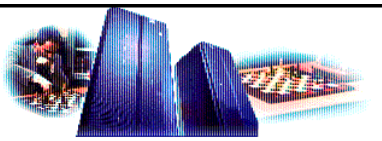
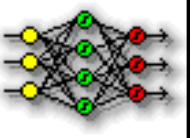




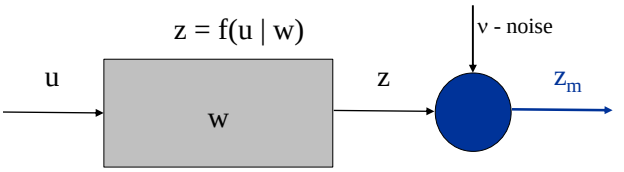
Sistemi Intelligenti
Supervised learning

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

Riassunto

- Supervised learning: predictive regression
- Regressione multi-scala
- Versione on-line
- Valutazione del modello

Modello



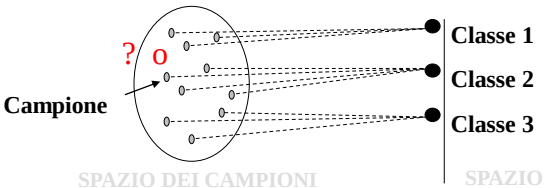
u – causa => z_m – effetto (misurato con errore)

Control / Classification / Prediction: determine {z} from {u},{w}
Inverse problem: determine cause {u} from $\{z_m\}$,{w}
Inverse problem: Identification: determine {w} from {u},$\{z_m\}$ - *Learning*

f(u|w) è un modello, rappresentazione di una realtà: policy, Value function, Environment…

Classificazione e regressione

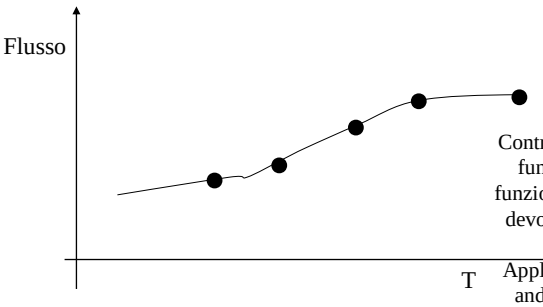
Mappatura dello spazio dei campioni nello spazio delle classi.



SPAZIO DEI CAMPIONI / DELLE FEATURES (CARATTERISTICHE)

SPAZIO DELLE CLASSI (identificate da un'etichetta)

Classifico ("aiuto alla clusterizzazione")



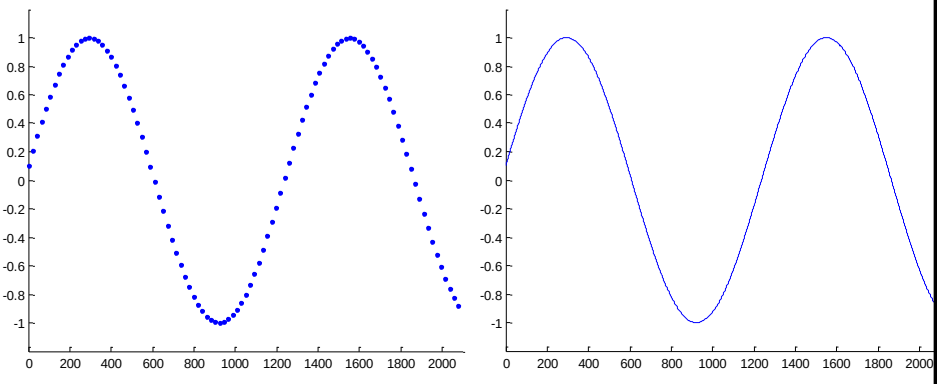
Quanto vale ?

Controllo della portata di un condizionatore in funzione della temperatura. "Imparo" una funzione continua a partire da alcuni campioni: devo imparare ad **interpolare** (regressione = predictive learning).
Applicaizoni alle serie temporali: ad esempio andamento borsa, previsioni del tempo,….

Ruolo dei modelli

- **Identificazione:** stimo i parametri di un modello a partire dai dati: identifico il modello.
- **Utilizzo 1:** utilizzo il modello per inferire informazioni su nuovi dati (controllo, regressione predittiva, classificazione).
- **Utilizzo 2:** utilizzo il modello per inferire informazioni sulla causa di un effetto.

Modello parametrico



I punti vengono fittati perfettamente da una sinusoide: $y = \mathbf{A} \sin(\boldsymbol{\omega} x + \boldsymbol{\phi})$. Devo determinare solo i 3 parametri della sinusoide (non lineare), i cui valori ottimali sono: $\omega = 1/200$, $\phi = 0.1$, $A = 1$. I parametri hanno un significato semantico.

