Riconoscimento e recupero dell’informazione per bioinformatica

Rappresentazione dei dati

Manuele Bicego

Corso di Laurea in Bioinformatica

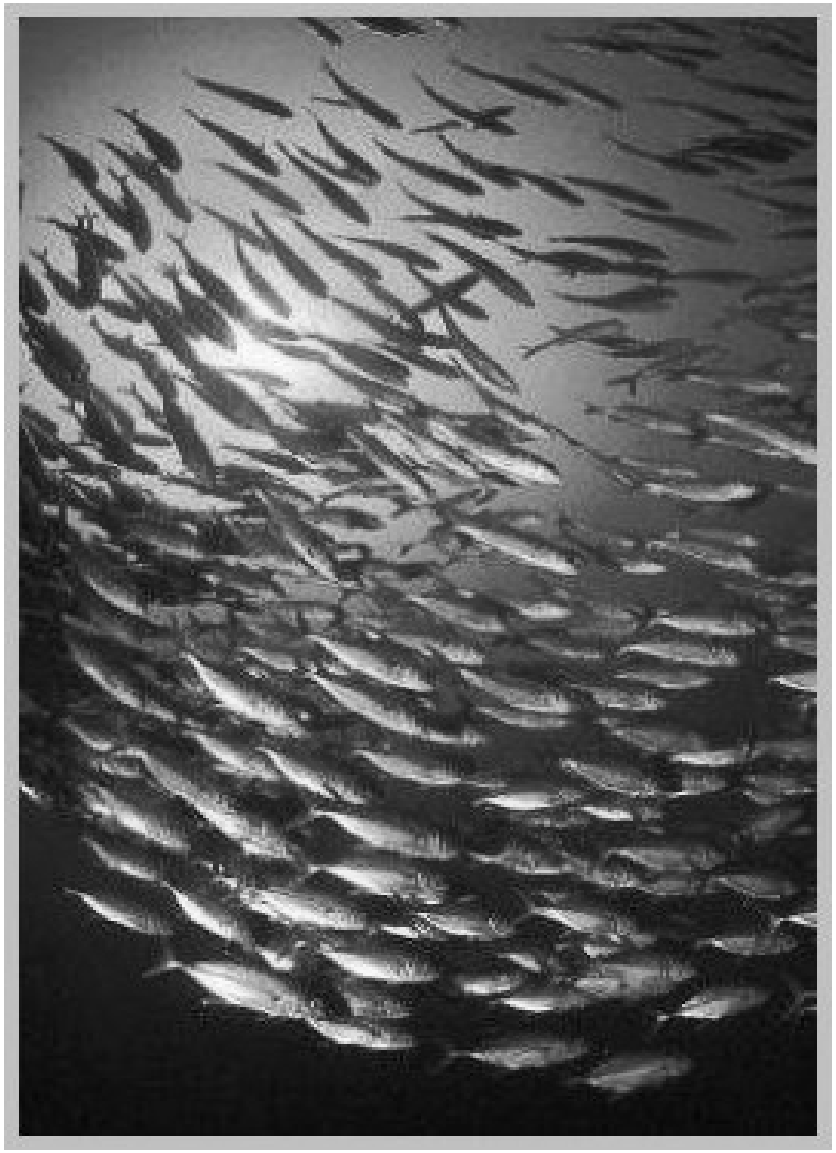
Dipartimento di Informatica - Università di Verona

Sommario

- Rappresentazione dei dati:
  - campionamento
  - Rappresentazione:
    - Estrazione delle features
    - Costruzione del pattern
  - preprocessing (scaling, riduzione del rumore, riduzione della dimensionalità)

La rappresentazione dei dati

- Rappresentazione dei dati: il problema di come rappresentare gli oggetti del problema in esame
  - Rappresentazione che un calcolatore possa capire
  - Rappresentazione utilizzata per costruire il modello, per fare il testing
  - Scelta cruciale



La rappresentazione dei dati

1. Campionamento (acquisizione dati)

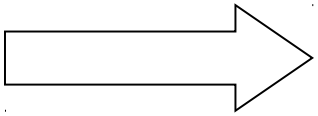
2. Rappresentazione: estrazione delle features e costruzione del pattern

3. Preprocessing: scaling, riduzione del rumore, riduzione della dimensionalità

Fase 1: campionamento

Rappresenta la raccolta dati vera e propria: effettuare delle misure sugli oggetti del problema

Esempio 1: modellare pesci

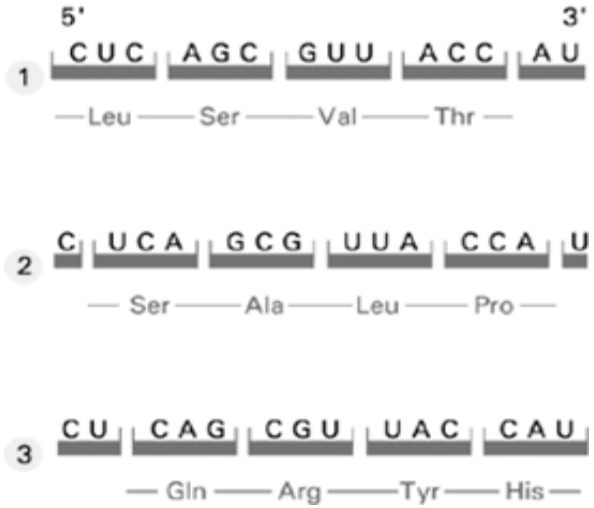


campionamento

immagine (dato grezzo)

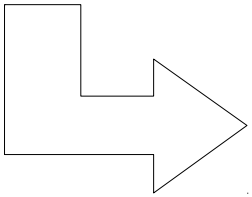
Fase 1: campionamento

- Esempio 2: filogenesi di microrganismi
  - l’obiettivo è sequenziare il gene scelto per fare filogenesi nei microrganismi in esame
  - far crescere i batteri (scelta del terreno di crescita,...)
  - estrarre il DNA e amplificare quello dei geni prescelti per la filogenesi (scelta dei primer, scelta dei parametri della PCR)
  - Sequenziare

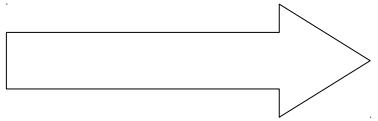


Fase 1: campionamento

⇨ Esempio 3: riconoscimento di volti



campionamento



Fase 1: campionamento

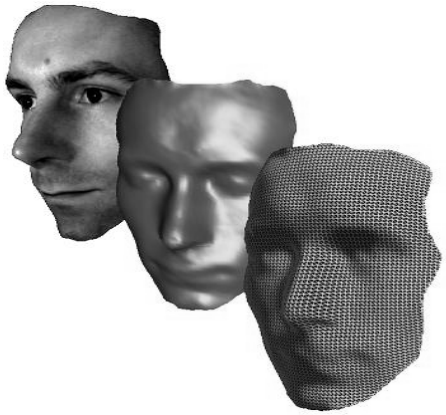
SCELTA CRUCIALE: il tipo di sensore da utilizzare

⇨ Esempio: riconoscimento di volti

telecamera tradizionale:

PRO: alta risoluzione

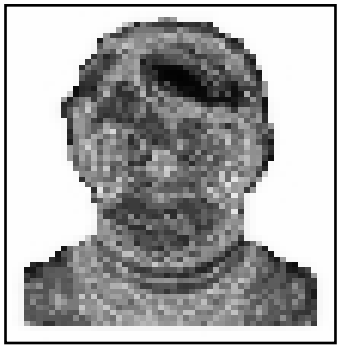
CON: sensibile ai cambi di illuminazione



Sensore 3D:

PRO: risolve il problema della liveness

CON: computazionalmente oneroso



telecamera a infrarosso:

PRO: funziona anche al buio

CON: risoluzione bassa

CON: Info interessante?

Fase 1: campionamento

Problemi da tenere in considerazione:

- ⇨ frequenza di campionamento: capacità di modellare evoluzione temporale
- ⇨ risoluzione (dipende dal sensore): quanti dettagli si riescono a recuperare
- ⇨ capacità di gestire cambiamenti di condizioni al contorno (eg. cambi di illuminazione)

il campionamento dipende strettamente dal problema (e tipicamente è deciso dall'esperto)

La rappresentazione dei dati

1. Campionamento (acquisizione dati)

2. Rappresentazione: estrazione delle features e costruzione del pattern

3. Preprocessing: scaling, riduzione del rumore, riduzione della dimensionalità

Fase 2: Rappresentazione

- Problema vero e proprio: rappresentare gli oggetti del problema in esame a partire dalle misurazioni fatte durante il campionamento

- Scelta semplice: si utilizzano direttamente i dati derivanti dal campionamento (dati grezzi)
  - Problema: presenza di informazione irrilevante

Fase 2: Rappresentazione

- Soluzione: si elaborano i dati provenienti dal campionamento: si aggregano misure, si estraggono nuovi dati, si scartano le informazioni irrilevanti, etc etc

- Due fasi:

  - Estrazione di features (caratteristiche)

  - Costruzione del pattern (insieme di features): aggregazione delle features

Esempio:

⇨ Opzione 1:

Features

Altezza e larghezza

Pattern:

vettore di due dimensioni [a,l]

⇨ Opzione 2:

Features

Coordinate del contorno dell'oggetto, a partire dalla coda

Pattern:

sequenza di vettori di due dimensioni (le coordinate di ogni punto del contorno)

x1,y1
x2,y2
x3,y3
...
xn,yn

Opzione 1

Vantaggi:

- ⇨ rappresentazione compatta
- ⇨ ogni oggetto è un punto in uno spazio vettoriale bidimensionale
- ⇨ non è difficile da calcolare

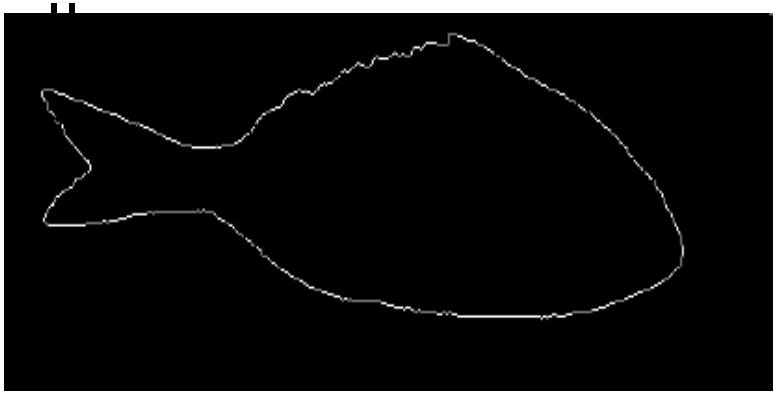
Svantaggi

- ⇨ troppo semplificata: non riesce a modellare la forma del pesce, il colore

Opzione 2

Vantaggi

- ⇨rappresentazione più ricca: modella la forma del pesce



Svantaggi

- ⇨più complicata da calcolare
- ⇨il pattern risultante è una sequenza (che può essere di dimensione diversa a seconda del pesce). Non siamo quindi più in uno spazio vettoriale. Non modella il colore

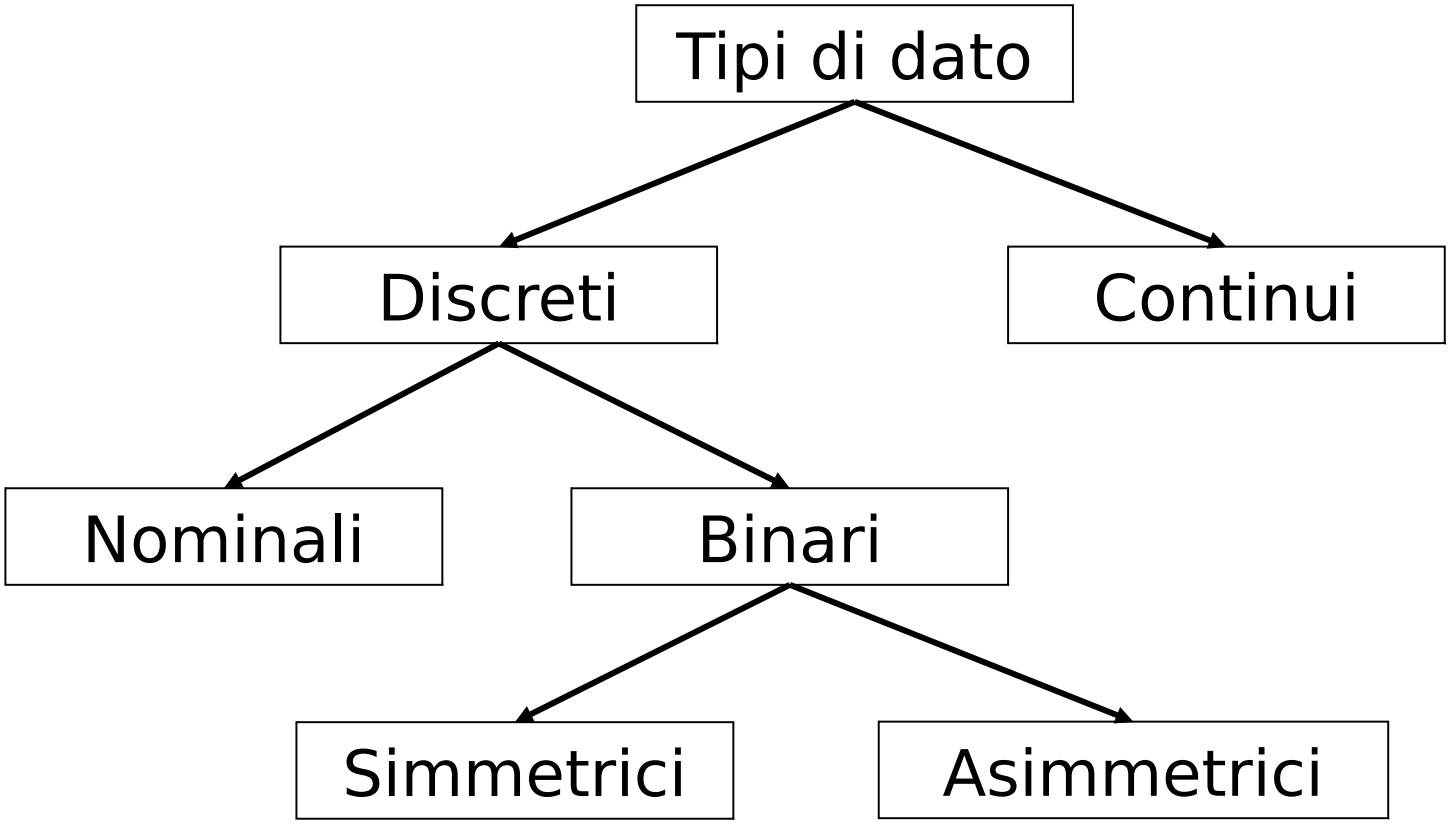
Estrazione delle features

Feature: caratteristica del problema in esame

- ⇨ rilevante
- ⇨ discriminante
- ⇨ quantificabile a partire dai dati grezzi
- ⇨ (interpretabile)

La scelta di queste features è chiaramente cruciale

Tipologie di features: tipi di dato



Tipi di dato

- Continui: il valore della feature può assumere un numero infinito di valori (e.g. numeri reali)
  - attenzione: campionamento!

- Discreti: il valore della feature può essere solo uno di un insieme finito di valori possibili

- Nominali: il valore della feature può essere uno di un insieme di nomi (o di simboli) – tipo di dato discreto
- EXE: sequenza di DNA:
  - T Timina
  - A Adenina
  - G Guanina
  - C Citosina

Tipi di dato

- Dati binari: dati che possono assumere solo due valori:
  - 0/1, vero/falso
- Dati binari simmetrici: i due valori sono ugualmente importanti
  - esempio: maschio/femmina
- Dati binari asimmetrici: uno dei due valori porta più informazione dell'altro
  - esempio:
    - sì: presenza di un attributo (ad esempio una malattia)
    - no: assenza

Costruzione del pattern

Pattern: insieme di features/misure relative allo stesso oggetto

Problema della costruzione del pattern: “Come mettere assieme le varie *features*”

⇨ Un vettore, una sequenza, un grafo....

