black pixels:
 - invariant to the rotation and translation of the object
 - little discriminative: it does not take into account the form
- use of a priori knowledge: I must distinguish between "a" and "b", and I know that the letter "b" is typically higher and longer than the "a".
- use the height/width ratio as a feature

Features

Feature extraction is the process of transforming raw data into measurable values suitable for modeling.

Feature transformation is the process of transforming (combining) existing features to improve modelling performances.

Feature selection is the process of selecting a subset of relevant features from the input data to be used to make decisions.

Choice of the Model

- Choice of logical structure and mathematical basis of the rules of classification.
- Typically, the classifier estimates, for each object, a value that indicates the degree of belonging to one or more classes on the basis of the feature vector that characterizes it.

Choice of the Model

You have to decide:

- Type of model
- Parameters
- Dimensionality
- Learning procedure (cost function, optimization algorithm)
- Validation strategy
- Indeed, understanding whether the model represents effectively the phenomenon under observation

GENERALIZATION

We want to make predictions on inputs we have never observed before, and we only know they belong to the same domain of the training data.

Choice of the Model

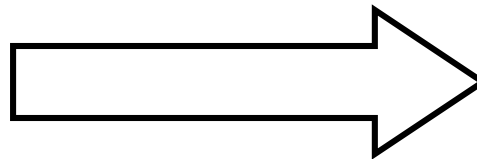
- There is no classifier that works for all applications
- *Example:* Use of a threshold classifier:
 - given an image I
 - calculate the height/width ratio $R(I)$;
 - If $R(I)$ is greater than a certain threshold θ , then the image is a "b", otherwise it is an "a".

Model training

- Synonyms:
 - classifier *training*
 - classifier *learning*
- Process using the data available (training set) to build the model

Examples drawn
from the problem
(*Training Set*)

A priori
knowledge



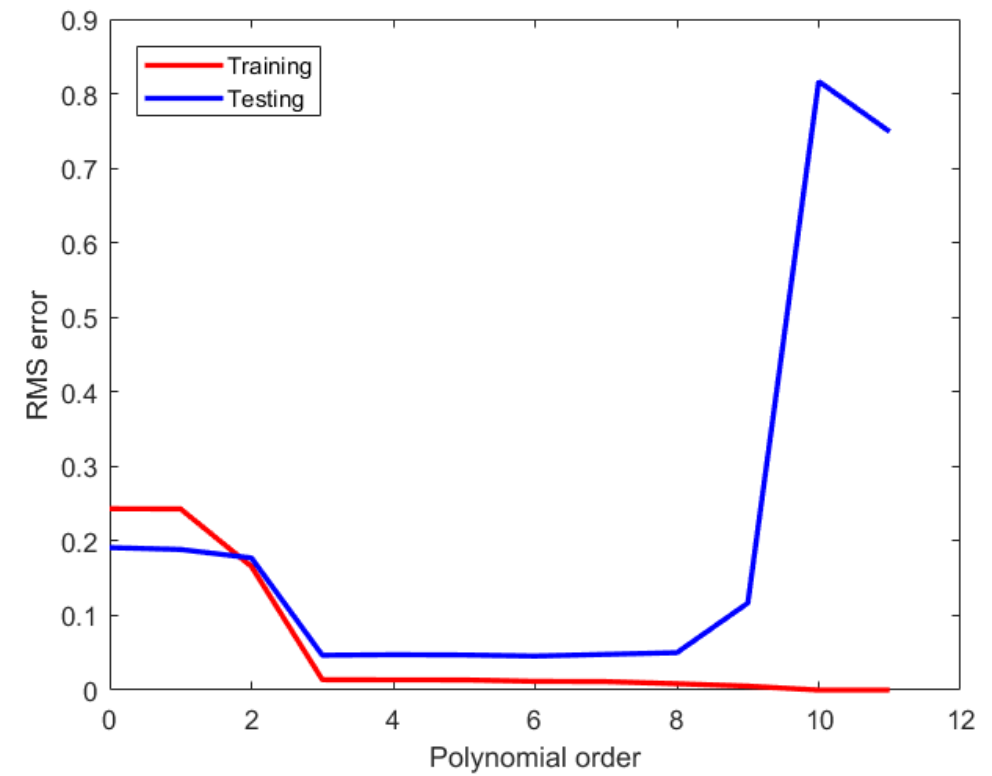
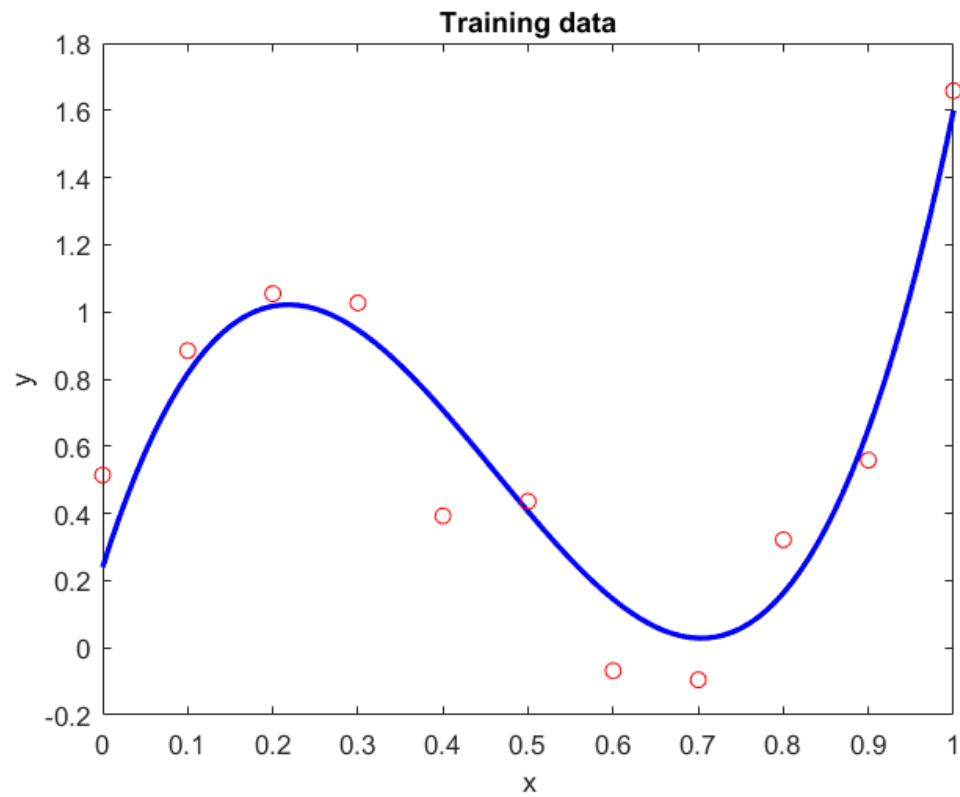
Rules governing
the phenomenon

Model training

Example

- Model training = threshold determination θ
 - A series of sample images is available for the characters "a" and "b" (*training set*)
 - $R(I)$ calculation for every image of the training set
 - Estimate a threshold θ "suitable" to separate the calculated $R(I)$ values

Overfitting



Supervised training

- Synonyms: *supervised learning, classification*
- Idea and goal:
 - the exact category of each training set element is known;
 - the goal is to create a system that can classify new objects.
- Problems:
 - understand if a training algorithm is able to find the optimal solution;
 - understand whether it converges and it is sufficiently scalable;
 - to understand if it can prefer simple solutions.

Supervised training

- *Example:*
 - the training set consists of a set of images depicting characters "a" and "b";
 - for each image we know the exact classification (i.e., if it is "a" or "b");
 - this information shall be used to determine the classifier threshold.

Unsupervised training

- Synonyms: *unsupervised learning, clustering*
- Ideal and goal:
 - no information about the categorization of training set elements;
 - the system must find the "natural" clusters (groups) within the training set, based on the "similarity" between patterns.
- Problems:
 - inherently more difficult than classification
 - “natural”?
 - “similarity”?

Unsupervised training

- *Example:*

- the training set consists of a set of images depicting characters "a" and "b";
- no information about image categorization;
- we try to create two groups, putting together those images that have similar value of $R(I)$ (the feature).

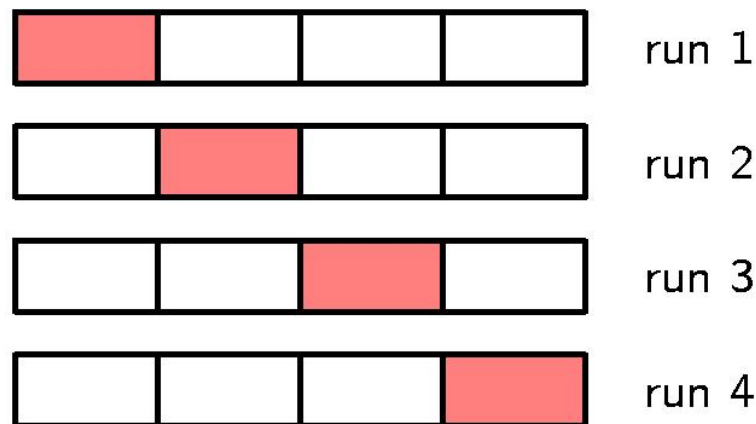
Training with Reinforcement

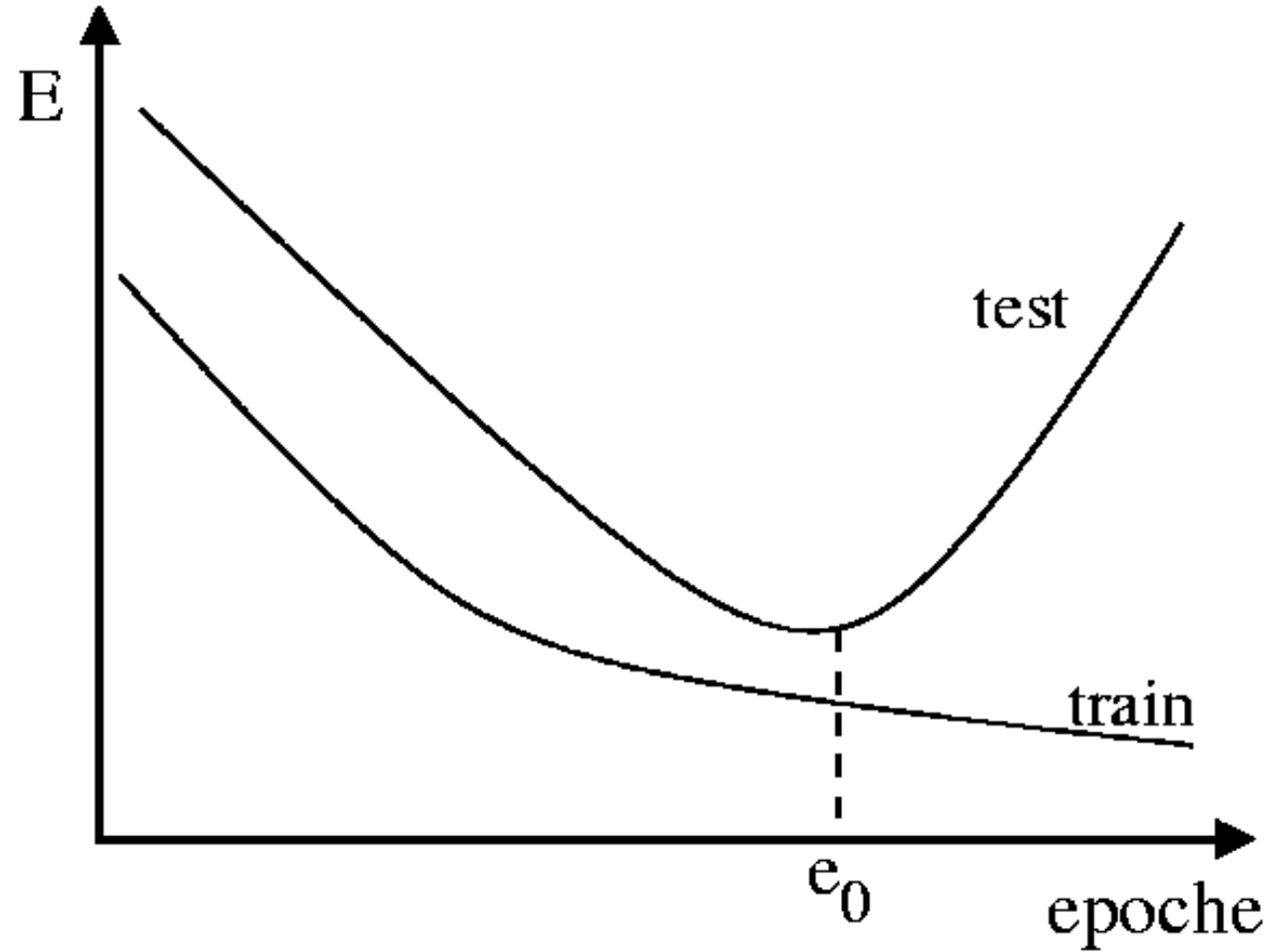
- Synonyms: *reinforcement learning, learning with a critic*
- Idea:
 - halfway between the two: no information on the exact category is provided, a judgment is given on the correctness of the classification
- The training strategy shall be modified:
 - a pattern is presented to the classifier
 - the classifier makes an attempt at classification
 - is told whether the attempt is correct or not
 - on the basis of judgment the classifier is modified

Validation and model selection

- Measurement of classifier performance
- *Generalization* performance: ability of the classifier to correctly classify also examples not present in the data set
- No error on the training set does not necessarily imply that you have obtained the optimal classifier (*overfitting, overtraining*)
- In order to avoid *overfitting* situations, it is always better to use two separate datasets, 1 for training and 1 for testing.

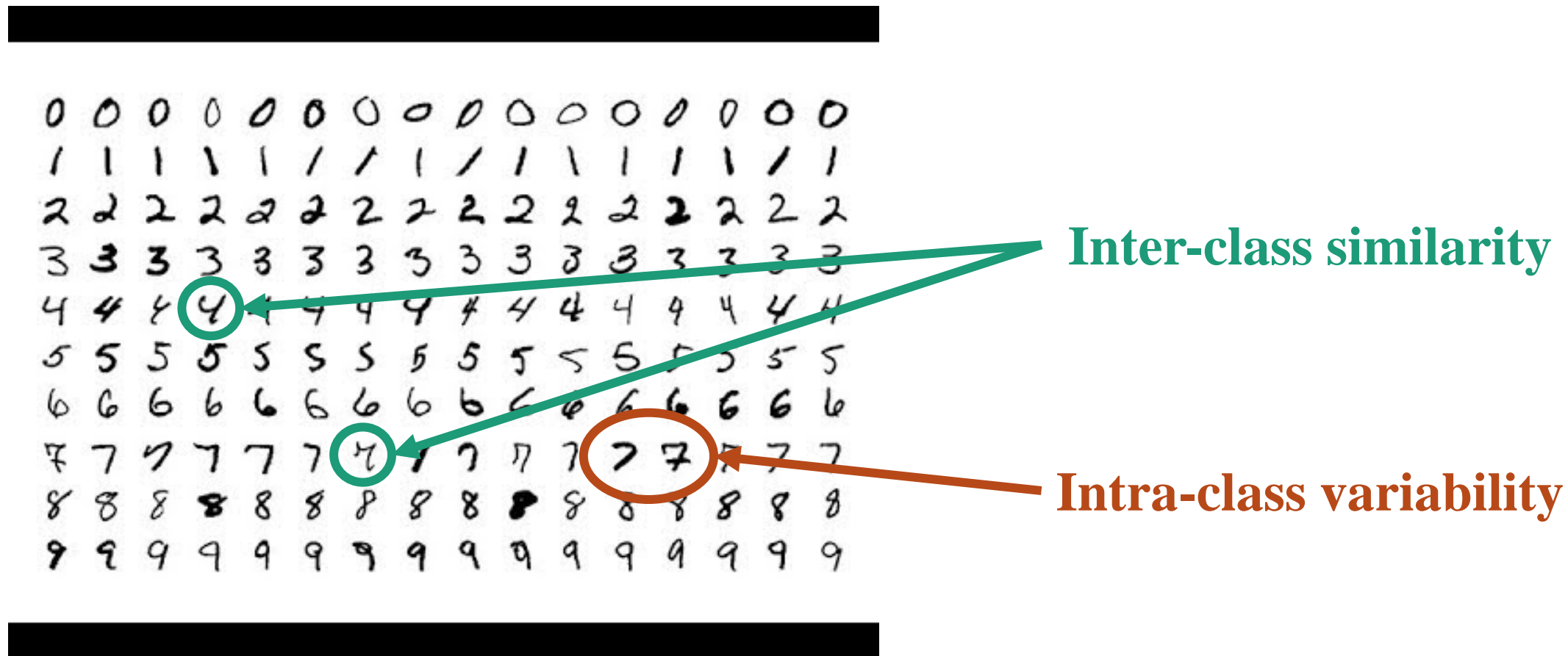
- Techniques for the choice of training set and testing set:
 - *Holdout*: the training set is randomly divided into two equal parts: one for training and one for testing, e.g., 50/50% or 80/20%
 - *Averaged Holdout*: multiple holdout partitions are made, and the result is averaged. This way you have independence from the particular chosen partition
 - *Leave-One-Out*: all patterns, except one used for testing, are used for training. It repeats for all possible combinations and averages.
 - *Leave-K-Out* (or *cross-folding* or *cross-validation*): like the previous one, it uses K elements for testing, instead of one.



