

Sistemi Intelligenti Soft Clustering

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



A.A. 2014-2015

1/73

<http://borghese.di.unimi.it/>



Riassunto



- **Soft Clustering**
- Mappe di Kohonen

A.A. 2014-2015

2/73

<http://borghese.di.unimi.it/>

K-means: addestramento

```

graph TD
    A[Inizializzazione C_j] --> B[Classificazione X_i]
    B --> C[Aggiornamento C_j]
    C --> D{I prototipi C_j si sono spostati significativamente?}
    D -- SI --> B
    D -- NO --> E[Termine addestramento]
  
```

Aggiornamento C_j : baricentro degli X_i classificati da C_j .

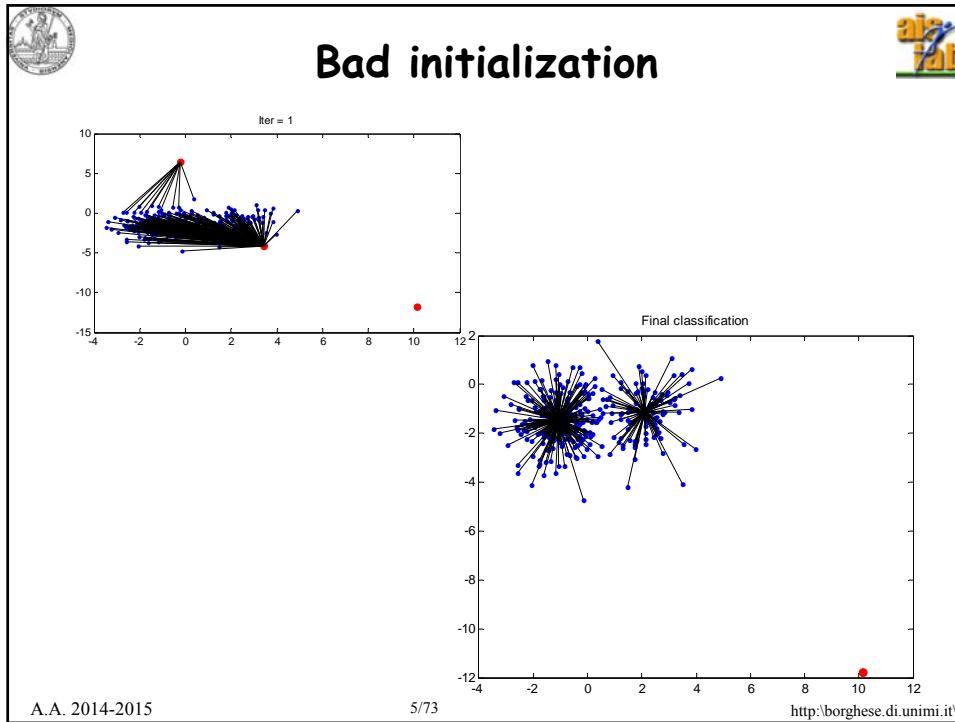
A.A. 2014-2015 3/73 <http://borghese.di.unimi.it/>

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. 2014-2015 4/73 <http://borghese.di.unimi.it/>



Principles of soft-clustering

- I centroidi vengono spostati e non posizionati
- Lo spostamento dei centroidi avviene analizzando iterativamente tutti i dati
- Lo spostamento viene ridotto via via che l'apprendimento procede

A.A. 2014-2015 6/73 <http://borgnese.di.unimi.it/>



Competitive learning



- All'interazione k- esima, si presenti al sistema **un (1)** dato, \mathbf{X}_i ;
- **Aggiornamento di tutti i prototipi \mathbf{W}_j ("neuroni")**
- Generalized competitive Learning Rule:
 - ◆ $\Delta \mathbf{W}_j = \eta_k \Lambda_k(i,j) (\mathbf{X}_i - \mathbf{W}_j)$

← AGGIORNAMENTO
PESI (POSIZIONE)
DEI NEURONI

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

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

← FUNZIONI DI
VICINATO



Competitive learning



- All'interazione k- esima, si presenti alla rete **un (1)** dato \mathbf{X}_i ; al termine dell'apprendimento.
- Unità vincente (associazione):
 - j^* t.c. $\|\mathbf{W}_{j^*} - \mathbf{X}_i\| = \min_j \|\mathbf{W}_j - \mathbf{X}_i\|$