… vediamo i vari tipi di pattern

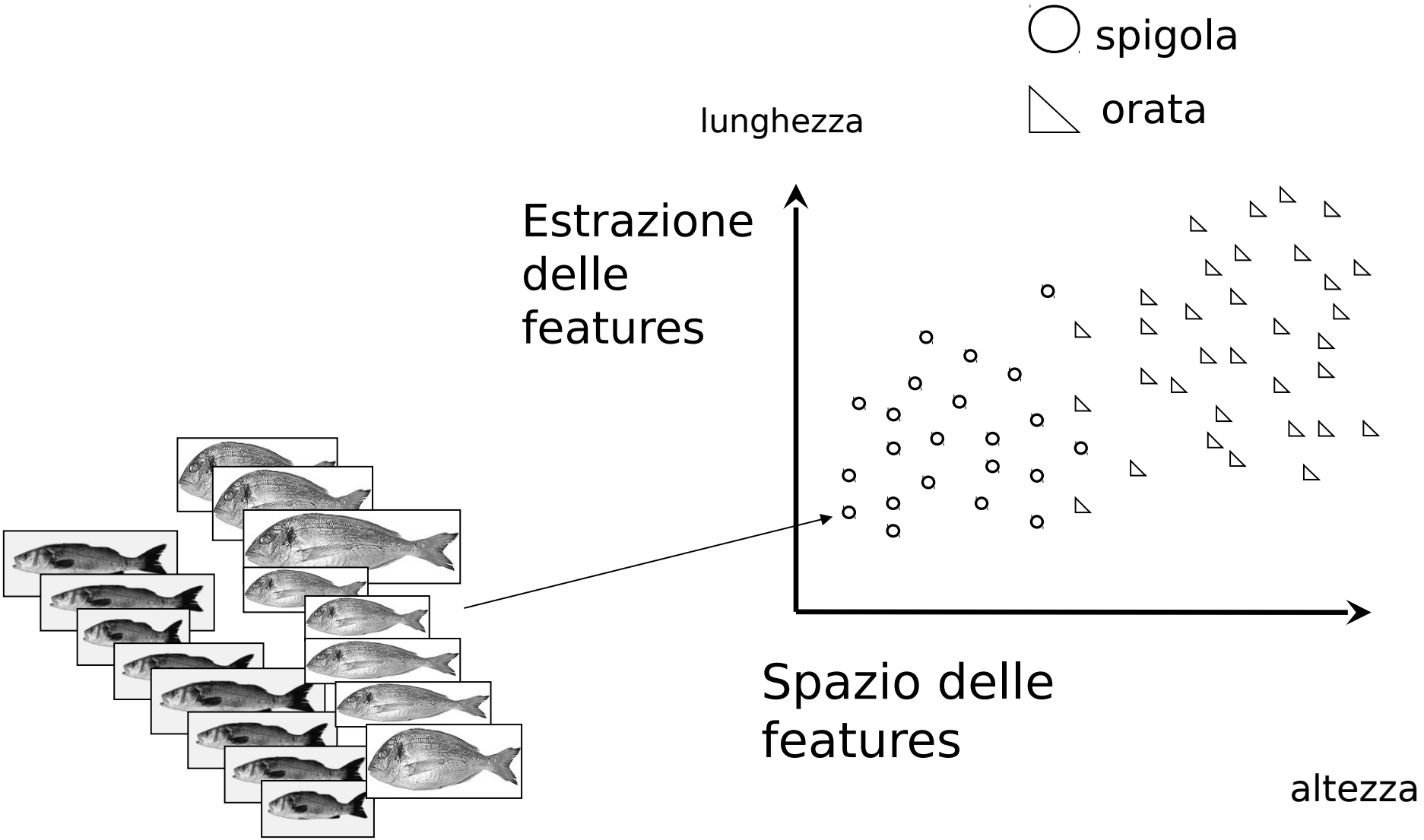
Tipi di pattern: i vettori

Dati vettoriali (formato più diffuso)

- ⇨ per ogni oggetto si ha un insieme prefissato di features, messe in ordine in un vettore
  - ⇨ Esempio:

- ⇨ il vettore è ordinato
- ⇨ l'oggetto viene proiettato in un punto in uno spazio d-dimensionale, detto "spazio delle features"
  - ⇨ "d" è il numero delle features

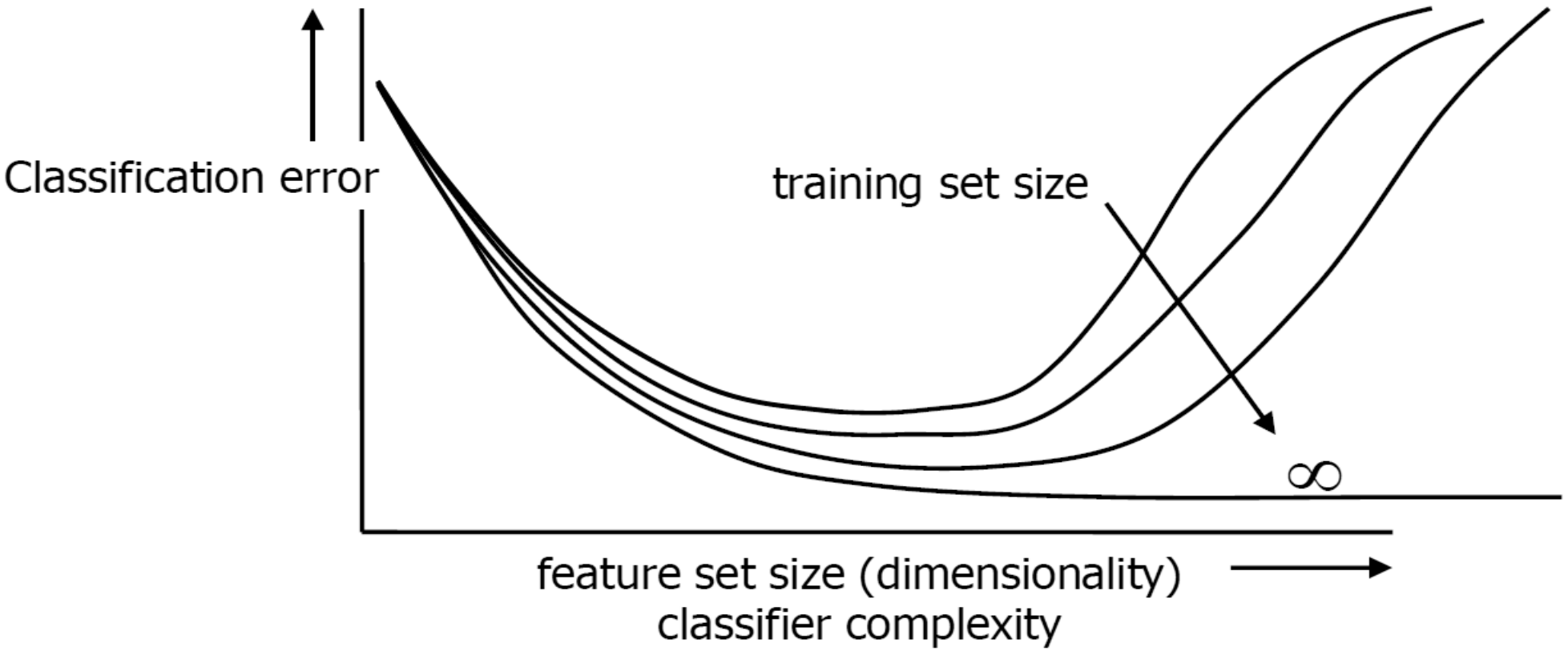
Tipi di pattern: i vettori



Commenti su spazi vettoriali

- La scelta delle features è cruciale
- Utilizzando molte features gli spazi diventano estremamente grandi
  - problema di visualizzare i dati
  - problema della curse of dimensionality
- Curse of dimensionality: insieme di problemi che possono nascere se lo spazio ha una dimensionalità troppo elevata rispetto al numero di oggetti o se il modello è troppo complesso
  - Esempio: 5 punti, 100 features
  - Esempio: 5 punti, polinomio di grado 10

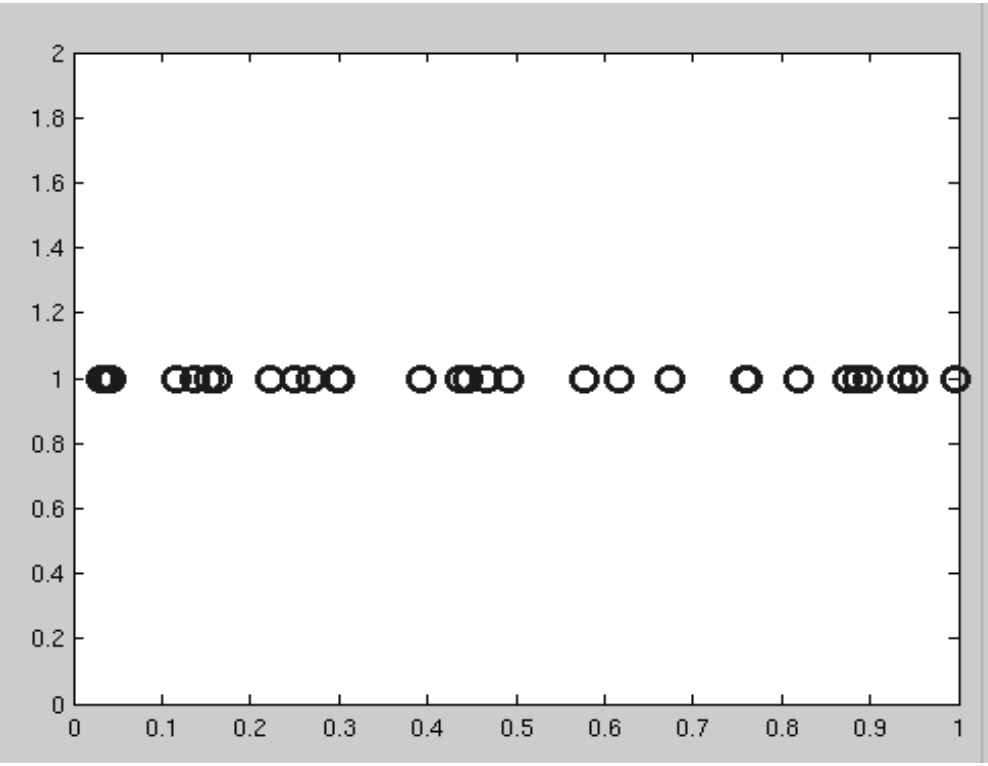
Curse of dimensionality: aumentando troppo il numero delle features l'errore di classificazione cresce



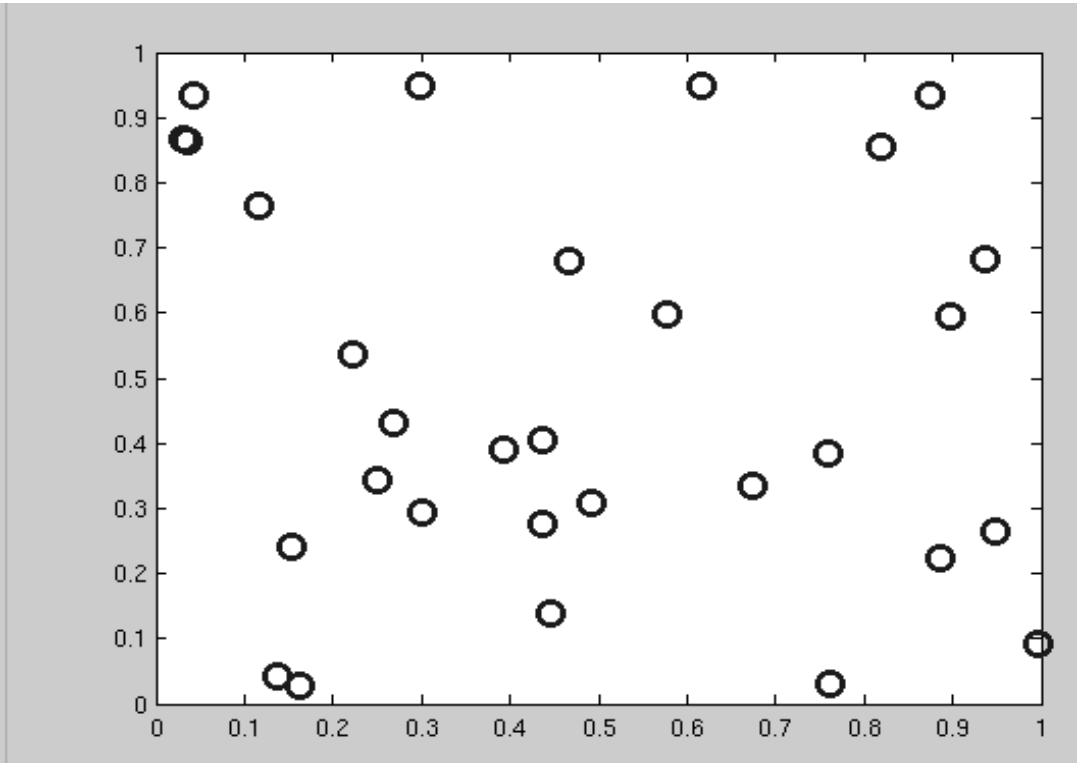
Peaking Phenomenon, Overtraining, Curse of Dimensionality, Rao's Paradox

Curse of dimensionality

- Problema 1: molte features, lo spazio diventa vuoto
  - Poco espressivo, poco denso, facile costruire classificatori che non generalizzano bene



