Expectation Maximization

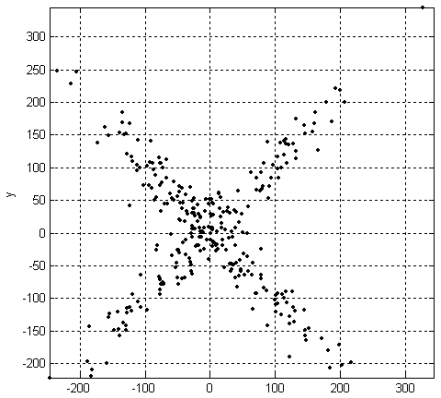
I. Frosio

Overview

- Stima di due rette (riassunto)
- Minimizzazione: metodi "di ordine zero" (simplesso, …)
- Minimizzazione: metodi "di primo ordine" (gradiente, …)
- Minimizzazione: metodi "di secondo ordine" (Newton, …)
- EM: conoscenze preliminari
- EM: derivazione
- EM: applicazione al problema di stima delle due rette

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 σ^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:

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

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

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

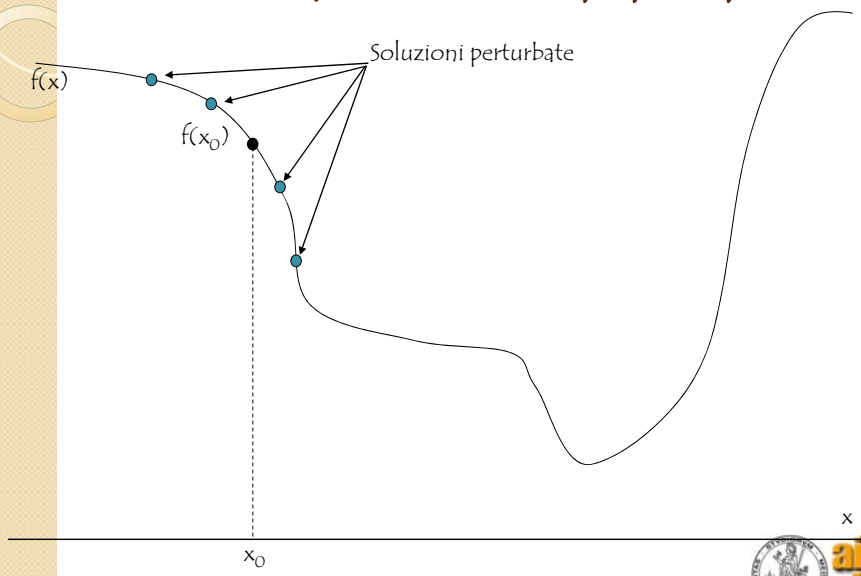
- Non può essere minimizzato analiticamente ponendo le derivate uguali a zero – è necessario usare un algoritmo di minimizzazione iterativo.

Minimizzazione: metodi di ordine 0

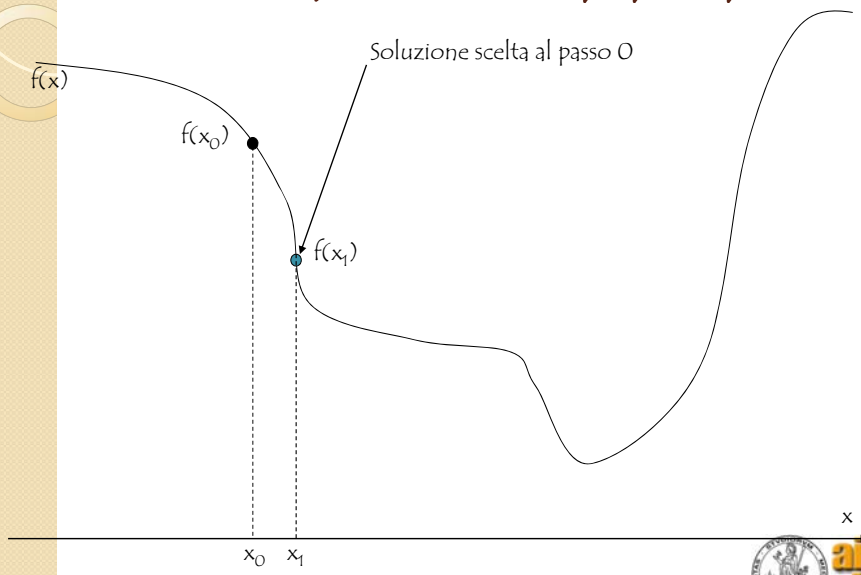
Strategia di base:

- 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 o meno "furbo", testando diverse perturbazioni Δx.
- Per ogni perturbazione Δx si calcola la $f(x_k+\Delta x)$.
- Si sceglie la Δx 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



Minimizzazione: metodi di ordine 0

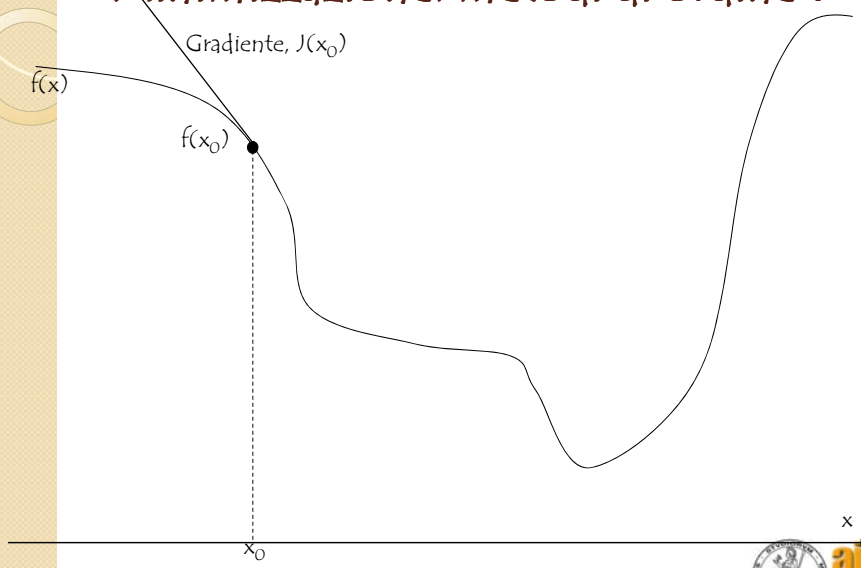


Minimizzazione: metodi di ordine 1

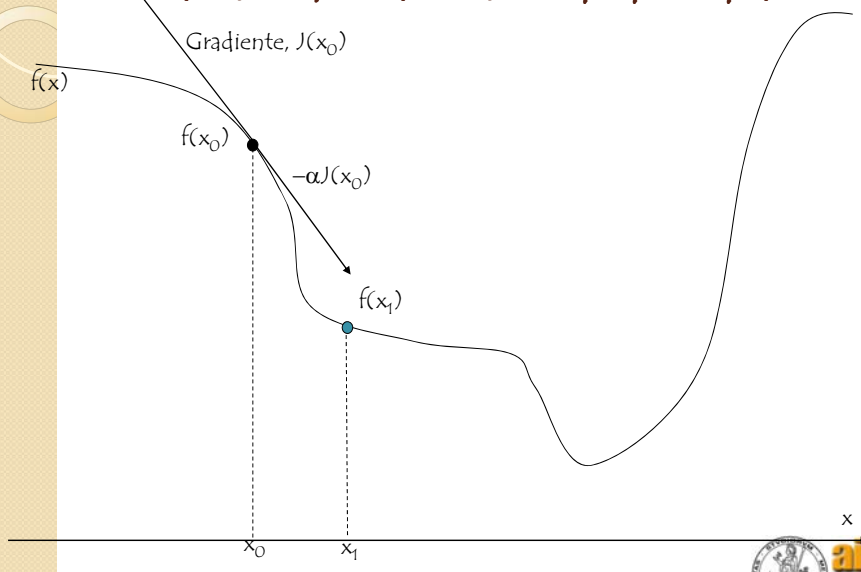
Strategia di base:

- 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 α (learning rate) determina la velocità di convergenza e la stabilità del metodo => Se ad ogni passo α viene in qualche modo ottimizzato, si parla di LINE SEARCH (determinazione della lunghezza ottimale del passo).

