

Dipartimento di Informatica

Università di Verona

A.A. 2014-15

# **Teoria e Tecniche del Riconoscimento**

**Stima dei parametri:  
approccio Maximum Likelihood,  
approccio Bayesiano,  
Expectation-Maximization**

# Introduzione

- Per creare un classificatore ottimale che utilizzi la regola di decisione Bayesiana è necessario conoscere:
  - Le *probabilità a priori*  $P(\omega_i)$
  - Le *densità condizionali*  $p(\mathbf{x} | \omega_i)$
- Le performance di un classificatore dipendono fortemente dalla bontà di queste componenti
- ***NON SI HANNO PRATICAMENTE MAI TUTTE QUESTE INFORMAZIONI!***

- Più spesso, si hanno unicamente:
  - Una *vaga conoscenza del problema*, da cui estrarre vaghe probabilità a priori.
  - *Alcuni pattern particolarmente rappresentativi, training data*, usati per *addestrare* il classificatore (spesso troppo pochi!)
- La stima delle probabilità a priori di solito non risulta particolarmente difficoltosa.
- La stima delle densità condizionali è più complessa.

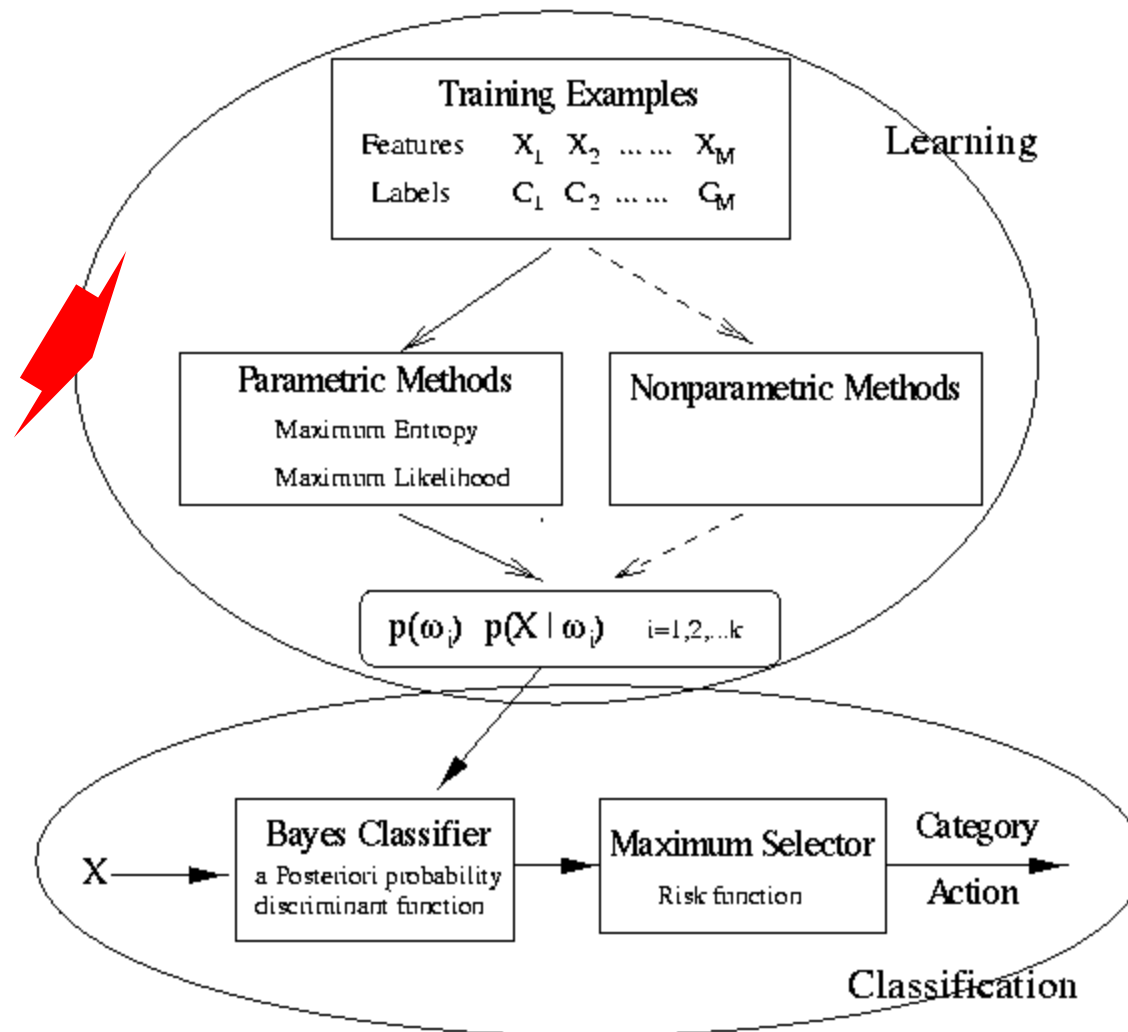
- Assunto che la conoscenza, benché approssimativa, delle densità a priori non presenta problemi, per quanto riguarda le densità condizionali le problematiche si possono suddividere in:
  1. *Stimare la funzione sconosciuta*  $p(\mathbf{x} | \omega_j)$
  2. *Stimare i parametri sconosciuti della funzione conosciuta*  $p(\mathbf{x} | \omega_j)$

Per es., stimare il vettore  $\boldsymbol{\theta}_j = (\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$  se  
 $p(\mathbf{x} | \omega_j) \approx N(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$

# Stima dei parametri

- Il secondo punto risulta di gran lunga più semplice (sebbene complesso!), e rappresenta un problema classico nella statistica.
- Trasferito nella *pattern recognition*, un approccio è quello di
  - 1) stimare i parametri dai dati di training
  - 2) usare le stime risultanti come se fossero valori veri
  - 3) utilizzare infine la teoria di decisione Bayesiana per usare un classificatore

# Uno sguardo d'insieme



# Stima dei parametri – Probabilità a priori

- Supponiamo di avere un insieme di  $n$  dati di training in cui ad ogni pattern è assegnata un'etichetta d'identità (ossia conosco per certo a quale stato  $\omega_j$  appartiene il pattern  $k$ -esimo)
- ➔ *problema di learning dei parametri supervisionato*

- Allora

$$P(\omega_i) = \frac{n_i}{n}$$

dove  $n_i$  è il numero di campioni con etichetta  $\omega_i$

# Stima dei parametri – Class conditional

- Supponiamo di avere  $c$  set di campioni  $D_1, D_2, \dots, D_c$  tracciati indipendentemente in accordo alla densità  $p(x|\omega_j)$ 
  - Assumiamo che  $p(x|\omega_j)$  abbia forma parametrica conosciuta
- Il problema di stima dei parametri consiste nello stimare i parametri che definiscono  $p(x|\omega_j)$
- Per semplificare il problema, assumiamo inoltre che:
  - i campioni appartenenti al set  $D_i$  non danno informazioni relative ai parametri di  $p(x|\omega_j)$  se  $i \neq j$ .



# Stima dei parametri – Due approcci

- Specificatamente, il problema può essere formulato come:
  - Dato un set di training  $D = \{x_1, x_2, \dots, x_n\}$
  - $p(x|\omega)$  è determinata da  $\theta$ , che è un vettore rappresentante i parametri necessari  
(p.e.,  $\theta = (\mu, \Sigma)$  se  $p(\mathbf{x} | \omega) \approx N(\mu, \Sigma)$  )
  - Vogliamo trovare il migliore  $\theta$  usando il set di training.
- Esistono due approcci
  - Stima **Maximum-likelihood (ML)**
  - Stima **parametrica Bayesiana**

# Stima dei parametri – Due approcci (2)

- Approccio Maximum Likelihood
  - I parametri sono *quantità fissate* ma sconosciute
  - La migliore stima dei loro valori è quella che *massimizza la probabilità di ottenere i dati di training*
- Approccio Bayesiano
  - I parametri sono *variabili aleatorie* aventi determinate probabilità a priori
  - Le osservazioni dei dati di training trasformano queste probabilità in probabilità a posteriori

# Stima dei parametri – Due approcci (3)

- Aggiungendo campioni di training il risultato è di rifinire meglio la forma delle densità a posteriori, causando un innalzamento di esse in corrispondenza dei veri valori dei parametri (fenomeno di *Bayesian Learning*).
- I risultati dei due approcci, benché proceduralmente diversi, sono qualitativamente simili.