30 punti in uno spazio 1D: molte sovrapposizioni



30 punti in uno spazio 2D: pochissime sovrapposizioni

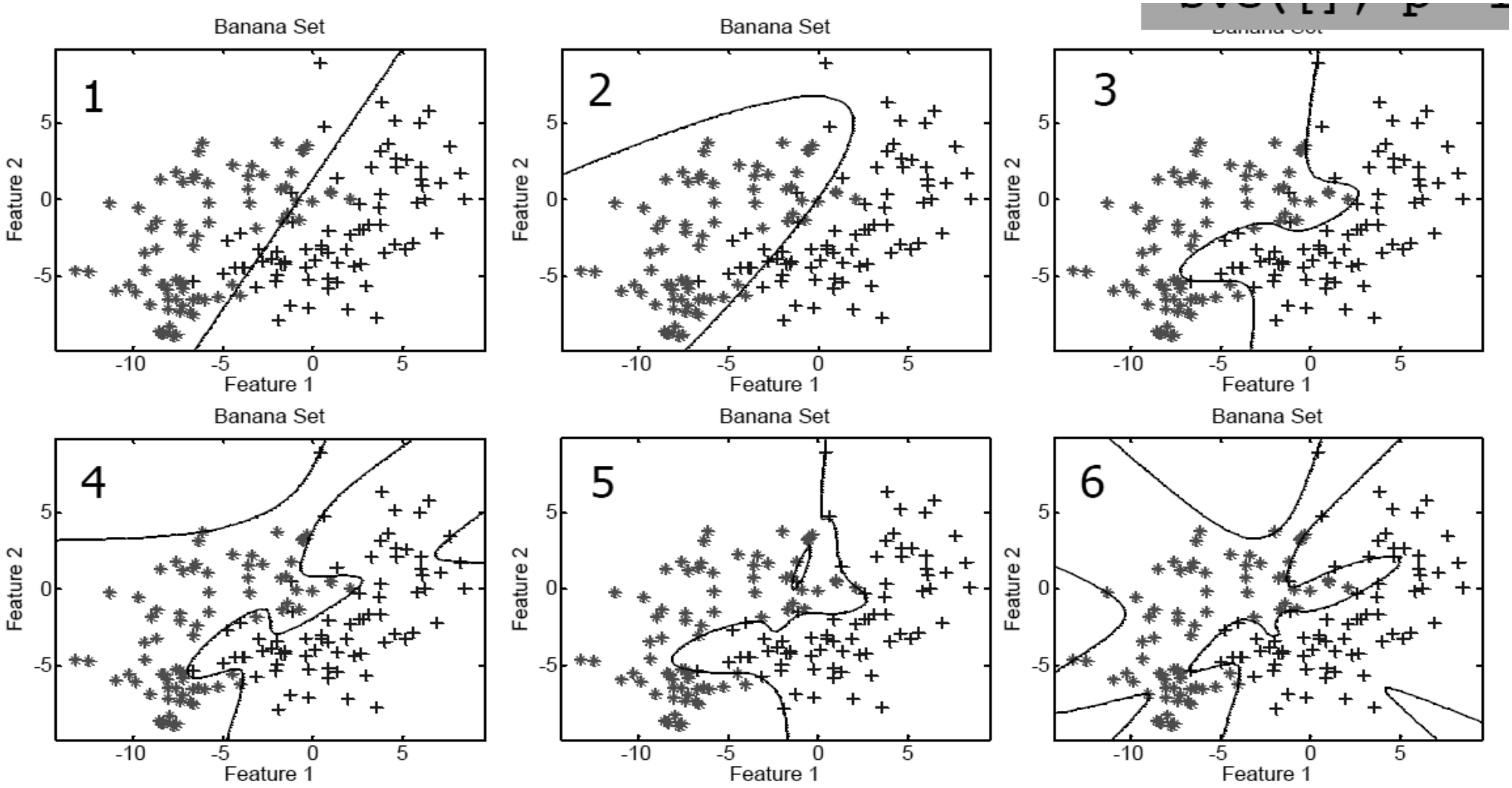
Curse of dimensionality

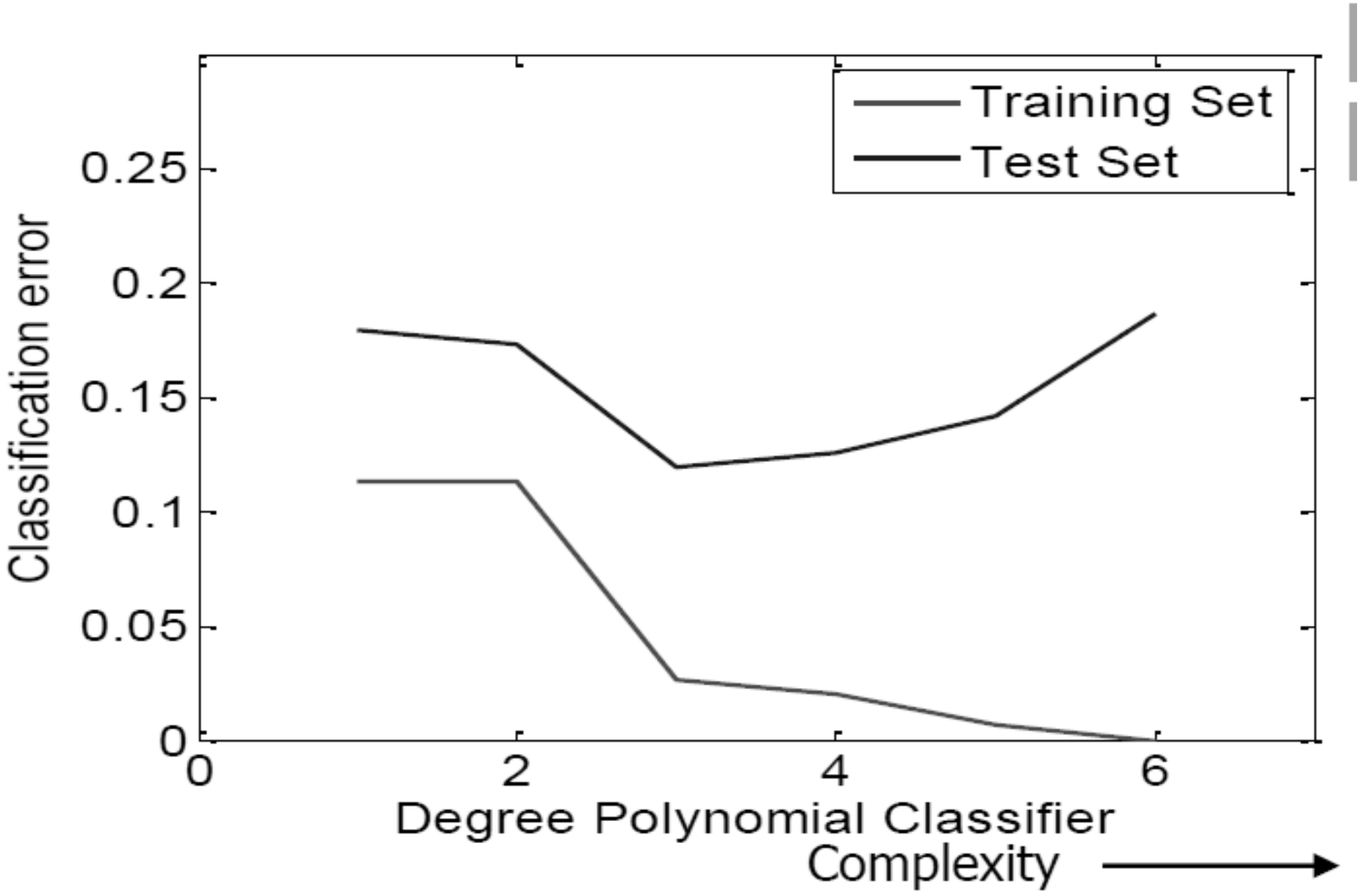
- Secondo problema: tipicamente è più difficile stimare i parametri
  - Meno “dati” per parametro
  - Esempio: una media in uno spazio bidimensionale è fatta di due valori, in uno spazio 10D di 10 valori, tutti da stimare a partire dallo stesso numero di oggetti

- Esistono metodi che riducono la dimensionalità di questi spazi (vedremo in seguito)

Curse of dimensionality

- Stesso fenomeno se si aumenta troppo la complessità di un classificatore



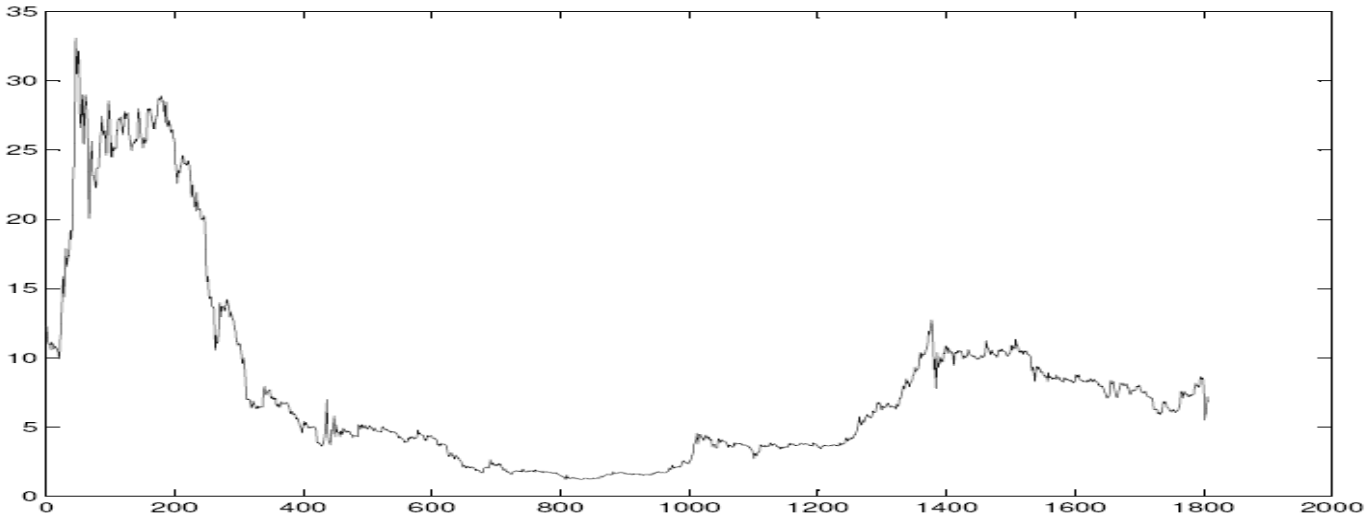

Training Set
Test Set
Classification error
0.25
0.2
0.15
0.1
0.05
0
0
2
4
6
Degree Polynomial Classifier
Complexity

Tipi di pattern: le sequenze

- Sequenze: dati che si presentano in forma ordinata e sequenziale (uno dopo l'altro): è importante l'ordine

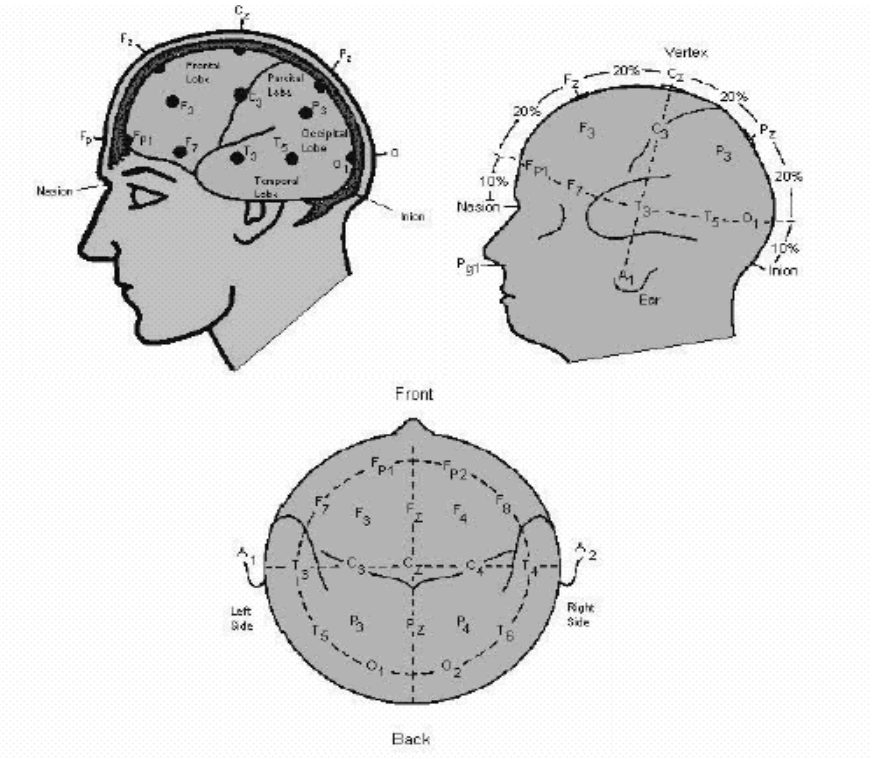
  $x_1, x_2, x_3, x_4 \ldots x_T$

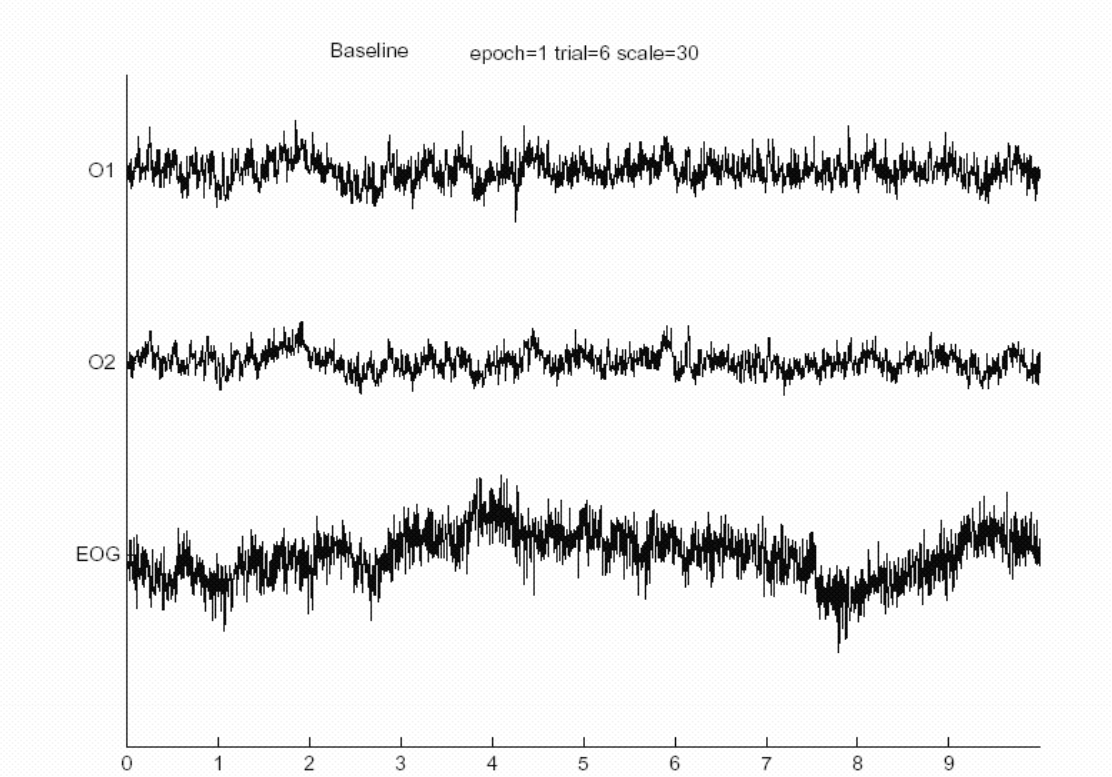
- Sequenze temporali:
  - sequenza di features che rappresentano la misurazione di un fenomeno a intervalli di tempo regolari



Esempio 1: indici di mercato (DowJones)

Tipi di pattern: le sequenze





⇨ Esempio 2: segnali elettro encefalografici (EEG)

Tipi di pattern: le sequenze

Tracking: “inseguire” le persone

Tipi di pattern: le sequenze

- Sequenze non temporali:
  - sequenze dove l’ordine non è dato dal tempo
- Esempio 1:
  - sequenze nulceotidiche

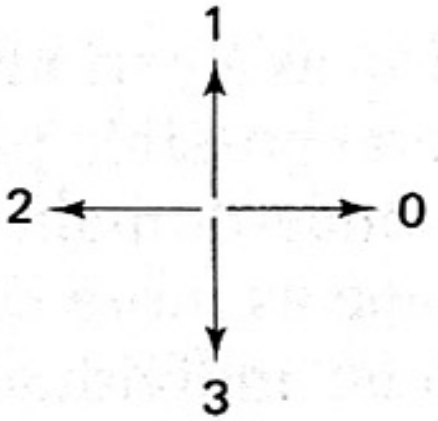
```
atgcgatcgatcgatcgatcaggcgcgctacgagcggcgaggacct
catcatcgatcag
```

  - sequenze aminoacidiche

```
MRPQAPGSLVDPNEDELRMAPWYWGRISREEAKSILHGKPDGSFLVRDALSMKGEYTLTLMKDGCEK
LIKICHMDRKYGFIETDLFNSVVEMINYYKENSLSMYNKTLDITLSNPIVRAREDEESQPHGDLCLL
SNEFIRTCQLLQNLEQNLENKRNSFNAIREELQEKKLHQSVFGNTEKIFRNQIKLNESFMKAPADA…
…
```

Tipi di pattern: le sequenze

⇨ Esempio 2: codifica di un contorno di una forma 2D



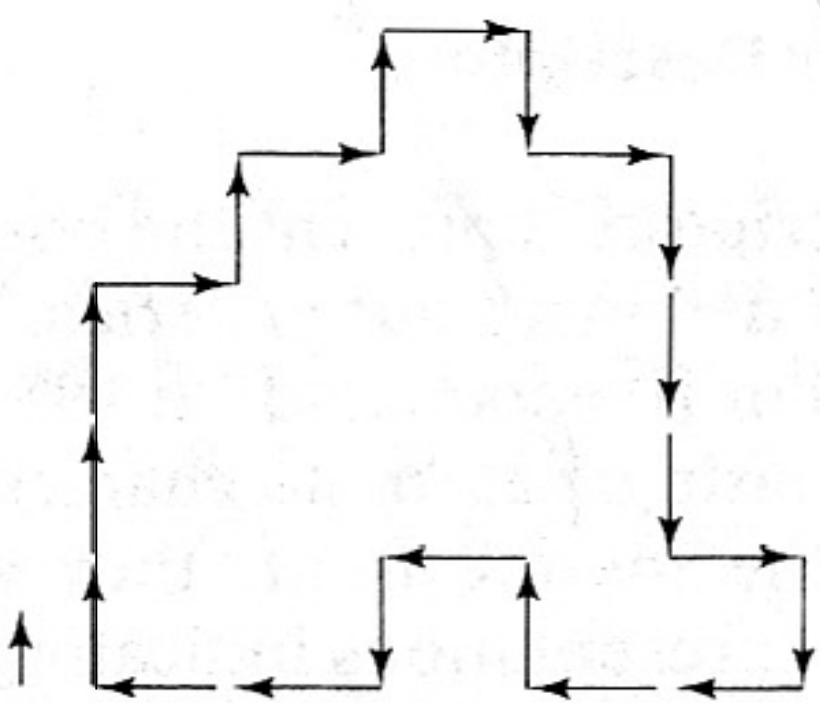
contorno della
forma

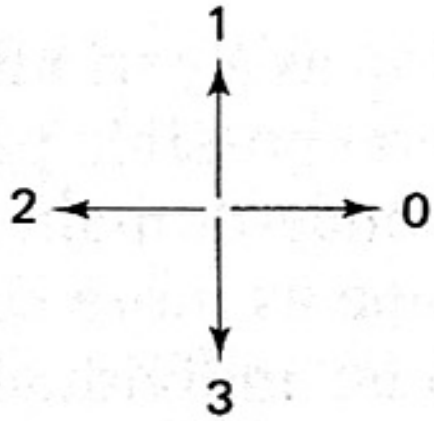
Chain code: specifica la direzione del contorno ad ogni punto di edge

le direzioni sono quantizzate in 4/8 valori

Rappresentazione: coordinate del punto iniziale e una sequenza di chain code che segue il contorno.

