

Analisi di Immagini e Video (Computer Vision)

Giuseppe Manco

Outline

- Segmentation
- Approcci classici
- Deep Learning for Segmentation

Crediti

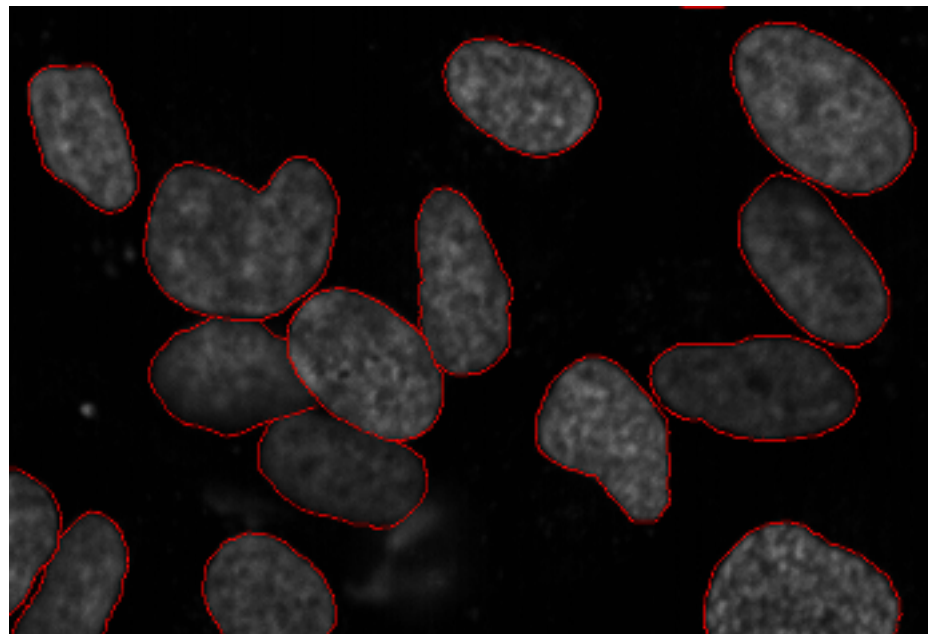
- Slides adattate da vari corsi e libri
 - Computational Visual Recognition (V. Ordonez), CS Virginia Edu
 - Computer Vision (S. Lazebnik), CS Illinois Edu
 - Computer Vision (I. Gkioulekas) - CS CMU Edu

Segmentation

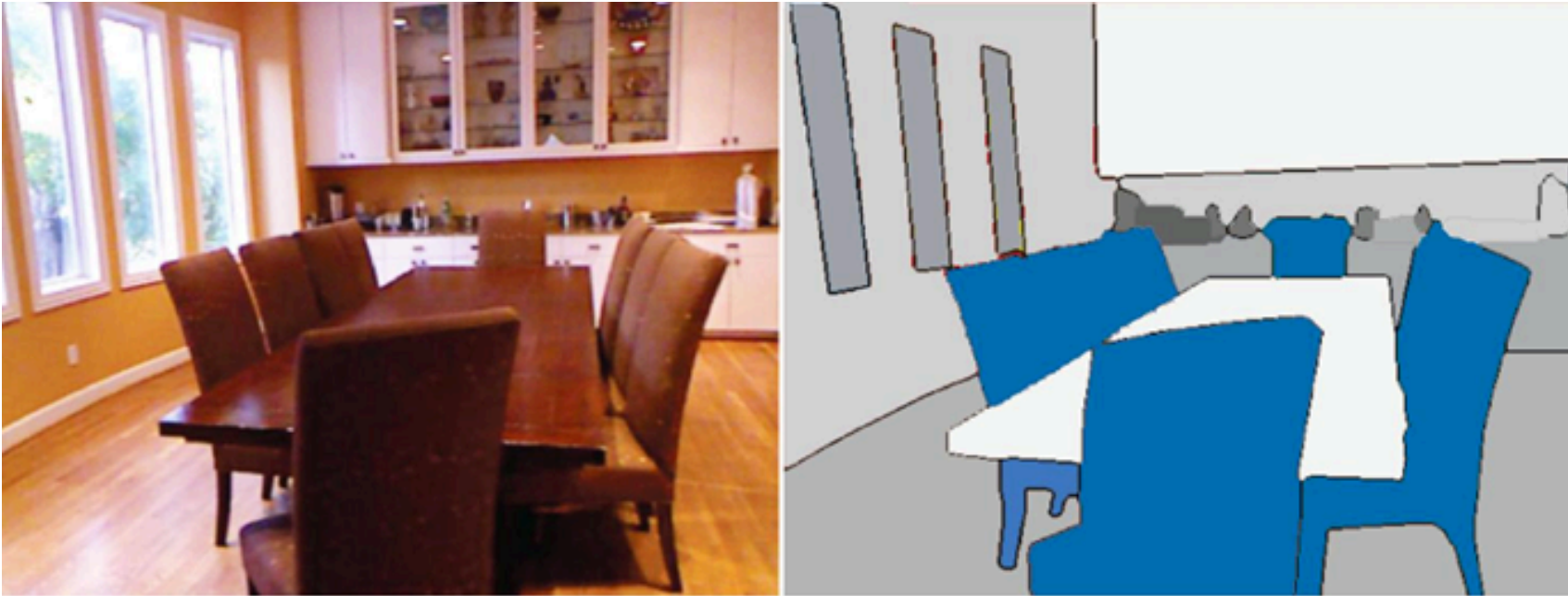
- Gruppi di pixel che formano regioni omogenee
 - Superfici
 - Oggetti
 - Parti di oggetti
- Separazione Foreground/background
- Semantic segmentation
- Instance segmentation

Separazione foreground-background

- Separazione di oggetti dallo sfondo



Semantic segmentation



Instance segmentation



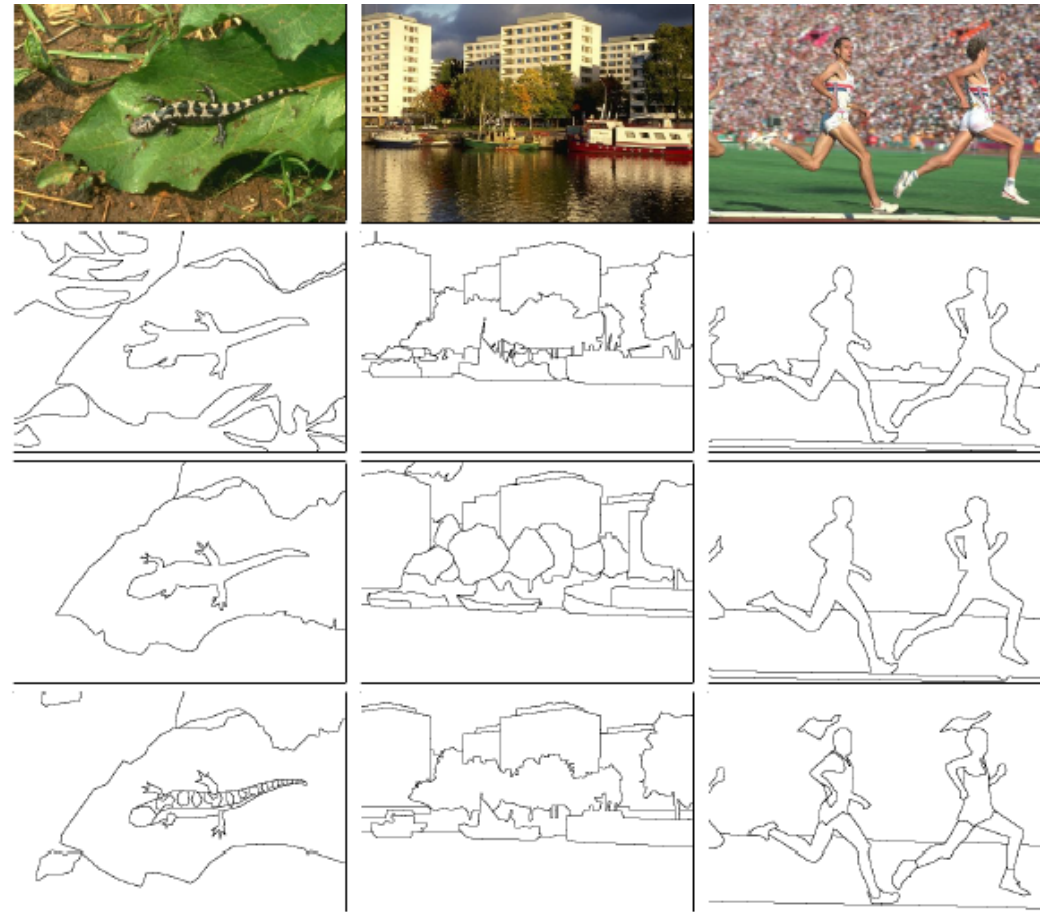
Cosa rende buona una segmentazione?

