

Sistemi Intelligenti 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. 2013-2014

1/73

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



Riassunto



- **Clustering partitivo**
- Mappe di Kohonen
- Regressione
- Regressione multi-scala
- Classificazione

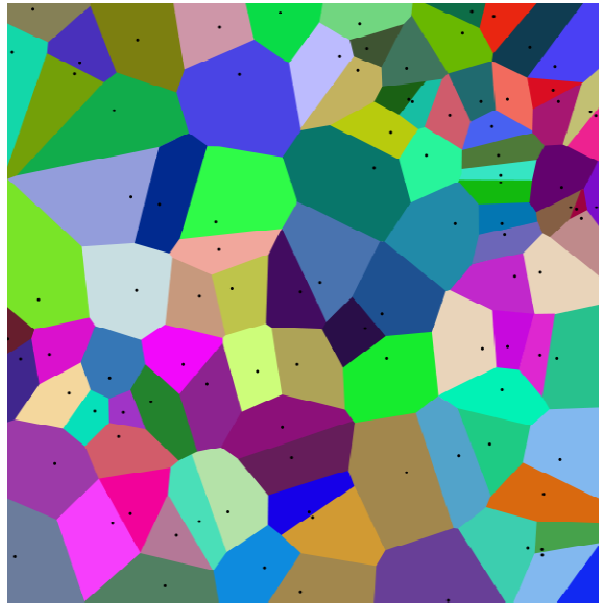
A.A. 2013-2014

2/73

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



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. 2013-2014

3/73

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



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. 2013-2014

4/73

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



Algoritmo K-means



L'obiettivo che l'algoritmo si prepone 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).

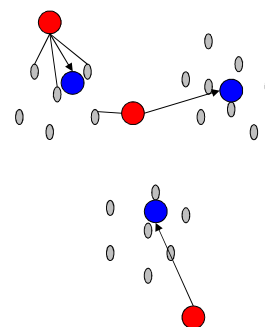
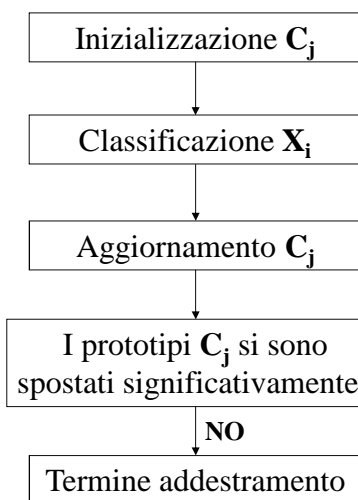
A.A. 2013-2014

5/73

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



K-means: addestramento



Aggiornamento C_j : baricentro degli X_i classificati da C_j .

A.A. 2013-2014

6/73

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



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:

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

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

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

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

A.A. 2013-2014

7/73

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



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. 2013-2014

8/73

<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

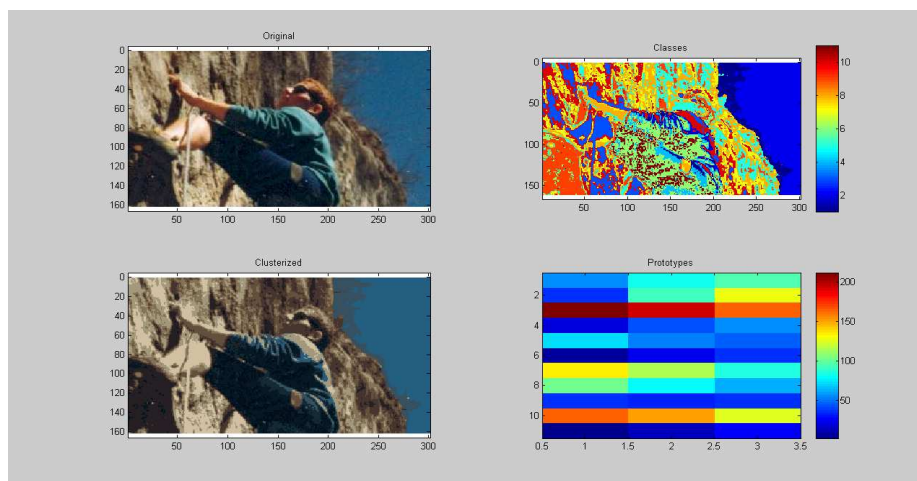
A.A. 2013-2014

9/73

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



K-Means per immagine RGB



Da 255 colori a 33 colori

A.A. 2013-2014

10/73

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



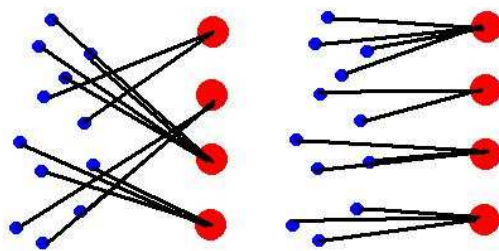
Riassunto



- Clustering partitivo
- **Mappe di Kohonen**
- Regressione
- Regressione multi-scala
- Classificazione



Clustering vs. feature mapping



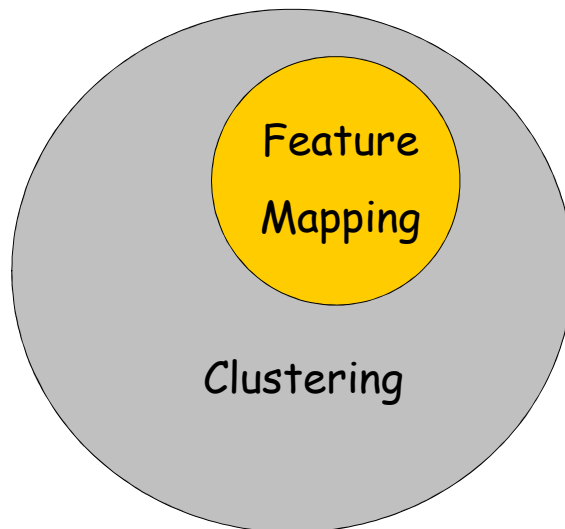
↑
Clustering

↑
Feature Mapping

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



Clustering e feature mapping



A.A. 2013-2014

13/73

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



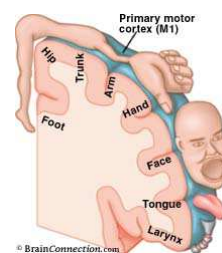
Feature mapping



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. 2013-2014

14/73

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



Self Organizing Maps (SOM)



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

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. 2013-2014 17/73 <http://borghese.di.unimi.it/>

