Computazione per l'interazione naturale: clustering e riduzione di dimensionalità

Corso di Interazione uomo-macchina II

Prof. Giuseppe Boccignone

Dipartimento di Informatica
Università di Milano

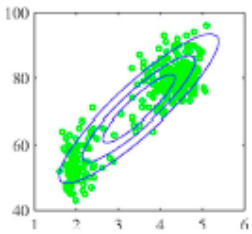
boccignone@di.unimi.it
http://boccignone.di.unimi.it/IUM2_2014.html

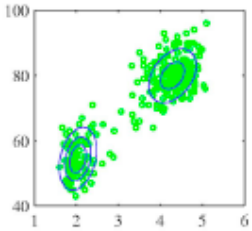
Apprendimento non supervisionato //Il clustering

	Supervised Learning	*Unsupervised Learning*
Discrete	classification or categorization Discrete Output y $\in \{1, \ldots, K\}$	clustering
Continuous	regression Continuous Output $y \in R$	dimensionality reduction

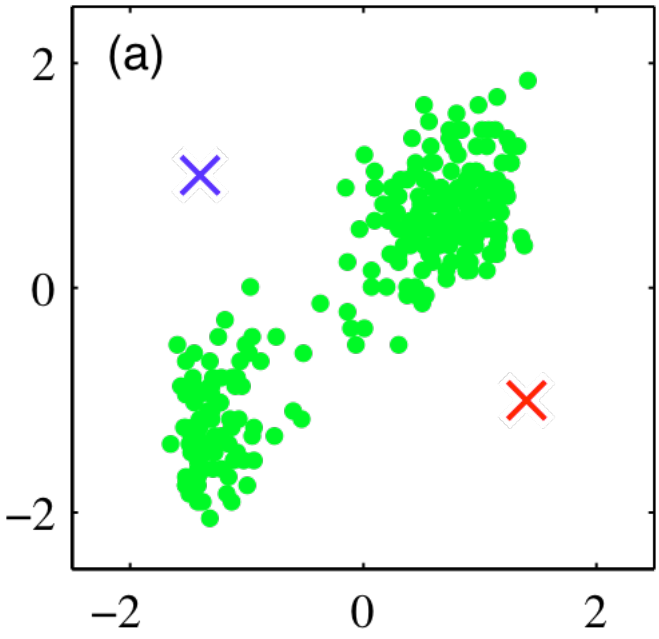
Il problema del clustering

- Trovare strutture coerenti nei dati
- Esempio: Old Faithful dataset
- Una singola Gaussiana è insufficiente
- Supponiamo di avere 2 clusters (agglomerati)
- Vogliamo assegnare ciascun punto ad uno dei due cluster
- Unmetodo di clustering semplice: k-means