# Approccio *Maximum Likelihood*

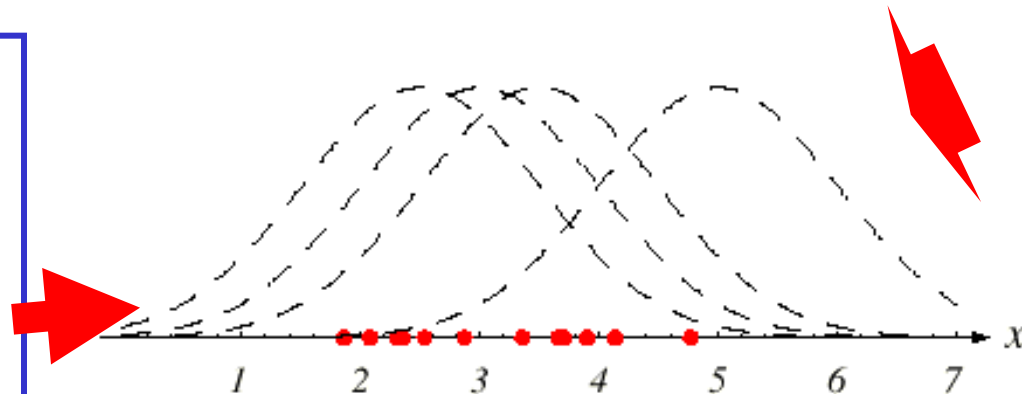
- In forza dell'ipotesi di partenza del problema, poiché i pattern del set  $\mathbf{D}$  sono i.i.d., abbiamo che:

$$p(\mathbf{D} | \boldsymbol{\theta}) = \prod_{k=1}^n p(x_k | \boldsymbol{\theta})$$

- Vista come funzione di  $\boldsymbol{\theta}$ ,  $p(\mathbf{D}|\boldsymbol{\theta})$  viene chiamata *likelihood* di  $\boldsymbol{\theta}$  rispetto al set di campioni  $\mathbf{D}$ .
- La stima di Maximum Likelihood di  $\boldsymbol{\theta}$  è, per definizione, il valore  $\hat{\boldsymbol{\theta}}$  che massimizza  $p(\mathbf{D}|\boldsymbol{\theta})$ ;
- Ricordiamo l'assunzione che  $\boldsymbol{\theta}$  è fissato ma sconosciuto

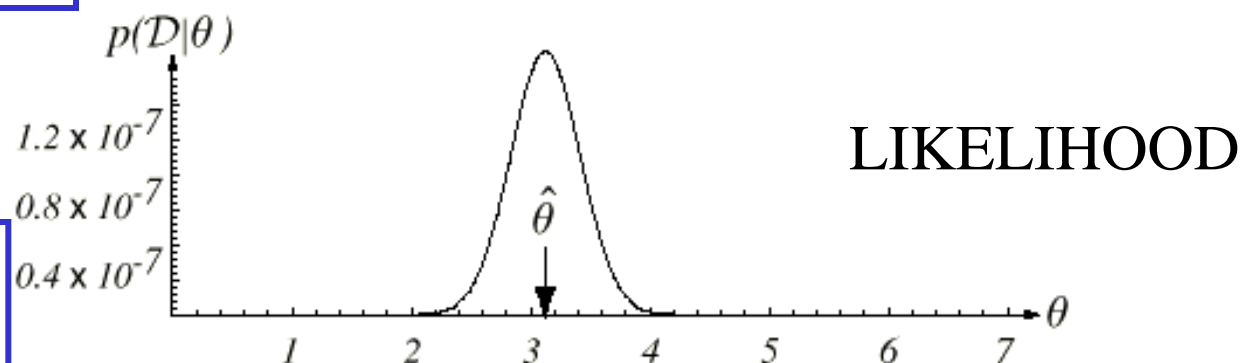
# Approccio Maximum Likelihood (2)

Punti di training 1-D  
assunti generati da una  
densità gaussiana di  
varianza fissata ma  
media sconosciuta

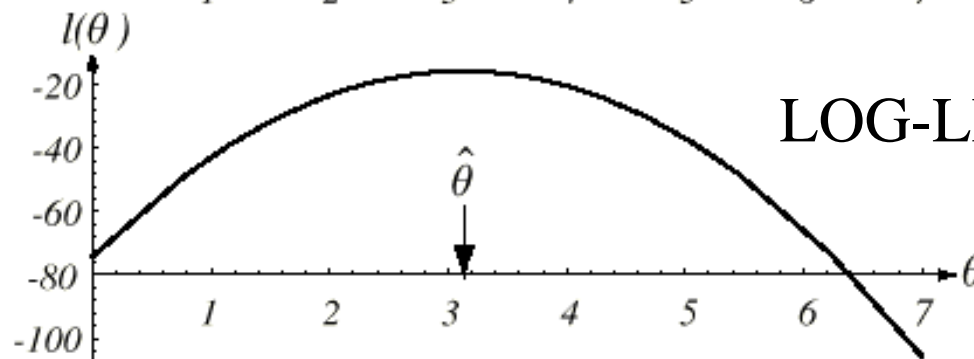


4 delle  
infinite  
possibili  
gaussiane

**NB: La likelihood**  
 **$p(D|\theta)$  è funzione di**  
 **$\theta$ , mentre la densità**  
**condizionale  $p(x|\theta)$**   
**funzione di  $x$**



LIKELIHOOD



LOG-LIKELIHOOD

# Approccio Maximum Likelihood (3)

- Se il numero di parametri da stimare è  $p$ , sia  $\boldsymbol{\theta} = (\theta_1 \dots \theta_p)^t$  e

$$\nabla \boldsymbol{\theta} \equiv \begin{bmatrix} \frac{\partial}{\partial \theta_1} \\ \vdots \\ \frac{\partial}{\partial \theta_p} \end{bmatrix}$$

- Per scopi analitici risulta più semplice lavorare con il logaritmo della likelihood.
- Definiamo quindi  $l(\boldsymbol{\theta})$  come *funzione di log-likelihood*

$$l(\boldsymbol{\theta}) \equiv \ln p(D \mid \boldsymbol{\theta}) = \sum_{k=1}^n \ln p(x_k \mid \boldsymbol{\theta})$$

# Approccio Maximum Likelihood (4)

- Lo scopo è di ottenere quindi il vettore

$$\hat{\boldsymbol{\theta}} = \arg \max_{\boldsymbol{\theta}} l(\boldsymbol{\theta})$$

in cui la dipendenza sul data set  $\mathbf{D}$  è implicita.

- Pertanto per ricavare il max:

$$l(\boldsymbol{\theta}) \equiv \ln p(\mathbf{D} | \boldsymbol{\theta}) = \sum_{k=1}^n \ln p(x_k | \boldsymbol{\theta})$$



$$\nabla_{\boldsymbol{\theta}} l(\boldsymbol{\theta}) = \sum_{k=1}^n \nabla_{\boldsymbol{\theta}} \ln p(x_k | \boldsymbol{\theta})$$

da cui vogliamo ottenere  $\nabla_{\boldsymbol{\theta}} l(\boldsymbol{\theta}) = 0$

# Approccio Maximum Likelihood (5)

- Formalmente, una volta trovato il set di parametri che rende vera, è necessario controllare che la soluzione trovata sia effettivamente un massimo globale, piuttosto che un massimo locale o un flesso o peggio ancora un punto di minimo.
- Bisogna anche controllare cosa accade ai bordi degli estremi dello spazio dei parametri
- Applichiamo ora l'approccio ML ad alcuni casi specifici.



# Maximum Likelihood: caso Gaussiano

- Consideriamo che i campioni siano generati da una popolazione normale multivariata di media  $\boldsymbol{\mu}$  e covarianza  $\boldsymbol{\Sigma}$ .
- Per semplicità, consideriamo il caso in cui solo la media  $\boldsymbol{\mu}$  sia sconosciuta. Consideriamo quindi il punto campione  $\mathbf{x}_k$  e troviamo:

$$\ln p(\mathbf{x}_k | \boldsymbol{\mu}) = -\frac{1}{2} \ln[(2\pi)^d |\boldsymbol{\Sigma}|] - \frac{1}{2} (\mathbf{x}_k - \boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1} (\mathbf{x}_k - \boldsymbol{\mu})$$



$$\nabla_{\boldsymbol{\mu}} \ln p(\mathbf{x}_k | \boldsymbol{\mu}) = \boldsymbol{\Sigma}^{-1} (\mathbf{x}_k - \boldsymbol{\mu})$$

## Maximum Likelihood: caso Gaussiano (2)

- Identificando  $\boldsymbol{\theta}$  con  $\boldsymbol{\mu}$  si deduce che la stima Maximum-Likelihood di  $\boldsymbol{\mu}$  deve soddisfare la relazione:

$$\sum_{k=1}^n \boldsymbol{\Sigma}^{-1} (\mathbf{x}_k - \hat{\boldsymbol{\mu}}) = 0$$

- Moltiplicando per  $\boldsymbol{\Sigma}$  e riorganizzando la somma otteniamo

$$\hat{\boldsymbol{\mu}} = \frac{1}{n} \sum_{k=1}^n \mathbf{x}_k$$

che non è altro che la semplice *media* degli esempi di training, altresì indicata con  $\hat{\boldsymbol{\mu}}_n$  per indicarne la dipendenza dalla numerosità del training set.

