

Università di Verona

A.A. 2020-21

Machine Learning & Artificial Intelligence

Introduzione alla
Pattern Recognition & Machine Learning & AI

Vittorio Murino

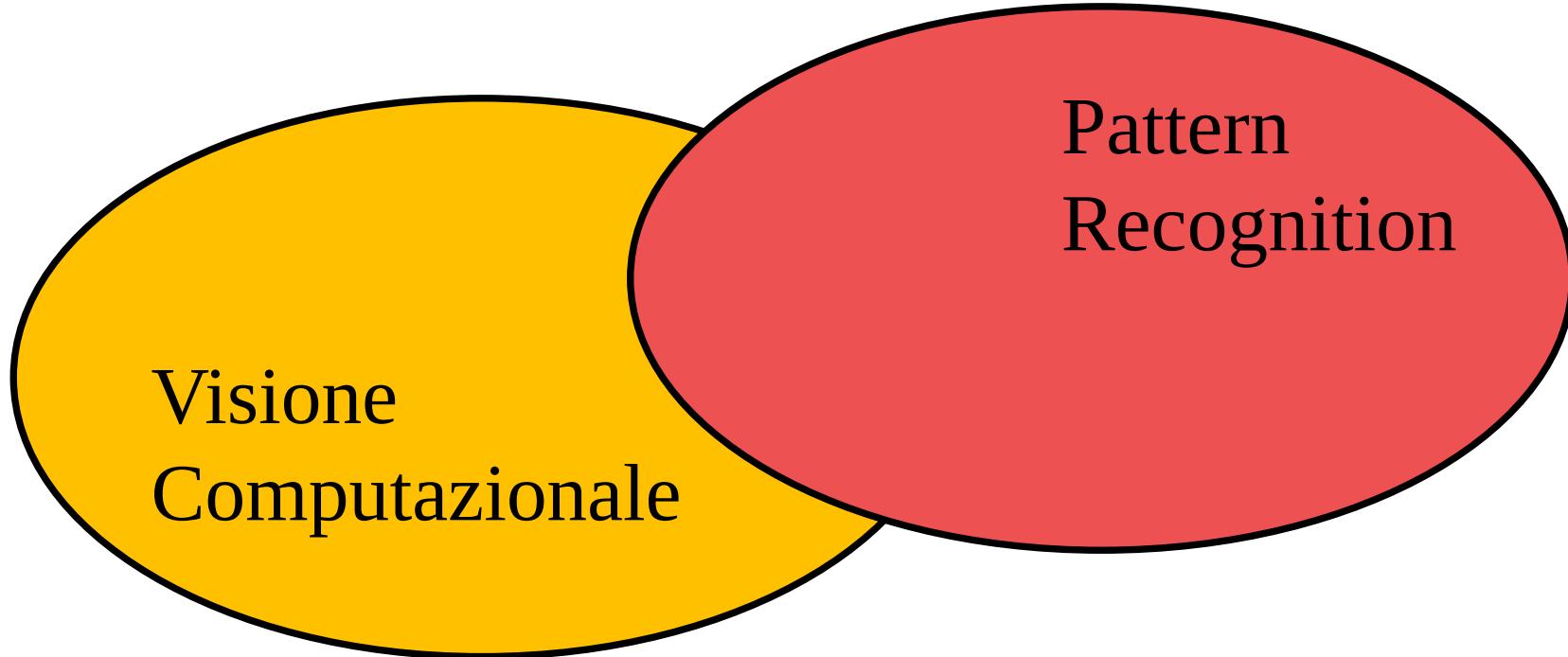
Inquadramento

- Sistemi di Pattern Recognition nell'uomo:
 - riconoscere la faccia di una persona conosciuta, anche se questo cambia pettinatura, ha gli occhiali da sole, ...
 - capire quello che una persona sta dicendo, anche se varia il tono della voce;
 - leggere una lettera scritta a mano;
 - ...
- Attività che l'essere umano risolve in modo molto naturale, per un calcolatore hanno invece notevole complessità

Alcune possibili definizioni

- *Pattern recognition*
 - studio delle problematiche connesse all'utilizzo dei calcolatori per il riconoscimento automatico di dati, altrimenti detti *pattern*.
- Studio di come le macchine possono osservare l'ambiente, imparare a distinguere i pattern di interesse dall'informazione di sfondo e prendere decisioni relative alla categoria dei pattern.
- *Sistema di Pattern Recognition*: il processo che prende in input dati grezzi (*raw*) ed effettua un'azione sulla base della “categoria” dei dati.

Pattern Recognition e Visione Computazionale



Il riconoscimento può essere visto come un problema di classificazione in cui le classi sono note o meno (e sono stimate dai dati)

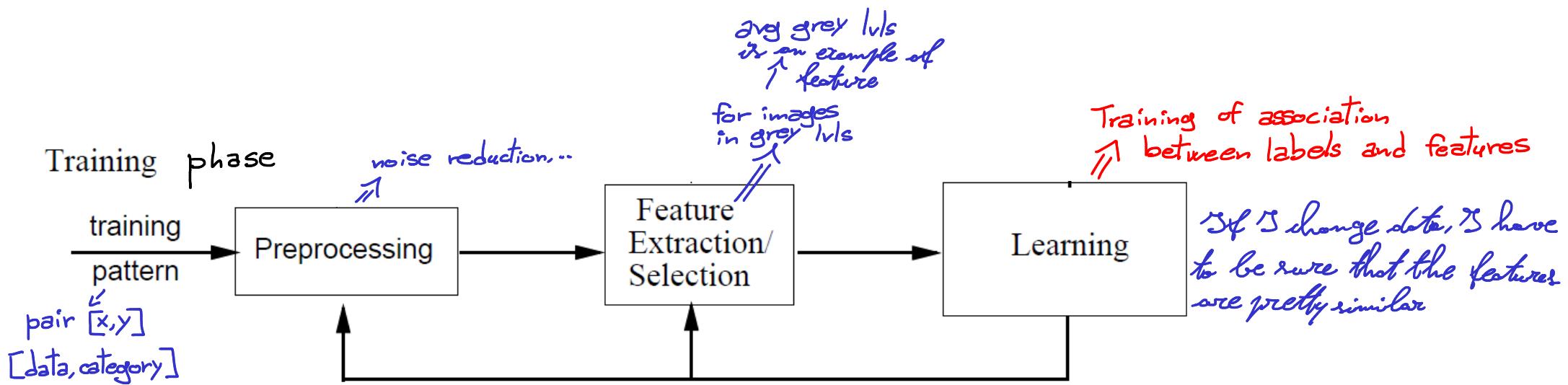
Supervised $\xrightarrow{\text{?}}$ *unsupervised*

Tabella 1: Applications

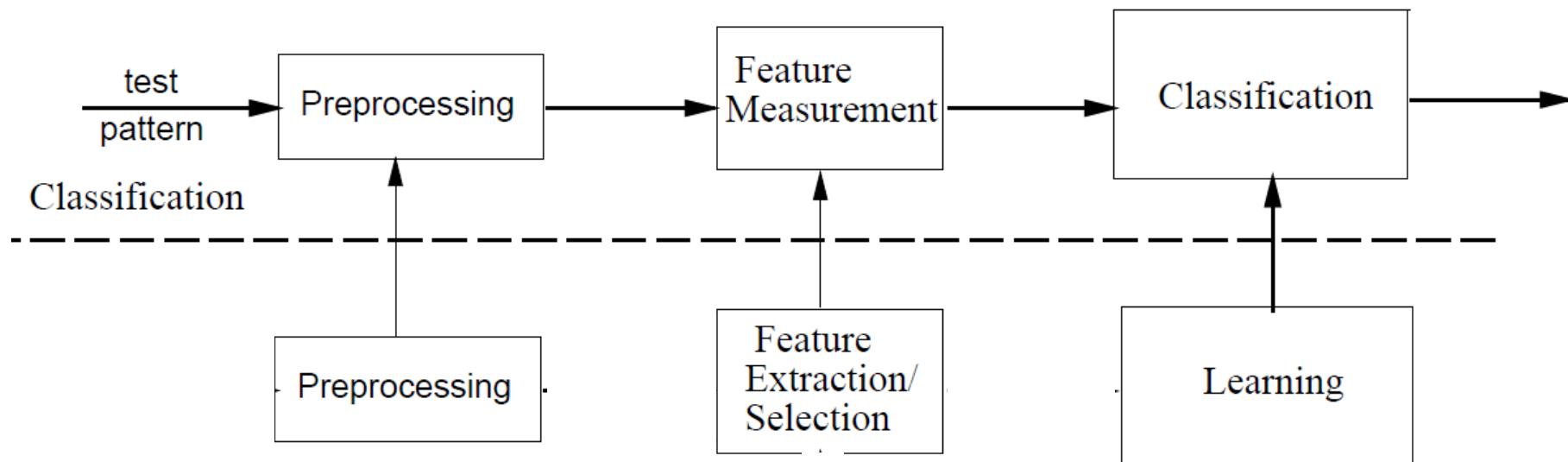
| Problem Domain | Application | Input Pattern | Pattern Classes |
|----------------------------------|---|---------------------------------------|--|
| Bioinformatics | Sequence Analysis | DNA/Protein sequence | Known types of genes/ patterns |
| Data mining | Searching for meaningful patterns | Points 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 |

Una punto comune di tali applicazioni è che le caratteristiche disponibili (*feature*, nell'ordine di migliaia) non sono suggerite da esperti ma devono essere estratte e ottimizzate da procedure *data-driven*