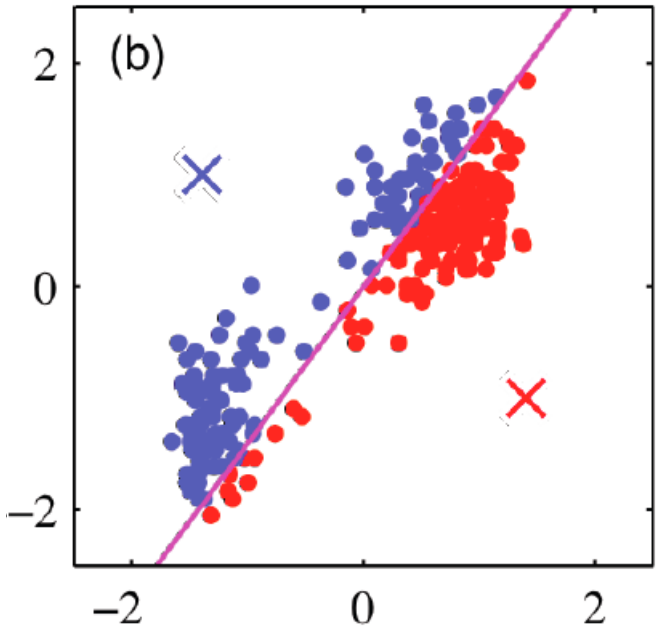




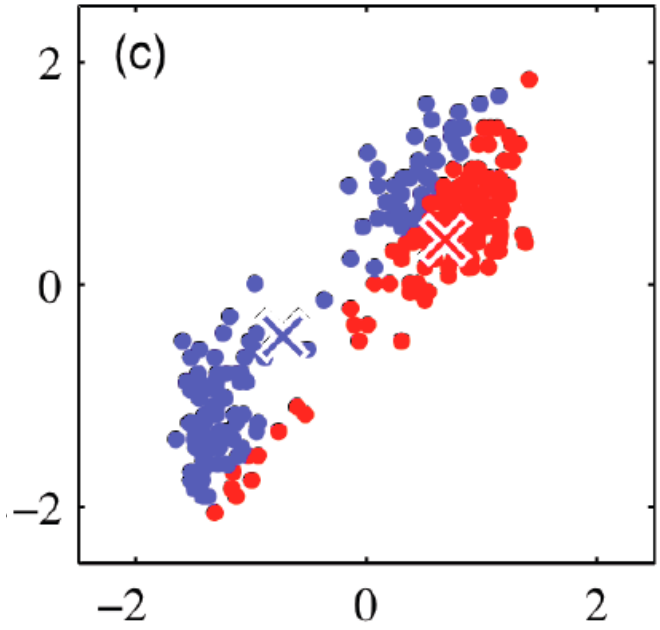
K-means clustering //Esempio



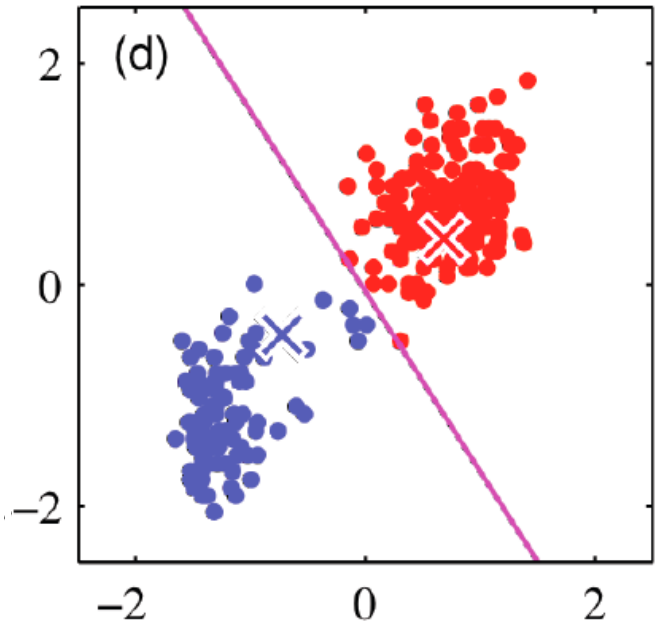
K-means clustering
//Esempio



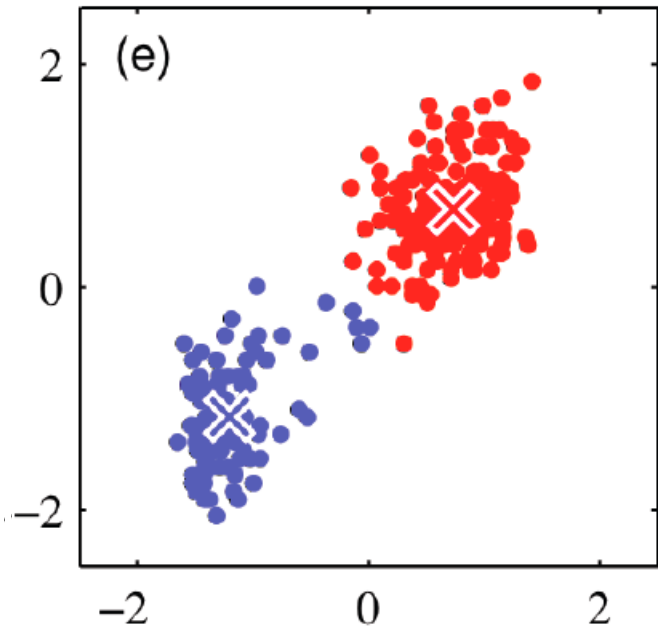
K-means clustering
//Esempio



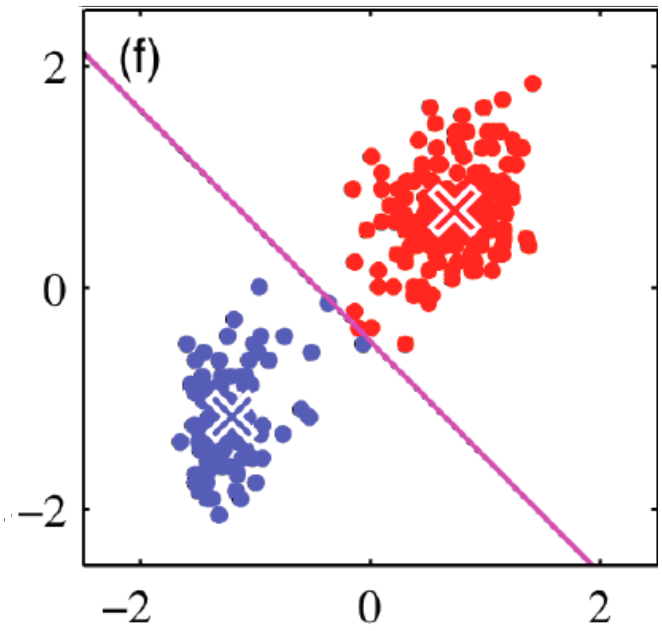
K-means clustering //Esempio



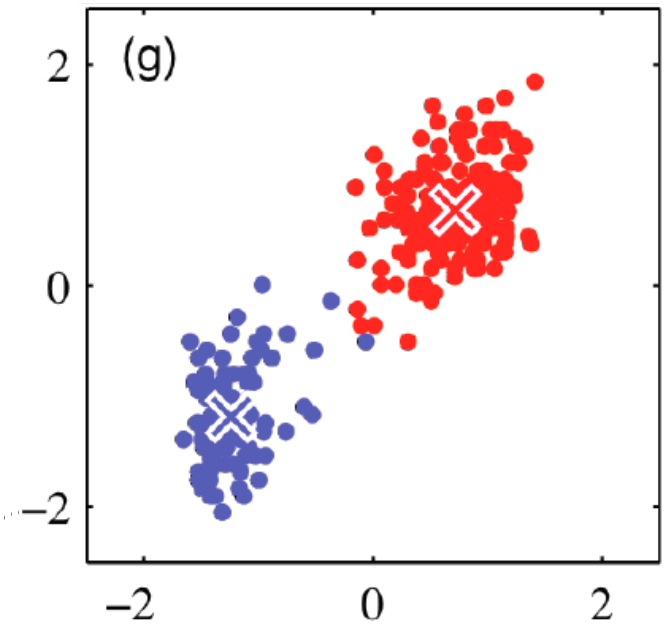
K-means clustering //Esempio



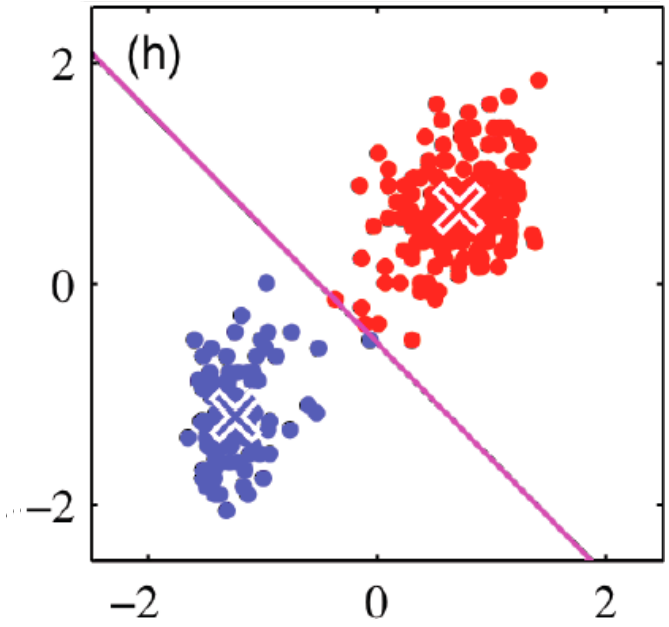
K-means clustering //Esempio



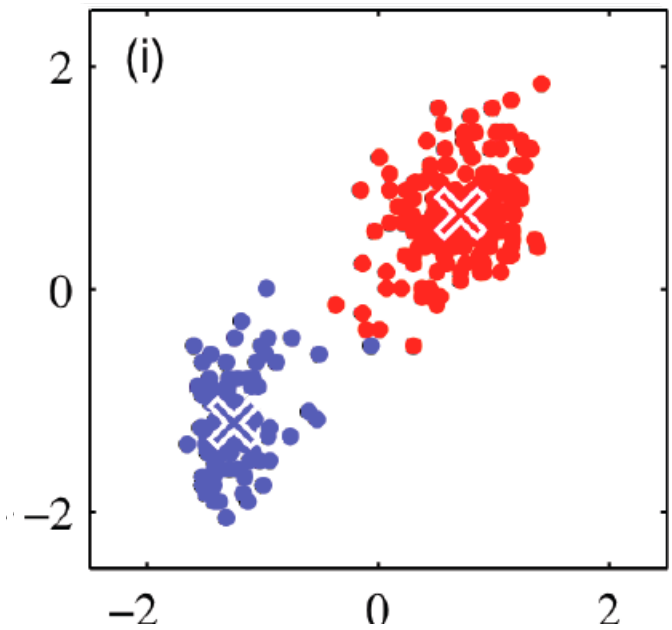
K-means clustering //Esempio



K-means clustering
//Esempio



K-means clustering
//Esempio



K-means clustering
//Algoritmo: ipotesi

- Dati (non etichettati) in uno spazio di features di dimensione D: $\mathbf{x}_n \in \mathbb{R}^D$
- Ipotizziamo K possibili clusters (è il nostro modello)
- Supponiamo che esista per ciascun punto n una variabile binaria che vale 1 se il punto n appartiene al cluster k

$$z_{kn} \in \{0, 1\}$$

- Questo tipo di variabile è detta indicatore (indicator variable), ed è in generale una variabile latente (non la conosciamo)

K-means clustering
//Algoritmo: misura di qualità del cluster

- Ci serve una misura di coerenza interna o compattezza per i punti di un cluster: distanza totale di tutti i punti dal baricentro del cluster

$$\sum_{\mathbf{x}_n \in \mathcal{C}_k} ||\mathbf{x}_n - \mathbf{m}_k||^2 = \sum_{n=1}^{N} z_{kn} ||\mathbf{x}_n - \mathbf{m}_k||^2$$

$$\mathbf{m}_k = \frac{1}{N_k} \sum_{\mathbf{x}_n \in \mathcal{C}_k} \mathbf{x}_n$$

- Numero di punti allocati nel cluster k: $N_k = \sum_{n=1}^{N} z_{kn}$
- Qualità complessiva del clustering: il criterio da ottimizzare

$$\mathcal{E}_K = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{kn} ||\mathbf{x}_n - \mathbf{m}_k||^2$$

K-means clustering //Algoritmo: ottimizzazione

- Due insiemi di parametri: $\mathbf{m}_k$, z_{kn}

- 1. Fissato $\mathbf{m}_k$ derivo $\mathcal{E}_K = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{kn}||\mathbf{x}_n - \mathbf{m}_k||^2$ rispetto a z_{kn} : il criterio si annulla quando
  - il punto è assegnato alla media più vicina
- 2. Fissato z_{kn} derivo $\mathcal{E}_K = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{kn}||\mathbf{x}_n - \mathbf{m}_k||^2$ rispetto a $\mathbf{m}_k$

$$\mathbf{m}_k = \frac{\sum_{n=1}^{N} z_{kn}\mathbf{x}_{kn}}{\sum_{n'=1}^{N} z_{kn'}}$$

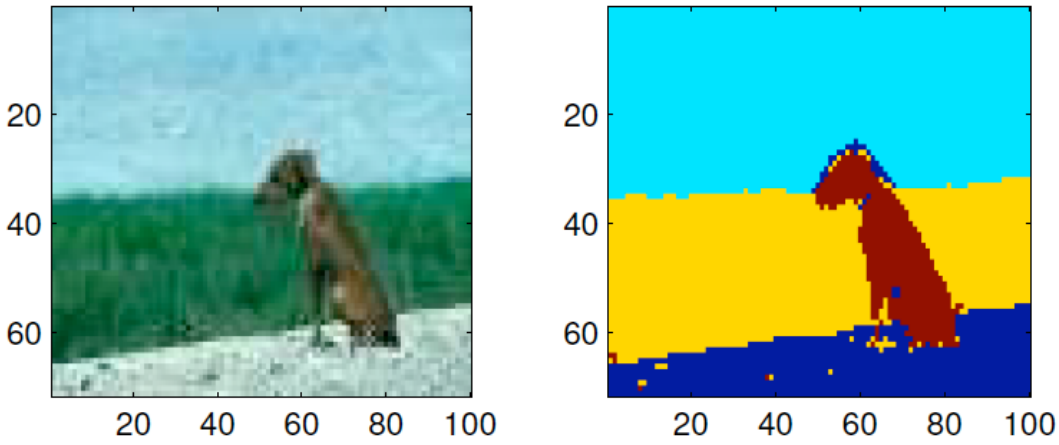
- Itero fino a convergenza

K-means clustering //Algoritmo

```
function [M,z,e] = kmeans(X,K,Max_Its)
%KMEANS Semplice implementazione del k-means clustering
% Input:
%  Data matrix X (N x D)
%  Numero di cluster K
%  Massimo numero di iterazioni Max_Its
% Output:
%   Matrice M (K x D) - le K medie
%   Vettore z (N x 1): z(n)=k indica il Cluster 1.. K a cui ciascun punto x_n e' stato assegnato
[N,D]=size(X);  %N - num punti, D dimensione dati
I=randperm(N);  %permutazione random  degli interi 1:N - usata
                %per settare in modo random le K medie iniziali
M=X(I(1:K),:);  %M matrice K x D dei mean values - selezionate in modo random
Mo = M;
for t=1:Max_Its
    %Crea la distance matrix  N x K: indica la distanza di ciascun punto X_n  dai K centri
    for k=1:K
        Dist(:,k) = sum((X - repmat(M(k,:),N,1)).^2,2);
    end
    %Minima distanza di K da ciascun punto.
    [i,z]=min(Dist,[],2);

    %Abbiamo gli assegnamenti ai  clusters: ricalcoliamo i centri dei
    %cluster
    for k=1:K
        if size(find(z==k))>0
            M(k,:) = mean(X(find(z==k),:));
        end
    end

    %Calcoliamo Z matrice N x K: indictor matrix - di elementi z_nk
    Z = zeros(N,K);
    for n=1:N    Z(n,z(n)) = 1; end
    %Calcola il valore corrente del  criterio di errore minimizzato da K-means
    e = sum(sum(Z.*Dist)./N);
    fprintf('%d Error = %f\n', t, e);
    Mo = M;
end
```

K-means clustering //Segmentazione dell'immagine come clustering

- K regioni/cluster, ad esempio di colore: ogni punto (pixel) è un vettore [R G B]
- Esempio: K=4



- Se si usa per codificare/comprimere: vector quantization

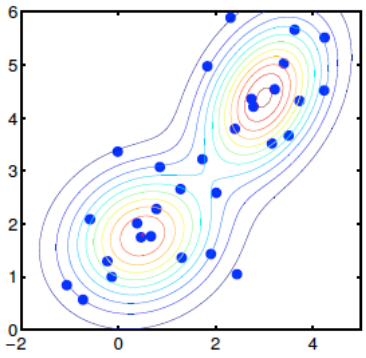
K-means clustering //Segmentazione dell'immagine come clustering

Cluster non Gaussiani

- Proviamo a generare una distribuzione di punti campionando da due Gaussiane 2D (due processi diversi) aventi i seguenti parametri

