


# Algoritmi di minimizzazione – EM – Mixture models

I. Frosio
AIS Lab.
frosio@dsi.unimi.it
http:\\homes.dsi.unimi.it\~frosio\

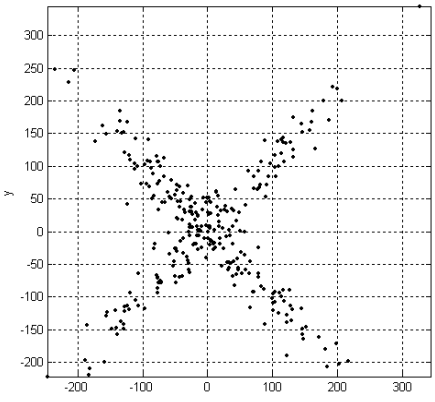



## Overview

- Stima di due rette (riassunto)
- Minimizzazione: metodi "di ordine zero" (simplesso, …)
- Minimizzazione: metodi "di primo ordine" (gradiente, …)
- Minimizzazione: metodi "di secondo ordine" (Newton, …)
- Minimizzazione: Expectation-Maximization
- Mixture models: introduzione
- Moxture models: segmentazione di radiografia cefalometrica

# Stima di due rette (riassunto)

- Si vogliono stimare i coefficienti angolari di due rette passanti per l'origine.
- I dati misurati yi possono provenire dall'una o dall'altra retta con la stessa probabilità.
- Sui dati misurati è presente rumore gaussiano con varianza $\sigma^2$.






# Stima di due rette (riassunto)

- Scriviamo la funzione di verosimiglianza:

$$p(yi) = P1 \cdot G(m1 \cdot xi, \sigma^2) + P2 \cdot G(m2 \cdot xi, \sigma^2)$$

$$G(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

- In pratica un punto yi può provenire dalla retta 1 con probabilità P1 o dalla retta 2 con probabilità P2. In ciascuno dei due casi il punto misurato ha una distribuzione gaussiana "centrata" sulla retta stessa.

# Stima di due rette (riassunto)

- Calcolo il logaritmo negativo della verosimiglianza:

$$f(m1,m2) = -\sum_{i=1}^{N} \ln[p(yi)] = -\sum_{i=1}^{N} \ln\left[P1 \cdot G\left(m1 \cdot xi, \sigma^2\right) + P2 \cdot G\left(m2 \cdot xi, \sigma^2\right)\right]$$

$$= -\sum_{i=1}^{N} \ln\left[P1 \cdot p_1\left(xi, \sigma^2\right) + P2 \cdot p_2\left(xi, \sigma^2\right)\right]$$

$$p_j\left(xi, \sigma^2\right) = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{1}{2}\left(\frac{x - m_j \cdot xi}{\sigma}\right)^2}$$

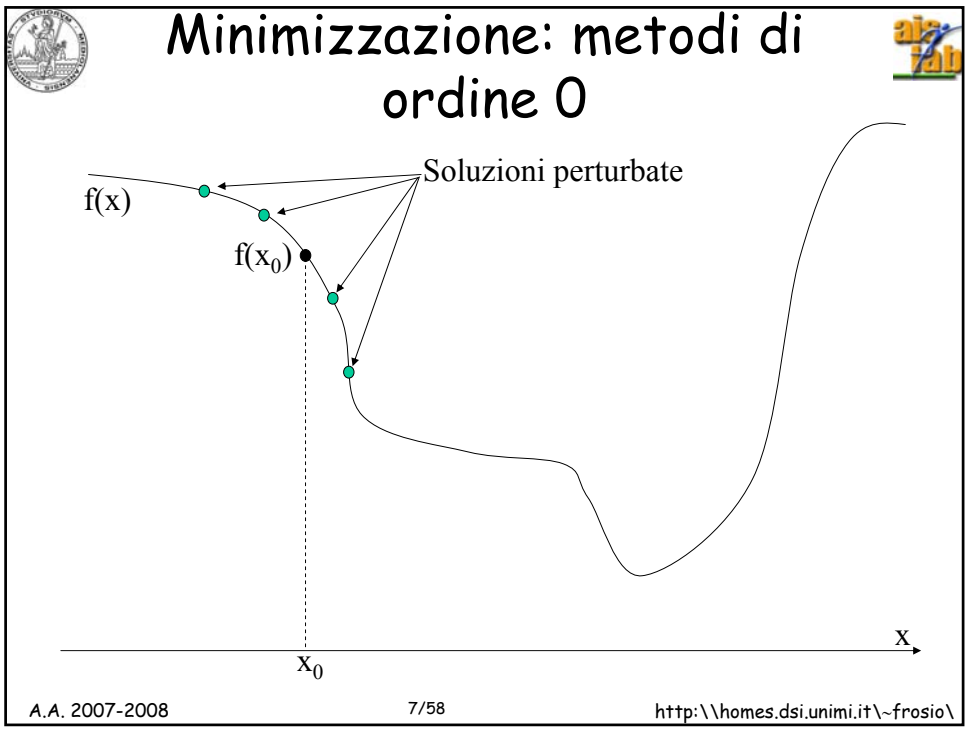
- Non può essere minimizzato analiticamente ponendo le derivate uguali a zero – è necessario usare un algoritmo di minimizzazione iterativo.