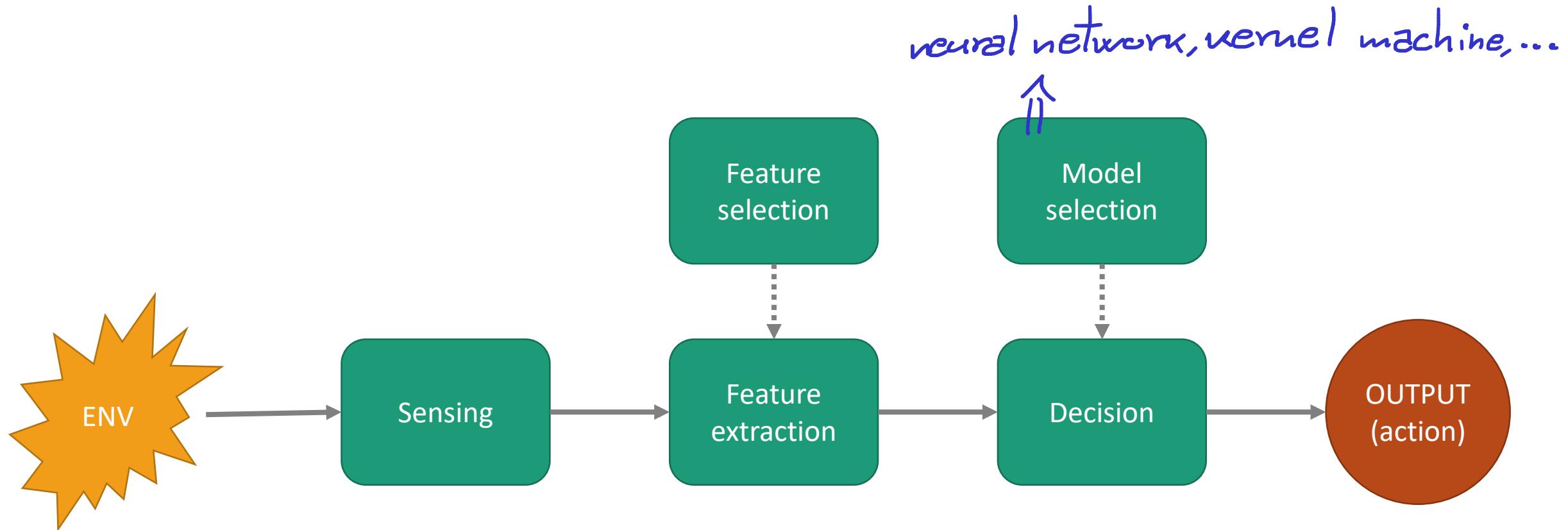
■ Modello di PR statistico



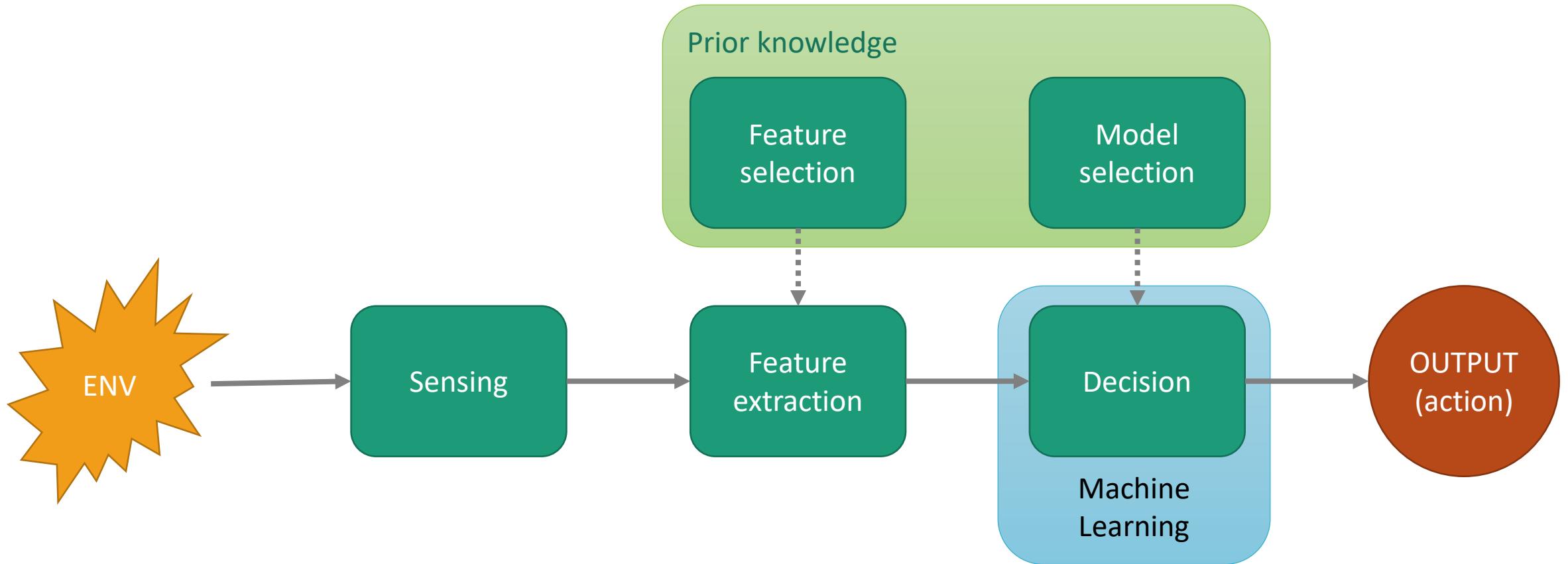
- Modello di PR statistico



Pattern recognition system

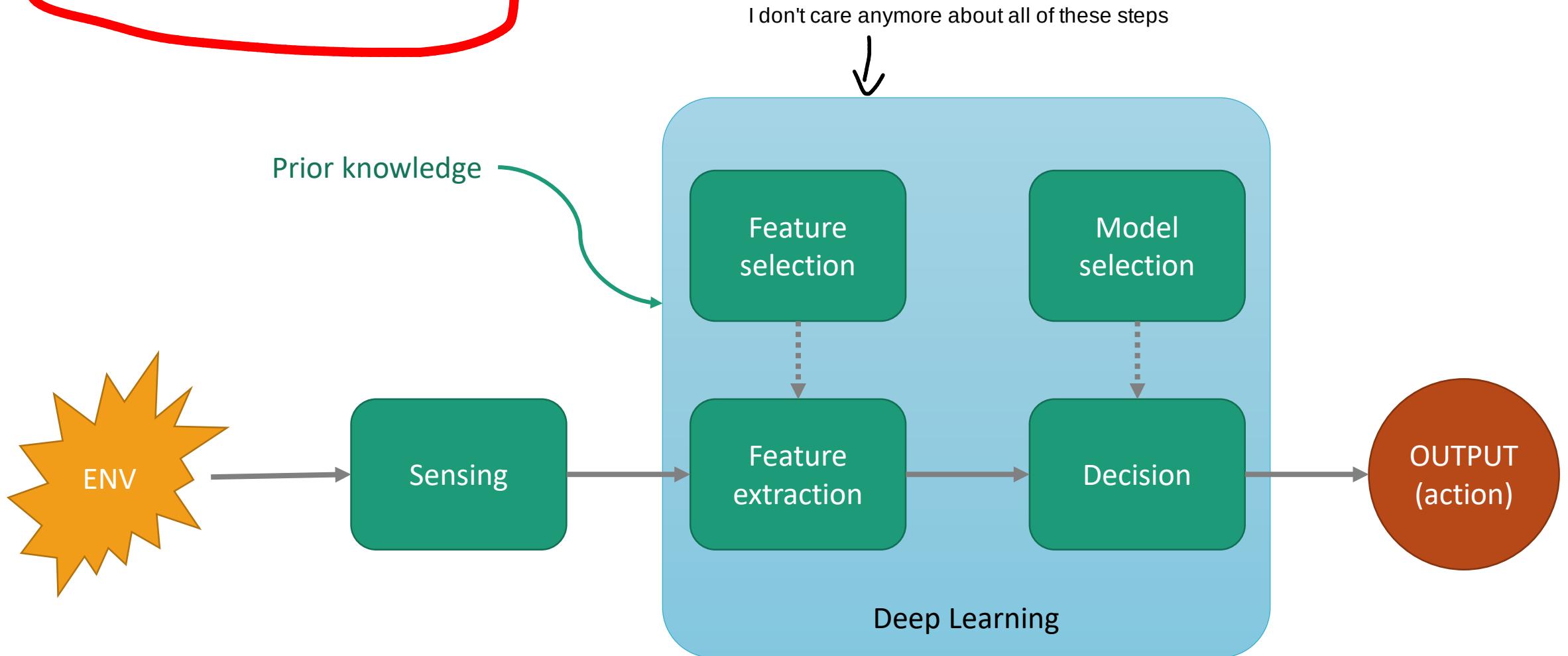


Pattern recognition systems: Traditional approach

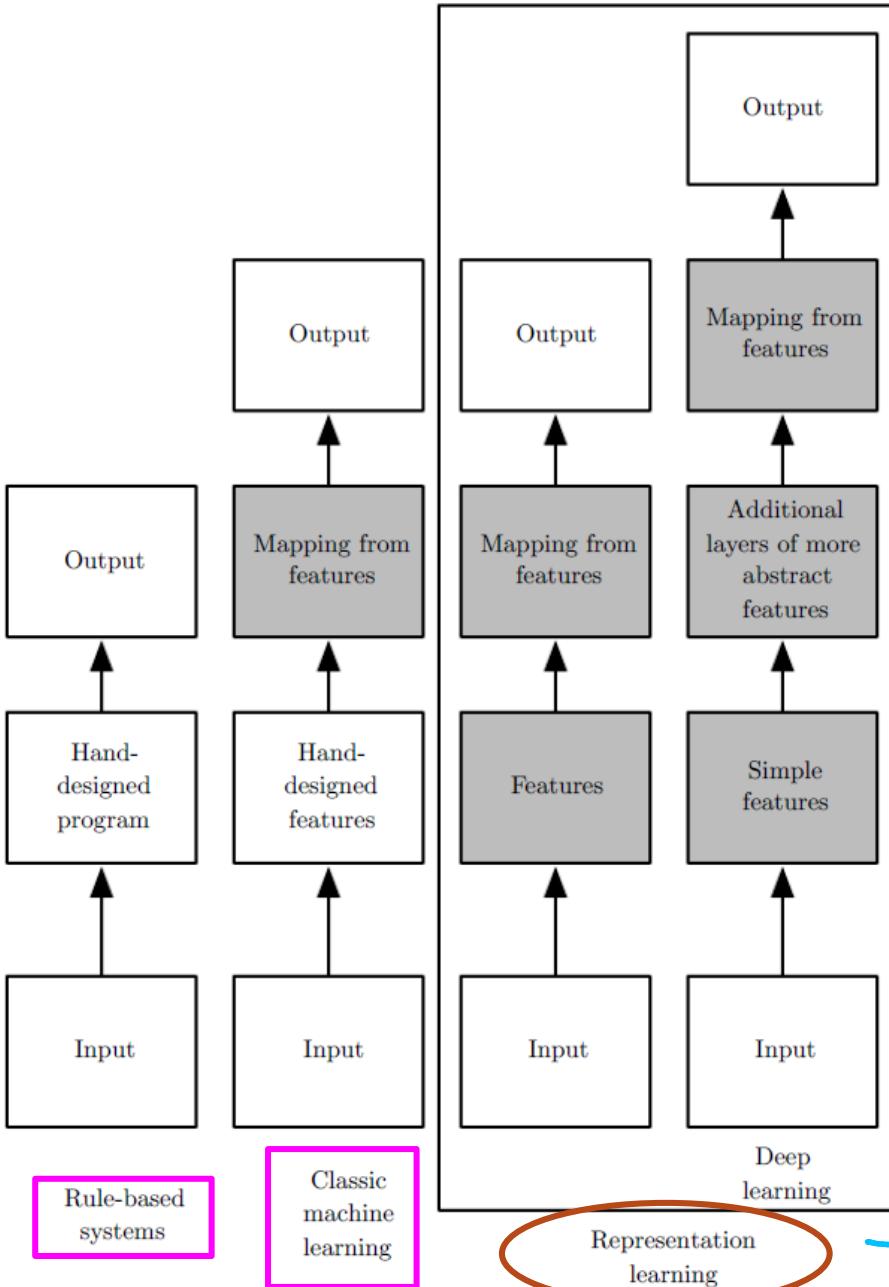


Pattern recognition systems:

Deep learning



Deep Learning



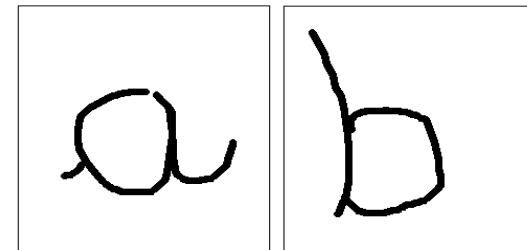
Higher abstraction layers mean increasing structured learning
E.g.: first level is recognition of eyes, nose, mouth
second level is to position them

Different abstract levels

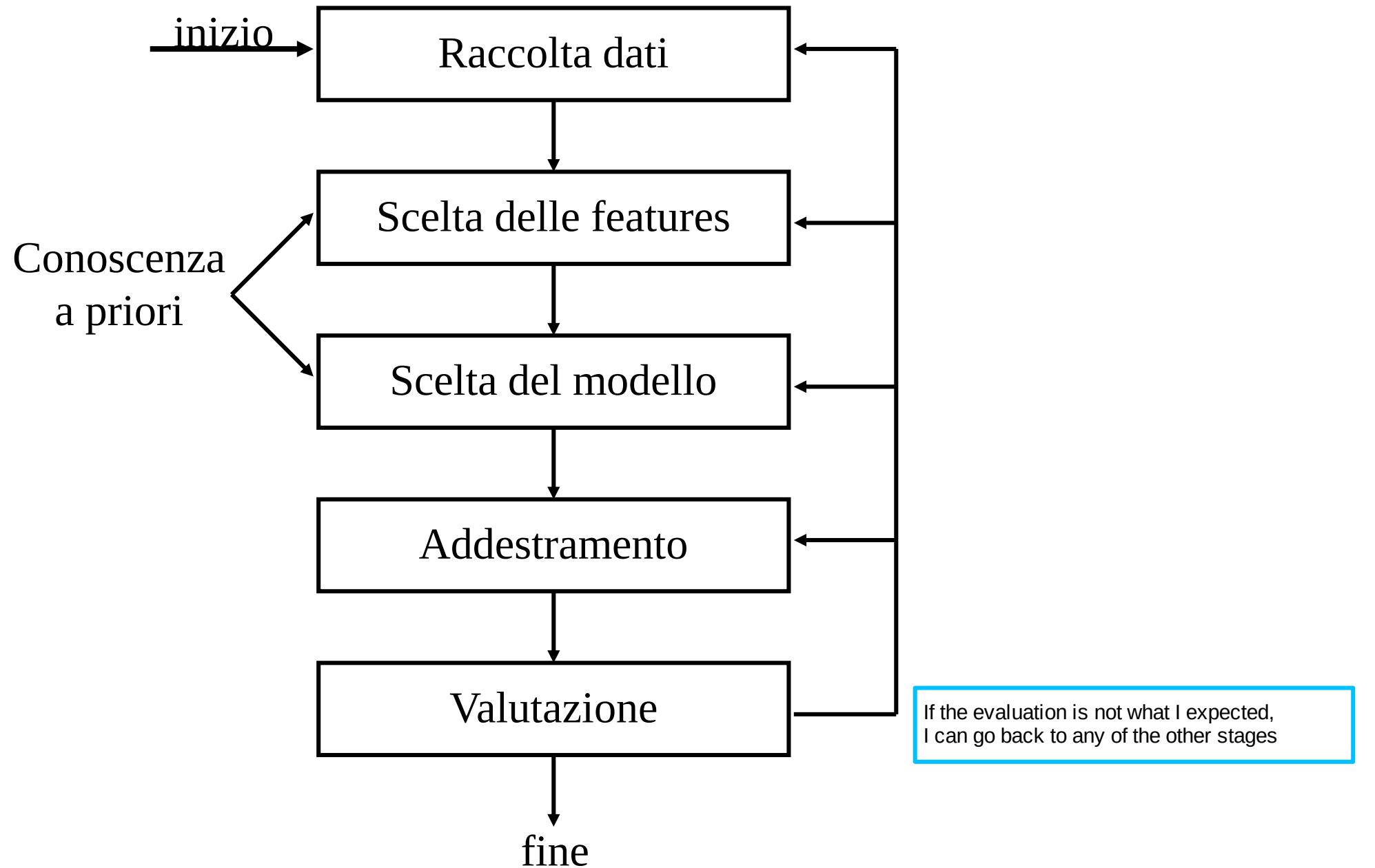
Deep learning can also learn the representation of the input data used in other classes, it can extract features that can be transferable to achieve other tasks

Sistema di Pattern Recognition

- Raccolta dati
- Scelta delle feature
- Scelta del modello
- Addestramento del modello
- Valutazione



Esempio guida: sistema che distingue tra le lettere scritte a mano “a” e “b”.



Raccolta Dati

dovremmo avere un numero sufficiente di istanze di coppie [dato, categoria] della cosa da imparare

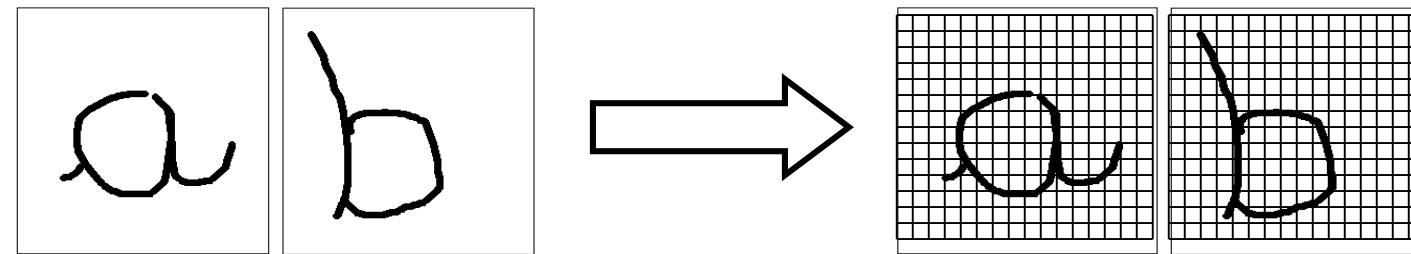


- Collezione di un insieme “**sufficiente**” e “**rappresentativo**” di esempi dal problema in esame.
 - “**sufficiente**”?
 - “**rappresentativo**”?
 - Problemi di sensoristica (risoluzione, banda, ...)
- se il dato è sempre lo stesso o varia di pochissimo, non può imparare molto
(se deve imparare a riconoscere la lettera 'a', non posso dargli in pasto la lettera scritta dalla stessa persona e basta, devo raccogliere un gran numero di lettere scritte da persone diverse)

Il deep learning funziona bene perché ho milioni di dati etichettati!

Normalmente sono umani che li etichettano, quindi questo può costare molti soldi e tempo!

- *Esempio:* un insieme di immagini contenenti le lettere “a” e “b” viene acquisito tramite una telecamera, e memorizzato nel computer



L’immagine viene rappresentata da un array di pixel, ogni pixel assume valore compreso tra 0 (completamente bianco) e 1 (completamente nero)

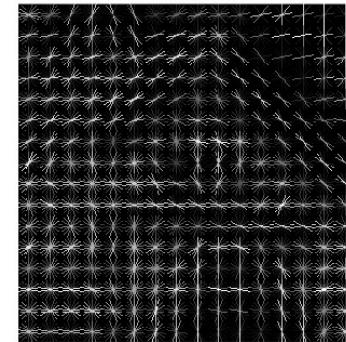
Scelta delle feature

- Non si possono utilizzare i dati così come sono (immagine 256x256 sono 65536 pixels)
- Feature: caratteristiche misurabili del fenomeno in esame (pattern = vettore di features):

le feature non devono essere troppo sensibili a piccole variazioni

- semplici da calcolare; → non posso spendere ore di tempo di calcolo solo per estrarre le feature
- invarianti a trasformazioni irrilevanti;
- affidabili; → tutte le feature presentano le stesse caratteristiche
- indipendenti; → non ci sono feature che dipendono da altre
- discriminanti; → le feature hanno un impatto pratico sul mio task, sono utili per la classificazione
- poche (problema della *curse of dimensionality*);

- In questa fase è molto utile l'**utilizzo della conoscenza a priori sul problema**



- *Esempio:*
 - ad esempio una feature potrebbe essere il numero totale di pixel neri:
 - invariante alla rotazione e traslazione dell'oggetto
 - poco discriminante: non tiene conto della forma
 - uso di conoscenza a priori: devo distinguere tra “a” e “b”, e so che la lettera “b” è tipicamente più alta e allungata della “a”.
 - uso come feature il rapporto altezza/larghezza

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.

Scelta del modello

- Scelta della struttura logica e la base matematica delle regole di classificazione.
- Tipicamente, il classificatore stima, per ogni oggetto, un valore che indica il grado di appartenenza ad una o più classi sulla base del vettore di feature che lo caratterizza.



Non solo la classe, ma anche il livello di appartenenza ad essa
Non vogliamo solo la risposta secca sì o no

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

IMPORTANT ↗

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.

Scelta del modello

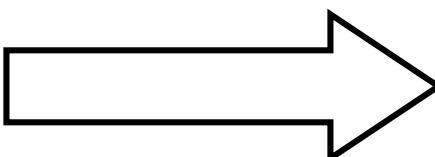
- Non esiste un classificatore che va bene per tutte le applicazioni
- *Esempio:* uso di un classificatore a soglia:
 - o data un'immagine I
 - o calcolo il rapporto altezza/larghezza $R(I)$;
 - o se $R(I)$ è maggiore di una certa soglia θ , allora l'immagine è una “b”, altrimenti è una “a”.

Addestramento del modello

- Sinonimi:
 - o training del classificatore
 - o learning del classificatore
- Processo con il quale si utilizzano i dati a disposizione (*training set*) per la costruzione del modello

Esempi tratti dal
problema
(*Training Set*)

Conoscenza a
priori



Regole che
governano il
fenomeno

Addestramento del modello

Esempio

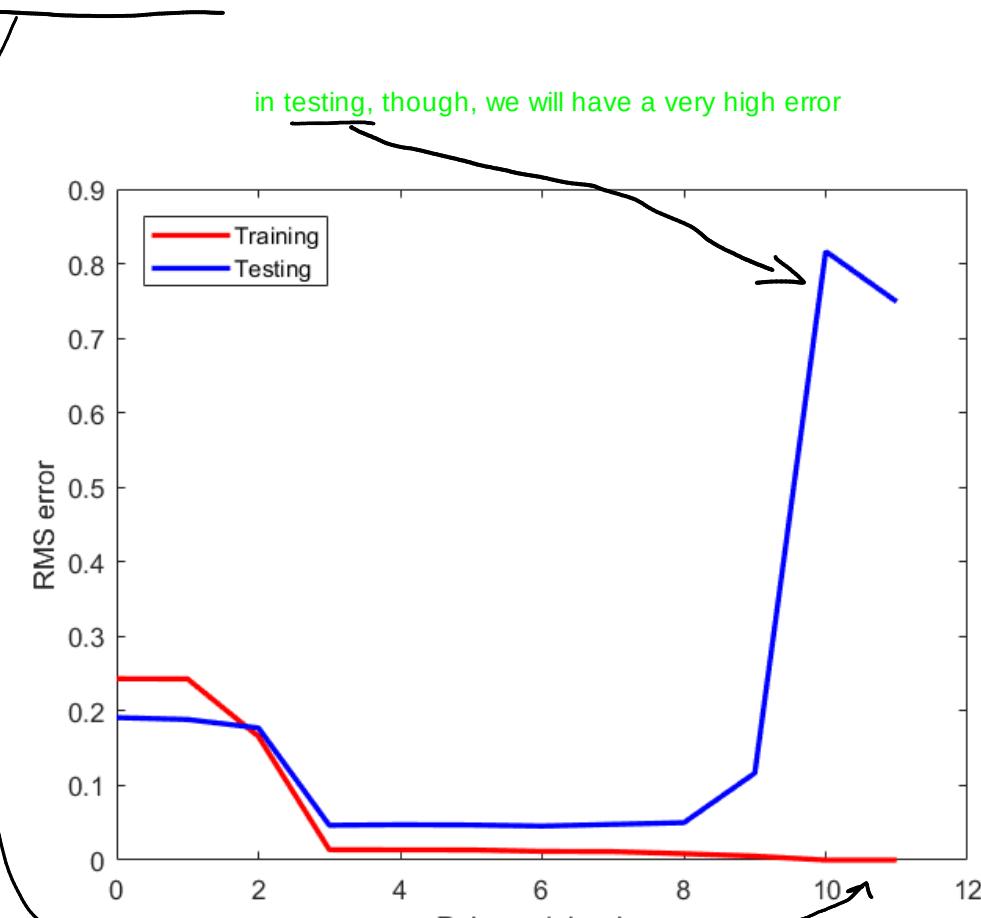
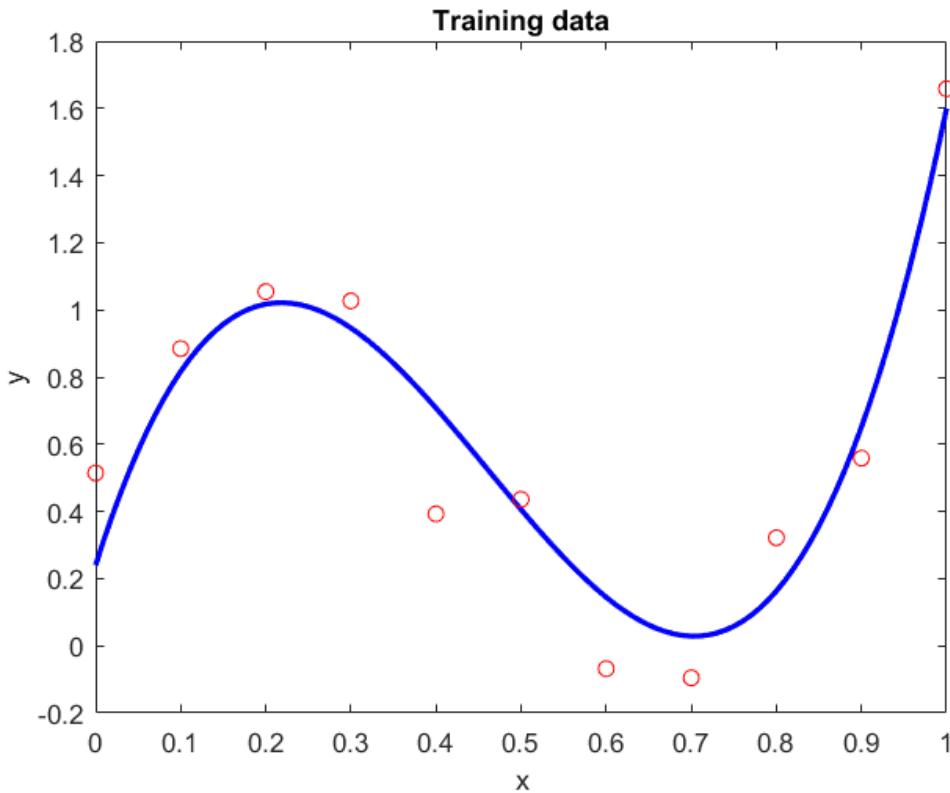
- Addestramento del modello = determinazione della soglia θ
 - Si ha a disposizione una serie di immagini di esempio per le lettere “a” e per le lettere “b” (*training set*)
 - calcolo $R(I)$ per tutte le immagini del training Set
 - determino una soglia θ “adatta” a separare i valori $R(I)$ calcolati

Overfitting

in this case the complexity of the polynomial is much higher than the training points

training too much on the training set and we are not generalizing,
but memorizing, so that IN TRAINING we have no error

full example on recorded lesson



Addestramento supervisionato

- Sinonimi: *supervised learning, classificazione*
- Idea e scopo:
 - di ogni elemento del training set si conosce l'esatta categoria.
 - L'obiettivo è quello di creare uno strumento in grado di classificare nuovi oggetti.
- Problemi:
 - capire se un algoritmo di training è capace di trovare la soluzione ottimale;
 - capire se converge, e se è sufficientemente scalabile;
 - capire se riesce a prediligere soluzioni semplici.