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 0.5 \\ 2.0 \end{bmatrix} \quad \mathbf{C}_1 = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$$

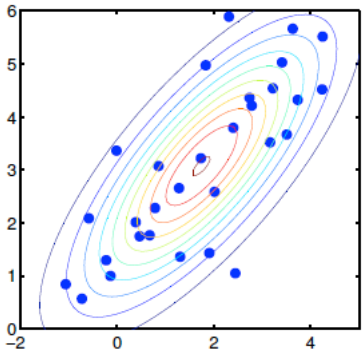
$$\boldsymbol{\mu}_2 = \begin{bmatrix} 3.0 \\ 4.0 \end{bmatrix} \quad \mathbf{C}_2 = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$$

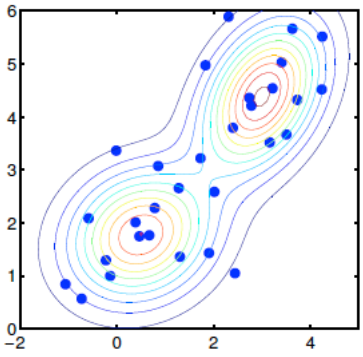


- La distribuzione risultante è non Gaussiana e non puo' essere fittata con un'unica Gaussiana (si può ma l'errore è elevato

Cluster non Gaussiani

- Proviamo a generare una distribuzione di punti campionando da due Gaussiane (due processi diversi) aventi i seguenti parametri

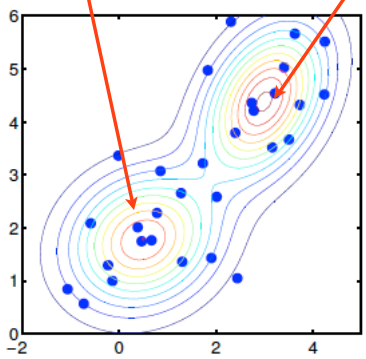




Misture di Gaussiane

- Possiamo rappresentare la distribuzione ottenuta come la mistura di due Gaussiane (mixture of Gaussians)

$$\begin{aligned} p(\mathbf{x}|\boldsymbol{\theta}) &= \pi p(\mathbf{x}|\boldsymbol{\theta}_1) + (1-\pi)p(\mathbf{x}|\boldsymbol{\theta}_2) \\ &= \pi \mathcal{N}_{\mathbf{x}}(\boldsymbol{\mu}_1, \mathbf{C}_1) + (1-\pi)\mathcal{N}_{\mathbf{x}}(\boldsymbol{\mu}_2, \mathbf{C}_2) \end{aligned}$$



$$\boldsymbol{\theta} = \{\pi, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$$

$$\boldsymbol{\theta}_1 = \{\boldsymbol{\mu}_1, \mathbf{C}_1\}$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 0.5 \\ 2.0 \end{bmatrix} \quad \mathbf{C}_1 = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} 3.0 \\ 4.0 \end{bmatrix} \quad \mathbf{C}_2 = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$$

$$\boldsymbol{\theta}_2 = \{\boldsymbol{\mu}_2, \mathbf{C}_2\}$$

π probabilità che x sia generato da $p(\mathbf{x}|\boldsymbol{\theta}_1)$

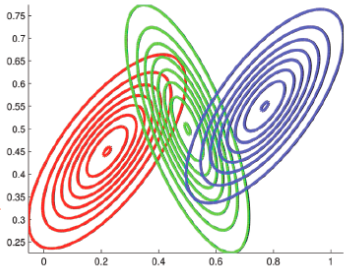
$1-\pi$ probabilità che x sia generato da $p(\mathbf{x}|\boldsymbol{\theta}_2)$

Misture di Gaussiane //caso generale

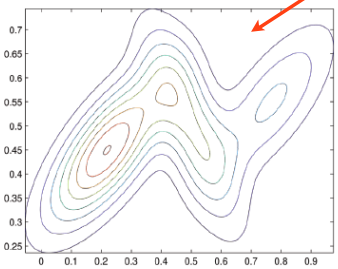
- Possiamo rappresentare una distribuzione non Gaussiana come mistura di M Gaussiane

$$\boldsymbol{\theta} = \{\pi_1 \cdots \pi_M, \boldsymbol{\theta}_1 \cdots \boldsymbol{\theta}_M\}$$

$$\sum_{m=1}^{M} \pi_m = 1$$



$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{m=1}^{M} \pi_m p(\mathbf{x}|\boldsymbol{\theta}_m)$$



Misture di Gaussiane //stima dei parametri (ML)

- Supponiamo di avere i dati

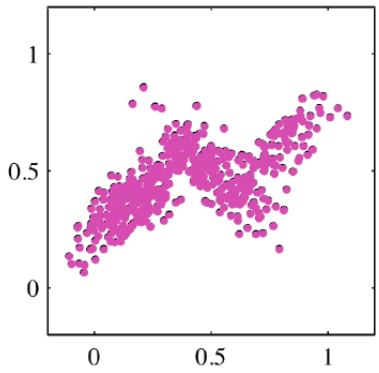
$$\mathcal{D} = \{\mathbf{x}_1 \cdots \mathbf{x}_N\}$$

- Assumiamo un modello MoG

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{m=1}^{M} \pi_m p(\mathbf{x}|\boldsymbol{\theta}_m)$$

- Vogliamo stimare

$$\boldsymbol{\theta} = \{\pi_1 \cdots \pi_M, \boldsymbol{\theta}_1 \cdots \boldsymbol{\theta}_M\}$$



Misture di Gaussiane //stima dei parametri (ML)

- Supponiamo di avere i dati

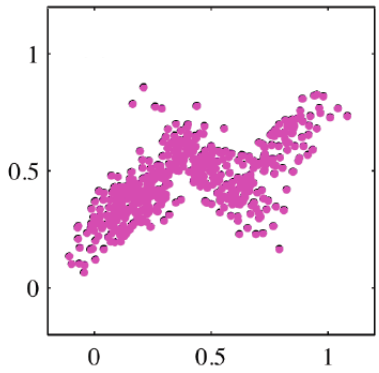
$$\mathcal{D} = \{\mathbf{x}_1 \cdots \mathbf{x}_N\}$$

- Assumiamo un modello MoG

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{m=1}^{M} \pi_m p(\mathbf{x}|\boldsymbol{\theta}_m)$$

- Vogliamo stimare

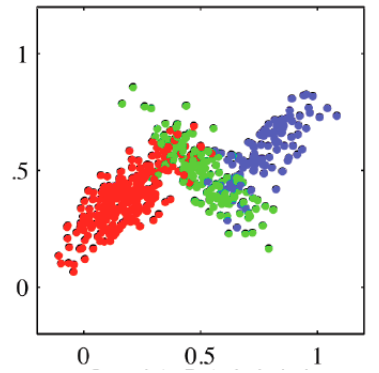
$$\boldsymbol{\theta} = \{\pi_1 \cdots \pi_M, \boldsymbol{\theta}_1 \cdots \boldsymbol{\theta}_M\}$$



- Supponiamo di avere la matrice Z degli indicatori:

$z_{mn} = 1$ il punto n è stato generato da Gaussiana m

$z_{mn} = 0$ il punto n non è stato generato da Gaussiana m



Misture di Gaussiane //stima dei parametri (ML)

- Supponiamo di avere la matrice Z degli indicatori:

  $z_{mn} = 1$ il punto n è stato generato da Gaussiana m

  $z_{mn} = 0$ il punto n non è stato generato da Gaussiana m

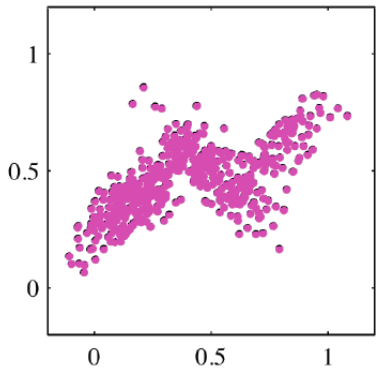
- La stima ML dei parametri $\boldsymbol{\theta}_m$ è la stima dei parametri di ciascuna Gaussiana m contando solo i punti che gli indicatori assegnano a quella Gaussiana

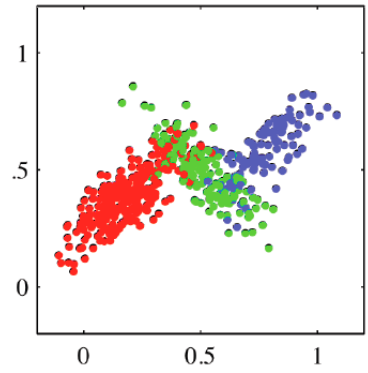
$$\widehat{\boldsymbol{\mu}}_m = \frac{\sum_{n=1}^{N} z_{nm}\mathbf{x}_n}{\sum_{n=1}^{N} z_{nm}} = \frac{1}{N_m}\sum_{n\in m}\mathbf{x}_n$$

$$\widehat{\boldsymbol{\Sigma}}_m = \frac{1}{N_m}\sum_{n=1}^{N} z_{mn}(\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_m)(\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_m)^{\mathsf{T}}$$

- Per π_m conto la frazione di punti nel cluster m

$$N_m = \sum_{n=1}^{N} z_{mn} \qquad \widehat{\pi}_m = \frac{N_m}{N}$$





Misture di Gaussiane //stima dei parametri (ML)

- Problema: NON ABBIAMO la matrice Z degli indicatori

  $z_{mn} = 1$ il punto n è stato generato da Gaussiana m

  $z_{mn} = 0$ il punto n non è stato generato da Gaussiana m

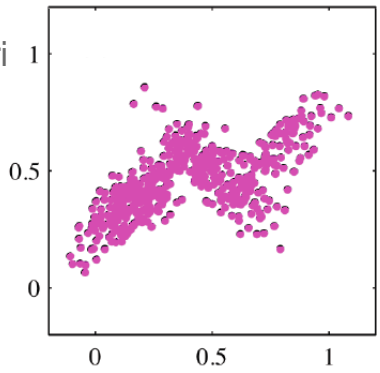
- z_{mn} sono variabili latenti o nascoste (hidden)