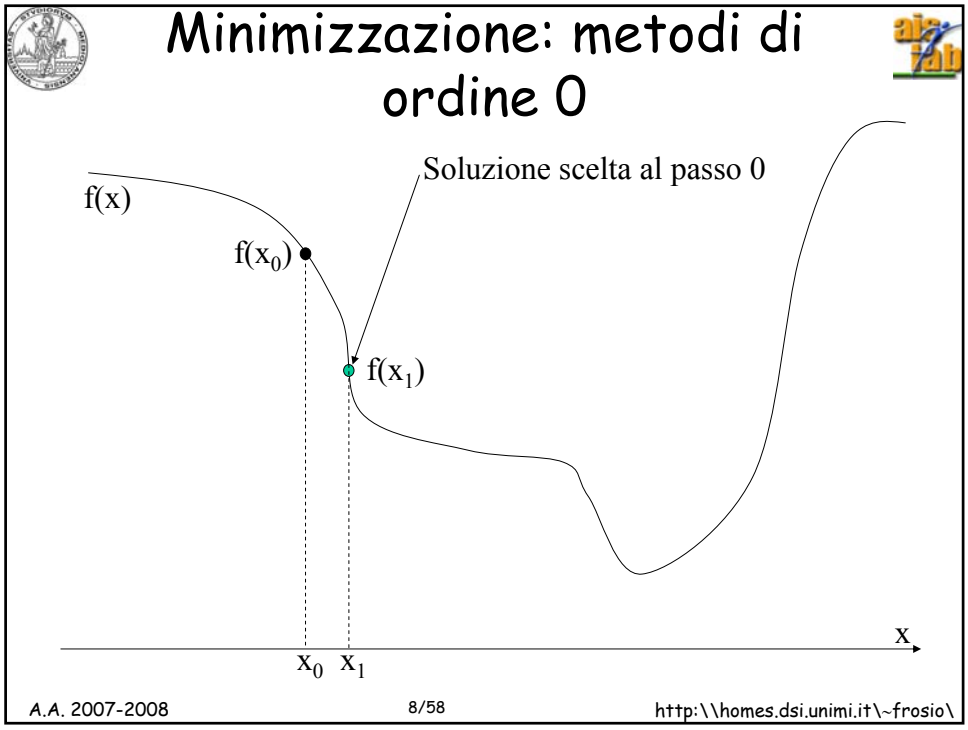


# Minimizzazione: metodi di ordine 0

Strategia di base:

- Sia f(x) la funzione da minimizzare, $x_0$ la soluzione di partenza;
- La soluzione viene perturbata ($x_{k+1}=x_k+\Delta x$) in modo più meno "furbo", testando diverse perturbazioni $\Delta x$.
- Per ogni perturbazione $\Delta x$ si calcola la $f(x_k+\Delta x)$.
- Si sceglie la $\Delta x$ tale per cui la $f(x_k+\Delta x)$ è minima e si aggiorna la soluzione ($x_{k+1}=x_k+\Delta x$).
- Esempi: metodo del simplesso, simulated annealing, algoritmi genetici…
- Se si utilizza un metodo di ordine 0 è necessario calcolare la sola f(x).
- Facile da implementare, bassa velocità di convergenza.


Minimizzazione: metodi di ordine 0
Soluzioni perturbate
f(x)
f(x0)
x
x0
A.A. 2007-2008
7/58
http:\\homes.dsi.unimi.it\~frosio\


Minimizzazione: metodi di ordine 0
Soluzione scelta al passo 0
f(x)
f(x0)
f(x1)
x
x0 x1
A.A. 2007-2008
8/58
http:\\homes.dsi.unimi.it\~frosio\

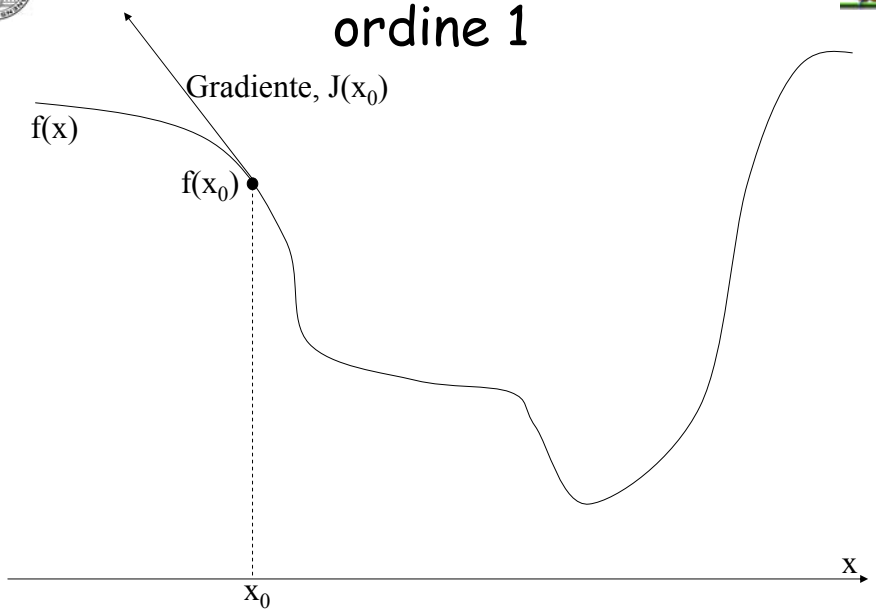
# Minimizzazione: metodi di ordine 1

Strategia di base:

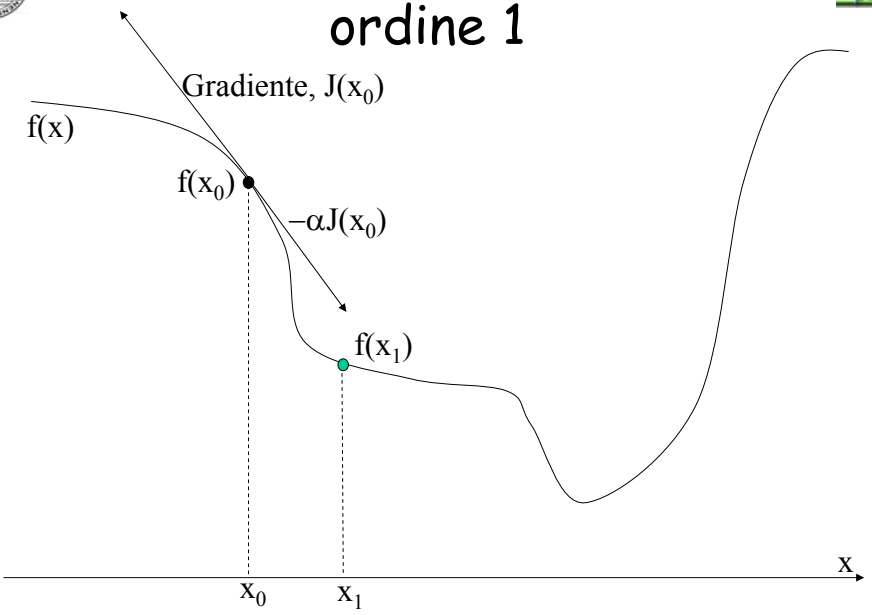
- Sia f(x) la funzione da minimizzare, $x_0$ la soluzione di partenza;
- Si calcoli il gradiente della f in $x_k$, $J(x_k)$.
- La soluzione viene aggiornata utilizzando l'informazione del gradiente.
- Nell'ipotesi più semplice (metodo del gradiente), muovendosi nella direzione opposta rispetto al gradiente ($x_{k+1}=x_k-\alpha J(x_k)$.)
- Esempi: metodo del gradiente, gradiente coniugato, …
- Se si utilizza un metodo di ordine 1 è necessario calcolare la f(x) e J(x).
- Abbastanza facile da implementare, velocità di convergenza media (dipende dalla strategia adottata).
- Il parametro scalare $\alpha$ (learning rate) determina la velocità di convergenza e la stabilità del metodo.



