

Sistemi Intelligenti Clustering

Alberto Borghese and Iuri Frosio

Università degli Studi di Milano
Laboratorio di Sistemi Intelligenti Applicati (AIS-Lab)
Dipartimento di Scienze dell'Informazione
Alberto.borghese@unimi.it



A.A. 2012-2013

1/52

<http://homes.dsi.unimi.it/~borghese/>



Riassunto



- **K means**
- Mappe di Kohonen

A.A. 2012-2013

2/52

<http://homes.dsi.unimi.it/~borghese/>



Tassonomia (sintetica) degli algoritmi di clustering



- Algoritmi gerarchici (agglomerativi, divisivi), e.g. **Hierarchical clustering**.
- Algoritmi partizionali, hard: **K-means, quad-tree decomposition**.
- Algoritmi partizionali, soft: fuzzy c-mean, neural-gas, enhanced vector quantization, **mappe di Kohonen**.
- Algoritmi statistici: **mixture models**.



Clustering



- Data, $\{X_1 \dots X_N\} \in \mathbb{R}^D$
- Cluster $\{C_1 \dots C_M\} \rightarrow \{P_1 \dots P_M\} \in \mathbb{R}^{D_u}$

P_j represents the set of data inside its cluster.

The set of data inside its cluster has to be determined



K-means (partitional): framework



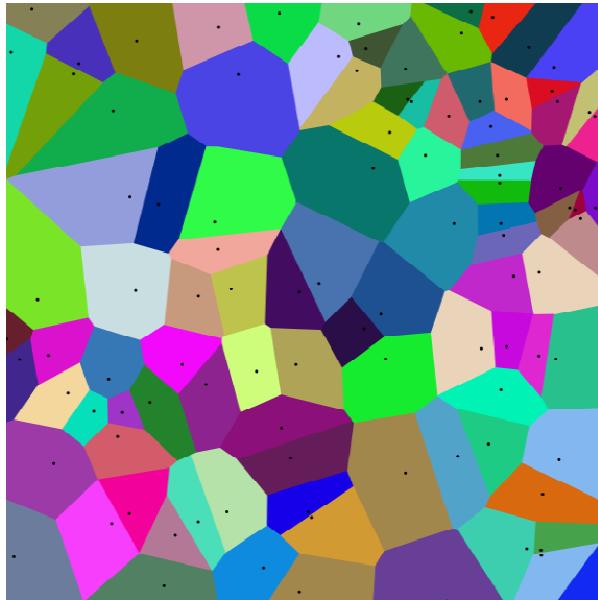
- Siano $\mathbf{X}_1, \dots, \mathbf{X}_D$ i dati di addestramento, features (per semplicità, definiti in \mathbb{R}^2);
- Siano $\mathbf{C}_1, \dots, \mathbf{C}_K$ i *prototipi* di K classi, definiti anch'essi in \mathbb{R}^2 ; ogni *prototipo* identifica il baricentro della classe corrispondente;
- Lo schema di classificazione adottato sia il seguente: “ \mathbf{X}_i appartiene a \mathbf{C}_j se e solo se \mathbf{C}_j è il *prototipo* più vicino a \mathbf{X}_i (distanza euclidea)”;
- L'algoritmo di addestramento permette di determinare le posizioni dei *prototipi* \mathbf{C}_j mediante successive approssimazioni.

A.A. 2012-2013

5/52



**Risultato del clustering è
un diagramma di Voronoj**



I poligoni azzurri rappresentano i diversi cluster ottenuti. Ogni punto marcato all'interno del cluster (cluster center) è rappresentativo di tutti i punti del cluster

A.A. 2012-2013

6/52

<http://homes.dsi.unimi.it/~borghese/>



Algoritmo K-means



L'obiettivo che l'algoritmo si propone è di minimizzare la varianza totale intra-cluster. Ogni cluster viene identificato mediante un centroide o punto medio. L'algoritmo segue una procedura iterativa. Inizialmente crea K partizioni e assegna ad ogni partizione i punti d'ingresso o casualmente o usando alcune informazioni euristiche. Quindi calcola il centroide di ogni gruppo. Costruisce quindi una nuova partizione associando ogni punto d'ingresso al cluster il cui centroide è più vicino ad esso. Quindi vengono ricalcolati i centroidi per i nuovi cluster e così via, finché l'algoritmo non converge (Wikipedia).

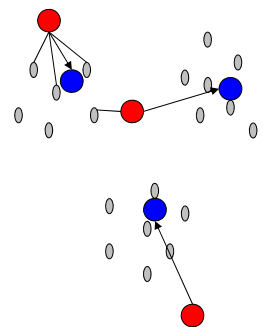
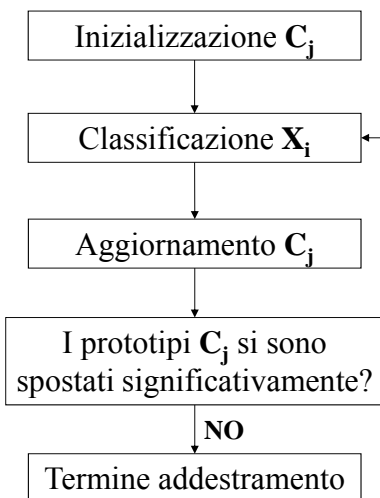
A.A. 2012-2013

7/52

<http://homes.dsi.unimi.it/~borghese/>



K-means: addestramento



Aggiornamento C_j : baricentro degli X_i classificati da C_j .

