

Computer Science Department

University of Verona

A.A. 2015-16

## **Pattern Recognition**

**Non parametric density estimation:  
static and dynamic cases**

# Stima non parametrica

- Problema della stima parametrica: si assume che la forma delle densità di probabilità sia nota, ma questa assunzione non può essere fatta in molti problemi di riconoscimento.
- In particolare, se la scelta della forma parametrica è sbagliata, la stima sarà molto povera
  - Esempio: distribuzione multimodale che viene assunta essere Gaussiana
- Soluzione: **metodi non parametrici**:
  - Nessuna assunzione sulla forma generale della pdf che modella la distribuzione dei dati di training di una classe

# Stima non parametrica

- Idea generale: stimare *localmente* la pdf andando ad analizzare le singole regioni dello spazio
  - in particolare, mi interessa la likelihood  $p(\mathbf{x} | \omega)$  : vado a considerare la regione attorno ad  $\mathbf{x}$  ed effettuo una stima considerando quella regione
- Esempio: *istogramma*
  - suddivido lo spazio delle features in regioni di larghezza uniforme
  - dato un insieme di punti  $D$  proveniente dalla distribuzione che devo stimare (ossia, i dati di training), per ogni regione conto il numero di punti che ci cade dentro: questo conteggio mi porta alla stima nonpar.

# Stima non parametrica della densità condizionale

Problema: stima di  $p(\mathbf{x} | \omega) = p(\mathbf{x})$  per semplicità di notazione

- La **probabilità** che un vettore  $\mathbf{x}$  sia in una regione (volume)  $\mathbf{R}$  è:

$$P = \int_{\mathbf{R}} p(\mathbf{y}) d\mathbf{y} \quad (1)$$

- $P$  è una versione *smoothed* (o mediata) della **densità**  $p(\mathbf{x})$ . Consideriamo un insieme di  $n$  campioni (i.i.d.) estratti secondo  $p$ : la probabilità che  $k$  punti su  $n$  siano in  $\mathbf{R}$  è data dalla legge binomiale:

$$P(k) = \binom{n}{k} P^k (1 - P)^{n-k} \quad (2)$$

che ha come parametro  $\theta = P$

– La stima ML di  $P$ , ossia

$$\max_{\theta} (P \mid \theta) \text{ è data da } \hat{\theta} = \frac{k}{n} \cong P$$

*Proof.* Immaginiamo di avere un set di  $N$  dati di training, e di osservare  $k$  estrazioni nella regione

$$L(P \mid n, X) = P^k (1 - P)^{n-k}$$

$$\log L(P \mid n, k) = +k \log P + (n - k) \log(1 - P)$$

Deriviamo rispetto a  $P$  ( $n$  è conosciuto)

$$\frac{\partial \log L}{\partial P} = \frac{1}{P} k - \frac{1}{1-P} (n - k)$$

pongo  $= 0$  e risolvo rispetto a  $P$

$$(1 - P)k - P(n - k) = 0$$

$$k - Pk - Pn + Pk = 0$$

$$P = \frac{k}{n} \quad (3)$$

- Quindi, il rapporto  $k/n$  è una buona stima per la probabilità  $P$ .
- Se  $p(\mathbf{y})$  è continua e la regione  $\mathbf{R}$  è così piccola che  $p$  non varia significativamente in essa (così da essere approssimabile ad una costante), possiamo scrivere:

$$\int_{\mathbf{R}} p(\mathbf{y}) d\mathbf{y} \cong p(\mathbf{x}) \cdot V \quad (4)$$

dove  $\mathbf{x}$  è un punto in  $\mathbf{R}$  e  $V$  è il volume incluso in  $\mathbf{R}$ .

Combinando le equazioni (1), (3), (4) stimiamo finalmente  $p$ :

$$P = \int_R p(\mathbf{y}) d\mathbf{y} \quad (1)$$

$$\int_R p(\mathbf{y}) d\mathbf{y} \cong p(\mathbf{x}) \cdot V \quad (4)$$

$$P = \frac{k}{n} \quad (3)$$

$$p(\mathbf{x}) \cong \frac{k}{nV}$$

# Stima non parametrica

- Data una regione  $R$  di volume  $V$ , dati  $n$  punti (di cui  $k$  cadono nella regione  $R$ ), si approssima  $p(\mathbf{x})$  in quella regione come:

$$p(\mathbf{x}) \approx \frac{k}{nV}$$

- **NOTA:** questa formula deriva da due approssimazioni, la cui bontà dipende da  $R$ 
  - $k/n$  stimatore di  $P$ : migliore per  $R$  grande
  - $p(\mathbf{x})$  costante in  $R$ : migliore per  $R$  piccola
- Scelta di  $R$  è quindi cruciale!

# Stima non parametrica

Due possibilità per determinare  $p(\mathbf{x})$  per ogni possibile  $\mathbf{x}$

1. (più intuitiva): si fissa la regione  $R$  centrata in  $\mathbf{x}$  (in particolare si fissa il suo volume  $V$ ), si calcola  $k$  dai dati e si stima  $p(\mathbf{x})$ 
  - più punti ci sono nel volume fissato  $V$ , più alta la probabilità
  - Parzen Windows (Esempio istogramma)
2. (meno intuitiva): si fissa  $k$ , si sceglie  $R$  in modo tale che contenga  $k$  punti attorno ad  $\mathbf{x}$ , si determina  $V$  e si stima  $p(\mathbf{x})$ 
  - più grande è la regione che devo considerare per trovare  $k$  punti, più bassa è la probabilità
  - K-Nearest Neighbor

# Parzen Windows

- Assumiamo che la regione  $R$  sia un ipercubo di lato  $h$  (in uno spazio  $d$ -dimensionale)

$$V = h^d$$

- Possiamo ottenere una forma analitica per  $k$ , il numero di punti che cadono nella regione  $R$ , definendo la seguente funzione

$$\gamma(\mathbf{y}) = \begin{cases} 1, & |y_i| < 1/2 \\ 0, & \text{altrimenti} \end{cases} \quad i = 1, \dots, D$$

che rappresenta un ipercubo di lato unitario centrato nell'origine (*window function*)

# Parzen Windows

- Questa funzione può essere specificata meglio definendo tre argomenti,  $\mathbf{x}$ ,  $\mathbf{x}_i$  e  $h$

$$\gamma\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right)$$

che vale 1 se  $\mathbf{x}_i$  cade nell'ipercubo di lato  $h$  centrato in  $\mathbf{x}$ , zero altrimenti

- Il numero  $k$  di punti che stanno nell'ipercubo di lato  $h$  (la regione  $R$ ) centrato in  $\mathbf{x}$  è quindi

$$k = \sum_{j=1}^N \gamma\left(\frac{\mathbf{x} - \mathbf{x}_j}{h}\right)$$

# Parzen Windows

- Sostituendo nella formula di prima otteniamo

$$p(\mathbf{x}) = \frac{k}{NV} = \frac{1}{N} \frac{1}{h^d} \sum_{j=1}^N \gamma\left(\frac{\mathbf{x} - \mathbf{x}_j}{h}\right),$$

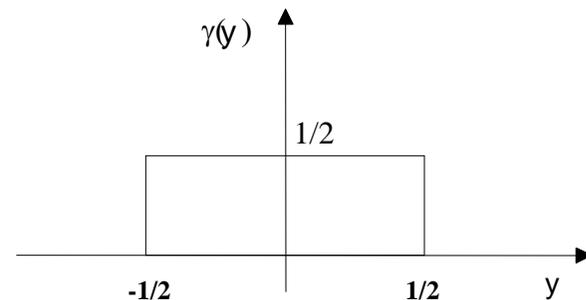
- Questo suggerisce un metodo più generale per stimare le densità di probabilità
  - Permettere più tipi di funzioni window
  - La stima della pdf è ottenuta come la media di queste funzioni di  $\mathbf{x}$  e  $\mathbf{x}_j$
  - In altre parole ogni campione contribuisce alla stima della pdf in un punto  $\mathbf{x}$
  - Il contributo è diverso a seconda della distanza da  $\mathbf{x}$

- Sia  $\gamma(\mathbf{x}, \mathbf{x}_i)$  la funzione potenziale per due campioni generici  $\mathbf{x}, \mathbf{x}_i$  (omettiamo  $h$ , ininfluyente per la definizione).
- Per costruire una buona approssimazione dobbiamo porre alcuni vincoli sulla forma di  $\gamma$ .
  - 1)  $\gamma(\mathbf{x}, \mathbf{x}_i) \geq 0$
  - 2)  $\arg \max_{\mathbf{x}} \gamma(\mathbf{x}, \mathbf{x}_i) = \mathbf{x}_i$ , cioè  $\gamma(\mathbf{x}, \mathbf{x}_i)$  è massima per  $\mathbf{x} = \mathbf{x}_i$ ;
  - 3)  $\gamma(\mathbf{x}, \mathbf{x}_1) \cong \gamma(\mathbf{x}, \mathbf{x}_2)$ , se  $|\mathbf{x}_2 - \mathbf{x}_1| < \varepsilon$ , cioè se i due vettori dei campioni sono "abbastanza" vicini (questo vincolo serve a garantire che  $p$  non vari bruscamente o possa avere discontinuità).
  - 4)  $\gamma(\mathbf{x}, \mathbf{x}_i)$  continua.
  - 5)  $\int_{-\infty}^{+\infty} \gamma(\mathbf{x}, \mathbf{x}_i) d\mathbf{x} = 1$  condizione di normalizzazione.
  - 6)  $\gamma(\mathbf{x}, \mathbf{x}_i) \cong 0$ , se  $\mathbf{x}$  è molto lontana da  $\mathbf{x}_i$ .

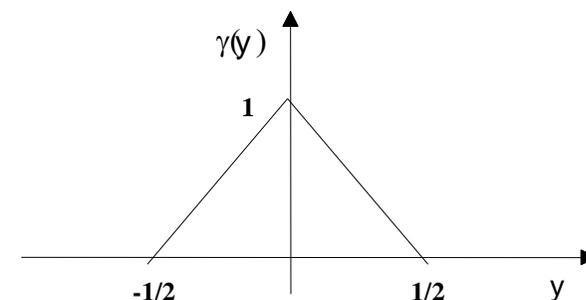
- Esistono diverse possibili forme di  $\gamma$  che tengono conto di questi vincoli, riferendoci al caso monodimensionale.
- Nel seguito, per semplicità, ritrasformo la funzione  $\gamma(\mathbf{x}, \mathbf{x}_i)$  in modo che abbia come variabile indipendente una sola variabile  $y$ , rappresentata dalla norma euclidea della differenza tra i due vettori  $\mathbf{x}$  e  $\mathbf{x}_i$ , i.e.,  $y := |\mathbf{x} - \mathbf{x}_i|$ .