## Maximum Likelihood: caso Gaussiano (3)

- Consideriamo ora il caso più tipico in cui la distribuzione Gaussiana abbia media e covarianza ignote.
- Consideriamo prima il caso univariato  $\boldsymbol{\theta} = (\theta_1, \theta_2) = (\mu, \sigma^2)$
- Se si prende un singolo punto abbiamo

$$\ln p(x_k | \boldsymbol{\theta}) = -\frac{1}{2} \ln[2\pi\theta_2] - \frac{1}{2\theta_2} (x_k - \theta_1)^2$$

la cui derivata è

$$\nabla_{\boldsymbol{\theta}} l = \nabla_{\boldsymbol{\theta}} \ln p(x_k | \boldsymbol{\theta}) = \begin{bmatrix} \frac{1}{\theta_2} (x_k - \theta_1) \\ -\frac{1}{2\theta_2} + \frac{(x_k - \theta_1)^2}{2\theta_2^2} \end{bmatrix}$$

## Maximum Likelihood: caso Gaussiano (4)

- Eguagliando a 0 e considerando tutti i punti si ottiene:

$$\sum_{k=1}^n \frac{1}{\hat{\theta}_2} (x_k - \hat{\theta}_1) = 0 \quad - \sum_{k=1}^n \frac{1}{\hat{\theta}_2} + \sum_{k=1}^n \frac{(x_k - \hat{\theta}_1)^2}{\hat{\theta}_2^2} = 0$$

dove  $\hat{\theta}_1$  e  $\hat{\theta}_2$  sono le stime ML per  $\theta_1$  e  $\theta_2$ .

- Sostituendo  $\hat{\mu} = \hat{\theta}_1$  e  $\sigma^2 = \hat{\theta}_2$  si hanno le stime ML di media e varianza

$$\hat{\mu} = \frac{1}{n} \sum_{k=1}^n x_k \quad \hat{\sigma}^2 = \frac{1}{n} \sum_{k=1}^n (x_k - \hat{\mu})^2$$

## Maximum Likelihood: caso Gaussiano (5)

- Il caso multivariato si tratta in maniera analoga con più conti. Il risultato è comunque:

$$\hat{\boldsymbol{\mu}} = \frac{1}{n} \sum_{k=1}^n \mathbf{x}_k \qquad \hat{\boldsymbol{\Sigma}} = \frac{1}{n} \sum_{k=1}^n (\mathbf{x}_k - \hat{\boldsymbol{\mu}})(\mathbf{x}_k - \hat{\boldsymbol{\mu}})^t$$

- Si noti tuttavia che la stima della covarianza è sbilanciata, i.e., il valore aspettato della varianza campione su tutti i possibili insiemi di dimensione  $n$  non è uguale alla vera varianza

$$E\left\{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2\right\} = \frac{n-1}{n} \sigma^2 \neq \sigma^2$$

# Maximum-Likelihood: altri casi

- Esistono, oltre alla densità Gaussiana, anche altre famiglie di densità che costituiscono altrettante famiglie di parametri:

- *Distribuzione esponenziale*

$$p(x | \theta) = \begin{cases} \theta e^{-\theta x} & x \geq 0 \\ 0 & \text{altrimenti} \end{cases}$$

- *Distribuzione uniforme*

$$p(x | \theta) = \begin{cases} 1/\theta & 0 \leq x \leq \theta \\ 0 & \text{altrimenti} \end{cases}$$

- *Distribuzione di Bernoulli multivariata*

# Maximum-Likelihood – Modello d'errore

- In generale, se i modelli parametrici sono validi, il classificatore *maximum-likelihood* fornisce risultati eccellenti.
- Invece, se si usano famiglie parametriche scorrette, il classificatore produce forti errori
  - Questo accade anche se è nota la famiglia parametrica da usare, per esempio se si stima all'interno di una distribuzione gaussiana come parametro una varianza troppo larga.

## Maximum-Likelihood – Modello d'errore (2)

- Di fatto *manca un modello d'errore che dia un voto alla parametrizzazione ottenuta.*
- Inoltre, per applicare la stima di Maximum-Likelihood, tutti i dati di training devono essere disponibili
  - Se vogliamo utilizzare nuovi dati di training, è necessario ricalcolare la procedura di stima Maximum-Likelihood.



# Stima di Bayes

- A differenza dell'approccio ML, in cui supponiamo  $\theta$  come fissato ma sconosciuto, *l'approccio di stima Bayesiana* dei parametri considera  $\theta$  come **una variabile aleatoria**.
- In questo caso il set di dati di training  $D$  ci permette di *convertire una distribuzione a priori*  $p(\theta)$  *su questa variabile in una densità di probabilità a posteriori*  $p(\theta|D)$

$$p(\theta) \quad \rightarrow \quad p(\theta|D)$$

- Data la difficoltà dell'argomento, è necessario un passo indietro al concetto di classificazione Bayesiana

# Approccio di stima Bayesiano – Idea centrale

- Il calcolo delle densità a posteriori  $P(\omega_i|\mathbf{x})$  sta alla base della classificazione Bayesiana
- Per creare un classificatore ottimale che utilizzi la regola di decisione Bayesiana è necessario conoscere:
  - Le *probabilità a priori*  $P(\omega_i)$
  - Le *densità condizionali*  $p(\mathbf{x}|\omega_i)$
- Quando queste quantità sono sconosciute, bisogna ricorrere a tutte le informazioni a disposizione.

# Approccio di stima Bayesiano – Idea centrale (2)

- Parte di queste informazioni può essere derivante da:
  - 1. Conoscenza a priori**
    - *Forma funzionale delle densità sconosciute*
    - *Intervallo dei valori dei parametri sconosciuti*
  - 2. Training set**
    - Sia  $D$  il *set totale di campioni*: il nostro compito si trasforma così nella stima di  $P(\omega_i | \mathbf{x}, D)$
- Da queste probabilità possiamo ottenere il classificatore Bayesiano.

## Approccio di stima Bayesiano – Idea centrale (3)

- Dato il set di training  $D$ , la formula di Bayes diventa:

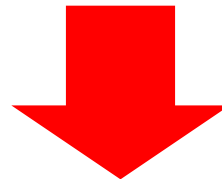
$$P(\omega_i | \mathbf{x}, D) = \frac{p(\mathbf{x} | \omega_i, D)P(\omega_i | D)}{\sum_{j=1}^c p(\mathbf{x} | \omega_j, D)P(\omega_j | D)}$$

- Assunzioni:
  - Ragionevolmente,  $P(\omega_i | D) \Rightarrow P(\omega_i)$
  - Dato il caso di learning supervisionato il set  $D$  è partizionato in  $c$  set di campioni  $D_1, D_2, \dots, D_c$  con  $i$  campioni in  $D_i$  appartenenti a  $\omega_i$
  - I campioni appartenenti al set  $D_i$  non danno informazioni sui parametri di  $p(\mathbf{x} | \omega_j, D)$  se  $i \neq j$ .

# Approccio di stima Bayesiano – Idea centrale (4)

- Queste assunzioni portano a due conseguenze:
  1. Possiamo lavorare con ogni classe indipendentemente, ossia

$$P(\omega_i | \mathbf{x}, D) = \frac{p(\mathbf{x} | \omega_i, D) P(\omega_i | D)}{\sum_{j=1}^c p(\mathbf{x} | \omega_j, D) P(\omega_j | D)}$$



$$P(\omega_i | \mathbf{x}, D) = \frac{p(\mathbf{x} | \omega_i, D_i) P(\omega_i)}{\sum_{j=1}^c p(\mathbf{x} | \omega_j, D_j) P(\omega_j)}$$

# Approccio di stima Bayesiano – Idea centrale (5)

2. Poiché ogni classe può essere trattata indipendentemente, si possono evitare le distinzioni tra le classi e semplificare la notazione **riducendola a  $c$  diverse istanze dello stesso problema**, ossia:

$$P(\omega_i | \mathbf{x}, D) = \frac{p(\mathbf{x} | \omega_i, D_i) P(\omega_i)}{\sum_{j=1}^c p(\mathbf{x} | \omega_j, D_j) P(\omega_j)}$$



$$p(\mathbf{x} | D)$$

# Distribuzione dei parametri

- Quello che vogliamo fare è effettivamente osservare come viene ottenuta  $p(\mathbf{x}|D)$  tramite l'ausilio di un modello di parametri *implicito*  $\theta$ .
- Ragionevolmente, abbiamo

$$p(\mathbf{x} | D) = \int p(\mathbf{x}, \theta | D) d\theta$$

dove l'integrazione si estende su tutto lo spazio dei parametri

# Distribuzione dei parametri

- Quindi

$$\begin{aligned} p(\mathbf{x} | D) &= \int p(\mathbf{x}, \boldsymbol{\theta} | D) d\boldsymbol{\theta} \\ &= \int p(\mathbf{x} | \boldsymbol{\theta}, D) p(\boldsymbol{\theta} | D) d\boldsymbol{\theta} \end{aligned}$$

- Poichè, per ipotesi, la probabilità di  $\mathbf{x}$  è indipendente dai campioni di training  $D$ , dato  $\boldsymbol{\theta}$ ,

$$p(\mathbf{x} | D) = \int p(\mathbf{x} | \boldsymbol{\theta}) p(\boldsymbol{\theta} | D) d\boldsymbol{\theta}$$



# Distribuzione dei parametri

- L'equazione precedente lega esplicitamente la densità condizionale  $p(\mathbf{x}|\mathbf{D})$  alla densità a posteriori  $p(\boldsymbol{\theta}|\mathbf{D})$  tramite il vettore sconosciuto di parametri  $\boldsymbol{\theta}$ .
- Se  $p(\boldsymbol{\theta}|\mathbf{D})$  si concentra fortemente su un valore, otteniamo una stima  $\hat{\boldsymbol{\theta}}$  del vettore più probabile, quindi

$$p(\mathbf{x}|\mathbf{D}) \approx p(\mathbf{x} | \hat{\boldsymbol{\theta}})$$

- Ma questo approccio *permette di tenere conto dell'effetto di tutti gli altri modelli*, descritti dal valore della funzione integrale, *per tutti i possibili modelli*.