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 cibernetica. Non si hanno informazioni sulla struttura dell'oggetto che vogliamo rappresentare.
- E' anche l'idea che sta alla base delle Reti Neurali Artificiali

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

Combinazione lineare di funzioni di base

Da calcolare

Classificazione

- Boosting. Si utilizza un insieme di classificatory binary, dove ciascun classificatore lavora su una singola feature. La classificazione avviene prendendo la maggioranza di voto dei classificatory.
- Reti neurali. Approccio black-box generale.
- Support Vector Machines. Calcolo la linea di separazione che massimizza il margine, cioè che passa più lontana dai punti delle due classi. La linea può essere una spezzata (lineare) oppure una curva (non-lineare).

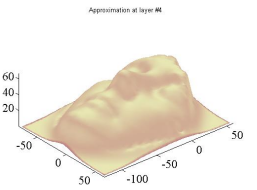
Modelli supportati da una base

- **Costituenti del modello equispaziati e tutti con gli stessi parametri (in questo caso σ).**
- (Il concetto di Base in matematica è definito mediante certe proprietà di approssimazione che qui non consideriamo, consideriamo solo l'idea intuitiva).
- Il concetto di base è simile a quello dei "replicating kernels".

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

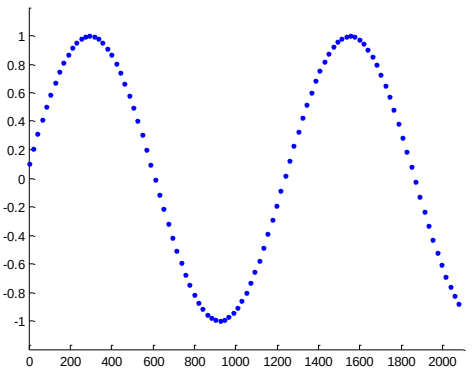
Combinazione lineare di funzioni di base

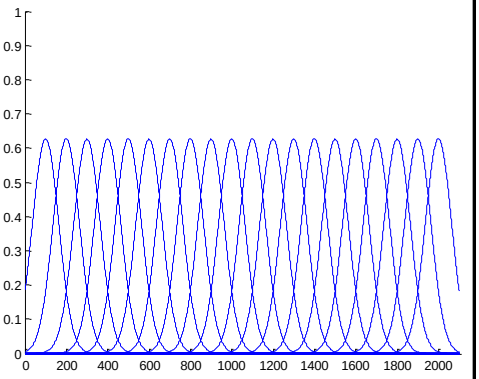
Da calcolare



Funzione di base (fissate)

Approssimazione mediante un modello semi-parametrico (lineare)

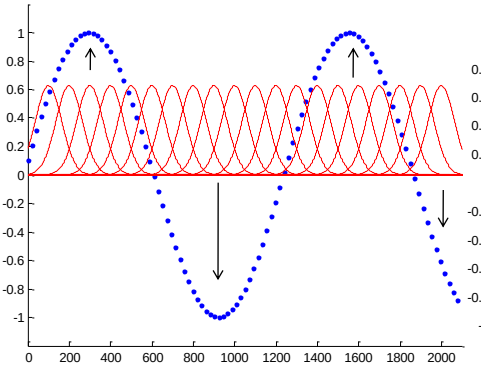


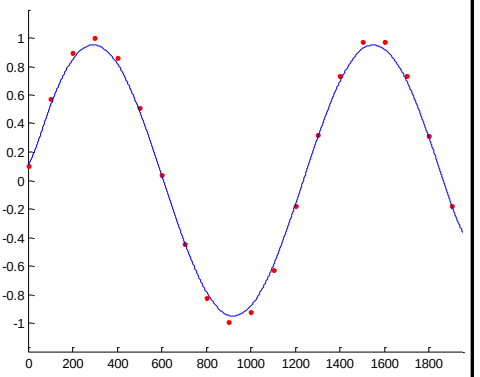


Sinusoide y = **A** sin(**ω**x + **ϕ**) con ω = 1/200, ϕ = 0.1.

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

Funzionamento di un modello semi-parametrico (lineare)





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

Devo definire, gli M $\{w_i\}$.
3 << M << N – numero punti.

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.

Model as a filter (convolution)

- Convolution: $\hat{f}(x)=\int_{\mathbb{R}} f(c)\,G(x-c|\sigma)\,\mathrm{d}c = f(x)*G(x;\sigma)$

we can construct output up to a certain scale (level of detail), provided an adequate small value of σ.

- Discrete convolution: $\hat{f}(x)= f_i * G(x-x_{k_i};\sigma)=\sum_{i=1}^{N} w_i G(x-x_{k_i};\sigma)$

The construction of the output, if G(.) is normalized, is obtained through digital filtering.

Extrapolation beyond the sample points. Continuos reconstruction up to a given scale.

Convolutional networks.

Filters and bases

$$\hat{f}(x) = \sum_k f_k * G(x - x_k; s)$$

$$\hat{f}(x) = \sum_{k=1}^{N} f_k G(x; x_k, \sigma)\Delta x = \frac{\Delta x}{\sqrt{\pi}\,\sigma} \sum_{k=1}^{N} f_k e^{-\frac{(x-x_k)^2}{\sigma^2}} \qquad \frac{\Delta x_k}{\sqrt{\pi}\sigma}$$ Normalization factor

Normalized Gaussians, filter = weighed sum of shifted (normalized) basis functions. Basis representation. Approximation space.