A.A. 2012-2013

8/52



Algoritmo K-means::formalizzazione



- Dati N pattern in ingresso $\{x_j\}$ e C_k prototipi che vogliamo diventino i centri dei cluster, x_j e $C_k \in \mathbb{R}^N$. Ciascun cluster identifica una regione nello spazio, P_k .
- Valgono le seguenti proprietà:

$$\bigcup_{k=1}^K P_k = Q \supseteq \mathbb{R}^D \quad \text{I cluster coprono lo spazio delle feature}$$

$$\bigcap_{k=1}^K P_k = \emptyset \quad \text{I cluster sono disgiunti.}$$

- $x_j \in C_k \quad \text{Se: } (x_j - C_k)^2 \leq (x_j - C_l)^2 \quad l \neq k$

- La funzione obiettivo viene definita come: $\sum_{i=1}^K \sum_{j=1}^N (x_j - C_k)^2$

A.A. 2012-2013

9/52

<http://homes.dsi.unimi.it/~borghese/>



Algoritmo K-means::dettaglio dei passi



- Inizializzazione.
 - ◆ Posiziono in modo arbitrario o guidato i K centri dei cluster.
- Iterazioni
 - ◆ Assegno ciascun pattern al cluster il cui centro è più vicino, formando così un certo numero di cluster ($\leq K$).
 - ◆ Calcolo la posizione dei cluster, C_k , come baricentro dei pattern assegnati ad ogni cluster, spostando quindi la posizione dei centri dei cluster.
- Condizione di uscita
 - I centri dei cluster non si spostano più.

A.A. 2012-2013

10/52

<http://homes.dsi.unimi.it/~borghese/>



K-means: :limiti

- Partitional, hard, deterministic;
- Veloce, semplice da implementare;
- Trova un minimo locale della funzione $f = \sum_j \sum_i [\text{dist}(x_i, \text{prot}_j)] / N_j$;
- Il risultato dipende dall'inizializzazione!
- Possono essere usati altri metodi (es. GA) per inizializzare K-means... es. GA per la minimizzazione di f , effettuano una ricerca globale, ma sono lenti!