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:

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

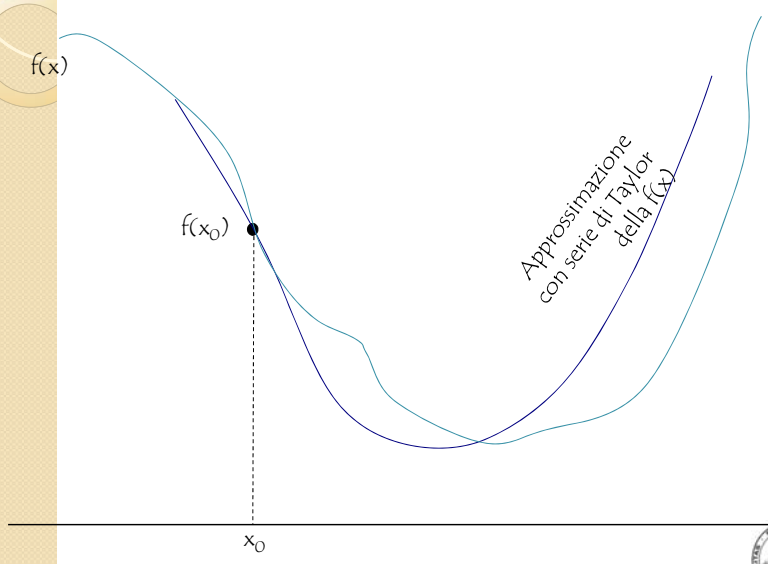
- (Esercizio → implementazione Matlab del metodo del gradiente – confronto con EM per vari valori di $0<\alpha<1$)

Minimizzazione: metodi di ordine 2

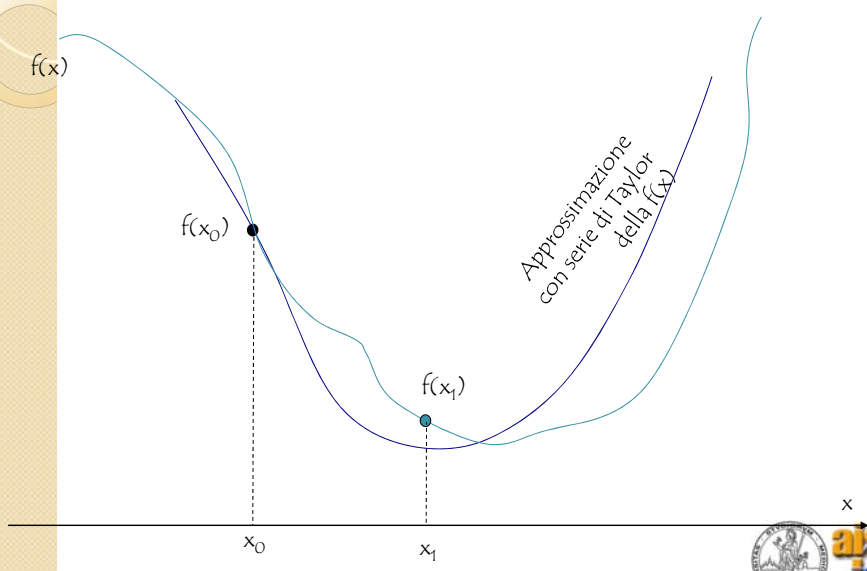
Strategia di base:

- 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):
- $J(x_k)+H(x_k)(\Delta x)=0 \rightarrow \Delta x = -H(x_k)^{-1}J(x_k)$
- E' oneroso calcolare l'hessiano, velocità di convergenza alta.

Minimizzazione: metodi di ordine 2



Minimizzazione: metodi di ordine 2



Minimizzazione – metodi iterativi (riassunto)

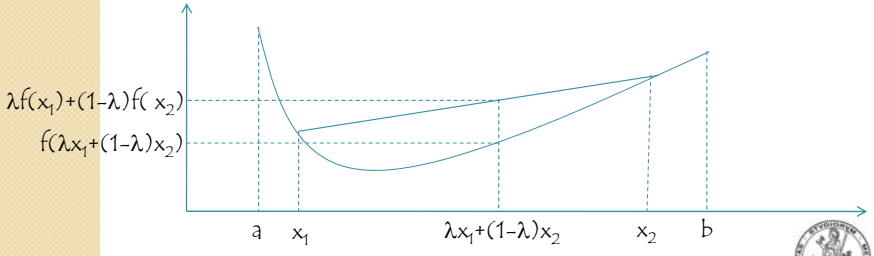
- Quando la funzione da minimizzare è fortemente non lineare, è necessario ricorrere ad un metodo iterativo.
- Si fanno delle ipotesi sull'andamento locale della f(x) e si aggiorna la soluzione x in modo da garantire che la f(x) decresca, fino al raggiungimento di un minimo locale.
- I metodi di ordine 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.