Il campo recettivo: effetto Hermann

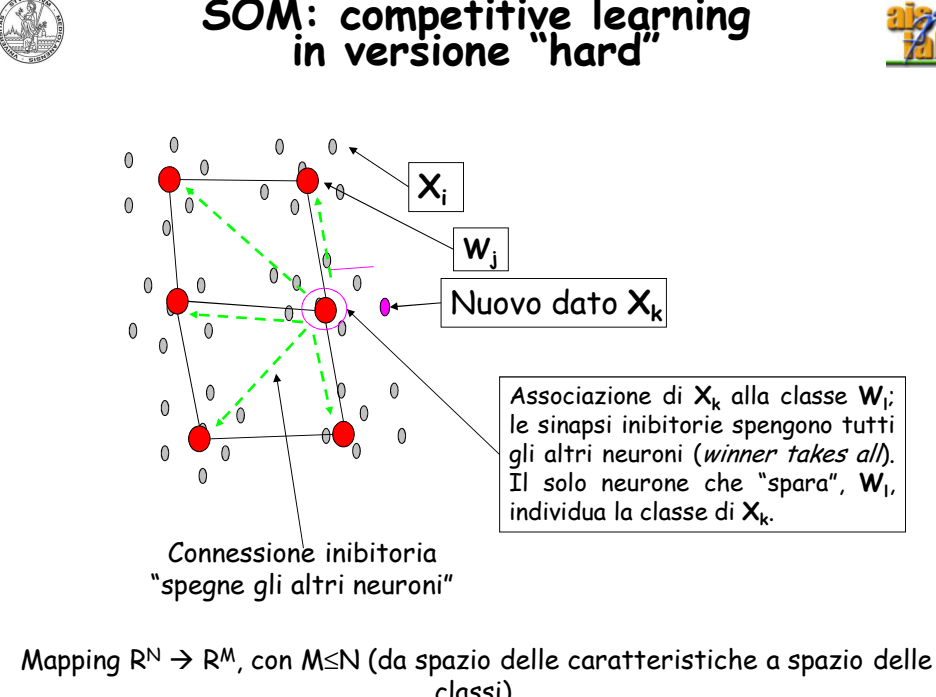
Pallini neri agli incroci delle linee bianche.

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

- - - - -→ Inibitori

A.A. 2013-2014 18/73 <http://borghese.di.unimi.it/>

SOM: competitive learning in versione "hard"



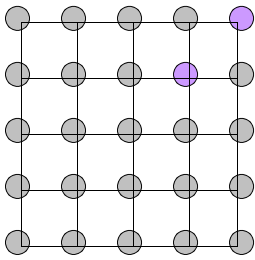
Connessione inibitoria "spegne gli altri neuroni"

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_k .

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.



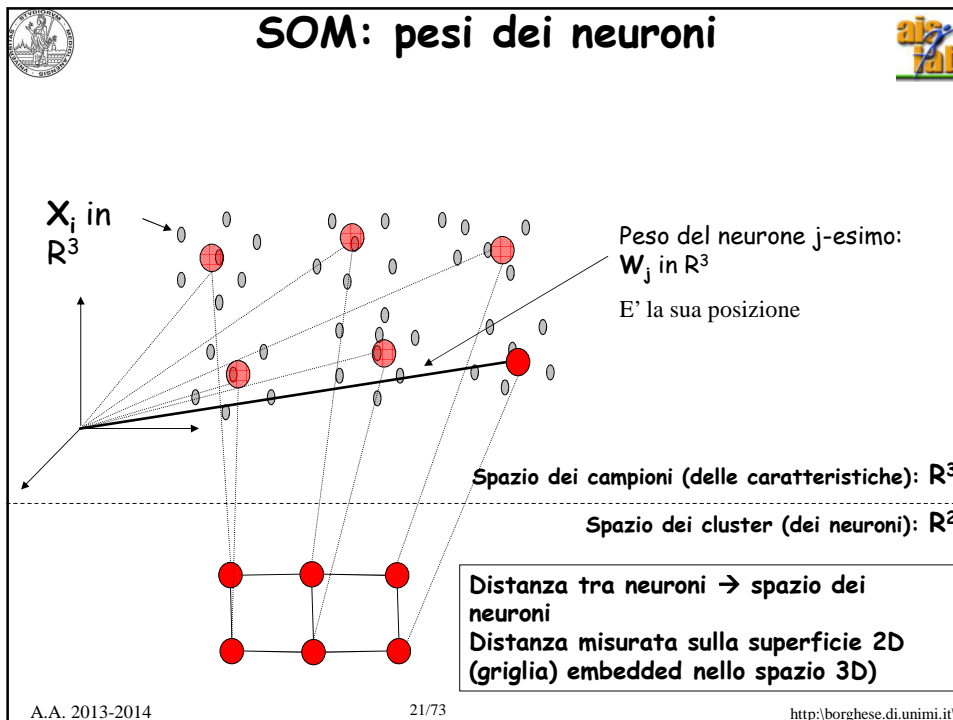
La distanza tra i due neuroni (nello spazio dei neuroni) è:

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

$|\Delta x| + |\Delta y| = 2$ [Manhattan]

...

A.A. 2013-2014 20/73 http://borghese.di.unimi.it/



-
- 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. 2013-2014 22/73 <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. $\|\mathbf{W}_{j^*} - \mathbf{X}_i\| = \min_j \|\mathbf{W}_j - \mathbf{X}_i\|$
- 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 $\underline{0}$ 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.



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):

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

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

UNITA' VINCENTE

AGGIORNAMENTO
PESI DEI NEURONI

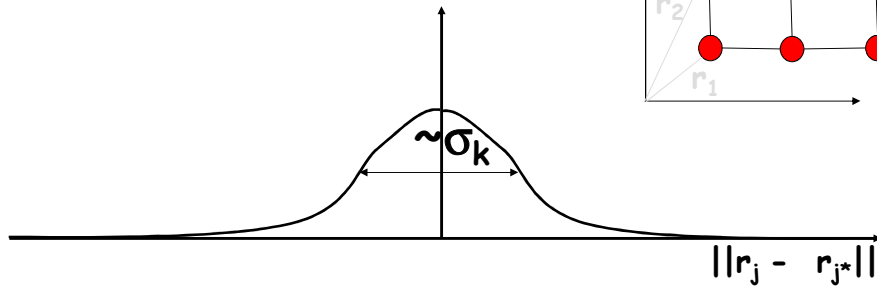
FUNZIONE DI
VICINATO

Calcolata nello spazio
dei prototipi



Funzione di vicinato

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



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

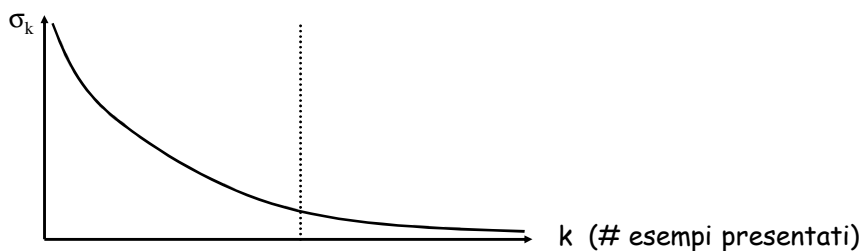
$\sigma_k \rightarrow 0$, l'esponenziale va rapidamente a 0, solo il neurone vincente, j^* , viene aggiornato.