Riesz basis, the approximation space is characterized by the scale of the basis that determines the amplitude of the space.

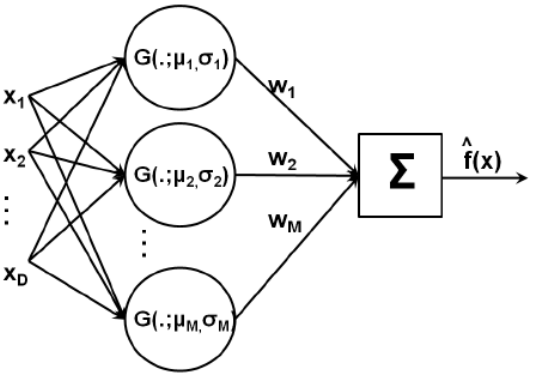
A sequence of spaces can be defined according to σ:

σ_0 -> V_0; σ_1 -> V_1; σ_2 -> V_2….

The number of representable functions increases.

RBF Network

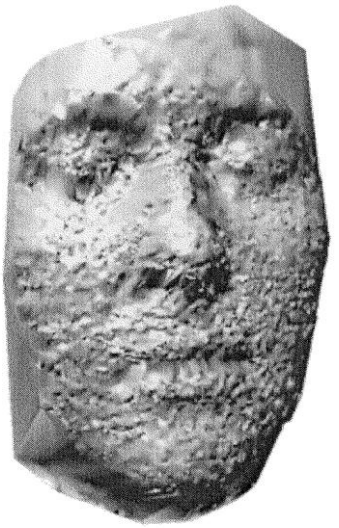
Connessionism. Simple processing units combined with simple operations to create complex functions.

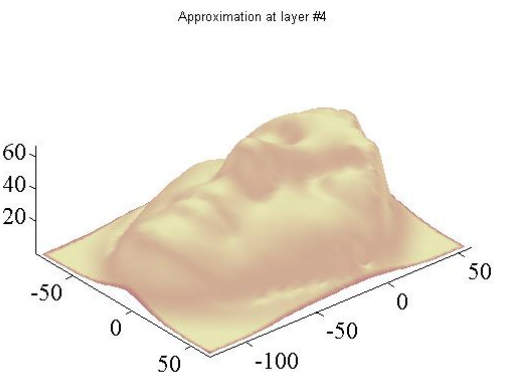


Perceptron

Esempio: scanner 3D

z = f(x,y | w) - altorilievo





Quante unità?

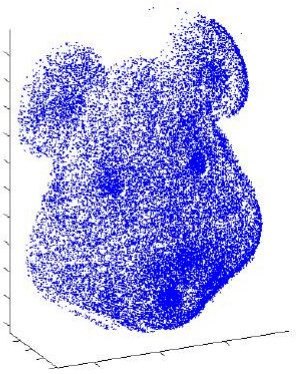
Problema dell'overfitting dovuto a sovra-parametrizzazione

Advantages and problems

$$\hat{f}(x) = \sum_{k=1}^{N} f_k G(x, x_k, \sigma)\Delta x = \frac{\Delta x}{\sqrt{\pi}\,\sigma} \sum_{k=1}^{N} f_k e^{-\frac{(x-x_k)^2}{\sigma^2}}$$

Filters interpolates data and reduces noise but...

Height of the surface on a grid crossing should be known.



Gridding

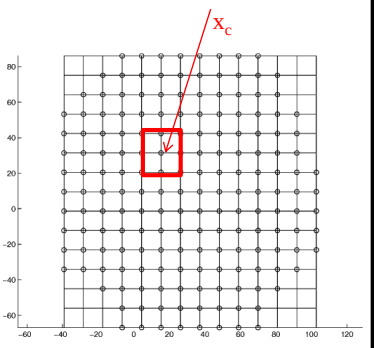
$$\hat{f}(x) = \sum_{k=1}^{N} f_k G(x; x_k, \sigma)\Delta x = \frac{\Delta x}{\sqrt{\pi}\,\sigma} \sum_{k=1}^{N} f_k e^{-\frac{(x-x_k)^2}{\sigma^2}} = \sum_{k=1}^{N} w_k e^{-\frac{(x-x_k)^2}{\sigma^2}}$$

How can we determine w_k from points clouds?

Local estimators. Nadaraya Watson estimator. *Lazy learning*.

$$\hat{f}(x_c) = \frac{\sum_i y_i K_\sigma(x_i, x_c)}{\sum_i K_\sigma(x_i, x_c)} = \frac{\sum_i y_i e^{-\frac{\|x_i - x_c\|^2}{\sigma^2}}}{\sum_i e^{-\frac{\|x_i - x_c\|^2}{\sigma^2}}} \setminus$$

$K_\sigma(.)$ Gaussiana



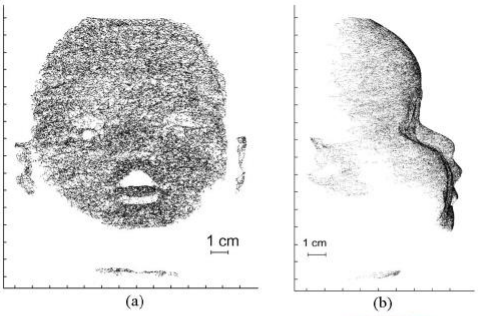
Parzen-window estimators.