# Esempi di funzioni potenziali

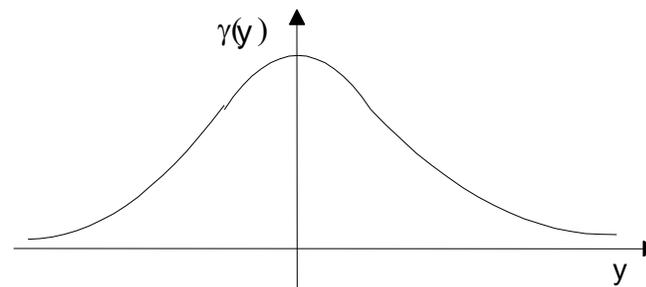
1)  $\gamma(\mathbf{y}) = \begin{cases} 0,5 & |\mathbf{y}| \leq 1 \\ 0 & |\mathbf{y}| > 1 \end{cases}$  Rettangolo



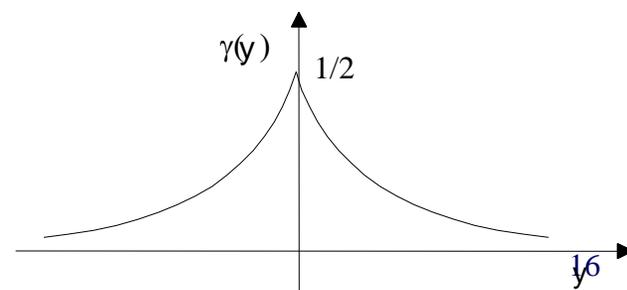
2)  $\gamma(\mathbf{y}) = \begin{cases} 1 - |\mathbf{y}| & |\mathbf{y}| \leq 1 \\ 0 & |\mathbf{y}| > 1 \end{cases}$  Triangolo



3)  $\gamma(\mathbf{y}) = (2\pi)^{-\frac{1}{2}} e^{-\left(\frac{\mathbf{y}^2}{2}\right)}$  Gaussiana

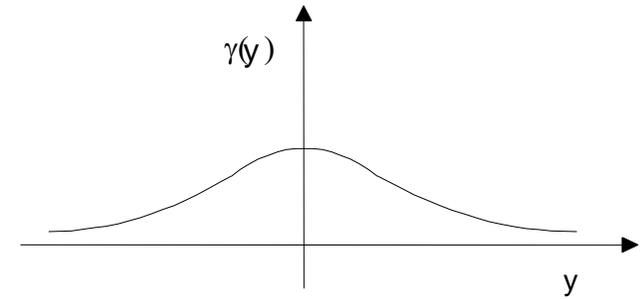


4)  $\gamma(\mathbf{y}) = \frac{1}{2} e^{-|\mathbf{y}|}$  Esponenziale decrescente



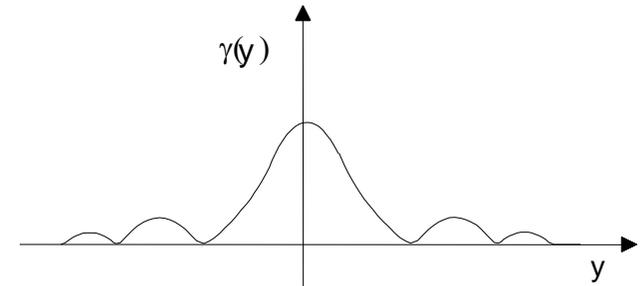
5)  $\gamma(\mathbf{y}) = [\pi(1 + \mathbf{y}^2)]^{-1}$

Distribuzione di Cauchy



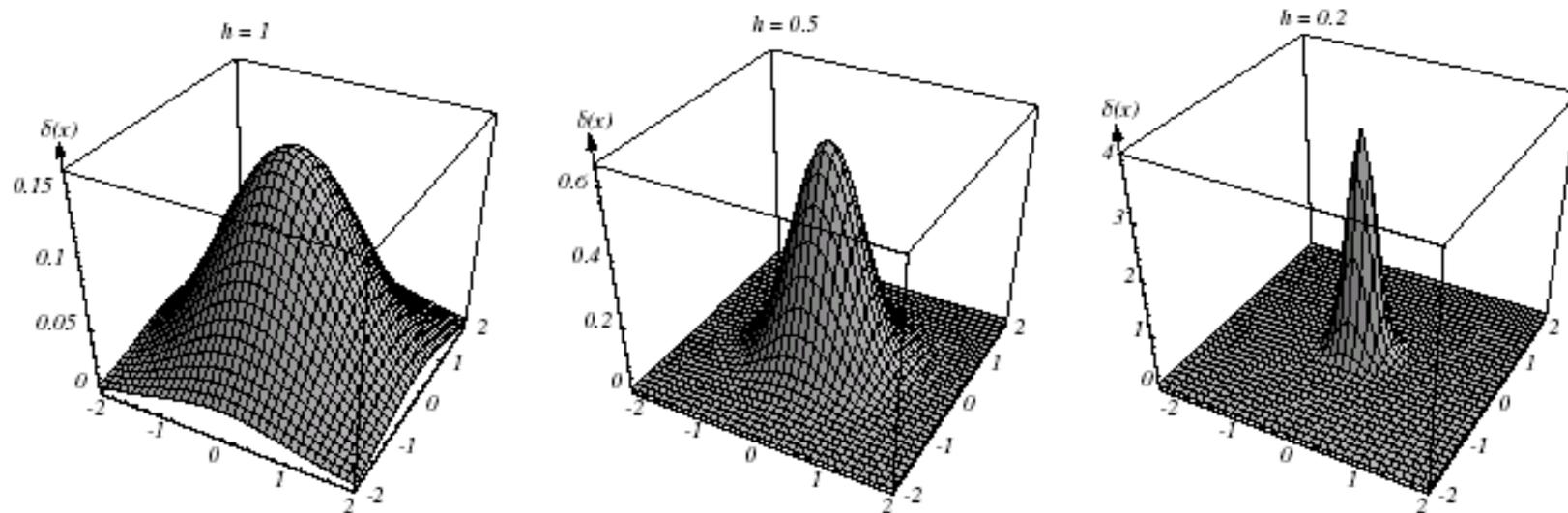
6)  $\gamma(\mathbf{y}) = (2\pi)^{-1} \left( \frac{\sin\left(\frac{\mathbf{y}}{2}\right)}{\frac{\mathbf{y}}{2}} \right)^2$

Funzione di tipo  $(\sin y/y)$



# Effetto dell'ampiezza $h$

- NOTA: solo i punti “vicini” ad  $\mathbf{x}$  influiscono sul calcolo della  $p(\mathbf{x})$
- $h$  determina l'ampiezza della finestra di interesse, cioè definisce in qualche modo il concetto di vicinato



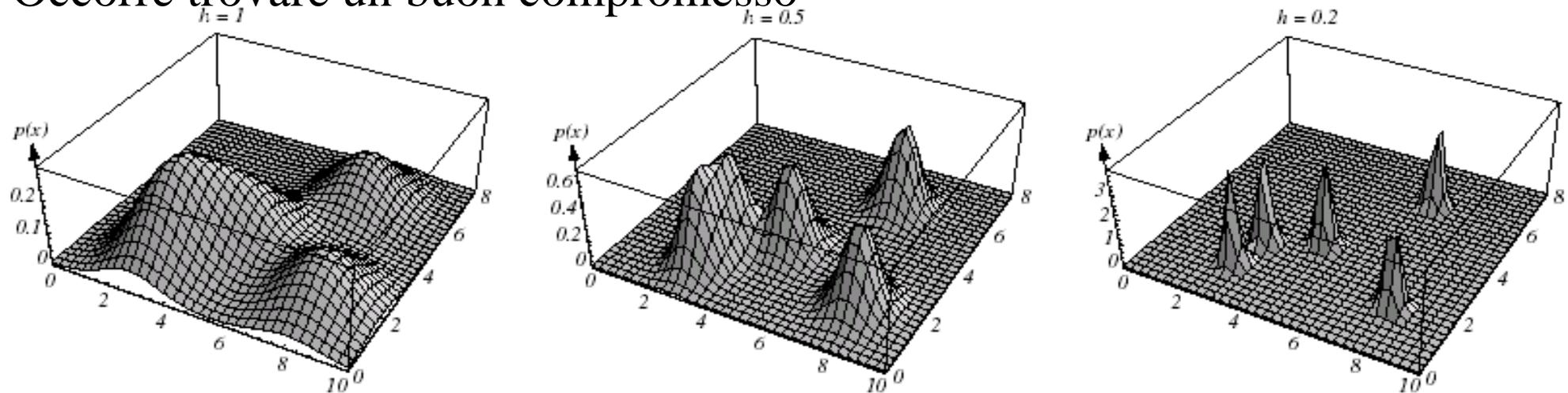
**FIGURE 4.3.** Examples of two-dimensional circularly symmetric normal Parzen windows for three different values of  $h$ . Note that because the  $\delta(\mathbf{x})$  are normalized, different vertical scales must be used to show their structure. From: Richard O. Duda, Peter E. H. Marco Cristani David G. Stork, *Pattern Classification e Tecniche del Riconoscimento* © 2001 by John Wiley & Sons, Inc.