← UNITA' VINCENTE

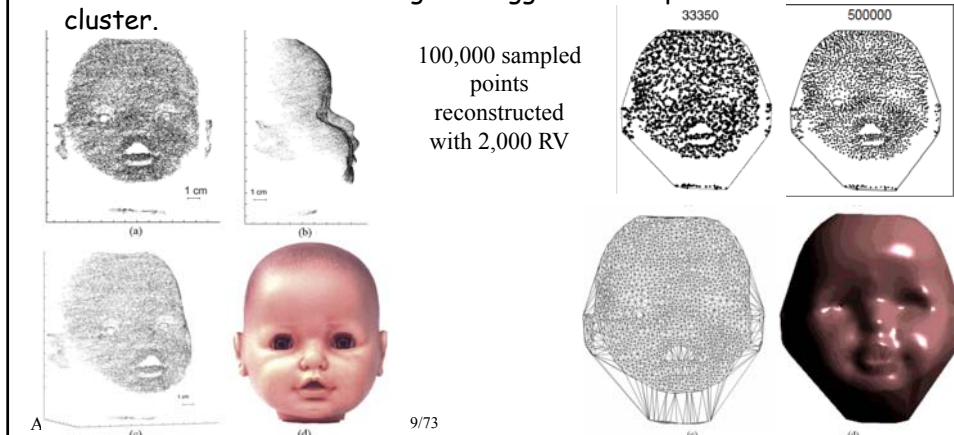
Anche qui viene indotta una tessellazione di Voronoj dallo spazio da tutte le unità vincenti.



Competitive learning ("First search then converge")



- 1) **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.
- 2) **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.



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 $\underline{0}$ dei prototipi (mappe di Kohonen).

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.



I problemi del soft-clustering



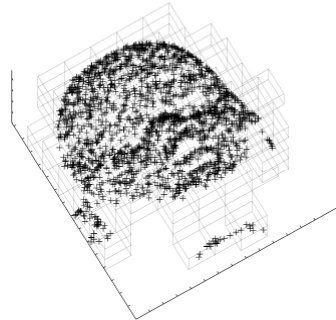
Dead-units: sono centroidi che non vengono aggiornati da un certo passo, k , in poi.

Inizializzazione guidata dai dati.

$$\rho_{Centroid} = \rho_{Data}^{\gamma} \quad \text{with} \quad \gamma = \frac{D}{D+2}$$

Partition of the input space and distribution of the number of centroids inside each box through a partitioning function:

$$M_k = M \frac{N_k^{\gamma}}{\sum_k N_k^{\gamma}}$$



Minimi locali.



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



Riassunto



- Soft Clustering
- **Mappe di Kohonen**

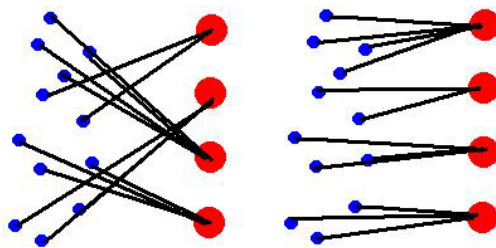
A.A. 2014-2015

13/73

<http://borghese.di.unimi.it/>



Clustering vs. feature mapping



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

↑
Clustering

↑
Feature Mapping

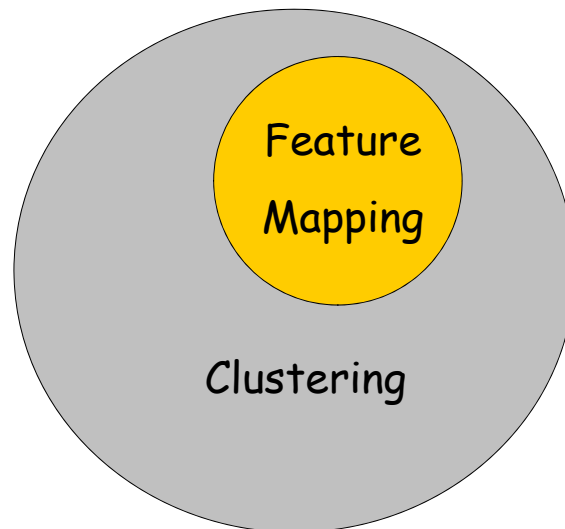
A.A. 2014-2015

14/73

<http://borghese.di.unimi.it/>



Clustering e feature mapping



A.A. 2014-2015

15/73

<http://borghese.di.unimi.it/>



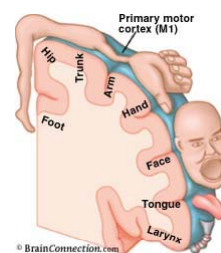
Feature mapping



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. 2014-2015

16/73

<http://borghese.di.unimi.it/>

Self Organizing Maps (SOM)

MODELLI ISPIRATI ALLA BIOLOGIA

NO SUPERVISORE

Una *struttura neurale autoorganizzante* per il *feature mapping*.