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

imi.it\



Funzione di vicinato nel tempo



$$\Delta w_j = \eta_k \Lambda_k(j, j^*) (X_i - w_j) \rightarrow w' = w + \eta_k \Lambda_k X_i - \eta_k \Lambda_k w$$

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

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

A.A. 2013-2014

28/73

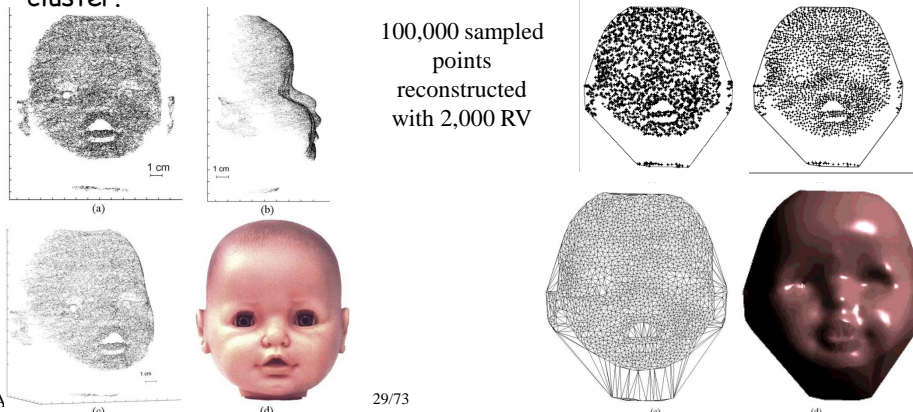
<http://borghese.di.unimi.it/>



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.



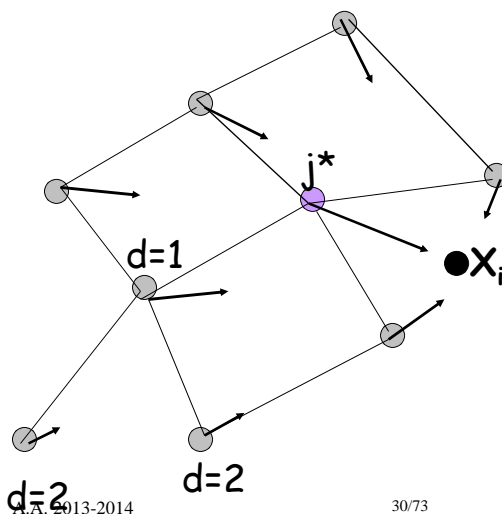
SOM: addestramento



In definitiva:

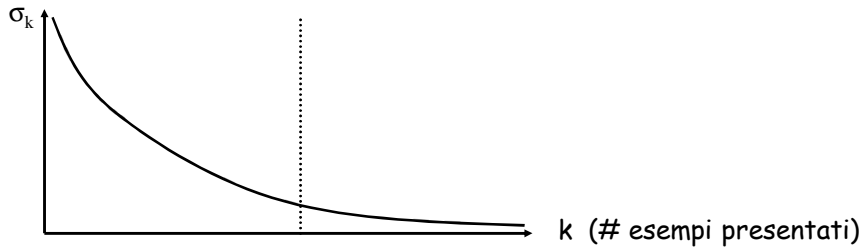
- Il neurone vincente si sposta verso X_i , trascinando i vicini in proporzione alla loro distanza nello spazio delle feature.

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





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.

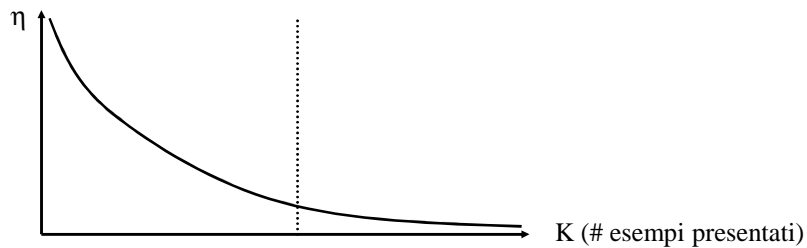
A.A. 2013-2014

31/73

<http://borgese.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. 2013-2014

32/73

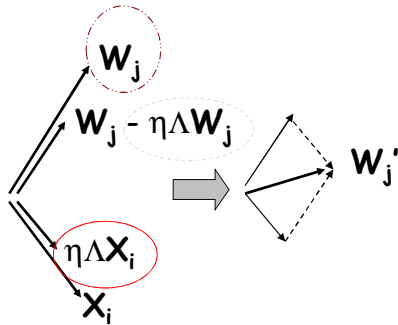
<http://borgese.di.unimi.it/>



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. 2013-2014

35/73

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

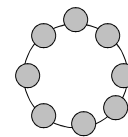


SOM per ordinamento



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. 2013-2014

36/73

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

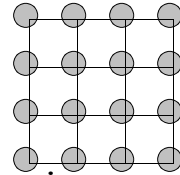


SOM per ricostruzione 3D



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

Topologia della SOM : griglia 2D



Parametri di addestramento
10x10, 0.5 \rightarrow
1 lin, ...

: # neuroni =
0.1 lin, 10 \rightarrow

[Movie here](#)

A.A. 2013-2014

37/73

<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. 2013-2014

38/73

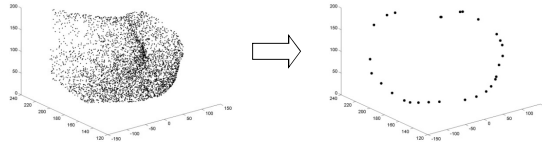
<http://borgese.di.unimi.it/>



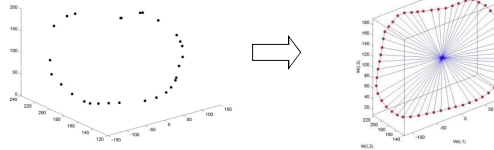
Boundary First Method



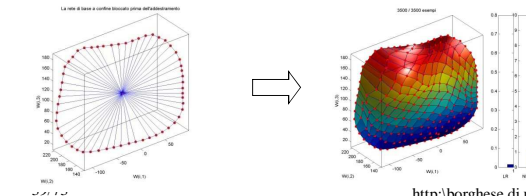
1) Individuazione dei punti di confine della superficie



2) Posizionamento dei neuroni di bordo



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



A.A. 2013-2014

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



SOM per ricostruzione 3D



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

A.A. 2013-2014

40/73

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



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:

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

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

UNITA' VINCENTE

AGGIORNAMENTO
PESI DEI NEURONI

FUNZIONE DI
VICINATO

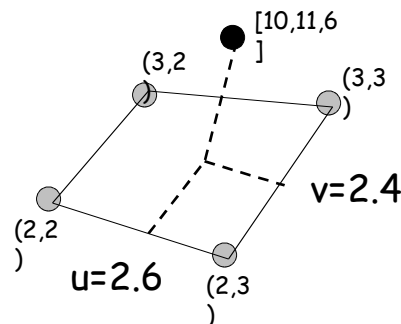
Calcolata nello spazio
dei dati, stesso spazio
dei prototipi.