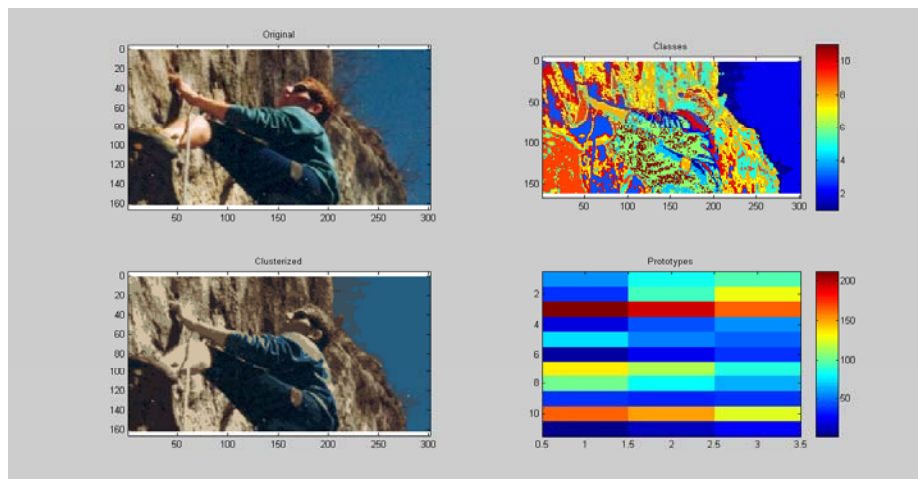
Sw in Matlab available

A.A. 2012-2013

11/52



K-Means per immagine RGB



Da 255 colori a 33 colori

A.A. 2012-2013

12/52



Riassunto



- K means
- **Mappe di Kohonen**

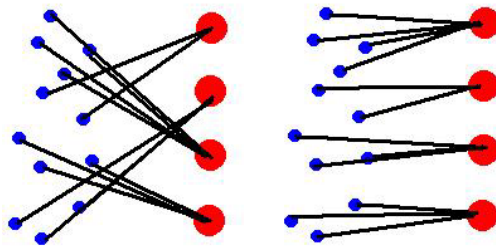
A.A. 2012-2013

13/52

<http://homes.dsi.unimi.it/~borghese/>



Clustering vs. feature mapping



↑
Clustering

↑
Feature Mapping

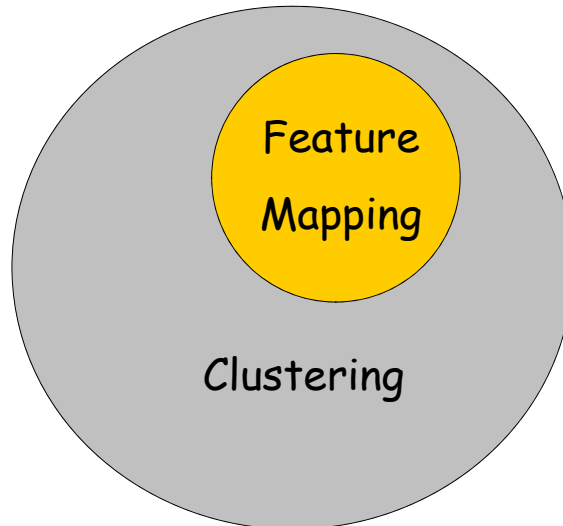
A differenza di quanto accade con il clustering, nel feature mapping vengono preservate le relazioni topologiche tra i dati.

A.A. 2012-2013

14/52



Clustering e feature mapping



A.A. 2012-2013

15/52



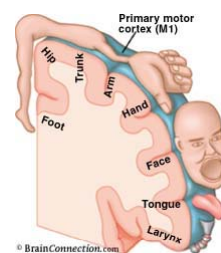
Feature mapping



Con il Feature Mapping si dà importanza alla posizione dei prototipi (xxx-topia, Homunculus motorio);

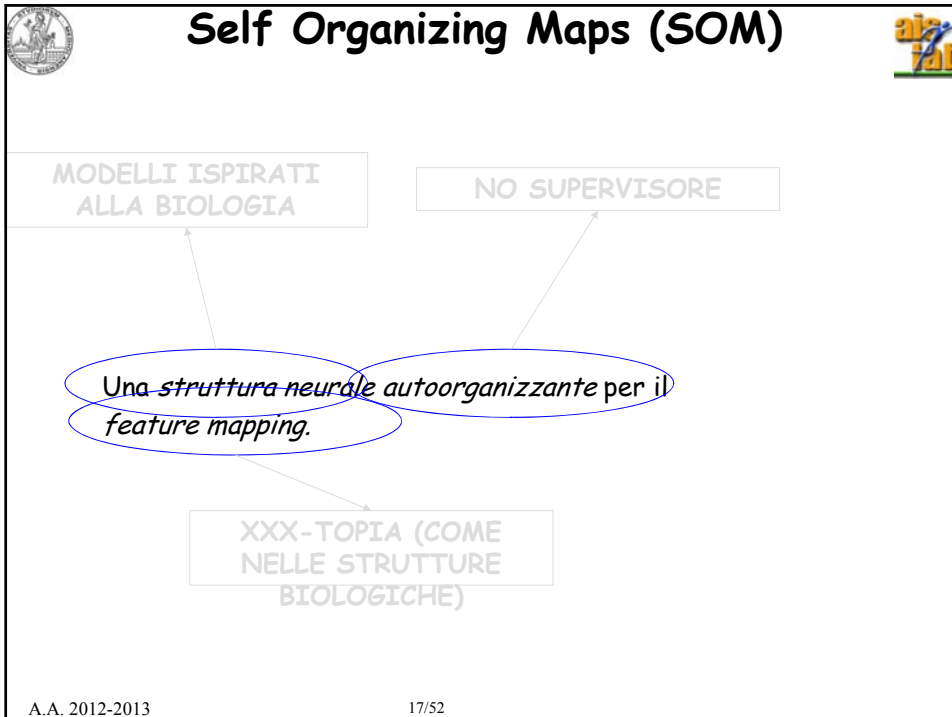
A uscite contigue corrispondono configurazioni d'ingresso contigue;

Il clustering opera una trasformazione tra lo spazio degli ingressi e lo spazio delle uscite (prototipi) che preserva le relazioni di vicinanza tra i vari elementi.



A.A. 2012-2013

16/52



Tipo di apprendimento

COMPETITIVE LEARNING. *Apprendimento competitivo. Dato un certo input, le unità competono tra loro per “aggiudicarsi” l’input.*

Questo meccanismo può essere hard. Nel caso estremo: “winner-take-all”, “spara” un solo neurone per volta (grandmother cell). Oppure può essere soft, le unità raggiungono un grado diverso di “vincita”.