- Per una stima ML, ci serve la La verosimiglianza dei dati $\mathbf{X} = \{\mathbf{x}_1 \cdots \mathbf{x}_N\}$

$$p(\mathbf{X}|\boldsymbol{\theta})$$

- Se avessimo i parametri $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1 \cdots \boldsymbol{\theta}_M\}$

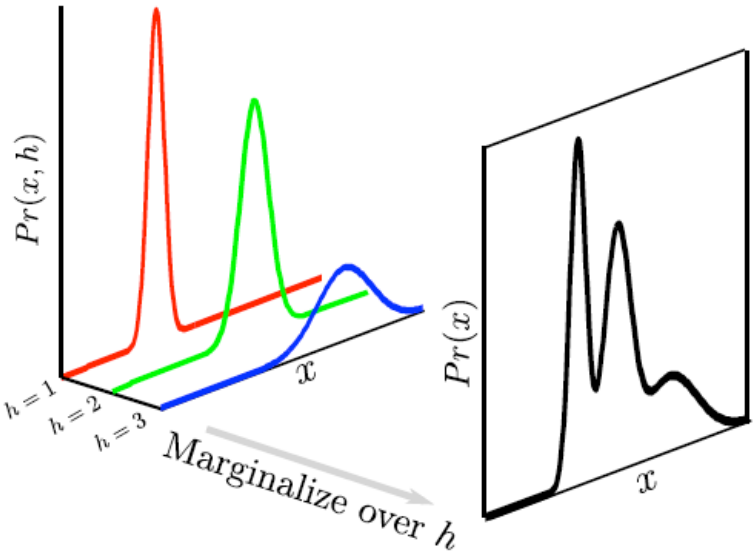
$$p(\mathbf{X}|\boldsymbol{\theta}) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$$



Misture di Gaussiane //stima dei parametri (ML)

- Se avessimo i parametri $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1 \cdots \boldsymbol{\theta}_M\}$

$$p(\mathbf{X}|\boldsymbol{\theta}) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$$



z_i = h variabili hidden

Misture di Gaussiane //stima dei parametri (ML)

- Passando alla log-verosimiglianza

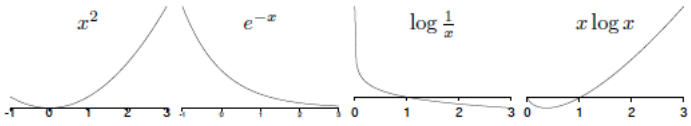
$$\begin{aligned} \log p(\mathbf{X}|\boldsymbol{\theta}) &= \log \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) \\ &= \log \sum_{\mathbf{Z}} P(\mathbf{Z}|\mathbf{X}) \frac{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}{P(\mathbf{Z}|\mathbf{X})} \end{aligned}$$

- Difficile da massimizzare in forma analitica per la presenza di $\log \sum_{\mathbf{Z}}$

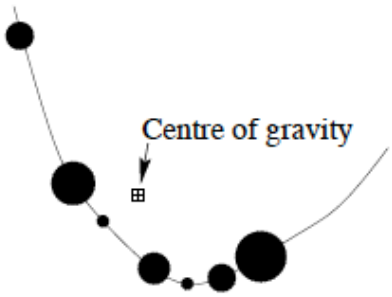
- Possiamo però determinare un lower bound più semplice e massimizzare il lower bound

Uno strumento per approssimare

- Disuguaglianza di Jensen: per una funzione convessa di x



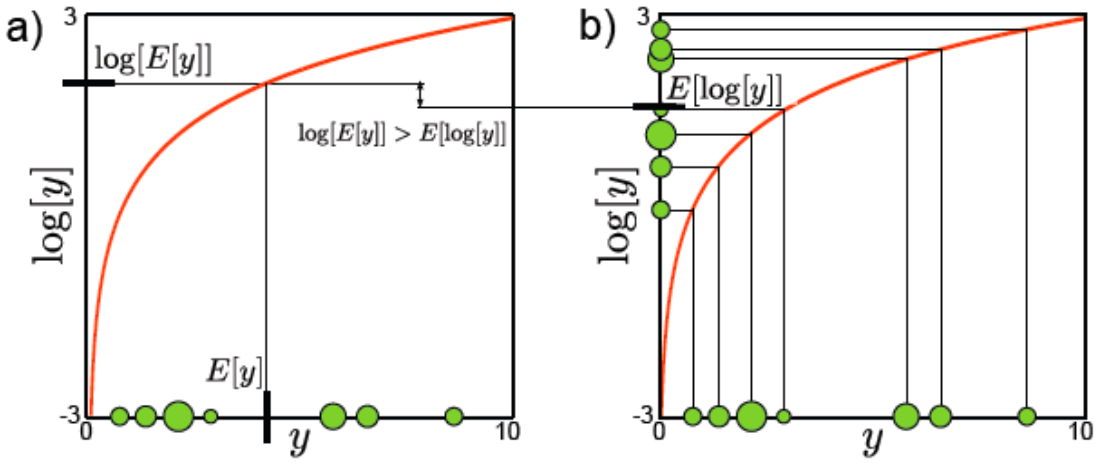
$$\mathcal{E}\left[f(x)\right] \geq f(\mathcal{E}[x])$$



Uno strumento per approssimare

- Per una funzione concava, si inverte la disuguaglianza
- Esempio: funzione logaritmica

$$\mathrm{E}[\log[y]] \leq \log(\mathrm{E}[y]$$



Misture di Gaussiane //stima dei parametri (ML)

- Passando alla log-verosimiglianza

$$\log p(\mathbf{X}|\boldsymbol{\theta}) = \log \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$$

$$= \log \sum_{\mathbf{Z}} P(\mathbf{Z}|\mathbf{X}) \frac{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}{P(\mathbf{Z}|\mathbf{X})} \quad \leftarrow E_{P(\mathbf{Z}|\mathbf{X})}\left\{\frac{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}{P(\mathbf{Z}|\mathbf{X})}\right\}$$

(Jensen) $$= \log E\{f(X)\} \geq E\{\log f(X)\}$$

$$= \sum_{\mathbf{Z}} P(\mathbf{Z}|\mathbf{X}) \log \frac{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}{P(\mathbf{Z}|\mathbf{X})}$$

Lower Bound

$$= \sum_{\mathbf{Z}} P(\mathbf{Z}|\mathbf{X}) \log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) - \sum_{\mathbf{Z}} P(\mathbf{Z}|\mathbf{X}) \log P(\mathbf{Z}|\mathbf{X})$$

Misture di Gaussiane //stima dei parametri (ML)

- Risultato se massimizzo il lower bound massimizzo la likelihood

$$\mathcal{L} = \log p(\mathbf{X}|\boldsymbol{\theta}) \geq \sum_{\mathbf{Z}} P(\mathbf{Z}|\mathbf{X}) \log \frac{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}{P(\mathbf{Z}|\mathbf{X})} = \mathcal{L}_B$$

Lower Bound

- Sotto l'ipotesi i.i.d

$$\mathcal{L}_B = \sum_{\mathbf{Z}} P(\mathbf{Z}|\mathbf{X}) \log \frac{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}{P(\mathbf{Z}|\mathbf{X})} = \sum_{m=1}^{M} \sum_{n=1}^{N} P(m|\mathbf{x}_n) \log \frac{p(\mathbf{x}_n|\boldsymbol{\theta}_m) P(m)}{P(m|\mathbf{x}_n)}$$

$$= \sum_{m=1}^{M} \sum_{n=1}^{N} P(m|\mathbf{x}_n) \log p(\mathbf{x}_n|\boldsymbol{\theta}_m) P(m) \quad \leftarrow \text{Probabilità che } z_{mn} = 1 \text{ per tutti gli n}$$

$$- \sum_{m=1}^{M} \sum_{n=1}^{N} P(m|\mathbf{x}_n) \log P(m|\mathbf{x}_n)$$

Probabilità che $z_{mn} = 1$

Misture di Gaussiane //stima dei parametri (ML): algoritmo EM

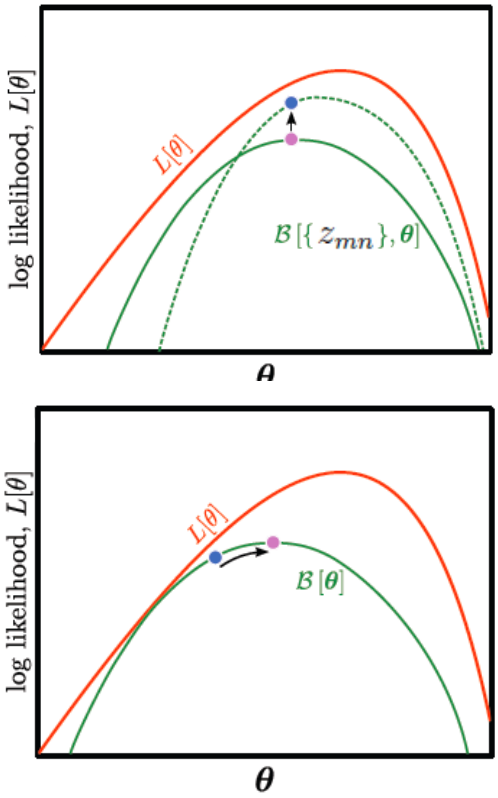
- Fino a convergenza:
  - E(xpectation) step: fissati i parametri, otteniamo gli indicatori z_{mn}

$$\frac{\partial \mathcal{L}_B}{\partial P(m|\mathbf{x}_n)} = 0$$

  - M(aximization) step: fissati gli indicatori z_{mn}, otteniamo i parametri

$$\frac{\partial \mathcal{L}_B}{\partial \boldsymbol{\theta}_m} = 0$$

$$\frac{\partial \mathcal{L}_B}{\partial P(m)} = 0$$



Misture di Gaussiane //stima dei parametri (ML): algoritmo EM

- Expectation

$$\mathcal{L}_B = \sum_{m=1}^{M}\sum_{n=1}^{N} P(m|\mathbf{x}_n)\log p(\mathbf{x}_n|\boldsymbol{\theta}_m)P(m) - \sum_{m=1}^{M}\sum_{n=1}^{N} P(m|\mathbf{x}_n)\log P(m|\mathbf{x}_n)$$