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)

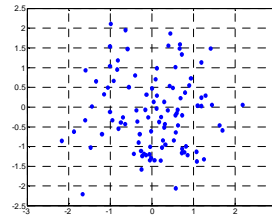




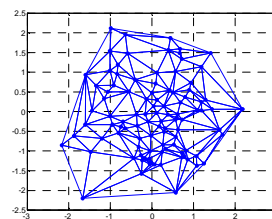
Triangolazione punti in $[u, v]$ (Delaunay)



- 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. 2013-2014

43/73

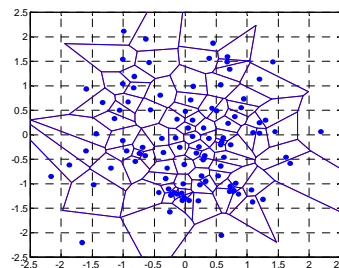
<http://borghese.di.unimi.it/>



Delaunay e Voronoi



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



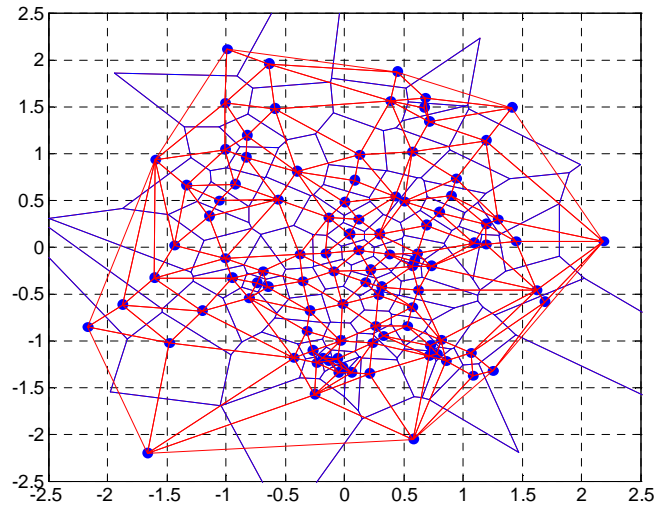
A.A. 2013-2014

44/73

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



Dealunay & Voronoi



A.A. 2011:

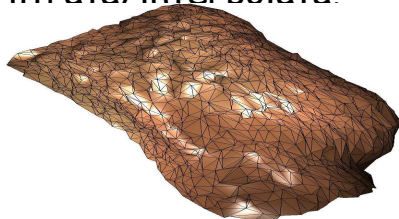
se.di.unimi.it/



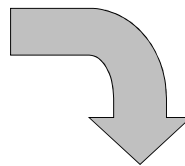
Parametrizzazione Points Cloud



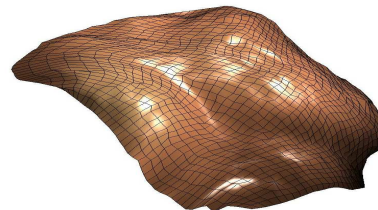
I punti 2D $[u,v]$ possono essere triangolati (ricostruzione a mesh di triangoli). La mesh viene poi filtrata/interpolata.



Ricostruzione a mesh di triangoli



Filtraggio o interpolazione





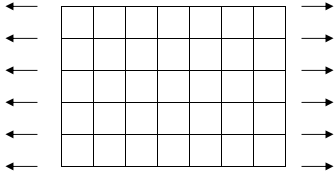
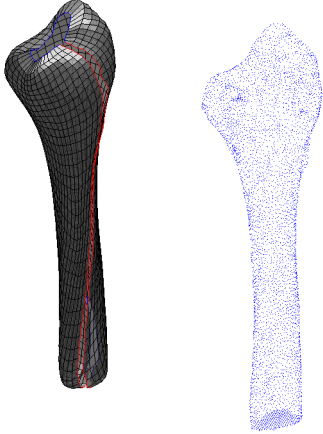
Ricostruzione Finale

A.A. 2013-2014

46/73



http://dorgnese.ai.unimi.it/

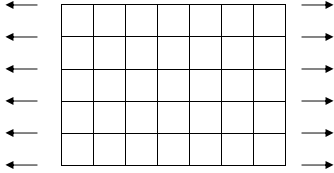
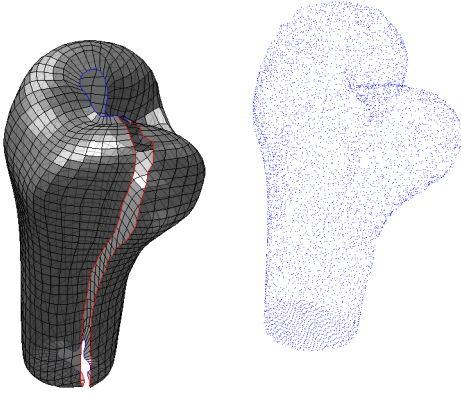
 **Tibia Left** 



Topology: cilindro

A.A. 2013-2014 47/73 <http://borgese.di.unimi.it/>

 **Femoral Bone Left** 



Topology: cilindro

A.A. 2013-2014 48/73 <http://borgese.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. 2013-2014

49/73

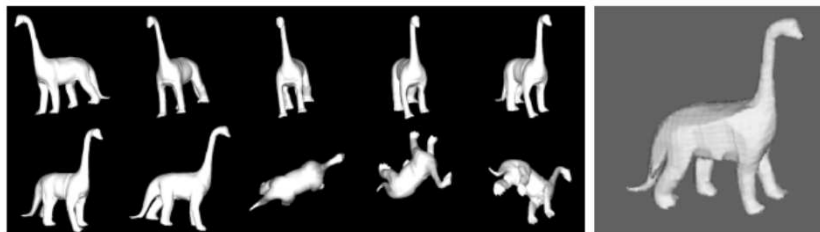
<http://borgese.di.unimi.it/>



Other methods



Balloons
Level sets
Volume carving



Pyon and Shin, 2002

Kinect fusion – camera motion + 3D reconstruction

A.A. 2013-2014

50/73

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



Riassunto



- Clustering partitivo
- Mappe di Kohonen
- **Regressione**
- Regressione multi-scala
- Classificazione