XXX-TOPIA (COME NELLE STRUTTURE BIOLOGICHE)

A.A. 2014-2015

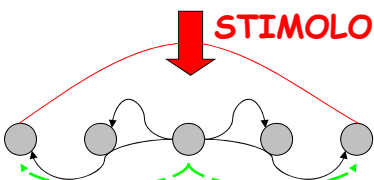
17/73

<http://borghese.di.unimi.it/>

L'ispirazione biologica : Il campo recettivo

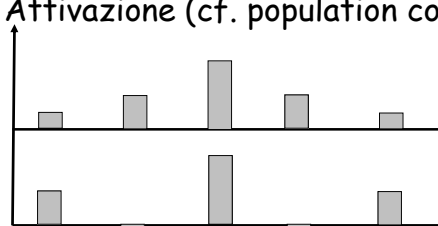
- **Sinapsi eccitatorie** verso i neuroni vicini;
- **sinapsi inibitorie** verso i neuroni lontani.



—→ Sinapsi eccitatorie

- - -→ Sinapsi inibitorie

Attivazione (cf. population code)



In assenza di sinapsi inibitorie

In presenza di sinapsi inibitorie

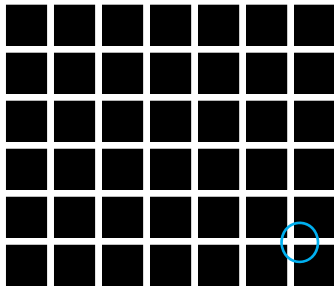
A.A. 2014-2015

18/73

<http://borghese.di.unimi.it/>



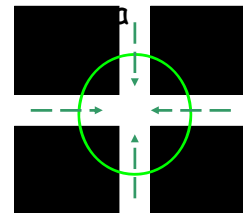
Il campo recettivo: effetto Hermann



Pallini neri agli incroci delle linee bianche.

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

---→ Inibitori



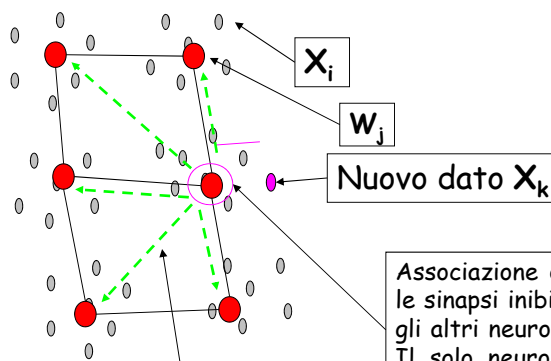
A.A. 2014-2015

19/73

<http://borgese.di.unimi.it/>



SOM: competitive learning in versione "hard"



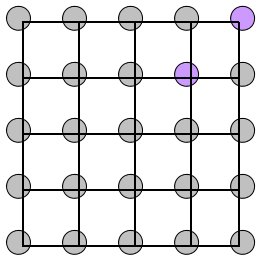
Mapping $R^N \rightarrow R^M$, con $M \leq N$ (da spazio delle caratteristiche a spazio delle classi).



SOM: organizzazione topologica



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



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

$$\sqrt{(\Delta x)^2 + (\Delta y)^2} = 1.4142 \quad [\text{Metrica Euclidea}]$$

$$|\Delta x| + |\Delta y| = 2 \quad [\text{Manhattan}]$$

...