Expectation Maximization (EM)

- Una procedura iterativa ed efficiente per il calcolo della stima alla massima verosimiglianza, nel caso di dati mancanti / nascosti.
- Paragonabile come complessità computazionale ad un metodo di ordine 1, utilizza però una funzione "surrogato" (Expectation) per la massimizzazione della verosimiglianza ad ogni iterazione.
- Ad ogni iterazione:
  - **Expectation step** -> stima dei dati mancanti sulla base dei dati osservati e della stima attuale dei parametri; costruzione della funzione "surrogato" (Expectation);
  - **Maximization step** -> massimizzazione della funzione "surrogato" (e quindi della verosimiglianza) sotto l'ipotesi che i dati mancanti siano noti.

Prima di derivare EM...

- Per la derivazione di EM abbiamo bisogno di richiamare la **nozione di convessità** e di dimostrare la **disuguaglianza di Jensen**;
- Convessità: una funzione $f(x)$ si dice convessa nell'intervallo $[a,b]$ se, $\forall x_1, x_2$ in $[a,b]$, $\forall \lambda \in [0,1]$: $f(\lambda x_1+(1-\lambda)x_2) \leq \lambda f(x_1)+(1-\lambda)f(x_2)$;
- $f''(x)<0$ -> $f(x)$ è concava.
- $d[\ln(x)]/dx = 1/x > 0$ -> $\ln(x)$ è convessa.



Prima di derivare EM...

- Estendendo la nozione di convessità a più di due punti, si dimostra la disuguaglianza di Jensen;

Jensen

- Sia f una funzione convessa definita nell'intervallo [a,b]; siano $x_1, x_2, \ldots, x_N \in [a,b]$; siano $\lambda_1, \lambda_2, \ldots, \lambda_N \geq 0$ tali che $\sum_{i=1..N} \lambda_i = 1$; allora: $f(\sum_{i=1..N} \lambda_i x_i) \leq \sum_{i=1..N} \lambda_i f(x_i)$;

- Dimostrazione per induzione della disuguaglianza di Jensen:
  - 1) dimostriamo che è vera per N=1;
  - 2) dimostriamo che, se è vera per N, allora è vera per N+1.

Prima di derivare EM...

- Sia f una funzione convessa definita nell'intervallo [a,b]; siano $x_1, x_2, \ldots, x_N \in [a,b]$; siano $\lambda_1, \lambda_2, \ldots, \lambda_N \geq 0$ tali che $\sum_{i=1..N} \lambda_i = 1$; allora: $f(\sum_{i=1..N} \lambda_i x_i) \leq \sum_{i=1..N} \lambda_i f(x_i)$;

- Per N=1, $\lambda_1=1$, $f(1 \cdot x_i) \leq 1 \cdot f(x_i)$;
- Per N=2, $f(\lambda x_1 + (1-\lambda) x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$ (direttamente dalla nozione di convessità).

Prima di derivare EM...

- Sia f una funzione convessa definita nell'intervallo [a,b]; siano x_1, x_2, ..., $x_N \in [a,b]$; siano $\lambda_1, \lambda_2, \ldots, \lambda_N \geq 0$ tali che $\sum_{i=1..N} \lambda_i = 1$; allora: $f(\sum_{i=1..N}\lambda_i x_i) \leq \sum_{i=1..N}\lambda_i f(x_i)$;

- Assumiamo che la disuguaglianza sia valida per un certo N, allora:

f convessa
$\lambda_{N+1} + (1-\lambda_{N+1}) = 1$

$$f(\sum_{i=1..N+1} \lambda_i x_i) = f(\lambda_{N+1}x_{N+1} + \sum_{i=1..N} \lambda_i x_i) =$$
$$f(\lambda_{N+1}x_{N+1} + (1-\lambda_{N+1})/(1-\lambda_{N+1})\cdot\sum_{i=1..N} \lambda_i x_i) \leq$$
$$\lambda_{N+1} f(x_{N+1}) + (1-\lambda_{N+1})\cdot f[1/(1-\lambda_{N+1})\cdot\sum_{i=1..N} \lambda_i x_i]$$

Prima di derivare EM...