# Minimizzazione: metodi di ordine 1



# Minimizzazione: metodi di ordine 1






# Minimizzazione: metodi di ordine 1

- Nel caso del problema delle due rette abbiamo già calcolato le derivate (si veda la lezione precedente).
- Utilizzando il metodo del gradiente, la soluzione può essere aggiornata come:

$$mj^{k+1} = mj^{k} - \alpha \cdot \frac{\partial f(m1, m2)}{\partial mj} = mj^{k} - \alpha \cdot \sum_{i=1}^{N} \frac{Pj}{p(xi)} \cdot p_j(xi) \cdot \frac{(x - m_j \cdot xi) \cdot xi}{\sigma^2}$$

- (Esercizio → implementazione Matlab del metodo del gradiente – confronto con EM per vari valori di $\alpha$)

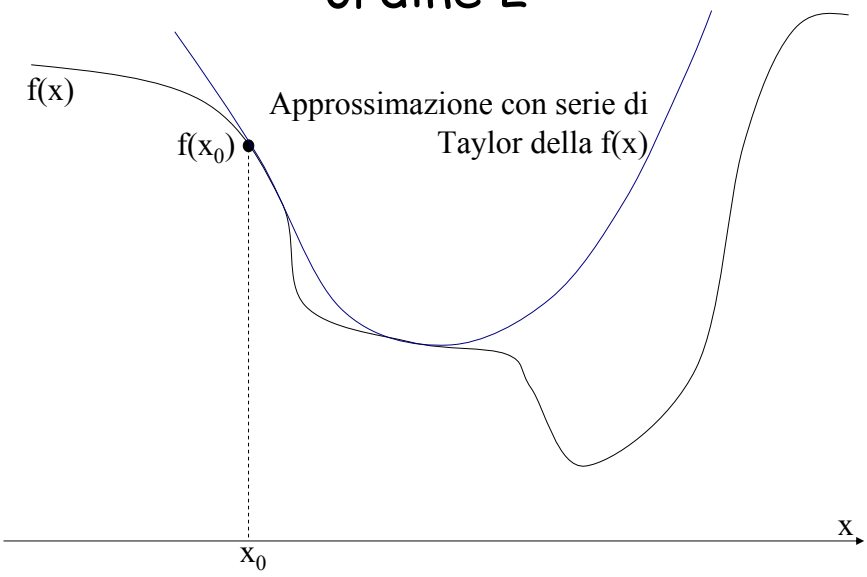
# Minimizzazione: metodi di ordine 2

Strategia di base:

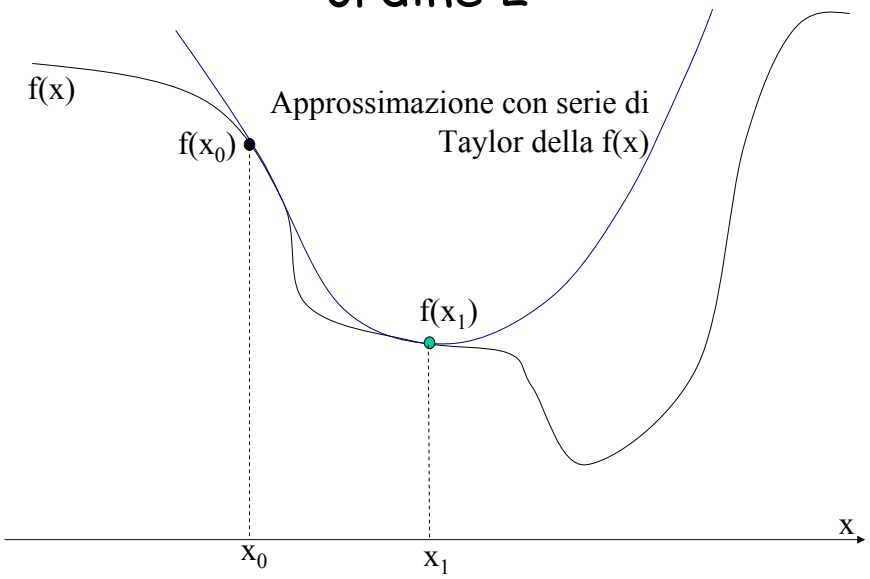
- Sia f(x) la funzione da minimizzare, $x_0$ la soluzione di partenza;
- L'espansione in serie di Taylor arrestata al 2° ordine di f(x) è la seguente:
  $f(x_k+\Delta x)=f(x_k)+J(x_k)\,(\Delta x)+1/2(\Delta x)^T H(x_k)(\Delta x)$,
  dove $H(x_k)$ è l'Hessiano di f.
- L'espansione in serie di Taylor dà un'approssimazione locale della f(x).
- E' possibile minimizzare analiticamente tale approssimazione; il minimo si ottiene derivando e ponendo la derivata uguale a zero (metodo di Newton):
- $J(x_k)+H(x_k)(\Delta x)=0 \rightarrow \mathbf{\Delta x = -H(x_k)^{-1}J(x_k)}$
- E' oneroso calcolare l'hessiano, velocità di convergenza alta.



# Minimizzazione: metodi di ordine 2



# Minimizzazione: metodi di ordine 2






# Minimizzazione – metodi iterativi (riassunto)

- Quando la funzione da minimizzare è fortemente non lineare, è necessario ricorrere ad un metodo iterativo.
- Si fanno delle ipotesi sull'andamento locale della f(x) e si aggiorna la soluzione x in modo da garantire che la f(x) decresca, fino al raggiungimento di un minimo locale.
- I metodi di ordine 0 utilizzano il calcolo della sola f(x) per studiarne l'andamento locale e scegliere l'aggiornamento della soluzione – facili da implementare, lenti nella convergenza.
- I metodi di ordine 1 utilizzano anche il gradiente per studiare l'andamento locale della soluzione – discretamente complessi, velocità di convergenza media.
- I metodi di ordine 2 utilizzano anche l'hessiano per descrivere con maggior cura l'andamento locale della soluzione (funzione "surrogato") e minimizzano ad ogni iterazione la funzione "surrogato" invece della f(x) – il calcolo dell'hessiano è complesso, velocità di convergenza alta.