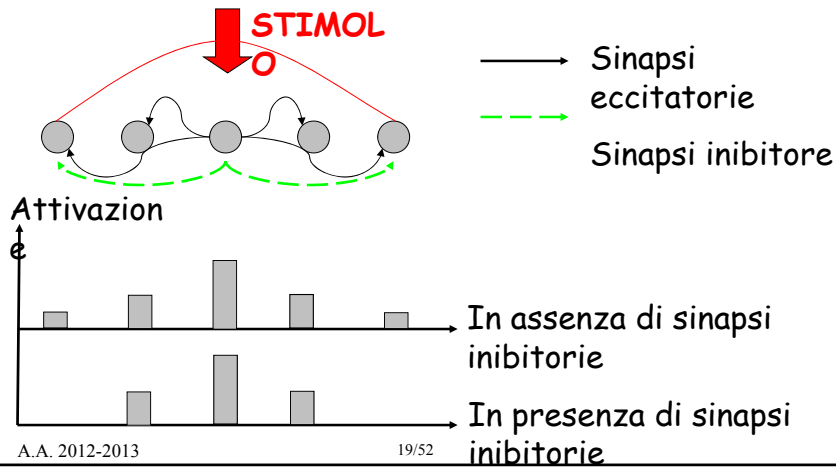
Winner-take-all → hard approach

More than one winner → soft approach

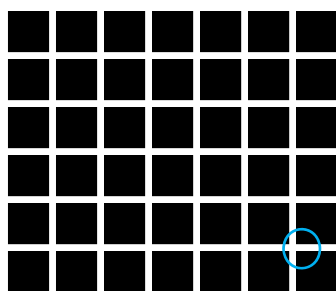
A.A. 2012-2013 18/52

L'ispirazione biologica : Il campo recettivo

- Sinapsi eccitatorie verso i neuroni vicini;
- sinapsi inibitorie verso i neuroni lontani.

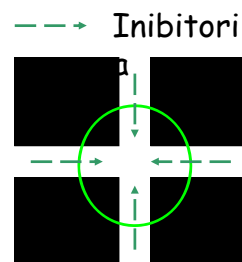


Il campo recettivo: effetto Hermann



Pallini neri agli incroci delle linee bianche.

Il neurone centrale viene "spento" dai neuroni vicini attivi → generazione della zona scura.



SOM: competitive learning in versione "hard"

X_i
 W_j
 Nuovo dato X_k
 Connessione inibitoria "spegne gli altri neuroni"
 Associazione di X_k alla classe W_i ; le sinapsi inibitorie spengono tutti gli altri neuroni (*winner takes all*). Il solo neurone che "spara", W_i , individua la classe di X .
 Mapping $R^N \rightarrow R^M$, con $M \leq N$ (da spazio delle caratteristiche a spazio delle classi).

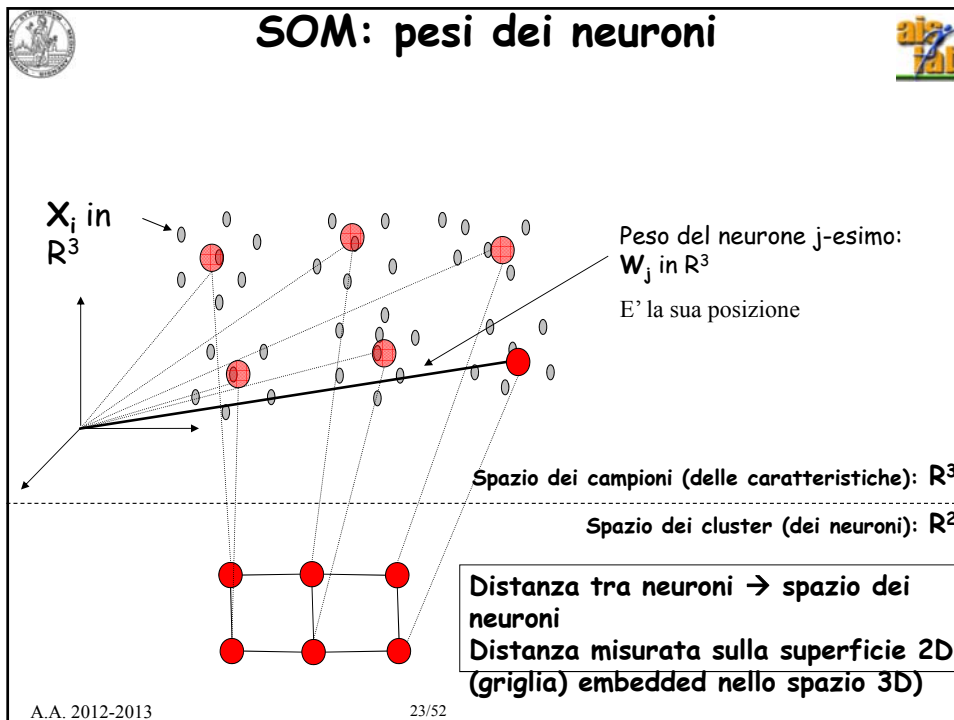
SOM: organizzazione topologica

I neuroni della SOM sono ordinati topologicamente nello spazio dei neuroni (es. griglia ordinata in R^2). In tale spazio viene definita la distanza tra neuroni.

La distanza tra i due neuroni (nello spazio dei neuroni) è:

$\sqrt{(\Delta x)^2 + (\Delta y)^2} = 1.4142$ [Metrica Euclidea]
 $|\Delta x| + |\Delta y| = 2$ [Manhattan]
 ...