- Sia f una funzione convessa definita nell'intervallo [a,b]; siano x_1, x_2, ..., $x_N \in [a,b]$; siano $\lambda_1, \lambda_2, \ldots, \lambda_N \geq 0$ tali che $\sum_{i=1..N} \lambda_i = 1$; allora: $f(\sum_{i=1..N}\lambda_i x_i) \leq \sum_{i=1..N}\lambda_i f(x_i)$;

Disug. vera per N
$\sum_{i=1..N} \lambda_i/(1-\lambda_{N+1}) = 1$

$$f(\sum_{i=1..N+1} \lambda_i x_i) \leq$$
$$\lambda_{N+1} f(x_{N+1}) + (1-\lambda_{N+1})\cdot f[1/(1-\lambda_{N+1})\cdot\sum_{i=1..N} \lambda_i x_i] =$$
$$\lambda_{N+1} f(x_{N+1}) + (1-\lambda_{N+1})\cdot f[\sum_{i=1..N} \lambda_i/(1-\lambda_{N+1})\cdot x_i] \leq$$
$$\lambda_{N+1} f(x_{N+1}) + (1-\lambda_{N+1})\cdot\sum_{i=1..N} \lambda_i/(1-\lambda_{N+1})\cdot f(x_i) =$$
$$\lambda_{N+1} f(x_{N+1}) + \sum_{i=1..N} \lambda_i f(x_i) =$$
$$\sum_{i=1..N+1} \lambda_i f(x_i)$$

QED

Prima di derivare EM...

- Possiamo applicare la disugaglianza di Jensen al caso della funzione logaritmica, quindi:

  $\ln(\sum_{i=1..N}\lambda_i x_i) \geq \sum_{i=1..N}\lambda_i \ln(x_i)$;

- Possiamo quindi costruire un minorante per il logaritmo di una somma (att.ne, ln(x) è concavo => -ln(x) è convesso...)
- L'espressione a destra è sempre minore dell'espressione a sinistra – massimizzare a destra implicare massimizzare a sinistra (tale proprietà verrà utilizzata in EM).

Derivazione di EM

- $\mathbf{X}$ -> vettore di variabili casuali;
- $p(\mathbf{X}|\boldsymbol{\theta}_n) = \sum_i p(x_i|\boldsymbol{\theta}_n)$-> probabilità di $\mathbf{X}$ dati i parametri $\boldsymbol{\theta}_n$ (all'n-esima iterazione);
- $f(\boldsymbol{\theta}_n) = \ln[L(\mathbf{X}|\boldsymbol{\theta}_n)]$ -> logaritmo della verosimiglianza;
- Vogliamo trovare una regola di aggiornamento dei parametri tale per cui: $f(\boldsymbol{\theta}_{n+1}) > f(\boldsymbol{\theta}_n)$ =>
  => $\ln[p(\mathbf{X}|\boldsymbol{\theta}_{n+1})] - \ln[p(\mathbf{X}|\boldsymbol{\theta}_n)] > 0$

Derivazione di EM

- In alcuni problemi, esistono alcune variabili nascoste (non osservate) – EM fornisce un framework naturale per la loro incusione;
- In altri casi, l'introduzione di alcune variabili nascoste "arbitrarie" permette di semplificare la formulazione matematica del problema.
- Consideriamo quindi un vettore di variabili nascoste **Z** e riscriviamo la probabilità di misurare i dati **X** considerando le variabili nascoste **Z**:

$$p(\mathbf{X}|\boldsymbol{\theta}_n) = \sum_z [p(\mathbf{X}|z,\boldsymbol{\theta}_n) \cdot p(z|\boldsymbol{\theta}_n)] \ ^*$$