$$p(\mathbf{x} | D) = \int p(\mathbf{x} | \boldsymbol{\theta}) p(\boldsymbol{\theta} | D) d\boldsymbol{\theta}$$

## Esempio: caso Gaussiano

$$p(\mathbf{x} | D) = \int p(\mathbf{x} | \boldsymbol{\theta}) p(\boldsymbol{\theta} | D) d\boldsymbol{\theta}$$

- Utilizziamo le tecniche di stima Bayesiana per calcolare la densità a posteriori  $p(\boldsymbol{\theta} | D)$ , e quindi la densità  $p(\mathbf{x} | D)$  per il caso in cui  $p(\mathbf{x} | \boldsymbol{\theta}) = p(\mathbf{x} | \boldsymbol{\mu}) \equiv p(x | \mu) \approx N(\mu, \sigma^2)$  *in cui l'unica quantità sconosciuta è la media  $\mu$ .*
- Devo quindi definire  $p(\boldsymbol{\theta} | D) = p(\mu | D)$

# Esempio: caso Gaussiano

- Con la regola di Bayes posso scrivere:

$$p(\mu | D) = \frac{p(D | \mu) p(\mu)}{\int p(D | \mu) p(\mu) d\mu}$$

**Densità  
riprodotta**

## PRIMO PASSO

$$p(\mu) \approx N(\mu_0, \sigma_0^2)$$

**Prior coniugato**

- in pratica  $\mu_0$  rappresenta la migliore scelta iniziale per il parametro  $\mu$ , con  $\sigma_0^2$  che ne misura l'incertezza.

# Esempio: caso Gaussiano

NOTA: la scelta del prior è arbitraria, ma:

- deve essere fatta (il prior deve essere noto)
- di solito si sceglie un prior coniugato
  - prior che assicura che la forma della posterior  $p(\mu/D)$  sia trattabile, cioè abbia la stessa forma della condizionale
  - Questo semplifica di molto l'analisi
  - Esempio: gaussiana per gaussiana, dirichlet per multinomiale

## Esempio: caso Gaussiano

- Supponiamo di avere  $n$  campioni di training  $D = \{x_1, x_2, \dots, x_n\}$  e riscriviamo la densità riprodotta come

$$\begin{aligned} p(\mu | D) &= \frac{p(D | \mu) p(\mu)}{\int p(D | \mu) p(\mu) d\mu} \\ &= \alpha \prod_{k=1}^n p(x_k | \mu) p(\mu) \end{aligned}$$

dove  $\alpha$  è un fattore di normalizzazione dipendente da  $D$ .

## Esempio: caso Gaussiano

- L'equazione mostra come l'osservazione del set di esempi di training influenzi la nostra idea sul vero valore di  $\mu$ ; essa relaziona la densità a priori  $p(\mu)$  con la densità a posteriori  $p(\mu/D)$ .

**SECONDO PASSO:** Svolgendo i calcoli, ci si accorge che, grazie al prior normale,  $p(\mu/D)$  risulta anch'essa normale, modificandosi in dipendenza del numero di campioni che formano il training set, evolvendosi in impulso di Dirac per  $n \rightarrow \infty$  (fenomeno di Learning Bayesiano).

- Formalmente si giunge alle seguenti formule:

# Esempio: caso Gaussiano

$$\begin{aligned} p(\mu|\mathcal{D}) &= \alpha \prod_{k=1}^n \overbrace{\frac{1}{\sqrt{2\pi}\sigma} \exp \left[ -\frac{1}{2} \left( \frac{x_k - \mu}{\sigma} \right)^2 \right]}^{p(x_k|\mu)} \overbrace{\frac{1}{\sqrt{2\pi}\sigma_0} \exp \left[ -\frac{1}{2} \left( \frac{\mu - \mu_0}{\sigma_0} \right)^2 \right]}^{p(\mu)} \\ &= \alpha' \exp \left[ -\frac{1}{2} \left( \sum_{k=1}^n \left( \frac{\mu - x_k}{\sigma} \right)^2 + \left( \frac{\mu - \mu_0}{\sigma_0} \right)^2 \right) \right] \\ &= \alpha'' \exp \left[ -\frac{1}{2} \left[ \left( \frac{n}{\sigma^2} + \frac{1}{\sigma_0^2} \right) \mu^2 - 2 \left( \frac{1}{\sigma^2} \sum_{k=1}^n x_k + \frac{\mu_0}{\sigma_0^2} \right) \mu \right] \right], \quad (29) \end{aligned}$$

## Esempio: caso Gaussiano

$$p(\mu | D) = \frac{p(D | \mu) p(\mu)}{\int p(D | \mu) p(\mu) d\mu} = \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left\{-\frac{(\mu - \mu_n)^2}{2\sigma_n^2}\right\}$$

$$\text{dove } \mu_n = \frac{n\sigma_0^2}{n\sigma_0^2 + \sigma^2} \left( \frac{1}{n} \sum_{k=1}^n x_k \right) + \frac{\sigma^2}{n\sigma_0^2 + \sigma^2} \mu_0$$

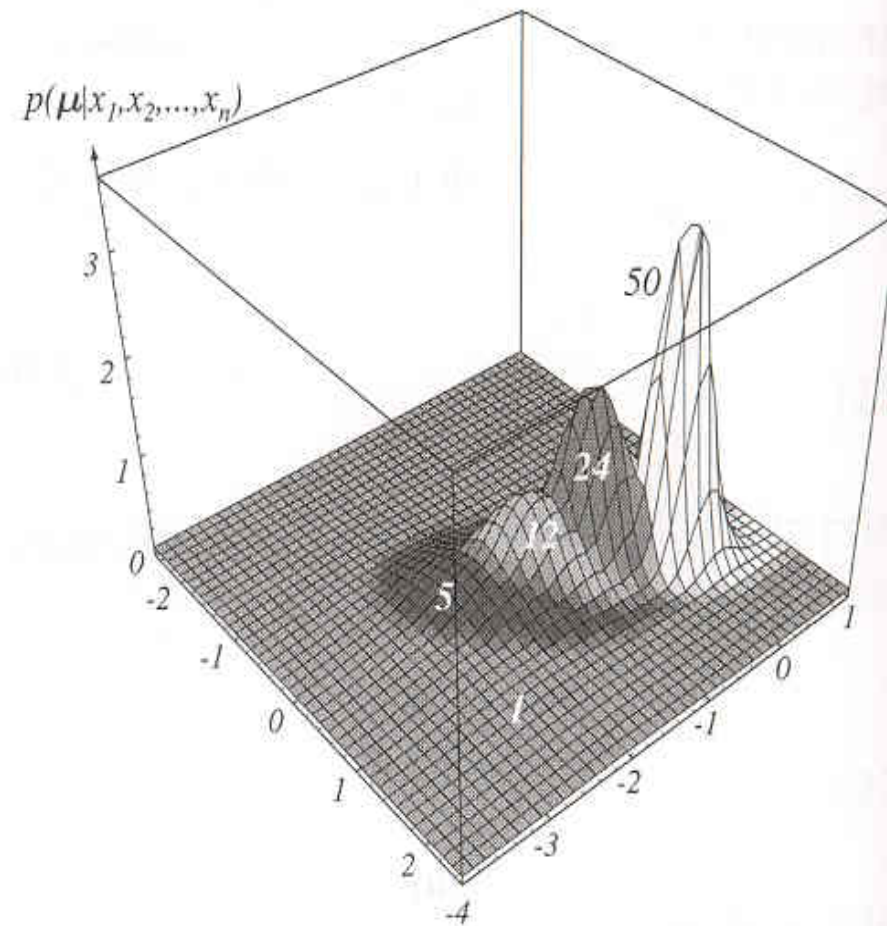
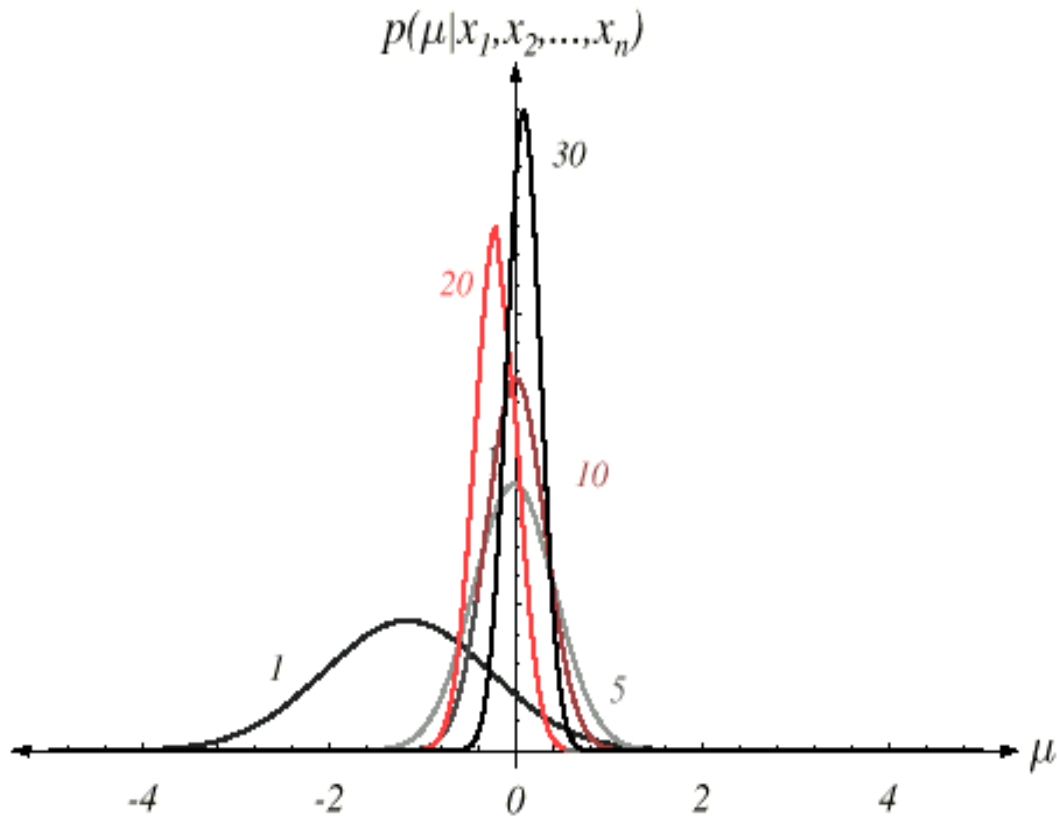
$$\sigma_n^2 = \frac{\sigma_0^2 \sigma^2}{n\sigma_0^2 + \sigma^2}$$