Example: 3D scanner

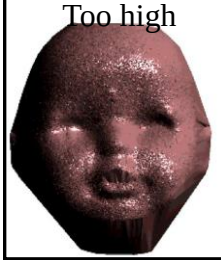
- Properties:

- Redundancy.
- Riesz basis (unique representation, given the height in the grid crossings).

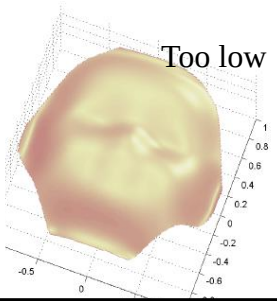
Which scale?

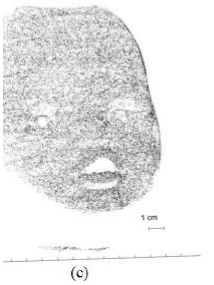


Too high



Too low



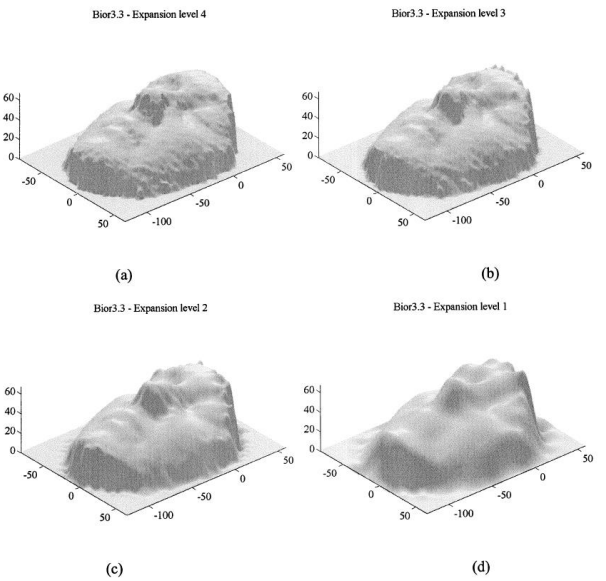


Riassunto

- Supervised learning: predictive regression
- Regressione multi-scala
- Versione on-line
- Valutazione del modello

Pyramidal reconstruction

- Which is the adequate scale?
- Which model is the closest to the true model?



Incremental strategy

- Acquire more data in the more complex areas, less smooth, higher frequency.
- Acquire less data in the less complex areas, more smooth, lower frequency.

$$\hat{f}(x) = \sum_{k=1}^{N} f_k \, G(x; x_k, \sigma) \Delta x = \frac{\Delta x}{\sqrt{\pi}\,\sigma} \sum_{k=1}^{N} f_k e^{-\frac{(x-x_k)^2}{\sigma^2}}$$

- Can we use a single Δx?

Incremental approximation with local adaptation.

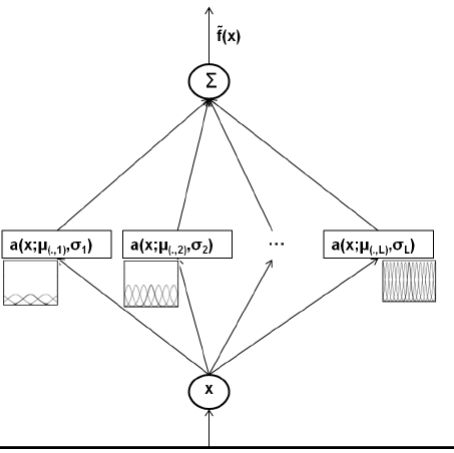
Start from low resolution

- Low resolution, small distance, $1/\Delta x > 2\nu_{Max}$

$$\hat{f}(x) = \sum_{k=1}^{N} f_k \, G(x; x_k, \sigma) \Delta x = \frac{\Delta x}{\sqrt{\pi}\,\sigma} \sum_{k=1}^{N} f_k e^{-\frac{(x-x_k)^2}{\sigma^2}}$$

σ determines the amount of overlap. It determines also the frequency content of the Gaussian G(.).

Once σ (or Δx is defined) the grid and mesh size are also defined.