- Ground truth



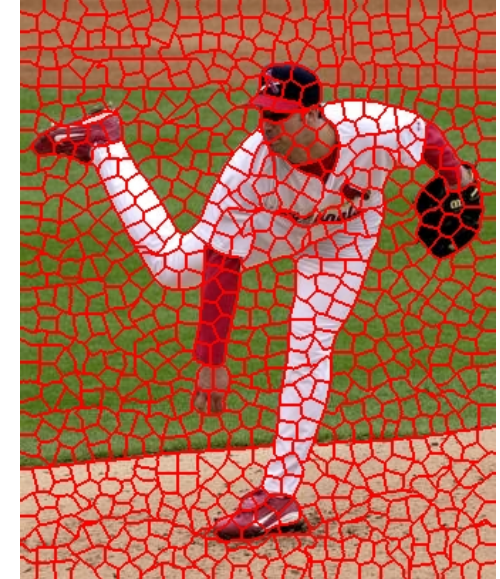
Cosa rende buona una segmentazione?

- Ground truth
 - Non necessariamente!



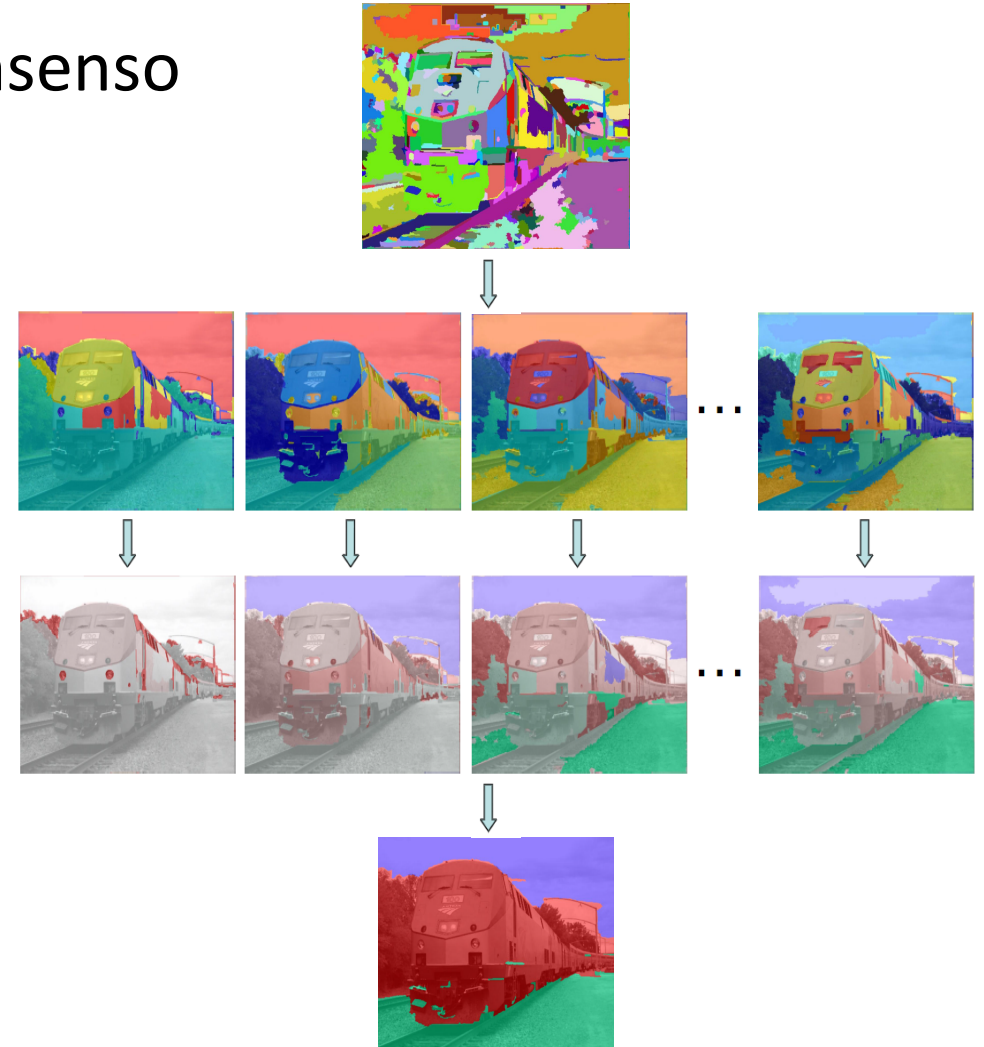
Superpixel segmentation

- Raggruppamenti a granularità grossa
 - Bottom-up
 - Non supervisionato
 - veloce
 - Oversegmentation
 - Ci basta che le regioni siano uniformi



Multiple segmentation

- Istanze multiple, etichettatura tramite consenso



Approccio

- Data un'immagine $I(\vec{x})$ caratterizzata da

$$\vec{F}(\vec{x}) = \begin{pmatrix} \vec{x} \\ I(\vec{x}) \\ \vec{L}(\vec{x}) \end{pmatrix}$$

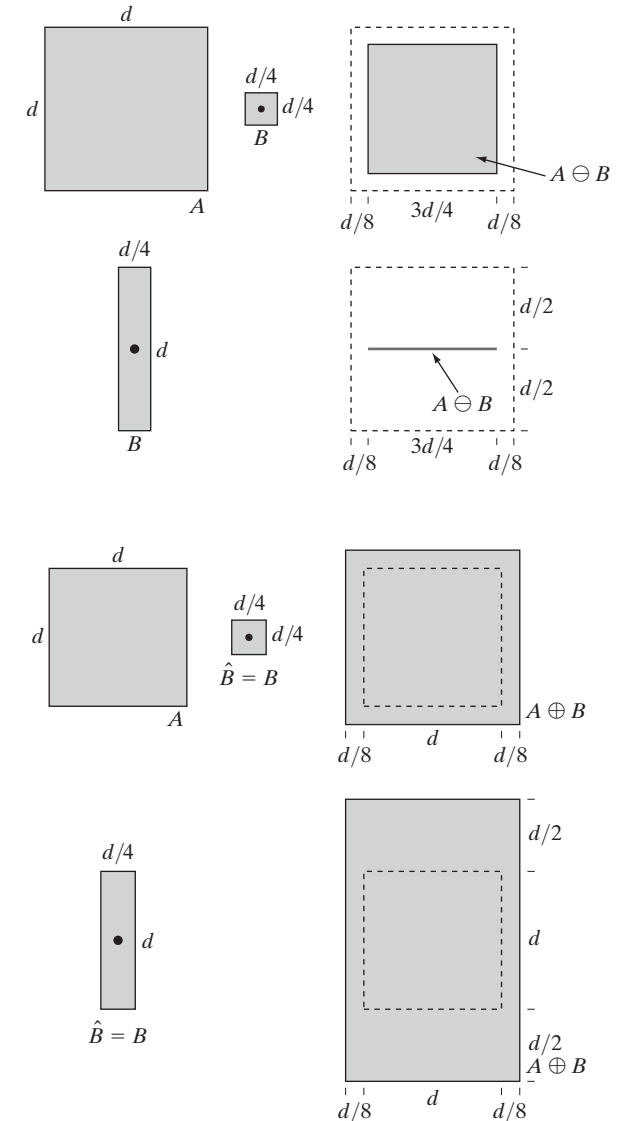
- $\vec{L}(\vec{x})$ è un insieme di features locali
 - Ad esempio la risposta a filtri passa-banda
- Trovare i raggruppamenti di \vec{x}