# Minimizzazione: Expectation-Maximization

- Expectation-Maximization: un algoritmo di massimizzazione della verosimiglianza per modelli con variabili latenti.
- Si consideri un problema di massimizzazione della verosimiglianza (es. stima delle due rette).
- Tale problema ha una formulazione complessa, ma…
- … Se vengono inserite alcune variabili nascoste nel modello, la formulazione matematica del problema può essere semplificata.



# Minimizzazione: Expectation-Maximization

- Sia $\theta$ il vettore dei parametri del modello da stimare;
- Scriviamo la verosimiglianza del problema introducendo delle variabili nascoste, Z, oltre alle variabili misurate, Y:

$$\ln\left[p(Y \mid \theta)\right] = \ln\left\{\sum_{Z} p(Y, Z \mid \theta)\right\}$$

# Minimizzazione: Expectation-Maximization

- Nel caso della stima delle due rette…
- … Cosa succederebbe se ci fosse data l'appartenenza di una misura yi all'una o all'altra retta?
- In tal caso il problema si ridurrebbe a due semplici problemi di stima di retta ai minimi quadrati.
- Per formulare EM, introduciamo delle variabili nascoste Z che descrivono l'appartenenza di una misura yi alla retta 1 o alla retta 2.



# Minimizzazione: Expectation-Maximization

- Dal momento che le variabili nascoste non sono note, devono essere stimate a partire dai dati.
- Utilizzando il teorema di Bayes, e considerando un vettore di parametri stimato $\theta^{old}$, possiamo stimare le probabilità delle Z, $p(Z|Y, \theta^{old})$.

# Minimizzazione: Expectation-Maximization

- Stimiamo le variabili nascoste per il problema di stima delle due rette.
- Dal teorema di Bayes:

Probabilità che sia stata la retta J-esima a generare il dato y

Probabilità del dato y quando generato dalla componente J

Probabilità della componente J

$$p(J \mid y) = \frac{p(y \mid J) \cdot p(J)}{p(y)}$$

Probabilità del dato y



# Minimizzazione: Expectation-Maximization

- Specificando per nostro caso:

$$p(J \mid yi) = \frac{p_j(yi) \cdot Pj}{P1 \cdot p_1(yi) + P2 \cdot p_2(yi)}$$

- La p(j|yi) descrive il "grado di appartenenza" del dato yi alla retta j.
- All'iterazione k-esima, può essere calcolato con i parametri m1, m2 dell'iterazione stessa.

# Minimizzazione: Expectation-Maximization

- Ciò consente di costruire la funzione di Expectation, che verrà massimizzata ad ogni passo - si richiede di **massimizzare la media su Z della funzione di verosimiglianza**:

Maximization

$$\theta^{new} = \arg\max_{\theta} Q(\theta, \theta^{old})$$

$$Q(\theta, \theta^{old}) = \sum_{Z} \{ p(Z \mid Y, \theta^{old}) \cdot \ln[p(Y, Z \mid \theta)] \}$$

Expectation



# Minimizzazione: Expectation-Maximization

- Scriviamo il passo di maximization per il problema di stima delle due rette:

$$Q(\theta, \theta^{old}) = \sum_{Z} \{ p(Z \mid Y, \theta^{old}) \cdot \ln[p(Y, Z \mid \theta)] \} \Rightarrow$$

$$\sum_{Z} \{ p(Z \mid Y, m1^{old}, m2^{old}) \cdot \ln[p(Y, Z \mid m1, m2)] \}$$

Costante, calcolata nella fase di Expectation → K(yi,j)

# Minimizzazione: Expectation-Maximization

$$Q = \sum_{i=1}^{N}\sum_{j=1}^{2} k(yi,j)\cdot \ln\left[p_j(yi)\right] = \sum_{i=1}^{N}\sum_{j=1}^{2} k(yi,j)\cdot \ln\left[\frac{1}{\sqrt{2\pi}\sigma}\cdot e^{-\frac{1}{2}\left(\frac{yi-mj\cdot xi}{\sigma}\right)^2}\right] =$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{2} k(yi,j)\cdot \ln\left[\frac{1}{\sqrt{2\pi}\sigma}\right] + \sum_{i=1}^{N}\sum_{j=1}^{2} k(yi,j)\cdot\left[-\frac{1}{2}\left(\frac{yi-mj\cdot xi}{\sigma}\right)^2\right] =$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{2} k(yi,j)\cdot \ln\left[\frac{1}{\sqrt{2\pi}\sigma}\right] - \frac{1}{2\sigma^2}\cdot\sum_{i=1}^{N}\sum_{j=1}^{2} k(yi,j)\cdot(yi-mj\cdot xi)^2$$



# Minimizzazione: Expectation-Maximization

- Cerchiamo allora il massimo di Q (Maximization) per avere l'aggiornamento della soluzione:

$$\frac{\partial Q}{\partial mj} = \frac{\partial}{\partial mj}\left\{\sum_{i=1}^{N}\sum_{j=1}^{2} k(yi,j)\cdot \ln\left[\frac{1}{\sqrt{2\pi}\sigma}\right] - \frac{1}{2\sigma^2}\cdot\sum_{i=1}^{N}\sum_{j=1}^{2} k(yi,j)\cdot(yi-mj\cdot xi)^2\right\} =$$

$$= 0 - \frac{1}{2\sigma^2}\cdot\frac{\partial}{\partial mj}\sum_{i=1}^{N}\sum_{j=1}^{2} k(yi,j)\cdot(yi-mj\cdot xi)^2 =$$

$$= -\frac{1}{2\sigma^2}\cdot\frac{\partial}{\partial mj}\sum_{i=1}^{N} k(yi,j)\cdot(yi-mj\cdot xi)^2 =$$

$$= -\frac{1}{2\sigma^2}\cdot\sum_{i=1}^{N} k(yi,j)\cdot 2\cdot(yi-mj\cdot xi)\cdot(-xi) =$$

$$= \frac{1}{\sigma^2}\cdot\sum_{i=1}^{N} k(yi,j)\cdot(yi-mj\cdot xi)\cdot(xi)$$

# Minimizzazione: Expectation-Maximization

- Ponendo la derivata di Q uguale a zero…

$$\frac{\partial Q}{\partial mj} = \frac{1}{\sigma^2} \cdot \sum_{i=1}^{N} k(yi, j) \cdot (yi - mj \cdot xi) \cdot (xi) = 0 \Rightarrow$$

$$\sum_{i=1}^{N} k(yi, j) \cdot (yi - mj \cdot xi) \cdot (xi) = 0 \Rightarrow$$

$$\sum_{i=1}^{N} k(yi, j) \cdot (yi \cdot xi) - mj \cdot \sum_{i=1}^{N} k(yi, j) \cdot (xi^2) = 0 \Rightarrow$$

$$mj = \frac{\sum_{i=1}^{N} k(yi, j) \cdot (yi \cdot xi)}{\sum_{i=1}^{N} k(yi, j) \cdot (xi^2)}$$

- Otteniamo le equazioni di aggiornamento per i parametri del modello secondo EM.