$\mu_n$  rappresenta la nostra migliore scelta per  $\mu$  dopo aver osservato  $n$  campioni.

$\sigma_n^2$  misura l'incertezza della nostra scelta.



# Esempio: caso Gaussiano



## Esempio: caso Gaussiano

**TERZO PASSO:** stima della densità condizionale  $p(x|\mathcal{D})$

$$p(x|\mathcal{D}) = \int p(x|\mu) p(\mu|\mathcal{D}) d\mu$$

$$= \int \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left[-\frac{1}{2}\left(\frac{\mu-\mu_n}{\sigma_n}\right)^2\right] d\mu$$

$$= \frac{1}{2\pi\sigma\sigma_n} \exp\left[-\frac{1}{2} \frac{(x-\mu_n)^2}{\sigma^2 + \sigma_n^2}\right] f(\sigma, \sigma_n), \quad (3)$$

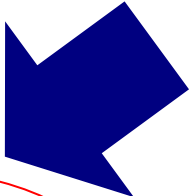
# Esempio: caso Gaussiano

dove

$$f(\sigma, \sigma_n) = \int \exp \left[ -\frac{1}{2} \frac{\sigma^2 + \sigma_n^2}{\sigma^2 \sigma_n^2} \left( \mu - \frac{\sigma_n^2 x + \sigma^2 \mu_n}{\sigma^2 + \sigma_n^2} \right)^2 \right] d\mu.$$

# Esempio: caso Gaussiano

- Concludendo, la densità  $p(\mathbf{x}/D)$  ( $=P(\mathbf{x} | \omega_i, D)$ ) ottenuta è la densità condizionale desiderata


$$P(\omega_i | \mathbf{x}, D) = \frac{p(\mathbf{x} | \omega_i, D)P(\omega_i)}{\sum_{j=1}^c p(\mathbf{x} | \omega_j, D)P(\omega_j)}$$

che assieme ai prior  $P(\omega_i)$  produce le informazioni desiderate per il design del classificatore, al contrario dell'approccio ML che restituisce solo le stime puntuali  $\hat{\mu}$  e  $\hat{\sigma}^2$

# Stima di Bayes: in generale

⇒ Riassumendo ed estendendole al caso generale, le formule principali viste sono:

$$p(\mathbf{x} | D) = \int p(\mathbf{x} | \boldsymbol{\theta}) p(\boldsymbol{\theta} | D) d\boldsymbol{\theta}$$

$$p(\mu | D) = \frac{p(D | \mu) p(\mu)}{\int p(D | \mu) p(\mu) d\mu} = \frac{p(D | \boldsymbol{\theta}) p(\boldsymbol{\theta})}{\int p(D | \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}} = p(\boldsymbol{\theta} | D)$$

$$p(D | \boldsymbol{\theta}) = \prod_{k=1}^n p(\mathbf{x}_k | \boldsymbol{\theta})$$

⇒ Si noti la somiglianza con l'approccio ML, con la differenza che qui non si cerca il max puntuale

# Conclusioni: Bayes vs ML

- ML restituisce una stima puntuale  $\hat{\Theta}$ , l'approccio Bayesiano una distribuzione su  $\theta$  (più ricca, tiene conto di tutti i possibili modelli)
- Bayes più accurato (in linea di principio), ML più fattibile in pratica
- Inoltre: ML, per un dataset abbastanza grande, produce risultati buoni
  - le stime risultano equivalenti per training set di cardinalità infinita (Al limite,  $p(\theta|D)$  converge ad una funzione delta)

# Conclusioni: Bayes vs ML

- In Bayes occorre stimare i prior  $\hat{\theta}$
- Praticamente, gli approcci sono differenti per vari motivi:
  - Complessità computazionale
  - Interpretabilità
  - Affidabilità delle informazioni a priori

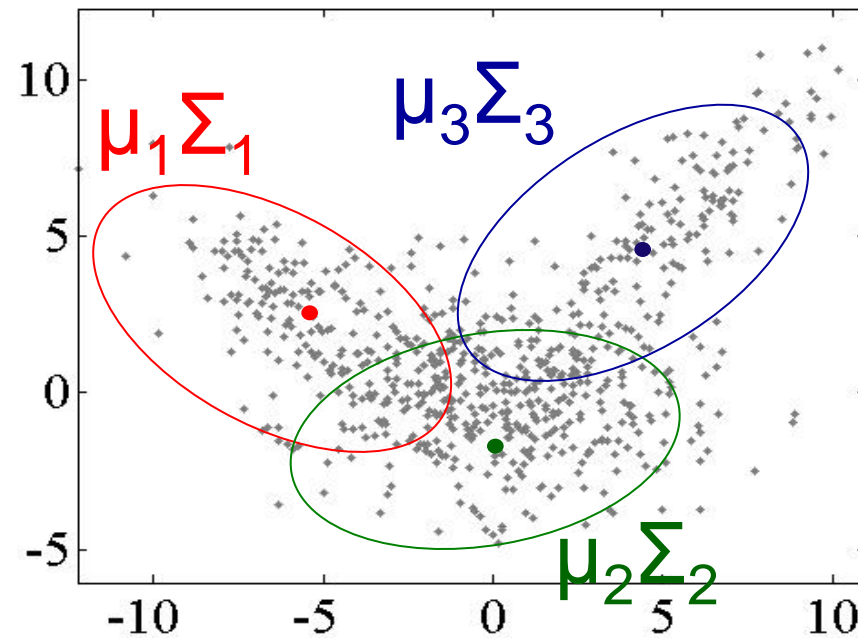
# Mapping slide-libro

- Cap.3 Duda until 3.4.2, 3.5.1





# Expectation-Maximization



$$p(v^{(t)} | h^\theta) = \sum_{i=1}^M \pi_i N(\mu_i, \Sigma_i)$$



# Introduction - Maximum Likelihood Estimation (MLE) problem

- INPUT:
  - A dataset of **observations**  $v = \{v^{(t)}\}_{t=1 \dots T}$
  - An *implicit* knowledge, i.e.
    - the dataset comes from a parametric random process
    - such random process has a known form (f.i. a mixture of Gaussians)
    - other (i.i.d. data, usually)
- OUTPUT:
  - the **set of parameters**  $h^\theta$  that maximizes the *likelihood*  $p(v|h^\theta)$  a.k.a. *objective function*  $L(h^\theta)$



# Introduction - MLE problem and EM solution

- Usually, the MLE is performed by **differentiating the likelihood** function with respect to the various parameters, and **solving for 0**

$$\frac{\partial \log p(v | h^\theta)}{\partial h_i^\theta} = 0 \quad i = 1 \dots I$$

- Sometimes, this solution is not feasible due to the **complex form of the likelihood**
- This is the situation in which the EM algorithm *helps*