Tipi di pattern: le sequenze





(b) Chain code: 1 1 1 0 1 0 1 0 3 0 3 3 3 0 3 2 2 1 2 3 2 2
(c) Derivative: 1 0 0 3 1 3 1 3 3 1 3 0 0 1 3 3 0 3 1 1 3 0

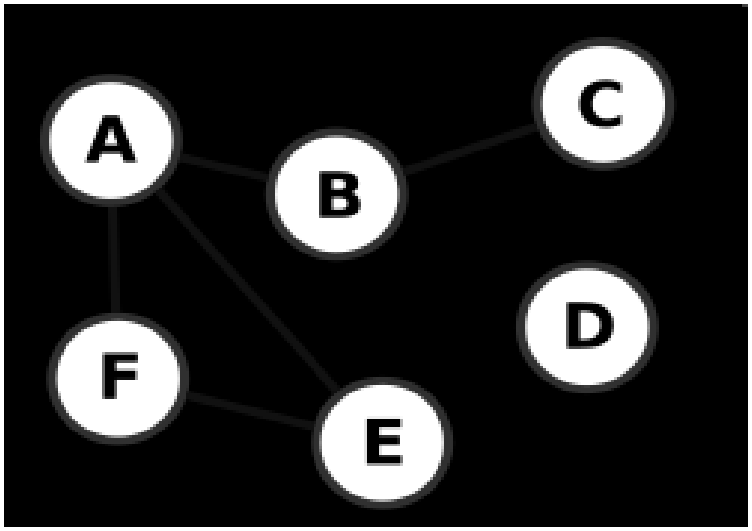
Tipi di pattern: le sequenze

Riassumendo le caratteristiche delle sequenze:

- ⇨ esiste un concetto di “sequenzialità” dei dati
- ⇨ l’ordine è importante
  - ⇨ “xyyyx” non è la stessa cosa di “yyyxx”
- ⇨ la lunghezza di due sequenze diverse potrebbe essere diversa
  - ⇨ e.g. lunghezza contorno di un oggetto
  - ⇨ e.g. lunghezza gene
- ⇨ Sequenze di lunghezza diversa non possono essere rappresentate in uno spazio vettoriale (gli spazi necessari sarebbero di dimensione diversa)

Tipi di pattern: i grafi

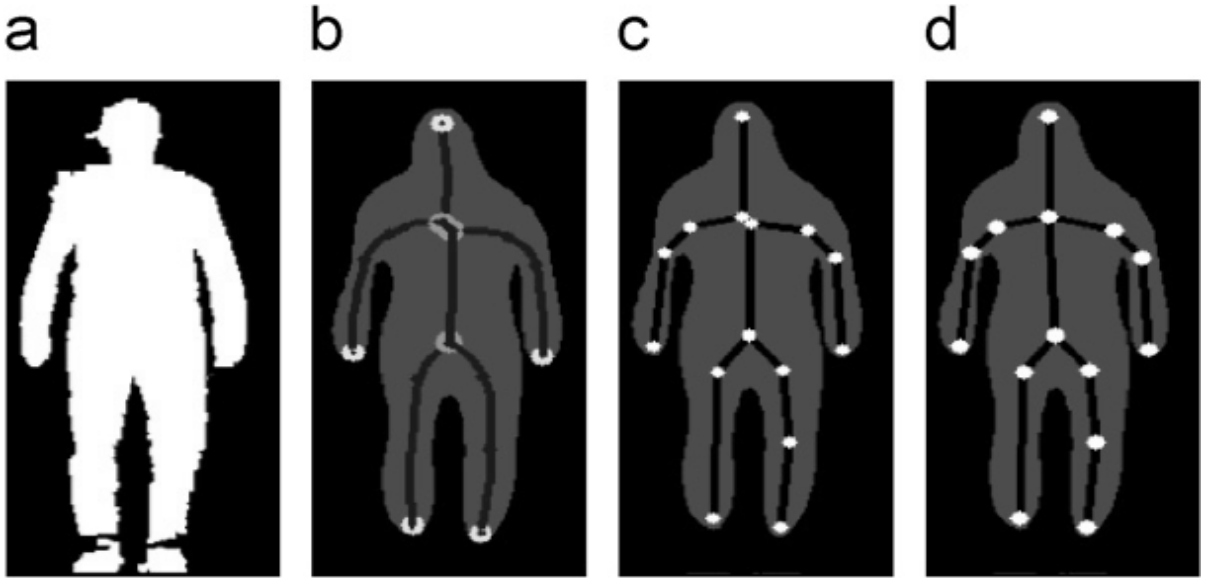
- I grafi (e gli alberi) rappresentano un insieme di nodi collegati da archi (vedi il corso di Algoritmi)
- Codificano la relazione tra parti
  - esempio: vicinanza, connettività, etc

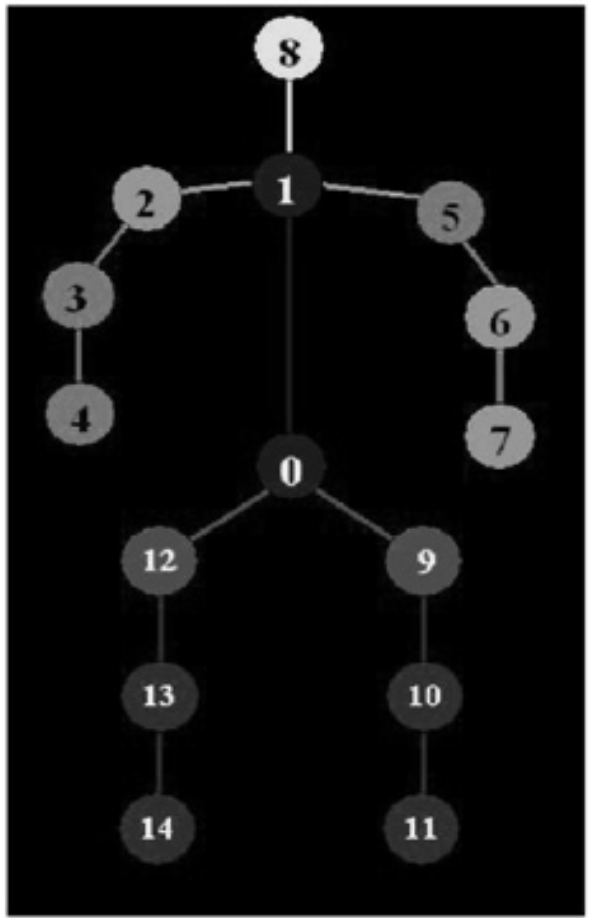


Maggiori dettagli nella parte del corso con la Prof.ssa Giugno

Tipi di pattern: i grafi

⇨ Esempio 1: modellare le diverse parti di un corpo umano



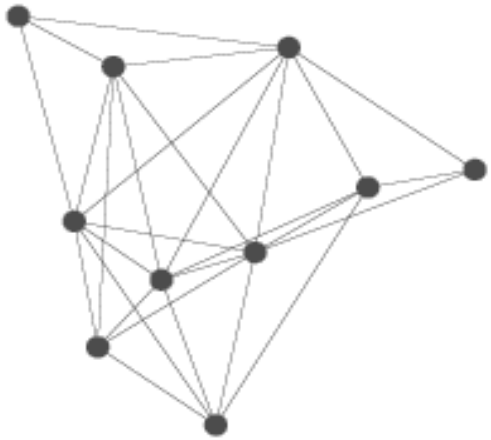


Tipi di pattern: i grafi

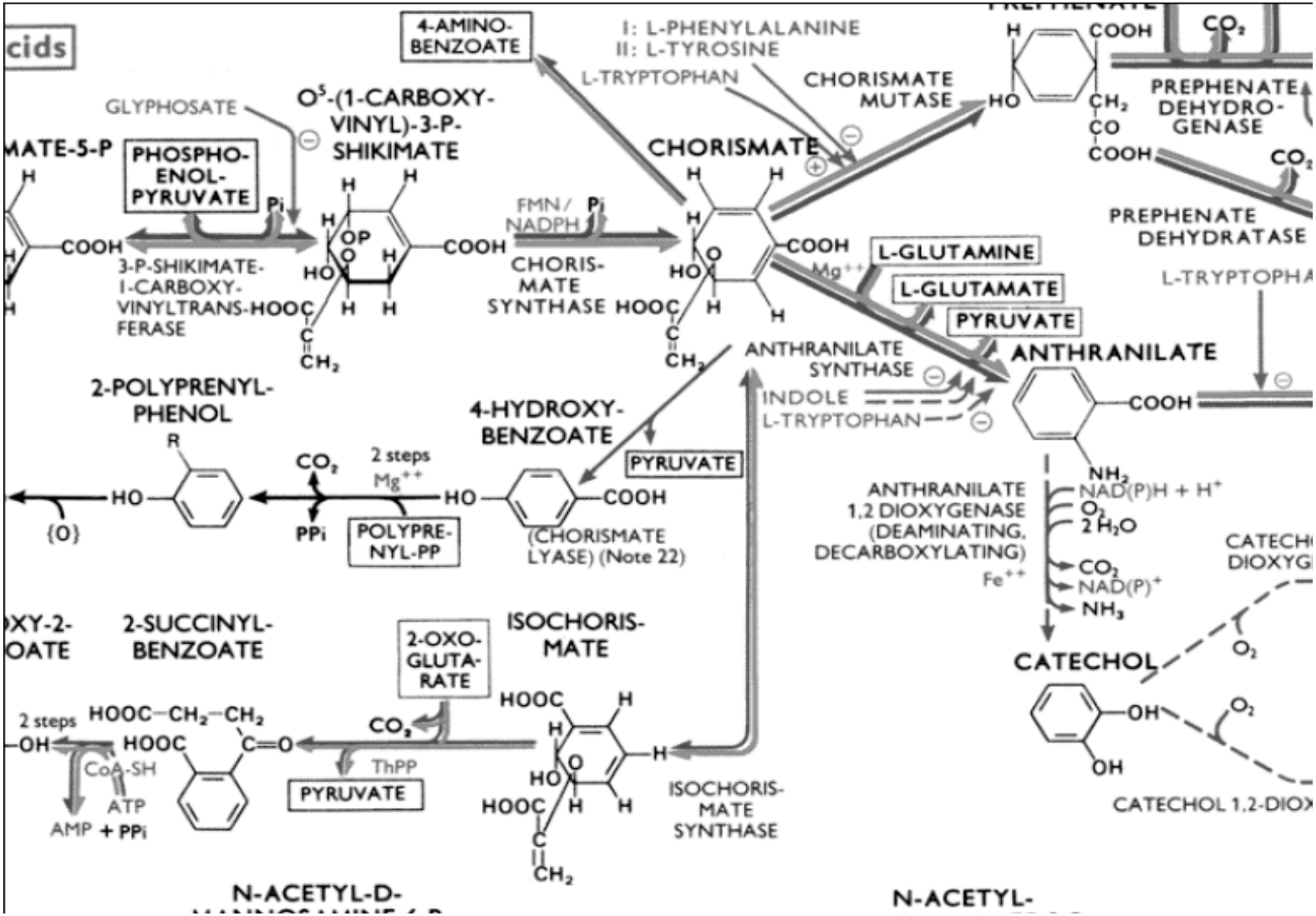
- Esempio 2: Protein-Protein Interaction Networks
- Grafi dove vengono visualizzate le interazioni tra le proteine
  - fondamentali in biologia

- list of N **proteins** (nodes)
- list of **protein pairs** (edges)

This is an **undirected, unweighted graph**



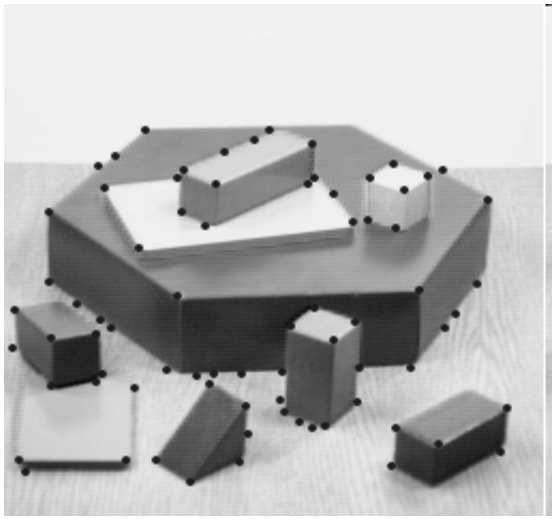
⇨ Esempio 3: percorsi metabolici: insieme di reazioni di composti chimici per svolgere un processo cellulare



Tipi di pattern: gli insiemi

- insiemi: collezione non ordinata di dati a cardinalità variabile
- insieme di descrittori tutti relativi alla stessa entità, ma non ordinati

- Esempio 1: angoli in un'immagine

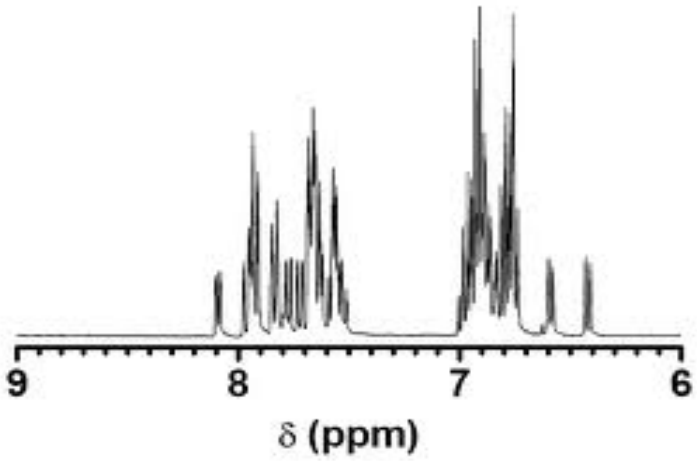


non c'è un ordine

in ogni immagine ce ne può essere un numero variabile

Tipi di pattern: gli insiemi

- Esempio 2: picchi in uno spettro NMR:
  - numero di picchi può essere diverso in due diversi spettri
  - i picchi non sono ordinati (si possono forse ordinare per ampiezza, o per ppm, ma in generale questo non è possibile)



Altri tipi di pattern

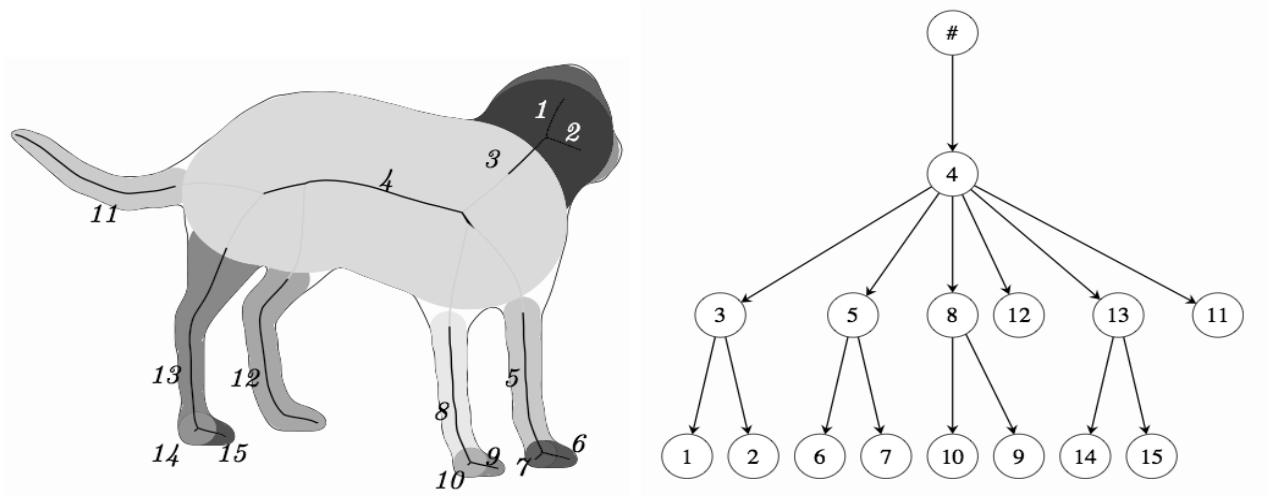
Transactions

- Dato un insieme di oggetti, una transaction è un sottoinsieme di questi oggetti
- Esempio 1: market basket analysis
  - analisi dei prodotti acquistati da un consumatore
  - dati N prodotti disponibili
  - solo M (M<N) vengono acquistati
- Esempio 2: dato l'insieme di geni, caratterizziamo quelli espressi ("attivati") in una determinata condizione (attenzione, livello di espressione)

- tipicamente rappresentati da un vettore binario lungo N
  - 0 indica l'assenza dell'oggetto
  - 1 indica la presenza di un oggetto

Summary: vettori vs altri tipi di pattern

- Tipicamente pattern complessi (sequenze, grafi, …) sono più “espressivi” dei vettori



- MA: molte delle tecniche di Pattern Recognition Statistica funzionano solo nel caso di spazi vettoriali (si pensi ai classificatori che usano la media!)
- In caso di dati non vettoriali la maggior parte delle assunzioni non vale

Summary: vettori vs altri tipi di pattern

Soluzioni possibili:

- ⇨ creazione/utilizzo di metodi di classificazione o di clustering che lavorano con questi tipi di dato
  - ⇨ Esempio: definizione di misure di similarità che riescano a tenere in considerazione la struttura del pattern
- ⇨ incapsulamento (embedding) in uno spazio vettoriale

Dettagli

- SOLUZIONE 1: creazione/utilizzo di metodi di classificazione/clustering capaci di lavorare con questi tipi di dato
  - metodi che non necessitano di uno spazio vettoriale
  - Esempio 1: Hidden Markov Models per sequenze
  - Esempio 2: metodi che basano il loro funzionamento sulla definizione di misure di distanza
    - il problema viene risolto con la definizione di misure di similarità che riescano a tenere in considerazione la struttura non vettoriale del pattern (e.g. distanza dynamic time warping per riconoscimento del parlato)
    - si vedrà meglio in seguito!