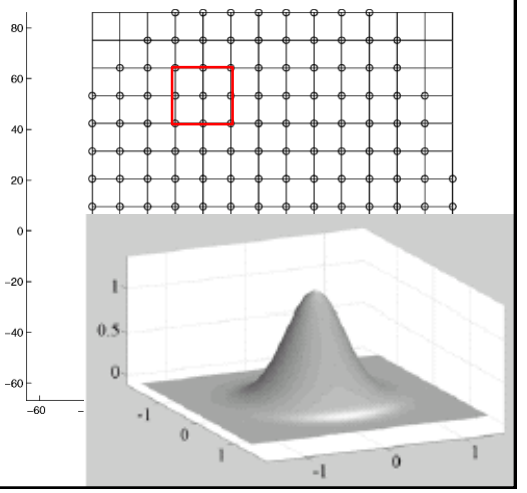
Determination of the surface height

How many points to consider? The Gaussian has infinite support. Splines have a limited support.

$$\hat{f}(x) = \sum_{k=1}^{N} f_k \, G(x, x_k, \sigma) \Delta x$$

Apply local estimator to the data points in the neighbourhood of a grid crossing (Gaussian center) to compute f_k.

Sorting of the data is made simple, they are subdivided into quads. Identified the points inside the neighbourhood is equivalent to extract all the points between two positions in the data vector.



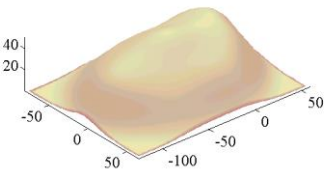
We can obtain a «poor» reconstruction

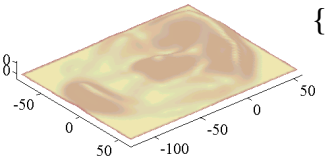
But it is a start. It can be seen as a modified support for successive approximations.

What can be done?

We can compute the residual for each data point.

Approximation at layer #1





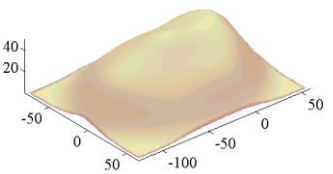
$\{r_1(\mathbf{x})\}$

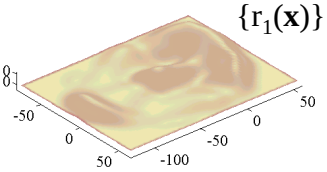
We evaluate the residual for each data point: $r_1 = dist\left(y_m, \hat{f}(x_m)\right)$

E.g.: $r_1 = \left(y_m - \hat{f}(x_m)\right)^2$ $r_1 = \left|y_m - \hat{f}(x_m)\right|$

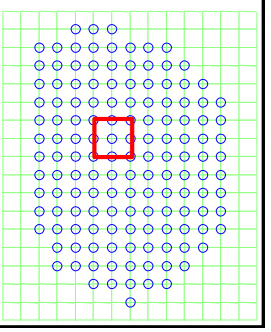
Is the residual adequate?

Approximation at layer #1





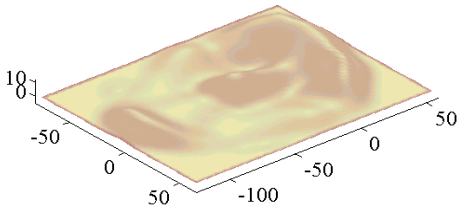
$\{r_1(\mathbf{x})\}$

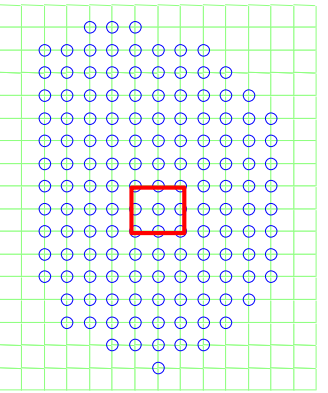


For each Gaussian the integral of the residual inside the "receptive field" of the Gaussian, is assumed as local approximation error associated to it. , is computed inside its "receptive field":

$$R(x_c) = \frac{\sum_m r_m}{N_k}$$

How can we evaluate the local adequacy of the reconstruction?





$$R(x_c) = \frac{\sum_m r_m}{N_k}$$

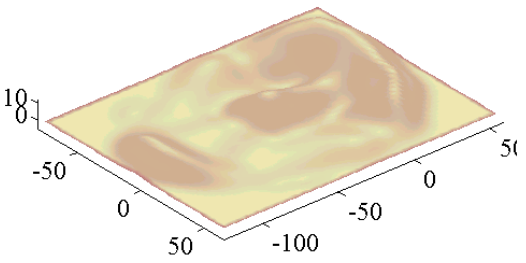
We compare the local residual it with a threshold:
- Degree of approximation
- Noise: RMS.