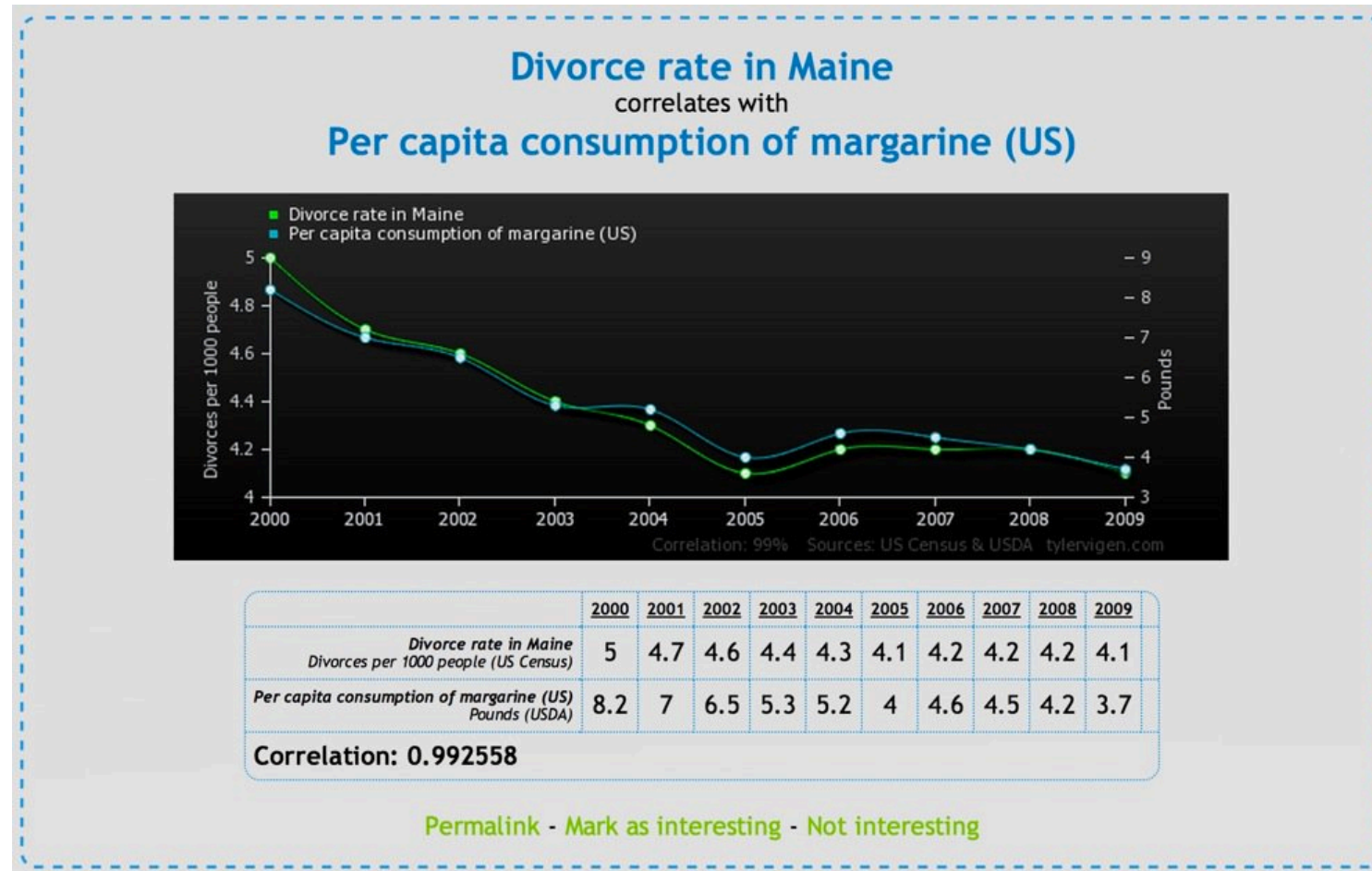


Training stops before overtraining phenomenon occurs (e_0)

Learning from *data*



Learning from *data: careful!!!*



<http://tylervigen.com/spurious-correlations>

“No free lunch” theorem

- **Lack of inherent superiority of any classifier**

- If we are interested solely in the generalization performance, are there any reasons to prefer one classifier or learning algorithm over another?
- If we make no prior assumptions about the nature of the classification task, can we expect any classification method to be superior or inferior overall?
- Can we even find an algorithm that is overall superior to (or inferior to) random guessing?

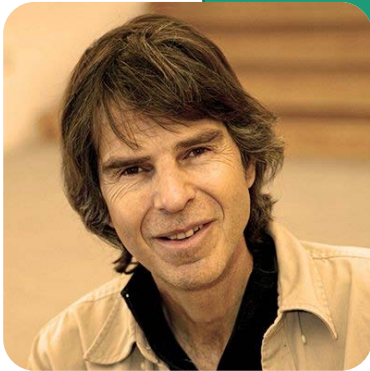
- The answer to these and several related questions is no: on the criterion of generalization performance, there are *no context-* or *problem-independent* reasons to favor one learning or classification method over another.

- The apparent superiority of one algorithm or set of algorithms is due to the nature of the problems investigated and the distribution of data.

“No free lunch” theorem

In a noise-free scenario where the loss function is the misclassification rate, if one is interested in off-training-set error, then there are no a priori distinctions between learning algorithms.

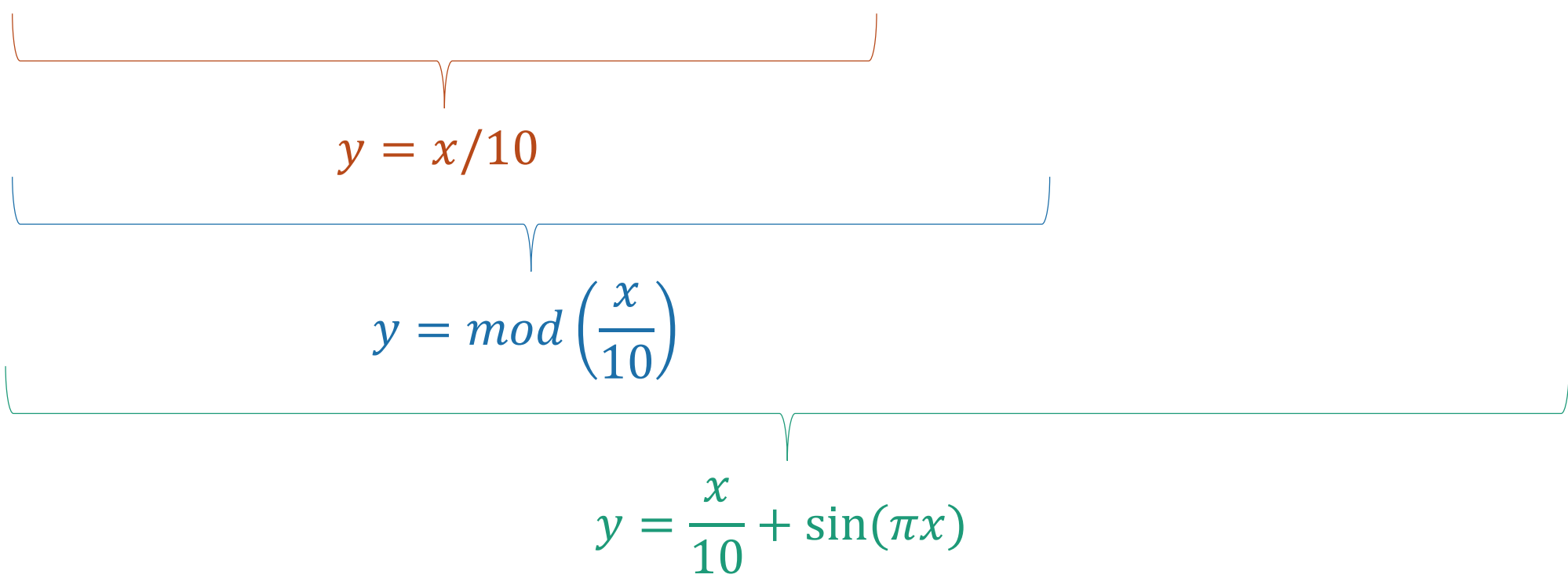
David H. Wolpert



It allows us, when confronting practical pattern recognition problems, to focus on the aspects that matter most – prior information, data distribution, amount of training data and cost or reward functions.

Inductive bias

In x	10	20	30	40	50	53.871...	61	66.5	70.2
Out y	1	2	3	4	5	5	6.1	7.65	7.6078


$$y = x/10$$

$$y = \text{mod}\left(\frac{x}{10}\right)$$

$$y = \frac{x}{10} + \sin(\pi x)$$

Ockham's razor

When presented with competing hypotheses to solve a problem, one should select the solution with the fewest assumptions.



William of Ockham

Performance metrics

CONFUSION MATRIX

		Ground Truth	
		True	False
Predictions	True	True Positives (TP)	False Positives (FP)
	False	False Negatives (FN)	True Negatives (TN)

ACCURACY

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

PRECISION

$$precision = \frac{TP}{TP + FP}$$

RECALL

$$recall = \frac{TP}{TP + FN}$$

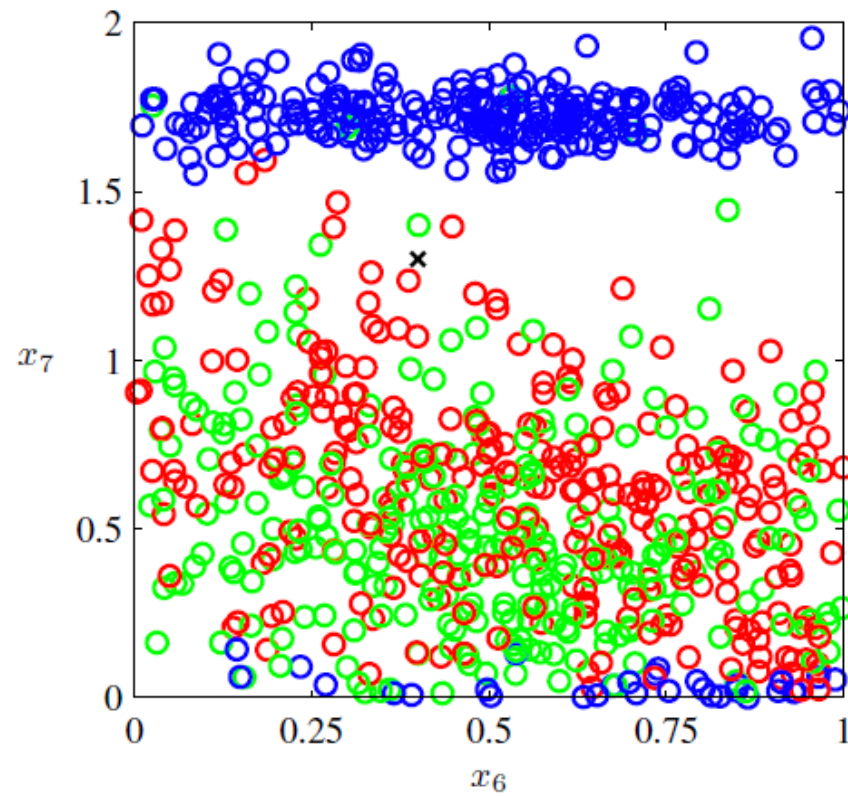
F-MEASURE

$$F_1 = 2 \frac{precision * recall}{precision + recall}$$

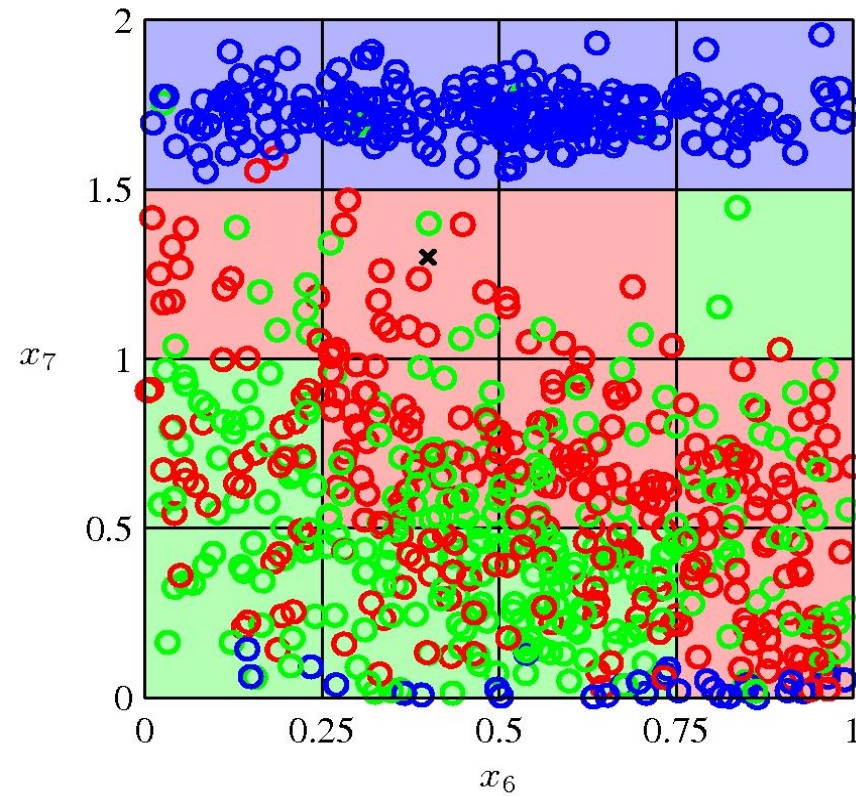
Curse of dimensionality

- Performance depends on the relationship between the number of samples, the number of *features*, and the complexity of the classifier.
- In theory, the probability of error does not increase if you add features.
- It has been shown that the $P(\text{err})$ tends to 0 if the number of features tends to infinity for a problem to 2 classes (and under the assumptions of normal pdf multivariate).
- In practice there are problems due to the fact that the assumptions are only approximations in real cases.
- In addition, the number of training samples must be exponential with respect to the number of features.
- All common classifiers suffer from this problem and guiding rules exist.

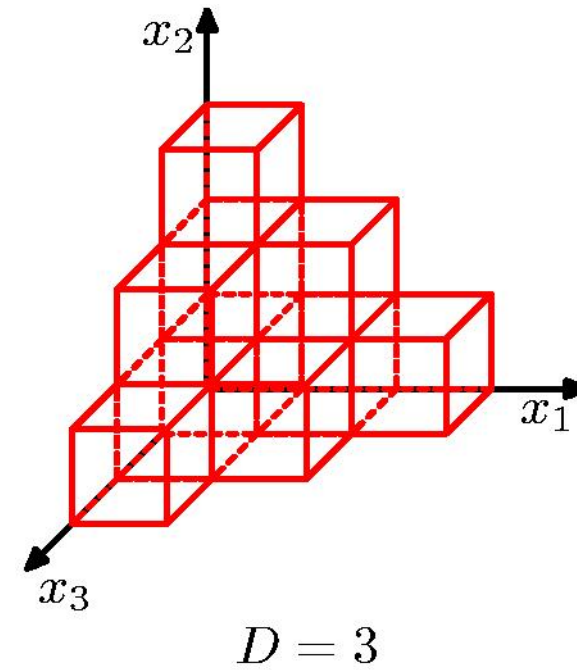
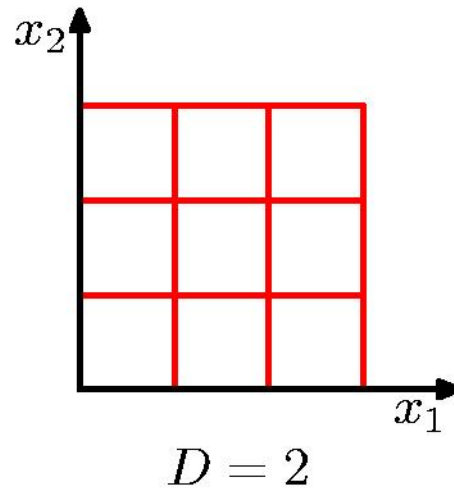
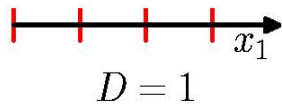
Curse of dimensionality