A.A. 2012-2013 22/52



- ## SOM: addestramento
- Siano X_1, \dots, X_D i dati di addestramento (per semplicità, definiti in \mathbb{R}^3);
 - siano W_1, \dots, W_K i *prototipi* di K classi, definiti anch'essi in \mathbb{R}^3 ; ogni *prototipo* identifica il peso di un neurone della SOM (W_j);
 - lo schema di associazione adottato sia il seguente: “ X_i appartiene a W_j se e solo se W_j è il *prototipo* (*peso del neurone*) più vicino a X_i , nello spazio dei campioni (delle caratteristiche, \mathbb{R}^3)”;
 - l'algoritmo di addestramento permette di determinare i pesi dei neuroni (le posizioni dei *prototipi*) W_j mediante successive approssimazioni;
 - L'algoritmo di addestramento tiene conto della topologia dei neuroni nello spazio dei neuroni (*feature mapping*).
- A.A. 2012-2013 24/52



SOM: output



- All'interazione k- esima, si presenti alla rete il dato \mathbf{X}_i ;
- Unità vincente (associazione):
 j^* t.c. $\|\mathbf{W}_{j^*} - \mathbf{X}_i\| = \min_j \|\mathbf{W}_j - \mathbf{X}_i\|$
- Uscita:
 - ◆ $u_{j^*} = 1$, se $j = j^*$
 - ◆ $u_j = 0$ se $j \neq j^*$

UNITA' VINCENTE



Competitive learning



- All'interazione k- esima, si presenti alla rete il dato \mathbf{X}_i ;
- **Aggiornamento di tutti i prototipi \mathbf{W}_j ;**
- Generalized competitive Learning Rule:
 - ◆ $\Delta \mathbf{W}_j = \eta_k \Lambda_k(i,j) (\mathbf{X}_i - \mathbf{W}_j)$

AGGIORNAMENTO
PESI DEI NEURONI

$\Lambda_k(i,j)$ è una funzione “campo recettivo”

- ◆ $\Lambda_k(i,j) = \exp(-\|f(\mathbf{X}_i) - f(\mathbf{W}_j)\|^2 / 2\sigma_k^2)$

FUNZIONE DI
VICINATO



Soft-clustering



$$\Delta \mathbf{W}_j = \eta_k \Lambda_k(i,j) (\mathbf{X}_i - \mathbf{W}_j)$$

$\Lambda_k(i,j)$ è l'elemento chiave. I "Campi recettivi" dei diversi neuroni sono parzialmente sovrapposti.

In "Competitive clustering" $\Lambda_k(i,j)$ è una Gaussiana nello spazio dei dati e dei prototipi.

In "Neural-gas" $\Lambda_k(i,j)$ è una ranking function nello spazio dei dati e dei prototipi.

In "Fuzzy c-means" $\Lambda_k(i,j)$ è realizzata mediante membership function nello spazio dei dati e dei prototipi.



SOM e competitive learning



- All'interazione k- esima, si presenti alla rete il dato \mathbf{X}_i ;

- Unità vincente (associazione):

$$j^* \text{ t.c. } \|\mathbf{W}_{j^*} - \mathbf{X}_i\| = \min_j \|\mathbf{W}_j - \mathbf{X}_i\|$$

- Competitive Learning Rule (SOM, Kohonen '81):

- ◆ $\Delta \mathbf{W}_j = \eta_k \Lambda_k(i,j) (\mathbf{X}_i - \mathbf{W}_j)$

- ◆ $\Lambda_k(i,j) = \exp(-\|\mathbf{W}_{j^*} - \mathbf{W}_j\|^2 / 2\sigma_k^2)$

UNITA' VINCENTE

← AGGIORNAMENTO PESI DEI NEURONI

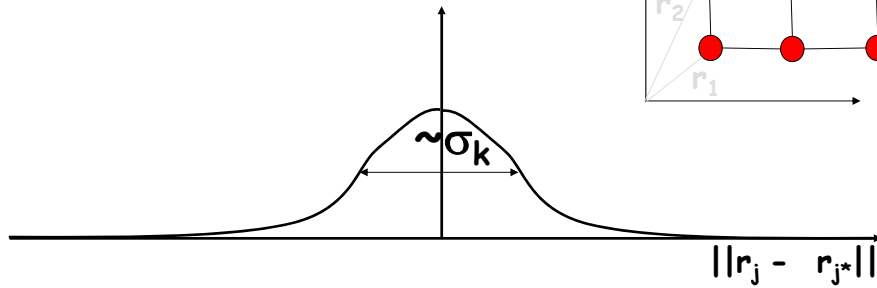
← FUNZIONE DI VICINATO

Calcolata nello spazio dei prototipi



Funzione di vicinato

$$\Lambda_k(j, j^*) = \exp(-|r_j - r_{j^*}|^2 / 2\sigma_k^2)$$



Neurone lontano dal neurone vincente: $\Lambda_k(j, j^*) \rightarrow 0$, $\Delta w_j \rightarrow 0$

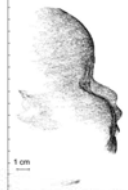
$\sigma_k \rightarrow 0$, l'esponenziale va rapidamente a 0, solo il neurone vincente, j^* , viene aggiornato.