Tecniche

- Non supervisionati, Bottom-up
 - Region-based
 - Edge-detection
 - Clustering
 - Graph-based
- Supervisionati, top-down
 - CRFs
 - Deep Learning

Thresholding e Morfologia

- Su immagini in scala di grigio
 - Thresholding:
 - $\theta(I, t) = \begin{cases} 1 & I > t \\ 0 & \end{cases}$
 - B structuring element
 - (un filtro binario di dimensione S)
 - $C = I * B$
 - Convoluzione di «conteggio»
 - Erosion
 - $A \ominus B = \theta(C, S)$
 - Dilation
 - $A \oplus B = \theta(C, 1)$
 - Opening
 - $(A \ominus B) \oplus B$
 - Closing
 - $(A \oplus B) \ominus B$



Thresholding

- Su immagini in scala di grigio

- Thresholding:

- $\theta(I, t) = \begin{cases} 1 & I > t \\ 0 & \end{cases}$

- Determinazione della soglia ottimale

- Otsu algorithm, basato sulla massimizzazione della varianza inter-segmento

- Soglie multiple

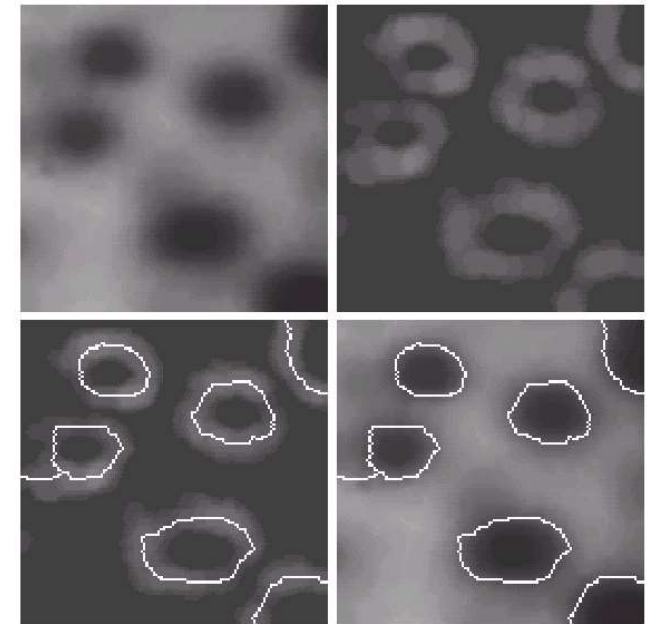
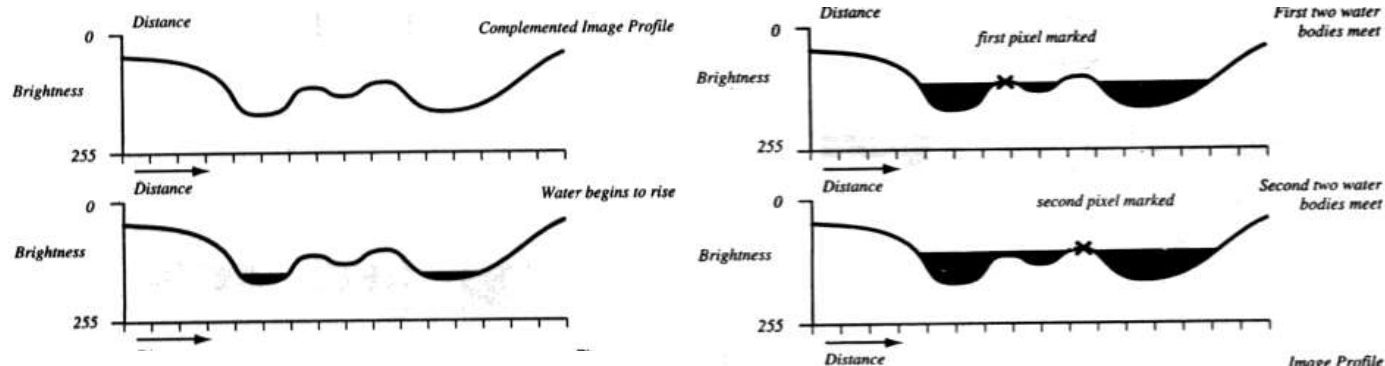
- Partizionamento
 - neighborhood

Edge Detection

- Metodi basati sull'individuazione di cambiamenti di intensità
 - Image smoothing per la riduzione del rumore
 - Detection di edge points
 - Individuazione di potenziali candidati, tramite filtri laplaciani o di gradiente
 - Thresholding, canny edge detection
 - Edge localization
 - Selezione dai candidati
 - Edge linking

Watershed segmentation

- Linee spartiacqua
 - Aumentiamo progressivamente e uniformemente il «livello dell'acqua»
 - Quando due bacini si fondono, marchiamo i pixel di separazione
 - Le linee spartiacqua sono costruite su tali linee
- Applicato sul gradiente



Clustering

- Raggruppiamo sulla base delle affinità di $\vec{F}(\vec{x})$
 - Esempi: k-means (Mixture of Gaussians)

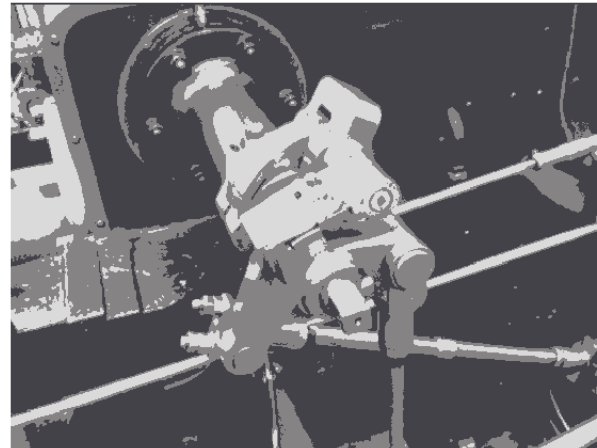
$$p(\vec{F}(\vec{x})|M) = \sum_k \pi_k p(\vec{F}(\vec{x})|\vec{m}_k, \Sigma_k)$$

- La segmentazione è stabilita dai parametri $\{\pi_k, \vec{m}_k, \Sigma_k\}_{k=1\dots K}$
 - numero di segmenti stabilito da penalized likelihood
- Segmento di appartenenza:

$$c(\vec{x}) = \operatorname{argmax}_k p(\vec{F}(\vec{x})|\vec{m}_k, \Sigma_k)$$

Clustering

- La qualità del clustering dipende dalla veridicità delle assunzioni
 - Distribuzioni gaussiane



Mean-Shift

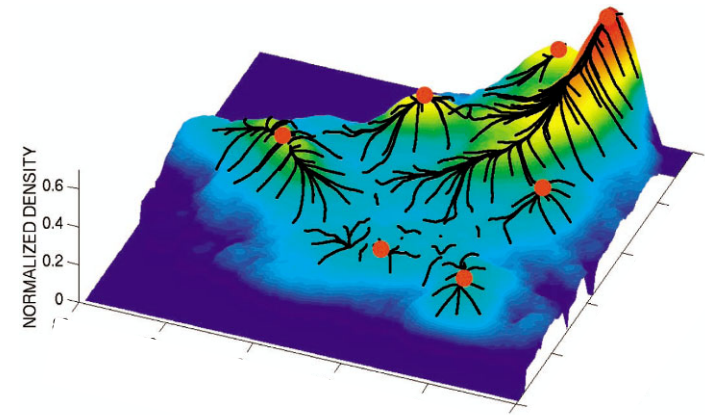
- Variante non-parametrica al clustering
- Utilizza Kernel-density estimation
 - $f(x) = 1/M \sum K(x - x_i)$
- Ogni punto è associato al minimo locale più vicino
 - Utilizzando gradient ascent, si può convergere
 - Calcolo ottimizzato:

$$y_{k+1} = \frac{\sum_i x_i K'(y_k - x_i)}{\sum_i K'(y_k - x_i)}$$

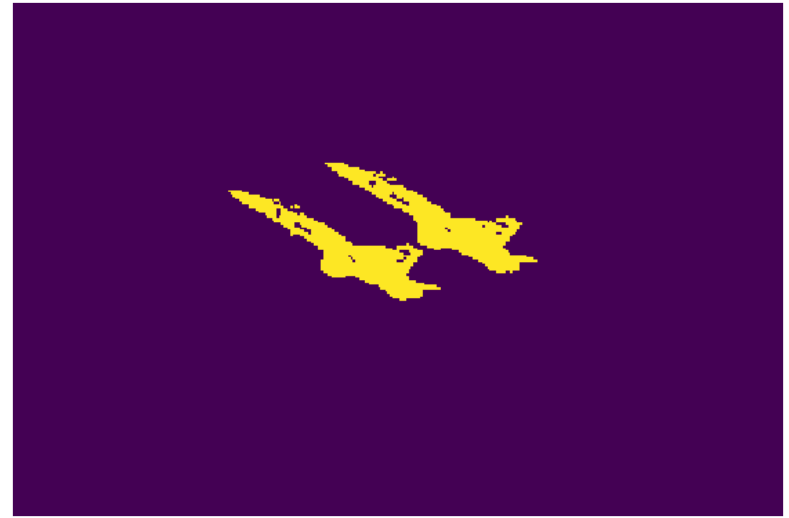
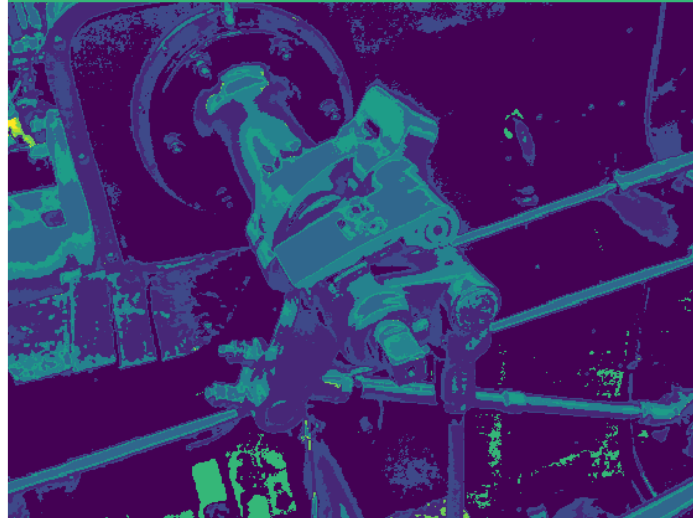
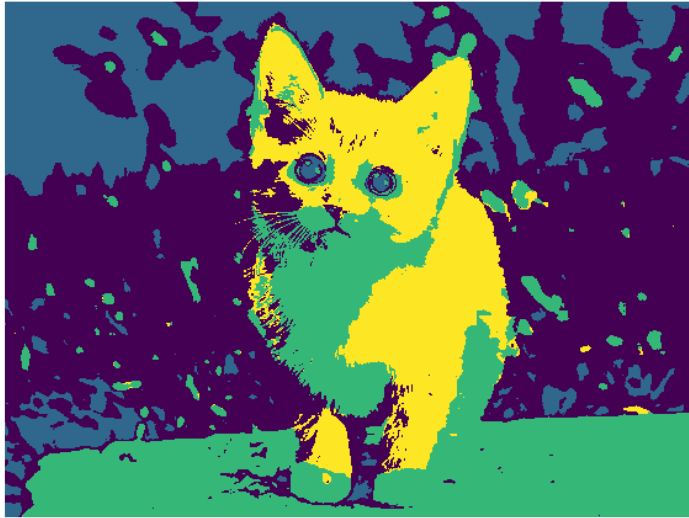
- converge su kernel gaussiani o Epanechnikov

$$K_G(r; h) = \exp(-r^2/2h)$$

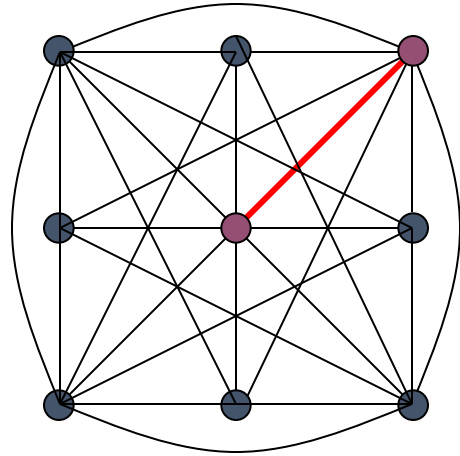
$$K_E(r; h) = \max(0, 1 - r/h)$$



Mean Shift

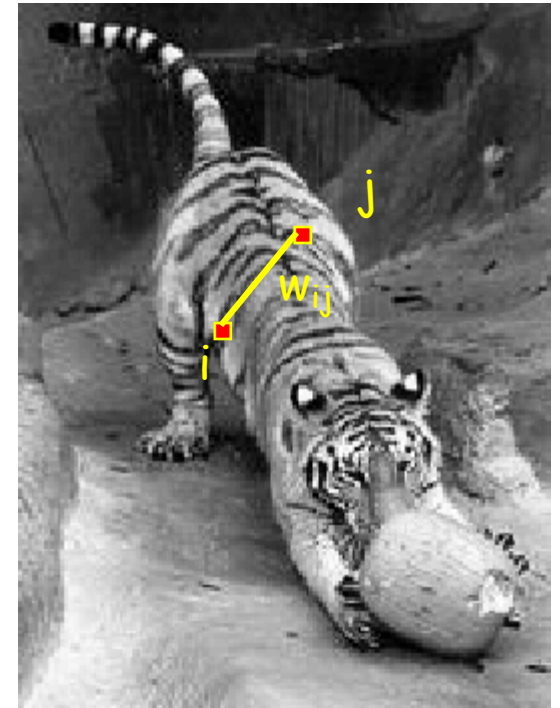


Graph-based segmentation

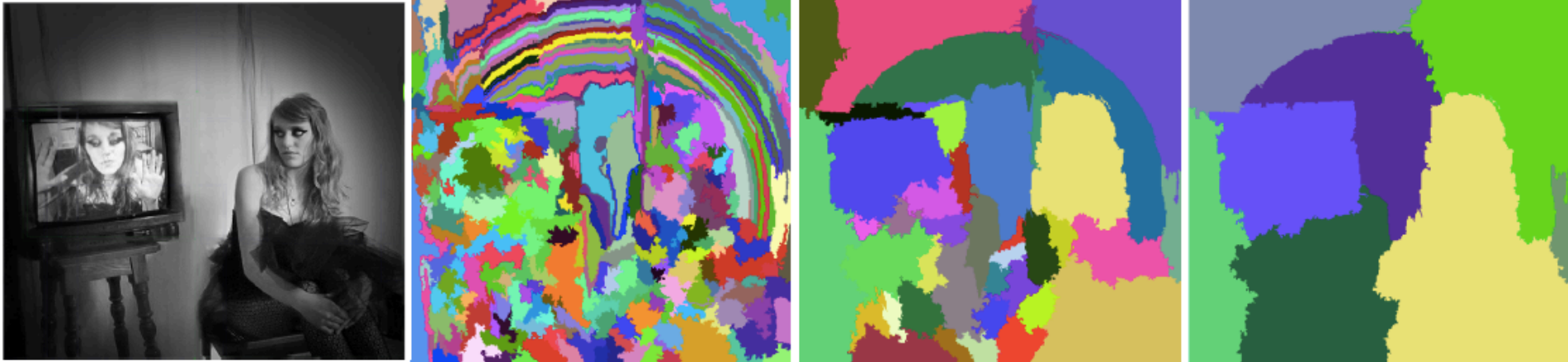


Un grafo pesato indiretto connesso

- **Nodi** = pixel
- **archi** = coppie di pixel vicini
- **Pesi sugli archi** = similarità dei nodi