Curse of dimensionality

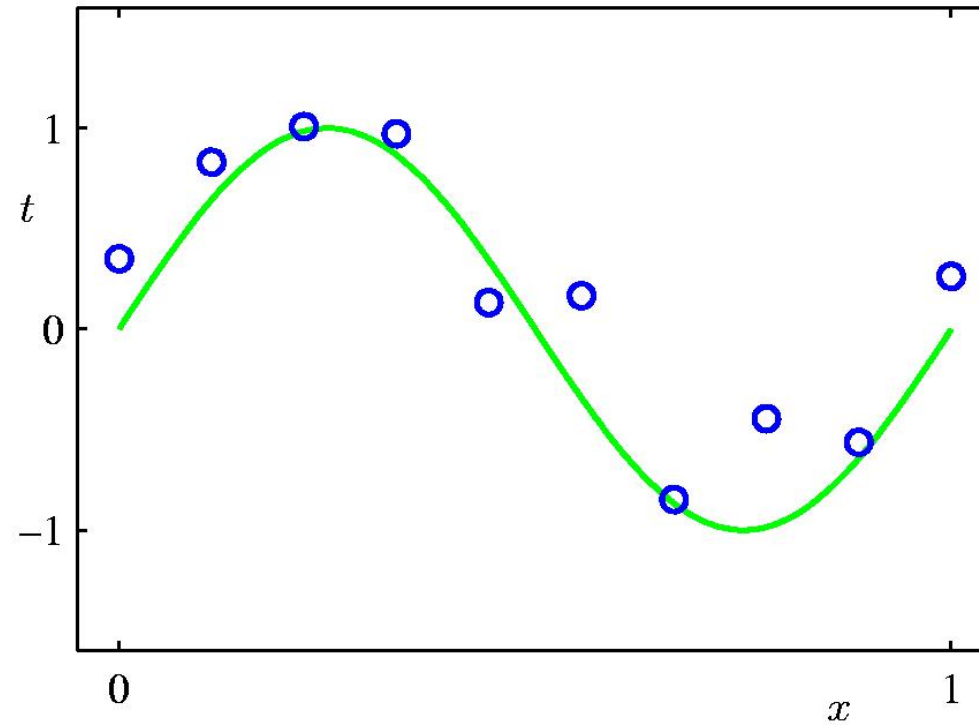


Curse of dimensionality



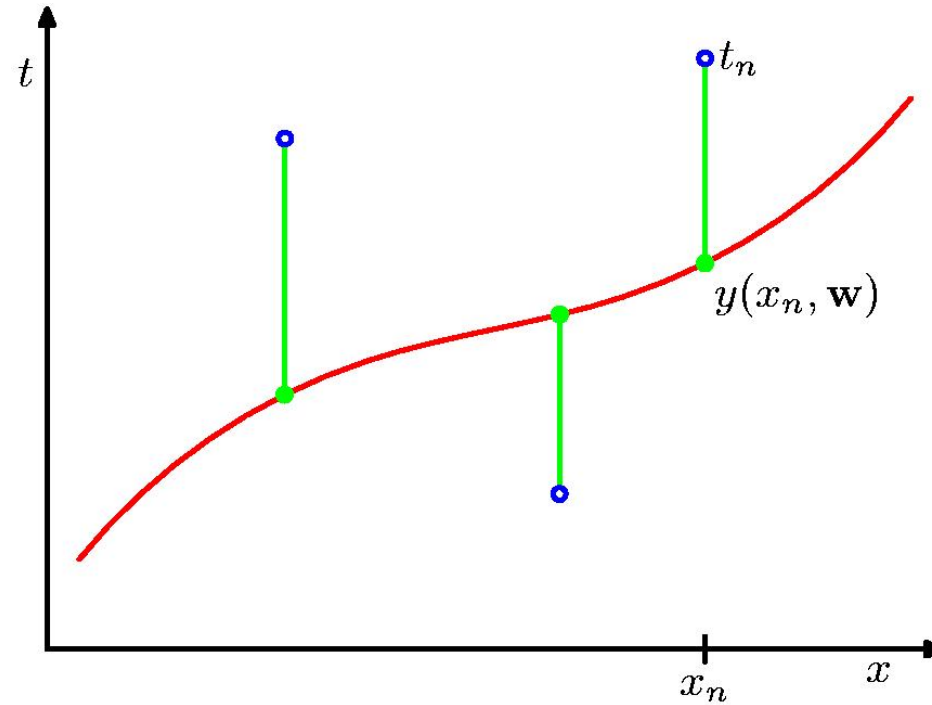
Example

Polynomial Curve Fitting



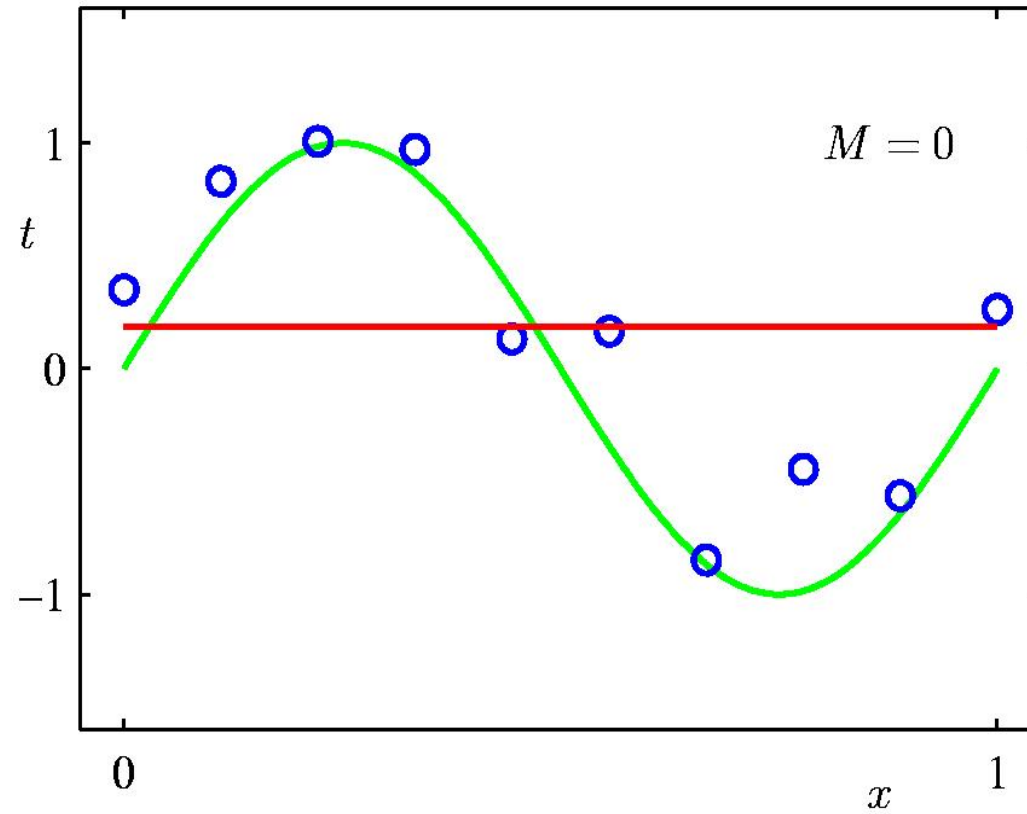
$$y(x, \mathbf{w}) = w_0 + w_1x + w_2x^2 + \dots + w_Mx^M = \sum_{j=0}^M w_jx^j$$