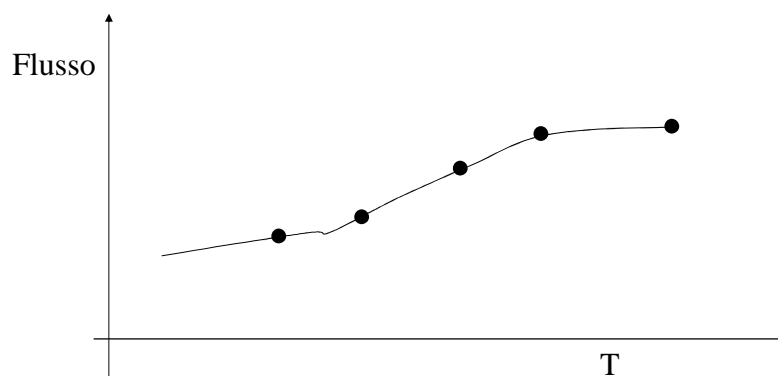
A.A. 2013-2014

51/73

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



Apprendimento supervisionato: **regressione = predictive learning**



Controllo della portata di un condizionatore in funzione della temperatura. “Imparo” una funzione continua a partire da alcuni campioni: devo imparare ad **interpolare**.

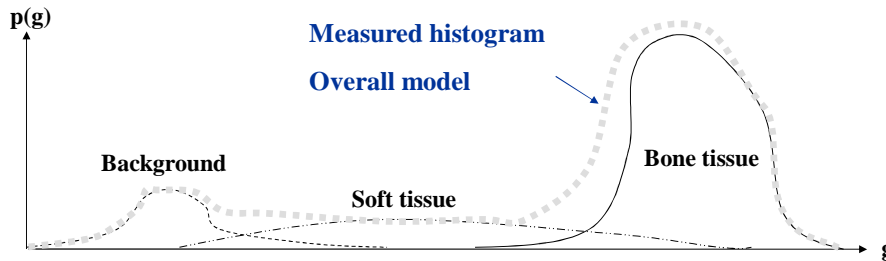
A.A. 2013-2014

52/73

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



I modelli parametrici



$$p(g) = \sum_{j=1}^M P(j) \cdot p(g | j) = \sum_{j=1}^M w_j \cdot p_j(g)$$

La probabilità di avere un livello di grigio g è la somma pesata delle tre probabilità di avere background, $p_1(g)$, tessuto molle, $p_2(g)$ o tessuto osseo, $p_3(g)$.



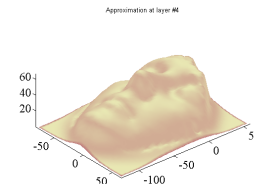
I modelli semi-parametrici

- L'approssimazione è ottenuta mediante funzioni "generiche", dette di **base**, soluzione molto utilizzata nelle NN e in Machine learning. E' anche associato all' approccio «black-box» in teoria dei controlli. Non si hanno informazioni sulla struttura dell'oggetto che vogliamo rappresentare.
- (Il concetto di Base in matematica è definito mediante certe proprietà di approssimazione che qui non consideriamo, consideriamo solo l'idea intuitiva).
- E' anche l'idea che sta alla base delle Reti Neurali Artificiali

$$z(p(x, y)) = \sum_i w_i G(p, p_i; \sigma)$$

Combinazione lineare di funzioni di base

Da calcolare



Funzione di base (fissate)



Modelli lineari e non lineari



Classificazione alternativa dei modelli. Vengono utilizzate classi molto diversi di algoritmi per stimare i parametri di questi due tipi di modelli.

$$z(p(x, y)) = f(x) = \sum_i w_i x \qquad z(p(x, y)) = \sum_i f_i(G(p, p_i; \sigma))$$

$f(\cdot) = w_i$ è funzione lineare

$f(\cdot)$ è funzione non lineare

e.g. $f(\cdot) = e^{G(\cdot)}$

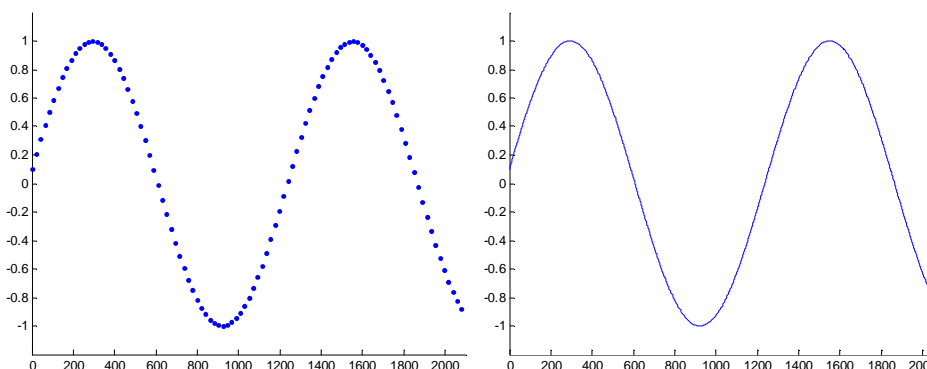
A.A. 2013-2014

55/73

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



Funzionamento di un modello parametrico (non-lineare)



I punti vengono fittati perfettamente da una sinusoide: $y = A \sin(\omega x + \phi)$.
Devo determinare i parametri della sinusoide (non lineare), i cui valori ottimali sono: $\omega = 1/200$, $\phi = 0.1$.

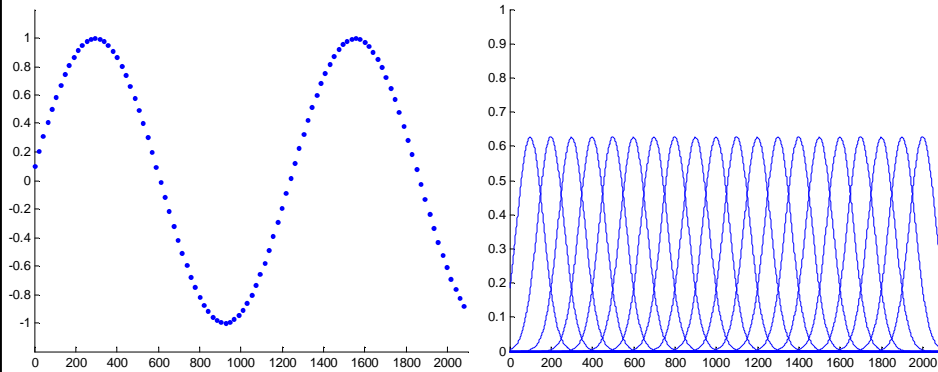
A.A. 2013-2014

56/73

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



Approssimazione mediante un modello semi-parametrico (lineare)



Vogliamo fittare i punti con l'insieme di Gaussiane riportate sulla dx. In questo caso hanno tutte $\sigma = 90$. Come le utilizzo?