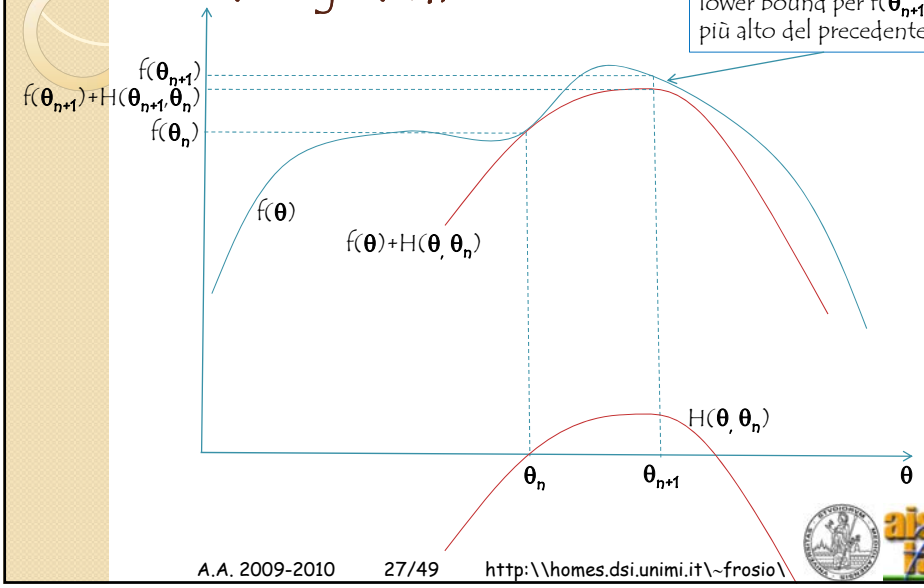
* Per calcolare la probabilità di un dato x, devo considerare tutti i possibili valori che può assumere z...

Derivazione di EM

- Utilizzando le variabili nascoste, vogliamo trovare una funzione $H(\boldsymbol{\theta}_{n+1},\boldsymbol{\theta}_n)$ t.c. $f(\boldsymbol{\theta}_{n+1}) \geq f(\boldsymbol{\theta}_n) + H(\boldsymbol{\theta}_{n+1},\boldsymbol{\theta}_n)$ e $H(\boldsymbol{\theta}_n,\boldsymbol{\theta}_n) = 0$.
- All'iterazione n-esima, $\boldsymbol{\theta}_n$ è fissato, mentre $\boldsymbol{\theta}_{n+1}$ è incognito.
- Massimizzando $H(\boldsymbol{\theta}_{n+1},\boldsymbol{\theta}_n)$, si massimizza la $f(\boldsymbol{\theta}_{n+1})$!

Derivazione di EM (e convergenza)

Massimizzando $H(\theta_{n+1},\theta_n)$ rispetto a θ_{n+1}, otteniamo un nuovo lower bound per $f(\theta_{n+1})$ più alto del precedente!



Derivazione di EM

- Ad ogni iterazione la f deve crescere [$f(\theta_{n+1}) > f(\theta_n)$]...
- $f(\theta_{n+1}) - f(\theta_n) =$
  $\ln\{\sum_z [p(X|z,\theta_{n+1})\cdot p(z|\theta_{n+1})]\} - \ln[p(X|\theta_n)] =$
  $\ln\{\sum_z [p(X|z,\theta_{n+1})\cdot p(z|\theta_{n+1}) \cdot p(z|X,\theta_n)/p(z|X,\theta_n)]\} - \ln[p(X|\theta_n)] =$
  $\ln\{\sum_z [p(z|X,\theta_n) \cdot p(X|z,\theta_{n+1})\cdot p(z|\theta_{n+1}) /p(z|X,\theta_n)]\} - \ln[p(X|\theta_n)] \geq ...$

  [con Jensen, $\lambda_z = p(z|X,\theta_n)$, $\sum_z \lambda_z = 1$, $x_z = p(X|z,\theta_{n+1})\cdot p(z|\theta_{n+1}) /p(z|X,\theta_n)$]

  $... \geq \sum_z \{p(z|X,\theta_n)\cdot \ln[p(X|z,\theta_{n+1})\cdot p(z|\theta_{n+1}) /p(z|X,\theta_n)]\} - \ln[p(X|\theta_n)] =$
  $\sum_z \{p(z|X,\theta_n)\cdot \ln[p(X|z,\theta_{n+1})\cdot p(z|\theta_{n+1})]\} - \sum_z \{p(z|X,\theta_n)\cdot \ln[p(z|X,\theta_n)]\} - \ln[p(X|\theta_n)]$

  $f(\theta_{n+1}) - f(\theta_n) \geq$
  $\sum_z \{p(z|X,\theta_n)\cdot \ln[p(X|z,\theta_{n+1})\cdot p(z|\theta_{n+1})]\} - \sum_z \{p(z|X,\theta_n)\cdot \ln[p(z|X,\theta_n)]\} - \ln[p(X|\theta_n)]$

Derivazione di EM

Abbiamo quindi trovato:

$H(\theta_n, \theta_{n+1}) = \sum_z \{p(z|X,\theta_n) \cdot \ln[p(X|z,\theta_{n+1}) \cdot p(z|\theta_{n+1})]\} - \sum_z \{p(z|X,\theta_n) \cdot \ln[p(z|X,\theta_n)]\} - \ln[p(X|\theta_n)]$

Verifichiamo che $H(\theta_n, \theta_n) = 0$...