# Effetto dell'ampiezza $h$

La scelta di  $h$  è cruciale

- $h$  troppo grande: molto smooth, tutto più o meno uguale
- $h$  troppo piccolo: un sacco di picchi singoli (dove  $\mathbf{x}=\mathbf{x}_i$ )

Occorre trovare un buon compromesso



**FIGURE 4.4.** Three Parzen-window density estimates based on the same set of five samples, using the window functions in Fig. 4.3. As before, the vertical axes have been scaled to show the structure of each distribution. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

# K-Nearest Neighbor

Secondo metodo per stimare non parametricamente  $p(\mathbf{x})$

$$p(\mathbf{x}) \approx \frac{k}{nV}$$

si fissa  $k$ , si sceglie  $R$  in modo tale che contenga  $k$  punti attorno ad  $\mathbf{x}$ , si determina  $V$  e si stima  $p(\mathbf{x})$

- Effettuando questa stima non parametrica delle posterior di tutte le classi, e applicando la regola di classificazione di Bayes, si ottiene il classificatore K-nearest Neighbor

# K-Nearest Neighbor

- Presentazione:
  - i. come funziona e su che intuizione si basa
  - ii. in che senso rappresenta il classificatore di Bayes nel caso di stima non parametrica delle posterior
- Funzionamento (molto semplice):
  - sia  $X$  un insieme di esempi etichettati (il training set, ogni punto ha la sua classe)
  - dato un punto  $\mathbf{x}$  da classificare, si calcola l'insieme  $U$  dei  $k$  punti dell'insieme  $X$  più vicini ad  $\mathbf{x}$  secondo una determinata metrica
  - Si calcola la classe  $C$  più frequente all'interno dell'insieme  $U$
  - Si assegna  $\mathbf{x}$  a  $C$

# Classificazione di Bayes con KNN

- Vediamo la probabilità a priori: data una classe  $\omega_j$ , si valuta in genere semplicemente la frequenza di occorrenza dei campioni  $N_j$  della classe  $j$  rispetto al numero totale di campioni  $N$ , i.e.,

$$P(\omega_j) \equiv \hat{p}(\omega_j) = \frac{N_j}{N}$$

# Stima della densità e regola dei punti vicini

- 2 classi,  $\omega_i$  e  $\omega_j$ , contenenti  $N_i$  ed  $N_j$  campioni,  $N = N_i + N_j$
- La stima locale di densità per  $\omega_i$  si calcola come (e analogamente per  $\omega_j$ )

$$\hat{p}(\mathbf{x}|\omega_i) = \frac{1}{V} \frac{k_i}{N_i}$$

i.e., rapporto tra  $k_i$  ( $k_j$ ) punti sugli  $N_i$  ( $N_j$ ) totali appartenenti alla classe  $\omega_i$  ( $\omega_j$ ) contenuti nel volume  $V$

- La regola di Bayes è  $p(\mathbf{x}|\omega_i)P(\omega_i) > p(\mathbf{x}|\omega_j)P(\omega_j)$ , allora

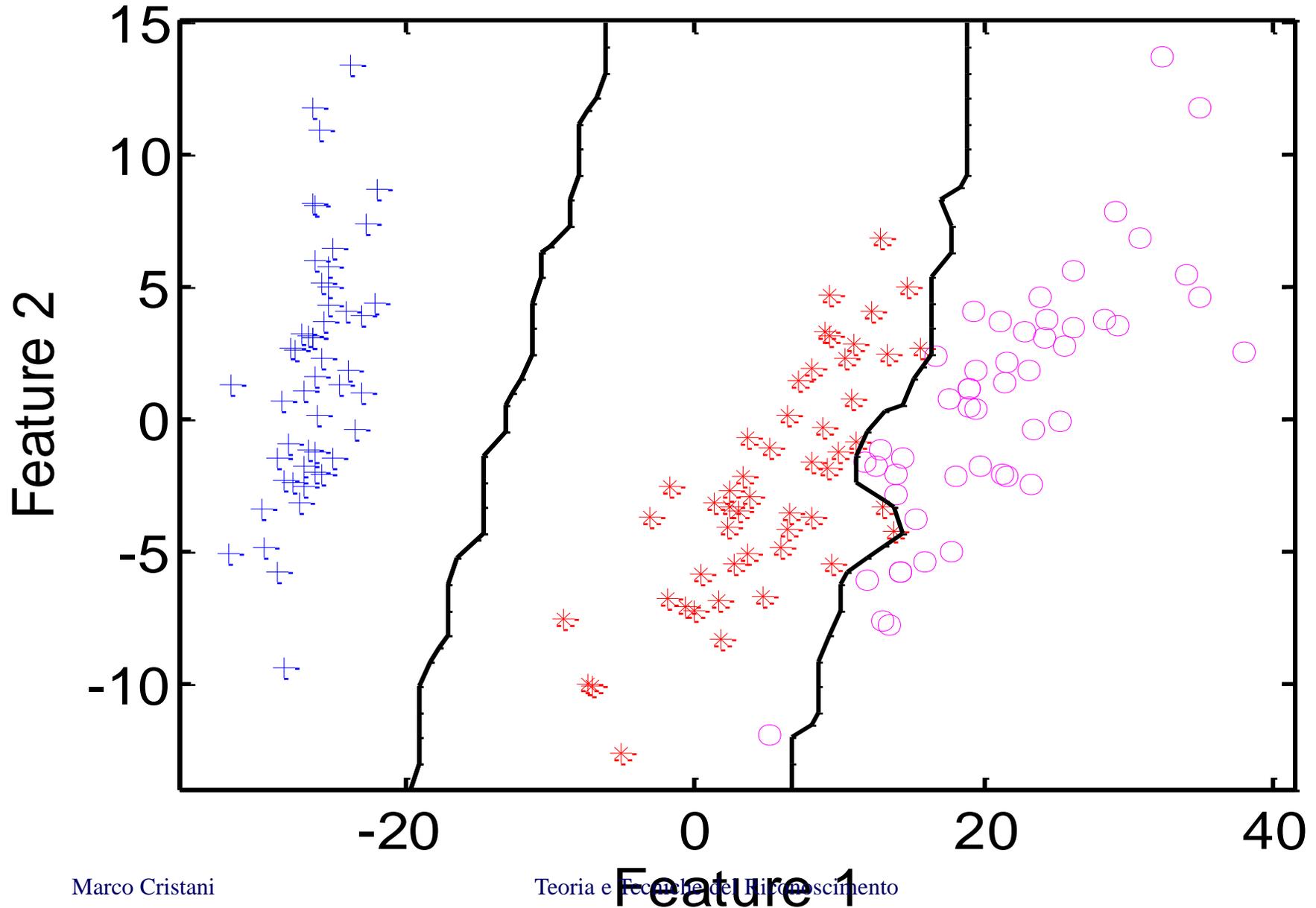
$$\begin{aligned} \hat{p}(\mathbf{x}|\omega_i)\hat{p}(\omega_i) &> \hat{p}(\mathbf{x}|\omega_j)\hat{p}(\omega_j) \\ \Rightarrow \frac{1}{V} \frac{k_i}{N_i} \frac{N_i}{N} &> \frac{1}{V} \frac{k_j}{N_j} \frac{N_j}{N} \quad \Rightarrow \quad k_i > k_j \end{aligned}$$

# K-Nearest Neighbor

## VANTAGGI

- tecnica semplice e flessibile
- tecnica intuitiva (assume che punti della stessa classe abbiano probabilmente caratteristiche simili, cioè una distanza bassa)
- tecnica che funziona anche per dati non vettoriali (basta avere una misura di distanza appropriata)
- ragionevolmente accurata (il confine di separazione è comunque non lineare)
- ci sono pochi parametri da aggiustare
- sono stati dimostrati molti risultati teorici su questa tecnica (asintoticità del comportamento)

# Iris plants



# K-Nearest Neighbor

## SVANTAGGI

- Tutti i punti del training set devono essere mantenuti in memoria
- vengono utilizzati solo pochi punti dello spazio per prendere la decisione (solo  $k$  punti)
- dipendentemente dalla metrica utilizzata, occorre pre-processare lo spazio
- Serve una misura di distanza buona
- La scelta di  $k$  spesso è cruciale ( $k = 1 \rightarrow$  Nearest Neighbor rule)

$\Rightarrow$  scelta tipica

$$k \cong \sqrt{N}$$

# K-Nearest Neighbor: note finali

## Determinazione di $k$

- è equivalente al parametro  $h$  di Parzen Windows
  - troppo piccolo si hanno stime troppo rumorose
  - troppo grande si hanno stime troppo grezze
- Metodo per stimare  $k$ :
  - crossvalidation sul training set (o su Validation Set)
    - si provano diversi valori e si tiene quello che funziona meglio
  - Metodi locali: si decide guardando la regione dove si sta operando (ad esempio guardando il  $k$  che funziona meglio localmente)

# Stima di una densità che evolve nel tempo - Tracking

- Definizioni equivalenti
  - Capire come si spostano gli oggetti in una sequenza
  - Inseguire gli oggetti distinti nella scena durante il loro movimento, stimandone la traiettoria
    - nel 3D della scena
    - nel piano immagine
- Opzioni per la telecamera:
  - statica-mobile
  - unica-multipla



# Passi fondamentali

- **Inizializzazione**, cattura nuovi soggetti entranti nella scena
- **Inseguimento**
  - insegue gli oggetti nella scena mantenendone l'identità
    - metodologia "forza bruta"
      - correlazione
      - correlazione + vincoli
    - metodologia classica
      1. predizione
      2. verifica, associazione



# Inizializzazione

- Trovare un oggetto da seguire
- Metodologie
  - Template matching
    - Output di un classificatore
      - Face detector, hand detector, altro
    - Cross-correlazione normalizzata
  - Sottrazione del background
  - Metodi ibridi



# Inseguimento – metodo forza bruta

- Due immagini:  $Z_t$  e  $Z_{t-1}$
- Associa ad ogni oggetto  $O_{i,t-1}$  il corrispondente  $O_{i,t}$  dove  $i=1,\dots,K$

t-1

t



come scegliere  
le associazioni?



# Correlazioni

- Soluzioni:
  1. esaustivamente (lento, impreciso)
  2. con vincoli
    - definire/stimare le caratteristiche degli oggetti
      - visuali
      - spaziali
      - di moto
        - velocità (intensità, direzione), accelerazione
    - predire la posizione di un oggetto al passo  $t$
    - pesare la correlazione



# Condensation

- Particle filtering
  - Metodi di Montecarlo
- Condensation classico
- Bramble: Condensation multi-oggetto

Visual Motion Analysis by Probabilistic Propagation of Conditional Density,  
Michael Isard, D.Phil. Thesis, Oxford University, 1998



# Particle filtering - Assunzioni

- Assumiamo che
  - lo **stato** del sistema che stiamo osservando al tempo  $t$  sia definito da
    - $x_t$ 
      - La posizione di un singolo oggetto puntiforme
    - $\mathbf{X}_t = \{x_{1,t}, x_{2,t}, \dots, x_{M,t}\}$ 
      - La forma di un oggetto, definita da  $M$  punti definiti sulla sua silhouette
      - $M$  oggetti puntiformi
  - la **storia** (discreta!!!) del sistema fino al tempo  $t$  sia  $\mathcal{X}_t = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_t\}$



# Particle filtering - Assunzioni

- lo **stato** del sistema fino al tempo  $t$  sia osservabile da un set di osservazioni  $\mathcal{Z}_t = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_t\}$
- La presenza di una storia indica un'evoluzione del sistema nel tempo
- Assumiamo l'evoluzione del sistema come
  - un processo stocastico
  - Markoviano tempo-indipendente di ordine 1

$$P(\mathbf{X}_t | \mathcal{X}_t) = P(\mathbf{X}_t | \mathbf{X}_{t-1})$$

- Esempio 1D

$$P(x_t | x_{t-1}) = \exp - \frac{1}{2} (x_t - x_{t-1} - 1)^2$$



# Particle filtering - Assunzioni

- lo **stato** del sistema fino al tempo  $t$  sia osservabile da un set di **osservazioni**  $\mathcal{Z}_t = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_t\}$ , le quali
  - non modificano lo stato del sistema

$$p(\mathcal{Z}_{t-1}, \mathbf{X}_t | \mathcal{X}_{t-1}) = p(\mathbf{X}_t | \mathcal{X}_{t-1}) \prod_{i=1}^{t-1} p(\mathbf{z}_i | \mathbf{X}_i)$$