Esempio



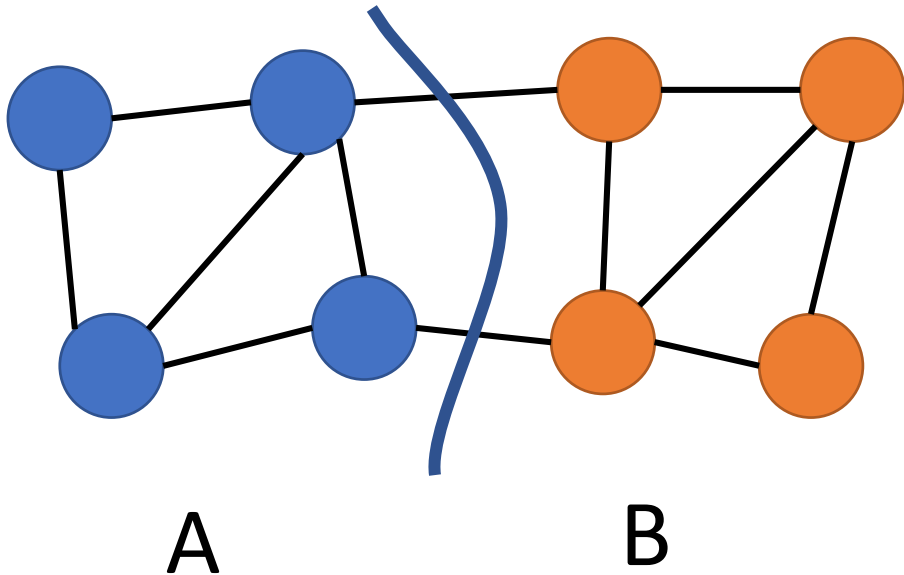
Felzenszwalb & Huttenlocher algorithm

- I pesi corrispondono alla dissimilarità nello spazio (x, y, r, g, b)
- L'algoritmo
 - Ogni vertice rappresenta una componente
 - Per ogni arco in ordine crescente di peso:
 - Se l'arco è tra vertici in due componenti A e B, fonda le componenti se il peso è minore della minima differenza interna intra-componente

$$Int(R) = \min_{e \in MST(R)} w(e) + k/|R|$$

Altri approcci: Graph-cut

- Si cerca il taglio ottimale cancellando gli archi che minimizzano il cut



$$Cut(A, B) = \sum_{i \in A, j \in B} w_{ij}$$
$$NCut(A, B) = \frac{Cut(A, B)}{\sum_{i \in A, j \in V} w_{ij}} + \frac{Cut(A, B)}{\sum_{i \in B, j \in V} w_{ij}}$$

- NP-Hard
- Soluzione tramite rilasciamento continuo

Probabilistic Graphical Model

- Formulazione probabilistica
 - $p(x, y)$ con y label e x punto nell'immagine
- Idea: formulare una funzione di costo che dipenda dalla quantità di «energia» individuata nell'etichettatura dei punti

$$E(Y|X) = \sum_i \phi(y_i|x_i; \theta) + \sum_{i,j} \psi(y_i, y_j|x_i, x_j; \theta)$$

Pixels

Unary potential (local data term): score for pixel i and label c_i

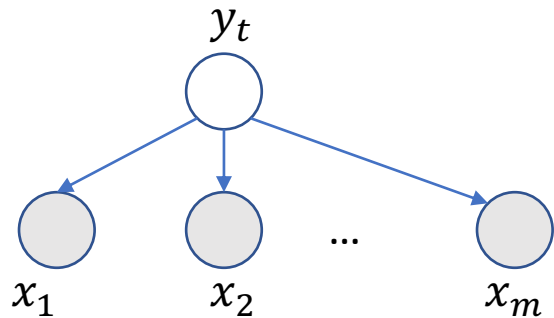
Neighboring pixels

Pairwise potential (context or smoothing term)

Dependencies, random fields

- Naive Bayes:

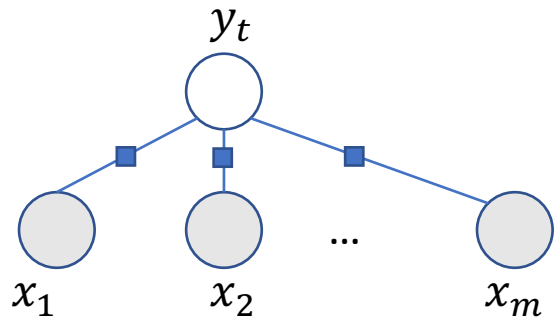
$$p(y|x) = \frac{\prod_i p(x_i|y)p(y)}{\sum_{\bar{y}} \prod_i p(x_i|\bar{y})p(\bar{y})} \rightarrow p(y = 1|x) = \frac{1}{1 + \exp \left\{ \log \frac{p(y = 1)}{1 - p(y = 0)} + \sum_i \log \frac{p(x_i|y = 1)}{p(x_i|y = 0)} \right\}}$$



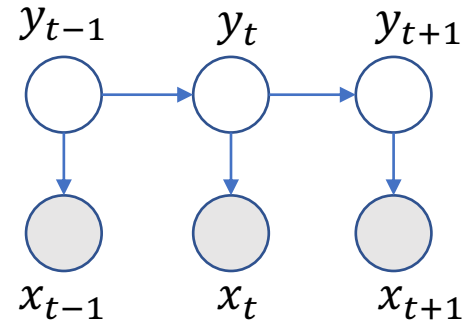
Dependencies, random fields

- Logistic regression:

$$p(y = 1|x) = \frac{1}{1 + \exp\{w_0 + \sum_i w_i x_i\}}$$

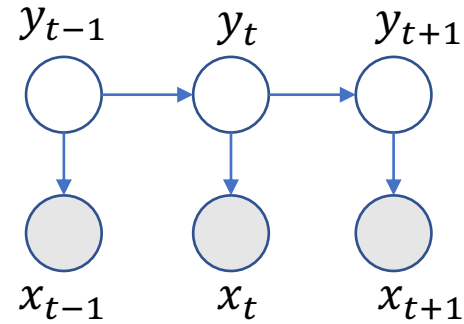


Hidden Markov Models



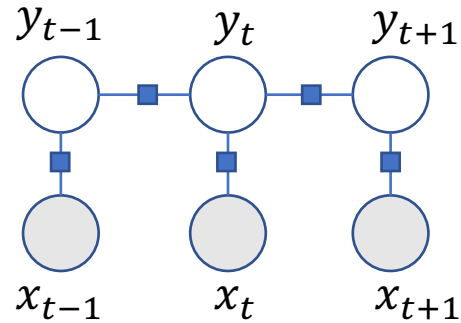
- Joint likelihood
 - $p(Y, X) = \prod_t p(x_t | y_t) p(y_t | y_{t-1})$
- Estensione «markoviana» di Naive Bayes
- Inferenza
 - Qual è l'insieme Y di etichette ottimale?
 - Qual è la probabilità $p(y_i | X)$?