- *Esempio:*
 - o il training set è costituito da un insieme di immagini “a” e “b”.
 - o di ogni immagine conosciamo l’esatta classificazione (cioè se è “a” oppure “b”)
 - o queste informazioni sono utilizzate per determinare la soglia del classificatore.

Addestramento non supervisionato

- Sinonimi: *unsupervised learning, clustering*
- Idea e scopo:
 - nessuna informazione sulla categorizzazione degli elementi del training set.
 - Il sistema deve trovare i clusters (gruppi) “naturali” all’interno del training set, sulla base della “similarità” tra patterns
- Problemi:
 - intrinsecamente più difficile della classificazione
 - “naturali”?
 - “similarità”?

Addestramento non supervisionato

- *Esempio:*
 - il training set è costituito da un insieme di immagini “a” e “b”.
 - nessuna informazione sulla categorizzazione delle immagini.
 - si cercano di creare due gruppi, mettendo assieme quelle immagini che hanno valore simile di $R(I)$ (la *feature*)

Di solito il numero delle classi è dato a priori

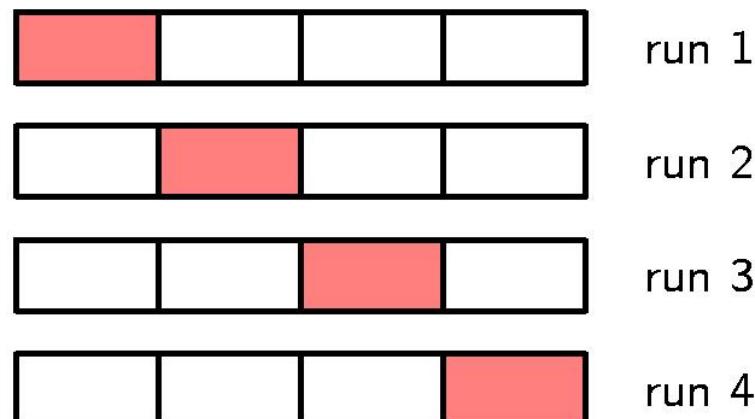
Addestramento con Rinforzo

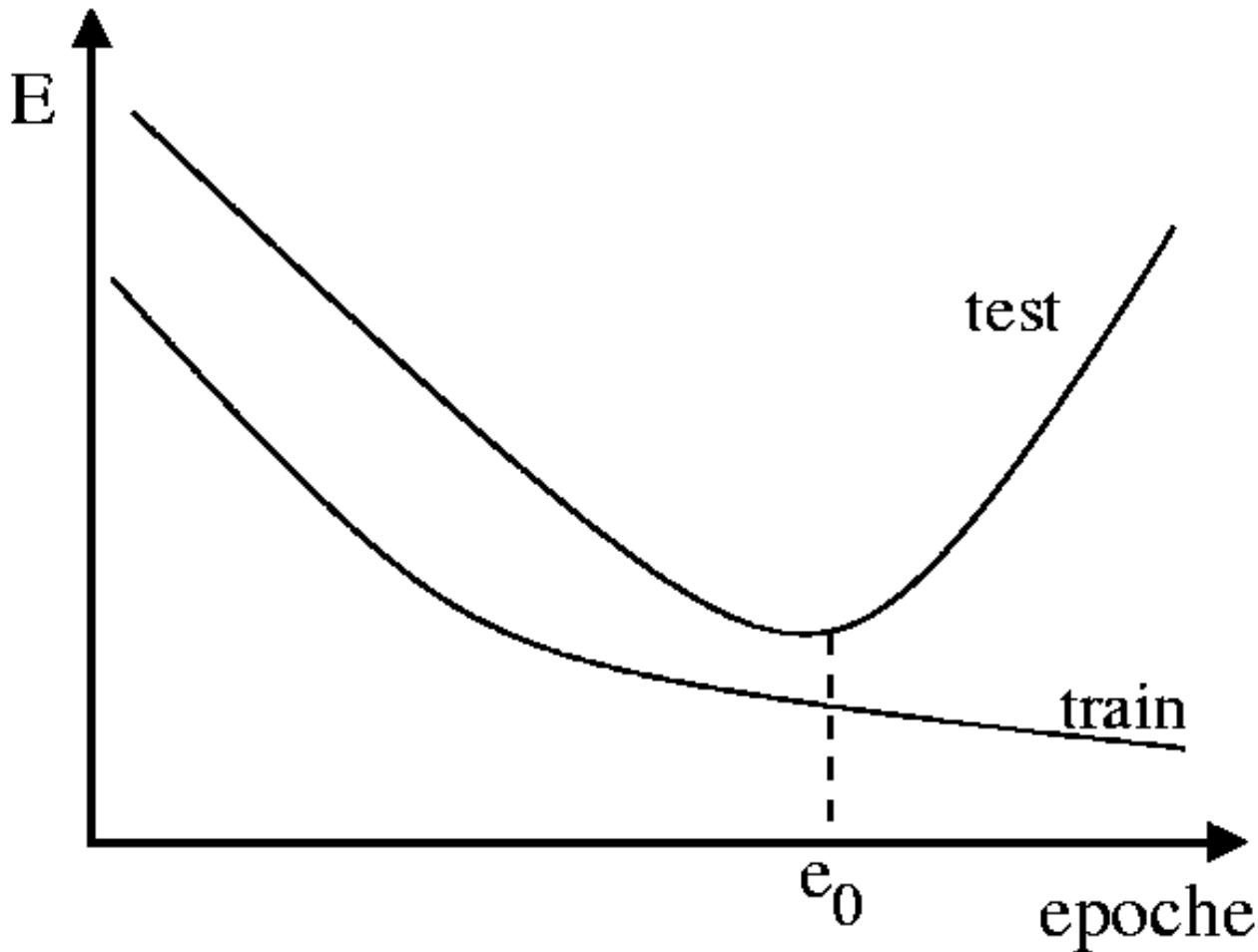
- Sinonimi: *reinforcement learning, learning with a critic*
- Idea:
 - o a metà strada tra le due: non viene fornita alcuna informazione sulla categoria esatta, viene dato un giudizio sulla correttezza della classificazione
- La strategia di addestramento viene modificata:
 - o si presenta un pattern al classificatore
 - o il classificatore fa un tentativo di classificazione
 - o viene detto se il tentativo è corretto o meno
 - o sulla base del giudizio si modifica il classificatore

Valutazione e model selection

- Misura delle prestazioni del classificatore
- Prestazioni di *generalizzazione*: capacitá del classificatore di classificare correttamente anche esempi non presenti nel data set
- Nessun errore sul training set non implica necessariamente aver ottenuto il classificatore ottimale (pb di *overfitting*, *overtraining*)
- Per evitare situazioni di *overfitting* è sempre meglio utilizzare due insiemi disgiunti in fase di learning, 1 per il training e 1 per il testing.

- Tecniche per la scelta del training set e del testing set:
 - Holdout: si suddivide casualmente il training set in due parti uguali: una per il training una per il testing, e.g., 50/50% oppure 80/20%
 - Averaged Holdout: si effettuano più partizioni holdout, e si media il risultato ottenuto. In questo modo si ha indipendenza dalla particolare partizione scelta
 - Leave-One-Out: per il training vengono utilizzati tutti i patterns tranne uno, utilizzato per il testing. Si ripete per tutte le possibili combinazioni e si media.
 - Leave-K-Out (or cross-folding or cross-validation): come il precedente, utilizza K elementi per il testing, invece che uno.





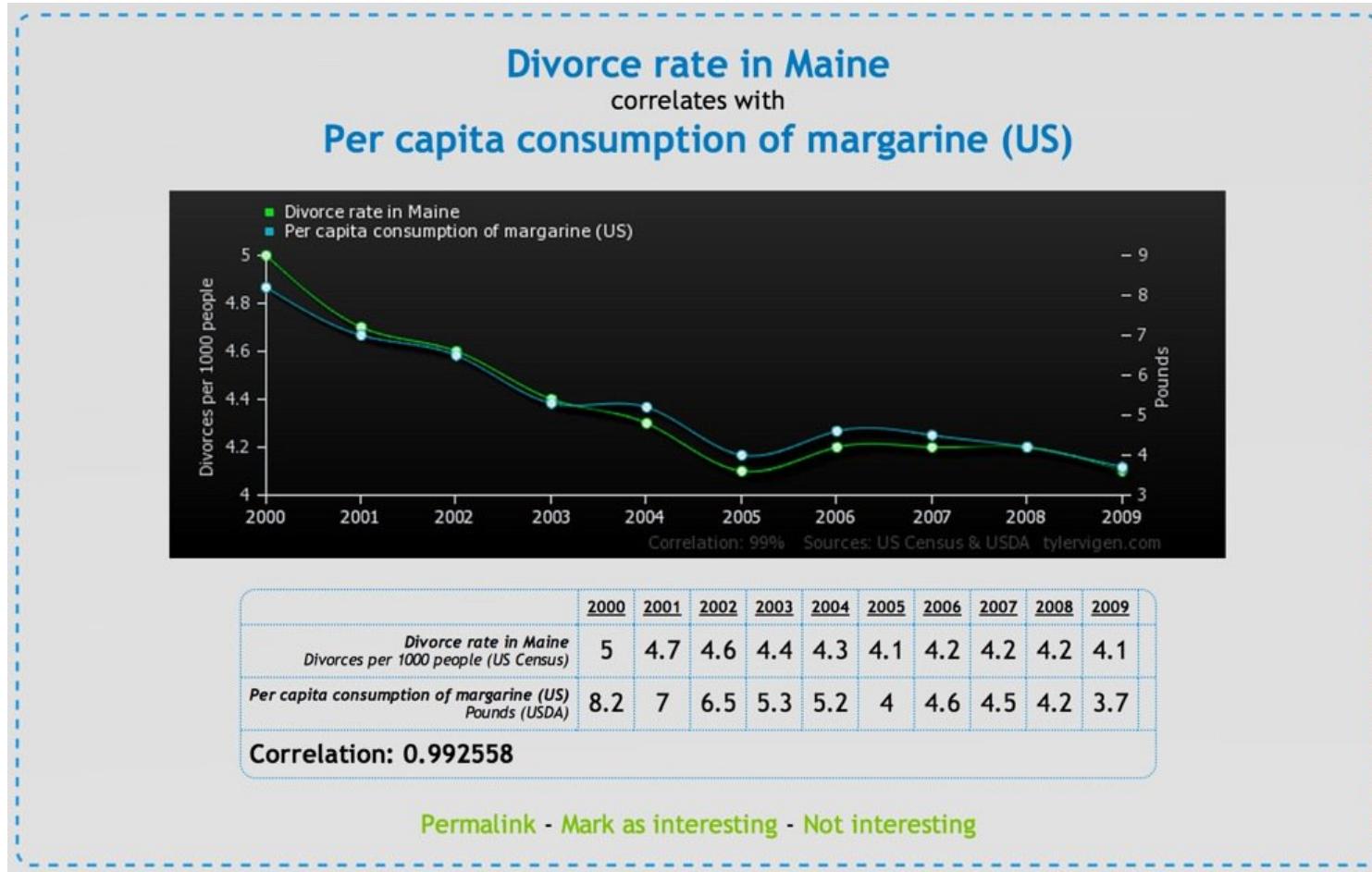
Si ferma l'addestramento prima del verificarsi del
fenomeno dell'overtraining (e_0)

Learning from *data*

Inter-class similarity

Intra-class variability

Learning from data: careful!!!



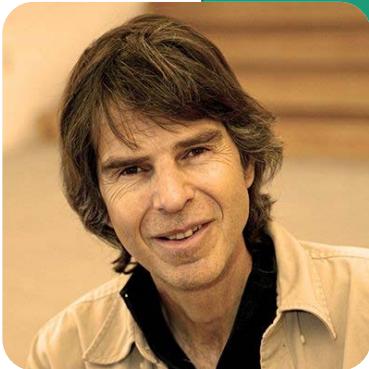
<http://tylervigen.com/spurious-correlations>

“No free lunch” theorem

- **Lack of inherent superiority of any classifier**
 - o If we are interested solely in the generalization performance, are there any reasons to prefer one classifier or learning algorithm over another?
 - o 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?
 - o 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

quanti dati sono classificati correttamente rispetto a tutti i dati

CONFUSION MATRIX

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

quanto è preciso un dato classificato come vero

classify as positive something that is negative

ACCURACY

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

PRECISION

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

RECALL

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

F-MEASURE

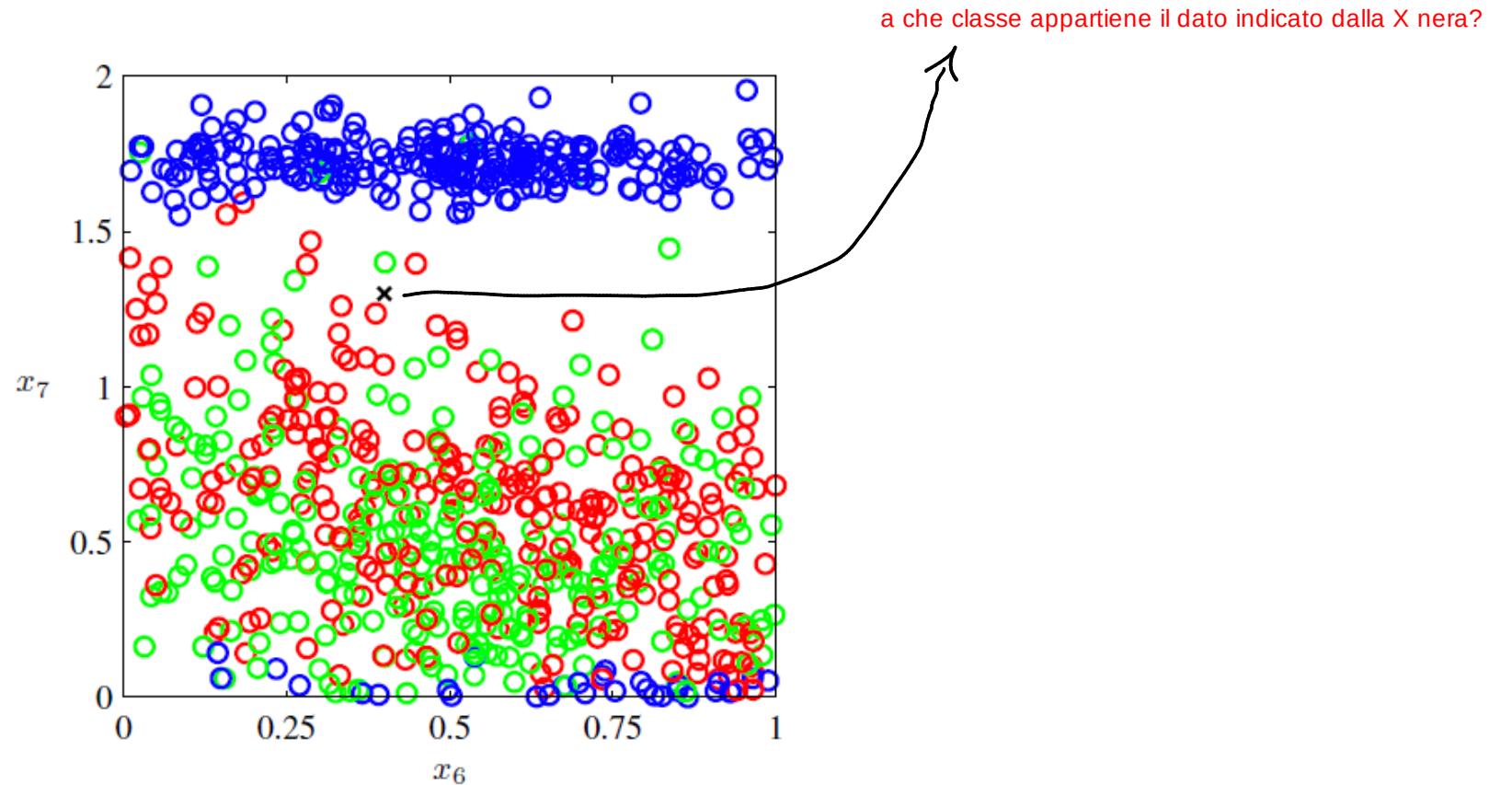
$$F_1 = 2 \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