$H(\theta_n,\theta_n) = \sum_z \{p(z|X,\theta_n) \cdot \ln[p(X|z,\theta_n) \cdot p(z|\theta_n)]\} - \sum_z \{p(z|X,\theta_n) \cdot \ln[p(z|X,\theta_n)]\} - \ln[p(X|\theta_n)]$

$\ln[p(X|\theta_n)] = \underbrace{\sum_z [p(z|X,\theta_n)]}_{=1} \cdot \ln[p(X|\theta_n)] = \sum_z \{p(z|X,\theta_n) \cdot \ln[p(X|\theta_n)]\}$

$H(\theta_n, \theta_n) = \sum_z (p(z|X,\theta_n) \cdot \ln[p(X|z,\theta_n) \cdot p(z|\theta_n) / [p(z|X,\theta_n) \cdot p(X|\theta_n)]]) = \ldots$

[ricordando che $p(X,z|\theta_n) = p(X|z,\theta_n) \cdot p(z|\theta_n) = p(z|X,\theta_n) \cdot p(X|\theta_n)$]

$\ldots = \sum_z \{p(z|X,\theta_n) \cdot \ln[p(X,z|\theta_n) / p(X,z|\theta_n)]\} = \sum_z \{p(z|X,\theta_n) \cdot \ln(1)\} = 0$ QED

Derivazione di EM

- Per derivare il passo di aggiornamento dei parametri, dobbiamo dunque massimizzare $H(\theta_{n+1},\theta_n)$ rispetto a θ_{n+1}.
- $\theta_{n+1} = \text{argmax}_\theta H(\theta,\theta_n)$.
- Nella massimizzazione, possiamo evitare di considerare i termini di $H(\theta,\theta_n)$ non dipendenti da θ [Q = H + cost].

$H(\theta_n, \theta_{n+1}) = Q(\theta,\theta_n) + \text{cost} =$
$\sum_z \{p(z|X,\theta_n) \cdot \ln[p(X|z,\theta_{n+1}) \cdot p(z|\theta_{n+1})]\} =$ [illegible]

$\theta_{n+1} = \text{argmax}_\theta \sum_z \{p(z|X,\theta_n) \cdot \ln[p(X|z,\theta_{n+1}) \cdot p(z|\theta_{n+1})]\} \Rightarrow$
$\theta_{n+1} = \text{argmax}_\theta \sum_z \{p(z|X,\theta_n) \cdot \ln[p(X,z|,\theta_{n+1})]\}$

Derivazione di EM

- Passo di aggiornamento:

$\theta_{n+1} = \text{argmax}_\theta \sum_z \{p(z|X,\theta_n) \cdot \ln[p(X,z|,\theta_{n+1})]\}$

1) Calcola il valore atteso (Expectation) di $\ln[p(X,z|,\theta_{n+1})$, utilizzando le probabilità delle veriabili nascoste calcolate al passo precedente $[p(z|X,\theta_n)]$.
2) Massimizza il valore atteso rispetto ai nuovi parametri θ_{n+1} (Maximization).

Cosa abbiamo guadagnato?

- Dalla problema di massimizzazione di $f(\theta)$ siamo passati al problema di massimizzazione di $Q(\theta)$.
- $Q(\theta)$ tiene in considerazione le variabili nascoste che non compaiono in $f(\theta)$.
- Se scelte opportunamente, le variabili nascoste portano ad una formulazione di $Q(\theta)$ che è più semplicemente ottimizzabile rispetto alla massimizzazione di $f(\theta)$.
- EM permette inoltre di stimare il valore delle variabili nascoste.

EM per la stima di 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.

- 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.