# Introduction - EM



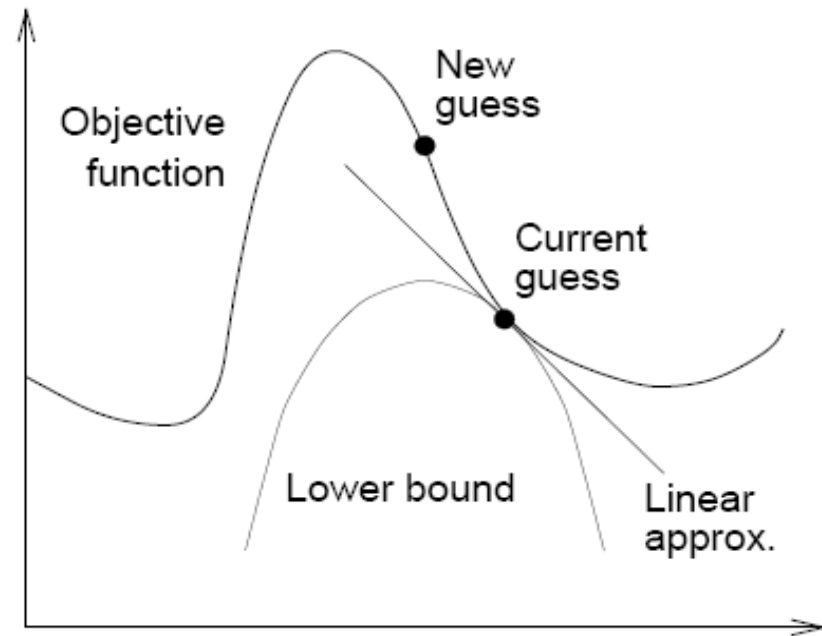
- Iterative process
- Each iteration is composed by 2 steps
  - E-step: Expectation
  - M-step: Maximization
- Convergent to a local maxima of the likelihood function
- Widespreadly used
  - genetics
  - statistics
  - econometrics



# Introduction -

## EM placement in the maximization methods literature

- *Gradient descent*: linear approximation to the  $L(h^\theta)$ 
  - we don't know how good is the approximation
  - we don't know how big the step to do
- *Newton methods*: quadratic approx
  - same problem as above
- *EM*:
  - at each E step it builds a local lower bound of the objective function
  - at each M step, a novel  $h^\theta$  which corresponds to a bigger value of the objective function

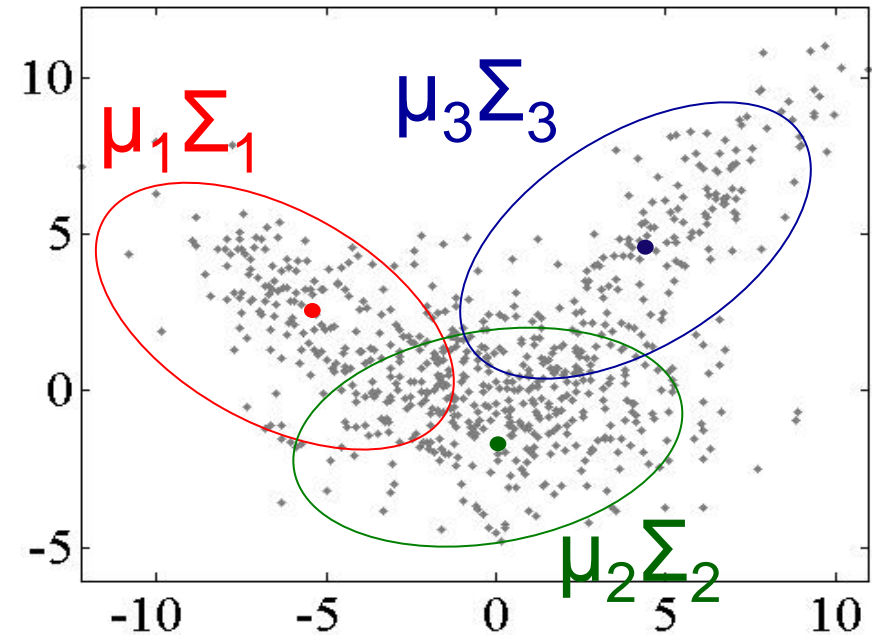




# Introduction -

## MLE example - Mixture of Gaussians (MoG)

- $p(v^{(t)} | h^\theta) = \sum_{i=1}^M \pi_i N(\mu_i, \Sigma_i)$
- $p(v | h^\theta) = \prod_{t=1}^T p(v^{(t)} | h^\theta)$
- $\log p(v | h^\theta) = \sum_{t=1}^T \log p(v^{(t)} | h^\theta)$
- $\log p(v | h^\theta) = \sum_{t=1}^T \log \sum_{i=1}^M \pi_i N(\mu_i, \Sigma_i)$





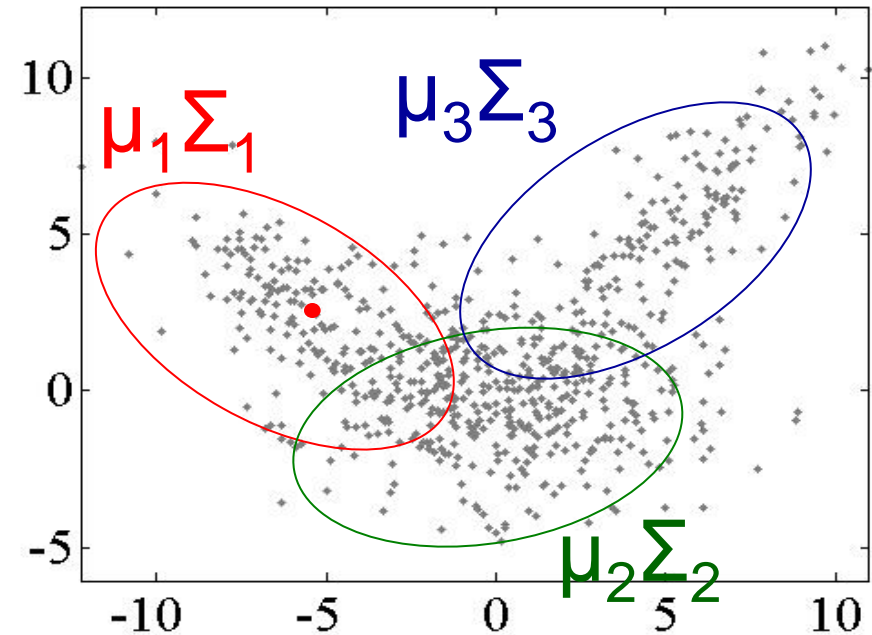
# Introduction - MLE example - MoGs (2)

- $p(v^{(t)} | h^\theta) = \sum_{i=1}^M \pi_i N(\mu_i, \Sigma_i)$

- $p(v | h^\theta) = \prod_{t=1}^T p(v^{(t)} | h^\theta)$

- $\log p(v | h^\theta) = \sum_{t=1}^T \log p(v^{(t)} | h^\theta)$

- $\log p(v | h^\theta) = \sum_{t=1}^T \log \sum_{i=1}^M \pi_i N(\mu_i, \Sigma_i)$



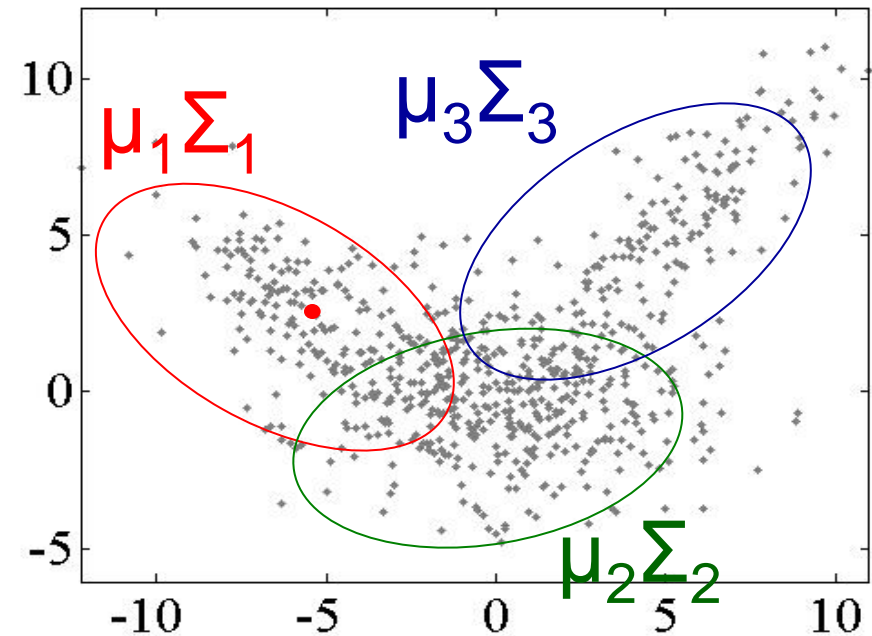
# Introduction - MLE example - MoGs (3)

- $$\log p(v | h^\theta) = \sum_{t=1}^T \log \sum_{i=1}^M \pi_i N(\mu_i, \Sigma_i)$$

- Goals

1. find  $h^\theta$
2. maximize  $\log p(v | h^\theta)$

$$\frac{\partial \log p(v | h^\theta)}{\partial h^\theta} = 0$$



## PROBLEMATIC

the parameters are coupled, due to the sum of the log: no closed form solution





# The algorithm - EM in one slide! - The EM trick

$$\log P(v | h^\theta) = \log P(v)$$

$$= \log \left( \int_h P(h, v) \right)$$

$$= \log \left( \int_h Q(h) \frac{P(h, v)}{Q(h)} \right)$$

$$\geq \int_h Q(h) \log \left( \frac{P(h, v)}{Q(h)} \right) = -F(Q(h), P(h, v))$$