Curse of dimensionality

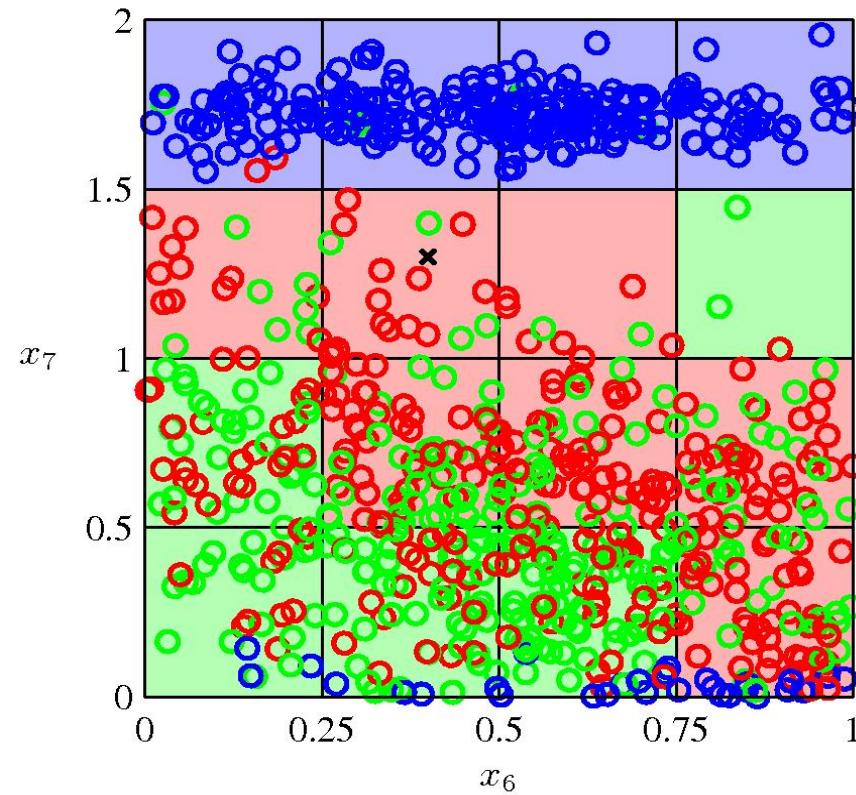
- Le prestazioni dipendono dalle relazioni tra il numero di campioni, numero di feature e dalla complessità del classificatore.
- In teoria, la probabilità di errore non aumenta se si aggiungono feature
- È dimostrato che la P(err) tende a 0 se il numero di feature tende a infinito per un problema a 2 classi (e sotto le ipotesi di pdf normali multivariate)
- In pratica si riscontrano dei problemi dovuti al fatto che le ipotesi sono solo approssimazioni nei casi reali
- Inoltre, il numero di campioni di training deve essere in relazione esponenziale rispetto al numero di feature
- Tutti i comuni classificatori soffrono di questo problema ed esistono regole guida

only in
theory

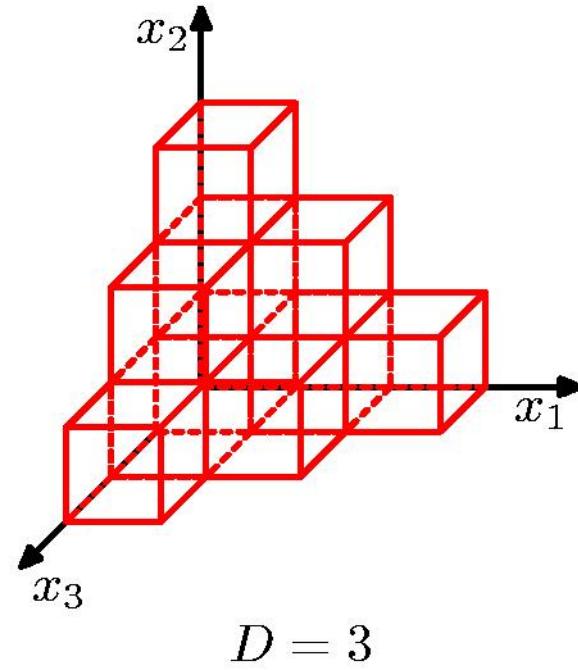
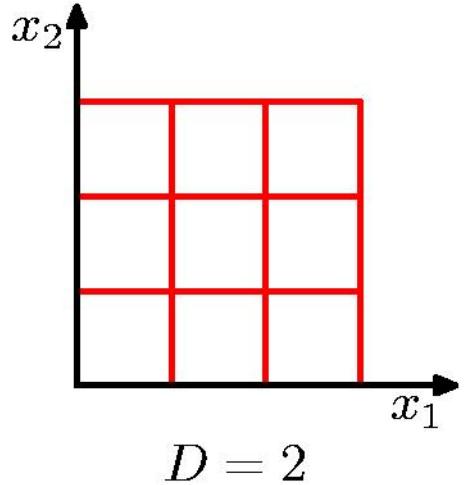
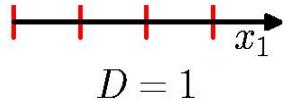
Curse of dimensionality



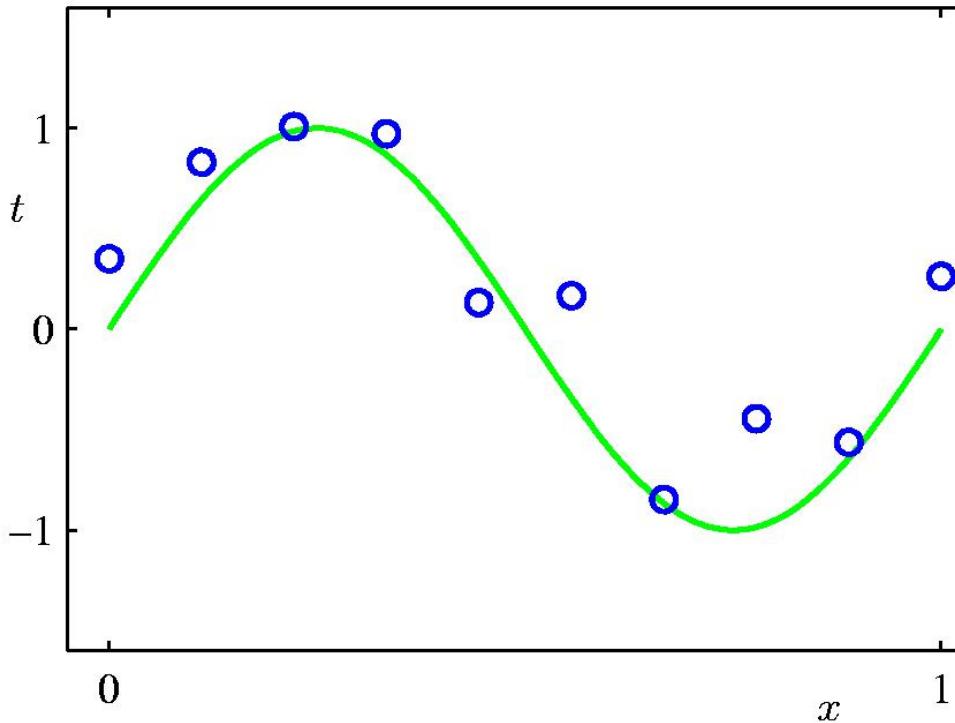
Curse of dimensionality



Curse of dimensionality

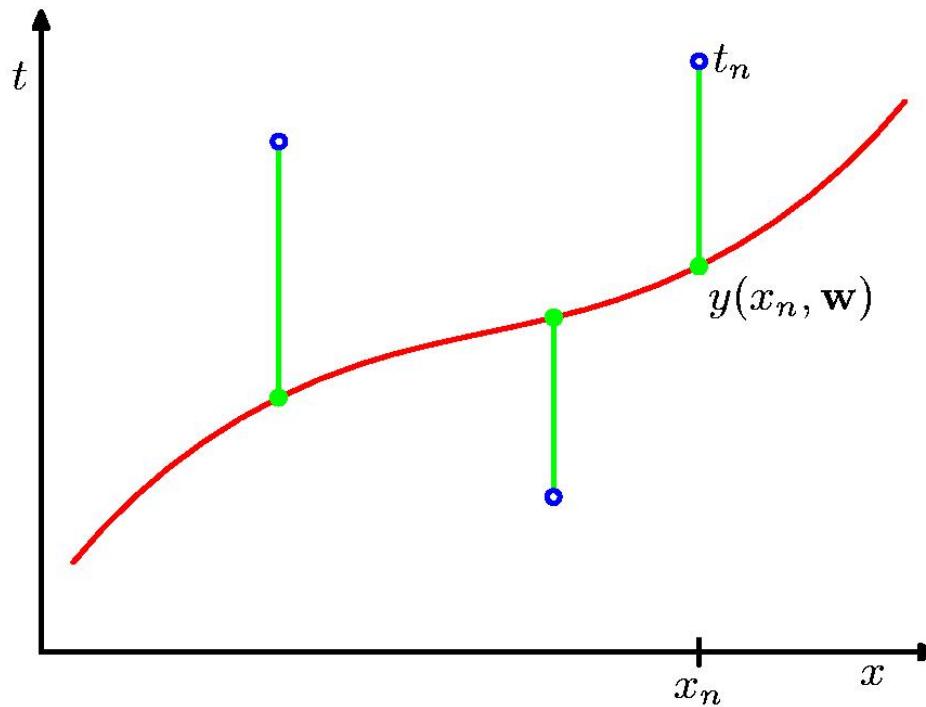


Example Polynomial Curve Fitting



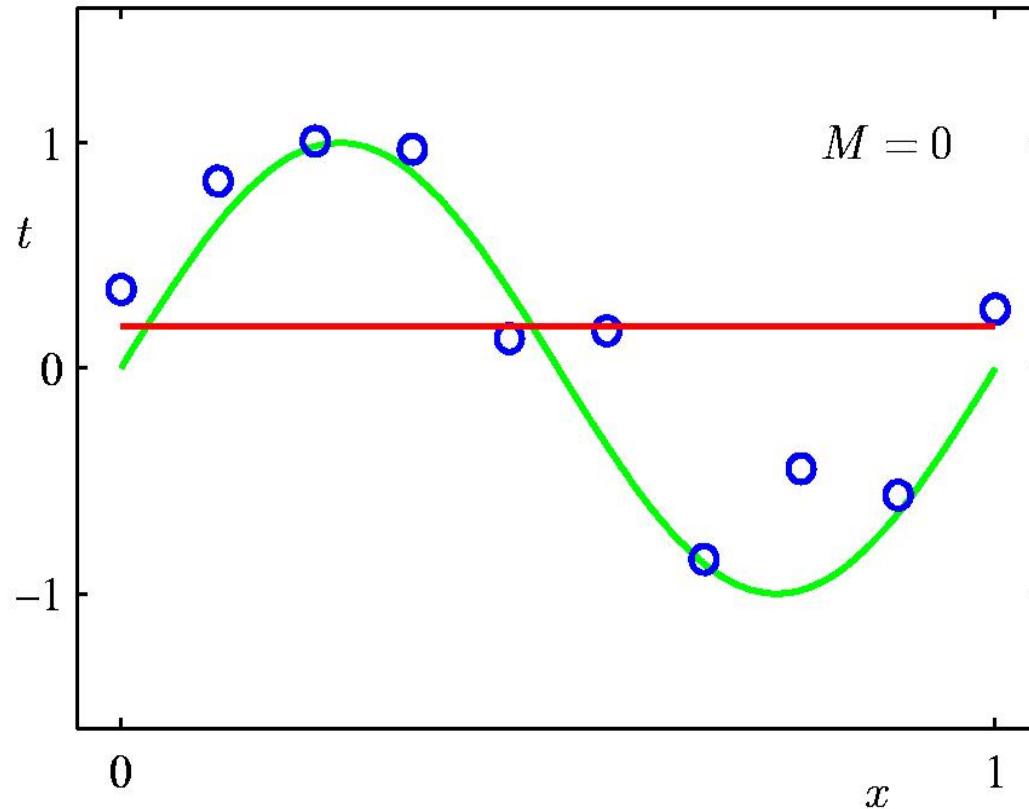
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \dots + w_M x^M = \sum_{j=0}^M w_j x^j$$