EM per la stima di due rette: funzione Q

- Dobbiamo **massimizzare la media su Z del logaritmo della funzione di verosimiglianza, condizionata a Z**:

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

Maximization

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

Expectation

EM per la stima di due rette: funzione Q

- Assumendo come variabile nascosta Z il fatto che cascun punto yi sia generato dall'una o dall'altra retta, otteniamo :

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

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

Costante [K(yi,j)] da calcolare nella fase di Expectation

EM per la stima di due rette: Expectation

- 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 θ^{old}, possiamo stimare le probabilità delle Z, $p(Z|Y, \theta^{old})$.

EM per la stima di due rette: Expectation

- Stimiamo le variabili nascoste utilizzando il 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

EM per la stima di due rette: Expectation

- 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.

EM per la stima di due rette: 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**:

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

Maximization

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

Expectation

EM per la stima di due rette: Maximization

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

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

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

EM per la stima di due rette: Maximization

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

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

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

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

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

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

EM per la stima di due rette: 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.

EM per la stima di due rette: Maximization

- Analizzando la funzione di aggiornamento per EM…

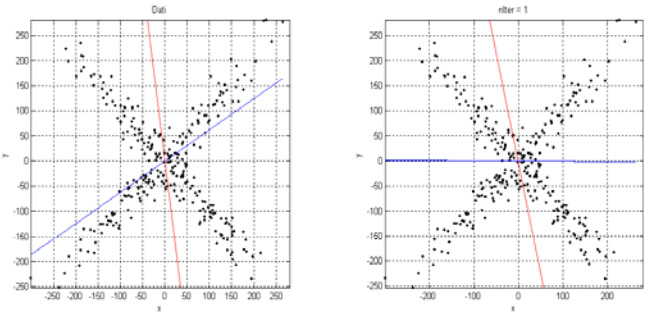
$$mj = \frac{\sum_{i=1}^{N} k(yi, j)\cdot(yi \cdot xi)}{\sum_{i=1}^{N} k(yi, j)\cdot(xi^2)}$$

- 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

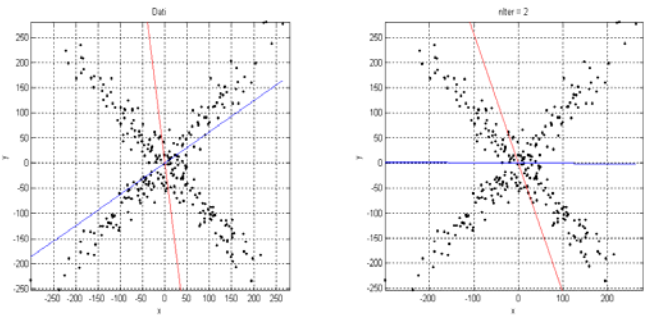
EM per la stima di due rette: interpretazione

- 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…
- Ricorda qualcosa??? **K-Means può essere interpretato come un EM "al limite"!!!**
- Nell'EM reale l'assegnazione è "soft" invece che "hard".
- Vedere anche → codice Matlab

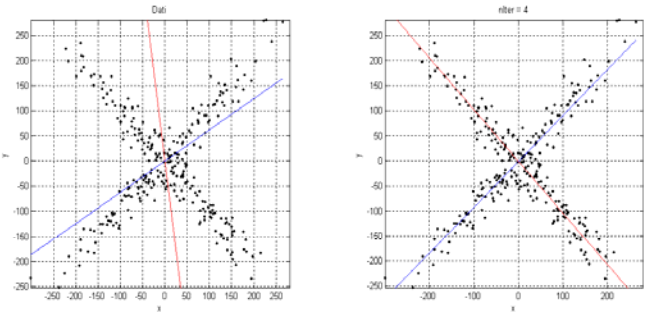
EM per la stima di due rette: esempio



EM per la stima di due rette: esempio



EM per la stima di due rette: esempio



EM - 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 ad esempio calcolato con il teorema di Bayes nella fase di costruzione della funzione di Expectation;
- Nel passo di Maximization, si massimizza la media su Z del logaritmo della verosimiglianza.

Riferimenti

- Derivazione di EM per lo più ispirata a:
  Sean Borman, The Expectation Maximization Algorithm, A short tutorial (disponibile free in rete).

- Riferimento alternativo o per approfondimenti: Cristopher M. Bishop, Pattern Recognition and Machine Learning, Capitolo 9.4.