Dettagli

- SOLUZIONE 2: incapsulamento (embedding) in uno spazio vettoriale:
  - L’idea è quella di riportare il problema in uno spazio vettoriale

- Vediamo due possibili classi di approcci:
  1. estrazione di features da dati non vettoriali
  2. incapsulamento (embedding) tramite misure di similarità

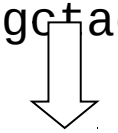
Dettagli

Approccio 1: estrazione di un insieme predeterminato di features dall'oggetto non vettoriale

ESEMPIO

- ⇨ Sequenza nucleotidica
- ⇨ Metodo di estrazione di features: si conta la frequenza di A, T, C, G

`atgcgatcgatcgatcgatcaggcgcgctacgagcggcgaggacctcatcatcgatcag`



[14,10,17,18]

Adesso ogni sequenza è un punto in uno spazio 4-dimensionale

Vantaggi: spazio vettoriale, facile

Svantaggi: che feature occorre estrarre? Quanta informazione si perde?

Dettagli

Approccio 2: incapsulamento in uno spazio dove vengono preservate le caratteristiche di similarità tra gli oggetti

ESEMPIO: Multidimensional Scaling

⇨ si calcolano le distanze tra gli oggetti non vettoriali

⇨ si “creano” dei punti in uno spazio vettoriale in modo che sia preservata la distanza

Multi dimensional Scaling (MDS)

- insieme di tecniche statistiche tipicamente utilizzate per visualizzare dati in uno spazio bi- o tri-dimensionale

- Il punto di partenza è una matrice di similarità (o dissimilarità/distanza) tra tutte le possibili coppie di pattern

$$\mathbf{D}_1 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0.0 & & & & \\ 2.0 & 0.0 & & & \\ 6.0 & 5.0 & 0.0 & & \\ 10.0 & 9.0 & 4.0 & 0.0 & \\ 9.0 & 8.0 & 5.0 & 3.0 & 0.0 \end{pmatrix} \end{array}$$

Multi dimensional Scaling (MDS)

- Dato un'insieme di oggetti x1..xN, abbiamo a disposizione la matrice delle distanze D = [$d_{i,s}$]
  - $d_{i,s}$ = distanza tra x_i e x_s
- GOAL: creare un insieme **Y** = [y_1..y_N] in uno spazio a 2 o 3 dimensioni tale che le distanze $d^{new}_{i,s}$ e $d_{i,s}$ siano il più simili possibili ($d^{new}_{i,s}$ = distanza tra y_i e y_s)
- Si definisce la funzione errore (errore di rappresentazione) in questo modo

$$E_{SAM} = {}^{1}\!\Big/\!_{C} \sum_{i,s} \frac{\left(d_{i,s} - d^{new}_{i,s}\right)^2}{d_{i,s}}$$

Multi dimensional Scaling (MDS)

- Algoritmo: (discesa lungo il gradiente per trovare gli y_i che minimizzano E_{SAM})

  PASSO 0: inizializzazione casuale di **Y** = [y^0_1.. y^0_N]

  Ripetere fino a convergenza

  PASSO 1: calcolare l'errore $E_{SAM}^{(t)}$ al tempo t

  PASSO 2: Aggiornare i vettori **Y** = [y_1..y_N] secondo la formula

$$y_{ij}^{t+1} = y_{ij}^{t} - \eta \Delta_{ij}^{t}$$

$$\Delta_{ij}^{t} \propto \frac{\partial E_{SAM}^{t}}{\partial y_{ij}^{t}}$$

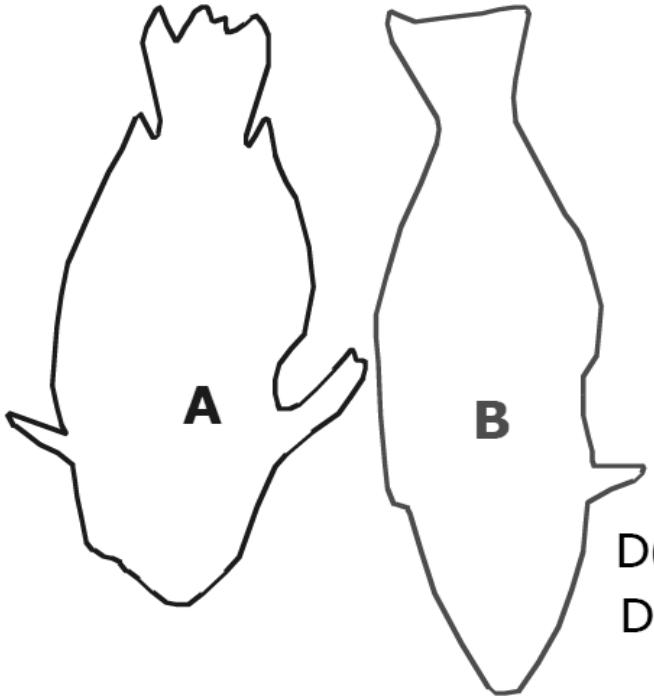
η learning rate

Problematiche

- Non è detto che si riesca a trovare uno spazio vettoriale in grado di preservare le distanze tra gli oggetti del problema
- Ci sono molte misure di dissimilarità che hanno senso da un punto di vista dell'applicazione ma che non sono “buone” da un punto di vista matematico (non sono metriche)

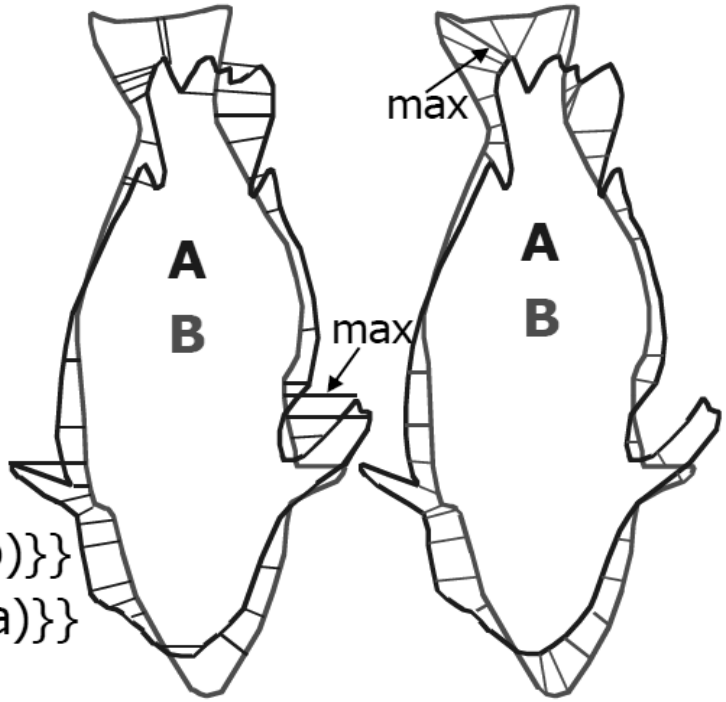
1. Positivity: $d_{ij} \geq 0$
2. Reflexivity: $d_{ii} = 0$
3. Definiteness: $d_{ij} = 0$ objects i and j are identical
4. Symmetry: $d_{ij} = d_{ji}$
5. Triangle inequality: $d_{ij} < d_{ik} + d_{kj}$

Problematiche



Dist(**A**,**B**):
a $\in$ **A**, points of **A**
b $\in$ **B**, points of **B**
d(a,b): Euclidean distance

D(**A**,**B**) = max_a{min_b{d(a,b)}}
D(**B**,**A**) = max_b{min_a{d(b,a)}}

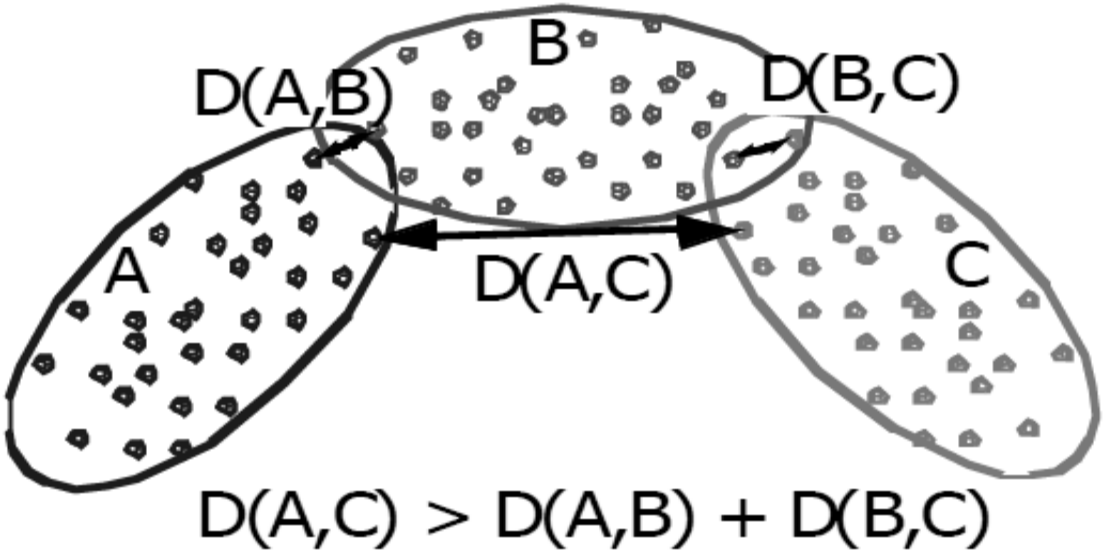


Esempio: distanza di Haussdorf per confrontare shapes

Molto utilizzata, funziona bene, ma... è una misura non simmetrica

Come si fa a creare uno spazio vettoriale dove la distanza tra A e B è diversa dalla distanza tra B e A???

Problematiche



Esempio: misura di distanza tra clusters (insiemi) – Single Linkage

Uno dei più famosi metodi di clustering si basa su questa distanza, ma.... questa misura non soddisfa la disuguaglianza triangolare!
(Non vale che D(A,C) <= D(A,B)+ D(B,C))

Problematiche

Caso ESTREMO:

D(Libro, tavolo) = 0
D(tazza, tavolo) = 0
D(Libro, tazza) > 0!

Anche in questo caso la distanza ha perfettamente senso!

Trovare metodi per gestire queste misure di dissimilarità altamente informative ma non metriche è un problema ancora aperto nella Pattern Recognition

Considerazioni finali sulla rappresentazione

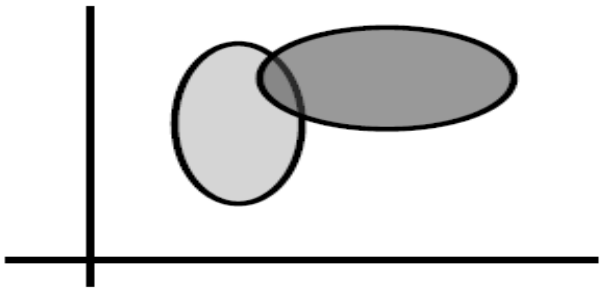
- La scelta della rappresentazione è cruciale, e rappresenta il cuore della Pattern Recognition
  - Goal della PR: dato un problema reale, l'obiettivo è trovare una rappresentazione che permetta di “generalizzare bene”

- Quali sono le caratteristiche di una rappresentazione adeguata?

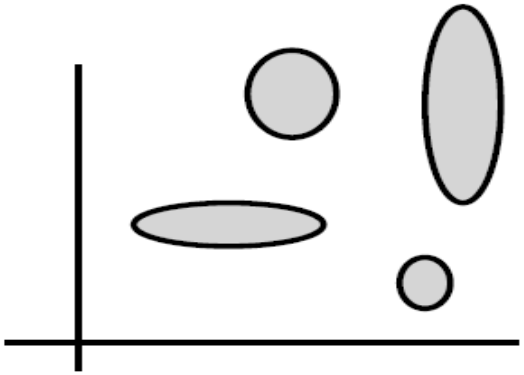
Considerazioni finali sulla rappresentazione

Buona rappresentazione:

- *Classe specifica*: classi differenti devono essere rappresentate in posizioni differenti nelllo spazio della rappresentazione

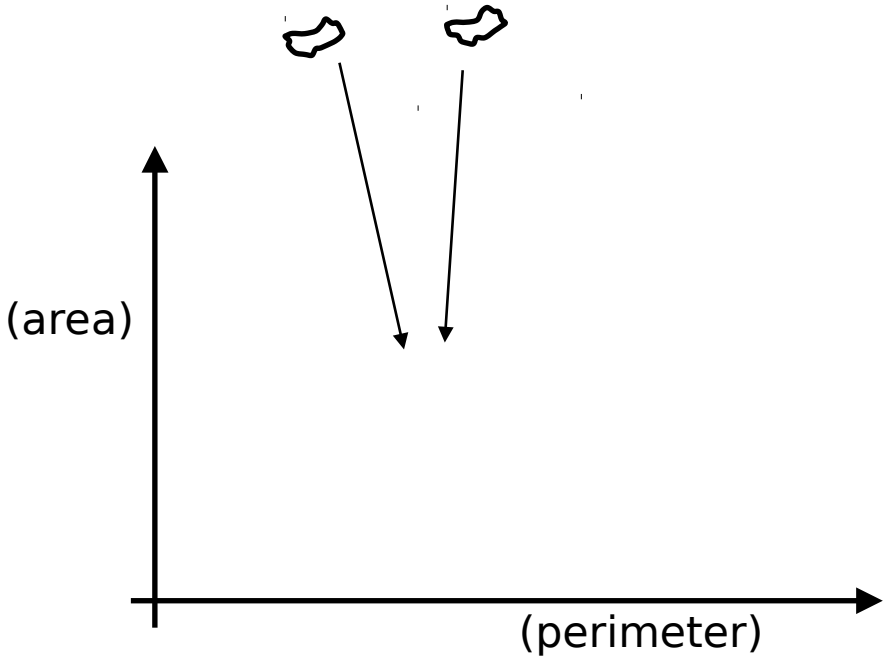


- *Compattezza*: ogni classe deve essere rappresentata in una zona piccola dello spazio



Considerazioni finali sulla rappresentazione

Compactness: "Representations of real world similar objects are close"

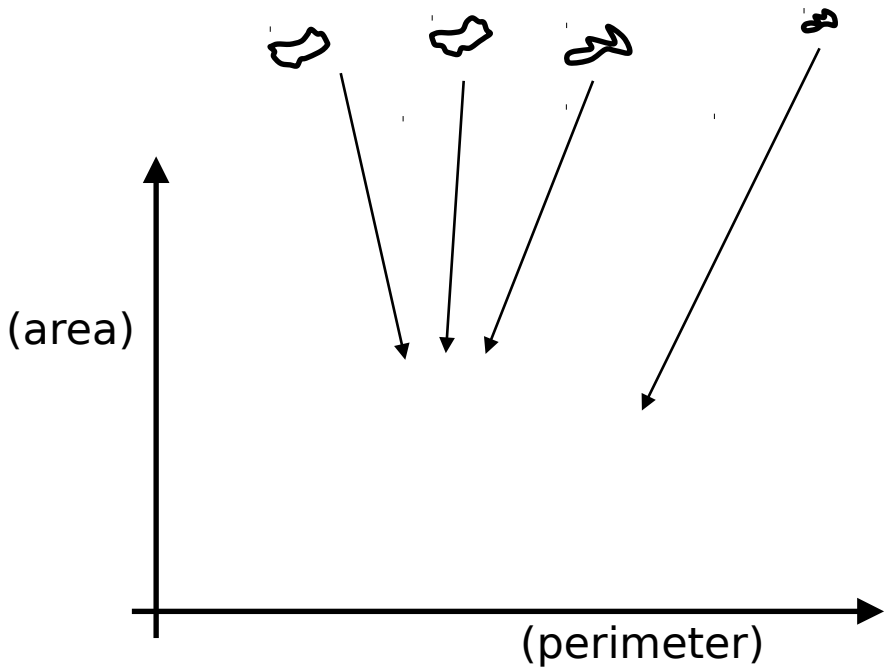


IPOTESI DI COMPATTEZZA: *Se due oggetti del problema sono simili allora hanno una rappresentazione simile (sono vicini nello spazio della rappresentazione)!*

"Nessuna possibilità di generalizzare se la rappresentazione viola questa regola"

(A.G. Arkedev and E.M. Braverman, *Computers and Pattern Recognition*, 1966.)