Sum-of-Squares Error Function

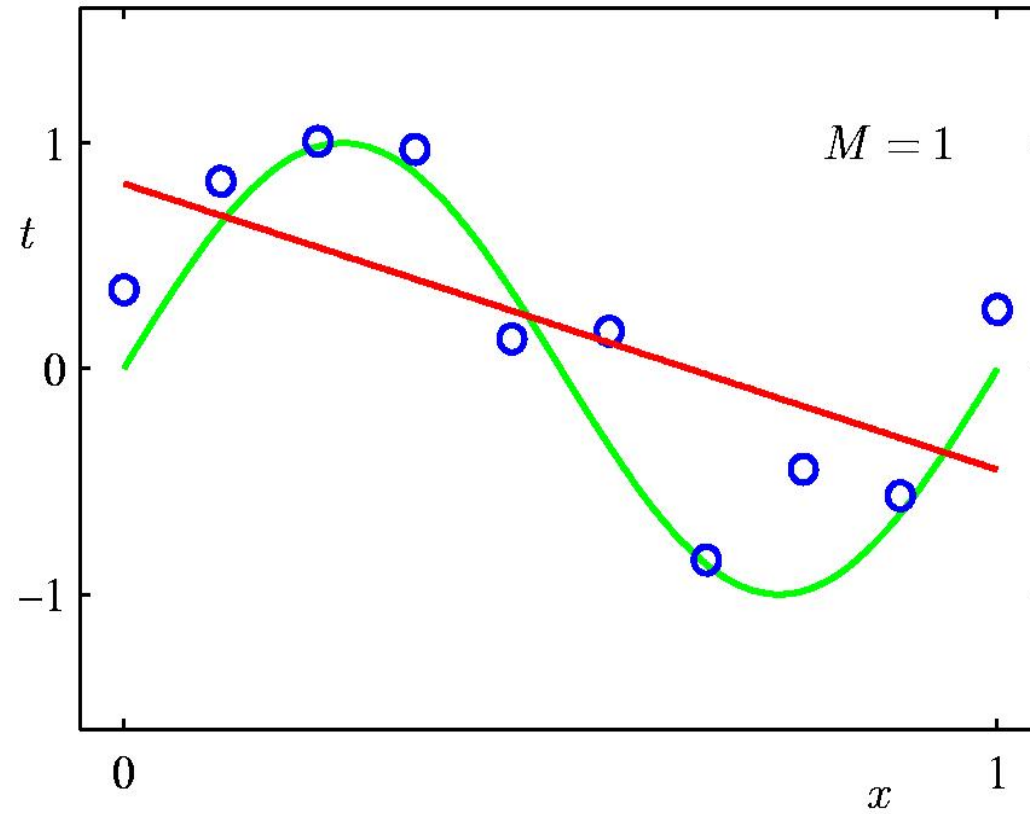


$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2$$

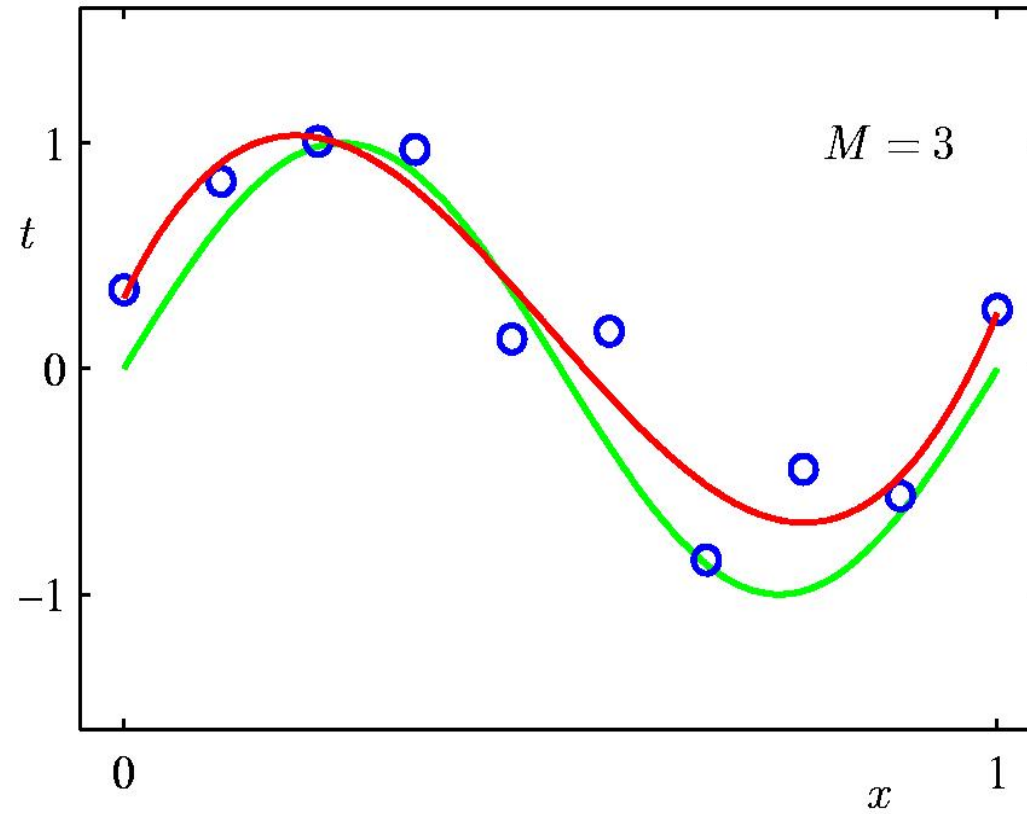
0th Order Polynomial



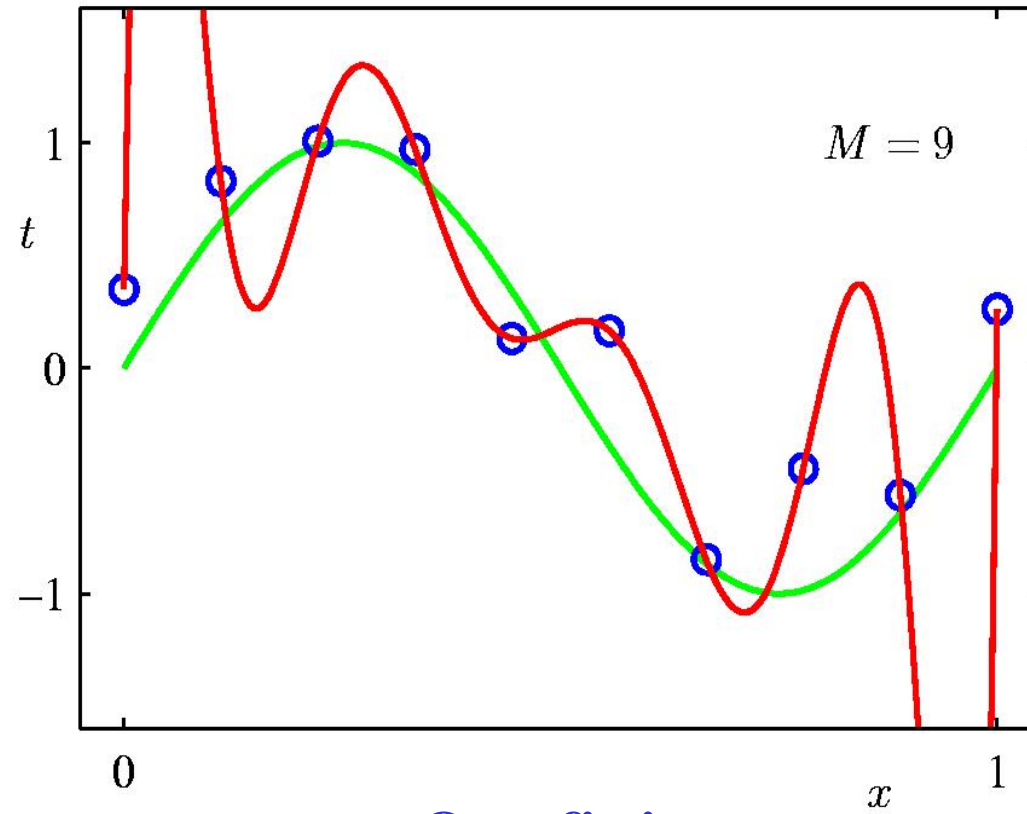
1st Order Polynomial



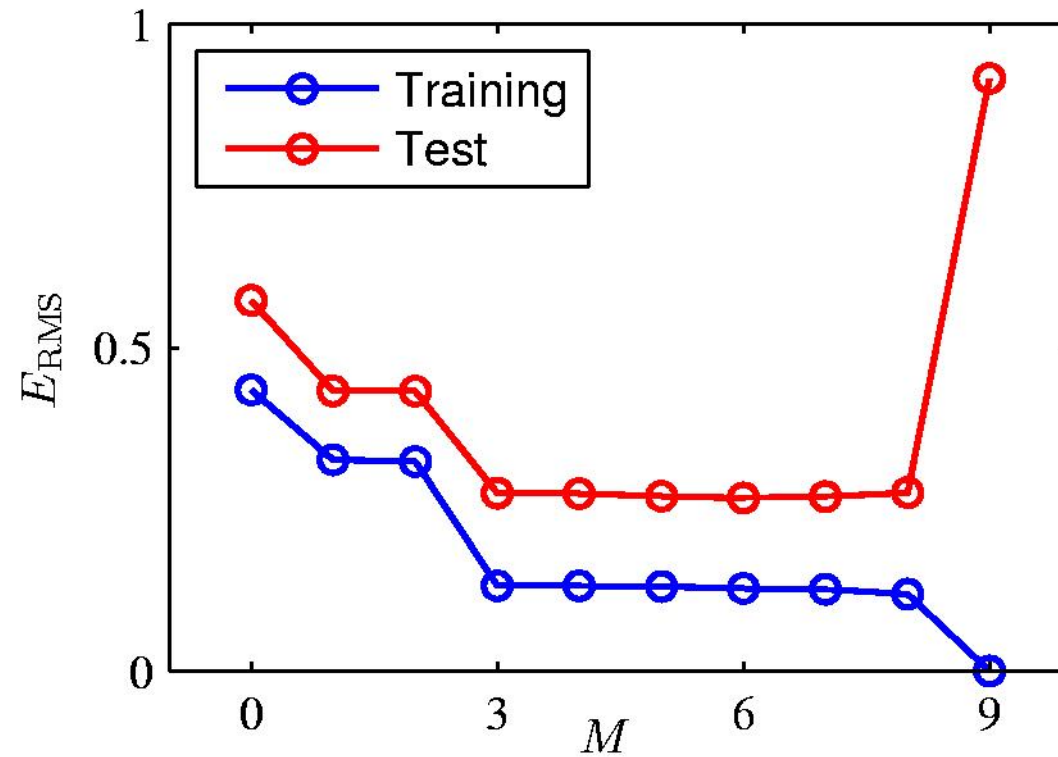
3rd Order Polynomial



9th Order Polynomial



Overfitting



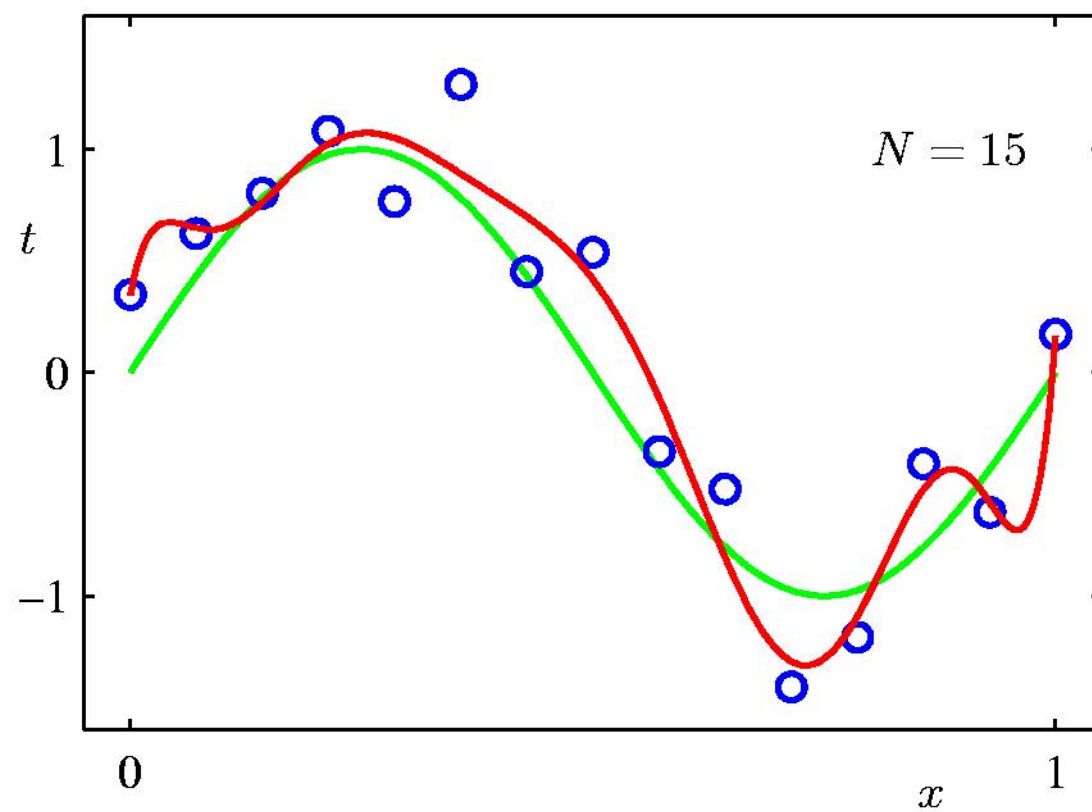
Root-Mean-Square (RMS) Error: $E_{\text{RMS}} = \sqrt{2E(\mathbf{w}^*)/N}$

Polynomial Coefficients

	$M = 0$	$M = 1$	$M = 3$	$M = 9$
w_0^*	0.19	0.82	0.31	0.35
w_1^*		-1.27	7.99	232.37
w_2^*			-25.43	-5321.83
w_3^*			17.37	48568.31
w_4^*				-231639.30
w_5^*				640042.26
w_6^*				-1061800.52
w_7^*				1042400.18
w_8^*				-557682.99
w_9^*				125201.43

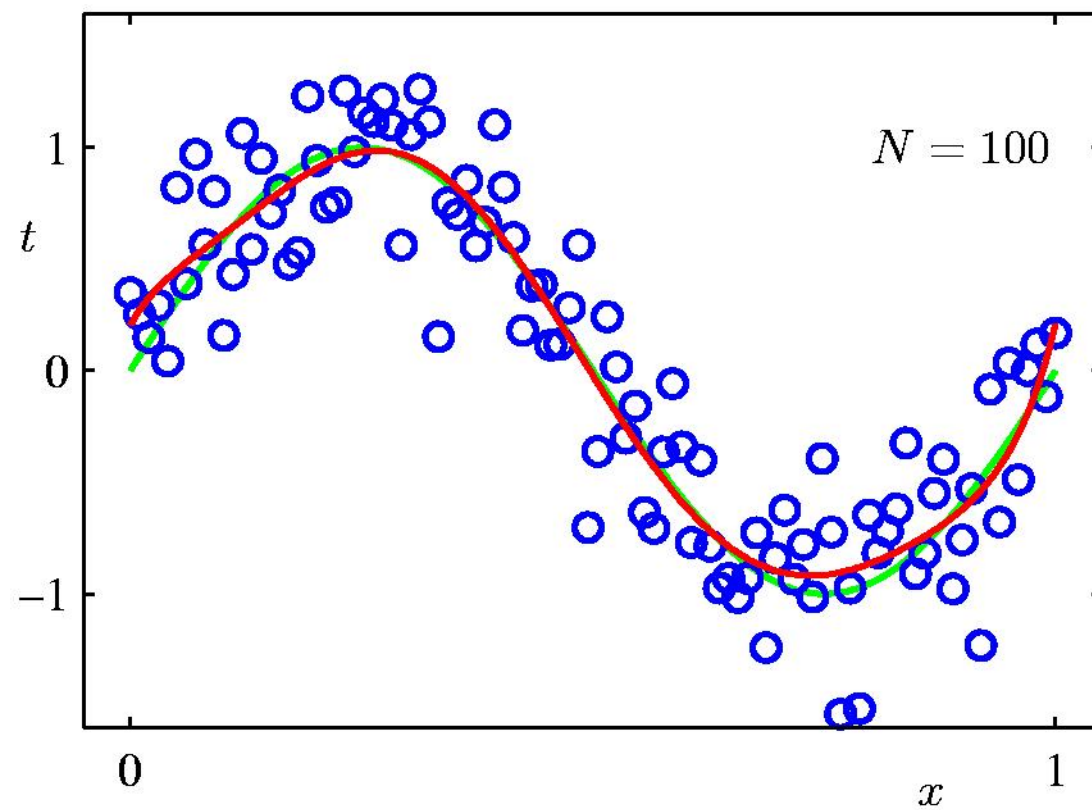
Data Set Size:

9th Order Polynomial $N = 15$



Data Set Size:

9th Order Polynomial $N = 100$

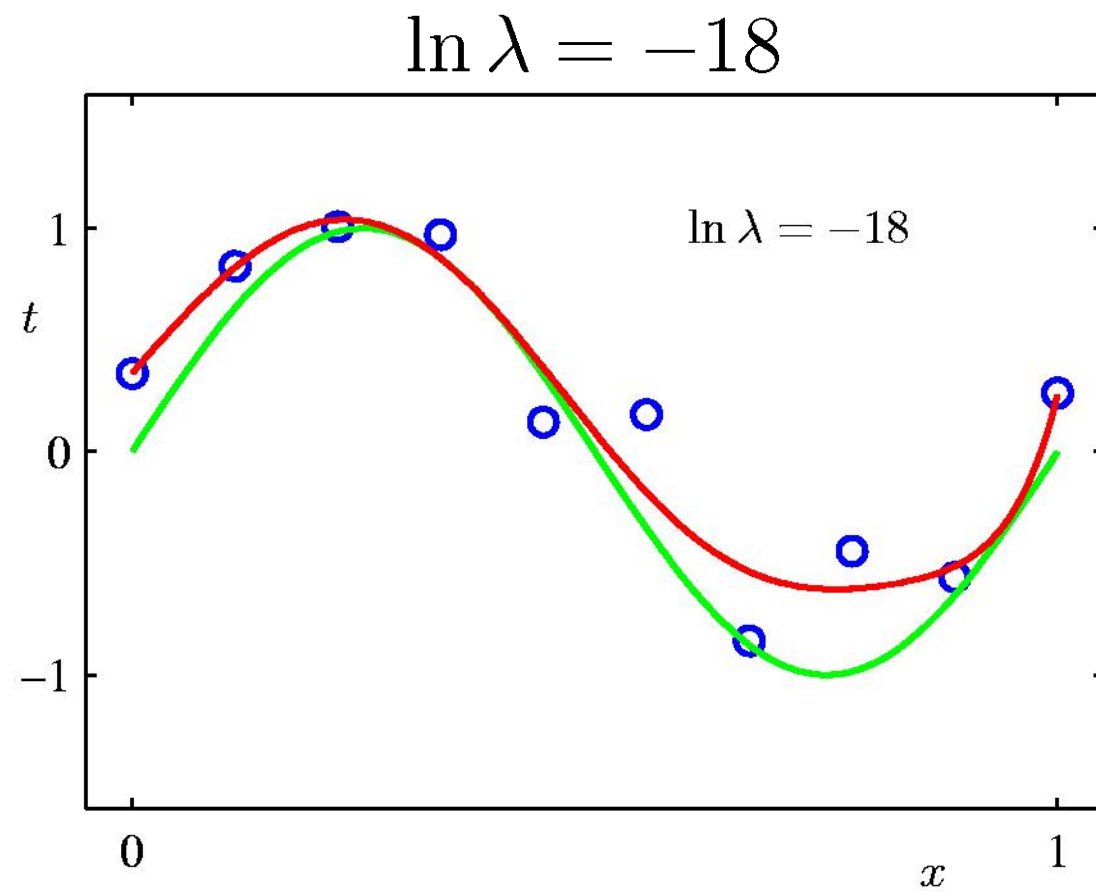


Regularization

- Penalize large coefficient values

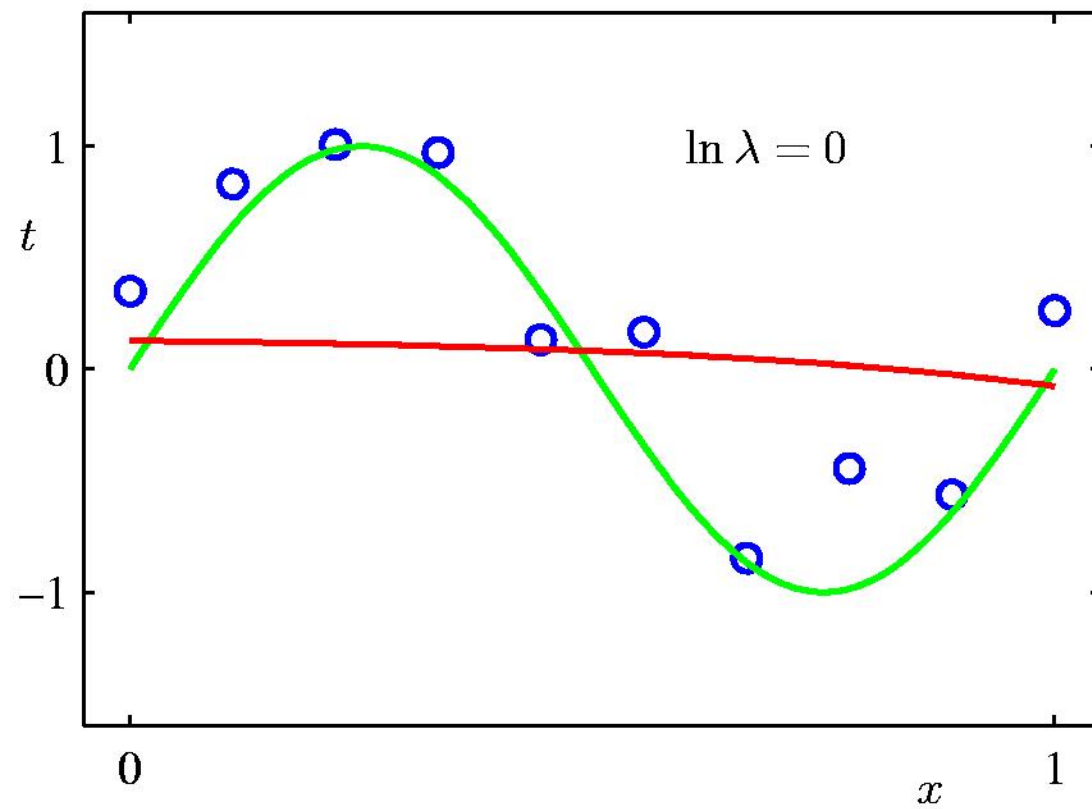
$$\tilde{E}(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2} \|\mathbf{w}\|^2$$

Regularization:



Regularization:

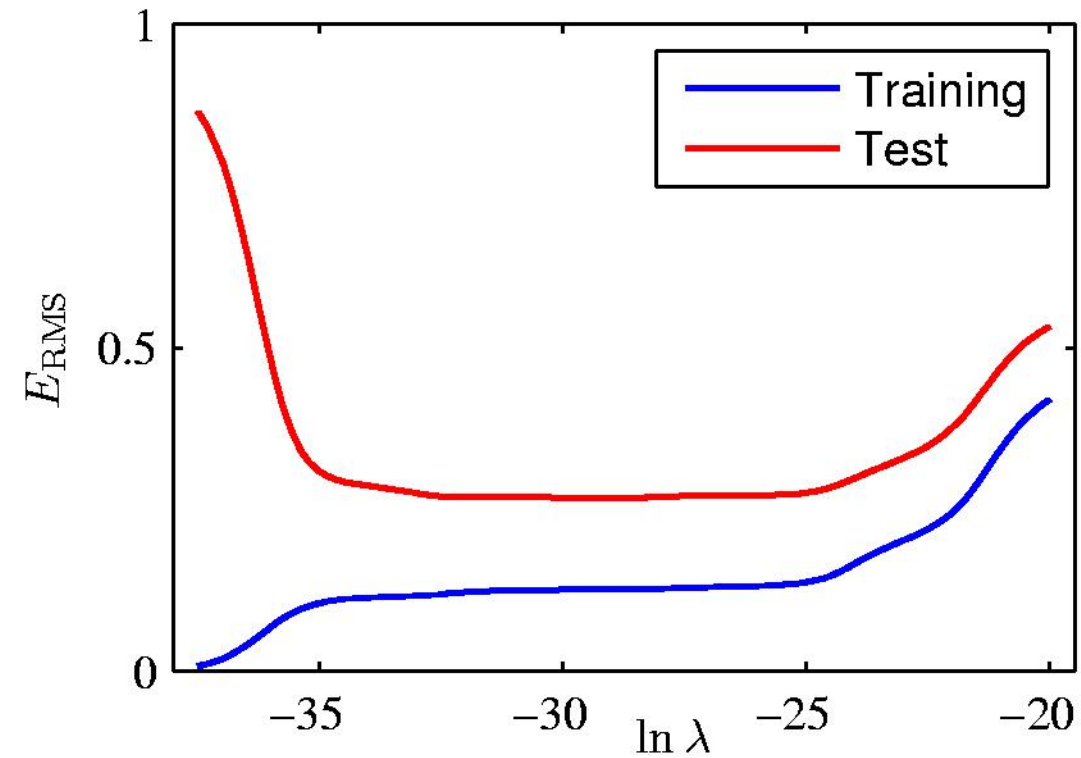
$$\ln \lambda = 0$$



Polynomial Coefficients vs. regularization

	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
w_0^*	0.35	0.35	0.13
w_1^*	232.37	4.74	-0.05
w_2^*	-5321.83	-0.77	-0.06
w_3^*	48568.31	-31.97	-0.05
w_4^*	-231639.30	-3.89	-0.03
w_5^*	640042.26	55.28	-0.02
w_6^*	-1061800.52	41.32	-0.01
w_7^*	1042400.18	-45.95	-0.00
w_8^*	-557682.99	-91.53	0.00
w_9^*	125201.43	72.68	0.01

Regularization: $\ln \lambda$ vs. E_{RMS}



Feature extraction and selection

- The number of features must be small to limit the cost of the measurement and not affect the accuracy of the classifier
- Feature Extraction: measure features from data or create new features from measured feature combinations
- Feature selection: best subset of features extracted
- Such features may have a better discriminative ability, but you lose the physical significance of these.
- Using a criterion function for reduction: typically the classification error of a subset of features.
- In addition, it is important to determine the size of the reduced space.

Feature extraction and projection methods

Method	Property	Comments
Principal Component Analysis (PCA)	Linear map; fast; eigenvector-based.	Traditional, eigenvector based method, also known as Karhunen-Loève expansion; good for Gaussian data.
Linear Discriminant Analysis	Supervised linear map; fast; eigenvector-based.	Better than PCA for classification; limited to $(c - 1)$ components with non-zero eigenvalues.
Projection Pursuit	Linear map; iterative; non-Gaussian.	Mainly used for interactive exploratory data-analysis.
Independent Component Analysis (ICA)	Linear map, iterative, non-Gaussian.	Blind source separation, used for de-mixing non-Gaussian distributed sources (features).
Kernel PCA	Nonlinear map; eigenvector-based.	PCA-based method, using a kernel to replace inner products of pattern vectors.
PCA Network	Linear map; iterative.	Auto-associative neural network with linear transfer functions and just one hidden layer.
Nonlinear PCA	Linear map; non-Gaussian criterion; usually iterative	Neural network approach, possibly used for ICA.
Nonlinear auto-associative network	Nonlinear map; non-Gaussian criterion; iterative.	Bottleneck network with several hidden layers; the nonlinear map is optimized by a nonlinear reconstruction; input is used as target.
Multidimensional scaling (MDS), and Sammon's projection	Nonlinear map; iterative.	Iterative; often poor generalization; sample size limited; noise sensitive; mainly used for 2-dimensional visualization.
Self-Organizing Map (SOM)	Nonlinear; iterative.	Based on a grid of neurons in the feature space; suitable for extracting spaces of low dimensionality.