A.A. 2014-2015

21/73

<http://borghese.di.unimi.it/>



SOM: pesi dei neuroni



X_i in \mathbb{R}^3

Peso del neurone j-esimo:
 W_j in \mathbb{R}^3

E' la sua posizione

Spazio dei campioni (delle caratteristiche): \mathbb{R}^3

Spazio dei cluster (dei neuroni): \mathbb{R}^2

Distanza tra neuroni \rightarrow spazio dei neuroni
Distanza misurata sulla superficie 2D (griglia) embedded nello spazio 3D

A.A. 2014-2015

22/73

<http://borghese.di.unimi.it/>



SOM: addestramento



- Siano $\mathbf{X}_1, \dots, \mathbf{X}_D$ 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: “ \mathbf{X}_i appartiene a W_j se e solo se W_j è il *prototipo* (*peso del neurone*) più vicino a \mathbf{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. 2014-2015

23/73

<http://borghese.di.unimi.it/>



SOM: output



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

UNITA'
VINCENTE

A.A. 2014-2015

24/73

<http://borghese.di.unimi.it/>



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):

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

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

UNITA' VINCENTE

AGGIORNAMENTO
PESI DEI NEURONI

FUNZIONE DI
VICINATO

Calcolata nello spazio
dei prototipi

A.A. 2014-2015

25/73

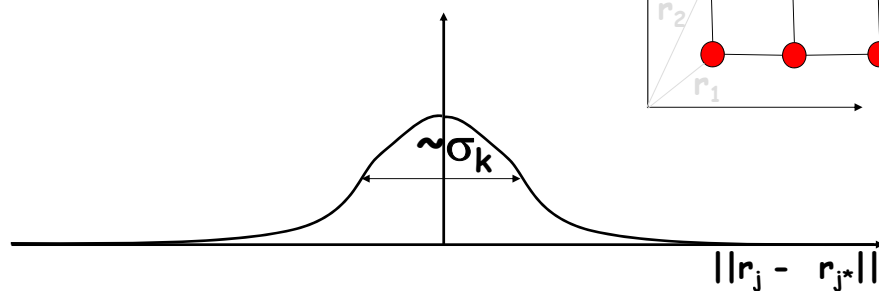
<http://borghese.di.unimi.it/>



Funzione di vicinato



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



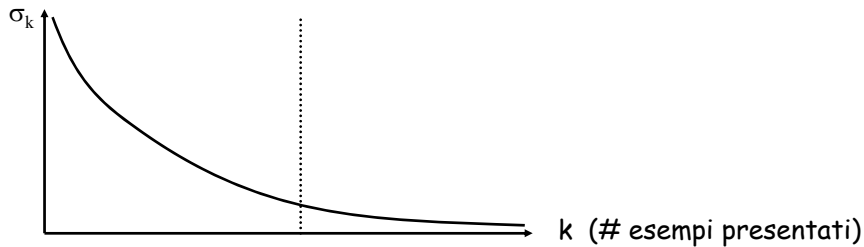
Neurone lontano dal neurone vincente: $\Lambda_k(j, j^*) \rightarrow 0$, $k \rightarrow \infty$
 $\sigma_k \rightarrow 0$, l'esponenziale va rapidamente a 0, solo il neurone vincente, j^* , viene aggiornato.

$\sigma_k \rightarrow \infty$ l'esponenziale rimane costantemente = 1. aggiornano tutti i neuroni di una quantità pari a: $\Delta \mathbf{W}_j = \eta_k (\mathbf{X}_i - \mathbf{W}_j)$

imi.it/



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.

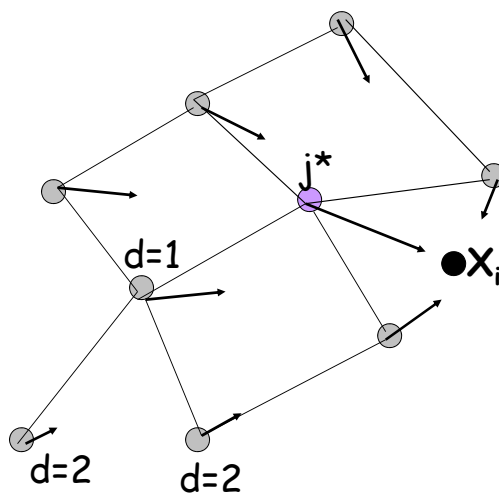
A.A. 2014-2015

27/73

<http://borgnese.di.unimi.it/>



SOM: addestramento



In definitiva:

-Il neurone vincente si sposta verso \mathbf{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.

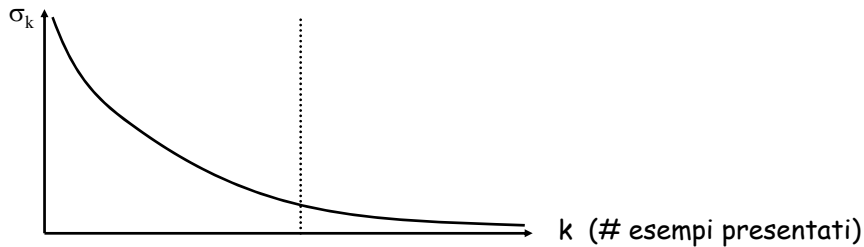
A.A. 2014-2015

28/73

<http://borgnese.di.unimi.it/>



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.