Sum-of-Squares Error Function

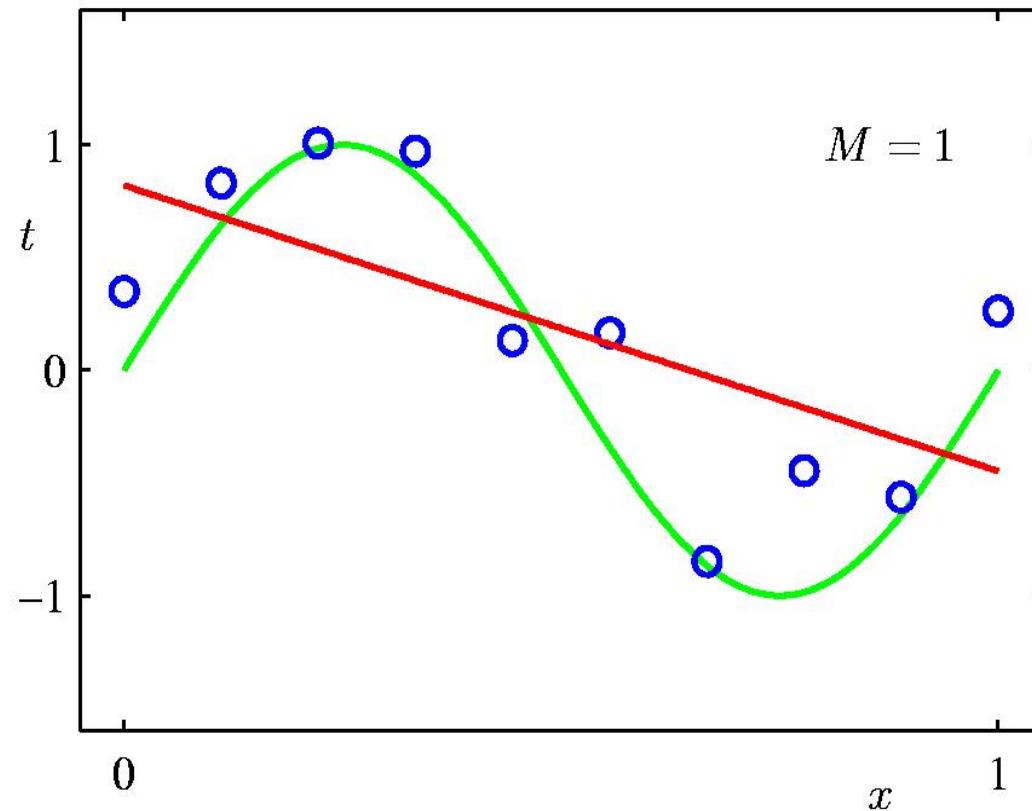


$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2$$

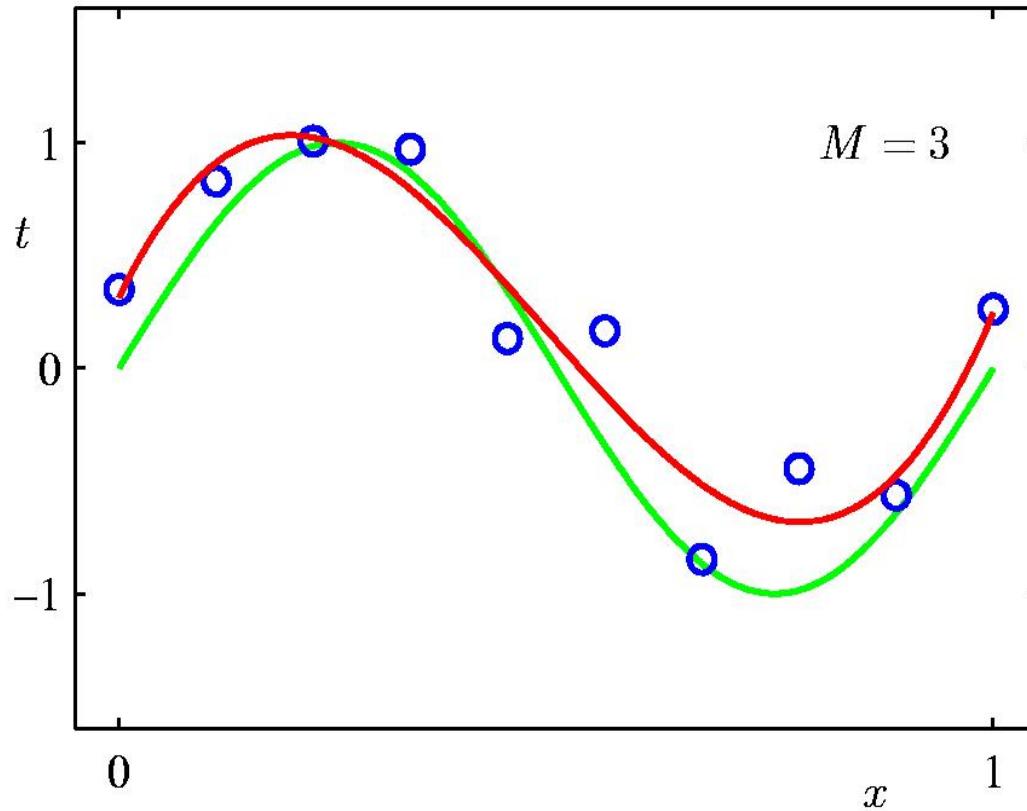
0th Order Polynomial



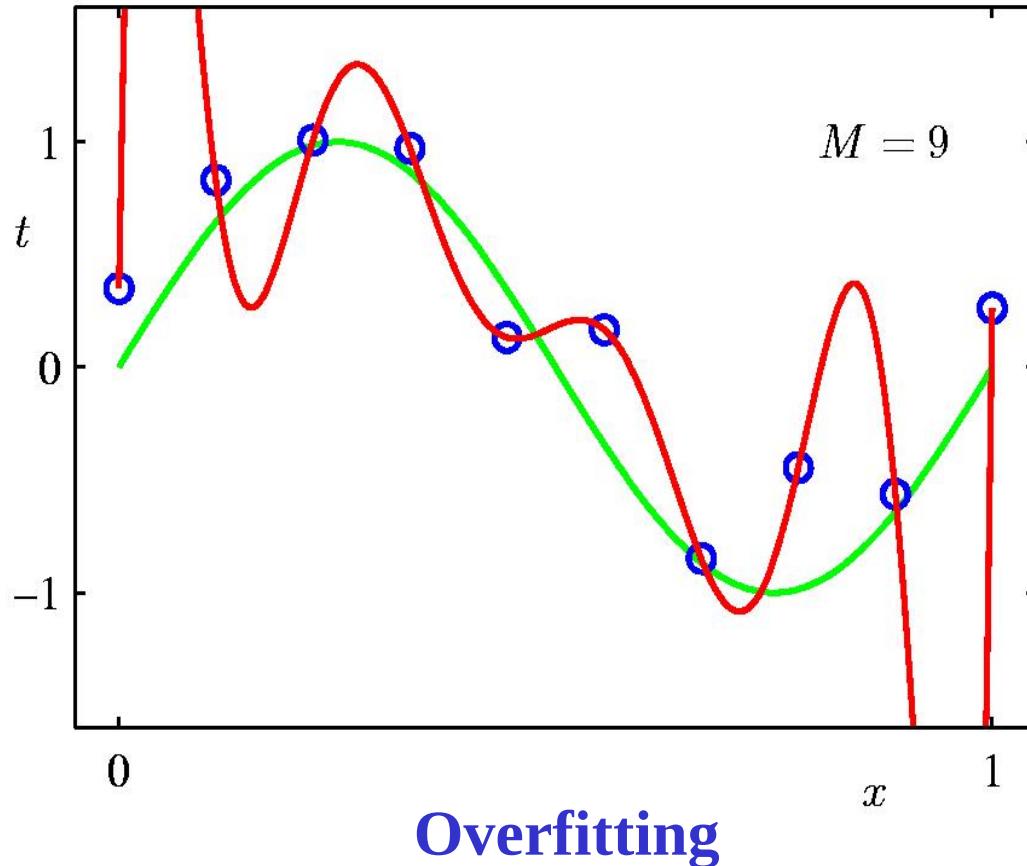
1st Order Polynomial



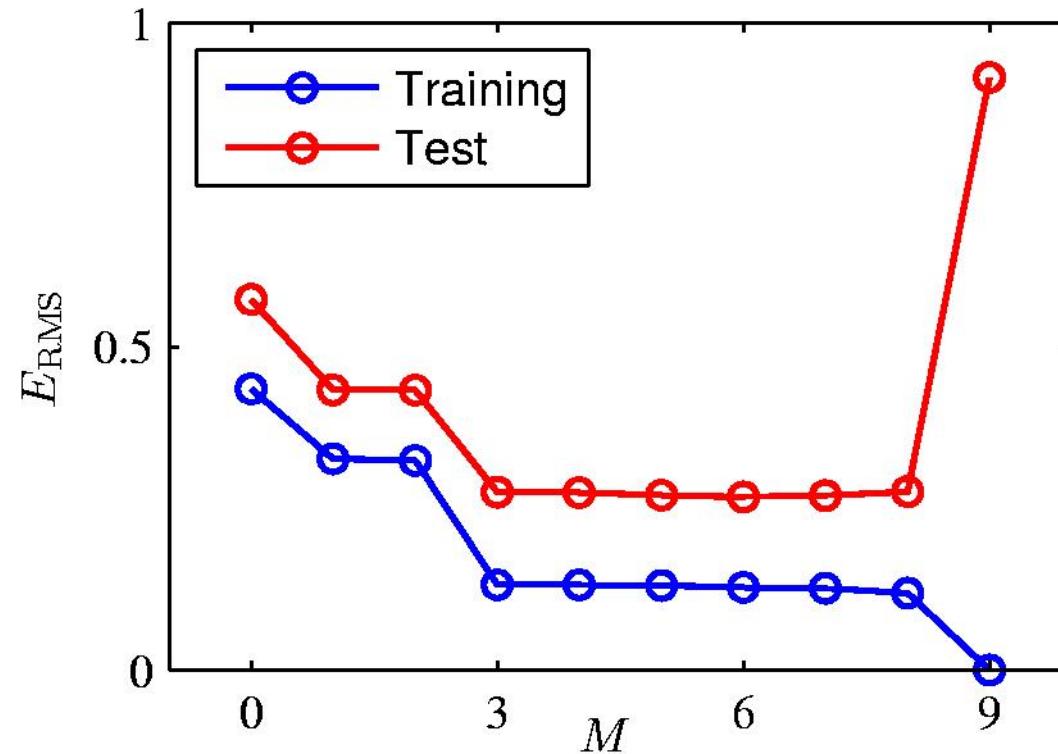
3rd Order Polynomial



9th Order Polynomial



Overfitting



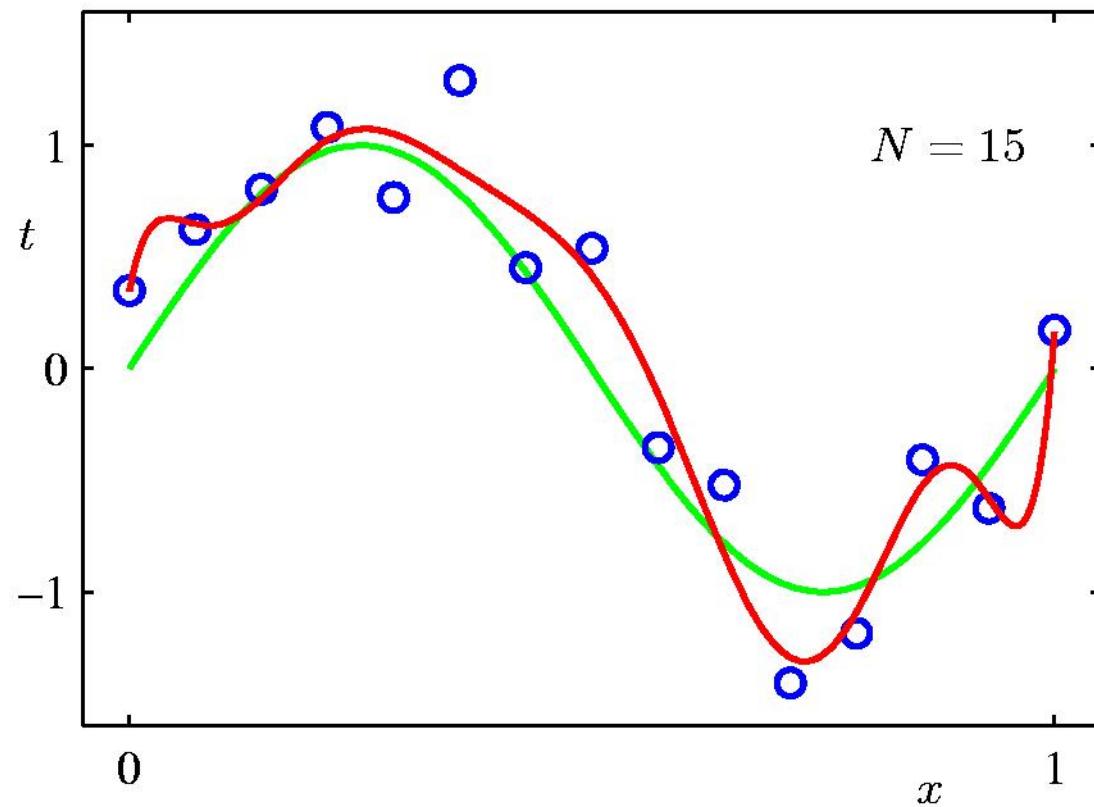
Root-Mean-Square (RMS) Error: $E_{\text{RMS}} = \sqrt{2E(\mathbf{w}^*)/N}$

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:

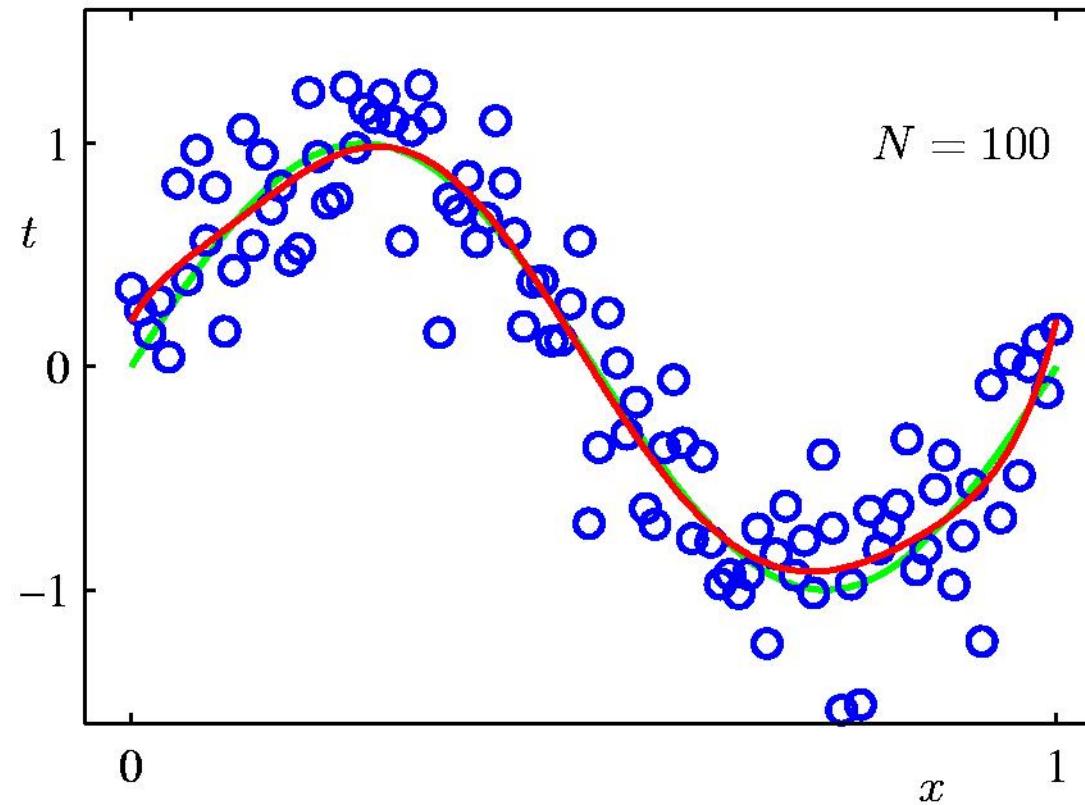
9th Order Polynomial $N = 15$



Data Set Size:

9th Order Polynomial

$N = 100$



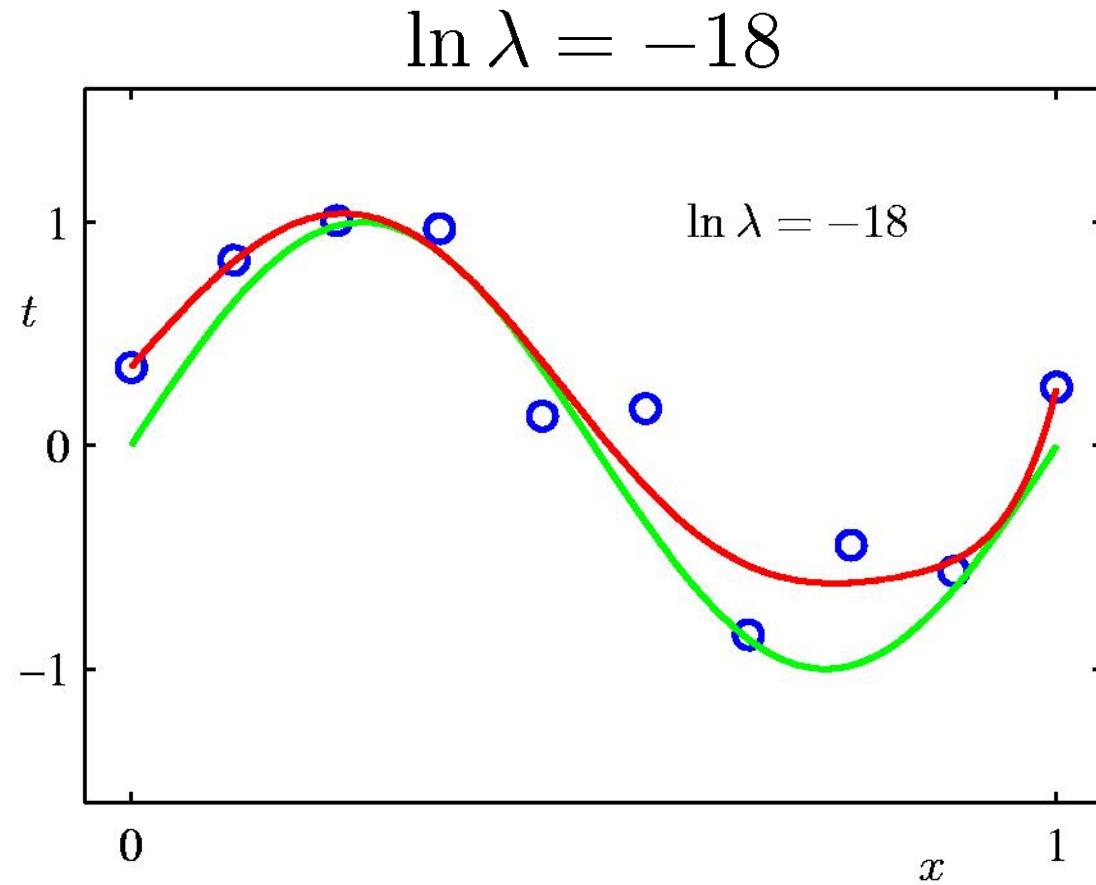
Regularization

- Penalize large coefficient values

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

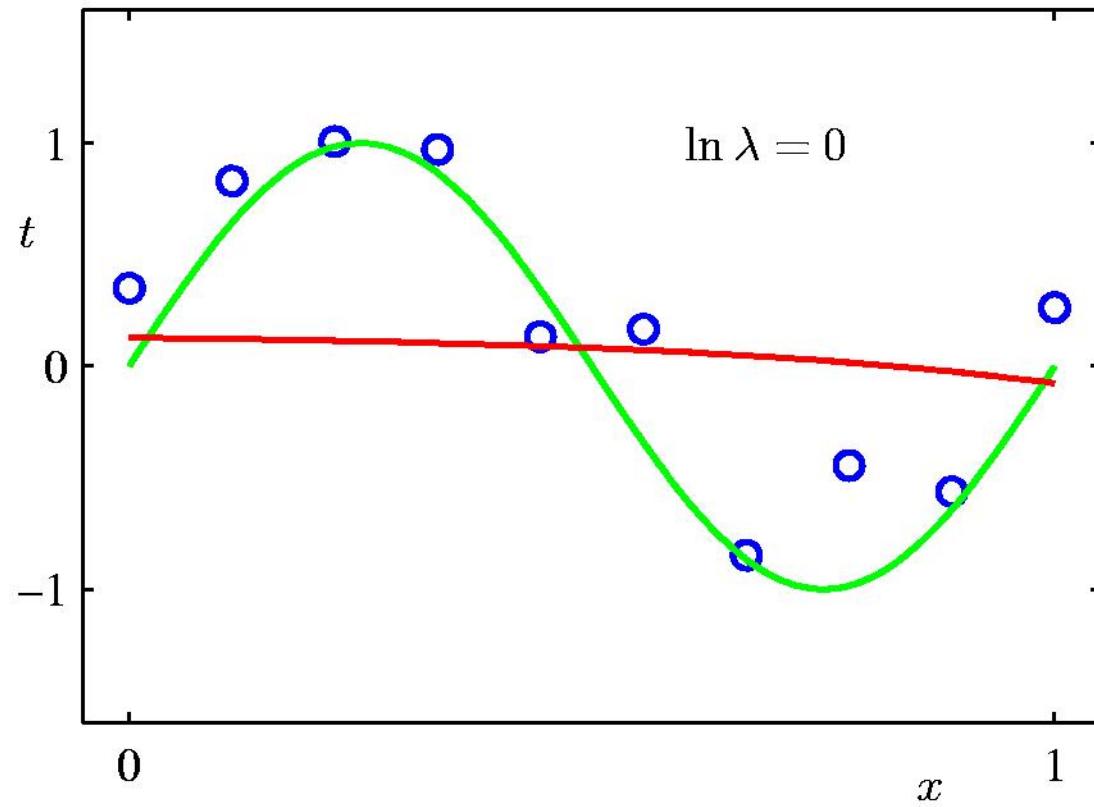
Regularization factor

Regularization:



Regularization:

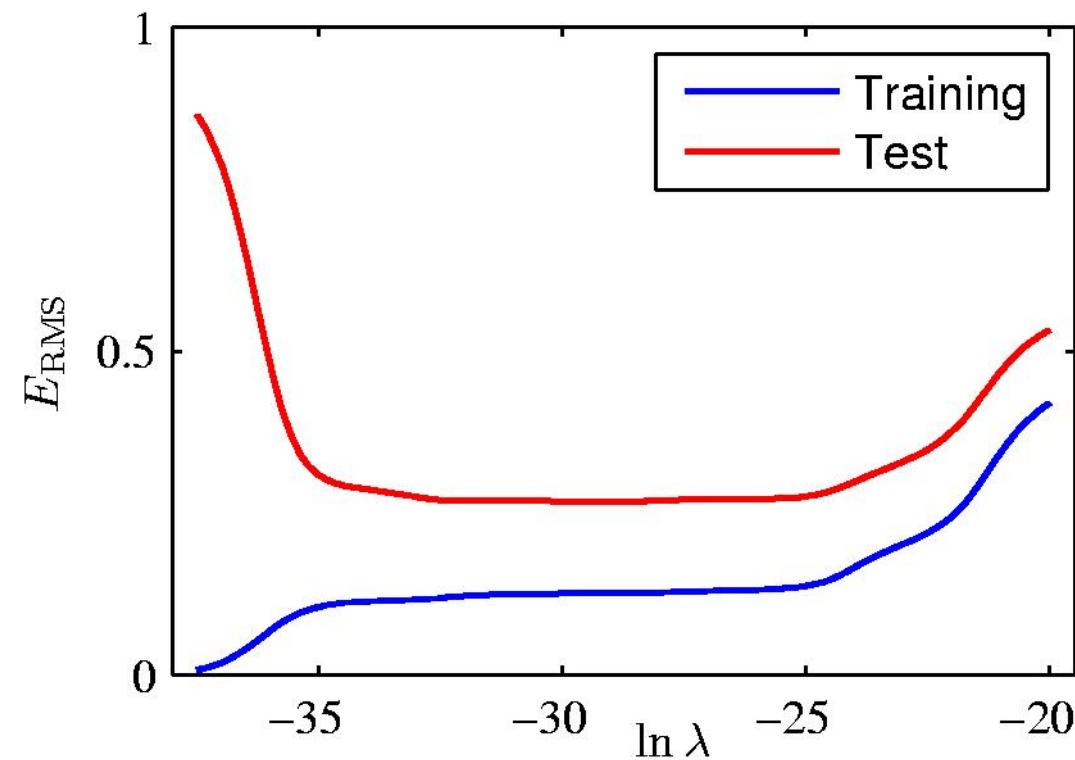
$$\ln \lambda = 0$$



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}



Estrazione e selezione delle feature

- Il numero di feature deve essere piccolo per limitare il costo della misura e non influire sull'accuratezza del classificatore
- Estrazione di feature: misura di caratteristiche dai dati o creazione di nuove feature da combinazioni di feature misurate
- Selezione di feature: migliore sottoinsieme delle feature estratte
- Tali feature possono aver una miglior capacità discriminativa, ma si perde il significato fisico di queste.
- Uso di una funzione criterio per la riduzione: tipicamente l'errore di classificazione di un sottoinsieme di feature.
- Inoltre, è importante determinare la dimensione dello spazio ridotto.

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-Loëve 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

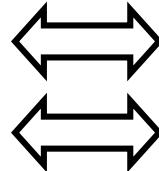
| Method | Property | Comments |
|--|---|--|
| Exhaustive Search | Evaluate all $\binom{d}{m}$ possible subsets. | Guaranteed to find the optimal subset; not feasible for even moderately large values of m and d . |
| Branch-and-Bound Search | Uses the well-known branch-and-bound search method; only a 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. |

Approcci alla Pattern Recognition

■ Approccio sintattico:

- o approccio gerarchico. Analogia tra la struttura dei patterns e la sintassi di un linguaggio:

- o patterns



- frasi di un linguaggio

- o sottopatterni primitivi

- alfabeto

■ Approccio statistico:

- o ad ogni pattern viene associato un vettore di feature che rappresenta un punto nello spazio multidimensionale del problema.
- o L'informazione sul problema, le dipendenze tra i vari fattori e i risultati prodotti sono tutti espressi in termini di probabilitá.