# Minimizzazione: Expectation-Maximization

- Analizzando la funzione di aggiornamento per EM…

$$mj = \frac{\sum_{i=1}^{N} k(yi, j) \cdot (yi \cdot xi)}{\sum_{i=1}^{N} k(yi, j) \cdot (xi^2)}$$

- I parametri vengono aggiornati secondo uno schema ai minimi quadrati pesati dai fattori k(yi,j) che descrivono l'appartenenza di un dato alla retta 1 o alla retta 2.
- Vedere anche → codice Matlab

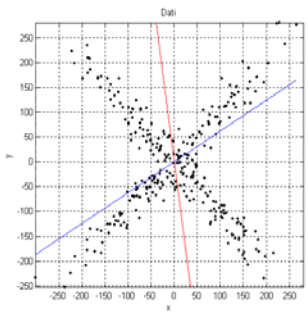
# Minimizzazione: Expectation-Maximization

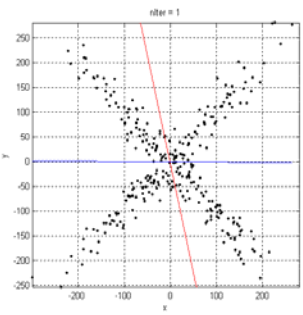
- Discorso "al limite" per interpretare EM…
- Si assegna un punto alla retta 1 o alla retta 2 sulla base della distanza dalle due rette (expectation);
- Si risolvono i due problemi di stima ai minimi quadrati (facili! - maximization);
- Si itera fino a convergenza…

- Nell'EM reale l'assegnazione è "soft" invece che "hard".
- Vedere anche → codice Matlab

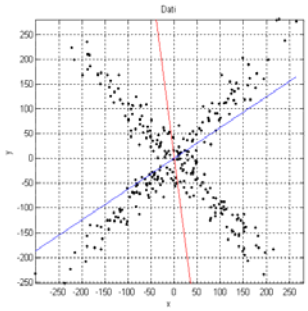


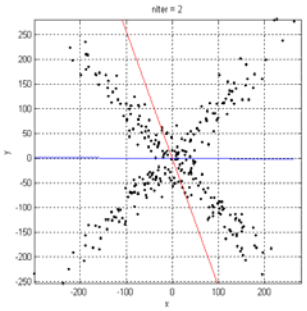
# Minimizzazione: Expectation-Maximization



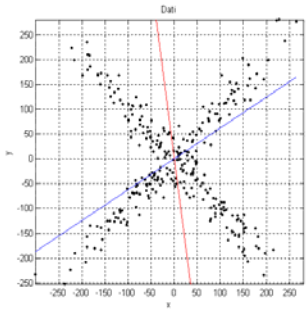
# Minimizzazione: Expectation-Maximization

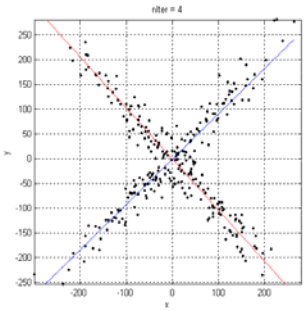





# Minimizzazione: Expectation-Maximization



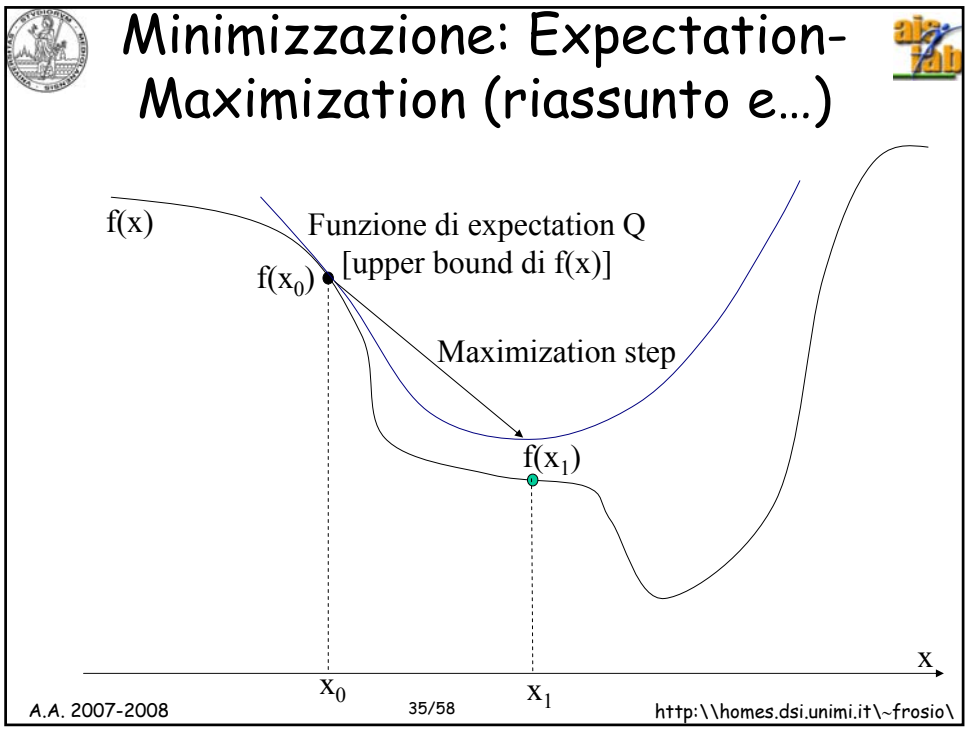
# Minimizzazione: Expectation-Maximization (riassunto)

- La funzione di verosimiglianza può portare ad una minimizzazione complessa;
- Introducendo delle variabili nascoste Z nel modello la formulazione può essere semplificata;
- Il valore delle variabili nascoste Z viene calcolato con il teorema di Bayes nella fase di costruzione della funzione di Expectation;
- Nel passo di Maximization, si massimizza la media su Z della verosimiglianza.