- sono indipendenti

$$p(\mathcal{Z}_t | \mathcal{X}_t) = \prod_{i=1}^t p(\mathbf{z}_i | \mathbf{X}_i)$$



# Particle filtering - Obiettivi

- stimare lo stato
  - più probabile
  - attesodel sistema  $\mathbf{X}_t$  date tutte le osservazioni  $\mathcal{Z}_t$
- per fare questo, è necessario valutare la distribuzione su tale probabilità, ossia recuperare  $p(\mathbf{X}_t | \mathcal{Z}_t)$
- Eseguendo questo controllo ad ogni istante  $t$ , ciò corrisponde a valutare l'evoluzione di una distribuzione nel tempo



# Particle filtering - Obiettivi

- Tramite la regola sulla legge condizionale (provate con  $t=3$ )

$$p(\mathbf{X}_t | \mathcal{Z}_t) = k_t p(\mathbf{Z}_t | \mathbf{X}_t) p(\mathbf{X}_t | \mathcal{Z}_{t-1})$$

dove

$$p(\mathbf{X}_t | \mathcal{Z}_{t-1}) = \int_{\mathbf{X}_{t-1}} p(\mathbf{X}_t | \mathbf{X}_{t-1}) p(\mathbf{X}_{t-1} | \mathcal{Z}_{t-1}) d\mathbf{X}_{t-1}$$



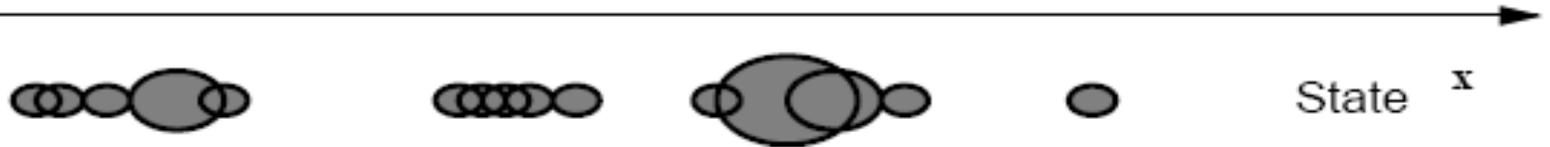
# Algoritmo di Condensation

- Assumo uno campo di esistenza  $R$  dello stato del sistema
  - 1D
  - finito
- Il sistema è una particella puntiforme su  $R$
- Ho un set di campioni  $\{s_{t-1}^{(n)}\}$   $n=1, \dots, N$  che rappresentano possibili stati del mio sistema (quindi posizioni in  $R$ ) al tempo  $t-1$



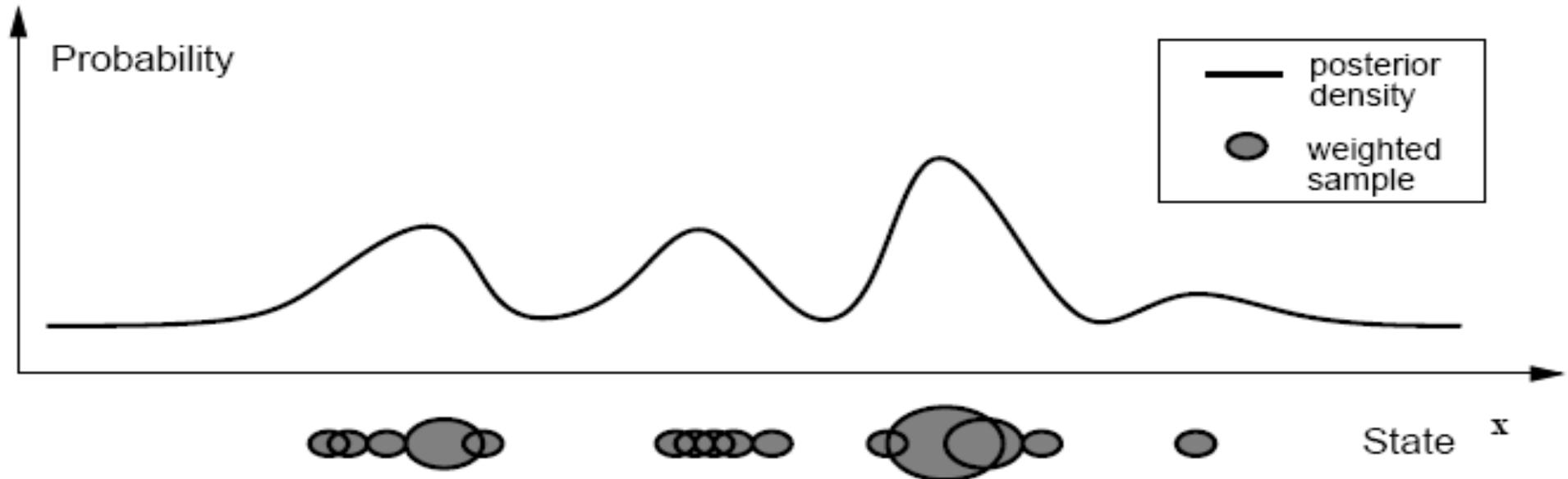
# Algoritmo di Condensation - Inizializzazione

- Ogni campione rappresenta un intorno spaziale locale
- Per modellare il fatto che in alcuni punti ho maggiore probabilità di avere il mio sistema, peso le particelle 1-1 con un set di pesi  $n=1, \dots, N$   $\{\pi_{t-1}^{(n)}\}$



# Algoritmo di Condensation - Inizializzazione

- A questo punto posso stimare una densità su tutto  $R$

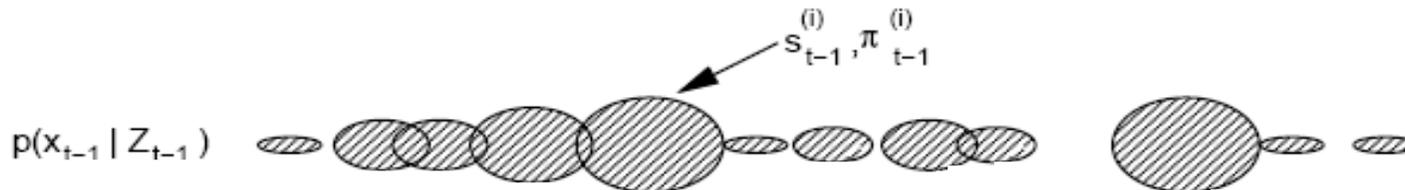


# Tracking via Condensation

- Supponendo di partire da questa conoscenza, il particle filtering fa evolvere il set di particelle
- 3 passi:

## 1. Selezione o campionamento

- campiona (estraggo)  $N$  campioni da  $\{s_{t-1}^{(n)}\}$
- il campione  $s_{t-1}^{(n)}$  viene selezionato con probabilità  $\pi_{t-1}^{(n)}$



# Tracking via Condensation (2)

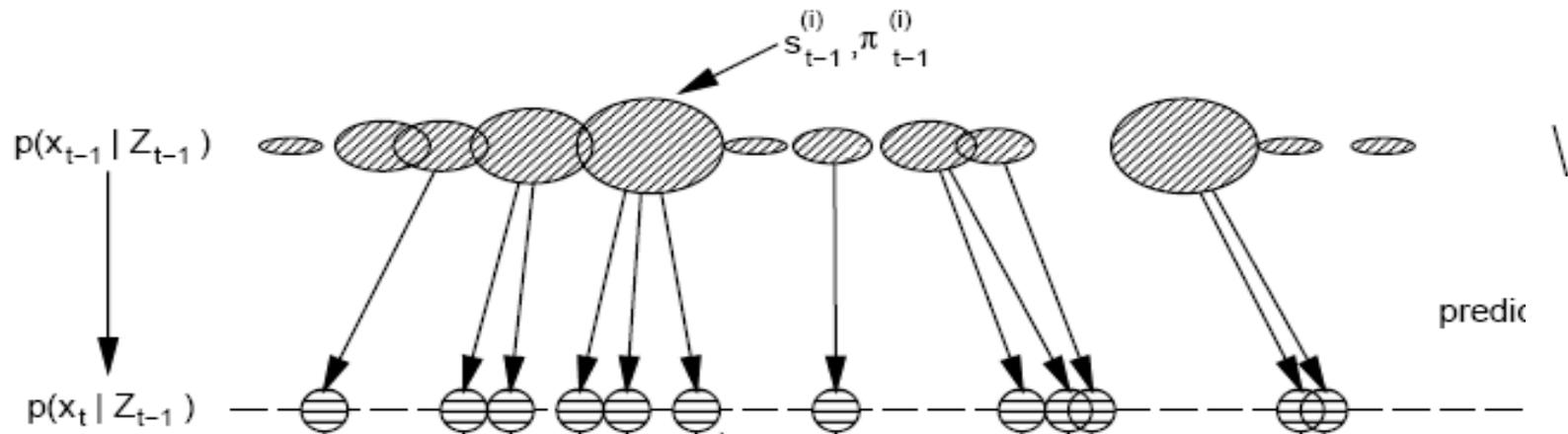
## 2. Applicazione della dinamica – per tutte le particelle

- Applico alla particella  $S_t^{(n)}$  un moto composto da due componenti:
  - Deterministica (*deterministic drift*)
    - Basato sulla storia della particella  $S_{t-1}^{(\tilde{n})}$  da cui  $S_t^{(n)}$  è stata estratta
    - Basata su un'unica dinamica pesata su tutte le particelle (vedi seguito)
  - Probabilistica (*diffuse*) aggiungo del rumore  $\mathcal{N}(0, \sigma_d^2)$  per modellare l'incertezza sul moto

ESEMPIO: 
$$p(x_t | x_{t-1}, \dot{x}_{t-1}) = \mathcal{N}(x_{t-1} + 0.8\dot{x}_{t-1}, \sigma_x^2)$$



# Tracking via Condensation (3)



- In tal maniera ottengo la parte predittiva (= a priori) della formulazione Bayesiana del filtro

$$p(\mathbf{X}_t | \mathcal{Z}_t) = k_t p(\mathbf{Z}_t | \mathbf{X}_t) p(\mathbf{X}_t | \mathcal{Z}_{t-1})$$

$$p(\mathbf{X}_t | \mathcal{Z}_{t-1}) = \int_{\mathbf{X}_{t-1}} p(\mathbf{X}_t | \mathbf{X}_{t-1}) p(\mathbf{X}_{t-1} | \mathcal{Z}_{t-1}) d\mathbf{X}_{t-1}$$