$\sigma_k \rightarrow \infty$ l'esponenziale rimane costantemente = 1. aggiornano tutti i neuroni di una quantità pari a: $\Delta W_j = \eta_k (X_i - W_j)$

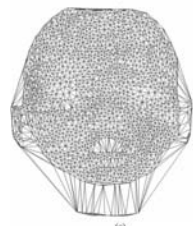
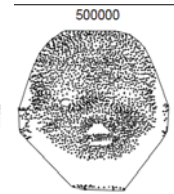


Competitive learning ("First search then converge")

- ORDERING PHASE:** η , σ grandi; ogni neurone può spostarsi molto verso l'ingresso X_i ; il neurone trascina con sé i vicini; in tale fase la rete si dispiega nello spazio R^N "spargendo" i suoi neuroni.
- TUNING PHASE:** η , σ piccoli; ogni neurone si muove da solo; è una fase di raffinamento in cui vengono raggiunti con precisione i centri dei cluster.



100,000 sampled points
reconstructed
with 2,000 RV

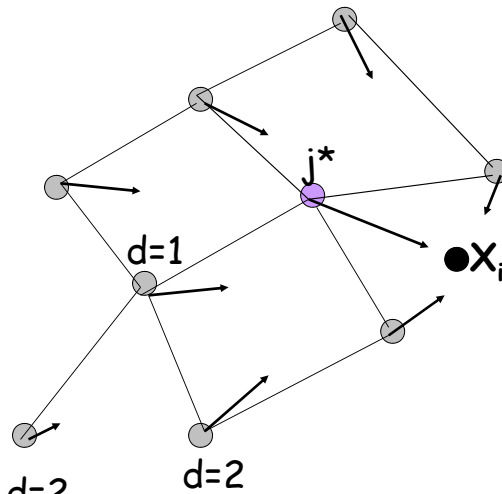


A

30/52



SOM: addestramento



In definitiva:

- Il neurone vincente si sposta verso X_i , trascinando i vicini in proporzione alla loro distanza nello spazio delle feature.

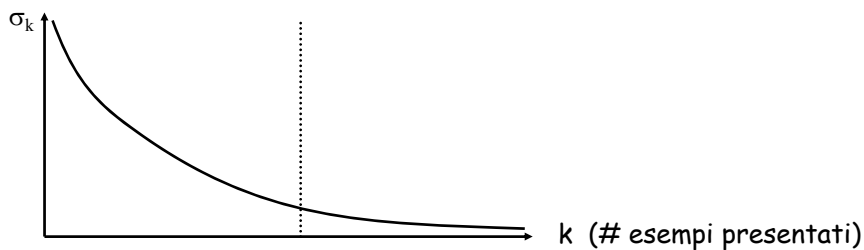
- L'ordinamento dei pesi dei neuroni nello spazio dei dati è simile all'ordinamento dei neuroni nello spazio dei neuroni.

A.A. 2012-2013

31/52



Funzione di vicinato nel tempo



$$\Delta \mathbf{w}_j = \eta_k \Lambda_k(j, j^*) (\mathbf{X}_i - \mathbf{w}_j) \rightarrow \mathbf{w}' = \mathbf{w} + \eta_k \Lambda_k \mathbf{X}_i - \eta_k \Lambda_k \mathbf{w}$$

$$\Lambda_k(j, j^*) = \exp(-\| \mathbf{r}_j - \mathbf{r}_{j^*} \|^2 / 2\sigma_k^2)$$

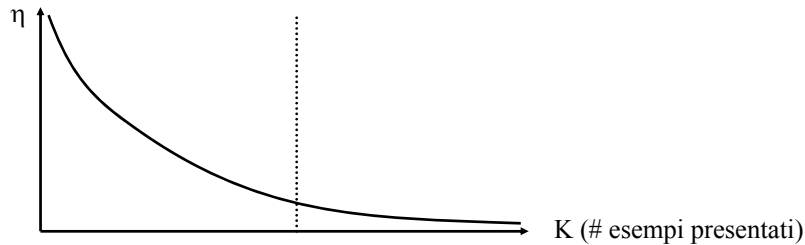
Procedendo nell'addestramento della rete, un neurone perde la capacità di spostare i suoi vicini.

A.A. 2012-2013

32/52



Learning rate nel tempo



$$\Delta w_j = \eta_k \Lambda_k(j, j^*) (\mathbf{X}_i - w_j) \rightarrow w' = w + \eta_k \Lambda_k \mathbf{X}_i - \eta_k \Lambda_k w_i$$

Procedendo nell'addestramento della rete, i pesi dei neuroni perdono la possibilità di muoversi => rete più stabile.

A.A. 2012-2013

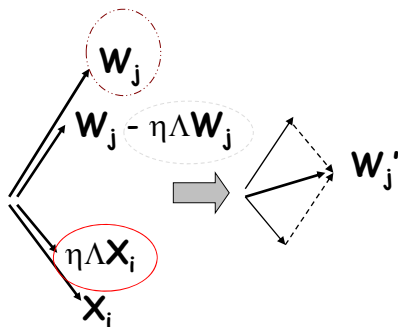
33/52



Competitive Learning come forze contrapposte



$$\Delta W_j = \eta_k \Lambda_k(i, j) (\mathbf{X}_i - W_j) \rightarrow W'_j = W_j + \eta_k \Lambda_k(i, j) \mathbf{X}_i - \eta_k \Lambda_k(i, j) W_j$$



Il peso W_j del neurone vincente j si sposta verso l'ingresso presentato X_i .