Hidden Markov Models



$$\log p(Y, X) = \sum_t \{\log p(x_t | y_t) + \log p(y_t | y_{t-1})\}$$

Conditional Random Field



$$E(Y|X) = \sum_i \phi(y_i|x_i; \theta) + \sum_{i,j} \psi(y_i, y_j|x_i, x_j; \theta)$$

Pixels

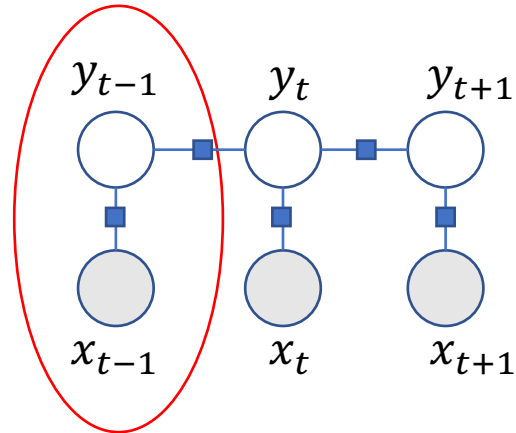
Unary potential (local data term): score for pixel i and label c_i

Neighboring pixels

Pairwise potential (context or smoothing term)

$$p(Y|X) = \frac{1}{Z} \exp\{E(Y|X)\}$$

Conditional Random Field



$$E(Y|X) = \sum_i \phi(y_i|x_i; \theta) + \sum_{i,j} \psi(y_i, y_j|x_i, x_j; \theta)$$

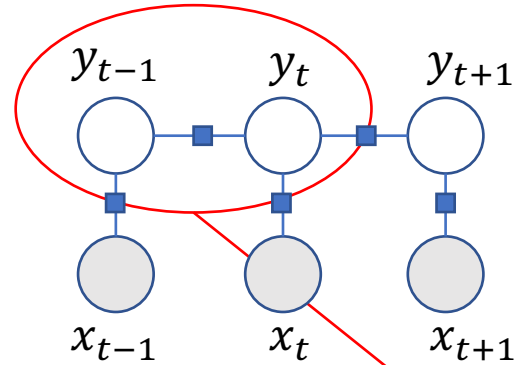
Pixels

Unary potential (local data term): score for pixel i and label c_i

Neighboring pixels

Pairwise potential (context or smoothing term)

Conditional Random Field



$$E(Y|X) = \sum_i \phi(y_i, x_i; \theta) + \sum_{i,j} \psi(y_i, y_j, x_i, x_j; \theta)$$

Pixels

Unary potential (local data term): score for pixel i and label c_i

Neighboring pixels

Pairwise potential (context or smoothing term)

Conditional Random Fields

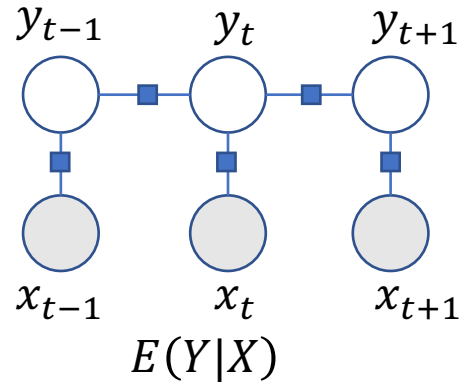
- $\phi(y_i|x_i; \theta) = -\log p(y_i|x_i; \theta)$
- $\psi(y_i, y_j|x_i, x_j; \theta) = w_{ij} \mathbf{1}_{y_i \neq y_j}$
 - $w_{ij} = \exp\left\{-\frac{|F(x_i) - F(x_j)|^2}{\sigma}\right\}$

$$E(Y|X) = \sum_i \phi(y_i, x_i; \theta) + \sum_{i,j} \psi(y_i, y_j, x_i, x_j; \theta)$$

Unary potential
Pixels (local data term):
score for pixel i
and label c_i

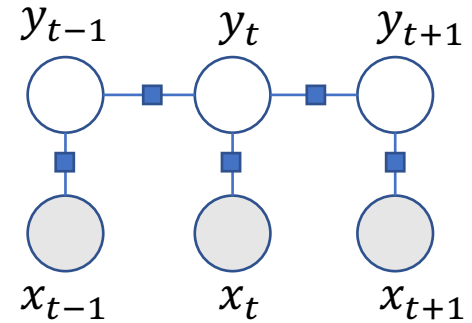
Pairwise potential
Neighboring
pixels
(context or smoothing
term)

Conditional Random Fields



- $p(Y|X) = \frac{1}{Z(X)} \exp\{E(Y|X)\}$
 - $E(Y|X) = \sum_i \phi(y_i|x_i; \theta) + \sum_{i,j} \psi(y_i, y_j|x_i, x_j; \theta)$
 - $Z(X) = \sum_Y \exp\{E(Y|X)\}$ è una costante di normalizzazione
- Inferenza
 - Qual è il labeling Y ottimale?
 - Qual è la probabilità $p(y_i|X)$?

Inferenza e CRF



- Un problema combinatorio
 - Qual è il labeling Y ottimale?
 - Qual è la probabilità $p(y_i|X)$?
 - ...
- Exact inference
 - esponenziale
- Approximate inference
 - Belief Propagation
 - MCMC
 - Variational Inference

Message Passing/Variational Inference

- Mean-field variational inference
- Approssimiamo $p(Y|X)$ con $q(Y) = \prod_i q(y_i)$
 - Troviamo la forma di $q(Y)$ che minimizza la divergenza di Kullback-Leibler con $p(Y|X)$

$$D_{KL}(q||p) \approx -E_{Y \sim q}[E(Y|X)] - E_{Y \sim q}[\log q(Y)]$$

- Soluzione ottimale:

$$q(y_i = l) = \frac{1}{Z_i} \exp \left\{ -\phi(y_i = l, x_i; \theta) - \sum_{j \in N(i)} \sum_{l'} q(y_j = l') \psi(y_i = l, y_j = l', x_i, x_j; \theta) \right\}$$

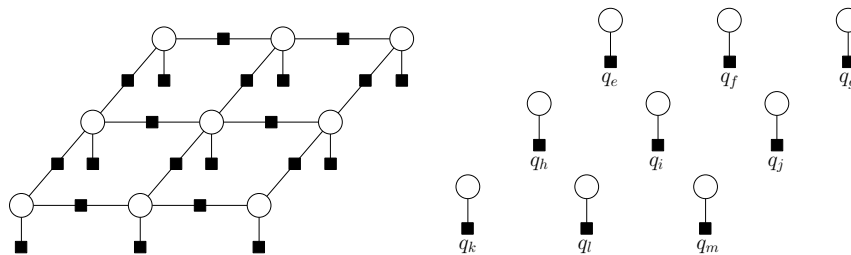
Message passing

Message Passing/Variational Inference

- Inizializza $q(Y)$
- Itera

$$q(y_i = l) = \frac{1}{Z_i} \exp \left\{ -\phi(y_i = l, x_i; \theta) - \sum_{j \in N(i)} \sum_{l'} q(y_j = l') \psi(y_i = l, y_j = l', x_i, x_j; \theta) \right\}$$