- Template matching
 - Reti neurali
 - Non sono tutti necessariamente indipendenti
- not statistical nor syntactic*

| Approach | Representation | Recognition Function | Typical Criterion |
|-------------------------|--------------------------------|-------------------------------|----------------------|
| Template matching | <u>Samples, pixels, curves</u> | 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 |

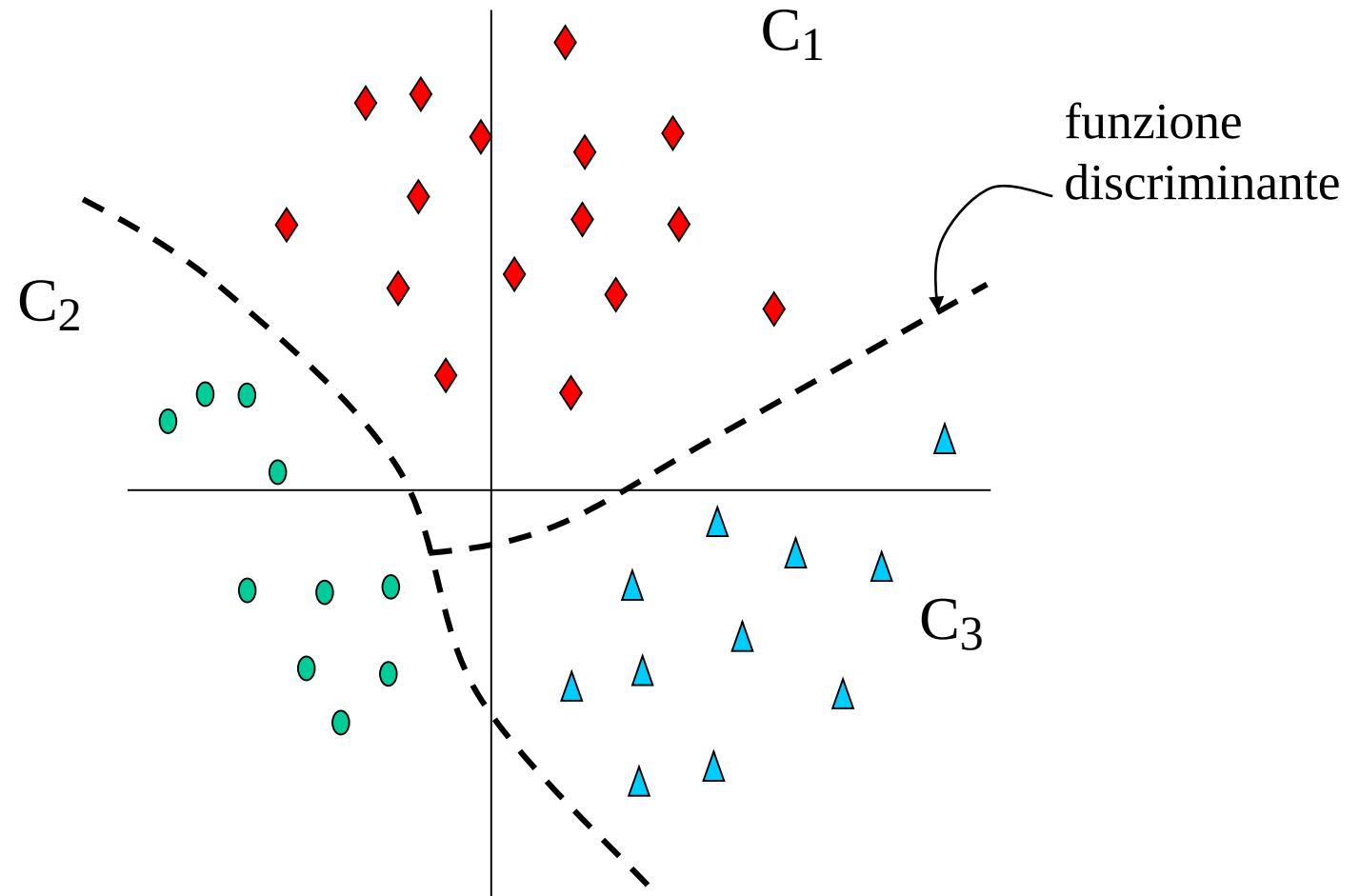
Template matching

simplest approach
brute-force approach

- Confronto di un modello (template, tipicamente una forma 2D) con i dati a disposizione per tutte le possibili istanze (diverse pose, scala).
- Misura di distanza (correlazione).
- Approccio forza bruta, computazionalmente oneroso, anche se esistono ottimizzazioni.

Classificazione statistica

- La descrizione statistica di oggetti utilizza descrizioni numeriche elementari chiamate *feature*, che formano i cosiddetti *pattern*, $\mathbf{x} = (x_1, x_2, \dots, x_n)$ o vettori di *feature*.
- L'insieme di tutti i possibili pattern forma lo spazio dei pattern o delle *feature*.
- Se esiste una (iper)superficie di separazione tra le classi il problema si dice con *classi separabili*.
- Se le iper-superfici sono iper-piani allora il problema si dice *linearmente separabile*.



Classificatore di Bayes \rightarrow classificatore ideale

- La classificazione statistica assume nota la PDF delle feature x in ogni classe $P(C_i|x)$ (nota dal problema o stimabile dato un training set)
 - ad esempio Gaussiana con media e varianza note o stimabili.
- Alternativamente si assumono note o stimabili la probabilità a priori delle classi $P(C_i)$ e la probabilità condizionata $p(x|C_i)$. $\xrightarrow{\text{a posteriori}}$
- In generale si deve minimizzare il rischio di Bayes (valore aspettato della funzione di perdita):

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

↑
Loss function

Probability f

- Se la funzione (matrice) di perdita L è binaria diagonale (1 se $i=j$, 0 altrimenti), allora si semplifica il tutto usando la teoria di decisione di Bayes.
- Queste probabilità sono legate dal **teorema di Bayes**:

$$P(C_i | x) = \frac{p(x | C_i)P(C_i)}{p(x)}$$

dove $p(x) = \sum_{i=1}^C p(x | C_i)P(C_i)$

- Il classificatore di Bayes classifica un nuovo oggetto x come appartenente alla classe C_k tale che

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

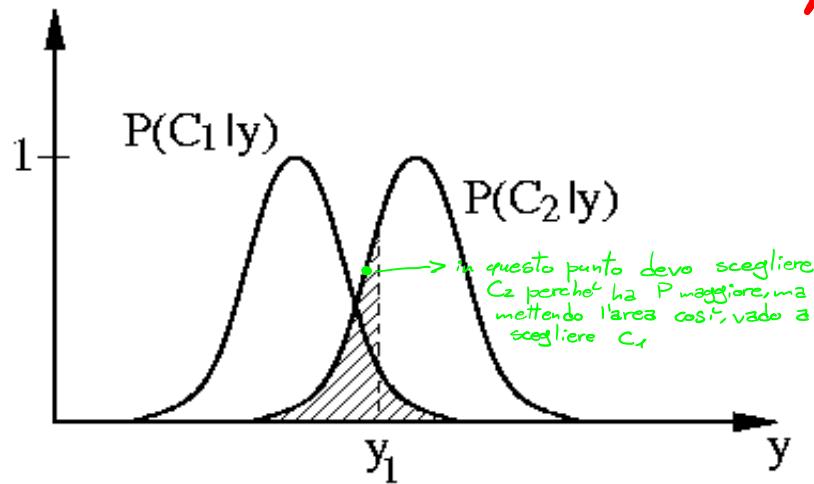
oppure

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

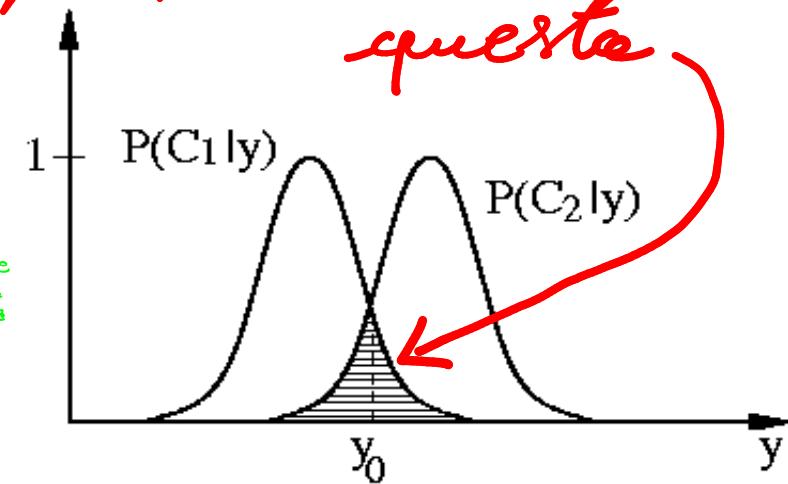
- Il classificatore di Bayes rappresenta l'ottimo teorico, in quanto minimizza la probabilità di commettere un errore.
- Problema: il problema è che le densità di probabilità non sono quasi mai note a priori, occorre stimarle dai dati a disposizione.
- Le prestazioni di questo classificatore dipendono dalla bontà di queste stime.

- Esempio: si vuole determinare la funzione discriminante per un problema a due classi, la cui distribuzione di probabilità $P(C_i|y)$ è rappresentata in figura
 - y_0 è la soglia scelta dal classificatore di Bayes e minimizza la probabilità di errore (area tratteggiata); per qualsiasi altra scelta (y_1), la probabilità di errore è maggiore.

L'area più piccola possibile è questa



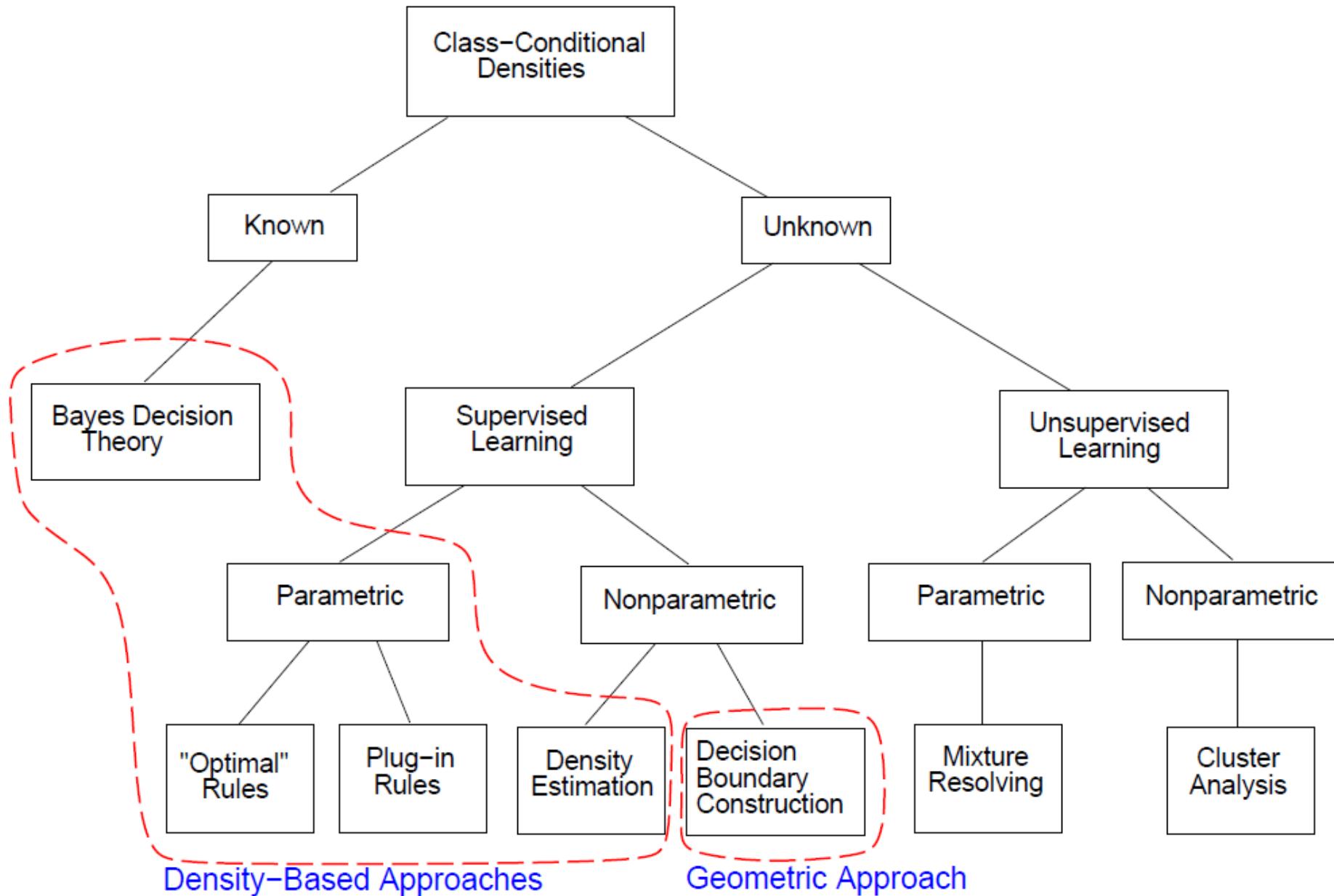
(a)



(b)

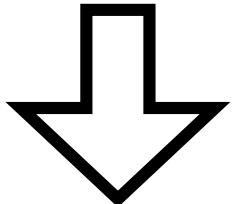
Stima delle pdf

- *Classificatori parametrici*
 - si fissa il modello della distribuzione e sulla base del training set se ne stimano i parametri;
 - esempio: classificatore gaussiano.
- *Classificatori non parametrici*
 - nessuna assunzione sulla forma della pdf, la stima si basa esclusivamente sui dati;
 - esempio: K-nearest Neighbor.
- *Classificatori semi-parametrici*
 - si ha una classe molto generale di modelli di pdf, in cui il numero di parametri può essere aumentato in modo sistematico per costruire modelli sempre più flessibili;
 - esempio: reti neurali.



K Nearest Neighbor (KNN)

- Classificatore **non parametrico**.
- Molto utilizzato per la sua semplicità, flessibilità e ragionevole accuratezza dei risultati prodotti.
- IDEA: due elementi della stessa classe avranno, molto probabilmente, caratteristiche simili, cioè saranno vicini nello spazio dei punti che rappresenta il problema



La classe di un punto può essere determinata analizzando la classe dei punti in un suo intorno

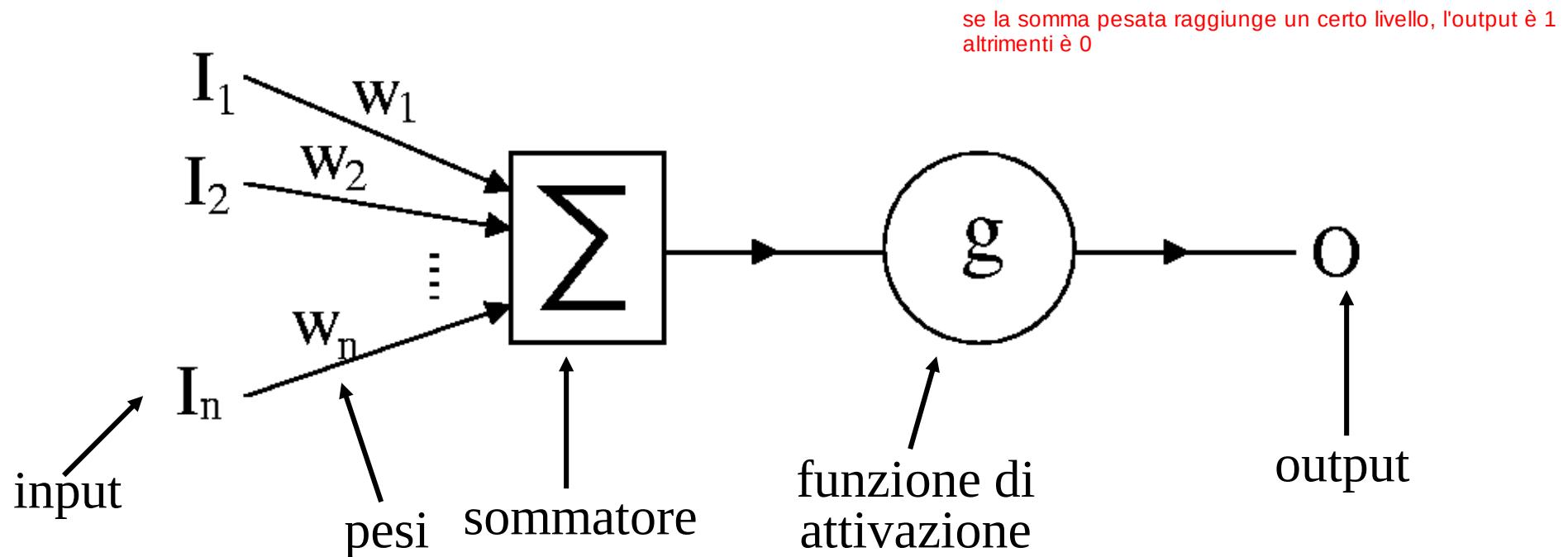
Algoritmo

- Dato un insieme di esempi X , dato un punto da classificare x_0 :
 - si calcola l'insieme U dei K punti di X più vicini a x_0 secondo una determinata metrica Σ (di solito la distanza euclidea);
 - si calcola la classe C più frequente all'interno dell'insieme U ;
 - x_0 verrà classificato come appartenente a C .
- Problema: scelta del parametro K e della metrica Σ .

Reti neurali: motivazioni

- Sistema artificiale di elaborazione dell'informazione che emula il sistema nervoso animale.
- Caratteristiche del sistema nervoso animale:
 - o robusto e resistente ai guasti;
 - o flessibile, si adatta a situazioni nuove imparando;
 - o lavora anche con informazione approssimata, incompleta o affetta da errore;
 - o permette un **calcolo altamente parallelo;**
 - o **piccolo e compatto.**

- Reti neurali: struttura complessa, composta da tante unità elementari di calcolo collegate tra loro in vario modo.
- Le unità elementari sono dette *neuroni*.
- I collegamenti sono detti *sinapsi*.



- L'output viene calcolato con

$$O = g\left(\sum_{i=1}^n w_i I_i - \theta\right)$$

Threshold

- Diverse possibilità per la funzione di attivazione

o Heaviside

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

o logistica

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

o tangente iperbolica

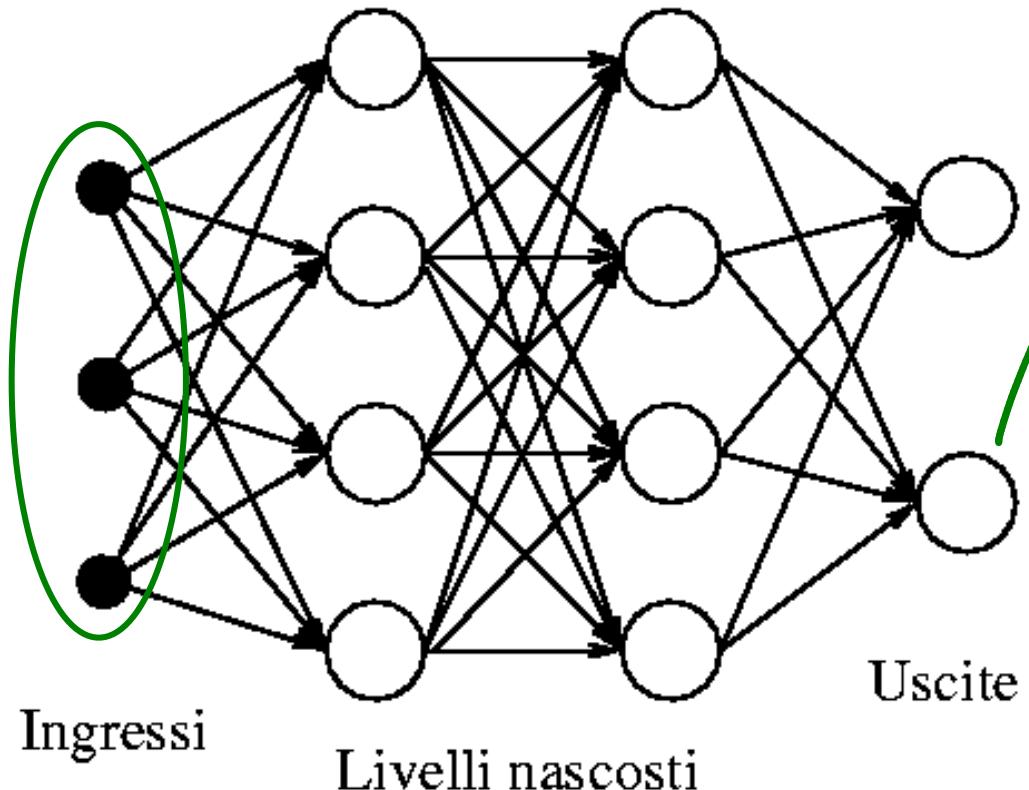
$$g(a) = \tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}}$$

• sigmoid function?