**Jensen Inequality**

**The trick**



# The algorithm - Novel objects in the MLE instance

$$\int_h Q(h) \log \left( \frac{P(h, v)}{Q(h)} \right) = -F(Q(h), P(h, v)) \leq \log(P(v))$$

- $h$  = *hidden variable*
  - a hidden *quality* of the single data point
- $P(h, v)$  = *complete data (hidden + visible) likelihood*
  - it explains how the hidden variables and the visible ones are coupled together
- $Q(h)$  = *support distribution on the hidden variables*
  - a distribution over the hidden variables, simpler than  $P(h, v)$



The algorithm -

Novel objects in the MLE instance (2)

$$\int_h Q(h) \log \left( \frac{P(h, v)}{Q(h)} \right) = -F(Q(h), P(h, v)) \leq \log(P(v))$$

● •  $F(Q(h), P(h, v))$

- a divergence between  $Q, P$  a functional
- an inferior bound with respect to the objective function  $L(h^\theta)$
- an object with  $Q(h)$  unknown
- an object with  $h^\theta$  unknown





# The algorithm - Minimization of the divergence

- I minimize  $F(Q,P)$  *alternatively (the sign – vanishes during minimization)*

1. with respect to  $Q(h)$ , with  $h^\theta$  fixed

2. with respect to  $h^\theta$ , with  $Q(h)$  fixed


$$\frac{\partial}{\partial Q(h)} \int_h Q(h) \log \left( \frac{P(h, v)}{Q(h)} \right) = 0$$


$$\frac{\partial}{\partial h^\theta} \int_h Q(h) \log \left( \frac{P(h, v)}{Q(h)} \right) = 0$$


$$Q(h) = P(h | v, h^\theta)$$


$$\int_h Q(h) \frac{\partial}{\partial h^\theta} \log P(h, v | h^\theta) = 0$$



# The algorithm -

## The core of the EM in practice

- **INITIALIZATION**: set an initial  $h^\theta$
- **STEP E**: Minimize  $F(Q,P)$  with respect to  $Q(h^{(t)})$  calculating for each possible value of  $h^{(t)}$

$$Q(h^{(t)}) \leftarrow P(h^{(t)} | v^{(t)}, h^\theta)$$

**EASY TO  
COMPUTE !!!**

for each  $t$

- **STEP M**: Minimize  $F(Q,P)$  with respect to  $h^\theta$  solving

$$\sum_{t=1}^T \left( \int_{h^{(t)}} Q(h^{(t)}) \frac{\partial}{\partial \hat{h}^\theta} \log P(h^{(t)}, v^{(t)} | h^\theta) \right) = 0$$

**EASY TO  
COMPUTE !!!**

for  $M$  parameters, this is a system of  $M$  equations.

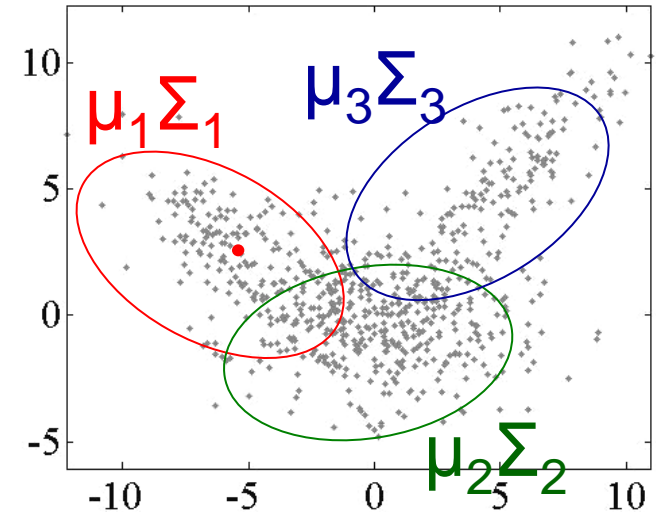


# The algorithm - Perplexities and practical receipts

- Cool, but when should I use EM?
  - with probabilistic problems, in which mixtures of *whatever* are involved, where each data point is generated by one of the components of the mixture
    - MoG (mixtures of Gaussian)
    - HMM (mixtures of states)
    - Bayes Net (mixtures of parents of a node)
- Crucial question: what is  $h^{(t)}$  ?
  - $h^{(t)}$  indicates what component of the mixture generates the data  $v^{(t)}$

# Applications - Back to the MoGs - the E-players

- $p(v^{(t)} | h^\theta) = \sum_{i=1}^M \pi_i N(\mu_i, \Sigma_i)$
- $p(v^{(t)} | h^{(t)}, h^\theta) = N(\mu_{h^{(t)}}, \Sigma_{h^{(t)}})$
- $p(h^{(t)} | h^\theta) = \pi_{h^{(t)}}$



**BAYES**

$$\begin{aligned}
 \bullet \quad P(h^{(t)} | v^{(t)}, h^\theta) &= \frac{P(v^{(t)} | h^{(t)}, h^\theta) P(h^{(t)} | h^\theta)}{P(v^{(t)} | h^\theta)} = \frac{P(v^{(t)} | h^{(t)}, h^\theta) P(h^{(t)} | h^\theta)}{\sum_{h^{(t)}} P(v^{(t)} | h^{(t)}, h^\theta) P(h^{(t)} | h^\theta)} \\
 &\quad \swarrow \quad \quad \quad \searrow \\
 &\quad Q(h^{(t)}) \quad \quad \quad = \frac{\pi_{h^{(t)}} N(\mu_{h^{(t)}}, \Sigma_{h^{(t)}})}{\sum_{h^{(t)}} \pi_{h^{(t)}} N(\mu_{h^{(t)}}, \Sigma_{h^{(t)}})}
 \end{aligned}$$

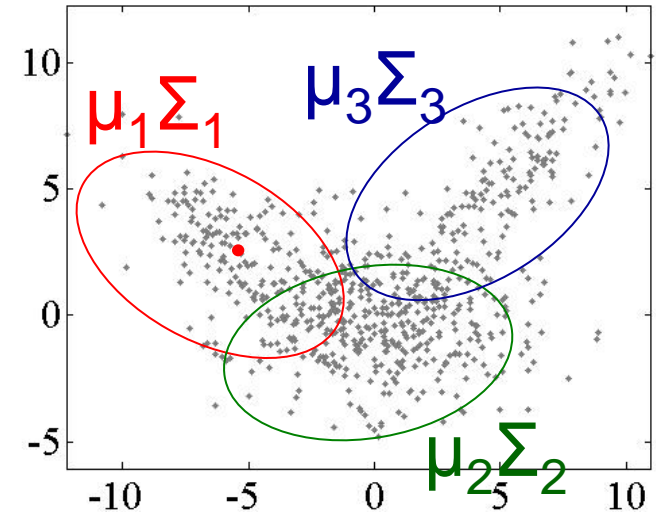
**Compute for each i, for each t**



# Applications - Back to the MoGs - the M-players

$$\bullet \sum_{t=1}^T \left( \int_{h^{(t)}} Q(h^{(t)}) \frac{\partial}{\partial \hat{h}^\theta} \log P(h^{(t)}, v^{(t)} | h^\theta) \right) = 0$$

$$= \sum_{t=1}^T \sum_{h^{(t)}=1}^K Q(h^{(t)}) \frac{\partial}{\partial \hat{h}^\theta} \log P(h^{(t)}, v^{(t)} | h^\theta) = 0$$



$$= \sum_{t=1}^T \sum_{h^{(t)}=1}^K P(h^{(t)} | v^{(t)}, \hat{h}^\theta) \frac{\partial}{\partial \hat{h}^\theta} \log \underbrace{P(v^{(t)} | h^{(t)}, h^\theta) P(h^{(t)} | h^\theta)}$$

$$\sum_{t=1}^T \sum_{h^{(t)}=1}^K P(h^{(t)} | v^{(t)}, \hat{h}^\theta) \frac{\partial}{\partial \hat{h}^\theta} \log P(v^{(t)} | h^{(t)}, h^\theta)$$

!!!

$$\sum_{t=1}^T \sum_{h^{(t)}=1}^K P(h^{(t)} | v^{(t)}, \hat{h}^\theta) \frac{\partial}{\partial \hat{h}^\theta} \log P(h^{(t)} | h^\theta)$$





$$\sum_{t=1}^T \sum_{h^{(t)}=1}^K P(h^{(t)} | v^{(t)}, \hat{h}^\theta) \frac{\partial}{\partial \hat{h}^\theta} \log P(v^{(t)} | h^{(t)}, h^\theta)$$