# Tracking via Condensation (4)

## 3. Valutazione/pesatura

- Calcolo la likelihood di ogni nuovo campione basandomi sulle osservazioni
- Le osservazioni rappresentano la possibilità di osservare *localmente* la realtà della scena
- Nel caso di moto monodimensionale, le osservazioni sono la possibilità di osservare l'intorno

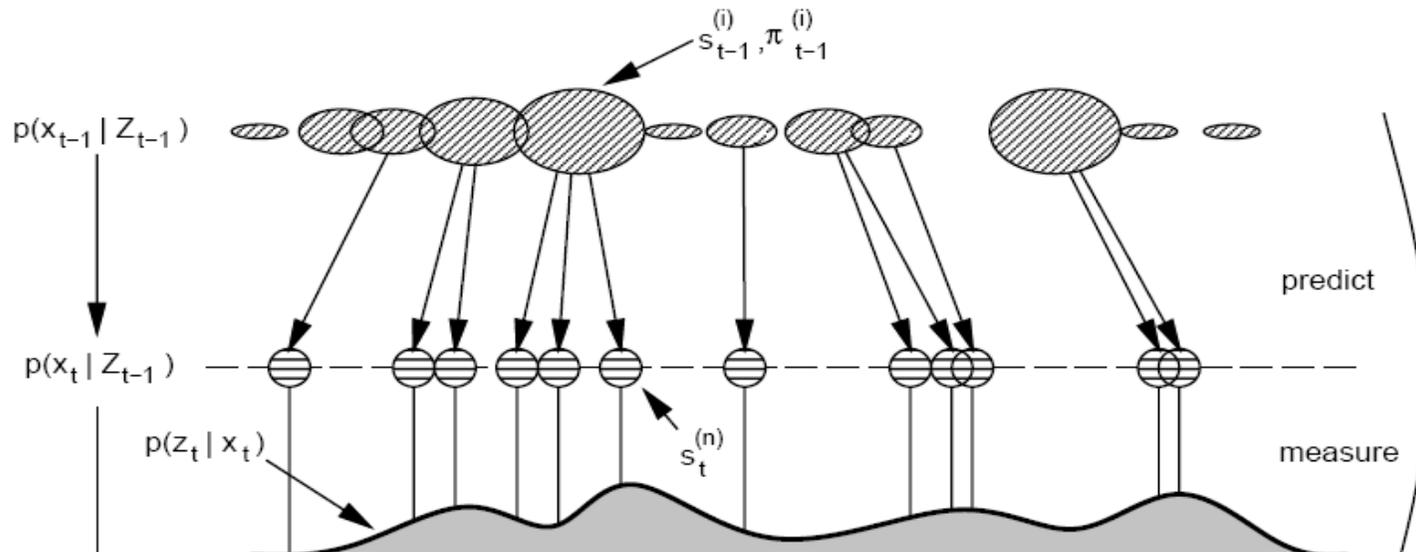
$$I(s_t^{(n)}) = [s_t^{(n)} - h, s_t^{(n)} + h], \quad h \in \mathfrak{R}$$



# Tracking via Condensation(5)

- Formalmente, si calcola  $p(z_t^{(n)} | s_t^{(n)})$
- Nel caso 1D, si puo' utilizzare una funzione di lik. Gaussiana

$$p(z_t^{(n)} | s_t^{(n)}) = \mathcal{N}(z_t^{(n)} | \mu = s_t^{(n)}, \sigma_w^2)$$



# Tracking via Condensation (6)

- Attenzione:

$$\sigma_w^2 > \sigma_d^2$$

altrimenti si perde l'obiettivo.

- La likelihood ottenuta serve per la creazione dei nuovi pesi associati alle particelle, ossia

$$\pi_t^{(n)} = p(z_t^{(n)} | s_t^{(n)}) \quad (+ \text{normalizzazione})$$

- In questo modo, prese tutte le particelle, si realizza la parte di likelihood del filtro bayesiano

$$p(\mathbf{X}_t | \mathcal{Z}_t) = k_t p(\mathbf{Z}_t | \mathbf{X}_t) p(\mathbf{X}_t | \mathcal{Z}_{t-1})$$



# Tracking via Condensation - definizione dell'obiettivo

- Decisione fondamentale: dove si trova l'oggetto all'istante  $t$ ?
  - Due soluzioni (generalmente nel caso multi-dim.):
    - Media pesata

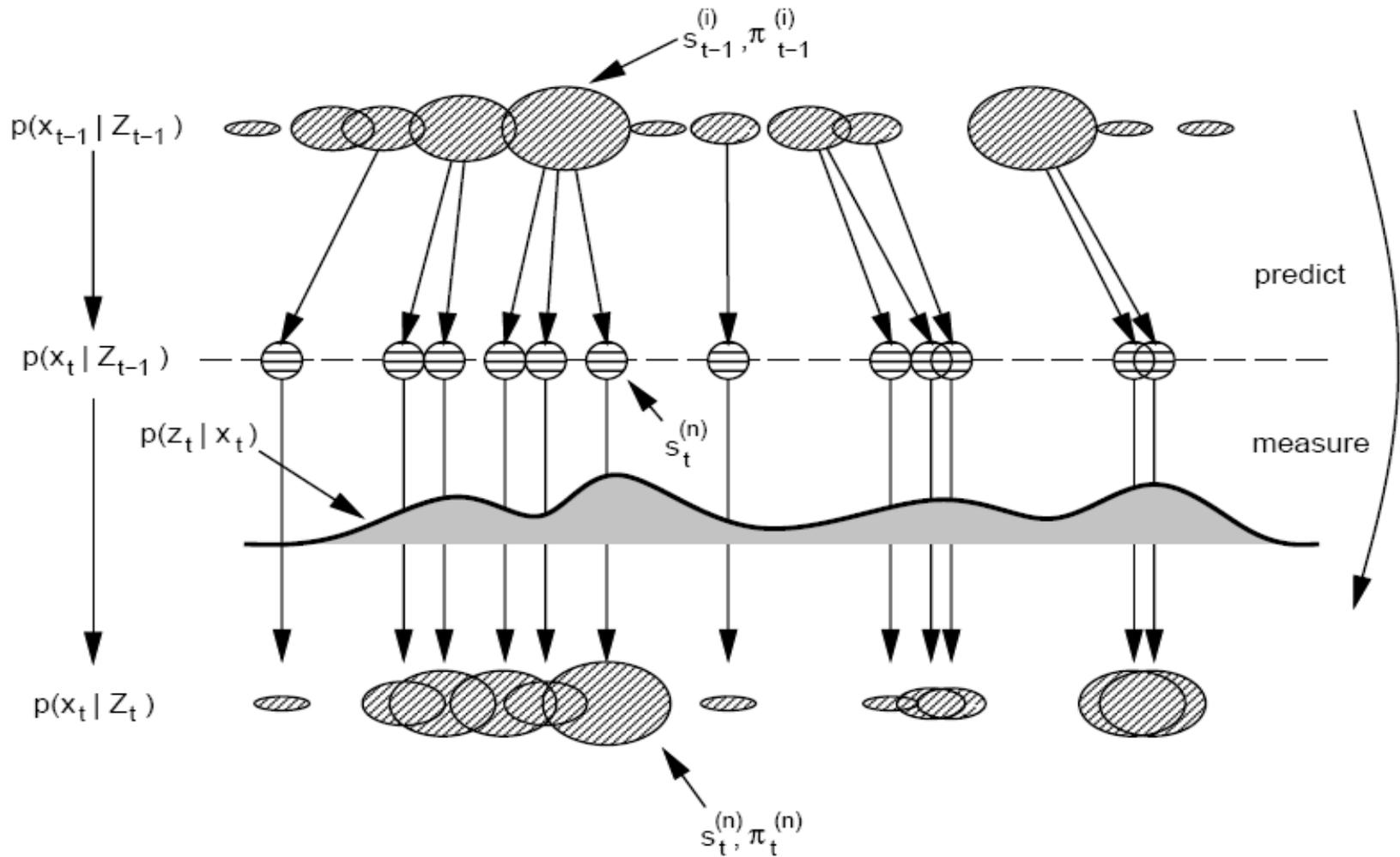
$$\mathcal{E}[f(\mathbf{X}_t)] = \sum_{n=1}^N \pi_t^{(n)} f(\mathbf{s}_t^{(n)}) \quad \text{con} \quad f(\mathbf{X}) = \mathbf{X}$$

- Maximum A Posteriori (MAP)