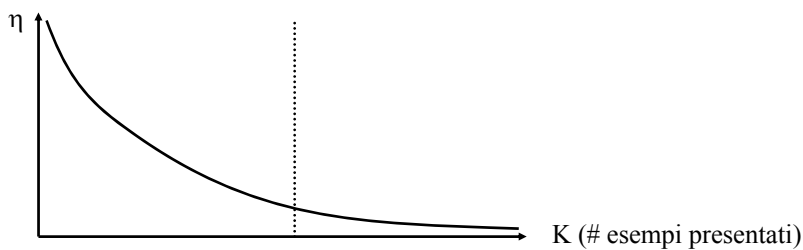
A.A. 2014-2015

29/73

<http://borghese.di.unimi.it/>



Learning rate nel tempo



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

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

A.A. 2014-2015

30/73

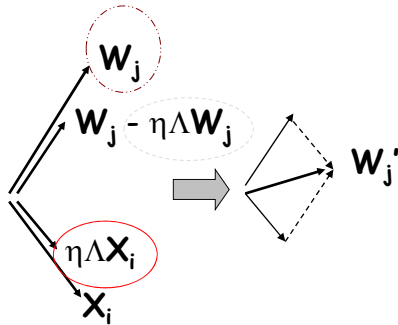
<http://borghese.di.unimi.it/>



Competitive Learning come forze contrapposte



$$\Delta W_j = \eta_k \Lambda_k(i,j) (X_i - W_j) \rightarrow W_j' = W_j + \eta_k \Lambda_k(i,j) X_i - \eta_k \Lambda_k(i,j) W_j$$



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.



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).



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. 2014-2015

33/73

<http://borghese.di.unimi.it/>

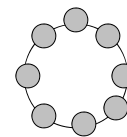


SOM per ordinamento



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. 2014-2015

34/73

<http://borghese.di.unimi.it/>

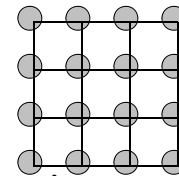


SOM per ricostruzione 3D



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

Topologia della SOM : griglia 2D



Parametri di addestramento : # neuroni =
10x10, 0.5 \rightarrow 0.1 lin, 10 \rightarrow
1 lin, ...

[Movie here](#)

A.A. 2014-2015

35/73

<http://borgese.di.unimi.it/>



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. 2014-2015

36/73

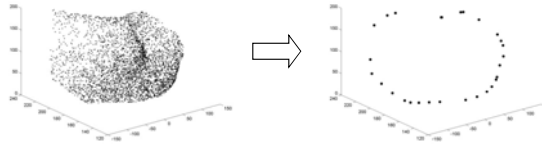
<http://borgese.di.unimi.it/>



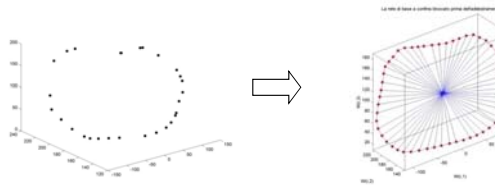
Boundary First Method



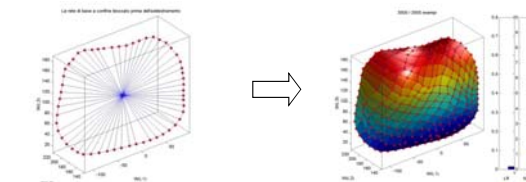
1) Individuazione dei punti di confine della superficie



2) Posizionamento dei neuroni di bordo



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



A.A. 2014-2015

37/12

<http://borghese.di.unimi.it/>



SOM per ricostruzione 3D



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

A.A. 2014-2015

38/73

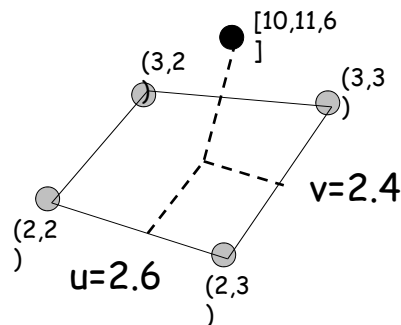
<http://borghese.di.unimi.it/>



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. 2014-2015

39/73

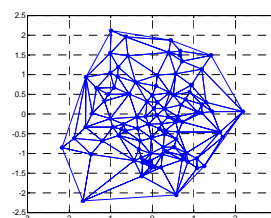
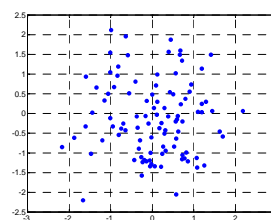
<http://borghese.di.unimi.it/>