- Derivando:

$$\frac{\partial \mathcal{L}_B}{\partial P(m|\mathbf{x}_n)} = \log P(m|\mathbf{x}_n) - \log p(\mathbf{x}_n|\boldsymbol{\theta}_m)P(m) - 1 \quad \Rightarrow \quad P(m|\mathbf{x}_n) = \frac{p(\mathbf{x}_n|\boldsymbol{\theta}_m)P(m)}{\sum_{m'=1}^{M} p(\mathbf{x}_n|\boldsymbol{\theta}_{m'})P(m')}$$

Misture di Gaussiane //stima dei parametri (ML): algoritmo EM

- Maximization:
  - nel caso Gaussiano

$$\begin{aligned}\mathcal{L}_B &= \sum_{m=1}^{M}\sum_{n=1}^{N} P(m|\mathbf{x}_n)\log p(\mathbf{x}_n|\boldsymbol{\theta}_m)P(m) - \sum_{m=1}^{M}\sum_{n=1}^{N} P(m|\mathbf{x}_n)\log P(m|\mathbf{x}_n) \\ &= -\frac{1}{2}\sum_{m=1}^{M}\sum_{n=1}^{N} P(m|\mathbf{x}_n)\log|\boldsymbol{\Sigma}_k| \\ &\quad -\frac{1}{2}\sum_{m=1}^{M}\sum_{n=1}^{N} P(m|\mathbf{x}_n)(\mathbf{x}_n-\boldsymbol{\mu}_m)^{\mathsf{T}}\boldsymbol{\Sigma}_m^{-1}(\mathbf{x}_n-\boldsymbol{\mu}_m) \\ &\quad +\sum_{m=1}^{M}\sum_{n=1}^{N} P(m|\mathbf{x}_n)\log P(m)\end{aligned}$$

  - Derivando:

$$\widehat{\boldsymbol{\mu}}_m = \frac{\sum_{n=1}^{N} P(m|\mathbf{x}_n)\mathbf{x}_n}{\sum_{n=1}^{N} P(m|\mathbf{x}_n)} \qquad \widehat{\boldsymbol{\Sigma}}_m = \frac{\sum_{n=1}^{N} P(m|\mathbf{x}_n)(\mathbf{x}_n-\widehat{\boldsymbol{\mu}}_m)(\mathbf{x}_n-\widehat{\boldsymbol{\mu}}_m)^{\mathsf{T}}}{\sum_{n=1}^{N} P(m|\mathbf{x}_n)}$$

$$P(m) = \frac{1}{N}\sum_{n=1}^{N} P(m|\mathbf{x}_n)$$

Misture di Gaussiane //Algoritmo Expectation-Maximization

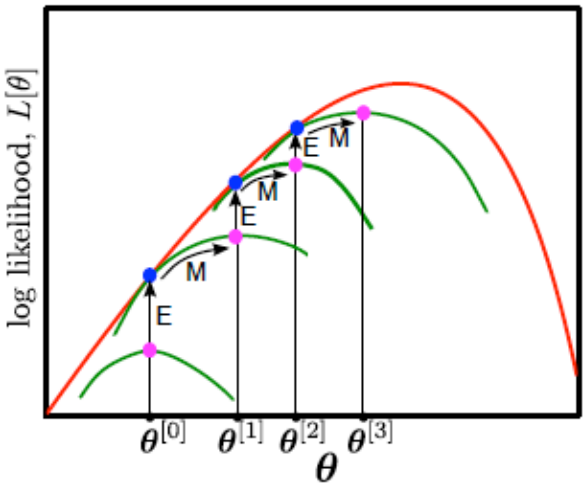
- Fino a convergenza:
  - E(xpectation) step: fissati i parametri, otteniamo gli indicatori z_{mn}

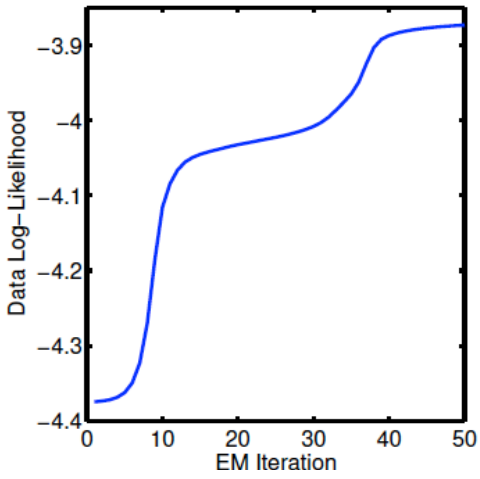
$$P(m|\mathbf{x}_n) = \frac{p(\mathbf{x}_n|\boldsymbol{\theta}_m)P(m)}{\sum_{m'=1}^{M} p(\mathbf{x}_n|\boldsymbol{\theta}_{m'})P(m')}$$

  - M(aximization) step: fissati gli indicatori z_{mn} , otteniamo i parametri

$$\begin{aligned}\widehat{\boldsymbol{\mu}}_m &= \frac{\sum_{n=1}^{N} P(m|\mathbf{x}_n)\mathbf{x}_n}{\sum_{n=1}^{N} P(m|\mathbf{x}_n)} \\ \widehat{\boldsymbol{\Sigma}}_m &= \frac{\sum_{n=1}^{N} P(m|\mathbf{x}_n)(\mathbf{x}_n-\widehat{\boldsymbol{\mu}}_m)(\mathbf{x}_n-\widehat{\boldsymbol{\mu}}_m)^{\mathsf{T}}}{\sum_{n=1}^{N} P(m|\mathbf{x}_n)} \\ P(m) &= \frac{1}{N}\sum_{n=1}^{N} P(m|\mathbf{x}_n)\end{aligned}$$

Misture di Gaussiane
//Algoritmo EM: log-likelihood durante iterazioni



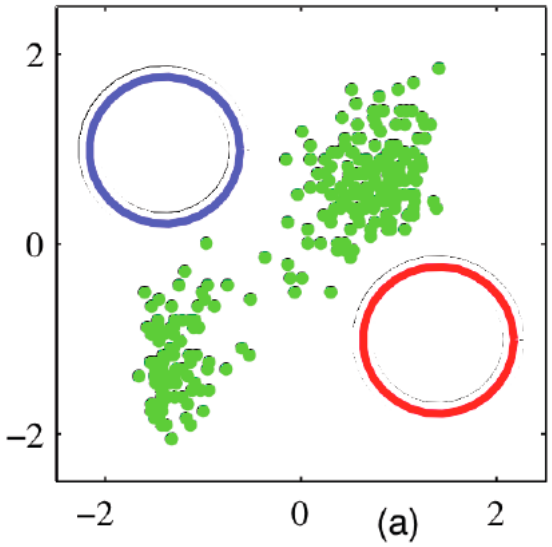


intuitivamente: quando L non cresce più di tanto,
posso terminare le iterazioni

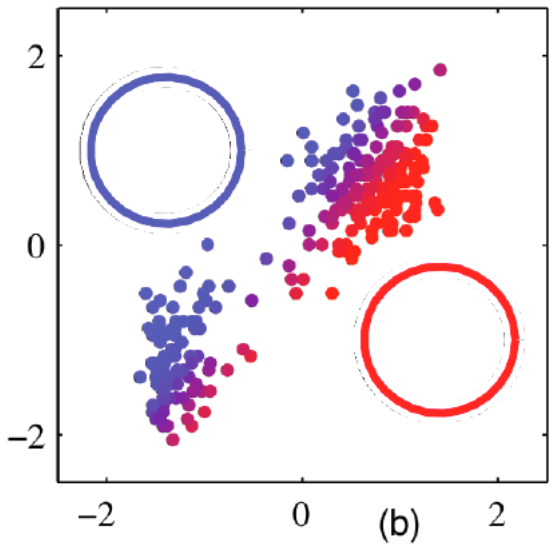
Misture di Gaussiane
//Algoritmo Expectation-Maximization (da PMTK3)

```
iter = 1;
done = false;
loglikHist = zeros(maxIter + 1, 1);
while ~done

    [ess, ll] = estep(model, data);

    loglikHist(iter) = ll;

    model = mstep(model, ess);

    done  = (iter > maxIter) || ( (iter > 1) && ...
        convergenceTest(loglikHist(iter), loglikHist(iter-1), convTol, true));

    iter = iter + 1;
end
loglikHist = loglikHist(1:iter-1);
```

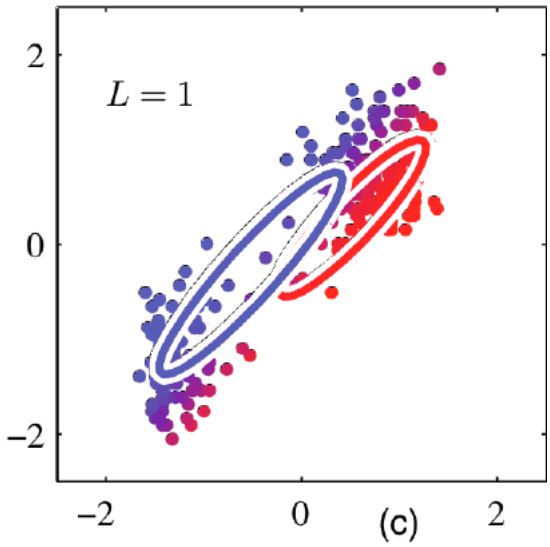
Misture di Gaussiane
//Algoritmo Expectation-Maximization



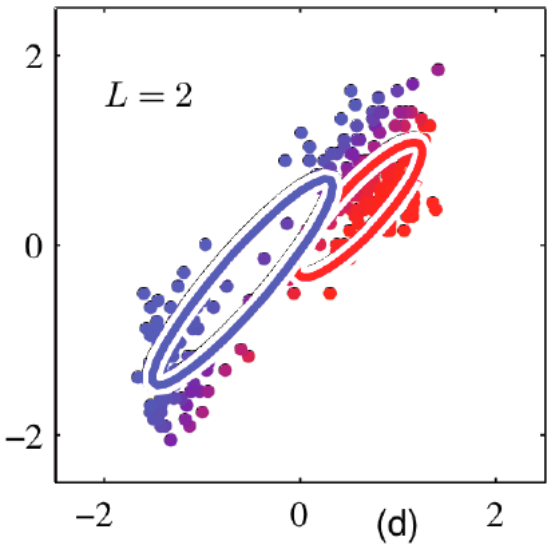
Misture di Gaussiane
//Algoritmo Expectation-Maximization