Considerazioni finali sulla rappresentazione

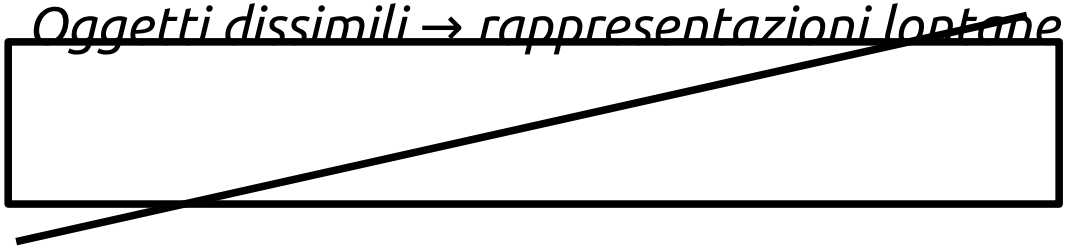


Attenzione: L'ipotesi di compattezza dice che:

Oggetti simili → rappresentazioni vicine

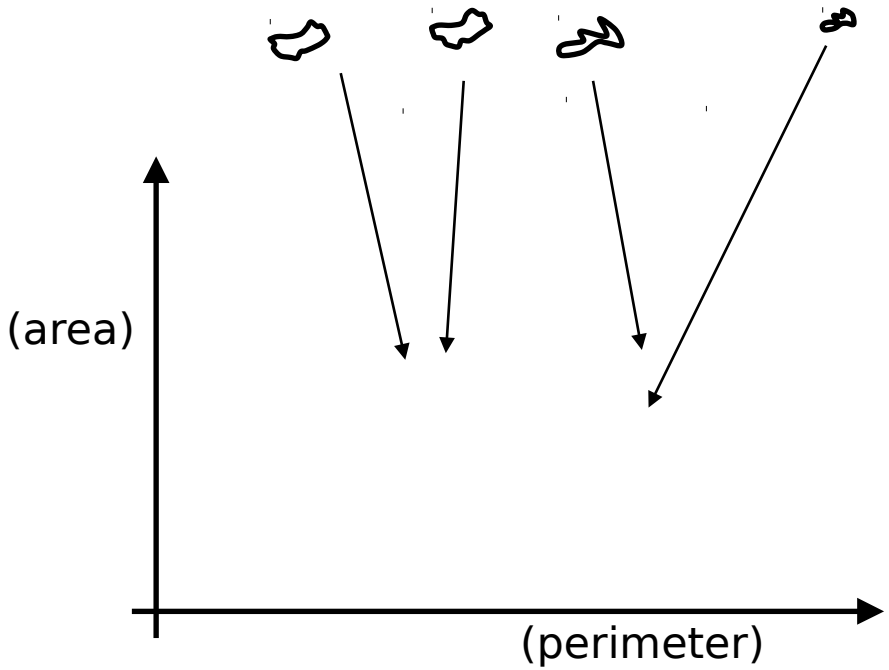
Questo in generale non è sufficiente per avere un classificatore perfetto, perché non è detto che valga anche il contrario

Oggetti dissimili → rappresentazioni lontane



Considerazioni finali sulla rappresentazione

Se vale anche il contrario allora si parla di "TRUE REPRESENTATIONS"



Oggetti simili → rappresentazioni vicine
Oggetti dissimili → rappresentazioni lontane

La rappresentazione dei dati

1. Campionamento (acquisizione dati)

2. Rappresentazione: estrazione delle features e costruzione del pattern

3. Preprocessing: scaling, riduzione del rumore, riduzione della dimensionalità

Fase 4: preprocessing

Preprocessing

- Concetto di scala
- Data standardization
- Data transformation
  - Riduzione della dimensionalità
- Riduzione del rumore

(ci concentriamo sul caso generico di spazi vettoriali)

Scala

⇨ Definizione di scala: significatività relativa dei numeri

ESEMPIO: si considerino due numeri 10, 12

⇨ sono molto simili in una scala [0-100]

⇨ sono molto diversi in una scala [10-13]

⇨ Riconoscere la scala di un problema è fondamentale:

- ⇨ nel calcolare la relazione tra due pattern
- ⇨ nell'interpretare i risultati (in particolare del clustering)

Scala

Tipi di scala:

- qualitativa: le misure non hanno significato
  - I numeri sono usati come nomi,
    - Esempio: (sì/no) può essere codificata come (1,0), (0,1) o (50,100).
  - oppure semplicemente conta l'ordine tra di essi (1,2,3 sono come 10,20,100)
    - Esempio lista di accesso all'università
- quantitativa: le misure hanno significato
  - I numeri hanno un valore (c'è un'unità di misura che dà il significato al valore)
  - Esempio: voto 9 (in decimi è buono, in trentesimi no!)

Problema

- Problema: a volte le variabili che descrivono un oggetto non sono nella stessa scala
- Ci sono metodi che soffrono se le features sono a scala diversa
- (Alla lavagna) esempio: calcolo della distanza euclidea tra due punti
- Soluzioni:
  - data standardization
  - data transformation

Data standardization

- La standardizzazione dei dati produce dati “senza dimensionalità”
  - tutta la conoscenza su scala e locazione dei dati viene persa dopo la standardizzazione
  - creazione di nuovi dati “in formato standard” – confrontabili

- E’ necessario standardizzare i dati!
  - esempio distanza euclidea visto in precedenza

Approcci di data standardization

- NOTA: la scelta dell'approccio da utilizzare dipende dal data set e dal campo di applicazione

- Alla lavagna:
  - notazione
  - formulazione generale
  - esempi di standardizzazione
  - intuizioni sugli effetti

Approcci di data standardization

- Ci sono due modi per standardizzare i dati
  - approcci globali standardizzano l’intero data set
  - approcci intra-gruppo standardizzano ogni gruppo (ATTENZIONE a non perdere tutta l'informazione!)
    - problema nel caso del clustering: i cluster non sono noti
    - possibile soluzione: standardizzazione iterativa (prima si calcolano i cluster, poi si standardizza, poi si ricalcolano i cluster e così via)

Data transformation

- In qualche modo legato alla standardizzazione dei dati
  - serve per migliorare la rappresentazione

- Data standardization: le operazioni sono implementate dimensione per dimensione

- Data transformation: le operazioni agiscono su tutte le dimensioni contemporaneamente (tipicamente in modo lineare)

Data transformation

In genere effettuata con i seguenti obiettivi:

1. ridurre la dimensionalità dello spazio delle features
   - ⇨ per visualizzare il dataset
   - ⇨ per ridurre il carico computazionale delle tecniche applicate
   - ⇨ per alleviare il problema della “curse of dimensionality”
   - ⇨ per eliminare la ridondanza di alcune direzioni del dataset

2. mettere in evidenza particolari strutture o migliorare le capacità discriminative dello spazio

Data transformation

- Approccio classico: trasformazione lineare dello spazio delle features

$$Y = A^{'} X$$

- Ogni nuova feature è una combinazione lineare di tutte le feature precedenti
  - Vediamo un esempio per un solo vettore x (una colonna della matrice **X**)

Esempio

A' x = y

A: matrice di trasformazione dei dati

Matrice 7x2

x: (dato) punto in uno spazio a dimensione 7

Matrice 7x1

y: (dato trasformato) punto in uno spazio a dimensione 2

Matrice 2x1

Data transformation

- Quindi: si vuole ridurre la dimensionalità dello spazio mantenendo la maggior quantità di informazione possibile

- “Informazione”: concetto che assume significati diversi a seconda della tecnologia utilizzata

Diversi approcci

- Approcci non supervisionati:
  - si utilizzano solo i dati
  - Esempio: Principal Component Analysis
- Approcci supervisionati:
  - si utilizzano altre informazioni (ad esempio le etichette)
  - Esempio: trasformata di Fisher
- La seconda classe di approcci si può utilizzare solo in caso di problema supervisionato (classificazione)
  - L’idea è che lo spazio viene “ridotto” tenendo conto del task finale di classificazione

Principal Component Analysis

- Approccio lineare non supervisionato alla riduzione della dimensionalità
- Obiettivo: mantenere la maggior aderenza ai dati originali
  - Minimizza lo scarto quadratico medio tra i dati originali e quelli ricostruiti
  - estrae le direzioni di massima varianza dei dati

Principal Component Analysis

- Cosa fa?
  - trasformazione lineare delle variabili che proietta i dati in uno spazio tale per cui
    - la prima direzione è quella di massima varianza
    - la seconda direzione è quella di massima varianza che sia però ortogonale alla prima
    - si continua così fino alla fine

- Come si realizza?
  - Esiste un algoritmo che ci permette di trovare direttamente la matrice di trasformazione A che realizza la PCA
  - Si basa sugli autovalori e autovettori della matrice di covarianza dei dati

Principal Component Analysis

(Algoritmo alla lavagna)

Principal Component Analysis

- Vantaggi:
  - migliore tecnica lineare di riduzione della dimensionalità di un insieme di dati
    - migliore in senso di “errore quadratico medio”
  - i parametri del modello possono essere ricavati direttamente dai dati
  - La proiezione nello spazio è un'operazione molto veloce (moltiplicazione di matrici)

Principal Component Analysis

- Svantaggi:
  - alto costo computazionale per il calcolo dei parametri del modello (soprattutto in caso di dimensionalità elevata)
  - non è chiaro come questa tecnica possa gestire il caso di dati incompleti
  - PCA non tiene conto della densità di probabilità dello spazio considerato (viene considerata solo la vicinanza del vettore trasformato al vettore originale)
  - non è detto in tutti i casi che le direzioni a varianza maggiore siano le direzioni ottimali

Principal Component Analysis

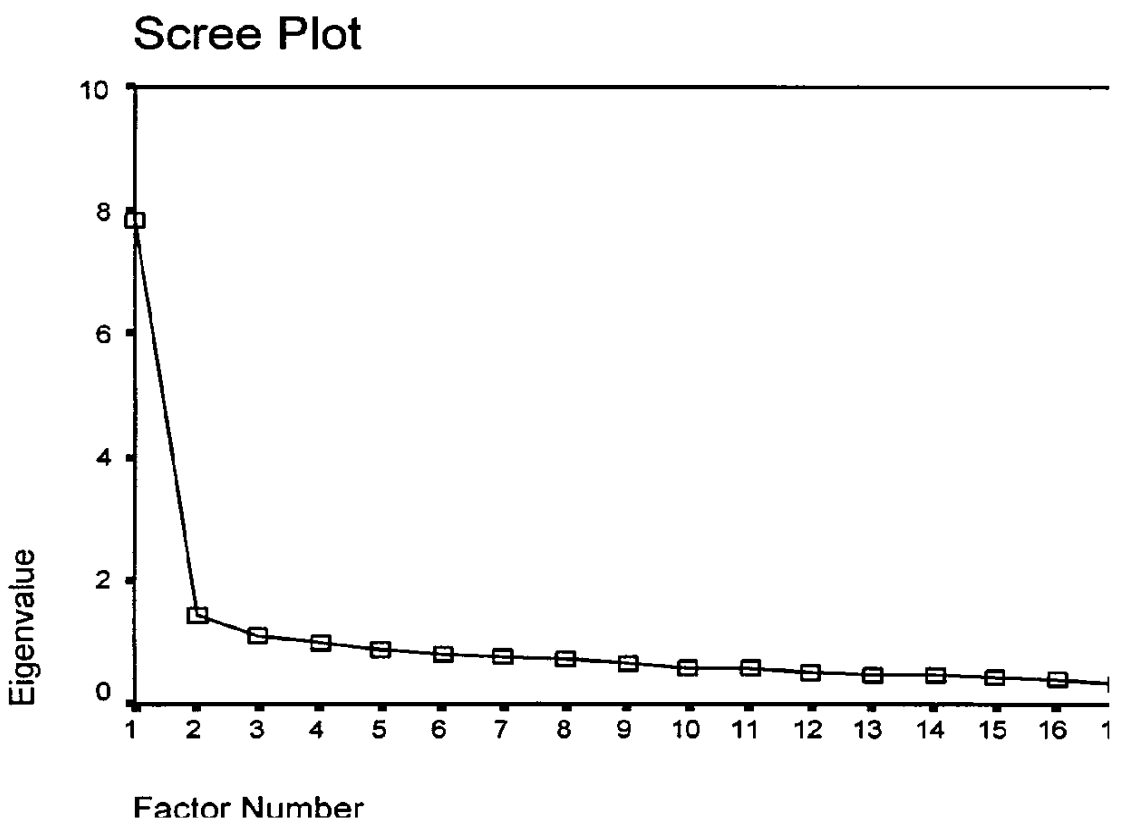
- Problema: come calcolare il numero di componenti principali ottimali?
- Soluzione 1. scegliere il numero di componenti che arrivano a coprire una certa percentuale di varianza (esempio 95%)
  - Come si misura? concetto di “importanza” di una componente principale

$$\text{imp}(i) = \frac{\lambda_i}{\sum_{i=1}^{d} \lambda_i}$$

$$\text{varianza}(1,..,\text{L}) = \sum_{i=1}^{L} \text{imp}(i) = \frac{\sum_{i=1}^{L} \lambda_i}{\sum_{i=1}^{d} \lambda_i}$$

Principal Component Analysis

- Soluzione 2: Cattell Scree test
  - plottare gli autovalori
  - trovare dove la decrescita degli autovalori non è più così marcata



Metodi supervisionati

- IDEA: utilizzo l'informazione del task finale
  - non sempre l'informazione "non supervisionata" estratta corrisponde alla migliore informazione per risolvere il problema
  - ESEMPIO: PCA vs Fisher

- Linear Discriminant Analysis (Fisher)
  - tecnica classica di riduzione della dimensionalità che mira a massimizzare la separabilità tra le classi nello spazio risultante
  - spesso utilizzata anche come classificatore lineare (dettagli in seguito)
  - anche chiamata:
    - Discriminant Analysis
    - Fisher Linear Analysis

Linear Discriminant Analysis

Occorre definire un criterio per misurare la “separabilità tra le classi”

- Diverse opzioni:
  - Esempio: la distanza tra le medie nello spazio risultante

    (IDEA: più sono separate le classi nello spazio dopo la trasformazione, più separate sono le classi)
  - Criterio non ottimale: non tiene conto della dispersione dei punti all'interno della classe

- Esistono criteri migliori: il criterio di Fisher!