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 triangolo
- 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. 2014-2015

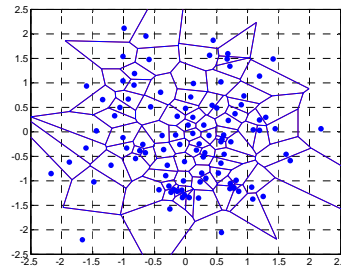
40/73

<http://borghese.di.unimi.it/>



Delaunay e Voronoi

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



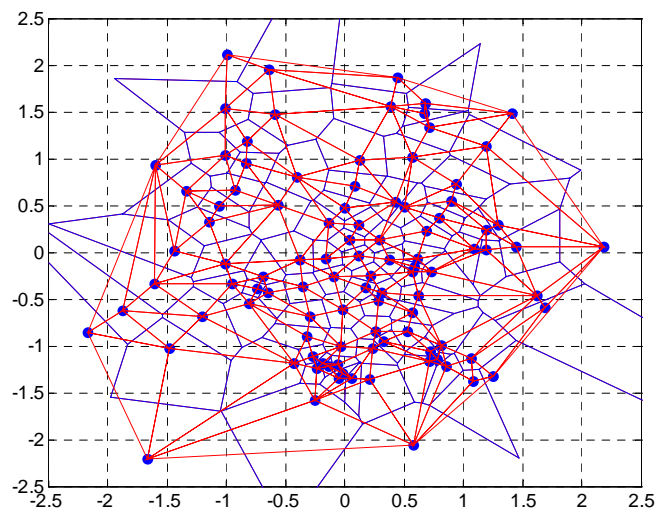
A.A. 2014-2015

41/73

<http://borghese.di.unimi.it/>



Delaunay & Voronoi



A.A. 2014

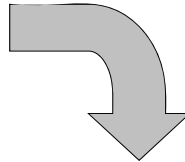
se.di.unimi.it/



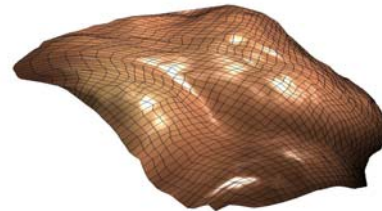
Parametrizzazione Points Cloud



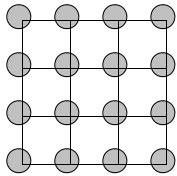
I punti 2D $[u,v]$ possono essere triangolati (ricostruzione a mesh di triangoli). La mesh viene poi filtrata/interpolata.



Filtraggio o interpolazione



Ricostruzione a mesh di triangoli



Ricostruzione Finale

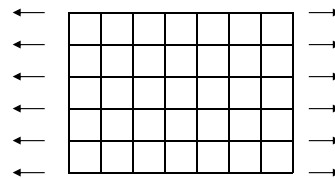
A.A. 2014-2015

43/73

<http://borghese.di.unimi.it/>



Tibia Left



Topologia: cilindro

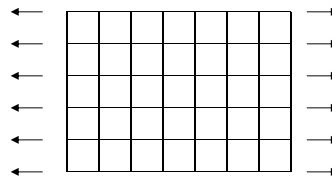
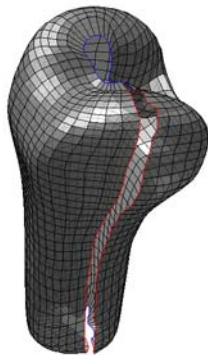
A.A. 2014-2015

44/73

<http://borghese.di.unimi.it/>



Femoral Bone Left



Topologia: cilindro

A.A. 2014-2015

45/73

<http://borghese.di.unimi.it/>



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. 2014-2015

46/73

<http://borghese.di.unimi.it/>



Other methods



Balloons

Level sets

Volume carving



Kinect fusion = camera motion + 3D reconstruction



Riassunto



- Soft Clustering
- Mappe di Kohonen