Feature selection methods

Method	Property	Comments
Exhaustive Search	Evaluate all $\binom{d}{m}$ possible subsets.	Guaranteed to find the optimal subset; not feasible for even moderately large values of m and d .
Branch-and-Bound Search	Uses the well-known branch-and-bound search method; only a fraction of all possible feature subsets need to be enumerated to find the optimal subset.	Guaranteed to find the optimal subset provided the criterion function satisfies the monotonicity property; the worst-case complexity of this algorithm is exponential.
Best Individual Features	Evaluate all the m features individually; select the best m individual features.	Computationally simple; not likely to lead to an optimal subset.
Sequential Forward Selection (SFS)	Select the best single feature and then add one feature at a time which in combination with the selected features maximizes the criterion function.	Once a feature is retained, it cannot be discarded; computationally attractive since to select a subset of size 2, it examines only $(d - 1)$ possible subsets.
Sequential Backward Selection (SBS)	Start with all the d features and successively delete one feature at a time.	Once a feature is deleted, it cannot be brought back into the optimal subset; requires more computation than sequential forward selection.
“Plus l -take away r ” Selection	First enlarge the feature subset by l features using forward selection and then delete r features using backward selection.	Avoids the problem of feature subset “nesting” encountered in SFS and SBS methods; need to select values of l and r ($l > r$).
Sequential Forward Floating Search (SFFS) and Sequential Backward Floating Search (SBFS)	A generalization of “plus- l take away- r ” method; the values of l and r are determined automatically and updated dynamically.	Provides close to optimal solution at an affordable computational cost.

Pattern Recognition approaches

■ Syntactic approach:

- hierarchical approach. Analogy between patterns structure and syntax of a language:

- patterns  phrases of a language
- subpattern primitive  alphabet

■ Statistical approach:

- each pattern is associated with a feature vector that represents a point in the multidimensional space of the problem.
- The information on the problem, the dependencies between the various factors and the results produced are all expressed in terms of probability.

- Template matching
- Neural networks

They are not all necessarily independent

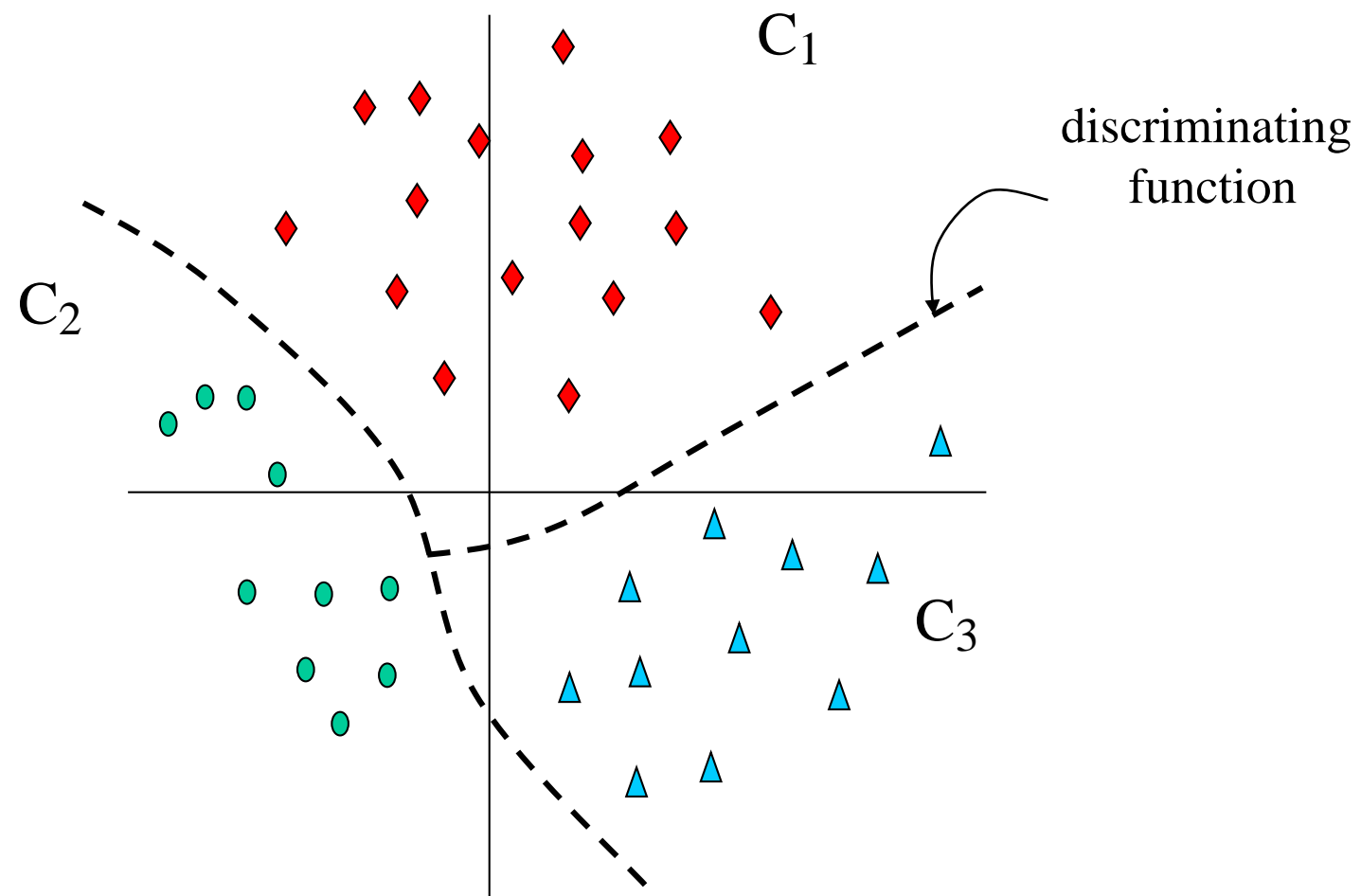
Approach	Representation	Recognition Function	Typical Criterion
Template matching	Samples, pixels, curves	Correlation, distance measure	Classification error
Statistical	Features	Discriminant function	Classification error
Syntactic or structural	Primitives	Rules, grammar	Acceptance error
Neural networks	Samples, pixels, features	Network function	Mean square error

Template matching

- Comparison of a model (template, typically a 2D form) with the data available for all possible instances (different poses, scale).
- Measurement of distance (correlation).
- Brute force approach, computationally onerous, although optimizations exist.

Statistical classification

- The statistical description of “objects” uses elementary numerical descriptors called *features*, which form the so-called *patterns*, $\mathbf{x} = (x_1, x_2, \dots, x_n)$ or *feature vectors*.
- The set of all possible patterns forms the space of patterns or *feature space*.
- If there is a (hyper-)surface of separation between classes the problem is said with *separable classes*.
- If hyper-surfaces are hyper-planes, then the problem is said to be *linearly separable*.



Bayes Classifier

- The statistical classification assumes known the Probability Density Function (PDF) of the feature x for each class $P(C_i|x)$ (known from the problem or estimated given a training set)
 - for example, a Gaussian with known or estimated mean and variance.
- Alternatively, let's assume known or estimated the a-priori probability of classes $P(C_i)$ and the conditional probability $p(x/C_i)$.
- In general, the Bayes risk (expected value of the loss function) should be minimized:

$$R(C_i|x) = \sum_{j=1}^C L(C_i, C_j) P(C_j | x)$$

- If the loss (matrix) function L is diagonal and binary (1 if $i=j$, 0 otherwise), then it can be simplified using Bayes' decision theory.
- These probabilities are linked by the **Bayes' theorem**:

$$P(C_i / x) = \frac{p(x / C_i)P(C_i)}{p(x)} \quad \text{where} \quad p(x) = \sum_{i=1}^c p(x / C_i)P(C_i)$$

- The Bayes classifier classifies a new object x as belonging to the class C_k such that

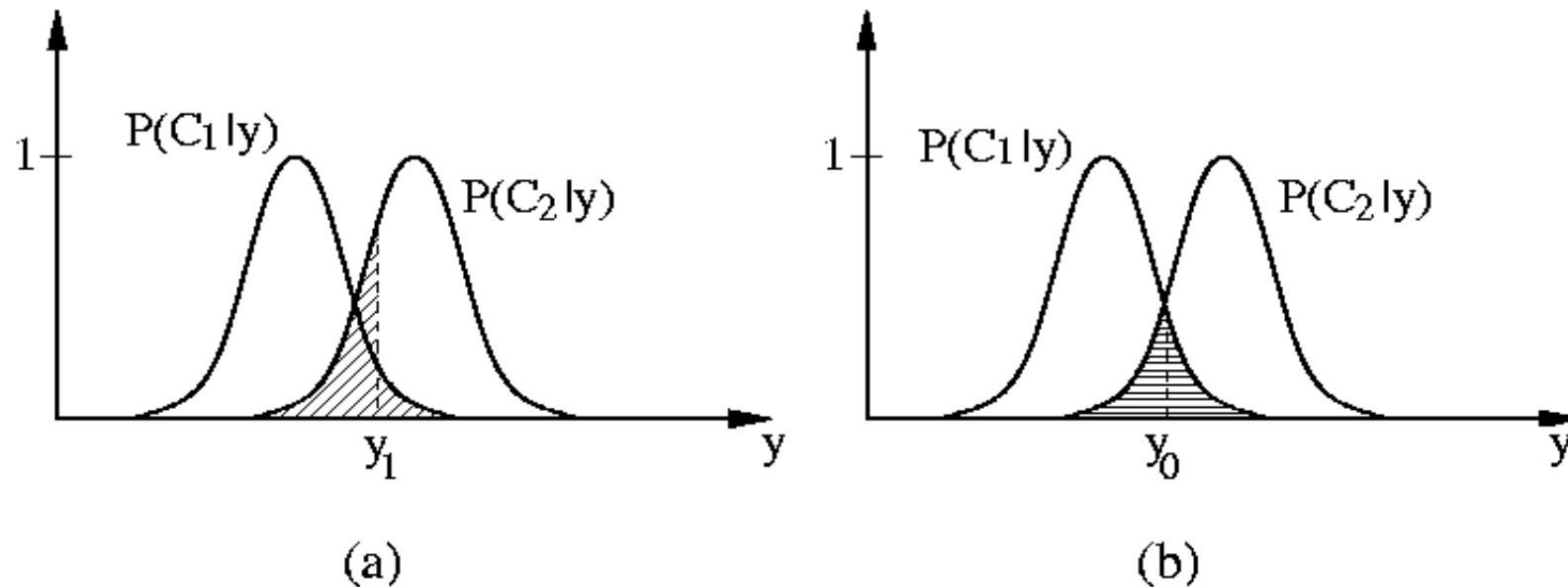
$$\forall i \neq k \quad P(C_k | x) > P(C_i | x)$$

oppure

$$P(x | C_k)P(C_k) > P(x | C_i)P(C_i)$$

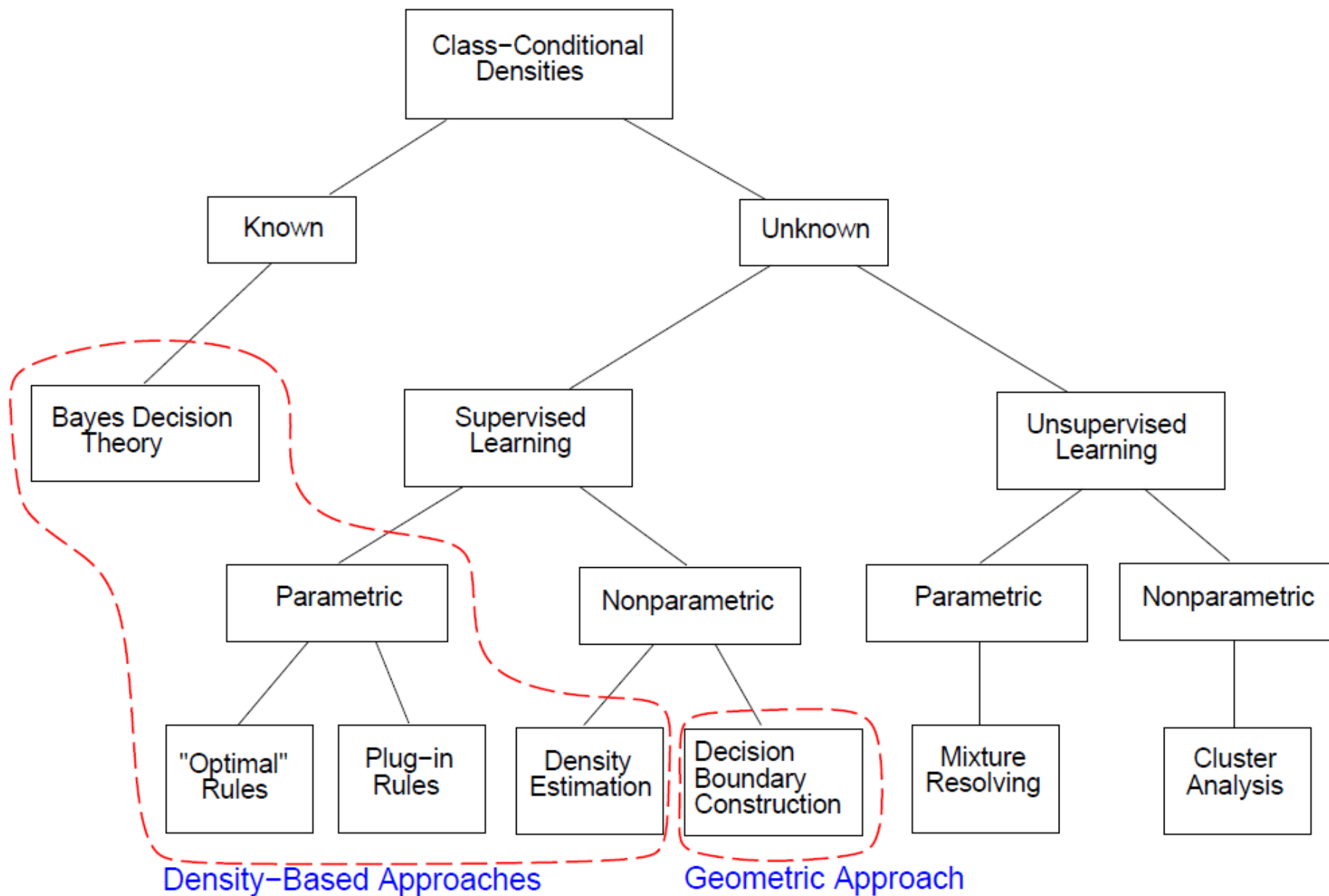
- The Bayes classifier is the theoretical optimum, as it minimizes the probability of making a mistake.
- Problem: the problem is that probability densities are almost never known a priori, it is necessary to estimate them from the available data.
- The performance of this classifier depends on the goodness of these estimates.

- Example: the purpose is to determine the discriminating function for a two-class problem, whose probability distribution $P(C_i|y)$ is shown in the figure below
 - y_0 is the threshold chosen by the Bayes classifier and minimizes the probability of error (dashed area); for any other choice (y_1), the probability of error is greater.



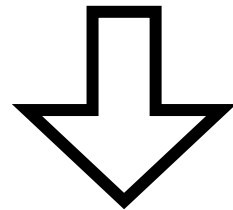
PDF estimation

- *Parametric classifiers*
 - the distribution model is fixed and its parameters are estimated on the basis of the training set;
 - example: Gaussian classifier.
- *Non parametric classifiers*
 - no assumption on the form of the pdf, the estimate is based exclusively on data;
 - example: K-nearest Neighbor.
- *Semi-parametric classifiers*
 - you have a very general class of pdf templates, in which the number of parameters can be increased in a systematic way to build more flexible models;
 - example: neural networks.