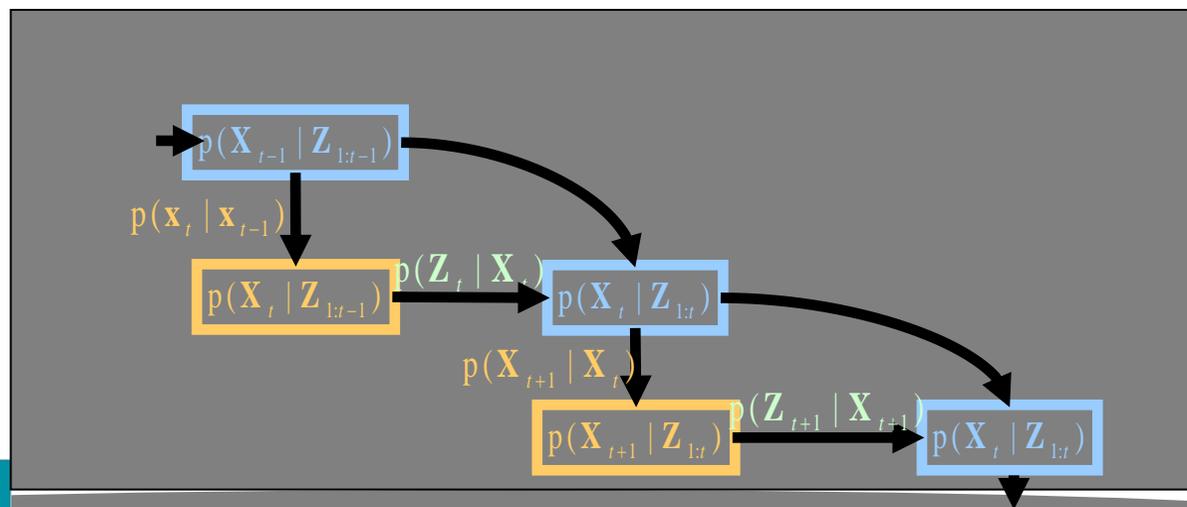
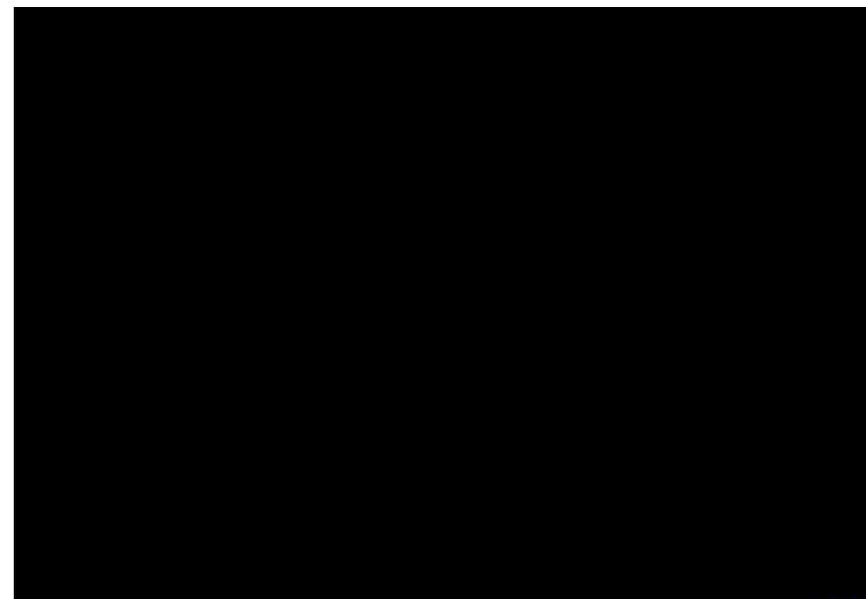
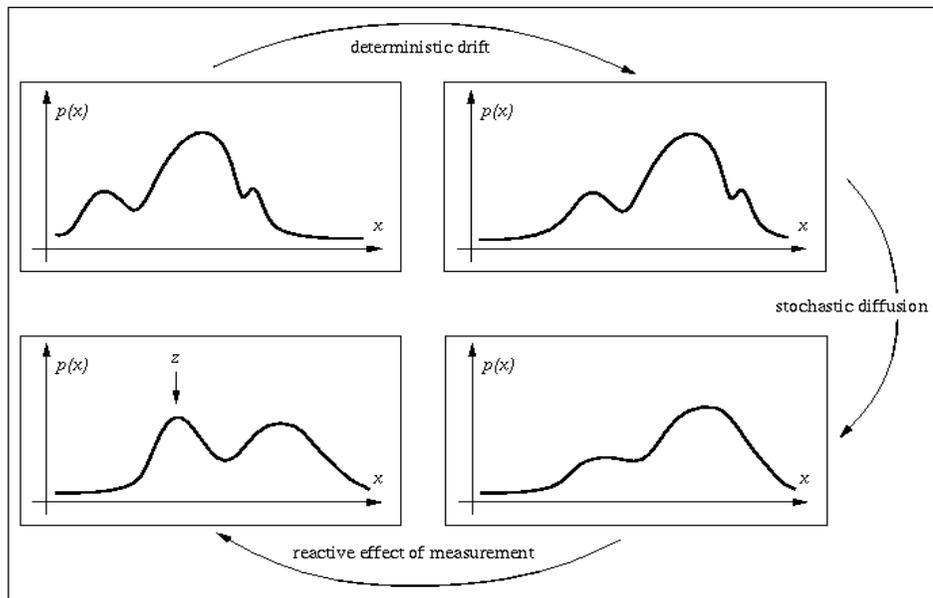
$$M[f(\mathbf{X}_t)] = f\left(\arg \max_{s_t^{(n)}} p(s_t^{(n)} | x_t^{(n)})\right)$$



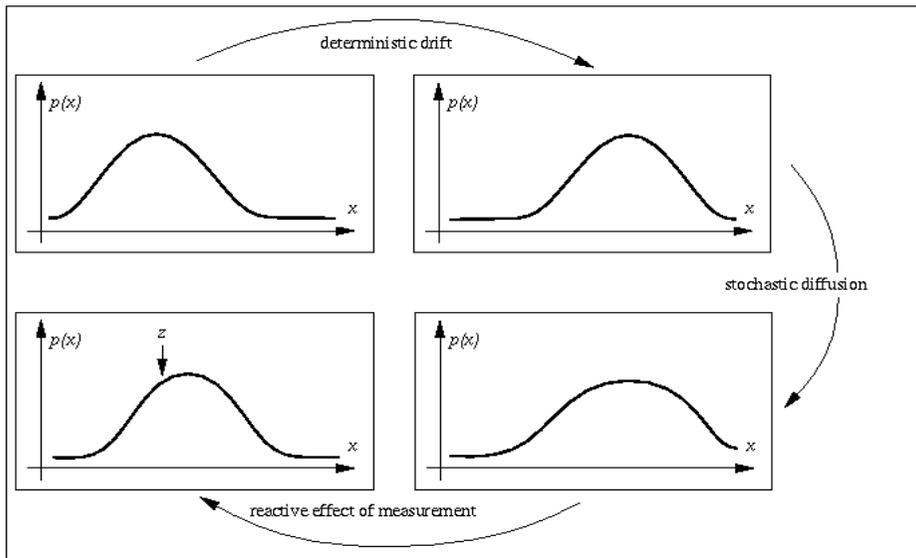
# Riassunto grafico



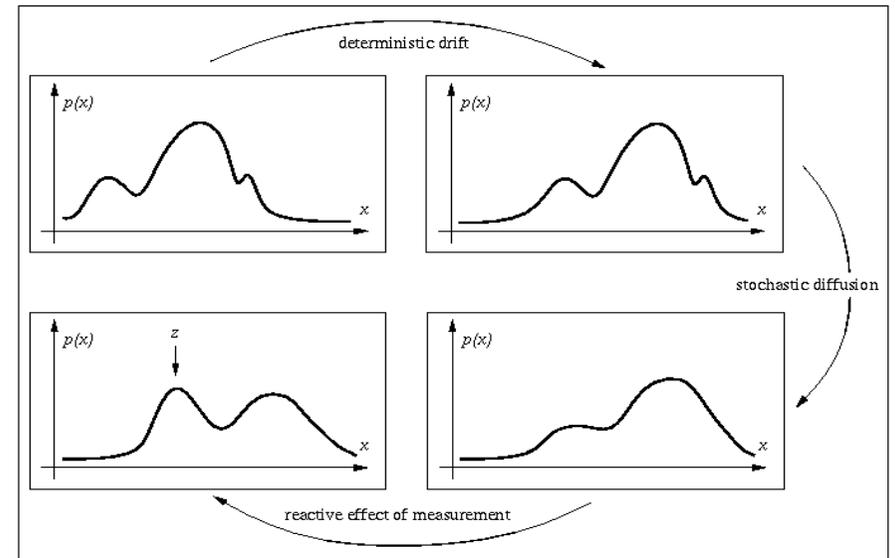
# Dimostrazione



# Richiamo al Kalman Filter



Kalman



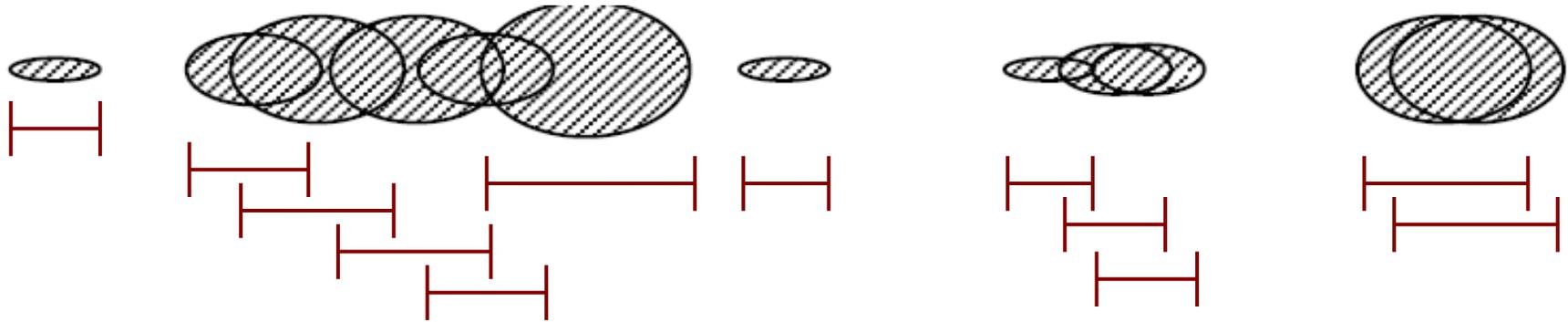
Condensation



# Dettagli: come campionare?

- Metodo della funzione di ripartizione

- Prendo tutti i pesi dei campioni  $\{\pi_t^{(n)}\}$



- Ne faccio la somma cumulativa, ottenendo i coeff.

$$\{c_t^{(n)}\}$$



# Dettagli: come campionare?

– Ossia

$$\begin{aligned}c_t^{(0)} &= 0, \\c_t^{(n)} &= c_t^{(n-1)} + \pi_t^{(n)} \quad (n = 1, \dots, N).\end{aligned}$$

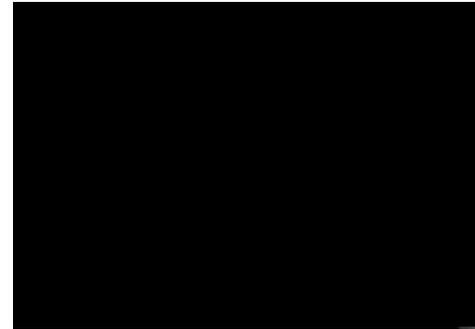
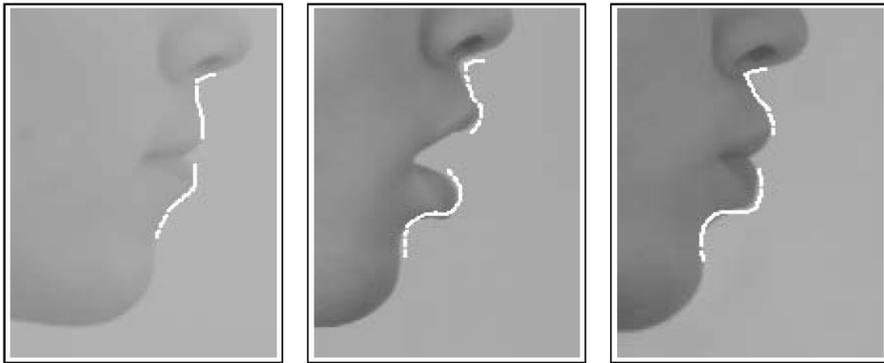
– Dopodiché (in modo efficiente –  $O(N \log N)$ )

1. generate a random number  $r \in [0, 1]$ , uniformly distributed.
2. find, by binary subdivision, the smallest  $j$  for which  $c_{t-1}^{(j)} \geq r$
3. set  $\mathbf{s}_t^{(n)} = \mathbf{s}_{t-1}^{(j)}$