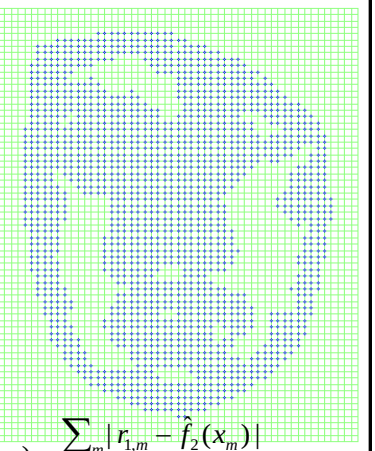
Layer 2

Input are the residuals, $r_{1,m} = \left| y_m - \hat{f}_1(x_m) \right|$
Output is the model that approximates $r_{1,m}$: $f_2(x_m) \rightarrow r_{1,m}$

Layer #2

Output of layer #2

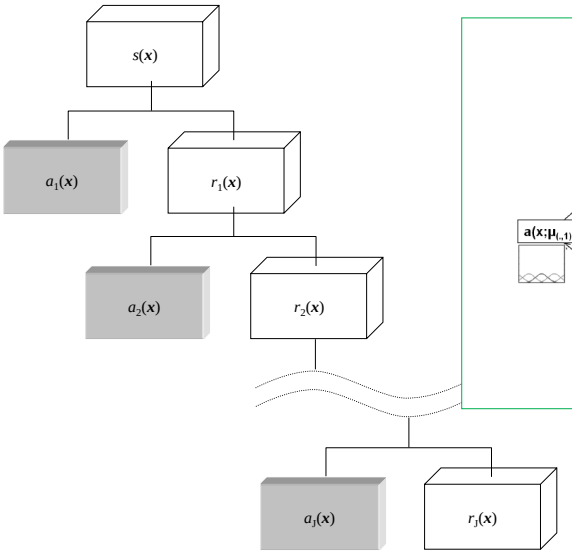


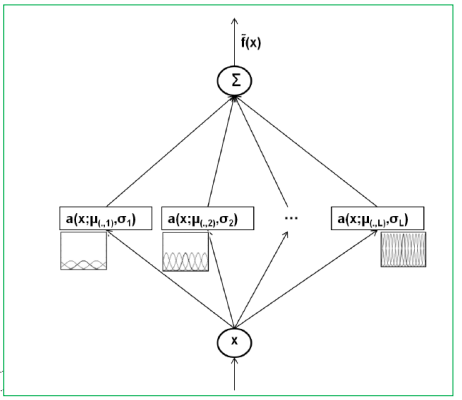


More packed Gaussians
There should be enough points to have a reliable local estimate of not filled grid.

$$R(x_c) = \frac{\sum_m \left| r_{1,m} - \hat{f}_2(x_m) \right|}{N_k}$$

Hierarchy construction

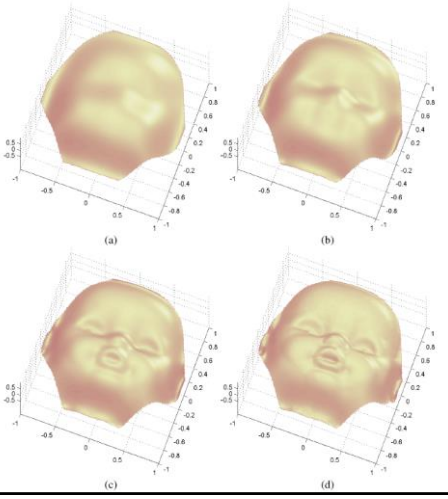


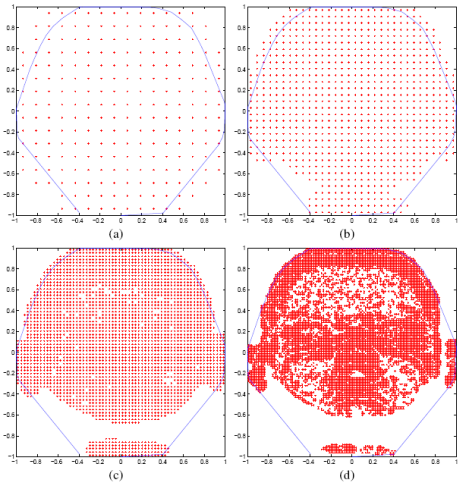


and use as a stack of layers

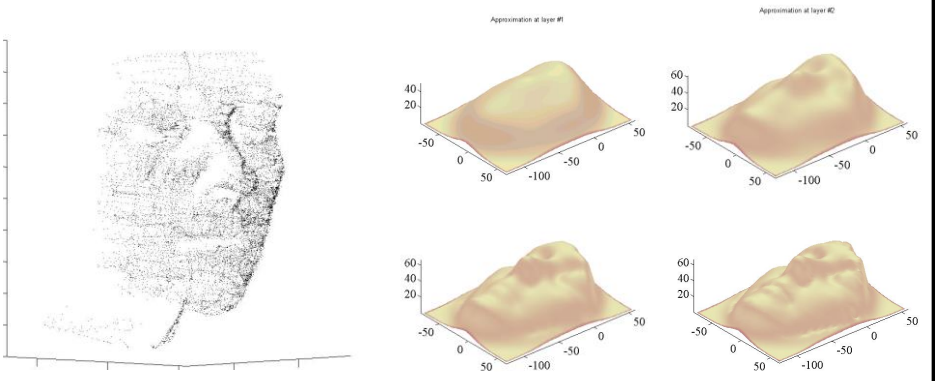
How to operate on large sets of data?