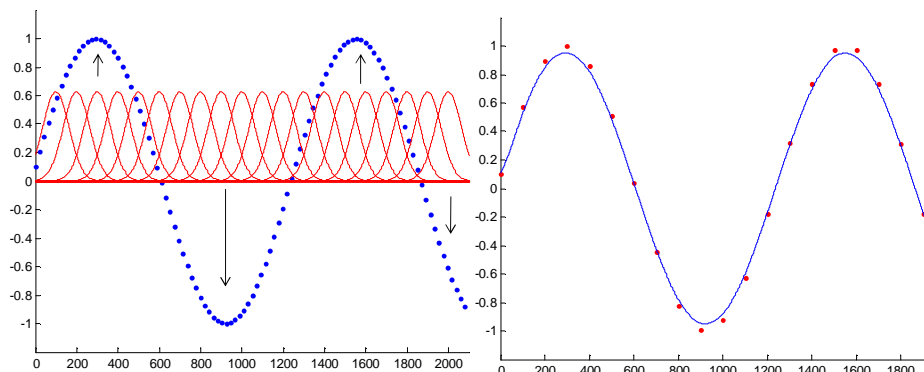
A.A. 2013-2014

57/73

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



Funzionamento di un modello semi-parametrico (lineare)



$$y(x) = \sum_{i=1}^{20} w_i G(x - x_{o_i}; 90)$$

Devo definire, i $\{w_i\}$

I σ sono tutti uguali ed uguali a 90, le Gaussiane sono equispaziate.
Le Gaussiane sono note tutte a priori, devono essere definiti i pesi.

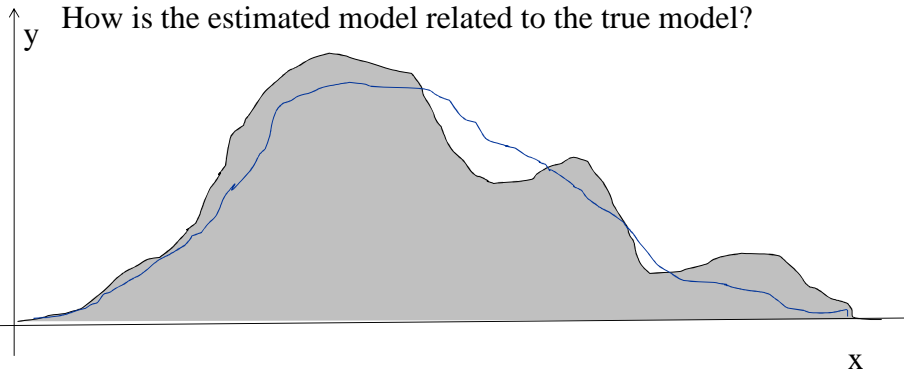
A.A. 2013-2014

58/73

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



How to classify the error introduced by a model?



Bias and variability trade-off

Bias is the distance of the model curve from the true unknown curve. It is associated to model error.

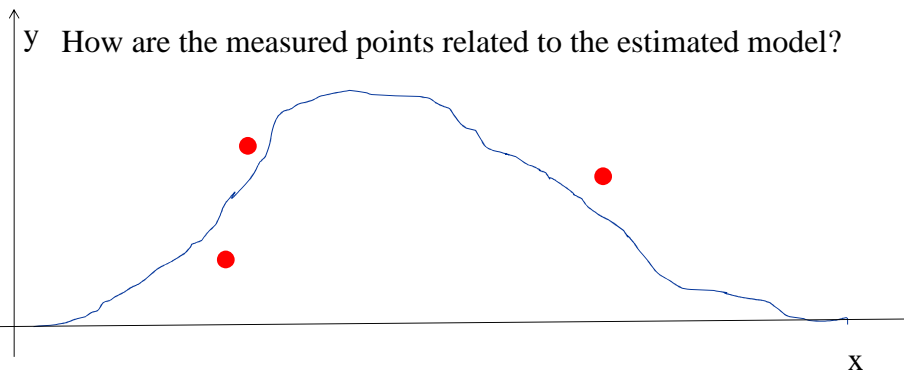
A.A. 2013-2014

59/73

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



Variability



Given $P_{mes}(x,y)$ and $y = y(x)$, the error is measured as:

- $\min \| P - y(x) \|$
- $\| y_{mes}(x_{mes}) - y(x_{mes}) \|$

It is associated to measurement error.

If variability goes to zero, bias increases and overfitting arises.

A.A. Variability approaches measurement noise.

di.unimi.it/



Problemi

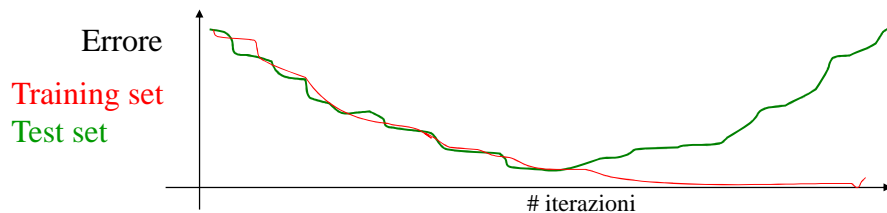
Quando si termina l'algoritmo di apprendimento?

Bootstrap – Vengono estratti pattern con ripetizioni.

Cross-Validation - Errore sull'insieme di training =
Errore sull'insieme di test.

Utilizzare lo “structural risk” invece dell’”empirical risk”.

Si vuole evitare che la rete si specializzi troppo sui pattern di training e non sia in grado di interpolare.



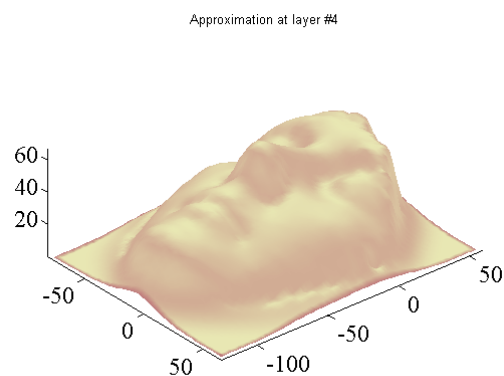
A.A. 2013-2014

61/73

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



Problema dell'overfitting dovuto a sovrapparametrizzazione



Quante unità?

A.A. 2013-2014

62/73

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



Problemi

Quando si termina l'algoritmo di apprendimento?