# Minimizzazione: Expectation-Maximization (riassunto e…)

- Derivazione rigorosa di EM:
  - Nel passo di expectation si costruisce un lower bound per la funzione di verosimiglianza (upper bound per -Log(L));
  - Nel passo di maximization si massimizza il lower bound;
- Per approfondire: Cristopher M. Bishop, Pattern Recognition and Machine Learning, Capitolo 9.4.



# Minimizzazione: Expectation-Maximization (riassunto e…)







# Mixture models: introduzione



- La funzione di verosimiglianza permette di stimare i parametri incogniti di una distribuzione (es. media e varianza di una gaussiana);
- Nei casi reali, la densità di probabilità può essere complessa a piacere;
- In particolare, si può considerare una combinazione lineare (mixture) di variabili casuali → **mixture models**

# Mixture models: introduzione

- In un mixture model, la d.d.p. complessiva è la combinazione lineare di M d.d.p. di base.

**Probabilità del dato x nella J-esima distribuzione**

**ddp complessiva**

$$p(\mathbf{x}) = \sum_{j=1}^{M} p_\theta(\mathbf{x} \mid j) P(j)$$

**Probabilità che il dato x provenga dalla J-esima distribuzione**

$$\sum_{j=1}^{M} P(j) = 1;\ 0 <= P(j) <= 1;\ \int p_\theta(\mathbf{x} \mid j) = 1$$



# Mixture models: introduzione

(incognite)

$$p(\mathbf{x}) = \sum_{j=1}^{M} p_\theta(\mathbf{x} \mid j) P(j)$$

**Dati**

La d.d.p. complessiva, p(x), viene misurata.

La forma delle d.d.p. di base, $p_q(x|j)$, viene scelta *a priori* (es. gaussiana).

**Incognite**

I parametri $\theta$ di ogni d.d.p. di base, $p_\theta(x|j)$, devono essere stimati.

Le probabilità per ogni d.d.p. di base, P(j), devono essere stimate.

# Mixture models: introduzione

- Per la stima del vettore dei parametri $\theta$ viene utilizzato…
- … L'approccio alla **massima verosimiglianza**.
- **Mixture model**:
- $p(x) = \Sigma_{j=1..M} p_\theta(x \mid j) \cdot P(j)$
- **Likelihood function**:
- $L = L(\theta) = p(x_1 / \theta) \cdot p(x_2 / \theta) \cdot \ldots p(x_D / \theta)$;
- **Negative log likelihood function**:
- $E = E(\theta) = -\log(L) = -\Sigma_{i=1..D} \log [p(x_i / \theta)] =$
  $= -\Sigma_{i=1..D} \log [\Sigma_{j=1..M} p_\theta(x_i \mid j) P(j)]$

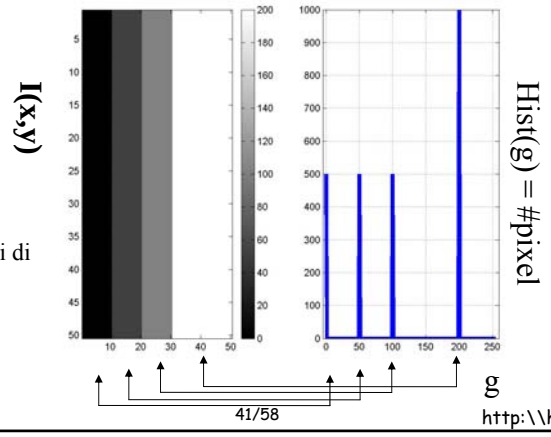


# Mixture models: introduzione (riassunto)

- Mixture models: una combinazione lineare di densità di probabilità utilizzata per descrivere la densità di probabilità degli elementi di un vettore di dati misurato.
- Le incognite sono i parametri di ogni ddp del mixture e la probabilità di ogni singola componente del mixture.
- I parametri vengono calcolati massimizzando la funzione di verosimiglianza.

# Mixture models: segmentazione radiografia cefalometrica

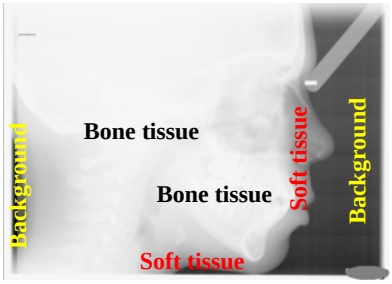
I(x,y) →immagine NRow x NCol, 8 bit (256 livelli di grigio g);
Hist(·) → istogramma, 256 componenti;
Hist(g) → # pixel t.c. I(x,y)=g;
Hist(g) / (NRow * NCol) = p(g). → **ddp**



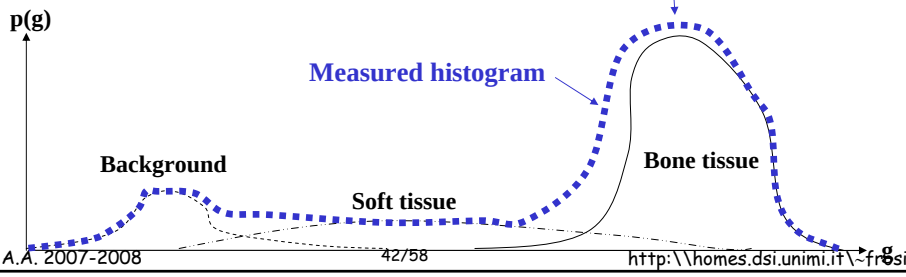

Solo 4 livelli di grigio



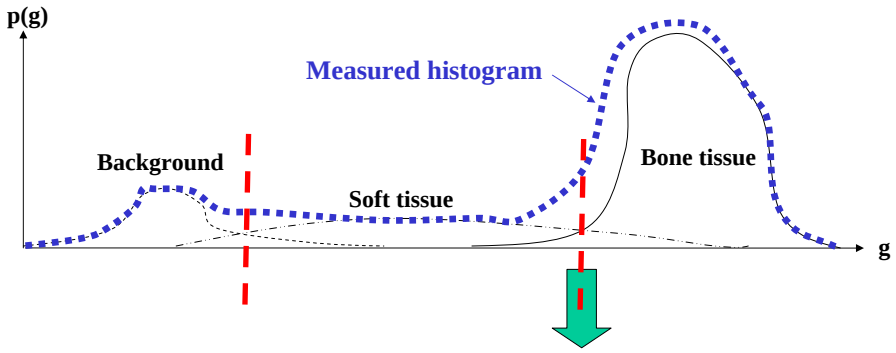
# Mixture models: segmentazione radiografia cefalometrica