Feed-forward neural network Multi-layer perceptron

Diverse topologie

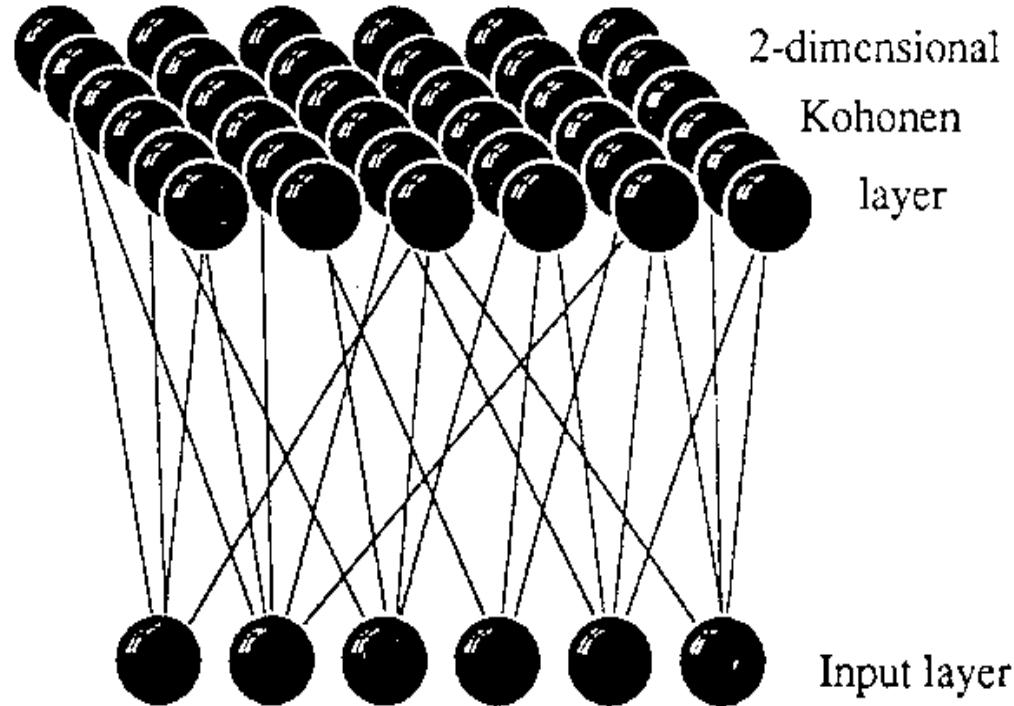
features
vector →



neurons corresponding
to the # of classes
*Feed forward
neural networks*

The main issue is defining the architecture, because we don't know what is in between the input and output (hidden levels)

Training phase: define the weights

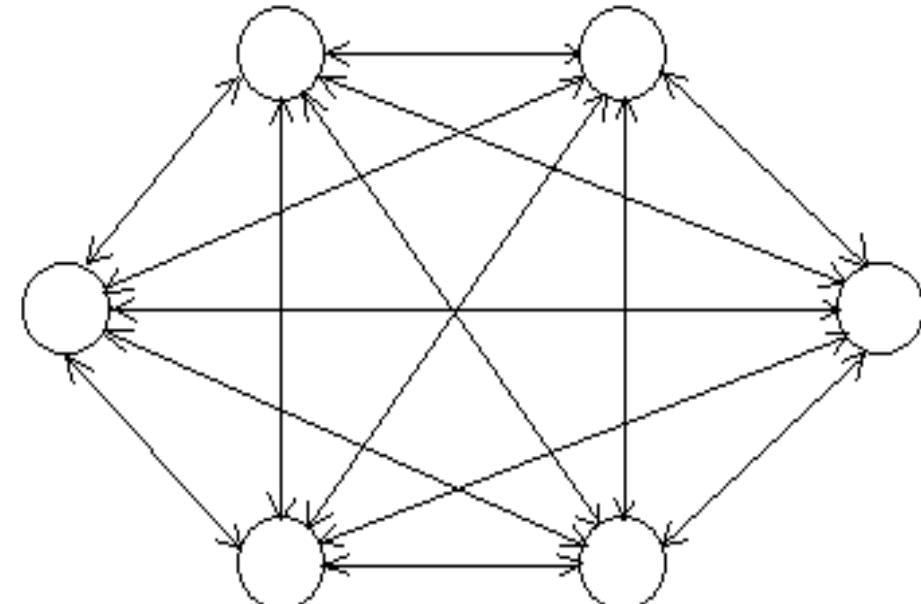


Only one input, feeding one layer

Self Organizing Maps:
utilizzate per fare clustering

Reti di Hopfield:

la rete evolve fino a convergere
in un determinato stato



Tassonomia metodi di classificazione

base element in neural nets

| 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. |
| <u>Perceptron</u> | 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

- Classificazione non supervisionata, non si conoscono le classi, non si conoscono dati di riferimento
- Non si conosce il numero delle classi
- Uno dei problemi è la definizione di un criterio di similarità che sia dipendente dai dati che dal contesto
- Due tecniche principali
 - o Agglomerativo gerarchico
 - o Iterativo partizionale

you can start each feat. vector as a cluster, then understand how to group them

you split the main cluster into sub-clusters in order to obtain the # of clusters I need

Tassonomia dei metodi di clustering

| 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). |

Combinazioni di classificatori

Ensemble methods

- Problema relativamente esplorato

- Assumono

- classificatori diversi e con prestazioni diverse e non ottime;
 - training set diversi;
 - diversi classificatori addestrati con uguali training set e quindi con prestazioni diverse;
 - uguali classificatori addestrati differentemente (NN) e che risultano avere differenti prestazioni.

- Obiettivo di aumentare le prestazioni

- Approcci:

- **parallelo**: si combinano i risultati dei singoli classificatori;
 - **seriale**: i risultati di uno sono input del successivo fino al risultato finale;
 - **gerarchico**: i classificatori sono strutturati ad albero e i risultati combinati adeguatamente.

Combining many classifiers improves a lot the accuracy

using different types

the lacks of one classifier are filled by another one

Serial



Per riassumere, sui metodi di Machine Learning ...

- Classificazione statistica
 - definizione del modello parametrico $M(w)$
- Questo implica
 - stima della distribuzione completa (congiunta parametri e dati), o della distribuzione a posteriori, ovvero
 - stima ottima dei parametri che massimizza la probabilità a posteriori, ovvero
 - stima delle probabilità marginali (rispetto ad alcuni parametri) o delle aspettative (expectation, predizioni) relativamente alla probabilità a posteriori
- Tutto può essere visto come un problema di ottimizzazione

- **Programmazione dinamica:** cerca il percorso più corto in un grafo appropriato con una metrica definita
 - algoritmo di Smith-Waterman (BLAST)
 - algoritmo di Needleman-Wunch
 - algoritmo di Viterbi
- **Discesa del gradiente:** ricerca del minimo di una funzione
 - stima dei parametri

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

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

- numerose varianti

- *Expectation-Maximization (EM) e Generalised EM*
 - Usato quando il modello coinvolge variabili latenti (nascoste), tipo Hidden Markov Model (HMM)
 - Passo E e passo M alternati
 - Il passo E stima la distribuzione delle variabili nascoste (date le osservazioni)
 - Il passo M, si aggiornano i parametri, data la distribuzione stimata prima
- *Markov-Chain Monte-Carlo, MCMC*
 - Deriva dalla fisica statistica
 - Stima del valore aspettato (expectation) di una distribuzione di probabilità multidimensionale $P(x_1, \dots, x_n)$, dove x_i possono essere parametri, parametri nascosti o dati osservati
 - Campionamento di tale distribuzione costruendo una catena di Markov che ha P come distribuzione nello stato di equilibrio
 - Possibili algoritmi: Gibbs sampling, algoritmo di Metropolis

- *Simulated annealing*
 - Derivato dalla meccanica statistica
 - Combina MCMC con un meccanismo di raffreddamento del “sistema” al fine di portarlo ad uno stato più stabile e robusto
- Metodi genetici ed evoluzionistici
 - Derivati dalle teorie dell’evoluzione
 - Si impongono al sistema alterazioni casuali (mutazioni) e si usa una funzione di fitness per valutare la qualità della mutazione ed eventualmente scartarla
 - Gli algorimi genetici, oltre alle mutazioni, permettono nuove generazioni di punti (crossover)

Inizio a stimare i parametri, in cui cerco di assumere che il prossimo stato possa essere uno qualsiasi. Durante il tempo, "raffreddo" il sistema in modo che i salti possano essere più piccoli, e man mano arrivo ad un punto in cui ho uno stato stabile .

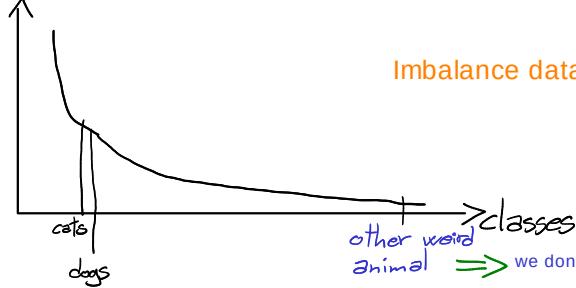
In sintesi ...

- ... nell'uso di questi algoritmi di apprendimento bisogna tener conto di
 - complessità del modello, *model selection*
 - fase di addestramento, *batch* o *online*
 - disponibilità e tipo di dati di training, test e validazione
 - meccanismi di stop della fase di learning (pb. di *overfitting*)
 - valutare l'uso di diversi modelli o dello stesso tipo di modello addestrato in modo diverso (insiemi di classificatori)
 - **bilanciamento dei dati a disposizione**

Finora abbiamo dato per scontato che i dati fossero bilanciati tra le varie classi, ma non è così in realtà. Averne troppi per una classe e pochi per l'altra non va per niente bene.
- Spesso, tutti questi modelli sono **adattati** allo specifico problema/applicazione da affrontare, sono spesso necessarie **metodi e modelli *ad hoc***

long tail distribution problem

#instances per class



Imbalance data learning

other weird
animal \Rightarrow we don't have enough instances for these classes

... sempre riguardo al bilanciamento dei dati a distribuzione

similar problem \rightarrow Few-shot learning \rightarrow few instances for each class

1-shot learning
 $\rightarrow \emptyset$ -shot learning

APPLICAZIONI

La rinascita della Pattern Recognition

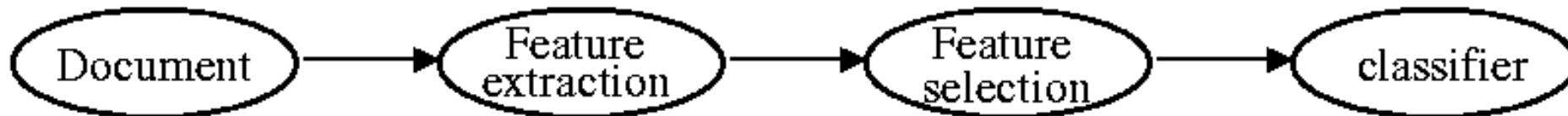
- Fattori che hanno decretato la rinascita della pattern recognition negli ultimi anni:
 - aumento della capacità computazionale dei calcolatori
 - presenza di grosse quantità di dati anche distribuite
 - nuovi sistemi di interazione uomo-macchina.

Applicazioni classiche

- Riconoscimento del parlato:
 - problematiche:
 - tono
 - voce
 - velocità
 - stato d'animo
 - applicazione: informazioni telefoniche senza l'assistenza di un operatore (chatbot);
- Riconoscimento di caratteri scritti a mano:
 - problematiche:
 - grafia
 - stato d'animo
 - applicazione: lettura automatica CAP nelle lettere

Classificazione di documenti

- Definizione:
 - classificazione di documenti sulla base dell'argomento contenuto (sport, economia, ...).
- Feature:
 - tipologie di parole, frequenze assolute e relativa
- Clustering
- Applicazioni: ricerche in internet, data mining.



Audio

- Speech vs. music vs. noise
- Speech recognition
- Speaker recognition
- 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 — Knowledge discovery

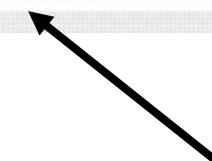
- Definizione:
 - estrazione di conoscenza da un insieme (tipicamente molto vasto) di dati multidimensionali.
- Scopi: predizione, classificazione, clustering, analisi delle associazioni, etc..
- Si noti che di solito i dati utilizzati per il Data Mining sono stati raccolti con un altro fine, diverso dal Data Mining.
- *Esempio:* dato un insieme di consumatori, raggrupparli in base a comportamenti di acquisto simili.

Image retrieval by content

- Definizione:
 - *image retrieval*: trovare, in un database, immagini o sequenze di immagini rispondenti ad una determinata query;
 - *by content*: la ricerca avviene sulla base del contenuto dell'immagine, non piú sulla base di un testo (annotato a mano su tutte le immagini del data set);
- Esempi di *query*:
 - “Trovami tutte le immagini simile ad un’immagine data”.
 - “Trovami tutte le immagini che contengono un cavallo”.

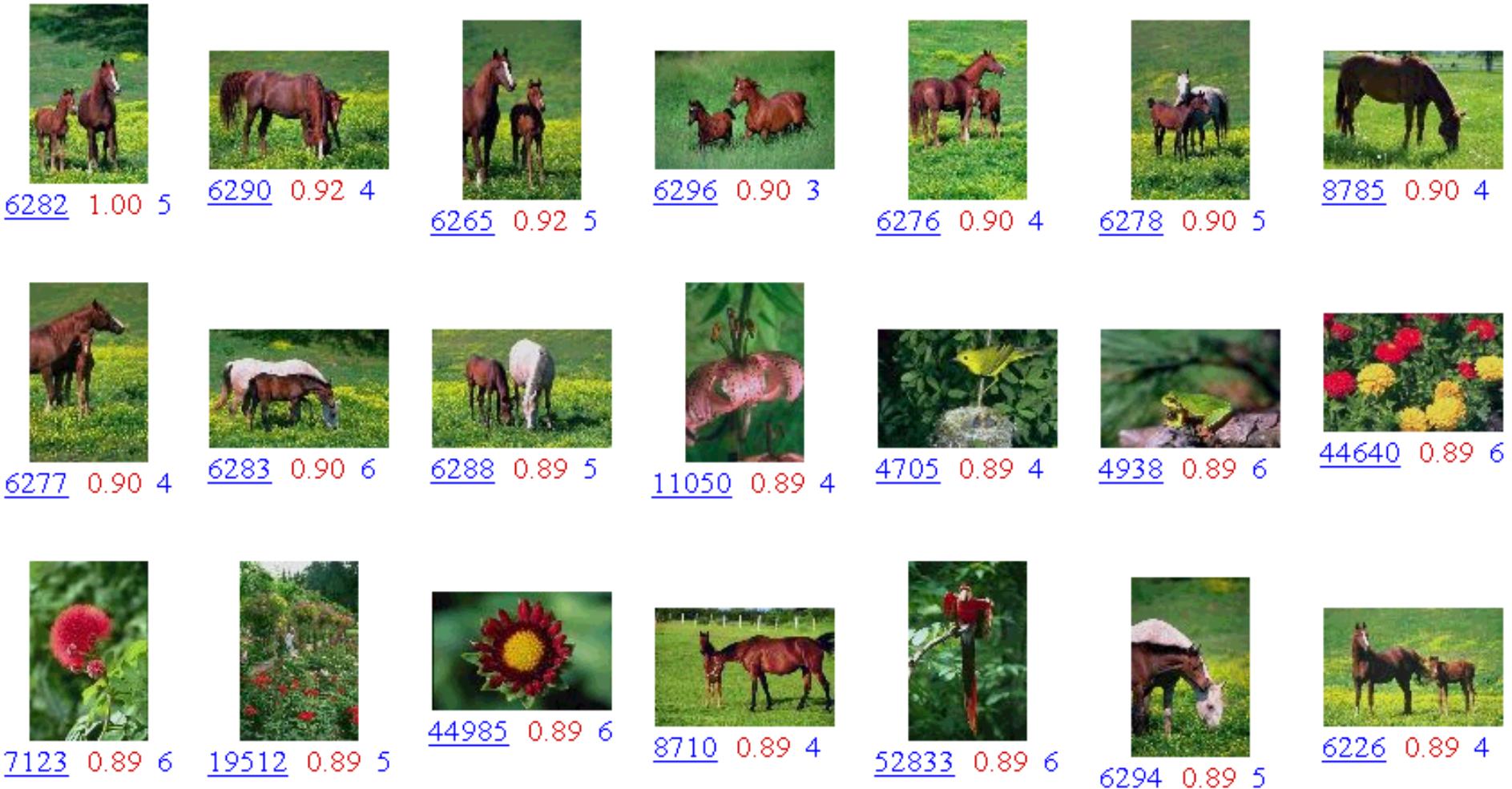
Esempio

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



Ricerca di tutte le foto simili a questa

Risultato: c'è anche un punteggio sull'affidabilitá del retrieval



Riconoscimento di gesti

- Sistema che identifica gesti umani e li utilizza per portare informazione, oppure per il controllo di dispositivi.
- Strumenti:
 - basato su guanti che tengono traccia della traiettoria;
 - basato su sistemi di computer vision che recuperano la traiettoria da informazioni stereo (con o senza marker).