K Nearest Neighbor (KNN)

- Non-parametric classifier.
- Widely used for its simplicity, flexibility and reasonable accuracy of the results produced.
- IDEA: two elements of the same class will, most likely, have similar characteristics, that is, they will be close in the space of the points that represents the problem



The class of a point can be determined by analyzing the class of points in its neighbourhood

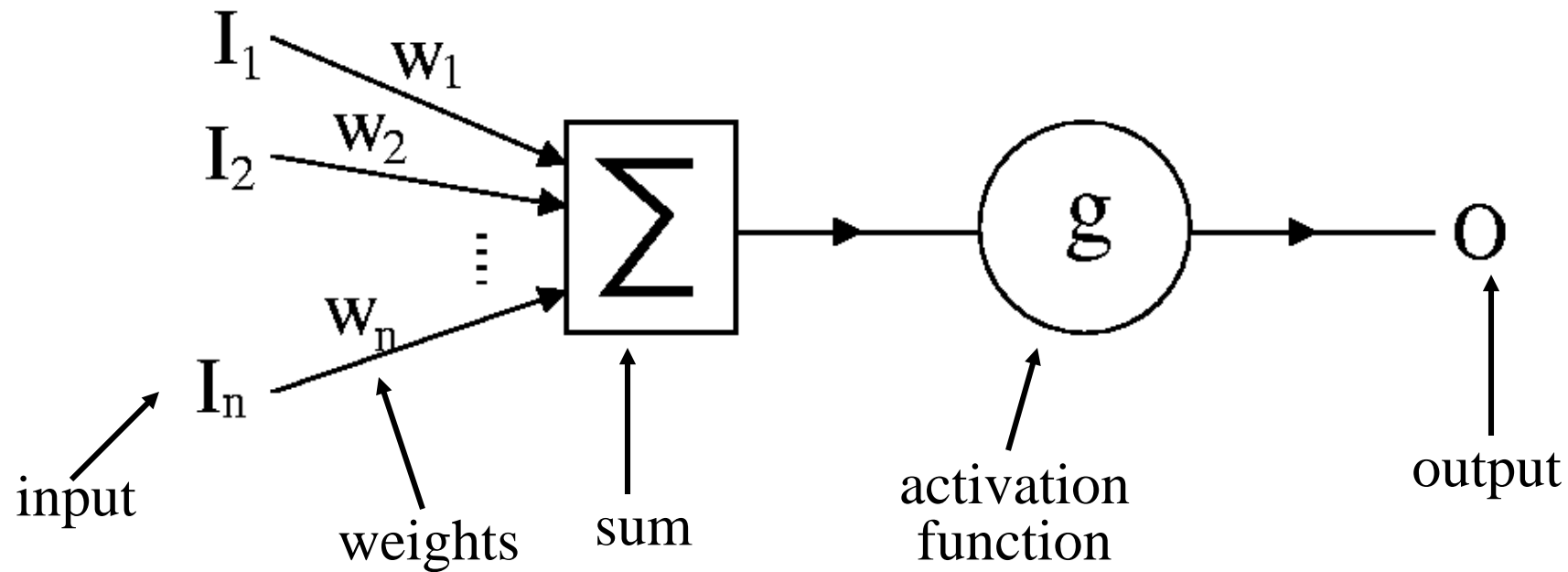
Algorithm

- Given a set of examples X , given a point to be classified x_0 :
 - We identify the set U of the K points belonging to X nearer to x_0 according to a certain metric Σ (usually the Euclidean distance);
 - the most frequent class C^* within the set U is calculated;
 - x_0 will be classified as belonging to C^* .
- Problem: choice of parameter K and metric Σ .

Neural networks: motivations

- Artificial information processing system that emulates the animal nervous system.
- Features of the animal nervous system:
 - robust and resistant to faults or damages
 - flexible, adapts to new situations by learning
 - also works with approximate information, incomplete or affected by errors
 - allows a highly parallel computation
 - small and compact
 - (small power consumption)

- Neural networks: complex structure, composed of many elementary processing units connected to each other in various ways.
- Elementary processing units are called *neurons*.
- The connections are called *synapses*.



- The output is calculated by $O = g(\sum_{i=1}^n w_i I_i - \theta)$

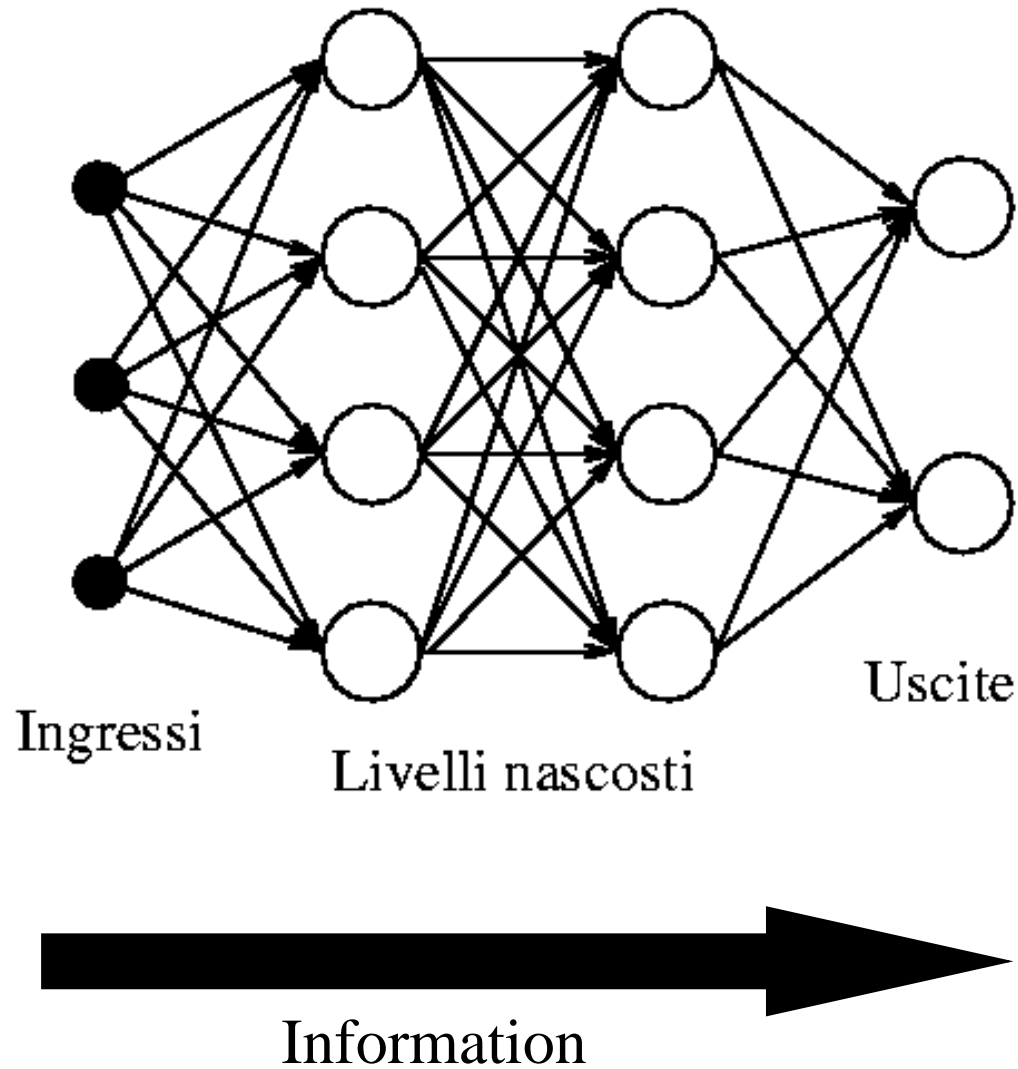
- Different possibilities for the activation function

- Heaviside
$$g(a) = \begin{cases} 0 & \text{se } a < 0 \\ 1 & \text{altrimenti} \end{cases}$$

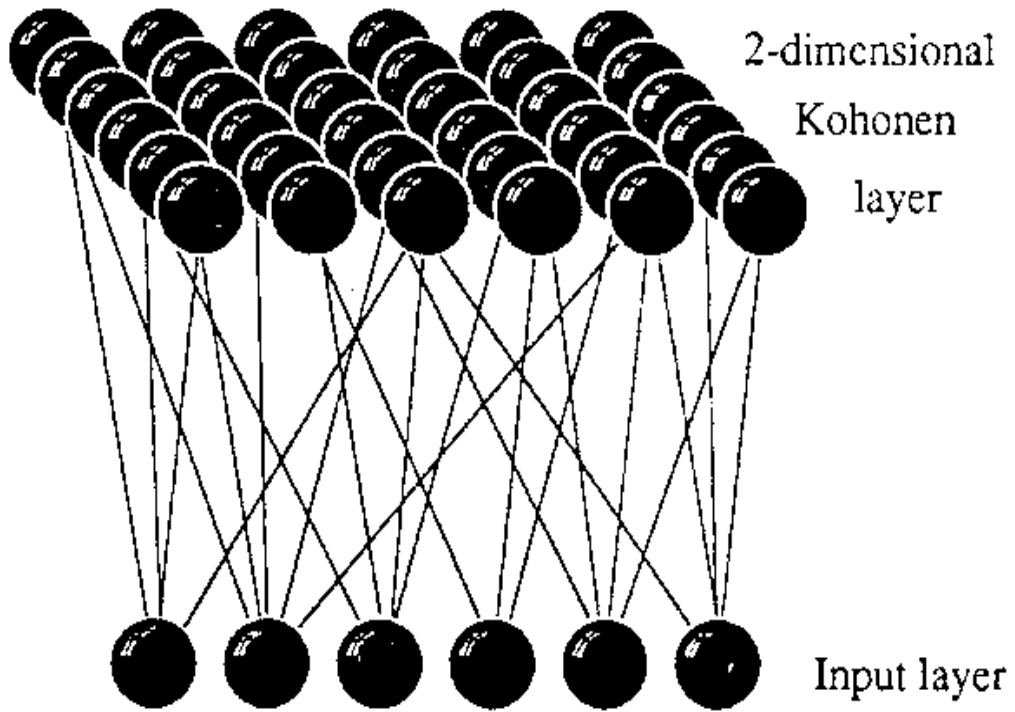
- Logistic
$$g(a) = \frac{1}{1 + e^{-a}}$$

- Hyperbolic tangent
$$g(a) = \tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}}$$

Different topologies

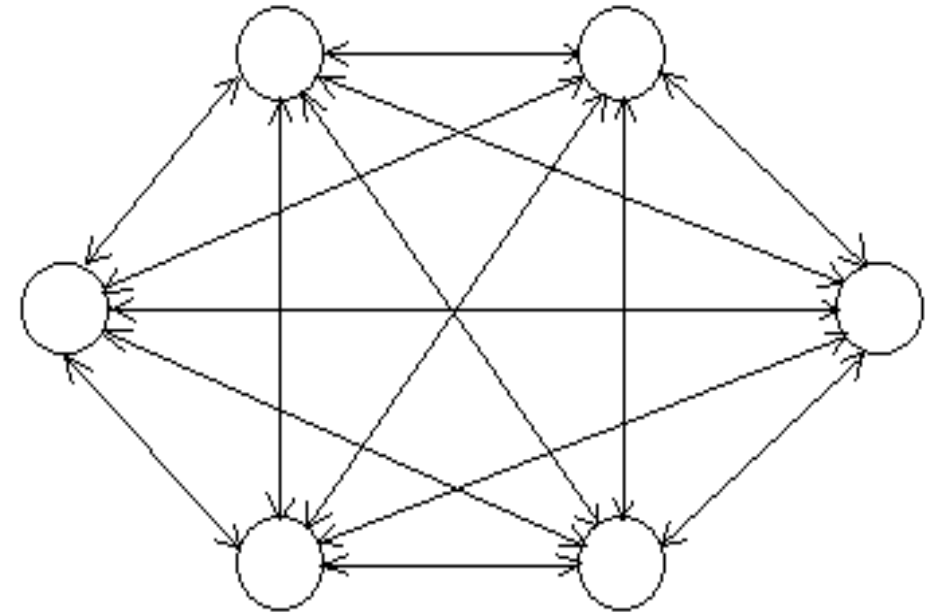


*Feed forward
neural networks*



Self Organizing Maps:
used for clustering

Reti di Hopfield:
the network evolves to
converge in a certain state



Taxonomy of classification methods

Method	Property	Comments
Template matching	Assign patterns to the most similar template.	The templates and the metric have to be supplied by the user; the procedure may include nonlinear normalizations; scale (metric) dependent.
Nearest Mean Classifier	Assign patterns to the nearest class mean.	Almost no training needed; fast testing; scale (metric) dependent.
Subspace Method	Assign patterns to the nearest class subspace.	Instead of normalizing on invariants, the subspace of the invariants is used; scale (metric) dependent.
1-Nearest Neighbor Rule	Assign patterns to the class of the nearest training pattern.	No training needed; robust performance; slow testing; scale (metric) dependent.
k-Nearest Neighbor Rule	Assign patterns to the majority class among k nearest neighbor using a performance optimized value for k.	Asymptotically optimal; scale (metric) dependent; slow testing.
Bayes plug-in	Assign pattern to the class which has the maximum estimated posterior probability.	Yields simple classifiers (linear or quadratic) for Gaussian distributions; sensitive to density estimation errors.
Logistic Classifier	Maximum likelihood rule for logistic (sigmoidal) posterior probabilities.	Linear classifier; iterative procedure; optimal for a family of different distributions (Gaussian); suitable for mixed data types.
Parzen Classifier	Bayes plug-in rule for Parzen density estimates with performance optimized kernel.	Asymptotically optimal; scale (metric) dependent; slow testing.
Fisher Linear Discriminant	Linear classifier using MSE optimization.	Simple and fast; similar to Bayes plug-in for Gaussian distributions with identical covariance matrices.
Binary Decision Tree	Finds a set of thresholds for a pattern-dependent sequence of features.	Iterative training procedure; overtraining sensitive; needs pruning; fast testing.
Perceptron	Iterative optimization of a linear classifier.	Sensitive to training parameters; may produce confidence values.
Multi-layer Perceptron (Feed-Forward Neural Network)	Iterative MSE optimization of two or more layers of perceptrons (neurons) using sigmoid transfer functions.	Sensitive to training parameters; slow training; nonlinear classification function; may produce confidence values; overtraining sensitive; needs regularization.
Radial Basis Network	Iterative MSE optimization of a feed-forward neural network with at least one layer of neurons using Gaussian-like transfer functions.	Sensitive to training parameters; nonlinear classification function; may produce confidence values; overtraining sensitive; needs regularization; may be robust to outliers.
Support Vector Classifier	Maximizes the margin between the classes by selecting a minimum number of support vectors.	Scale (metric) dependent; iterative; slow training; nonlinear; overtraining insensitive; good generalization performance.

Clustering

- Unsupervised classification, no known classes, unknown reference data
- Unknown number of classes
- One of the problems is the definition of a similarity criterion that is dependent on the data *and* the context
- Two main techniques
 - Hierarchical agglomerative
 - Iterative partitional

Taxonomy of clustering methods

Algorithm	Property	Comments
K -means	Identifies hyperspherical clusters; could be modified to find hyper-ellipsoidal clusters using Mahalanobis distance; computationally efficient.	Need to specify K and the initial cluster centers. Additional parameters for creating new clusters, merging existing clusters and outlier detection can be provided.
Fuzzy K -means	Similar to K -means except that every pattern has a degree of membership into the K clusters (fuzzy partition).	Need to specify K , initial cluster centers and cluster membership function.
Minimum Spanning Tree (MST)	Clusters are formed by deleting inconsistent edges in the MST of the data.	Need to provide the definition of an inconsistent edge.
Mutual Neighborhood	Compute the mutual neighborhood value (MNV) for every pair of patterns. If x_j is the p^{th} near neighbor of x_i and x_i is the q^{th} near neighbor of x_j , then $MNV(x_i, x_j) = p + q$; $p, q = 1, \dots, K$.	Need to specify the neighborhood depth, K .
Single-Link (SL)	A hierarchical clustering algorithm which accepts a $n \times n$ proximity matrix; output is a dendrogram or a tree structure; a single-link cluster is a maximally connected subgraph on the patterns.	Single-link clusters easily chain together and are often “straggly”; need a heuristic to cut the tree to form clusters (a partition).
Complete-Link (CL)	A hierarchical clustering algorithm which accepts a $n \times n$ proximity matrix; output is a dendrogram or a tree structure; a complete-link cluster is a maximally complete subgraph on the patterns.	Complete-link clusters tend to be small and compact which combine nicely into layer clusters even when such a hierarchy is not warranted; need a heuristic to form clusters (a partition).
Mixture Decomposition	Each pattern is assumed to be drawn from one of K underlying populations, or clusters; population parameters are estimated from unlabelled data.	The form and the number of underlying population (K) densities are assumed to be known; K can be estimated using a number of criteria (see Section 8.2).

Combinations of classifiers

- Problem relatively investigated
- Assumes:
 - several classifiers with different and sub-optimal performances;
 - different training sets;
 - different classifiers trained with equal training sets and therefore with different performances;
 - equal classifiers trained differently (NN) and that turn out to have different performances.
- The goal is to increase performance
- Approaches:
 - *parallel*: the results of the individual classifiers are combined;
 - *serial*: the results of one are input of the next one, until the final result;
 - *hierarchical*: the classifiers are tree-structured and the results are combined appropriately.

To summarize, about Machine Learning methods ...