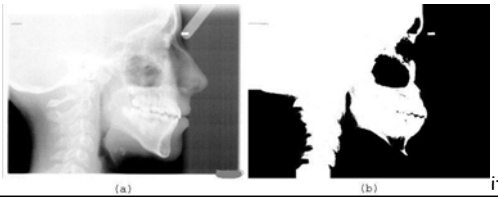

Istogramma tipico immagine radiografica cefalometrica



# Mixture models: segmentazione radiografia cefalometrica



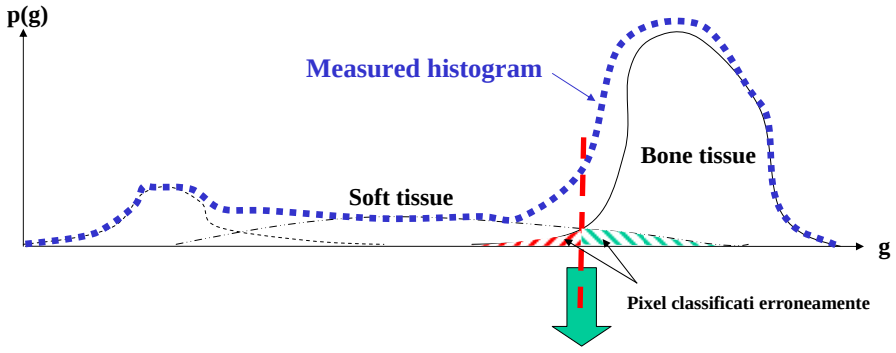

**Livelli di grigio diversi identificano tessuti diversi (soglia → segmentazione).**

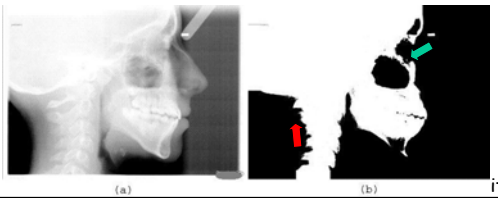


A.A. 2007-2008

it\~frosio\

# Mixture models: segmentazione radiografia cefalometrica



A.A. 2007-2008

it\~frosio\

# Mixture models: segmentazione radiografia cefalometrica

- L'istogramma di una radiografia cefalometrica è composto da tre componenti principali;
- Il problema di segmentazione dell'immagine si presta ad essere trattato con un approccio tipo mixture model;
- Ogni componente del mixture model sarà responsabile rispettivamente della generazione dei dati relativi a background, soft tissue e bone tissue;
- Massimizzazione della verosimiglianza: EM (massimizzazione di lower bound).



# Mixture models: segmentazione radiografia cefalometrica

- Funzione da minimizzare (incognita θ):

$$E = -\ln(L) = -\ln \prod_{n=1}^{N} p_{\vartheta}\left(x^n\right) = -\sum_{n=1}^{N} \ln\, p_{\vartheta}\left(x^n\right)$$

- Per ogni iterazione, i parametri vengono aggiornati (old → new, indice θ omesso):

$$E^{new} - E^{old} = -\sum_{n=1}^{N} \ln\, p^{new}\left(x^n\right) - \left[ -\sum_{n=1}^{N} \ln\, p^{old}\left(x^n\right) \right] = -\sum_{n=1}^{N} \ln \left[ \frac{p^{new}\left(x^n\right)}{p^{old}\left(x^n\right)} \right]$$

# Mixture models: segmentazione radiografia cefalometrica

$$E^{new} - E^{old} = -\sum_{n=1}^{N} \ln p^{new}(x^n) - \left[ -\sum_{n=1}^{N} \ln p^{old}(x^n) \right] = -\sum_{n=1}^{N} \ln \left[ \frac{p^{new}(x^n)}{p^{old}(x^n)} \right]$$

Ricordando che:

$$p(x) = \sum_{j=1}^{M} P(j) \cdot p(x \mid j)$$

Si ottiene:

$$E^{new} - E^{old} = \sum_{n=1}^{N} - \ln \left[ \sum_{j=1}^{M} \frac{P^{new}(j) p^{new}(x^n \mid j)}{p^{old}(x^n)} \cdot \frac{P^{old}(j \mid x^n)}{P^{old}(j \mid x^n)} \right]$$

1



# Mixture models: segmentazione radiografia cefalometrica

- La disuguaglianza di Jensen dice che:

$$dati \quad \lambda_j^2 \quad t.c. \quad \sum_{j=1}^{M} \lambda_j^2 = 1$$

$$\ln\left( \sum_{j=1}^{M} \lambda_j^2 K_j \right) \geq \sum_{j=1}^{M} \lambda_j^2 \ln(K_j) \Rightarrow$$

$$\Rightarrow -\ln\left( \sum_{j=1}^{M} \lambda_j^2 K_j \right) \leq -\sum_{j=1}^{M} \lambda_j^2 \ln(K_j)$$

# Mixture models: segmentazione radiografia cefalometrica

$$\sum_{j=1}^{M} \lambda_j^2 = 1 \quad \longleftrightarrow \quad \sum_{j=1}^{M} P^{old}\left(j \mid x^n\right) = 1, \forall n$$

Jensen — Mixture model

E' possibile applicare la disuguaglianza di Jensen ai mixture model, ove i $P^{old}(j \mid x^n)$ giocano il ruolo dei $\lambda_j^2$.