Tracking del corpo

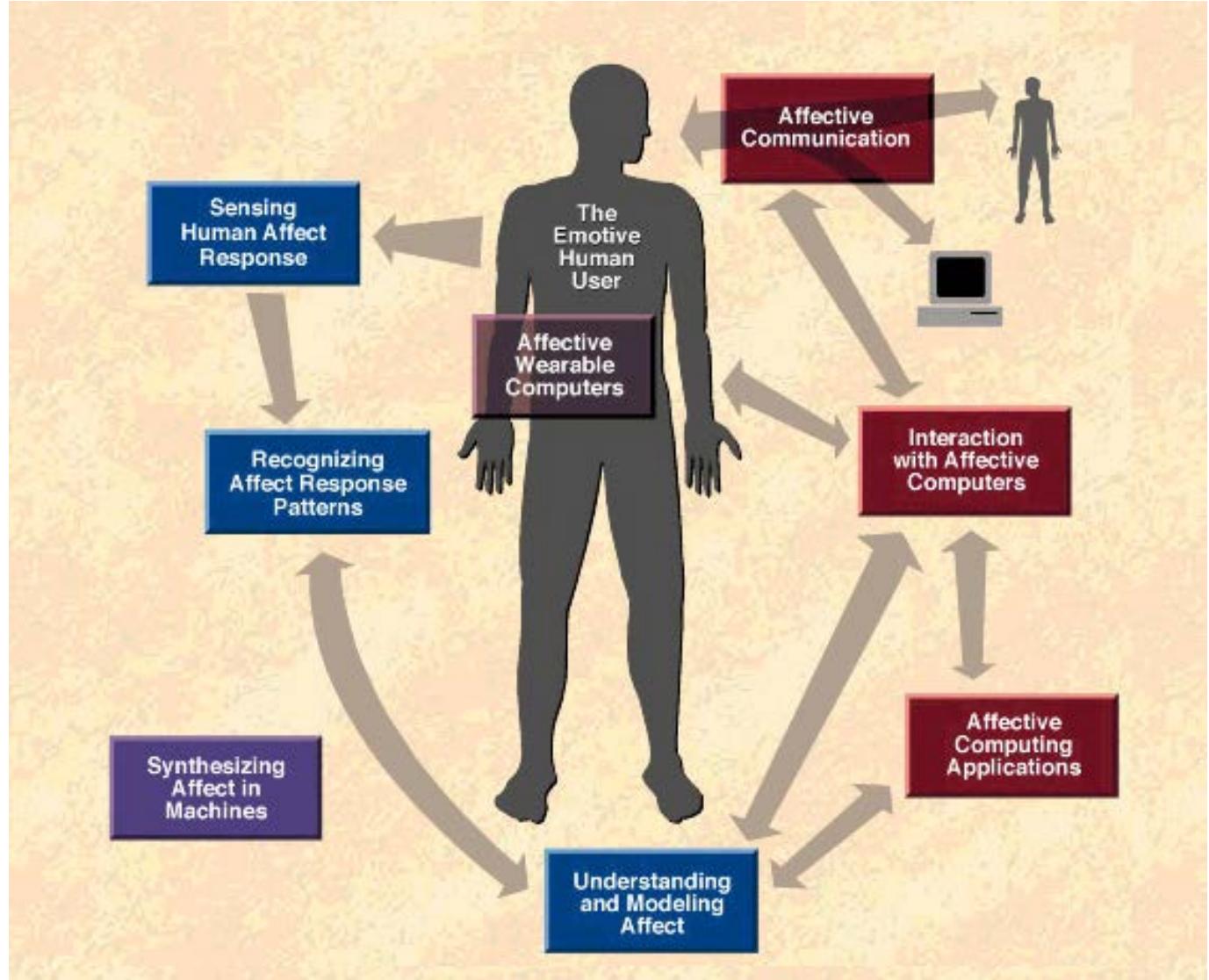


Recent evolution ...



Affective Computing

<https://youtu.be/sRh8AUakO90>



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

■ Sentic modulation

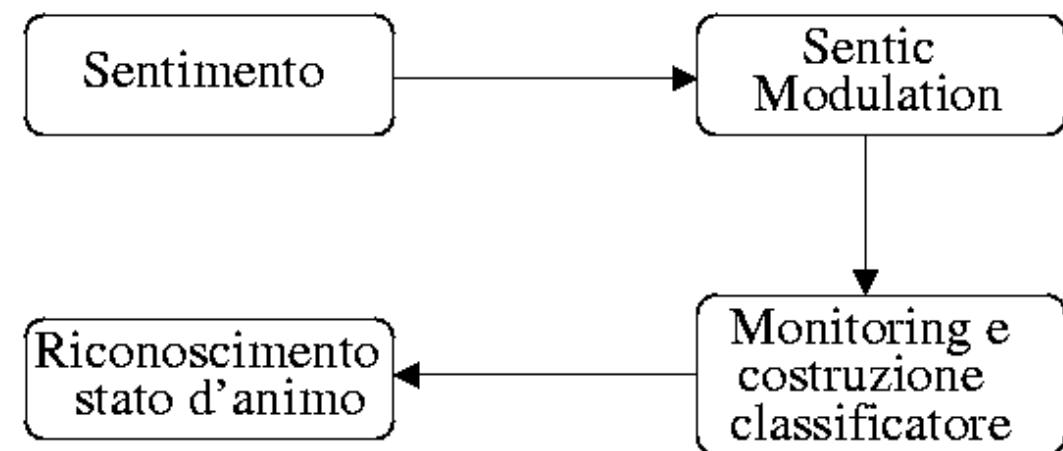
- o espressione fisica di un sentimento:
inflessione della voce, espressioni del viso, battito cardiaco, postura.

■ Problemi:

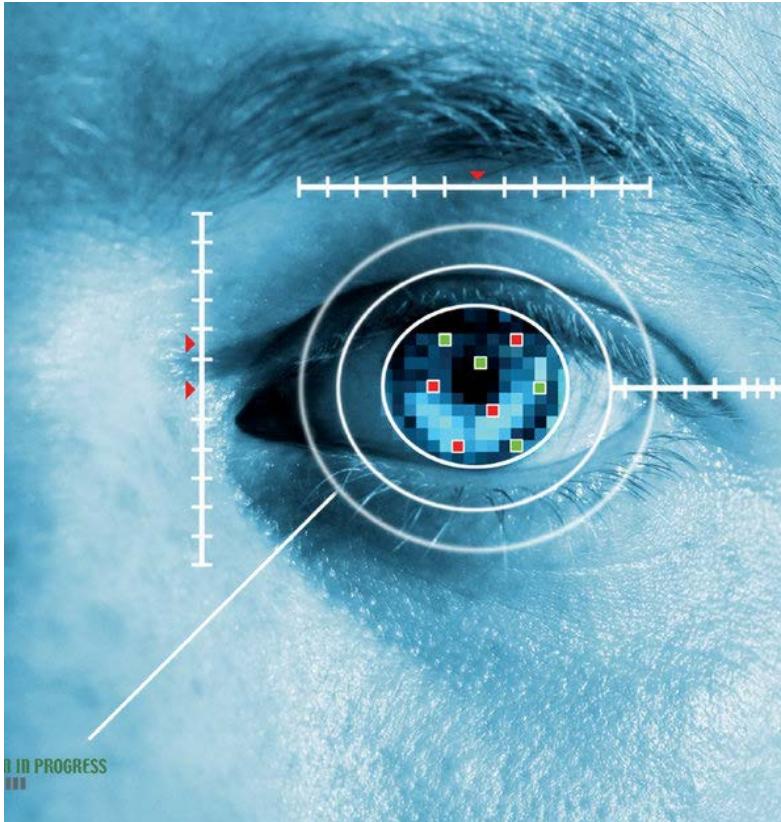
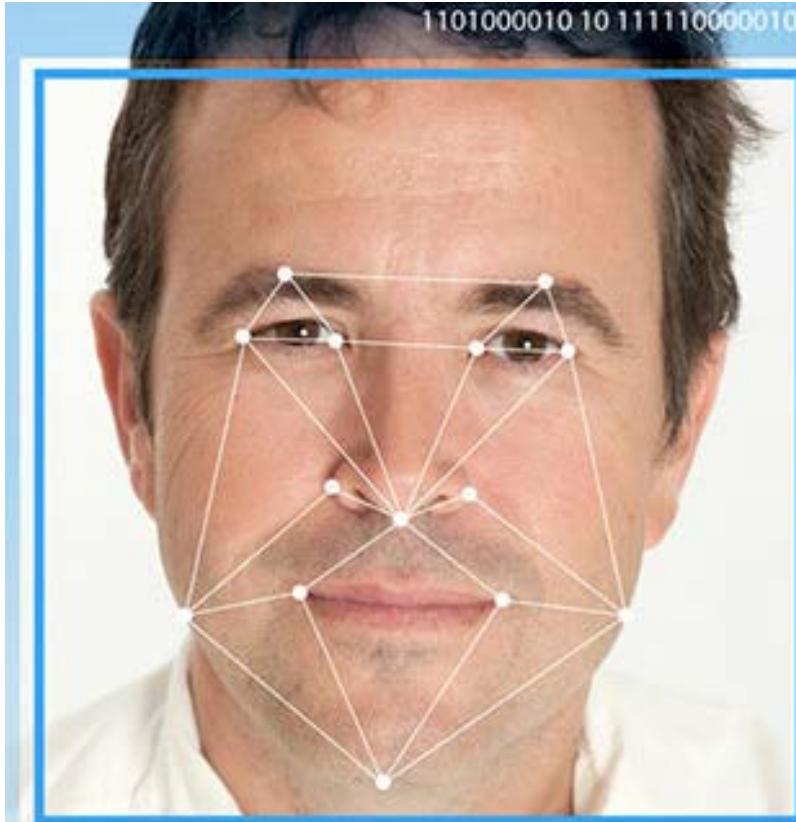
- o *monitoring sentic modulation*;
- o contesto;
- o libera espressione sentic modulation.

■ Applicazioni:

- o mail expressive;
- o video compressione di facce.



Biometrics



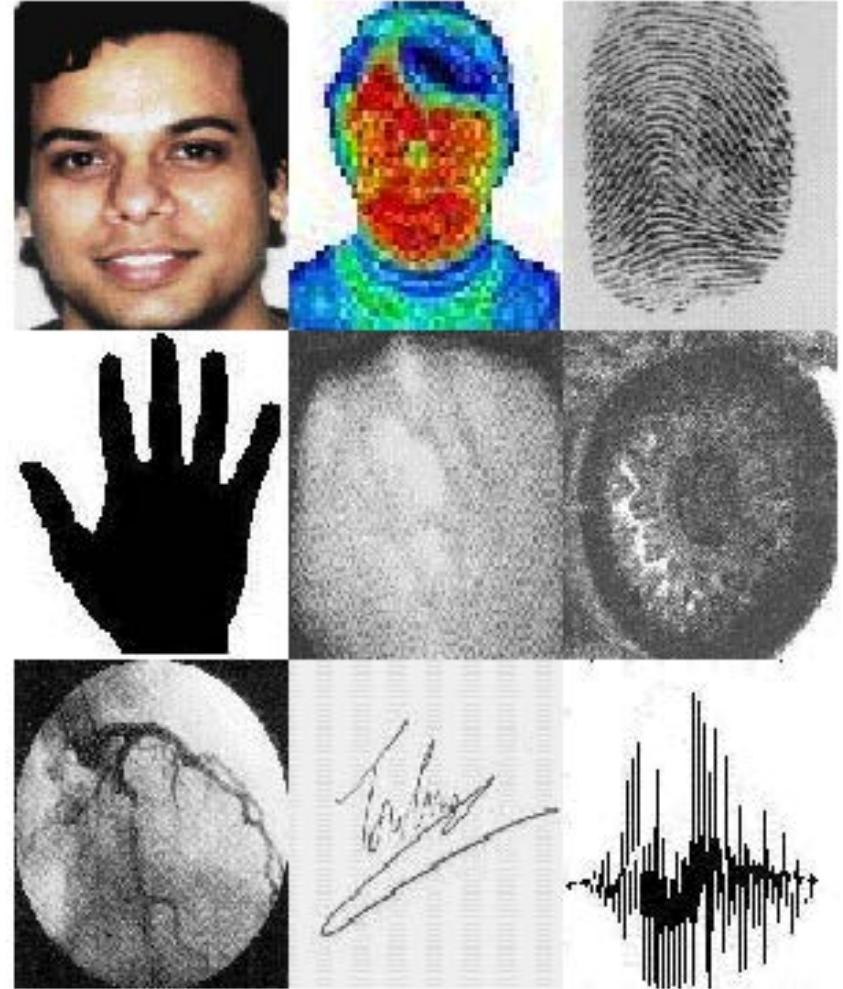
Biometrics

- Definizione:
 - identificazione delle persone attraverso l'analisi delle sue caratteristiche fisiologiche e/o comportamentali.

- Caratteristiche del fattore biometrico:
 - universale (presente in ogni individuo);
 - unico (diverso in ogni individuo);
 - permanente (non rimovibile);
 - quantificabile (misurabile).

Biometrics

- Fattore biometrico:
 - faccia
 - termogramma facciale
 - impronte digitali
 - geometria della mano
 - firma
 - voce
 - iride
- Valutazione di un sistema di biometria:
 - performance;
 - sicurezza;
 - accettabilitá.



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

Biometrics

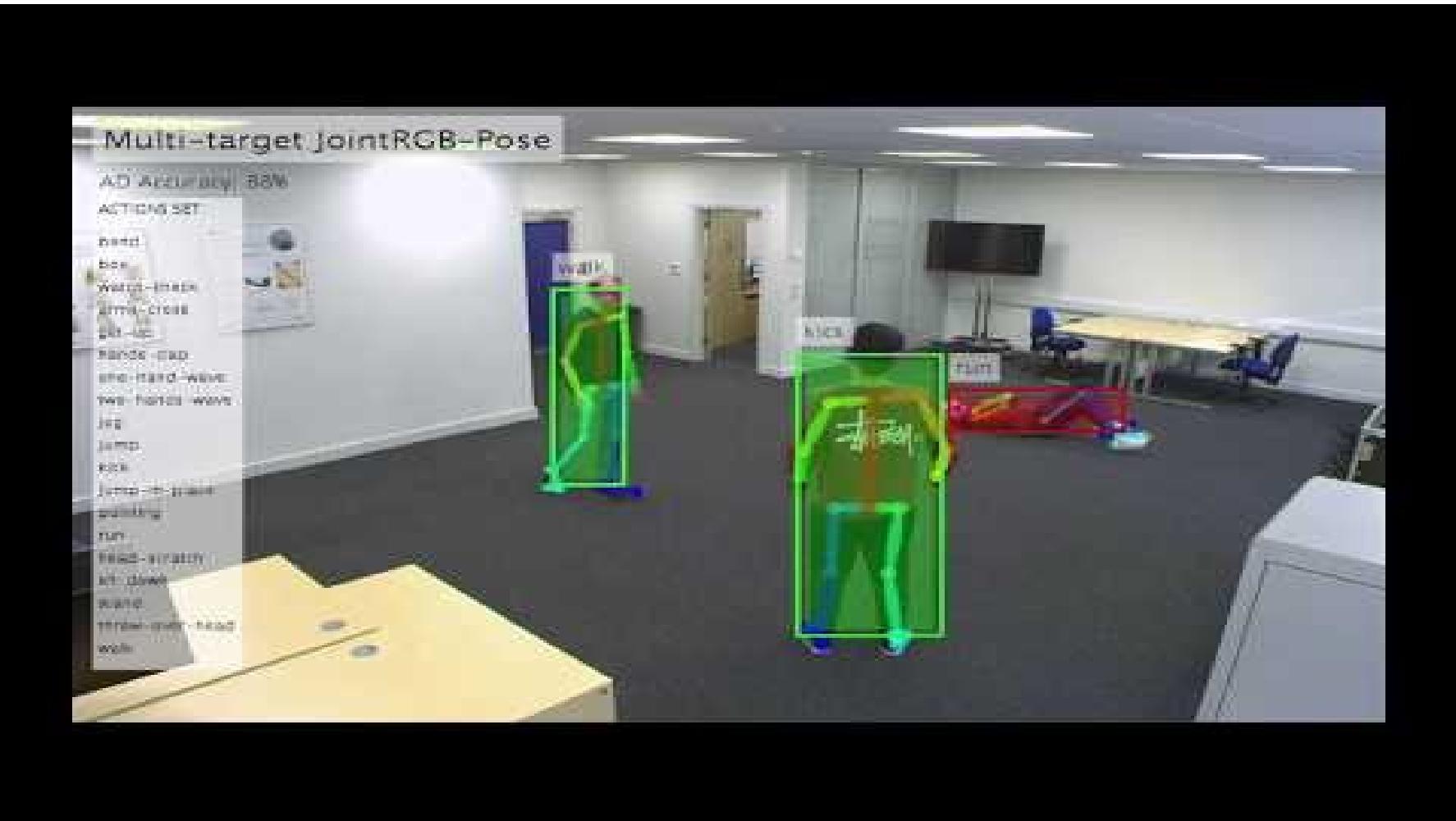
- <https://youtu.be/aE1kA0Jy0Xg>

Classificazione di attività

- Analisi di sequenze video:
 - tracking di oggetti che si muovono;
 - inseguimento nella sequenza;
 - classificazione delle traiettorie;
 - riconoscimento di comportamenti.

- Applicazioni:
 - video sorveglianza;
 - analisi del traffico.

Examples



Examples

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

Bioinformatics

- Scopo:
 - realizzazione di tecniche in grado di estrarre in modo automatico la maggior conoscenza possibile da una sequenza di DNA;
 - attualmente la definizione assume un significato molto ampio.

- Base di partenza scientifica:
 - *Genoma*: insieme del patrimonio genetico di un organismo.
 - *Cromosoma*: struttura superorganizzata di DNA.
 - *DNA*: doppia elica formata da una sequenza di basi nucleotidiche.
 - *Gene*: sequenza di DNA che codifica una proteina.
 - *Proteina*: molecola che svolge una ben determinata funzione nell'organismo.

Problematiche

- *Identificazione di geni*
 - data una sequenza di DNA occorre determinare se rappresenta un gene oppure no
 - 2 approcci: modello per gene/confronto in banca dati.
- *Classificazione di geni*
 - data una sequenza di DNA riconosciuta come gene occorre classificarla sulla base della funzione che esprime.
 - 2 approcci: gene-based (allineamento) e protein-based (modelli per proteine/protein threading).
- *Pattern Discovery*
 - scoprire pattern inusuali, rari e significativi.

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

the optimal situation is having lots of data
for each class

- Learning with scarce data (long tail distribution)
- Zero- and few-shot learning
- Domain adaptation e.g.: In lab, the images are taken during day, but the actual use is during night
→ training and test sets differ from real world scenarios, so I have to adapt my net
- Disentangling representations, fairness
- Multimodal learning *multiple data streams*
- Unsupervised learning
- Self-supervised learning
- Meta-learning (learning to learn)
- Continual, lifelong learning

combining them together in real time
in order to improve the performance,
one fills the lacks of the other and
vice versa