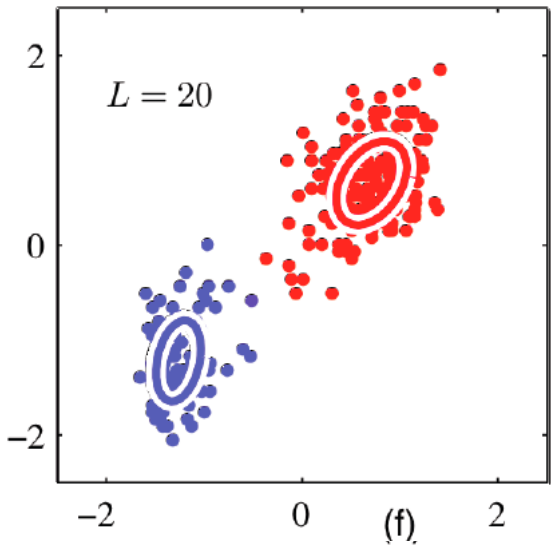
Misture di Gaussiane
//Algoritmo Expectation-Maximization



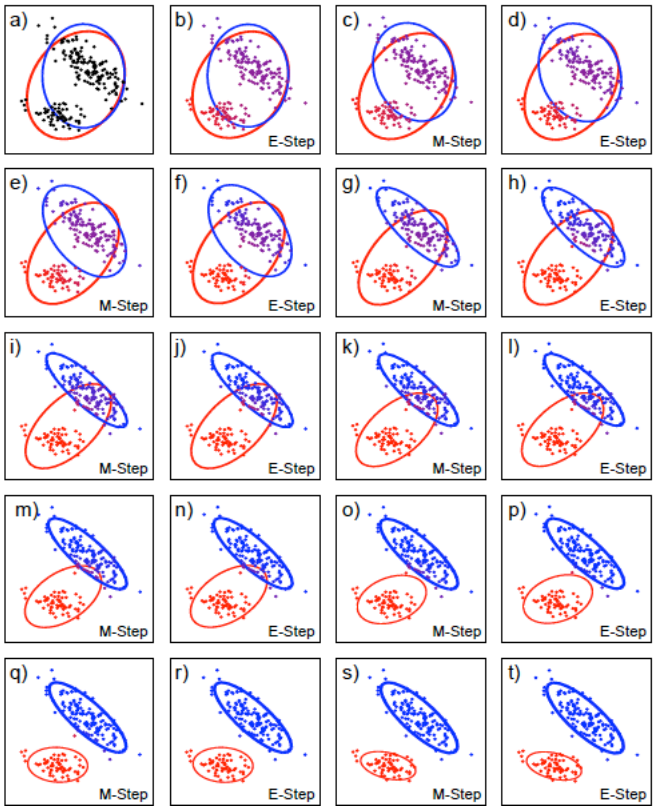
Misture di Gaussiane
//Algoritmo Expectation-Maximization



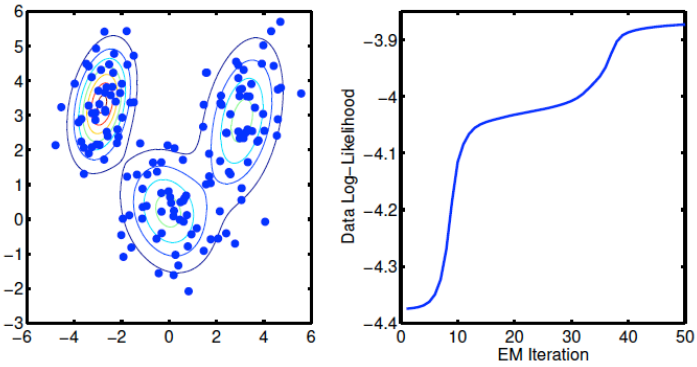
Misture di Gaussiane
//Algoritmo Expectation-Maximization



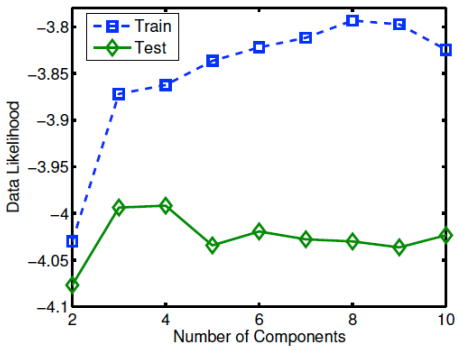
Misture di Gaussiane
//Algoritmo Expectation-Maximization



Misture di Gaussiane //Algoritmo Expectation-Maximization



massimizzazzione della log-likelihood durante le iterazioni di EM



Variazione della max log-likelihood al variare del numero di componenti M

Misture di Gaussiane //Algoritmo Expectation-Maximization

- Predizione: dopo aver appreso i parametri e la matrice Z, se volessi classificare un nuovo dato è sufficiente usare il passo di Expectation che corrisponde a Bayes e classificare in base alla regola map

$$\arg\max_{m} \; P(m|\mathbf{x}_n) = \frac{p(\mathbf{x}_n|\boldsymbol{\theta}_m)P(m)}{\sum_{m'=1}^{M} p(\mathbf{x}_n|\boldsymbol{\theta}_{m'})P(m')}$$

EM e k-means

- Il modo di funzionare ci ricorda quello del k-means
- Infatti il k-means è un caso particolare (degenere) di EM, che si ottiene considerando Gaussiane isotrope/sferiche (con matrice di covarianza fissata = matrice identità)
- In questo caso devo stimare solo le medie.
  - Passo E

$$E\{z_{kn}\} = P(k|\mathbf{x}_n) \propto \exp\left(-\frac{1}{2}||\mathbf{x}_n - \mathbf{m}_k||^2\right)$$

  - Passo M

$$\mathbf{m}_k = \frac{\sum_{n=1}^{N} P(k|\mathbf{x}_n)\mathbf{x}_n}{\sum_{m=1}^{N} P(k|\mathbf{x}_n)}$$

EM e k-means

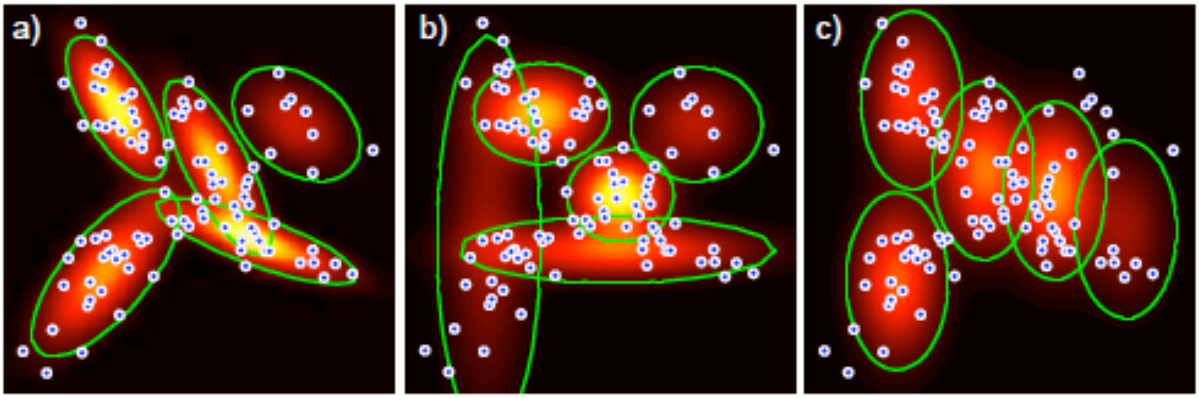


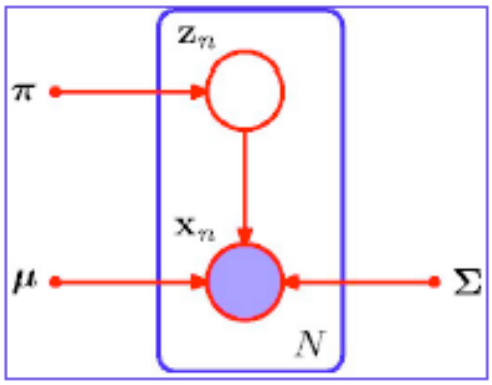
Figure 7.11 Covariance of components in mixture models. a) Full covariances. b) Diagonal covariances. c) Identical diagonal covariances.