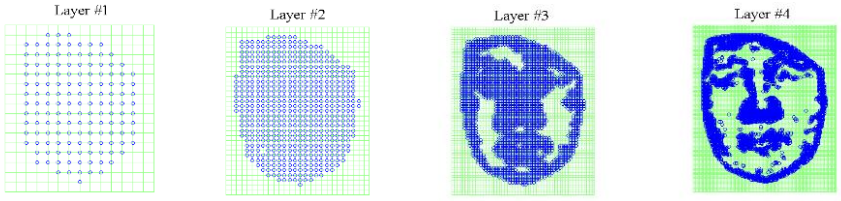
Recursive splitting of the quad domain -> local re-ordering of the data.





Applicazione della regressione



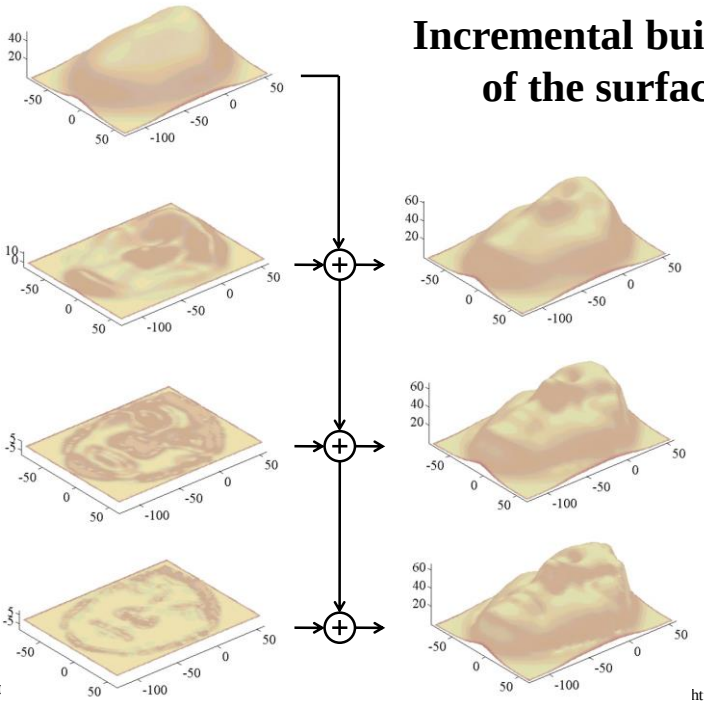


Characteristics of HRBF networks

- Not fully occupied layers
- Adaptive local scale
- Adaptive allocation of the resources
- Uniform convergence to a residual error
- Residual bias is recovered in the next layers.
- Relatively dense data sets are required to obtain a robust local estimate.
- Riesz basis, with a high degree of redundancy between the coefficients. The angle between two approximating spaces is not 90, but it is considerably smaller

$$\cos\,\alpha_j = \sup_{f(\cdot)\in V_j, h(\cdot)\in V_{j+1}} \frac{\langle f(\cdot), h(\cdot)\rangle}{\|f(\cdot)\|_2 \|h(\cdot)\|_2} = \cos\alpha_{j-1}.$$

Incremental building of the surface



Riassunto

- Supervised learning: predictive regression.
- Regressione multi-scala
- Versione on-line
- Valutazione del modello

On-line version

- Data do not arrive all together (batch)
- One data at a time.
- Growing while scanning

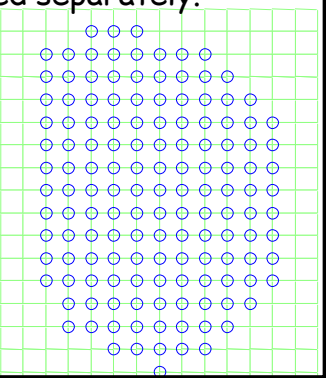
hrbf_online.wmv

Observation

- Each new point, $y=f(x_k)$, modifies at least f_1 around x_k.
- This in turns can modify 4 values in the next layer and so forth.

Recomputation can be simplified:
Numerator and denominator are stored separately.

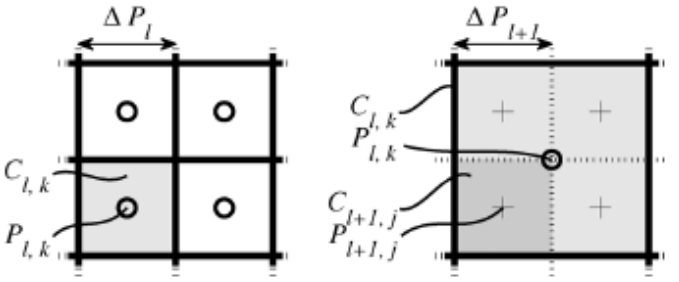
$$\hat{f}(x) = \frac{\sum_i y_i K_\sigma(x_i,x)}{\sum_i K_\sigma(x_i,x)} = \frac{\sum_i y_i e^{\frac{-||x_i-x||^2}{\sigma^2}}}{\sum_i e^{\frac{-||x_i-x||^2}{\sigma^2}}}$$



For each new point a new term is added and the ratio is recomputed.