- Statistical classification
 - Parametric model definition $M(w)$
- This implies
 - estimation of the complete distribution (joint, parameters and data), or the posterior distribution, or
 - optimal parameter estimation that maximizes the a-posteriori probability, or
 - estimation of marginal probabilities (related to some parameters) or expectations (predictions) related to the posterior probability.
- Everything can be seen as an optimization problem

- **Dynamic programming:** search for the shortest path in an appropriate graph with a defined metric
 - Smith-Waterman algorithm (BLAST)
 - Needleman-Wunch algorithm
 - Viterbi algorithm

- **Gradient descent:** search for the minimum of a function
 - parameter estimation

$$w^{t+1} = w^t - \eta \frac{\partial f(w)}{\partial w} \bigg|_{w^t}, \quad \eta \text{ learning rate}$$

$$f(w) = -\log P(w|D)$$
 - Numerous variants

- *Expectation-Maximization (EM) e Generalised EM*
 - Used when the model involves latent (hidden) variables, such as Hidden Markov Model (HMM)
 - Alternating E and M steps
 - Step E estimates the distribution of hidden variables (given the observations)
 - Step M, update parameters, given the distribution previously estimated

- *Markov-Chain Monte-Carlo, MCMC*
 - Derives from statistical physics
 - Estimation of the expected value (expectation) of a multidimensional probability distribution $P(x_1, \dots, x_n)$, where x_i can be parameters, hidden parameters or observed data
 - Sampling of such a distribution by building a Markov chain that has P as distribution in the equilibrium state
 - Possible algorithms: *Gibbs sampling, Metropolis algorithm*

- *Simulated annealing*

- Derived from statistical mechanics
- Combines MCMC with a cooling mechanism of the "system" in order to bring it to a more stable and robust state

- Genetic and evolutionary methods

- Derived from the theories of evolution
- Random changes (mutations) are imposed on the system and a *fitness* function is used to assess the quality of the mutation and possibly discard it
- Genetic algorithms, in addition to mutations, allow new generations of points (crossover)

In a nutshell..

- ... for the use of these learning algorithms, one must take into account
 - model complexity, *model selection*
 - training phase, *batch* or *online*
 - availability and type of training, testing and validation data
 - learning phase stop mechanisms (pb. *overfitting*)
 - evaluate the use of different models or the same type of model trained differently (classifiers ensemble)
 - data balancing
- Often, all these models are **adapted** to the specific problem/application to be addressed, ***ad hoc* methods and models** are often needed

APPLICATIONS

The rebirth of Pattern Recognition/Machine Learning

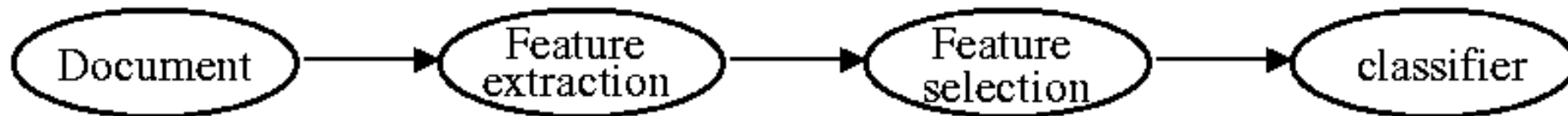
- Factors that have decreed the resurgence of pattern recognition in recent years:
 - increased computational capacity of computers
 - presence of large quantities of data, also distributed
 - new human-machine interaction systems

Typical applications

- Speech Recognition:
 - Problems:
 - tone
 - Voice/pitch
 - speed
 - mood
 - Application: remote (phone) information without the assistance of an operator (chatbot);
- Recognition of handwritten characters:
 - Problems:
 - handwriting
 - mood
 - Application: automatic CAP reading in letters

Classification of documents

- Definition:
 - classification of documents on the basis of the content topic (sport, economy, ...).
- Feature:
 - types of words, absolute and relative frequency
- Clustering
- Applications: research on the internet, data mining.



Audio

- Speech vs. music vs. noise
- Speech recognition
- Speaker recognition
- Voice Activity Detection (VAD)
- Turn taking
- Sentiment analysis



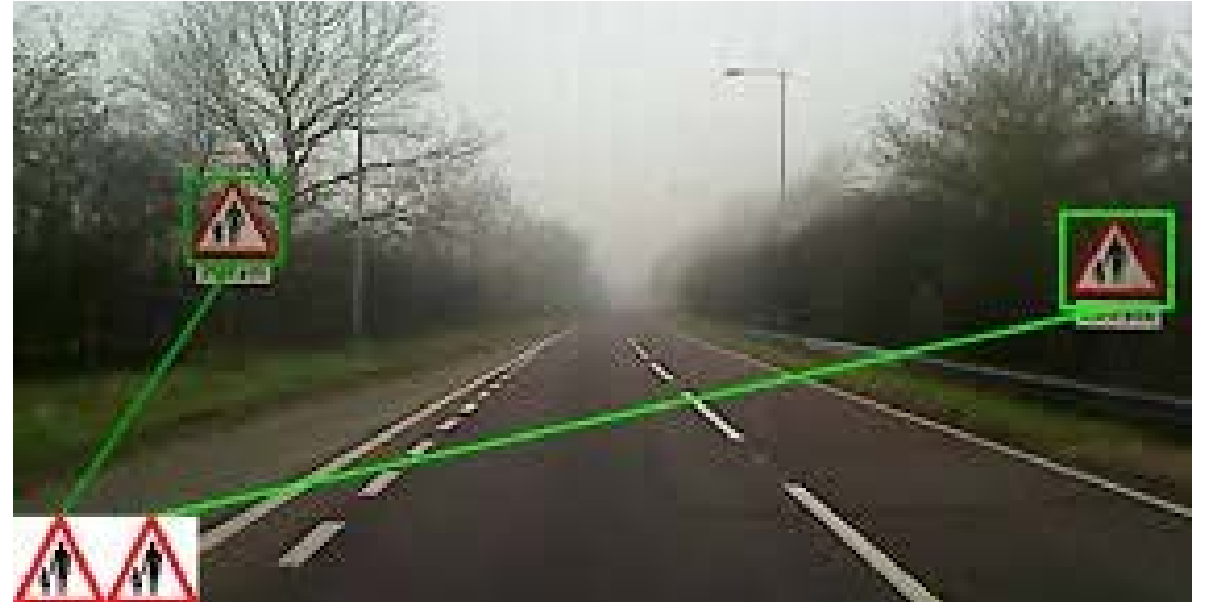
Visual inspection



Visual inspection

- <https://youtu.be/UY6xbrcViVw>
- <https://youtu.be/L7LtNabIZw0>

Autonomous driving



Autonomous driving

- <https://youtu.be/xMH8dk9b3yA?t=40>
- Kitti dataset: https://youtu.be/KXpZ6B1YB_k

Data Mining

- Definition:
 - knowledge extraction from a (typically very large) set of multidimensional data.
- Purposes: prediction, classification, clustering, association analysis, etc.
- Note that usually the data used for Data Mining has been collected for another purpose, other than Data Mining.
- *Example*: given a set of consumers, group them according to similar purchase behaviour.

Image retrieval by content

- Definition:

- *image retrieval*: find, in a database, images or sequences of images responding to a given query;
- *by content*: the search is based on the content of the image, no longer based on text (manually annotated on all images in the data set) or meta-data;

- Examples of *query*:

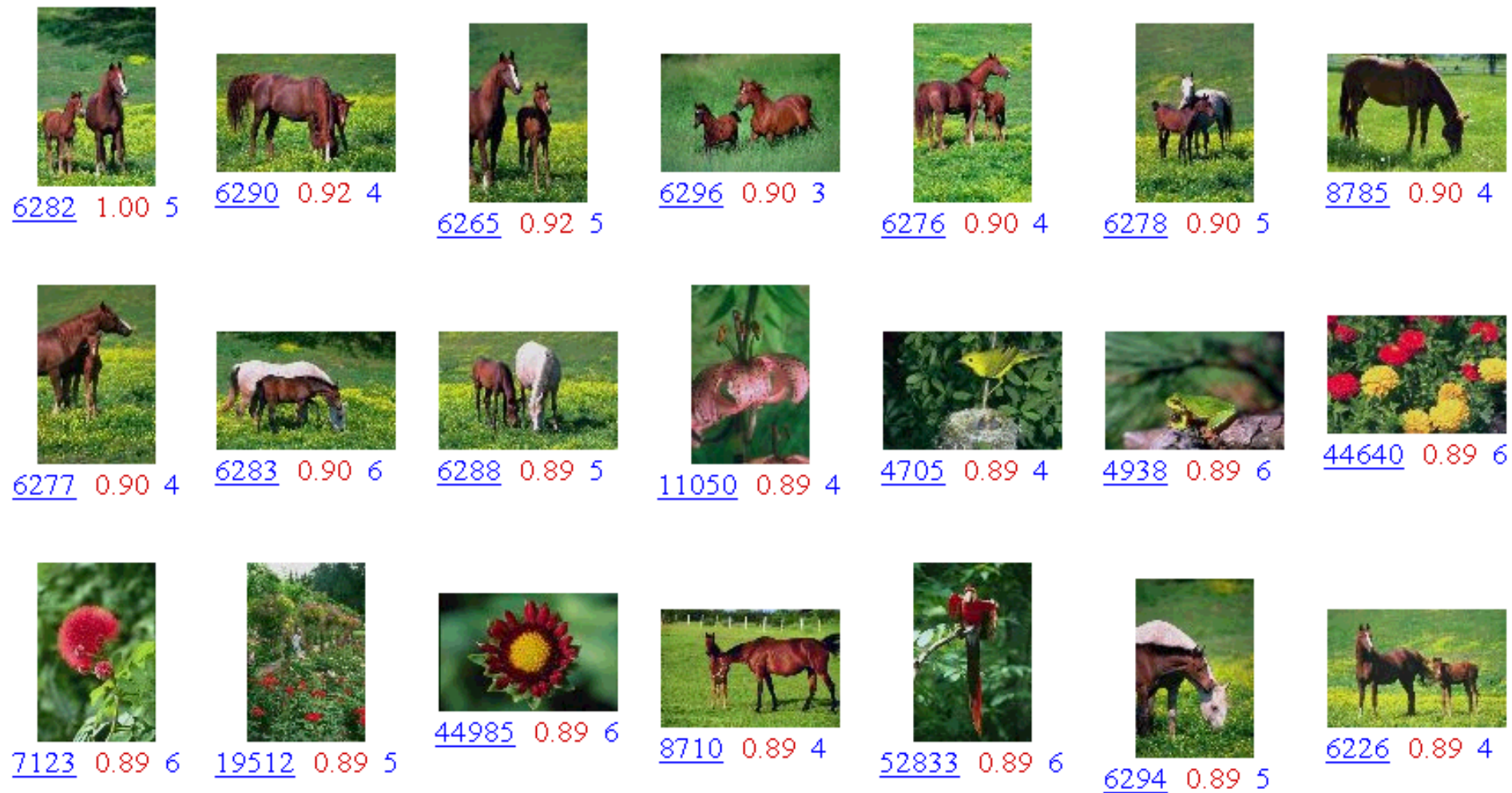
- "Find me all the images similar to a given image".
- "Find me all the pictures that contain a horse".

Example

Main Image Class ->	Photographs	Graphs
Option 1	Click <i>Random</i>	Click <i>Random</i>
Option 2	Query image URL or ID <input type="text"/>	Query image URL or ID <input type="text"/>
Option 3	Start with 	Start with 

Search for all photos similar to this

Result: there is also a score on the reliability of the retrieval



Gesture recognition

- System that identifies human gestures and uses them to carry information, or to control devices.
- Instruments:
 - based on gloves/trackers that keep track of the trajectory;
 - based on computer vision systems that retrieve the trajectory from stereo information (with or without markers).

Tracking del corpo

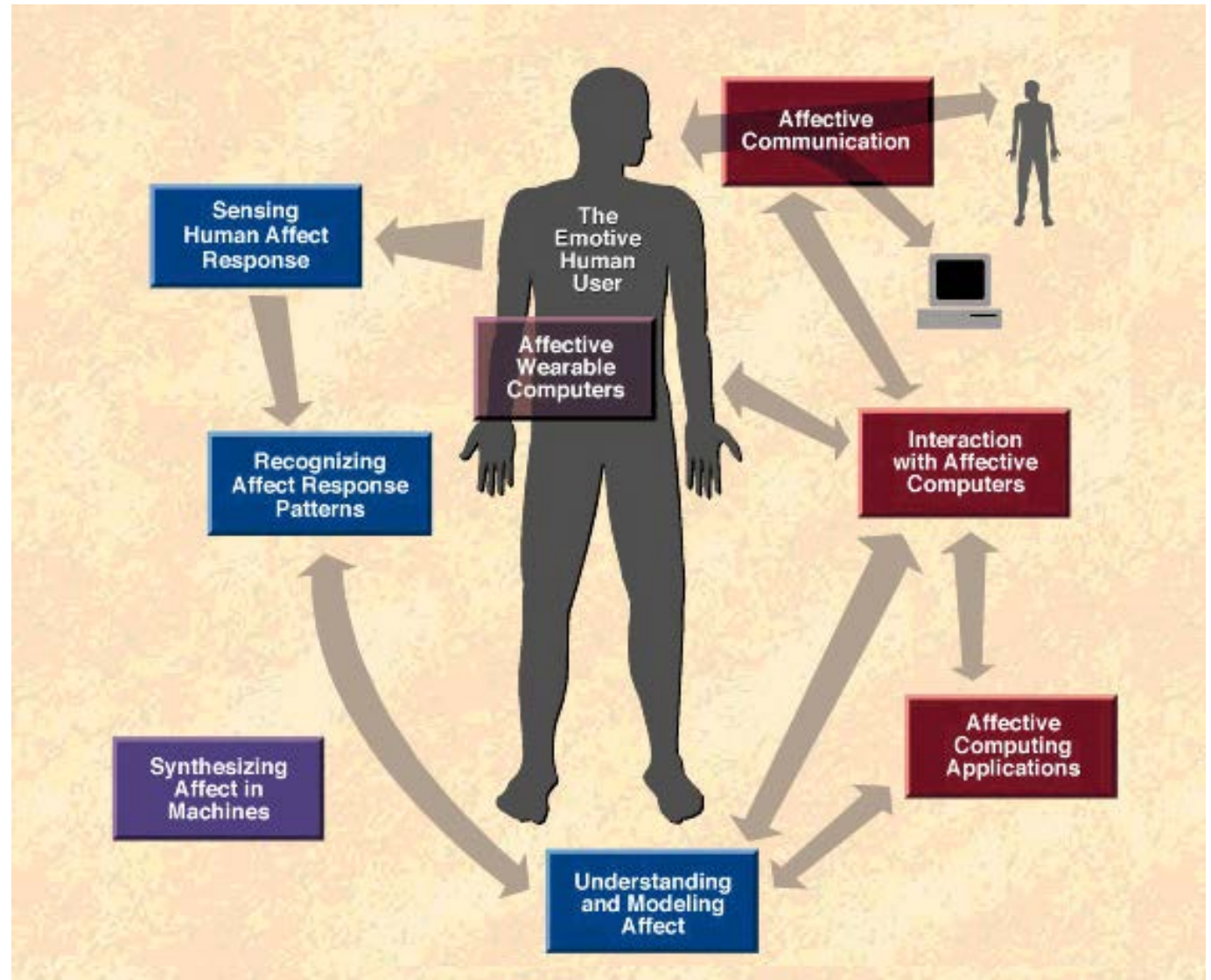


Recent evolution...



Affective Computing

<https://youtu.be/sRh8AUakO90>



http://affect.media.mit.edu/AC_research/

■ *Sentic modulation*

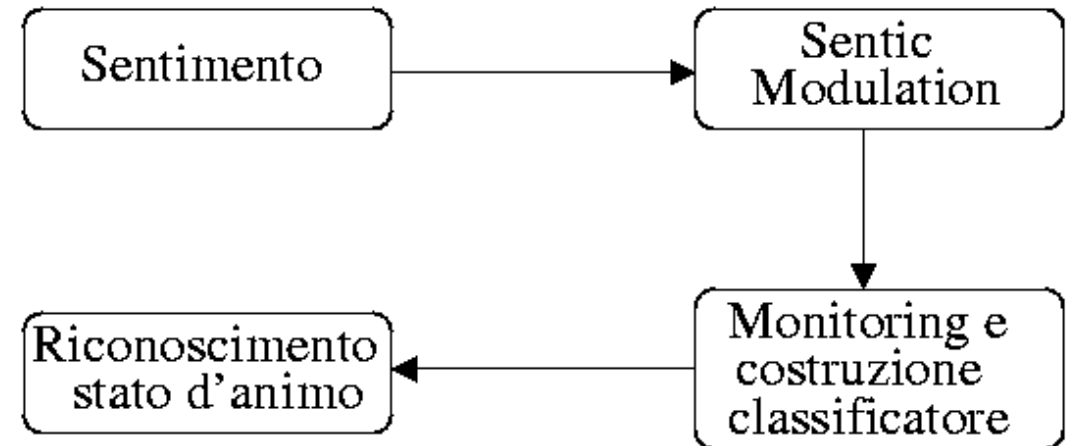
- physical expression of a feeling: inflection of voice, facial expressions, heartbeat, posture, gesture.