- Fino a convergenza

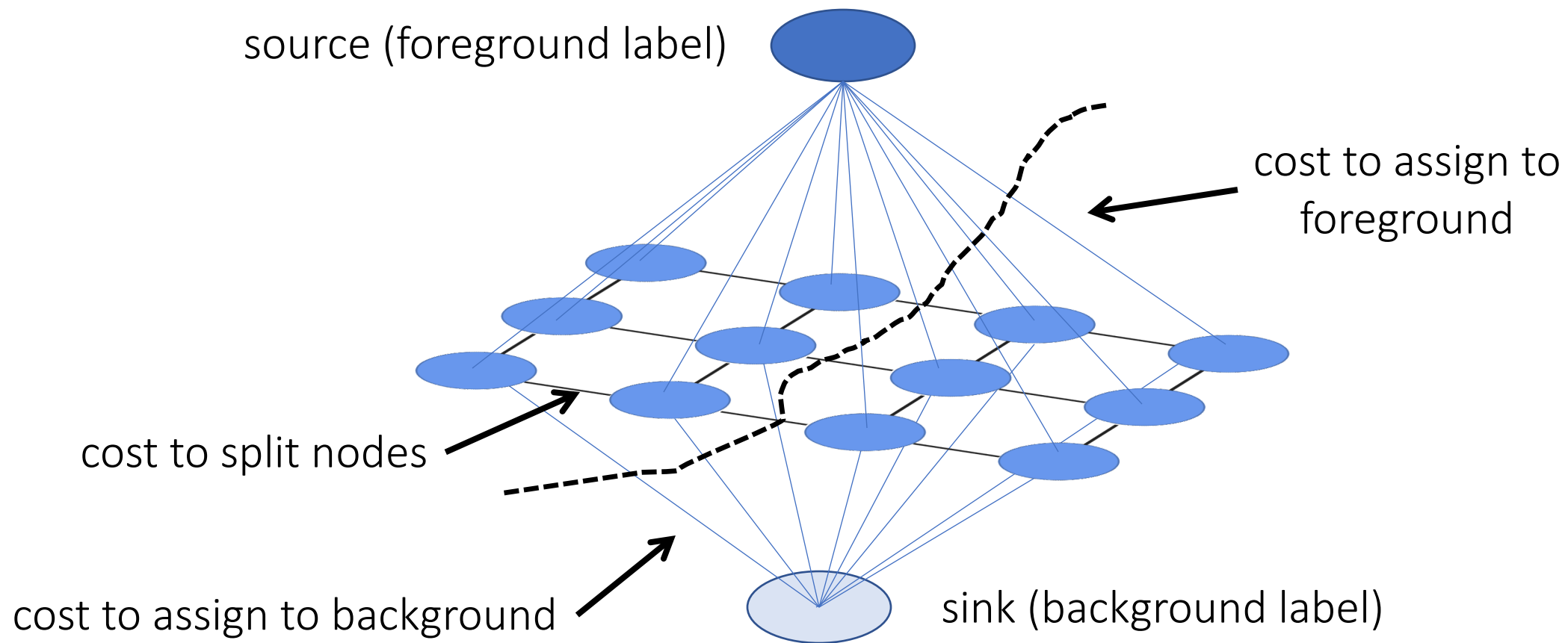


CRF e graph cuts

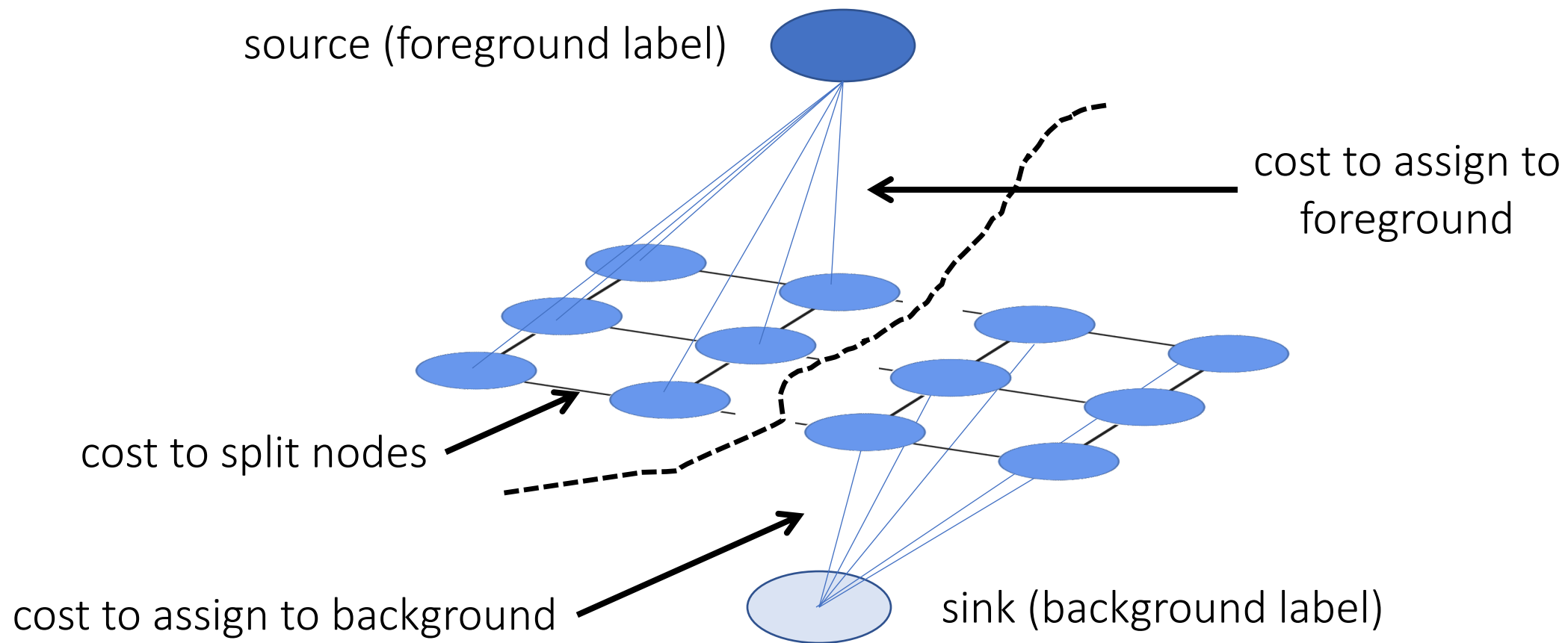
- Semi-supervisionato
 - Esplicita $\phi(y_i, x_i; \theta)$ per alcuni punti specifici