$-\eta W_j$ evita che il peso W_j cresca a dismisura.

A.A. 2012-2013

34/52



SOM: addestramento



- L'addestramento avviene presentando alla rete i vettori (dati) $X_i \in \mathbb{R}^N$ per un numero di epoche E ;
- Per ogni esempio presentato X_i , vengono aggiornati i pesi dei neuroni della rete;
- Durante l'addestramento il learning rate η e la neighborhood distance σ decrescono;
- Se presentiamo alla rete un nuovo esempio X_i alla fine dell'addestramento, la rete lo associa al neurone vincente (clustering);

In più:

- Categorie simili sono rappresentate da neuroni vicini (feature mapping).

A.A. 2012-2013

35/52



SOM: problemi



- E' necessario scegliere η , σ , numero di epoche, durata della ordering phase \rightarrow metodi empirici(!);
- Scelta della topologia e del numero di neuroni corretti;
- I dati di addestramento devono presentare una certa ridondanza;
- Unità "morte" (dead-units);

A.A. 2012-2013

36/52

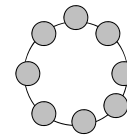


SOM per ordinamento



Spazio dei dati X_i (e dei pesi w) : \mathbb{R}^3

Topologia della SOM : circolare



Parametri di addestramento : # neuroni, $\eta(t)$, $\sigma(t)$, ...

[Movie here](#) (with RealPlayer)

A.A. 2012-2013

37/52

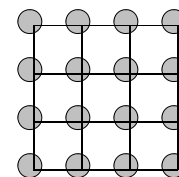


SOM per ricostruzione 3D



Spazio dei dati q (e dei pesi w) : \mathbb{R}^3

Topologia della SOM : griglia 2D



Parametri di addestramento : # neuroni =
 10×10 , $0.5 \rightarrow$ 0.1 lin, $10 \rightarrow$
1 lin, ...

[Movie here](#)

A.A. 2012-2013

38/52



SOM per ricostruzione 3d



Problemi:

- Oscillazioni della rete all'inizio dell'addestramento;

Sol.: Scelta accurata di $\eta(t)$, $\sigma(t)$

- Raggiungimento dei confini della superficie aperta;
Sol.: Boundary First Method + η , σ modificati

- Numero insufficiente di neuroni;

Sol.: Parametrizzazione della points cloud

A.A. 2012-2013

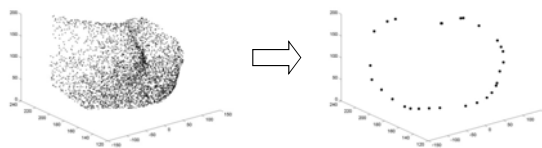
39/52



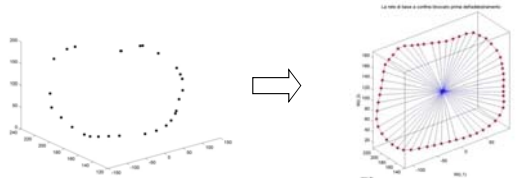
Boundary First Method



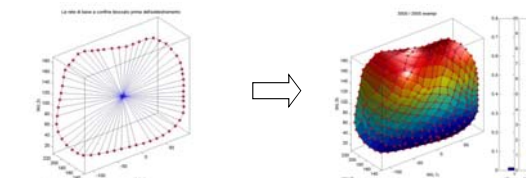
1) Individuazione dei punti di confine della superficie



2) Posizionamento dei neuroni di bordo



3) Addestramento della SOM (neuroni di bordo bloccati), η, σ maggiori ai lati



A.A. 2012-2013

40/52



SOM per ricostruzione 3D



- [Movie here](#)
- [Movie here](#)

A.A. 2012-2013

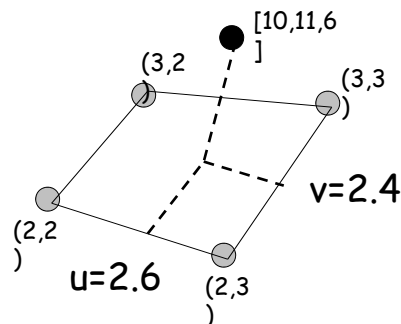
41/52



Parametrizzazione Points Cloud



- Ad ogni punto 3D $[x,y,z]$ vengono assegnate le coordinate 2D $[u,v]$ corrispondenti nello spazio 2D della SOM tramite una proiezione (parametrizzazione)



A.A. 2012-2013

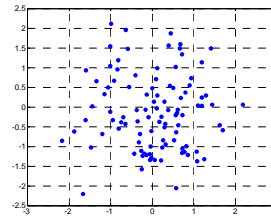
42/52



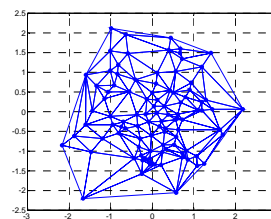
Triangolazione punti in $[u, v]$ (Delaunay)



- Dato un set di punti nello spazio 2D, è possibile trovare un set di triangoli tale per cui (triangolazione di Delaunay):
 - ◆ ogni punto si trova al vertice di uno o più triangoli
 - ◆ nessun punto si trova all'interno di un triangoli
- Utile in computer graphics.