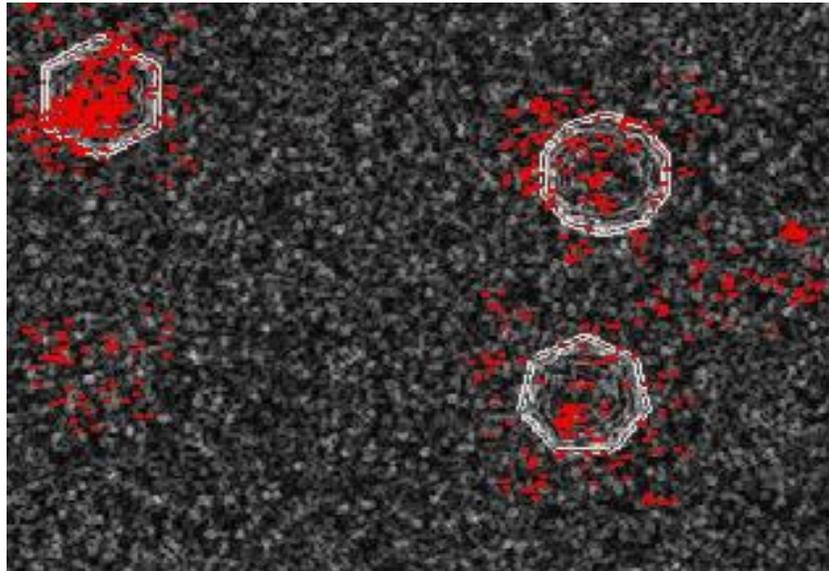
# Stato del sistema:

- Non esclusivamente una posizione 1D:
  - Posizione multidimensionale
  - Setdi coefficienti di B-spline, per modellare una forma



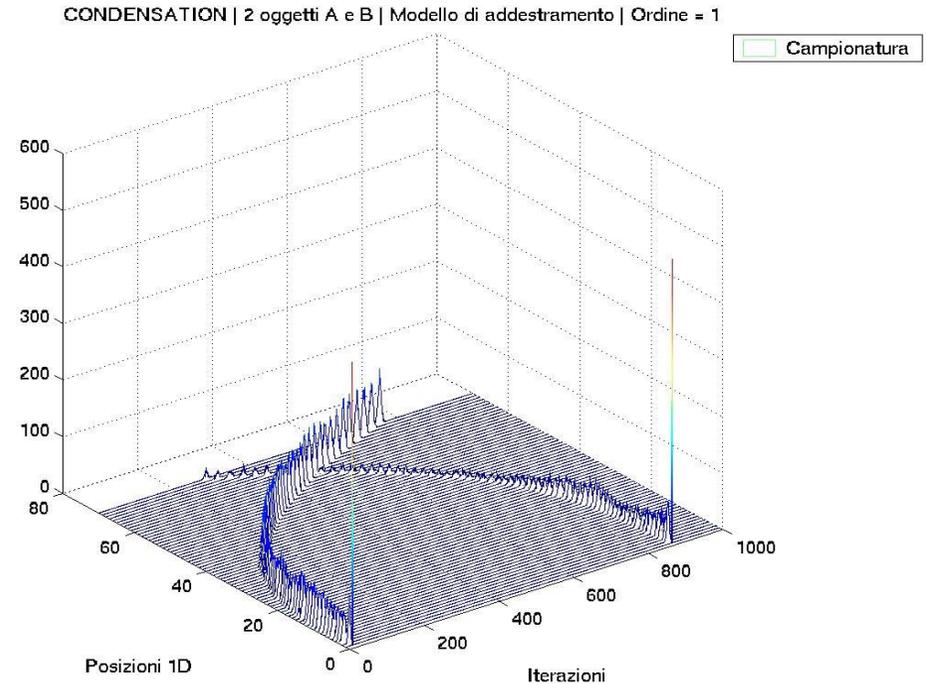
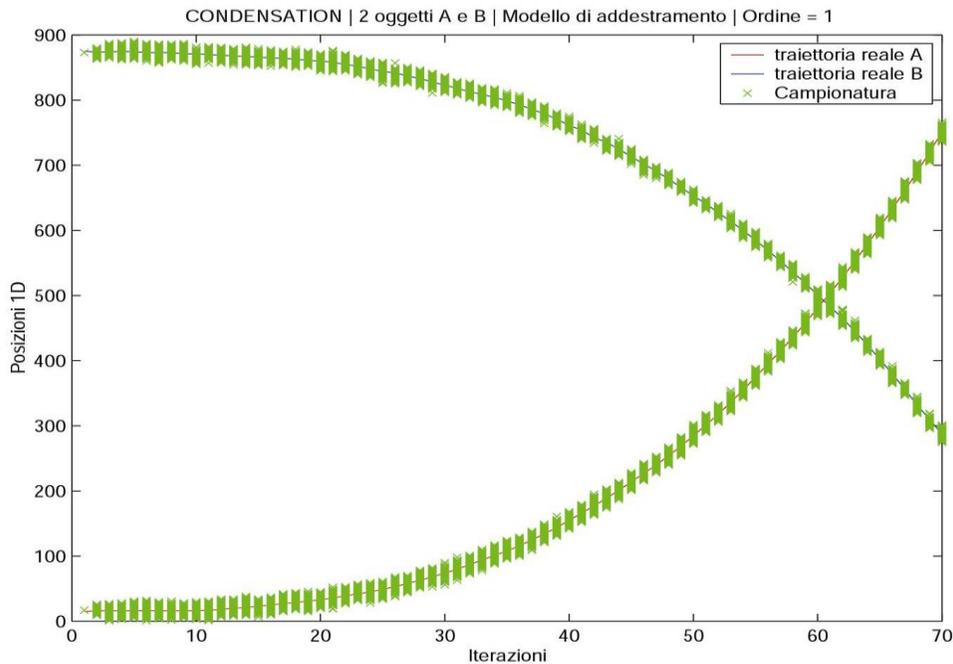
# Note – multi-object tracking

- Cosa accade nel caso in cui ci siano piu' di un oggetto da inseguire?



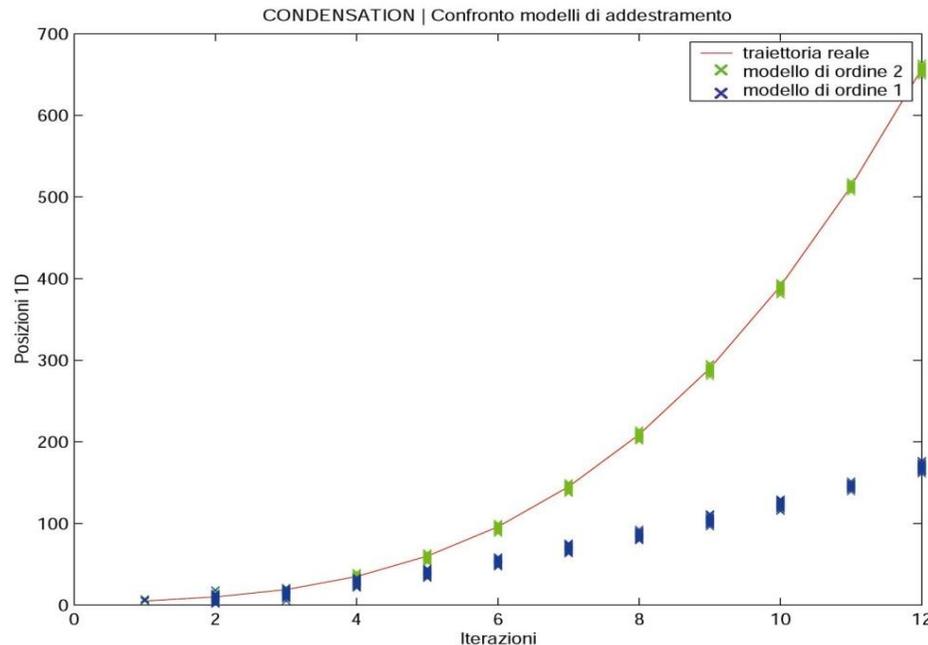
# Note – multi-object tracking

- L'algoritmo tende effettivamente a seguire un solo oggetto.



# Importanza della dinamica

- Una dinamica errata porta a perdere l'oggetto



- Soluzione: si stima la dinamica in modo robusto, off-line
  - manualmente, via correlazione esaustiva su tutto il piano immagine



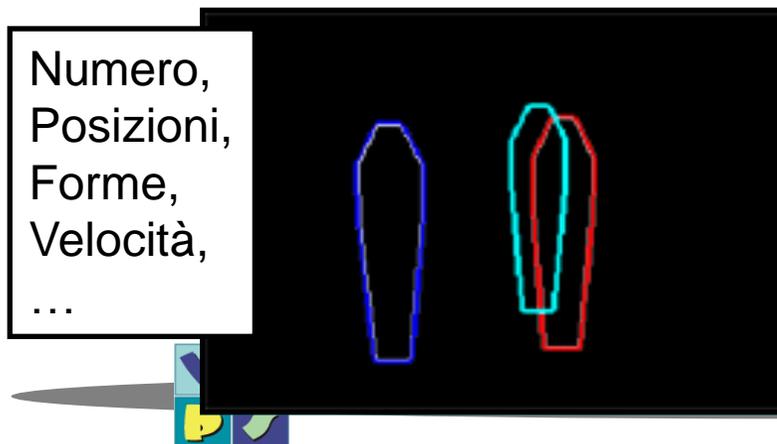
# Condensation multi oggetto - Bramble

- Bayesian Multiple-BLob (BraMBLe) tracker è l'evoluzione di Condensation.
- Al solito, stima

$$p(\mathbf{X}_t \mid \mathbf{Z}_t, \mathbf{Z}_{t-1}, \dots, \mathbf{Z}_1)$$

Stato all'istante  $t$

Sequenza di immagini



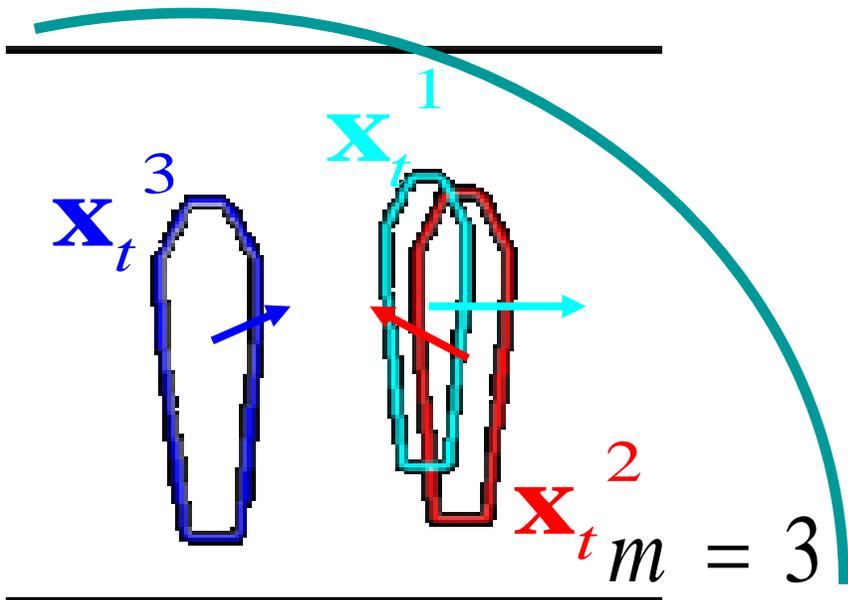
# Bramble - ingredienti

- Stato del sistema  $\mathbf{X}$

$$\mathbf{X}_t = \left( m, \quad \mathbf{x}_t^1, \quad \dots, \quad \mathbf{x}_t^m \right)$$