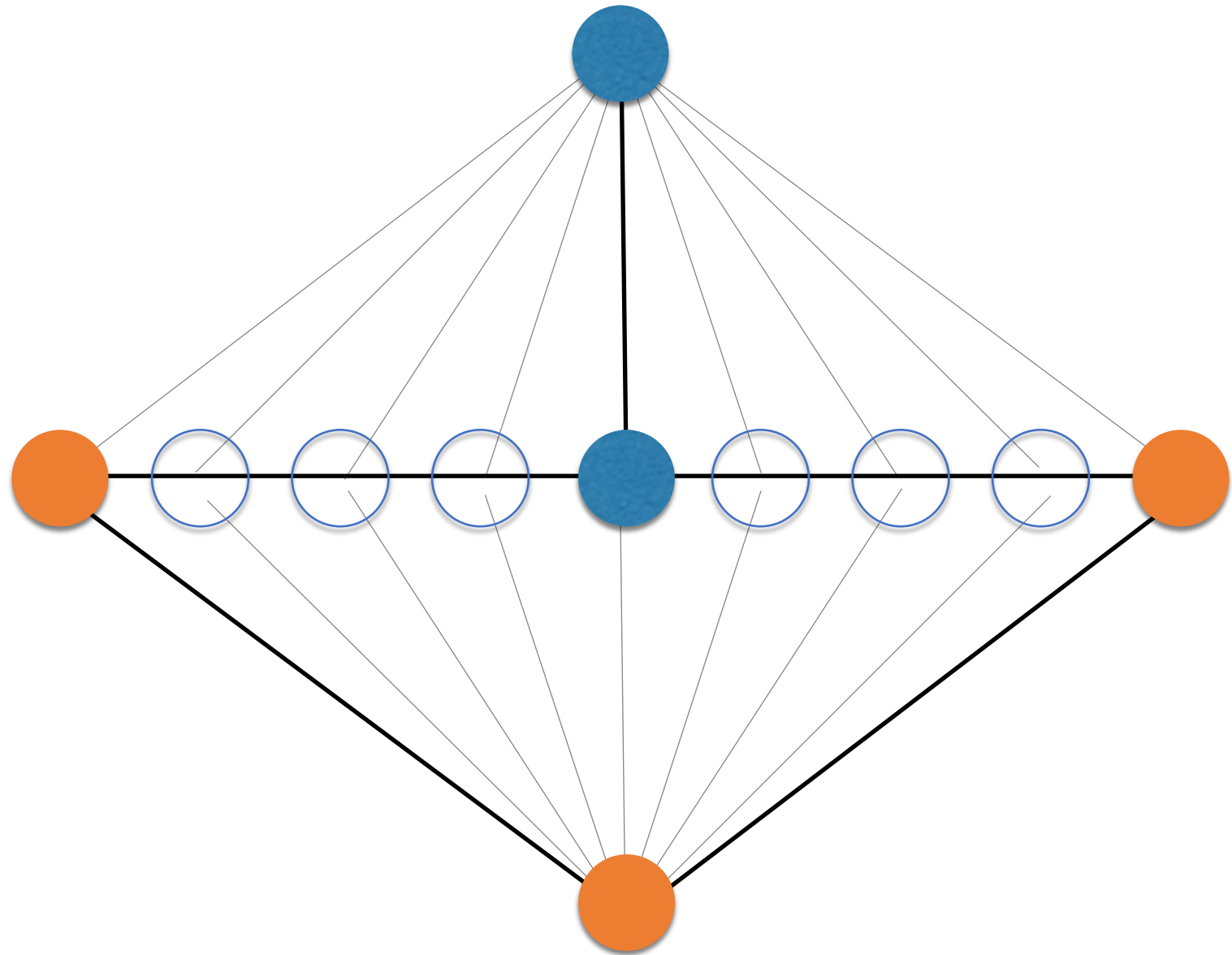


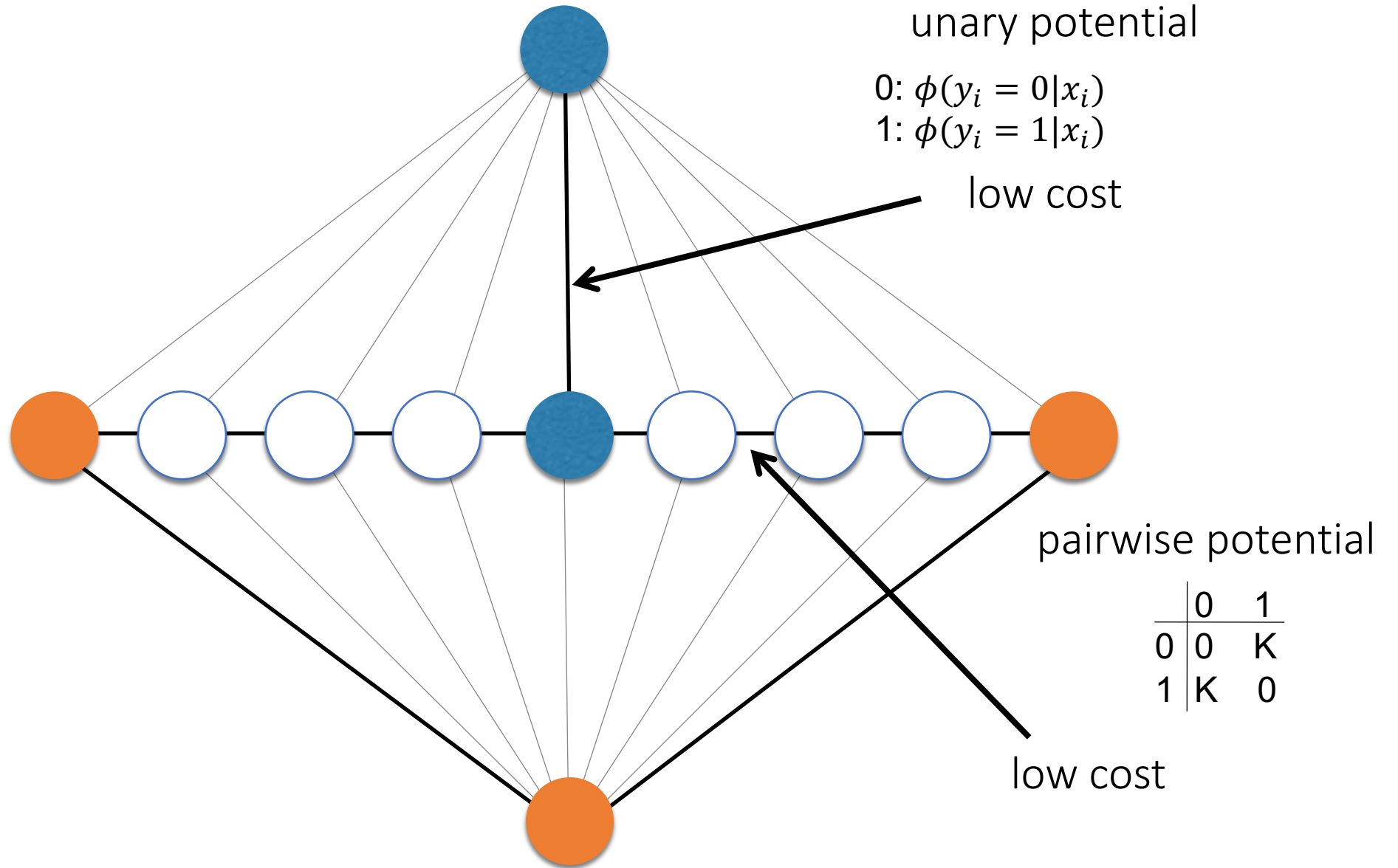
Graph cuts

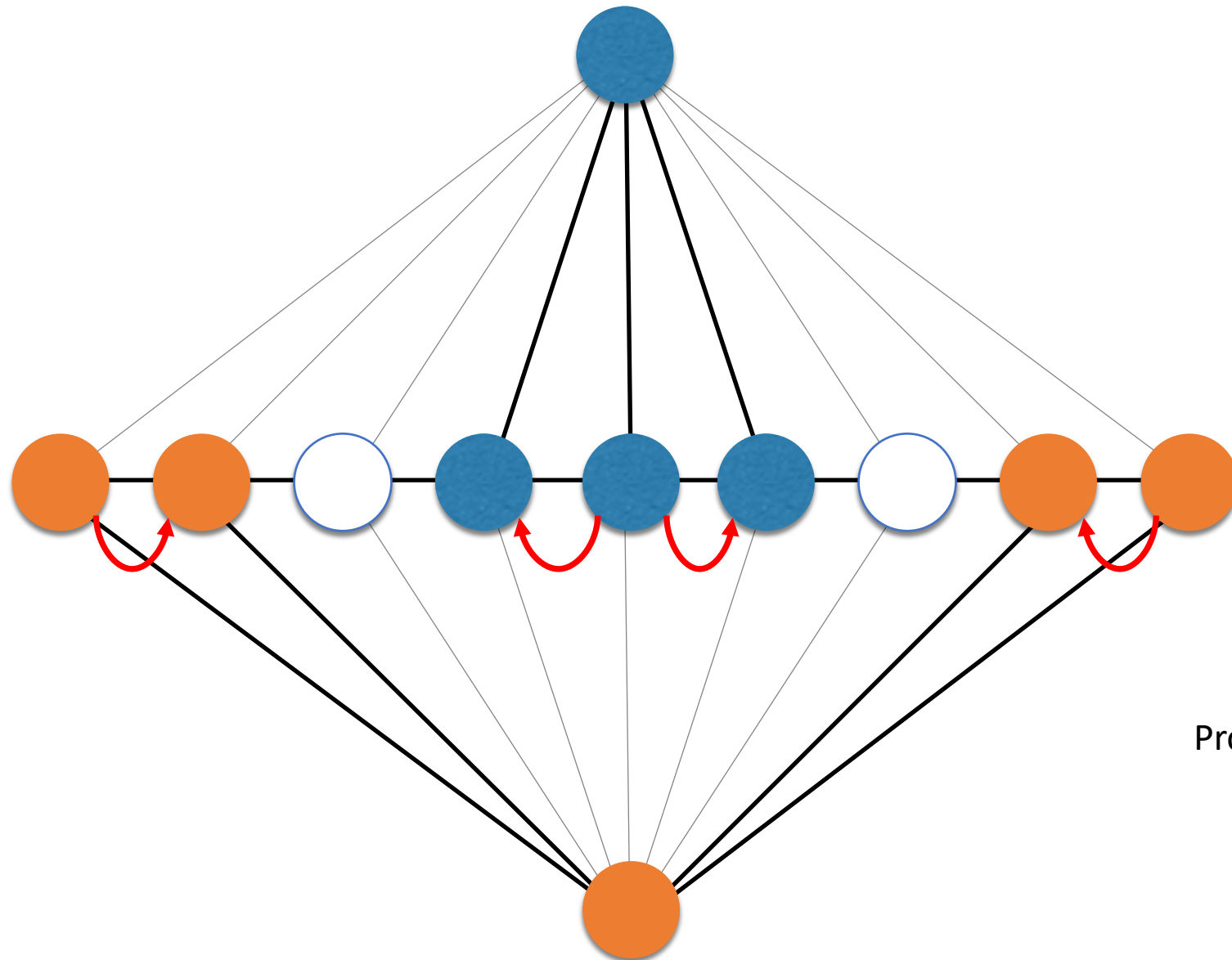


Graph cuts









Propagazione