- In modo più rigoroso:
a Delaunay triangulation for a set P of points in the plane is a triangulation $DT(P)$ such that no point in P is inside the circumcircle of any triangle in $DT(P)$. Delaunay triangulations **maximize the minimum angle of all the angles of the triangles** in the triangulation; they tend to **avoid "sliver" triangles**.



A.A. 2012-2013

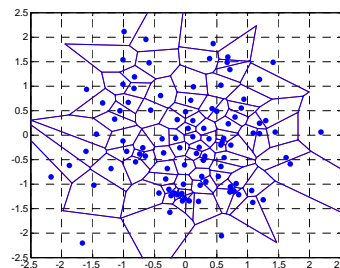
43/52



Delaunay e Voronoi



- Duale della triangolazione di Delaunay → Tessellazione di Voronoi.
- Suddivisione "regolare" dello spazio.

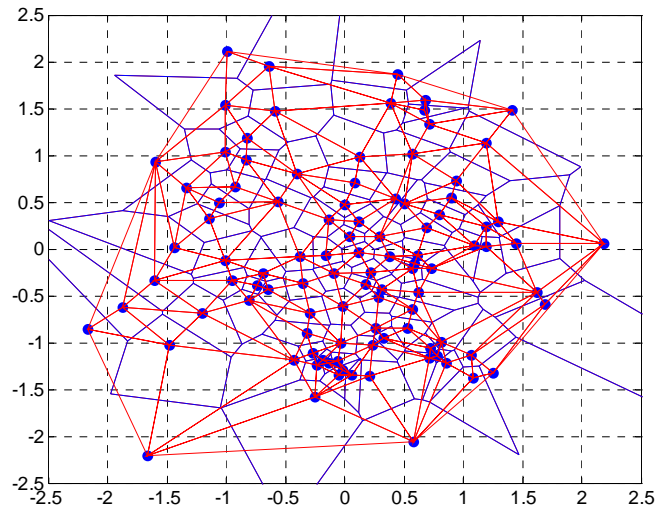


A.A. 2012-2013

44/52



Dealunay & Voronoi



A.A. 2011



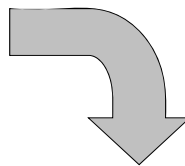
Parametrizzazione Points Cloud



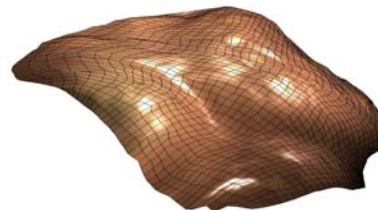
I punti 2D $[u,v]$ possono essere triangolati (ricostruzione a mesh di triangoli). La mesh viene poi filtrata/interpolata.



Ricostruzione a mesh di triangoli





Filtraggio o interpolazione

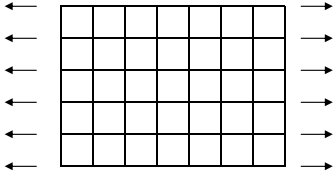




Ricostruzione Finale

A.A. 2012-2013



46/52

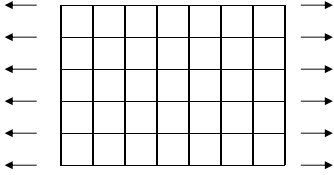

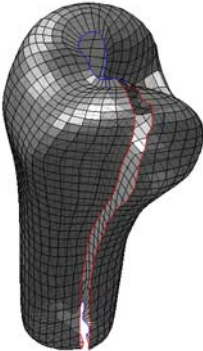
 **Tibia Left** 



Topologia: cilindro

A.A. 2012-2013 47/52

 **Femoral Bone Left** 



Topologia: cilindro

A.A. 2012-2013 48/52



Varianti SOM



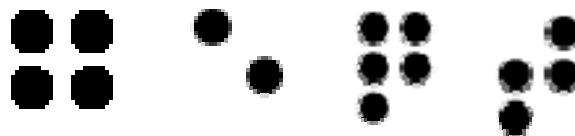
- SELF CREATING MAP
 - Aggiunta di un neurone:
 - Vicino al neurone vincente con f maggiore;
 - Vicino al neurone con curvatura massima;
- SOM SUPERVISIONATE (per la classificazione);
- ...

A.A. 2012-2013

49/52



Software available



Clustering topologico dei caratteri Braille

SW available

A.A. 2012-2013

50/52

<http://homes.dsi.unimi.it/~borghese>



Bibliografia



- J. Hertz, A. Krogh, R. G. Palmer, “Introduction to the theory of neural computation”, Santa Fe Institute Editorial Board, 1990
 - ◆ Feature mapping & SOM: cap. 9.4



Riassunto



- K means
- Mappe di Kohonen



Sommario



- Clustering a feature mapping
- SOM