Numero di oggetti

Stato  
dell'oggetto

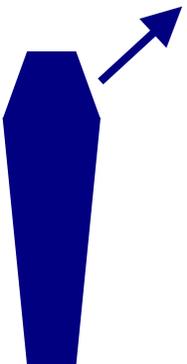


# Bramble - ingredienti

Stato dell'oggetto  $\mathbf{X} = (i, \mathbf{l}, \mathbf{v}, \mathbf{s})$

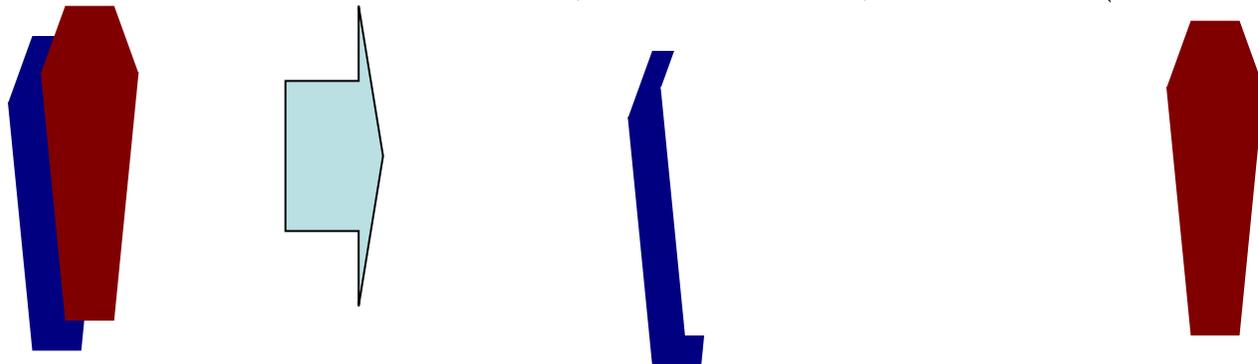
↓                      ↓                      ↓                      ↓

Identità                      Localazione                      Velocità                      Appearance



- Gestione delle occlusioni (grazie al 3D)

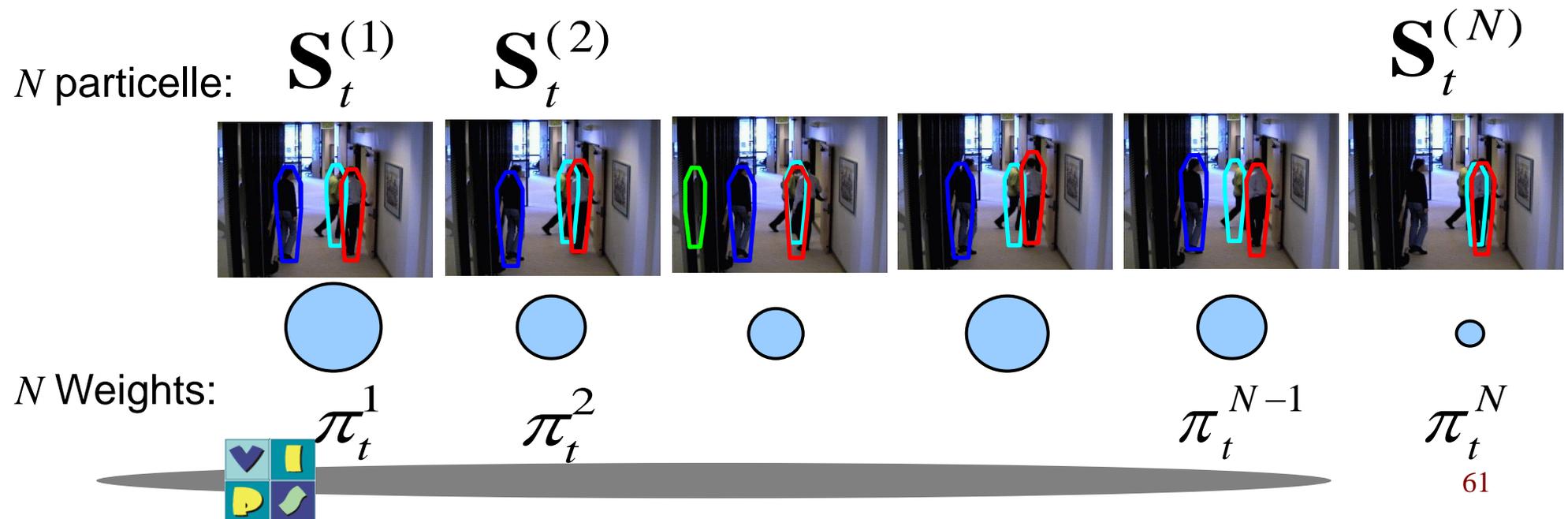
$$\mathbf{x}^1 = (i, \mathbf{l}, \mathbf{v}, \mathbf{s}) \quad \mathbf{x}^2 = (i, \mathbf{l}, \mathbf{v}, \mathbf{s})$$



# Bramble - ingredienti

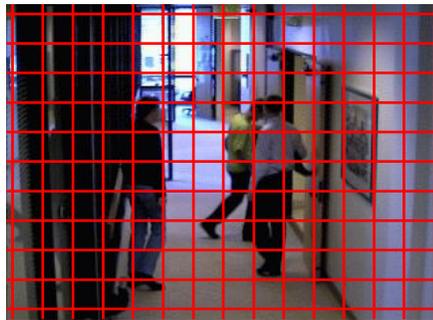
- Lo stato del sistema si approssima con un set di  $N$  particelle-configurazioni

$$\mathbf{X}_t \approx \left\{ \mathbf{S}_t^{(n)}, \pi_t^{(n)} \right\}$$



# Bramble - ingredienti

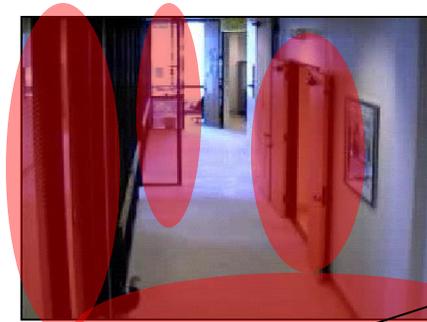
- Osservazioni **Z**
  - sottocampiono la matrice video in un set di  $G$  filtri  $z_g$  con una granularità spaziale sufficiente a poter valutare tutti i filtri in tempo reale
  - Ogni filtro corrisponde ad una determinata posizione 3D



# Bramble - fasi

**1. Selezione o campionamento** : la stessa di Condensation: si estraggono configurazioni grazie ai pesi. In più

- Ogni oggetto ha una probabilità variabile (in funzione della posizione) e costante rispetto al tempo di uscire dalla scena,  $\tau_{out}(l)$
- Esiste una probabilità variabile (in funzione della posizione) e costante rispetto al tempo di entrare nella scena  $\tau_{in}(l)$



alta probabilità



# Bramble - fasi

- 2. Applicazione della dinamica:** simile a Condensation  
– la dinamica di ogni oggetto di ogni configurazione è determinata in base
1. alla sua etichetta
  2. alla storia precedente (media pesata di tutte le particelle) di quella etichetta



# Bramble - fasi

## 3. Valutazione/pesatura

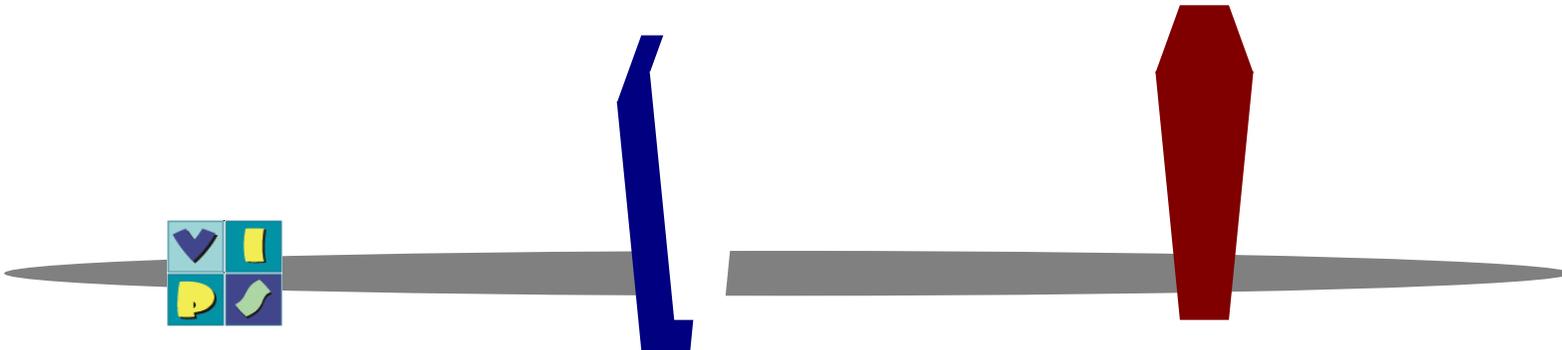
- La valutazione di una configurazione avviene controllando sequenzialmente i filtri, assunti indipendenti tra loro

$$p(\mathbf{Z} | S^{(n)}) = \prod_g p(z_g | S^{(n)}).$$

la produttoria si cambia in sommatoria logaritmica per stabilità numerica

- la valutazione tiene presente della posizione

$$\mathbf{x}^1 = (i, \mathbf{l}, \mathbf{v}, \mathbf{s}) \quad \mathbf{x}^2 = (i, \mathbf{l}, \mathbf{v}, \mathbf{s})$$



# Mapping libro di testo

- Stima non parametrica: Cap. 4 libro Duda
- Tracking: Cap. 3 Tesi di Isard (Allegato alle slide)