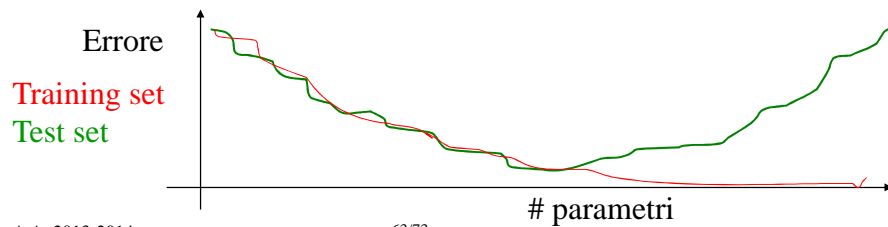
Bootstrap – Vengono estratti pattern con ripetizioni.

Cross-Validation - Errore sull'insieme di training =

Errore sull'insieme di test.

Utilizzare lo “structural risk” invece dell’”empirical risk”.

Si vuole evitare che la rete si specializzi troppo sui pattern di training e non sia in grado di interpolare.



A.A. 2013-2014

63/73

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



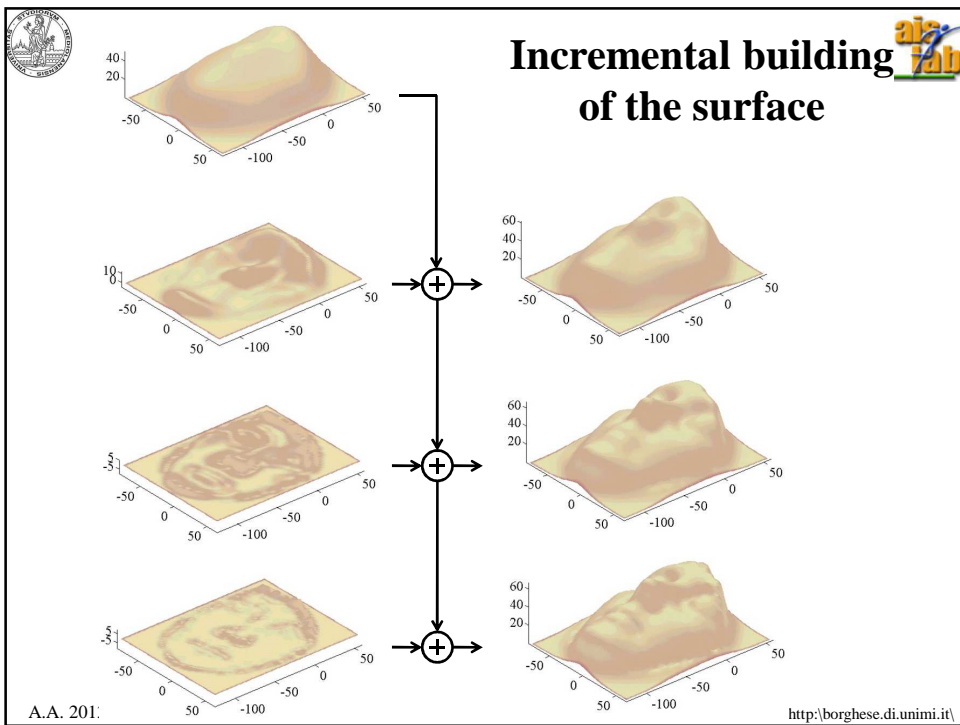
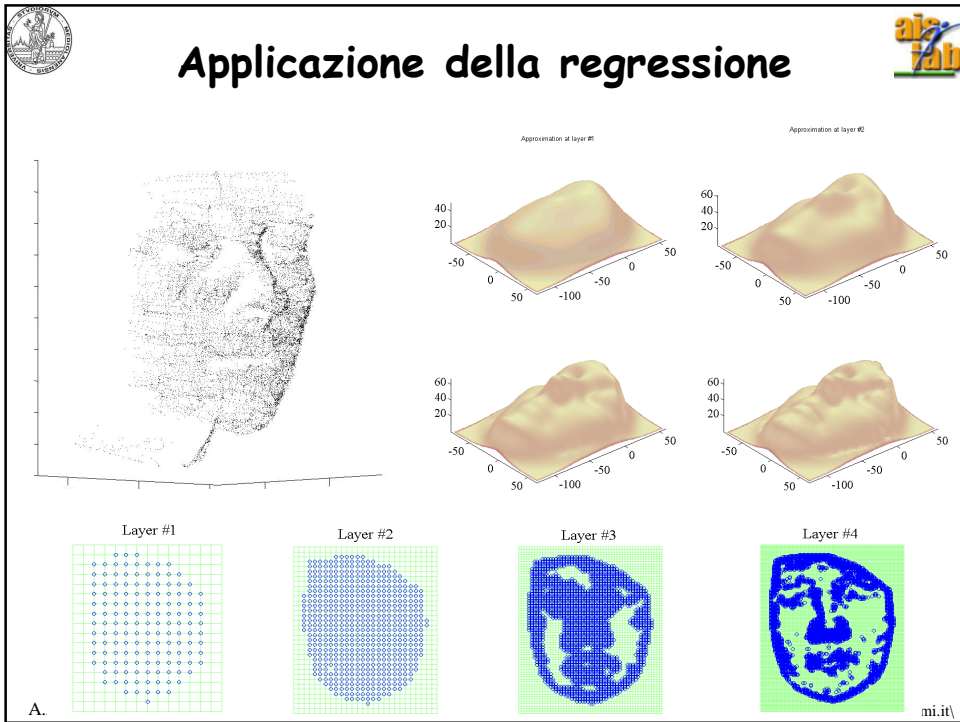
Riassunto