Mistura di Gaussiane
//punto di vista generativo

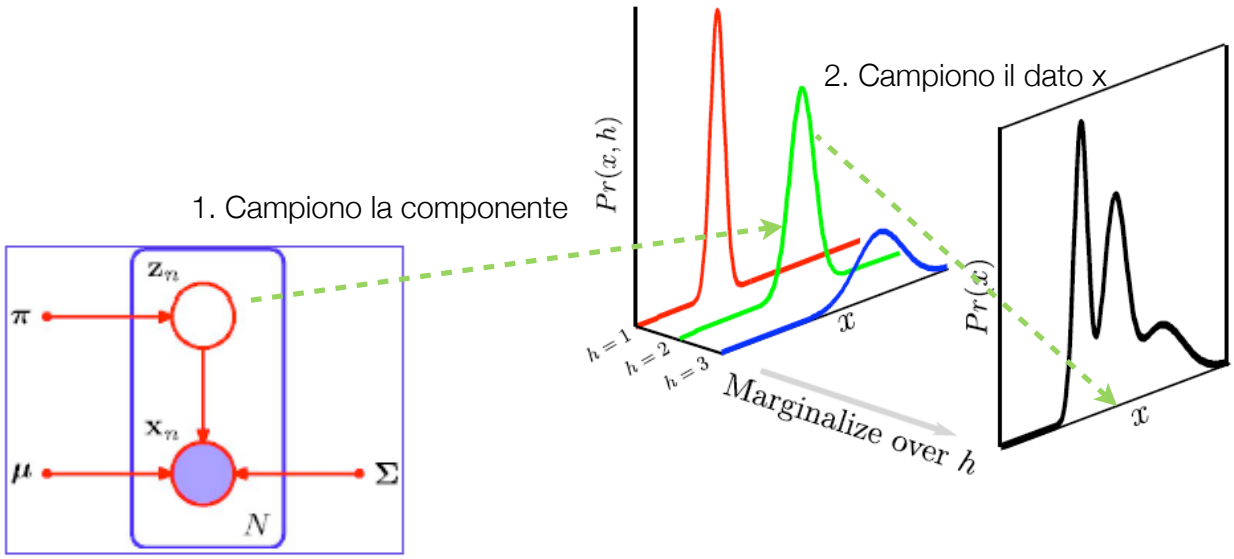
$$p(\mathbf{X}, \mathbf{Z} | \pi, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = p(\mathbf{X}|\mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) p(\mathbf{Z}|\pi)$$

$$p(\mathbf{Z}|\pi) = \prod_{n=1}^{N} \prod_{k=1}^{K} \pi^{z_{nk}}$$

$$p(\mathbf{X}|\mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1})^{z_{nk}}$$



Mistura di Gaussiane
//punto di vista generativo



Mistura di Gaussiane
//punto di vista generativo Bayesiano

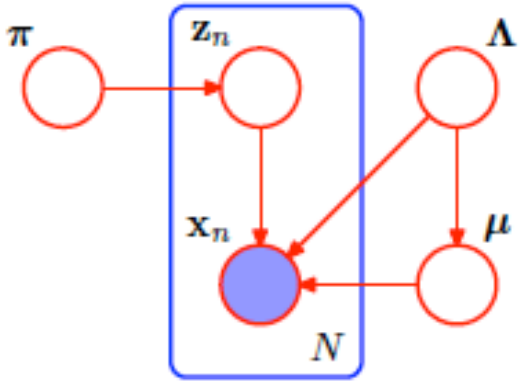
$$p(\mathbf{X}, \mathbf{Z}, \pi, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = p(\mathbf{X}|\mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) p(\mathbf{Z}|\pi) p(\boldsymbol{\mu}, \boldsymbol{\Lambda})$$

$$p(\pi) = \text{Dir}(\pi|\alpha_0)$$

$$p(\mathbf{Z}|\pi) = \prod_{n=1}^{N} \prod_{k=1}^{K} \pi^{z_{nk}}$$

$$p(\boldsymbol{\mu}, \boldsymbol{\Lambda}) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{\mu}_k|m_0, (\beta_0 \boldsymbol{\Lambda}_k)^{-1}) \mathcal{W}(\boldsymbol{\Lambda}_k|\mathbf{W}_0, \nu_0)$$

$$p(\mathbf{X}|\mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1})^{z_{nk}}$$

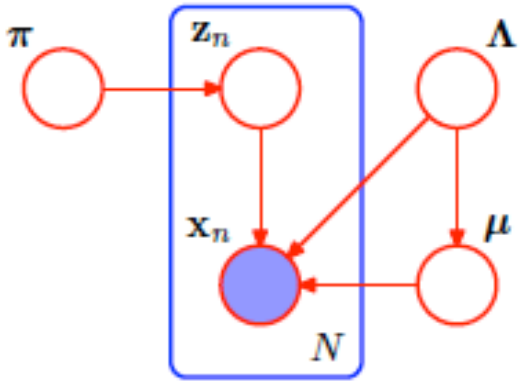


Mistura di Gaussiane
//punto di vista generativo Bayesiano

- Se intraprendiamo un'inferenza Bayesiana sulla congiunta, otteniamo una forma non integrabile analiticamente
- Adottiamo un'approssimazione variazionale basata sulla disuguaglianza di Jensen: Variational Bayes
- Se nelle formule di VB assumiamo che

$$p(\pi) \qquad p(\boldsymbol{\mu}, \boldsymbol{\Lambda})$$

- siano delta di Dirac (ML), otteniamo le equazioni di EM come caso particolare



Variational Bayes ⟶ EM ⟶ k-means

Modelli per la riduzione di dimensionlità

	Supervised Learning	*Unsupervised Learning*
Discrete	classification or categorization Discrete Output y $\in \{1, \ldots, K\}$	clustering
Continuous	regression Continuous Output $y \in R$	dimensionality reduction

Riduzione di dimensionalità

- Qual è la dimensionalità intrinseca a questi due data set?

Analisi per componenti principali
Principal Component Analysis (PCA)

- Trovare un piccolo numero (dimensione) di direzioni che spiega le correlazioni nei dati di input: lo spazio latente
- Si possono rappresentare i dati proiettandoli su tali direzioni
- I dati sono continui, il mapping tra lo spazio latente e lo spazio dei dati osservati è lineare
- Utile per:

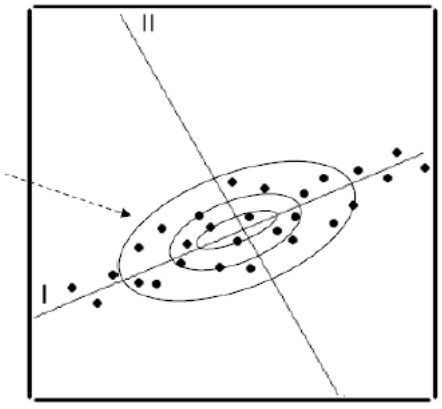
  - Visualization
  - Preprocessing
  - Modeling
  - Compression

Analisi per componenti principali
//Descrizione intuitiva

- N vettori di dati di dimensionalità D: $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}, \quad \mathbf{x}_n \in \Re^D$
- Spazio di proiezione (latente) di dimensionalità k<<D
- Si cercano le direzioni ortogonali di massima varianza
- La struttura dei vettori è rappresentata nella matrice di covarianza empirica

$$C = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$$

- Le direzioni cercate sono gli autovettori di C



Esempio

$$\boldsymbol{X}^1 = \begin{pmatrix} -1 \\ 3 \\ 1 \end{pmatrix}, \ \boldsymbol{X}^2 = \begin{pmatrix} 2 \\ 1 \\ -1 \end{pmatrix}, \ \boldsymbol{X}^3 = \begin{pmatrix} 2 \\ 2 \\ 3 \end{pmatrix}$$

$$M = \frac{1}{N}\sum_{k=1}^{N} X^k \longrightarrow M = \frac{1}{3}\begin{pmatrix} 3 \\ 6 \\ 3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

$$\boldsymbol{X}^1 - M = \begin{pmatrix} -2 \\ 1 \\ 0 \end{pmatrix}, \boldsymbol{X}^2 - M = \begin{pmatrix} 1 \\ -1 \\ -2 \end{pmatrix}, \boldsymbol{X}^3 - M = \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix}$$

$$C = \frac{1}{N}\sum_{k=1}^{N}(\boldsymbol{X}^k - M)(\boldsymbol{X}^k - M)^T \longrightarrow C = \frac{1}{3}\left\{\begin{bmatrix} 4 & -2 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & -1 & -2 \\ -1 & 1 & 2 \\ -2 & 2 & 4 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 2 \\ 0 & 0 & 0 \\ 2 & 0 & 4 \end{bmatrix}\right\}$$

$$= \frac{1}{3}\begin{bmatrix} 6 & -3 & 0 \\ -3 & 2 & 2 \\ -2 & 2 & 8 \end{bmatrix}$$

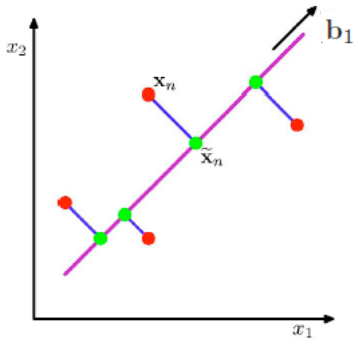
Analisi per componenti principali //Descrizione intuitiva

- Si selezionano gli autovettori di C

$$\mathbf{b}_1, \ldots, \mathbf{b}_K \qquad \mathbf{b}_j \in \mathbb{R}^d$$

- Si proiettano i vettori di input nel sottospazio:

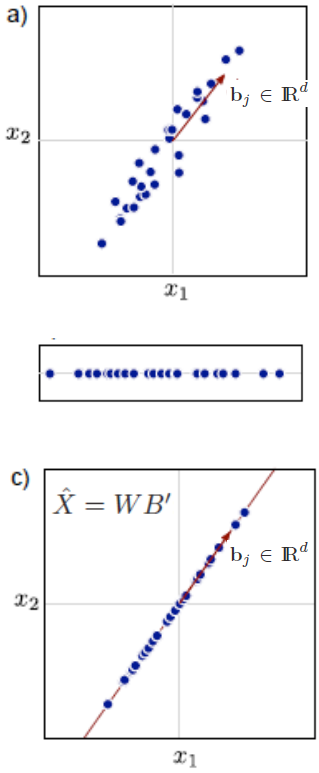
$$\mathbf{w}_i = \mathbf{x}_i^T \mathbf{b} \qquad W = XB$$



- Si puo' ricostruire x usando K autovettori

$$\mathbf{x}_i \approx \sum_{j=1}^{K} w_{ij}\mathbf{b}_j \qquad \hat{X} = WB'$$

Analisi per componenti principali //Descrizione intuitiva



$\mathrm{b}_1, \ldots, \mathrm{b}_K$

$\mathrm{w}_i = \mathrm{x}_i^T \mathrm{b}$

$\mathrm{x}_i \approx \sum_{j=1}^{K} w_{ij} \mathrm{b}_j$

Analisi per componenti principali //Descrizione intuitiva

Calcolo della base

$$\hat{C} = \frac{1}{n} \sum_{i=1}^{n} \mathrm{x}_i \mathrm{x}_i^T$$

```
K = rank(X);
options.disp = 0;
[Evec, evals] = eigs(cov(X), K, 'LM', options);
%[Evec, evals] = eig(cov(X));
% Sort so largest eval is first
[evals,index]=sort(Eval); index=flipud(index);  Evec = Evec(:, index);
B = Evec(:,1:K);
```