# Mixture models: segmentazione radiografia cefalometrica

- Applicando Jensen:

$$-\ln\left(\sum_{j=1}^{M} \lambda_j^2 K_j\right) \leq -\sum_{j=1}^{M} \lambda_j^2 \ln\left(K_j\right)$$

$$E^{new} - E^{old} = \sum_{n=1}^{N} -\ln\left[\sum_{j=1}^{M} \frac{P^{new}(j) p^{new}\left(x^n \mid j\right)}{p^{old}\left(x^n\right)} \cdot \frac{P^{old}\left(j \mid x^n\right)}{P^{old}\left(j \mid x^n\right)}\right]$$

# Mixture models: segmentazione radiografia cefalometrica

$$E^{new} - E^{old} = \sum_{n=1}^{N} -\ln\left[\sum_{j=1}^{M} \frac{P^{new}(j)p^{new}(x^n \mid j)}{p^{old}(x^n)} \cdot \frac{P^{old}(j \mid x^n)}{P^{old}(j \mid x^n)}\right]$$

$$\leq -\sum_{j=1}^{M} P^{old}(j \mid x^n) \cdot \ln\left[\frac{P^{new}(j) \cdot p^{new}(x^n \mid j)}{p^{old}(x^n) \cdot P^{old}(j \mid x^n)}\right]$$



# Mixture models: segmentazione radiografia cefalometrica

$$E^{new} - E^{old} \leq -\sum_{n=1}^{N}\sum_{j=1}^{M} P^{old}(j \mid x^n) \ln\left\{\frac{P^{new}(j)p^{new}(x^n \mid j)}{p^{old}(x^n) \cdot P^{old}(j \mid x^n)}\right\}$$

$$E^{new} \leq E^{old} + Q$$

***Minimizzando Q si minimizza E!***

$P^{old}(x^n)$, $P^{old}(j \mid x^n)$ → costanti…

… dunque $Q = Q(\theta^{new})$!

# Mixture models: segmentazione radiografia cefalometrica

Eliminando i termini costanti (old) nella somma (! trasf. Logaritmica!), è sufficiente minimizzare ad ogni iterazione:

$$\widetilde{Q} = -\sum_{n=1}^{N}\sum_{j=1}^{M} P^{old}\left(j \mid x^n\right)\ln\left\{P^{new}(j)\,p^{new}\left(x^n \mid j\right)\right\}$$

Tenendo inoltre conto del fatto che:

$$\sum_{j=1}^{M} P^{new}\left(j \mid x^n\right) = 1, \forall n$$



# Mixture models: segmentazione radiografia cefalometrica

Si minimizza allora (***metodo dei moltiplicatori di Lagrange***):

$$f = \widetilde{Q} + \psi\left(\sum_{j=1}^{M} P^{new}(j) - 1\right)$$

***Vincolo***

Cioè:

$$\begin{cases} \dfrac{\partial f}{\partial \mu_j^{new}} = 0 \\ \dfrac{\partial f}{\partial \sigma_j^{new}} = 0 \\ \dfrac{\partial f}{\partial P^{new}(j)} = 0 \\ \dfrac{\partial f}{\partial \psi} = 0 \quad \text{(vincolo)} \end{cases} \longrightarrow$$

Da questo sistema possono essere ricavate le equazioni per l'aggiornamento dei parametri del mixture model ad ogni iterazione.

# Mixture models: segmentazione radiografia cefalometrica

- Aggiornamento P(j):

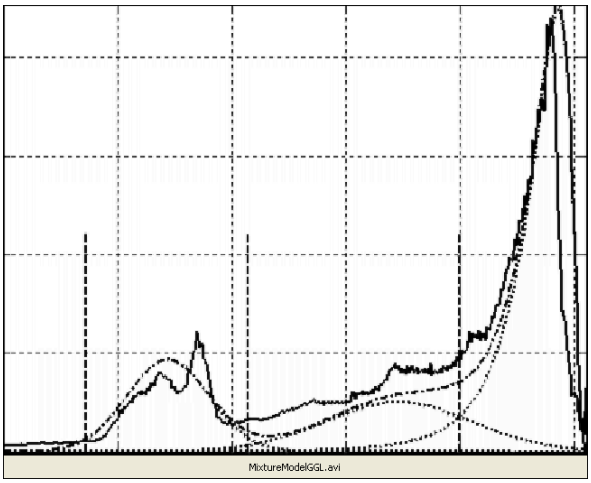
$$P^{new}(j) = \frac{1}{N}\sum_{n=1}^{N} P^{old}(j \mid x^n)$$

- Nel caso di ddp gaussiane:

$$\mu_j^{new} = \frac{\sum_{n=1}^{N} P^{old}(j \mid x^n) x^n}{\sum_{n=1}^{N} P^{old}(j \mid x^n)} \qquad (\sigma_j^{new})^2 = \frac{\sum_{n=1}^{N} P^{old}(j \mid x^n)(x^n - \mu_j)^2}{\sum_{n=1}^{N} P^{old}(j \mid x^n)}$$

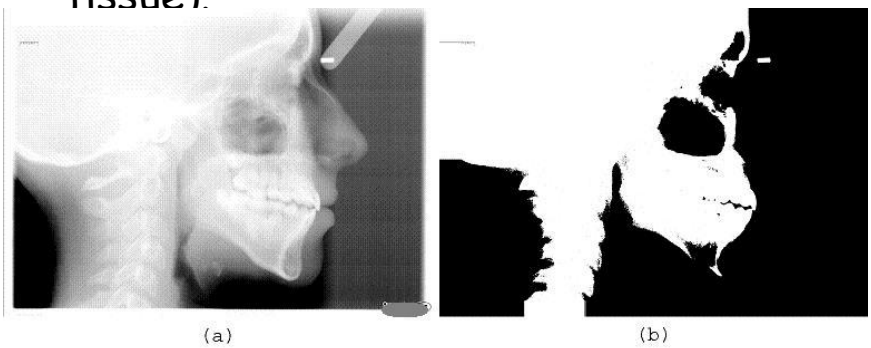


# Mixture models: segmentazione radiografia cefalometrica



# Risultato (bone tissue)

- Clusterizzazione immagine in 3 zone (background, soft tissue, bone tissue).






# Bibliografia


- Cristopher M. Bishop, Pattern Recognition and Machine Learning, Capitolo 2.3.9 (mixture di gaussiane), Capitolo 9.1, 9.2, 9.3 (K-means, mixture models, EM).
- I. Frosio, G. Ferrigno, N. A. Borghese, "Enhancing Digital Cephalic Radiography with Mixture Model and Local Gamma Correction," IEEE Transaction on Medical Imaging, Vol. 25, No. 1, Jan. 2006, pp. 113-121 (mixture models e radiografia cefalometrica).
- *Poul Erik Frandsen, Kristian Jonasson, Hans Bruun Nielsen, Ole Tingleff, Unconstrained Optimization, disponibile in rete, algoritmi di minimizzazione (approfondimento).*
- *Cristopher M. Bishop, Pattern Recognition and Machine Learning, Capitolo 9.4, EM come minimizzazione di un lower bound (approfondimento).*