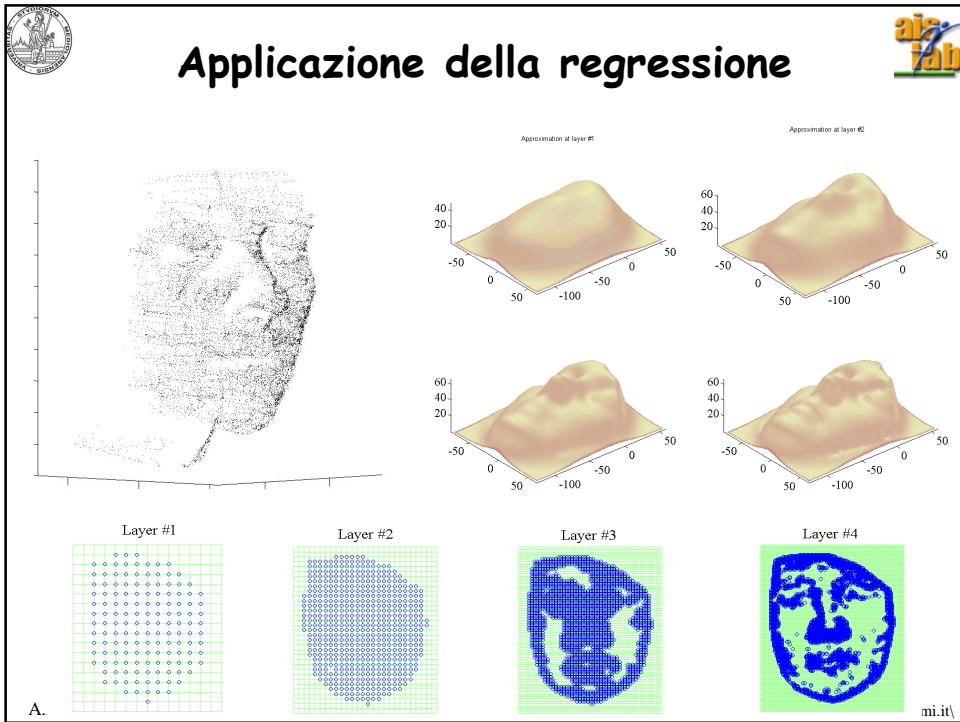
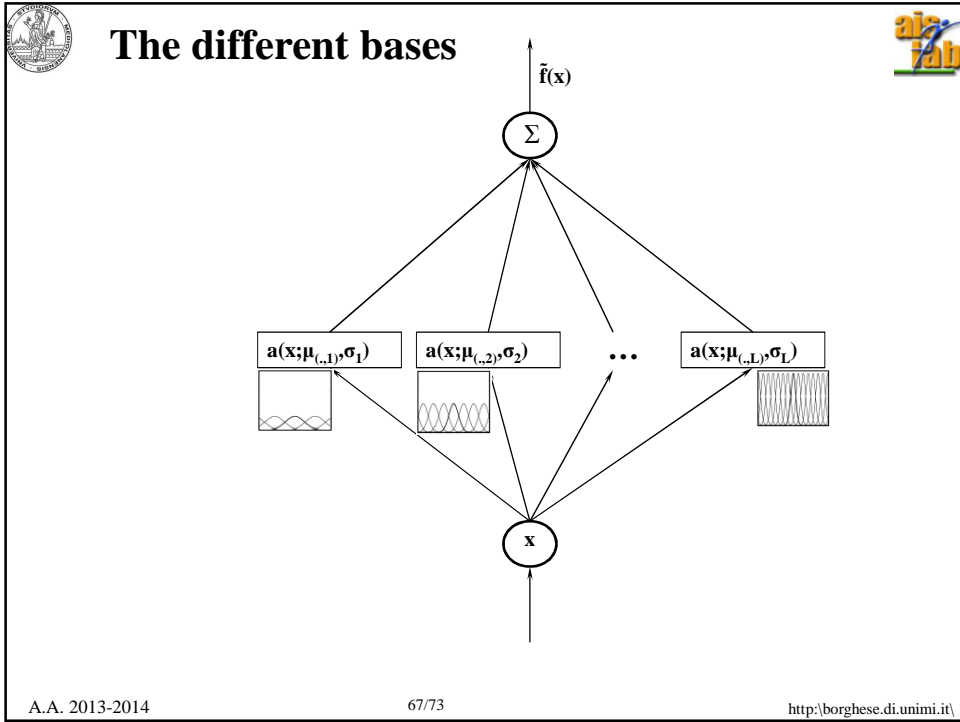
- Clustering partitivo
- Mappe di Kohonen
- Regressione
- **Regressione multi-scala**
- Classificazione

A.A. 2013-2014

64/73

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







Riassunto

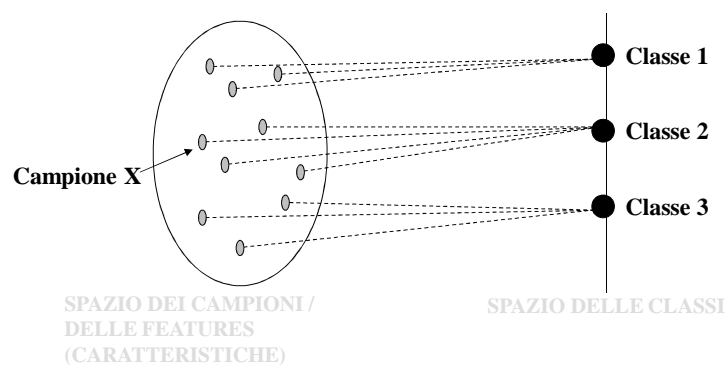
- Clustering partitivo
- Mappe di Kohonen
- Regressione
- Regressione multi-scala
- **Classificazione**





Classificazione

Un'interpretazione geometrica:

Mappatura dello spazio dei campioni nello spazio delle classi.





Apprendimento Supervisionato:
Classificazione

b
 b
 b
 b
 b
 → B

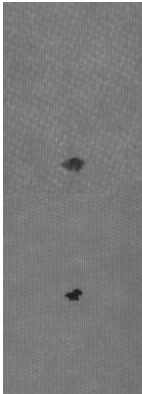
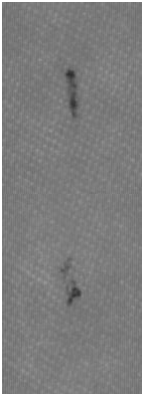
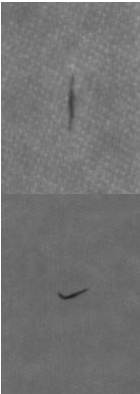
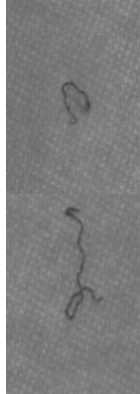
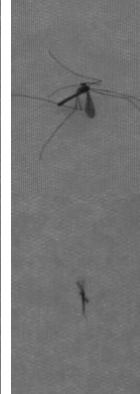
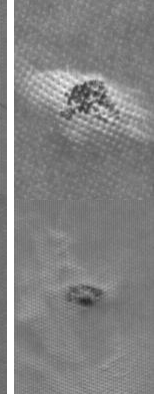
a
 a
 a
 a
 a
 → A

Task di classificazione
 Uscita intera (etichetta o label della classe)

A.A. 2013-20 http://borgese.di.unimi.it/

CLASSIFICAZIONE: Riconoscimento difetti in linee di produzione
 (progetto finanziato da Electronic Systems: 2006-2007)

					
regolari	irregolari	allungati	fili	insetti	macchie su denso

Difetti – Classificazione real-time e apprendimento mediante **boosting**.
 Committee (linear combination) of weak (binary) classifiers.

A.A. 2013-2014 http://borgese.di.unimi.it/



Riassunto



- Clustering partitivo
- Mappe di Kohonen
- Regressione
- Regressione multi-scala
- Classificazione