Linear Discriminant Analysis

Criterio di Fisher

$$J(A,X) = \text{traccia} \{ S_{Wy}^{-1} \cdot S_{By} \}$$

Dove:

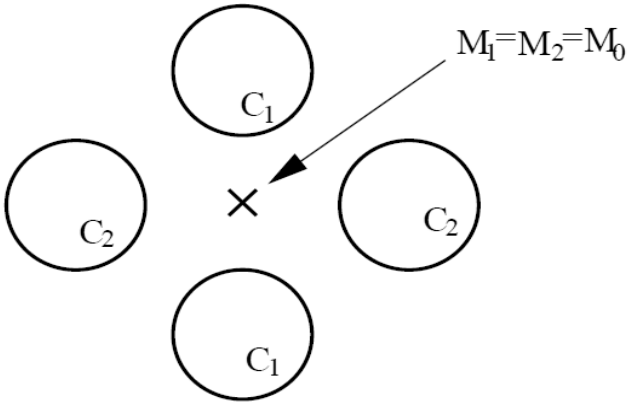
- S_{Wy}: “Within Class Covariance” nello spazio finale: dispersione interna alle classi
- S_{By}: “Between Class Covariance” nello spazio finale: dispersione tra le diverse classi

Linear Discriminant Analysis

- L'idea è di:
  - Massimizzare la Covarianza tra le classi (classi ben separate tra di loro)
  - Minimizzare la Covarianza interna alle classi (classi ben compatte)
- Come si trova la A che massimizza $\mathbf{J(A,X)}$?
  - Con un procedimento simile a quello usato per la PCA
  - A è formata dagli m autovettori corrispondenti agli m autovalori più grandi della matrice $\mathbf{S_{wx}^{-1} \cdot S_{bx}}$
  - $\mathbf{S_{wx}}$ $\mathbf{S_{bx}}$ hanno lo stesso significato di $\mathbf{S_{wy}}$ $\mathbf{S_{by}}$, (Within-class e Between-class) ma sono calcolati nello spazio originale X

Linear Discriminant Analysis

- Vantaggi: massimizza la separabilità tra le classi
- Svantaggi:
  - Principale: dato un problema a C classi, la massima dimensionalità dello spazio risultante è C-1
    - se abbiamo un problema binario allora possiamo proiettare i dati in uno spazio monodimensionale (molto restrittivo!)
  - Secondario: il criterio di Fisher non funziona se le classi sono multimodali e condividono la stessa media



Feature Selection

- Approccio alternativo alla riduzione della dimensionalità
  - rid. dimensionalità: considera tutte le feature e le trasforma
  - feature selection: sceglie solo alcune feature (in base ad un criterio di ottimalità)
  - Vantaggio: spesso alcune features sono irrilevanti / dannose
  - Svantaggio: computazionalmente oneroso

Feature Selection

- Problemi da risolvere
  - scegliere il criterio di ottimalità: informazione (come si misura?), varianza, capacità di classificazione (solo per problemi supervisionati)
  - come trovare il sottoinsieme ottimale senza provarli tutti (troppo oneroso computazionalmente)

- Esempio: Sequential Forward Feature Selection

Feature Selection

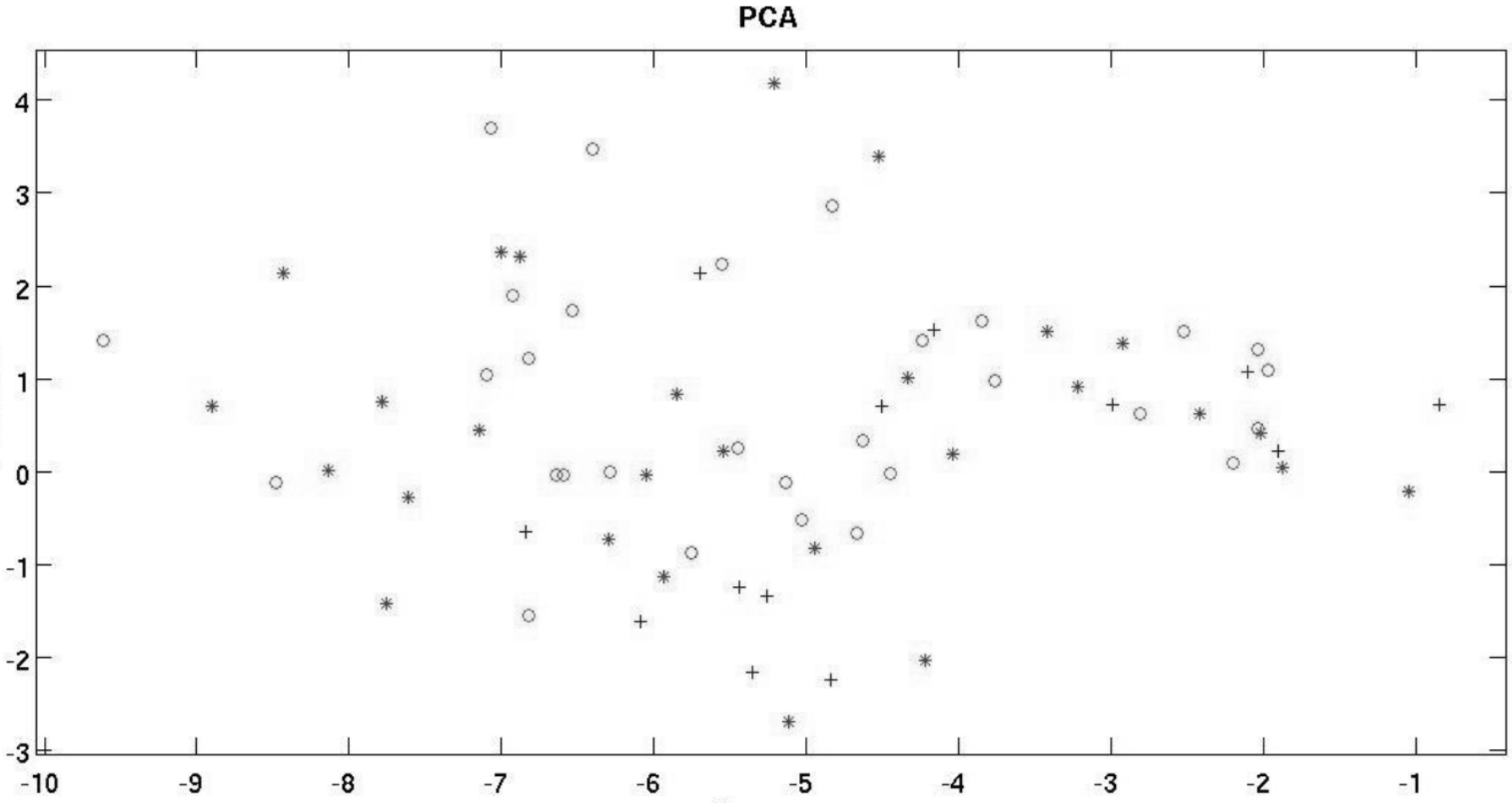
- Forward Sequential Feature Selection
  - Algoritmo greedy per trovare l'insieme ottimale di features
- Schema:
  - si valuta il criterio per tutte le features singolarmente
  - si sceglie la feature che massimizza il criterio (chiamata f1)
  - si valuta il criterio per tutte le coppie (f1,fx) – cioè tenendo fissata f1
  - si sceglie la coppia che massimizza il criterio (la coppia (f1,f2))
  - si valuta il criterio per tutte le terne (f1,f2,fx) – cioè tenendo fissate f1 e f2
  - ...
- Algoritmo greedy – ad ogni istante la scelta migliore
  - computazionalmente efficiente
  - sub ottimale

Esempio: neuropatologie

- Obiettivo: analisi di dati derivanti da pazienti affetti da Miopatie infiammatorie (malattie autoimmuni) per classificare sani/malati (o le diverse tipologie)
  - gli anticorpi del sistema immunitario riconoscono particolari proteine (normalmente presenti) del siero di un paziente, come agenti estranei (antigeni), scatenando una risposta infiammatoria causa della malattia.
  - Analisi di laboratorio permettono di individuare le proteine muscolari colpite dalla risposta immunitaria.
  - Ogni paziente è caratterizzato da un insieme di proteine (profilo sierologico), alcune delle quali vengono riconosciute e colpite dalla risposta immunitaria, altre no.
- Dal nostro punto di vista: una proteina è una feature!

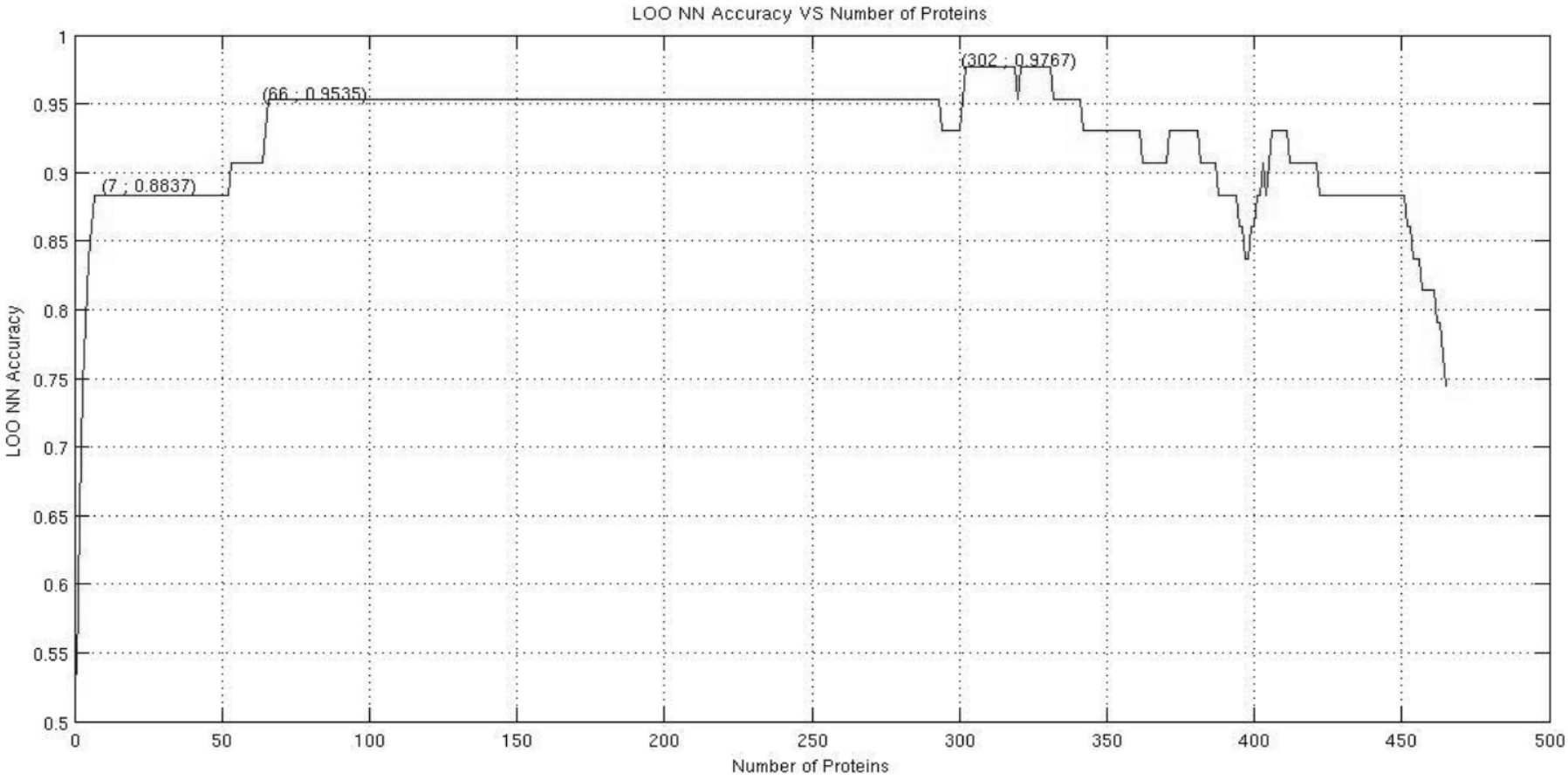
Esempio: neuropatologie

⇨ Usando tutte le features e applicando la PCA



Esempio: neuropatologie

⇨ feature selection



Visualizzazione dei dati

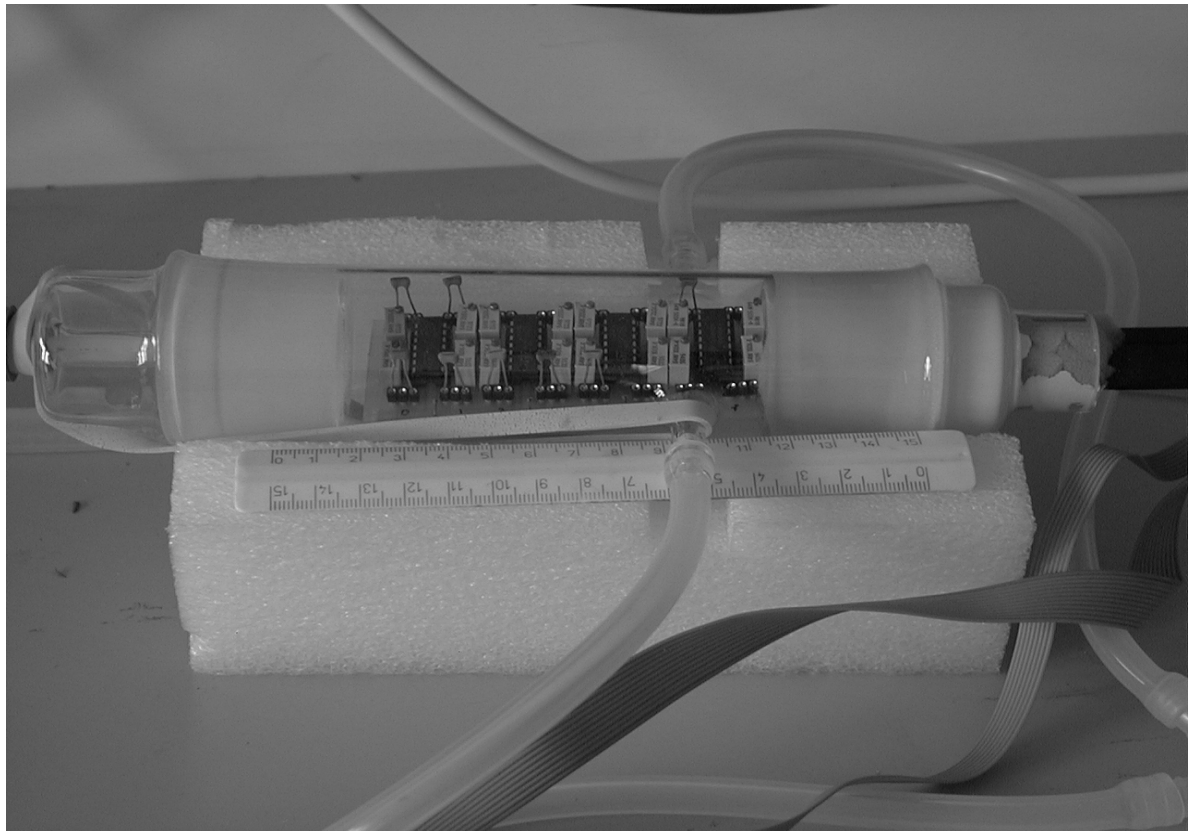
- Studio di metodologie per visualizzare dati/informazioni in modo grafico
- Fondamentale per:
  - validazione/interpretazione dei risultati
- Il nostro punto di vista:
  - rappresentare dati in uno spazio vettoriale a dimensione 2/3 per poter vedere la relazione tra i dati

- Approccio classico: ridurre la dimensionalità con le tecniche viste precedentemente, tipo PCA