(ricorda!  $--> P(v^{(t)} | h^{(t)}, h^\theta) = N(\mu_{h^{(t)}}, \Sigma_{h^{(t)}})$  )

$$\begin{aligned} & \sum_{t=1}^T \sum_{h^{(t)}=1}^K P(h^{(t)} | v^{(t)}, \hat{h}^\theta) \frac{\partial}{\partial \mu_k} \log N(\mu_{h^{(t)}}, \Sigma_{h^{(t)}}) \\ &= \sum_{t=1}^T P(h^{(t)} | v^{(t)}, \hat{h}^\theta) \Sigma^{-1} (v^{(t)} - \mu_k) = 0 \\ &= \sum_{t=1}^T P(h^{(t)} | v^{(t)}, \hat{h}^\theta) \Sigma^{-1} v^{(t)} - P(h^{(t)} | v^{(t)}, \hat{h}^\theta) \Sigma^{-1} \mu_k = 0 \\ & \mu_k = \frac{\sum_{t=1}^T P(h^{(t)} | v^{(t)}, \hat{h}^\theta) v^{(t)}}{\sum_{t=1}^T P(h^{(t)} | v^{(t)}, \hat{h}^\theta)} \end{aligned}$$



$$\mu_k = \frac{\sum_{t=1}^T P(h^{(t)} | v^{(t)}, \hat{h}^\theta) v^{(t)}}{\sum_{t=1}^T P(h^{(t)} | v^{(t)}, \hat{h}^\theta)}$$

$$\Sigma_k = \frac{\sum_{t=1}^T P(h^{(t)} | v^{(t)}, \hat{h}^\theta) (v^{(t)} - \mu_k)(v^{(t)} - \mu_k)^T}{\sum_{t=1}^T P(h^{(t)} | v^{(t)}, \hat{h}^\theta)}$$

$$p(h^{(t)} | h^\theta) = \pi_{h^{(t)}}$$

$$\alpha_i = \frac{1}{T} \sum_{t=1}^T P(h^{(t)} | v^{(t)}, \hat{h}^\theta)$$



# The algorithm -

## The core of the EM in practice

- **INITIALIZATION**: set an initial  $\{\mu_i^{(init)}, \Sigma_i^{(init)}, \pi_i^{(init)}\}$
- **STEP E**: Minimize  $F(Q,P)$  with respect to  $Q(h^{(t)})$  calculating for each possible value of  $h^{(t)}$

$$Q(h^{(t)}) \leftarrow P(h^{(t)} | v^{(t)}, h^\theta)$$

**EASY TO  
COMPUTE !!!**

for each  $t$

- **STEP M**: Minimize  $F(Q,P)$  with respect to  $h^\theta$ , obtaining

$$\{\mu_i^{(new)}, \Sigma_i^{(new)}, \pi_i^{(new)}\}$$

- **ITERATION** Loop until convergence!



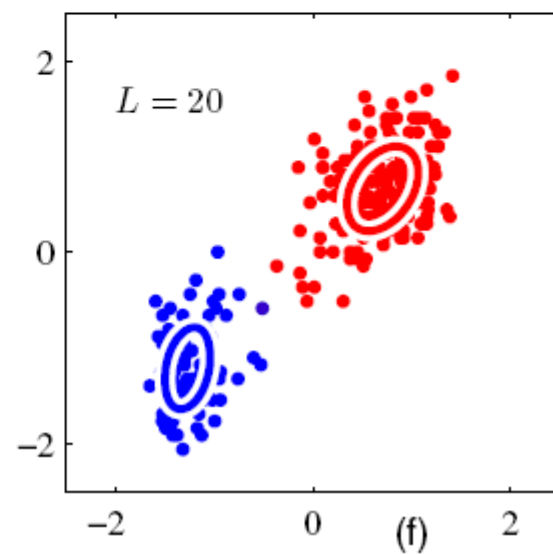
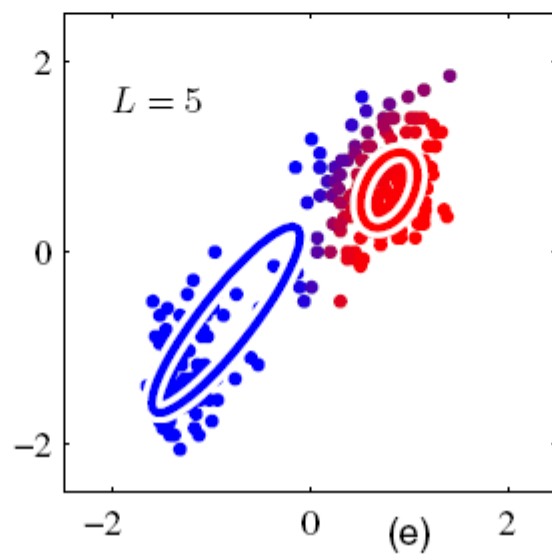
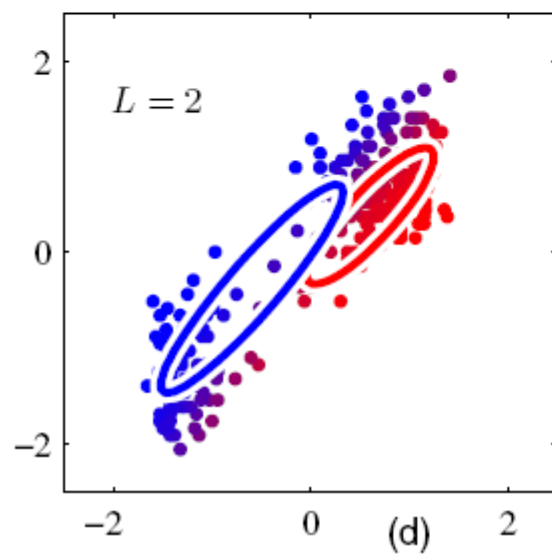
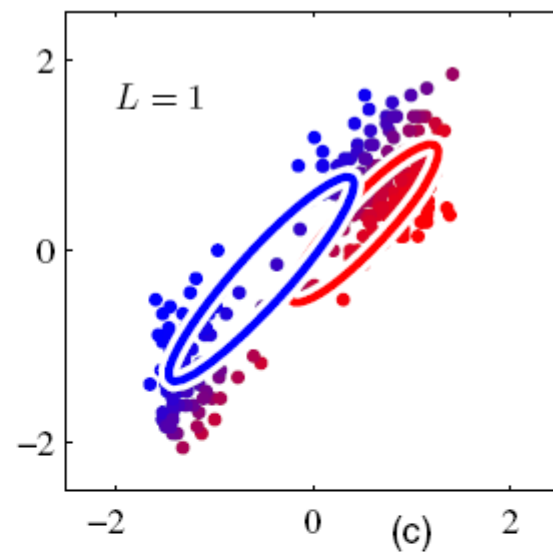
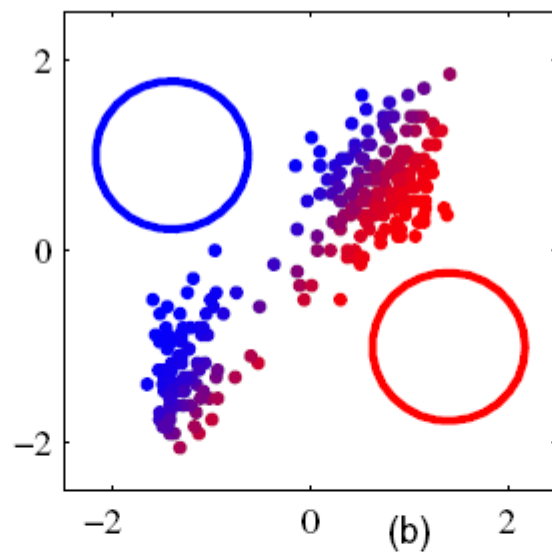
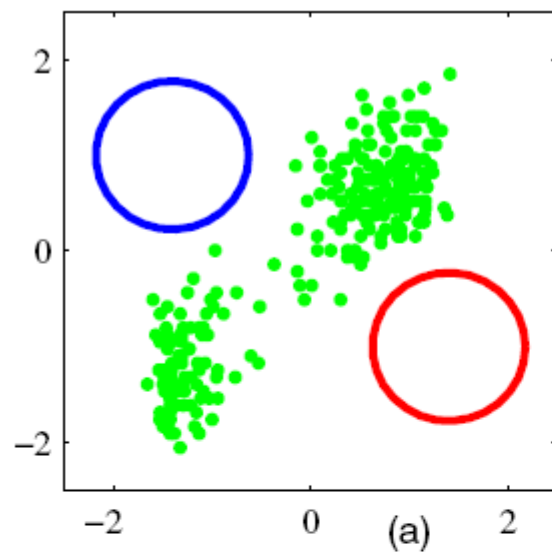
# The m-players

- The idea: introduce hidden variables which knowledge simplifies the computation of the parameters
- The hidden variables are related with the visible variables
- The decision of the hidden quantities is not an automatic process, and relies on the scientist
- In genera, the EM well apply when we have to deal with mixtures



# Remarks

- The idea: introduce hidden variables which knowledge simplifies the computation of the parameters
- The hidden variables are related with the visible variables
- The decision of the hidden quantities is not an automatic process, and relies on the scientist
- In genera, the EM well apply when we have to deal with mixtures





# Material

- Brendan J. Frey and Nebojsa Jojic. 2005. A Comparison of Algorithms for Inference and Learning in Probabilistic Graphical Models. *IEEE Trans. Pattern Anal. Mach. Intell.* 27, 9 (September 2005), 1392-1416.  
DOI=10.1109/TPAMI.2005.169  
<http://dx.doi.org/10.1109/TPAMI.2005.169>
- A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models (1998) by Jeff Bilmes