$\mathrm{b}_1, \ldots, \mathrm{b}_K$

Proiezione e ricostruzione

```
mu = mean(X);
[n d] = size(X);
XC = X-repmat(mu,n,1);
W = XC*B; % W is n*K
Xhat = W*B' + repmat(mu,n,1);
```

$\mathrm{w}_i = \mathrm{x}_i^T \mathrm{b} \quad W = XB$

$\mathrm{x}_i \approx \sum_{j=1}^{K} w_{ij} \mathrm{b}_j \quad \hat{X} = WB'$

Analisi per componenti principali //Esempi

- Dati input D= 19x19

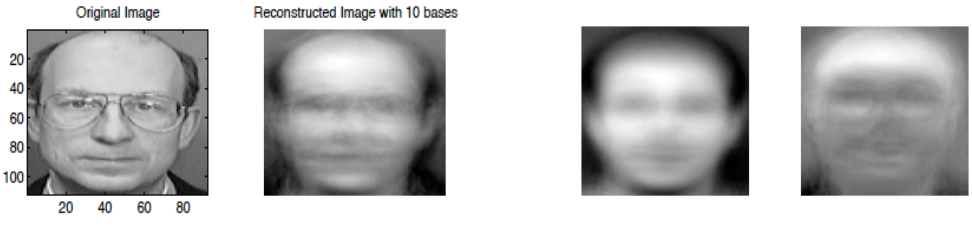
- Autovettori (K=48)

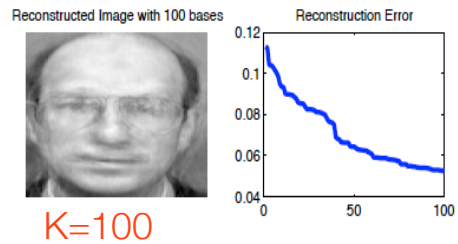
$\mathbf{b}_1, \ldots, \mathbf{b}_K$

Analisi per componenti principali //Esempi

$$\mathbf{x}_i \approx \sum_{j=1}^{K} w_{ij}\mathbf{b}_j$$

K=10



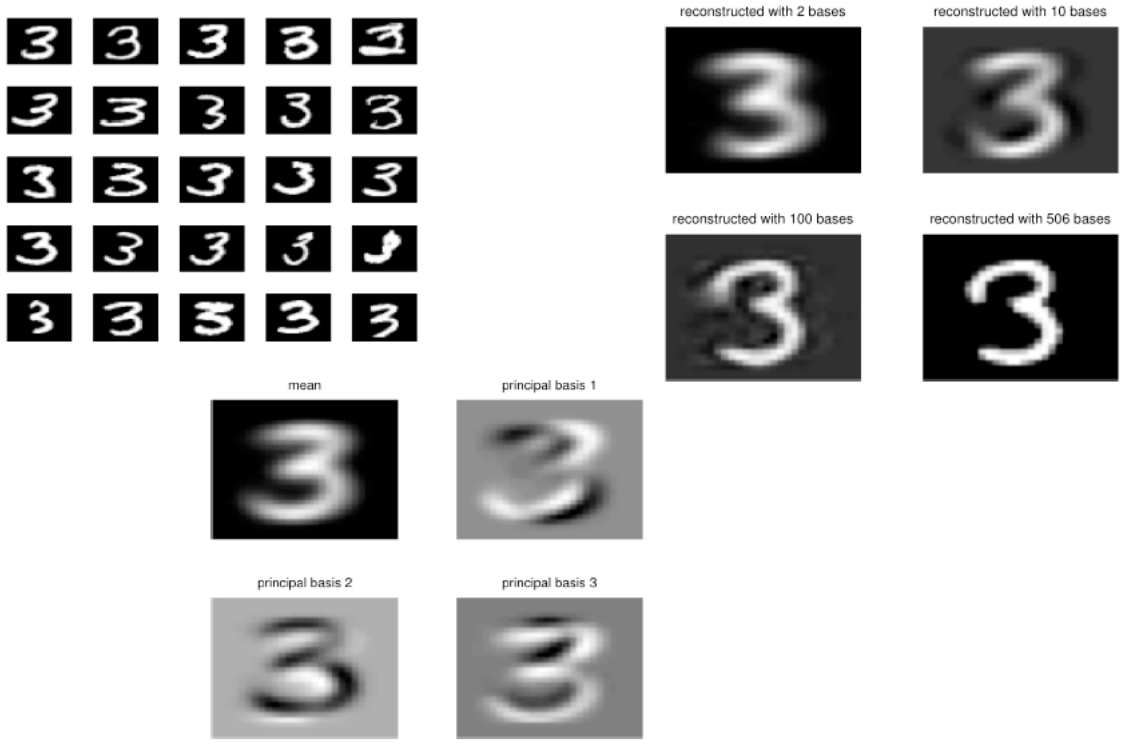


K=100

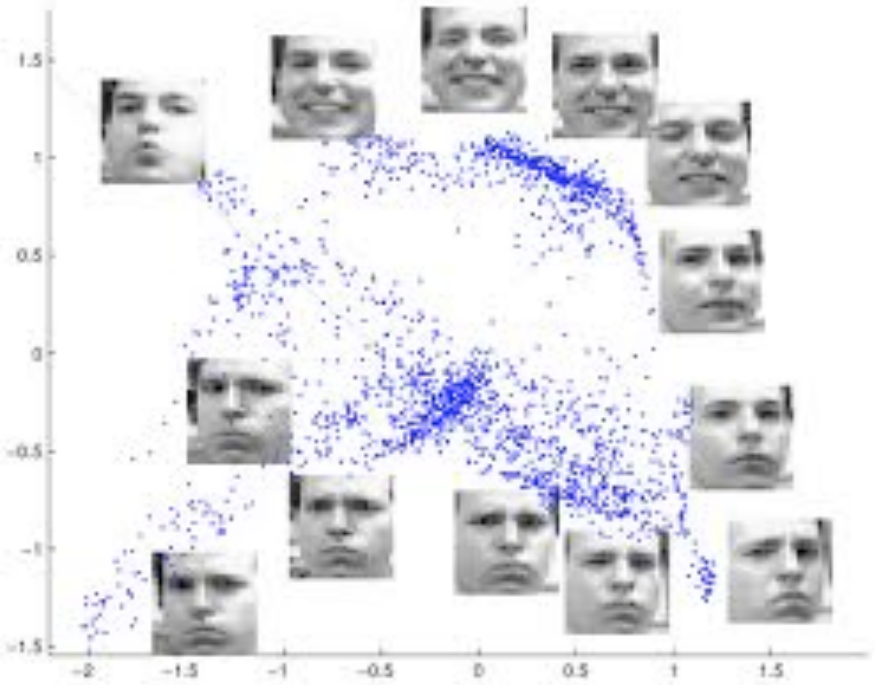
$$\mathbf{x}_i \approx \sum_{j=1}^{K} w_{ij}\mathbf{b}_j$$

$\mathbf{b}_1, \ldots, \mathbf{b}_K$

Analisi per componenti principali //Esempi



Analisi per componenti principali //Esempi



PCA (1)
//Direzione di massima varianza

- Consideriamo il mapping dallo spazio latente x a y $y_n = \mathbf{w}^\top \mathbf{x}_n$
- Trovare w in maniera da massimizzare var(y)

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \text{var}(y) = \arg\max_{\mathbf{w}} E_{\mathcal{D}}(y^2) = \arg\max_{\mathbf{w}} \frac{1}{N}\sum_n y_n^2$$

$$\begin{aligned} \frac{1}{N}\sum_n y_n^2 &= \frac{1}{N}\sum_n (\mathbf{w}^\top \mathbf{x}_n)^2 = \frac{1}{N}\sum_n \mathbf{w}^\top \mathbf{x}_n \mathbf{x}_n{}^\top \mathbf{w} \\ &= \mathbf{w}^\top \left(\frac{1}{N}\sum_n \mathbf{x}_n \mathbf{x}_n{}^\top\right)\mathbf{w} = \mathbf{w}^\top C \mathbf{w} \qquad C = \tfrac{1}{N}\textstyle\sum_n \mathbf{x}_n \mathbf{x}_n{}^\top \end{aligned}$$

- sotto il vincolo $\|\mathbf{w}\|^2 = \mathbf{w}^\top \mathbf{w} = 1$

$$\boxed{\mathbf{w}^* = \arg\max_{\mathbf{w}} \; \mathbf{w}^\top C \mathbf{w} - \lambda(\mathbf{w}^\top \mathbf{w} - 1)}$$

- Si può dimostrare che w* è autovettore di C

PCA (2)
//Minimizzazione dell'errore di ricostruzione

- Consideriamo il mapping dallo spazio latente x a y $y_n = \mathbf{w}^\top \mathbf{x}_n$
- Vogliamo minimizzare l'errore quadratico di ricostruzione

$$\min_{\{\alpha_n\},\mathbf{w}} \|\mathbf{x}_n - \alpha_n \mathbf{w}\|^2$$

- Risolvendo per alfa

$$\alpha_n = \frac{\mathbf{w}^\top \mathbf{x}_n}{\mathbf{w}^\top \mathbf{w}}$$

- e sostituendo $\min_{\mathbf{w}} \sum_n \|\mathbf{x}_n - (\mathbf{w}^\top \mathbf{x}_n)\mathbf{w}\|^2$
- Con il vincolo $\|\mathbf{w}\|^2 = \mathbf{w}^\top \mathbf{w} = 1$

$$\boxed{\mathbf{w}^* = \arg\max_{\mathbf{w}} \; \mathbf{w}^\top C \mathbf{w} - \lambda(\mathbf{w}^\top \mathbf{w} - 1)}$$

Modelli a variabili latenti //PCA probabilistica (PPCA)

- Si utilizza un modello generativo

$$\mathbf{z}_i \sim \mathcal{N}(0, \mathbf{I})$$
$$\boldsymbol{\epsilon}_i \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$$
$$\mathbf{x}_i = \mathbf{W}\mathbf{z}_i + \boldsymbol{\mu} + \boldsymbol{\epsilon}_i$$