Esempio

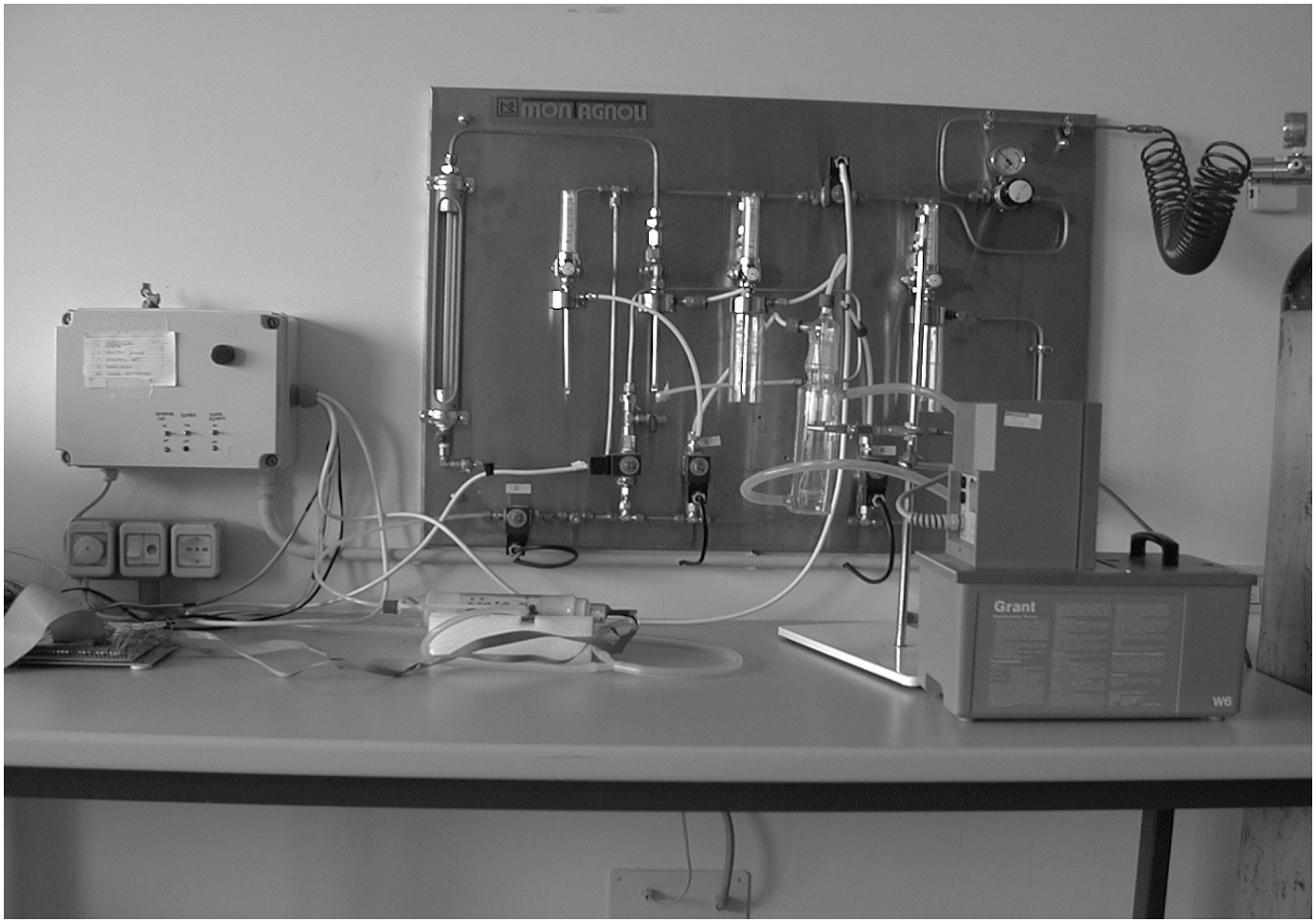
Riconoscimento di odori

⇨ Array di sensori chimici, ognuno sensibile a composti diversi



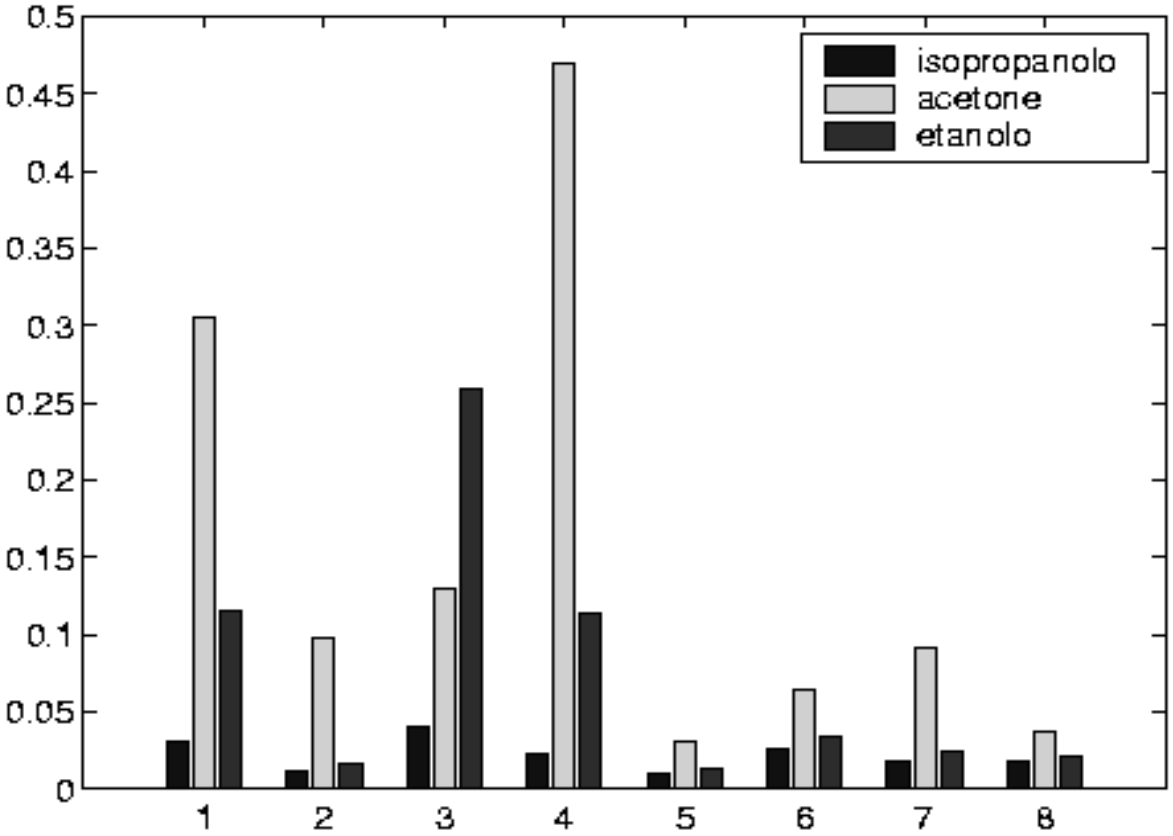
Esempio

- Dato un composto da analizzare, viene fatto passare sopra il “naso elettronico”



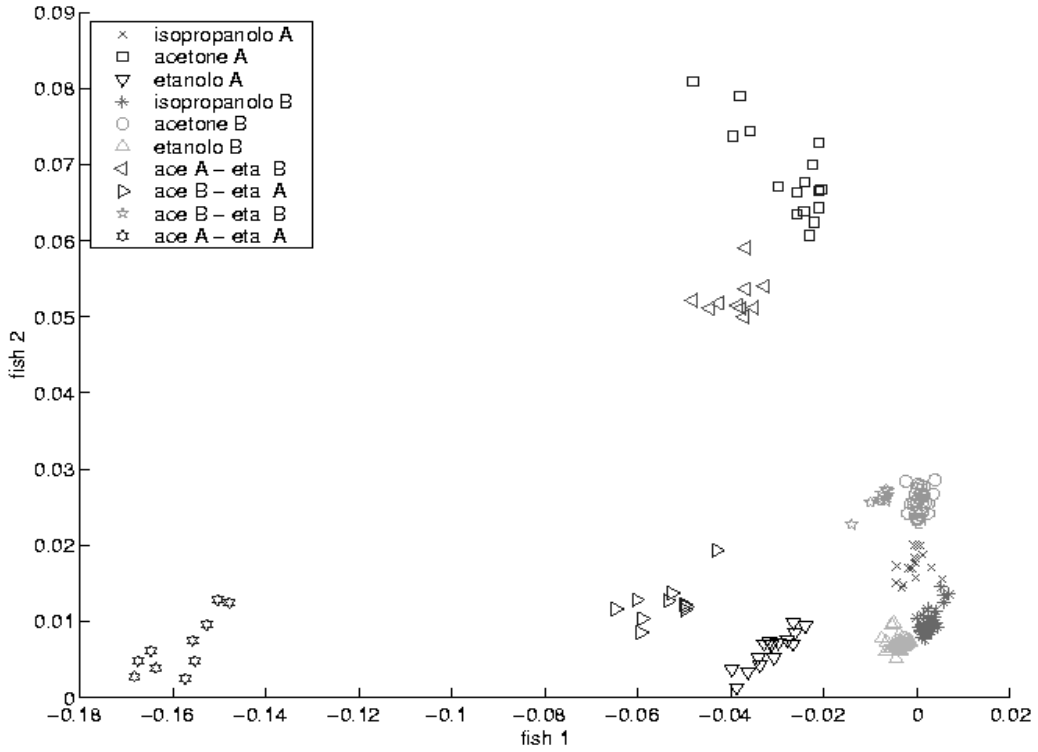
Esempio

⇨ Il risultato è un punto in uno spazio 8-dimensionale



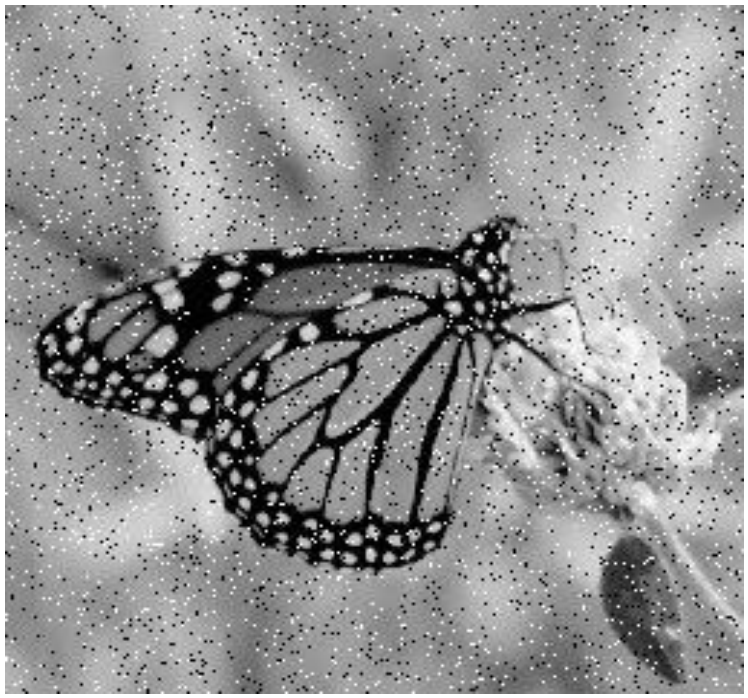
Esempio

⇨ Per visualizzare tutti gli esperimenti: PCA + plot



Riduzione del rumore

- Rumore: informazione irrilevante (o dannosa) nei dati
- Per rimuovere il rumore vengono utilizzate tecniche di filtraggio (molto utilizzate in signal/image processing)

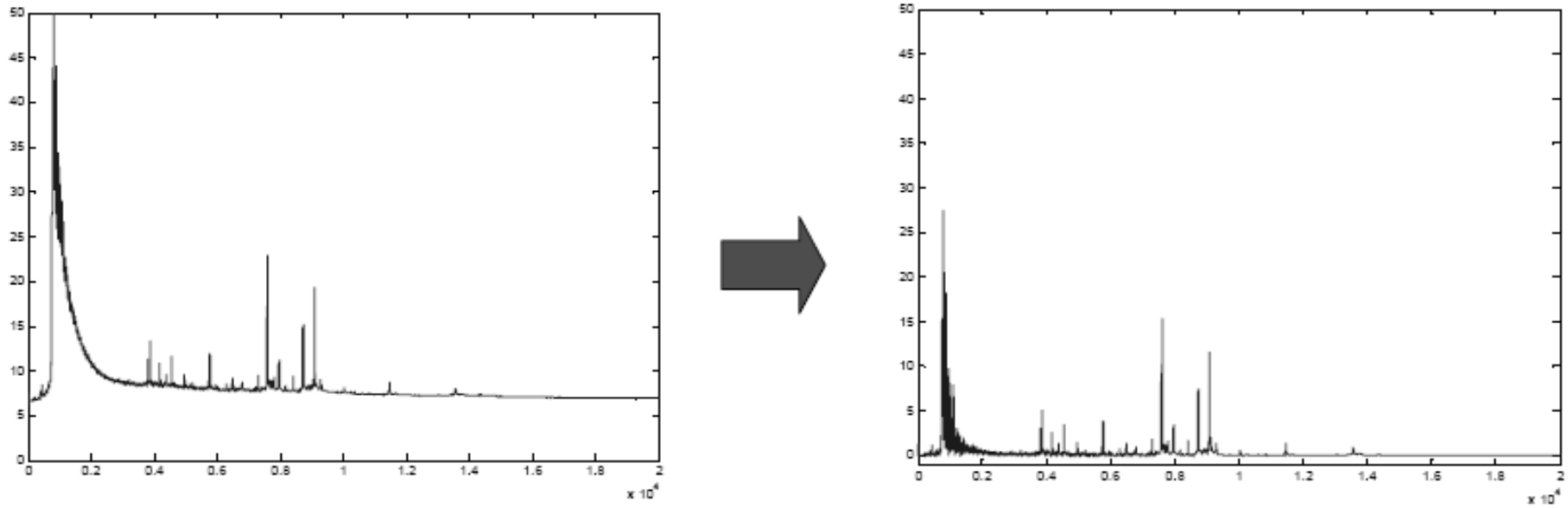


rumore sale e pepe

immagine “ripulita”

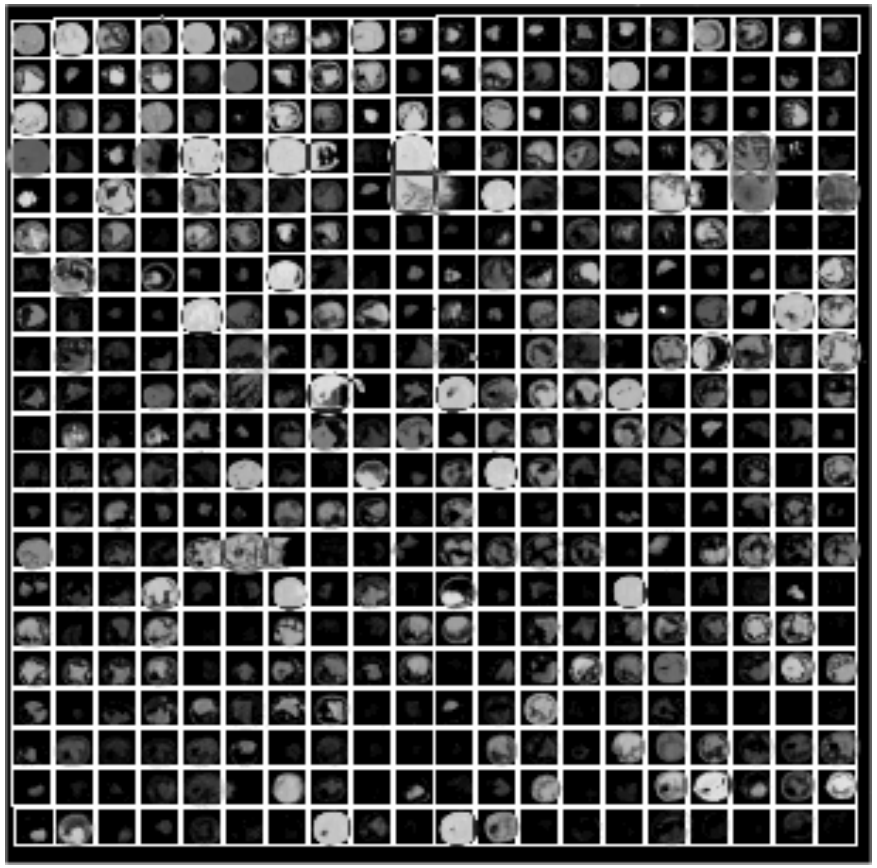
Riduzione del rumore

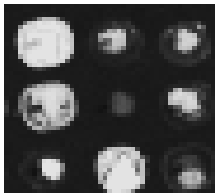
- Mass spectrometry: esiste un bias di intensità sistematico per il quale il profilo osservato differisce da zero



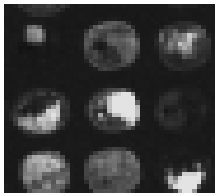
Riduzione del rumore

⇨ Microarray: errori nelle immagini degli spot

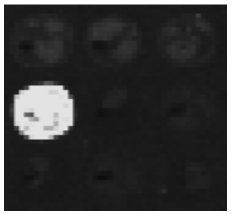




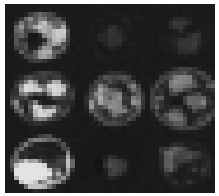
dimensione



rotondità



intensità



distribuzione
dei pixel

Riduzione del rumore

- NOTA: E’ necessario eliminare il rumore preservando l’informazione

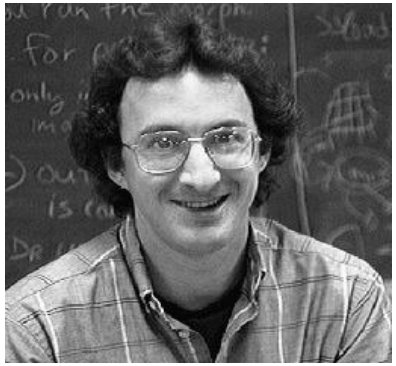
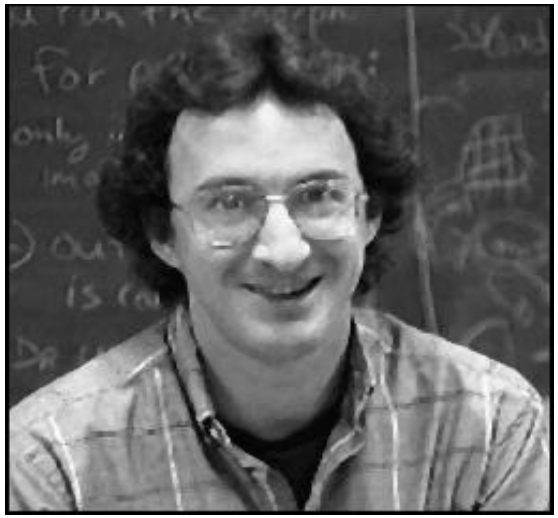
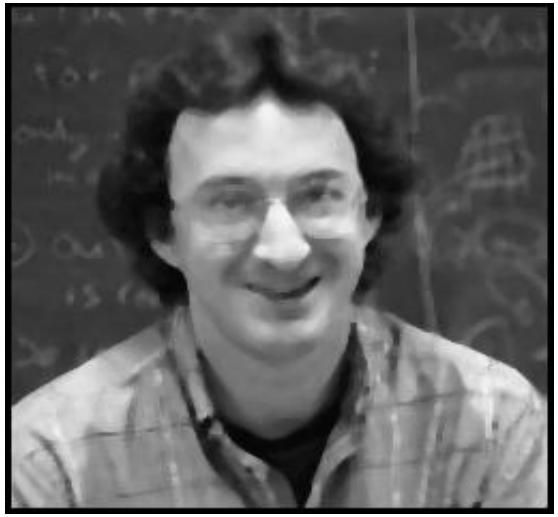


immagine originale



filtro mediano 3x3



filtro mediano 5x5

filtro mediano 7x7