■ Problems:

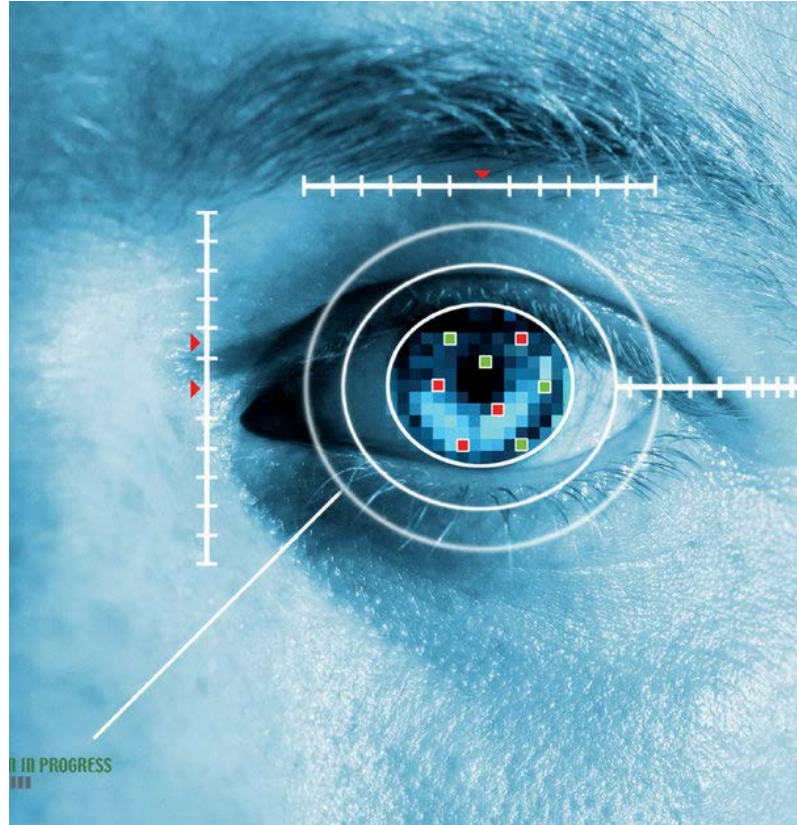
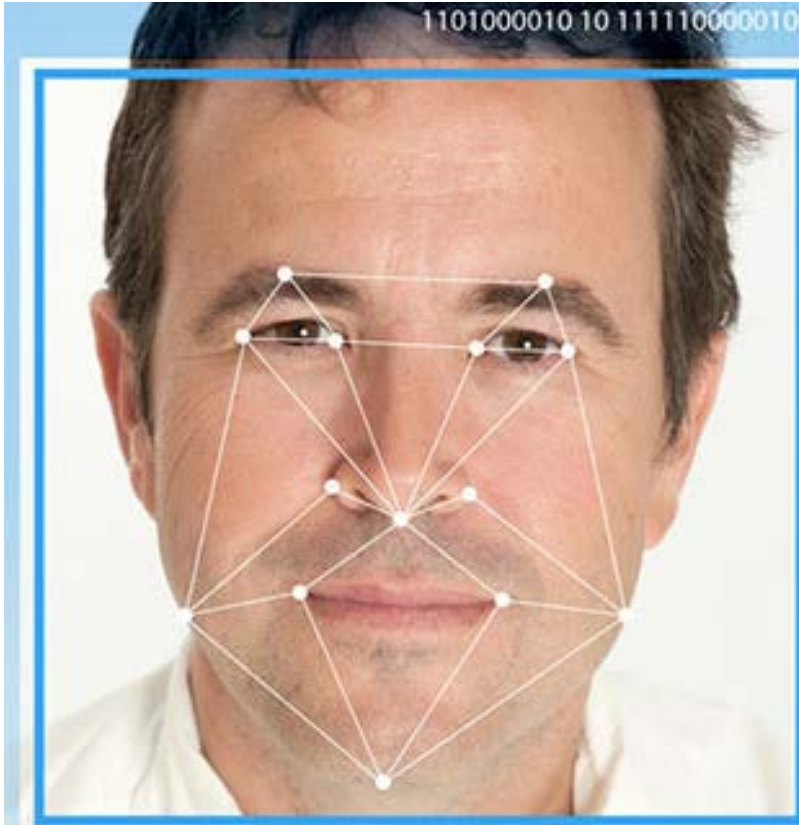
- *monitoring sentic modulation*;
- context;
- free expression sentic modulation.

■ Applications:

- expressive emails;
- video compression of faces.



Biometrics



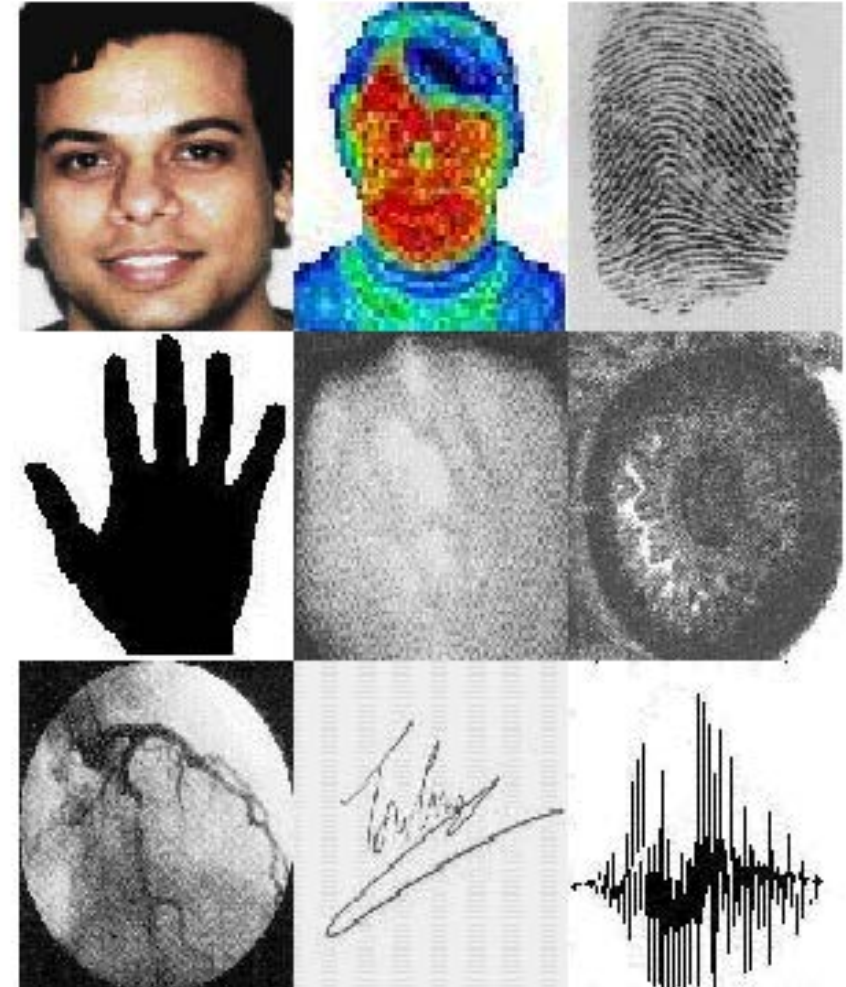
Biometrics

- Definition:
 - identification of persons through the analysis of their physiological and/or behavioural characteristics.

- Features of the biometric factor:
 - universal (present in every individual);
 - unique (different in each individual);
 - permanent (not removable);
 - quantifiable (measurable).

Biometrics

- Biometric factor:
 - face
 - facial thermogram
 - fingerprinting
 - hand geometry
 - signing
 - voice
 - iris
- Evaluation of a biometrics system:
 - performance;
 - safety;
 - acceptability.



<http://biometrics.cse.msu.edu/>

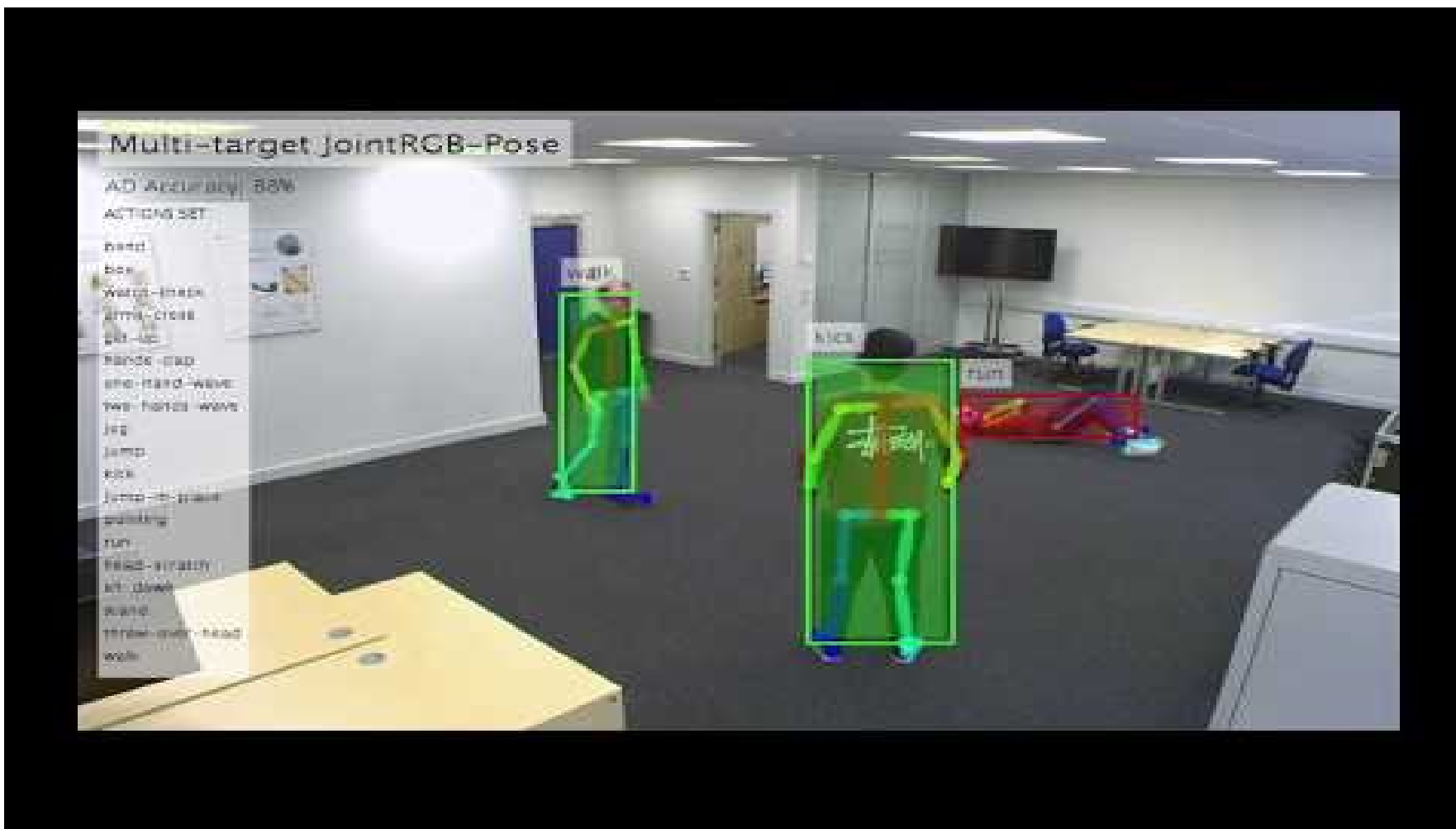
Biometrics

- <https://youtu.be/aE1kA0Jy0Xg>

Action/activity classification

- Analysis of video sequences:
 - objects tracking
 - tracking
 - classification of trajectories;
 - recognition of behaviours.
- Applications:
 - video surveillance;
 - traffic analysis.

Examples



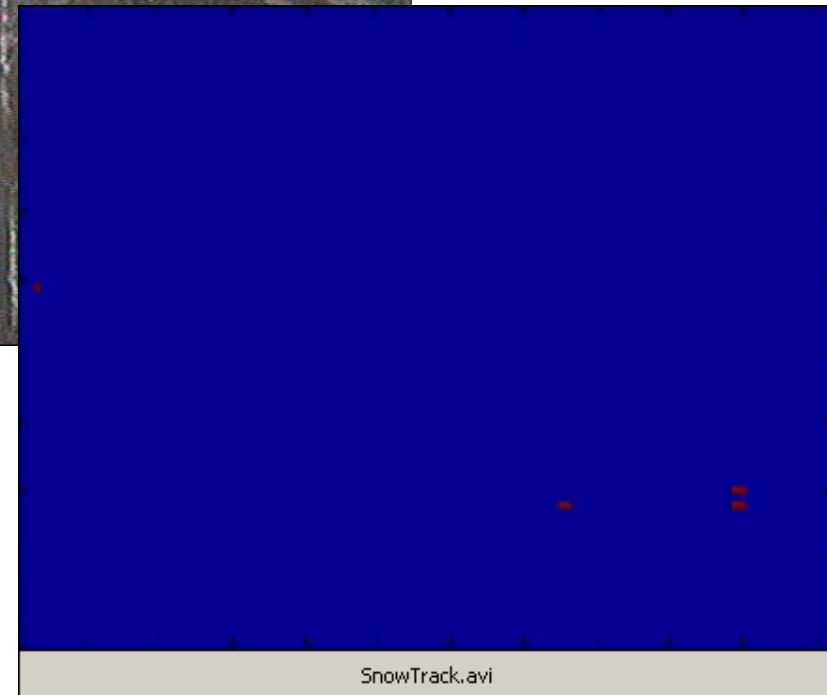
Examples

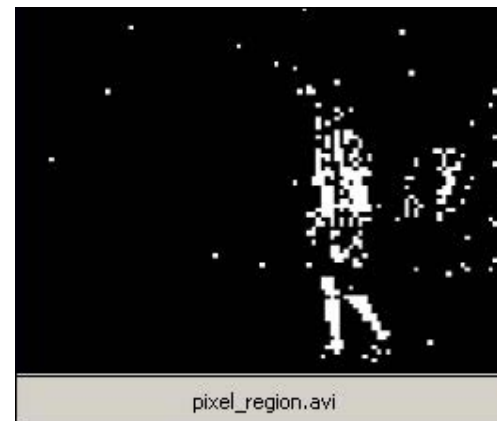
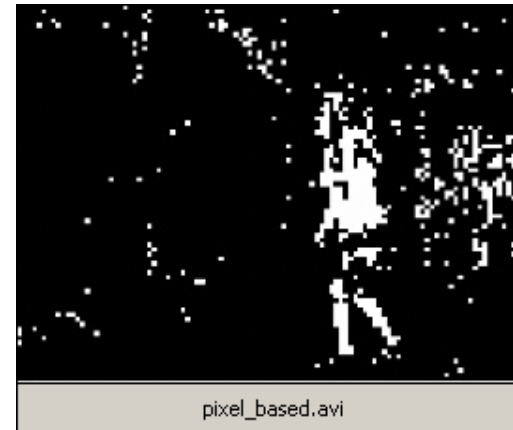
- https://youtu.be/hs_v3dv6OUI
- <https://youtu.be/PEziTgHx4cA>

Multi-object tracking









Bioinformatics

- Goal:
 - implementation of techniques capable of automatically extracting as much knowledge as possible from a DNA sequence;
 - the definition has nowadays a very broad meaning.

- Basis of scientific starting:
 - *Genome*: set of the genetic heritage of an organism.
 - *Chromosome*: superorganized structure of DNA.
 - *DNA*: double helix formed by a sequence of nucleotide bases.
 - *Gene*: DNA sequence encoding a protein.
 - *Protein*: a molecule that performs a specific function in the body.

Problems

■ *Identification of genes*

- given a DNA sequence it is necessary to determine whether it represents a gene or not
- 2 approaches: gene model/database comparison.

■ *Classification of genes*

- given a DNA sequence recognised as a gene, it should be classified on the basis of the function it expresses.
- 2 approaches: gene-based (alignment) and protein-based (protein/protein threading models).

■ *Pattern Discovery*

- discover unusual, rare and significant patterns.

Frontiers of Pattern Recognition

Topic	Examples	Comments
Model selection and generalization	Bayesian learning, MDL, AIC, marginalized likelihood, structural risk.	Make full use of the available data for training.
Mixture modeling and EM algorithm	Clustering density estimation.	Soft membership; better than k -means clustering.
New objective functions for classification	Maximum margin (SVMs), regularized cost.	Provide low VC dimension and good generalization.
Optimization methods	Quadratic programming; linear programming.	Leads to support vectors; built-in feature selection.
Local decision boundary learning	SVMs, Boosting, mixture of local experts.	Focus on boundary patterns.
Sequential pattern recognition	Hidden Markov Models (HMMs), recurrent networks.	Successfully applied to speech and handwriting recognition.
Local-invariant (dis)similarity measures	Deformable template matching, tangent distance.	Invariant to local distortions.
Independent component analysis	Blind source separation, feature extraction.	Extract statistically independent components.
Combining multiple classifiers	See Table 7.	Improve recognition accuracy.
Emerging applications	Data mining and KDD, Document categorization, Image database retrieval, Financial forecasting, Biometric recognition (fingerprint, iris, face, voice, handwriting and signature).	Large volume, high dimension, mixed data types, missing data, data modeling, model selection.

Frontiers of Pattern Recognition/Machine Learning

- Learning with **scarce** data (long tail distribution)
- **Zero- and few-shot** learning
- **Domain adaptation**
- **Disentangling representations**, fairness
- **Multimodal** learning
- **Unsupervised** learning
- **Self-supervised** learning
- **Meta-learning** (learning to learn)
- **Continual, lifelong learning**