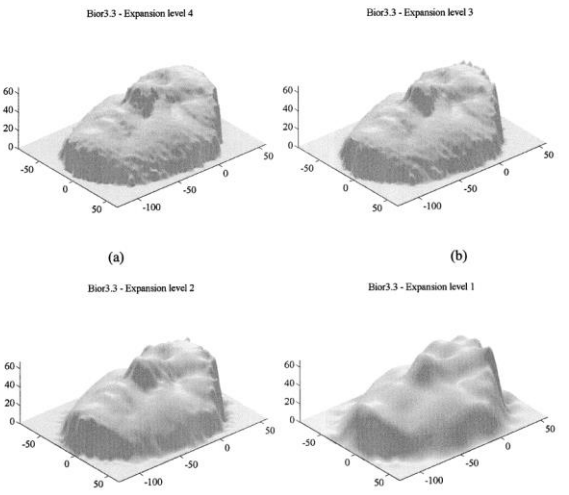
Local operations

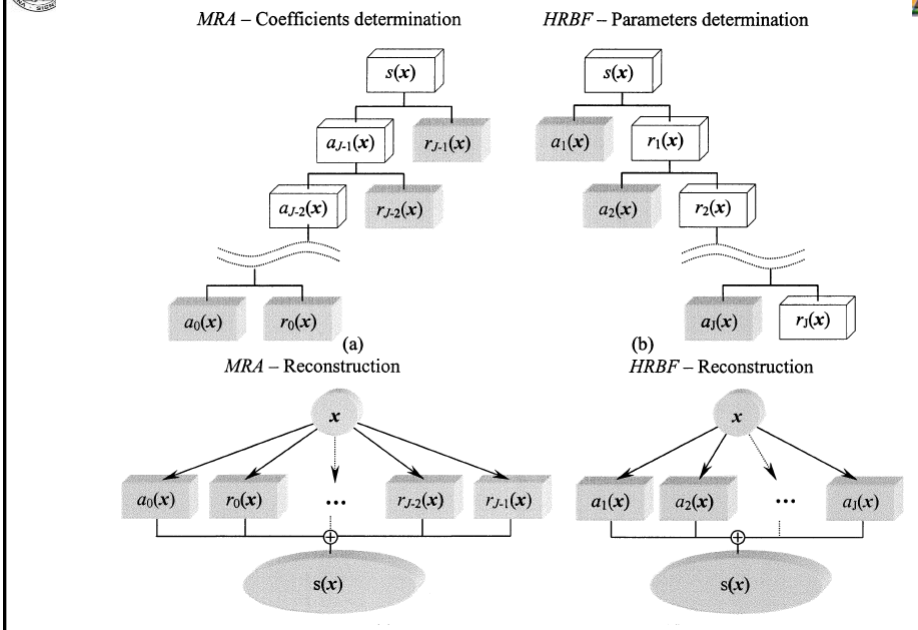
- Local splitting of each quad is achieved when:
  - Residual is higher than threshold
  - Enough points have been sampled



Comparison with Wavelets

- Fast incorporation of the content (high angles between approximating spaces -> 90 degrees)
- No control on the residual.





Beyond Wavelet

Portilla et al., Image Denoising Using Scale Mixtures of Gaussians in the Wavelet Domain, 2003.

Coefficients reduction through a model of the noise.

RBF and Wavelet have excellent for CUDA implementation as all bases with limited support.

Riassunto

- Supervised learning: predictive regression.
- Regressione multi-scala
- Versione on-line
- Valutazione del modello

How to classify the error introduced by a model?

Is the model good enough?

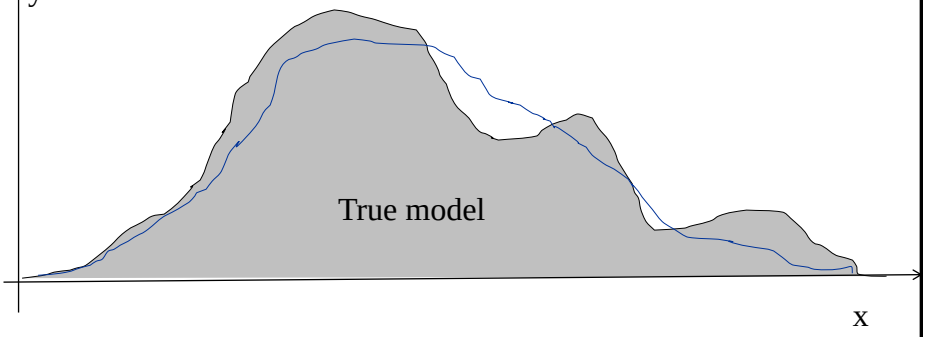
Does it have enough parameters?
Does it cover the input domain (in all dimensions)?

This is not enough to obtain a good model!!
The model should be properly tuned to the data

Source of errors:
- Under-parameterization
- Bias
- Variability

How to classify the error introduced by a model?

How is the estimated model related to the true model?

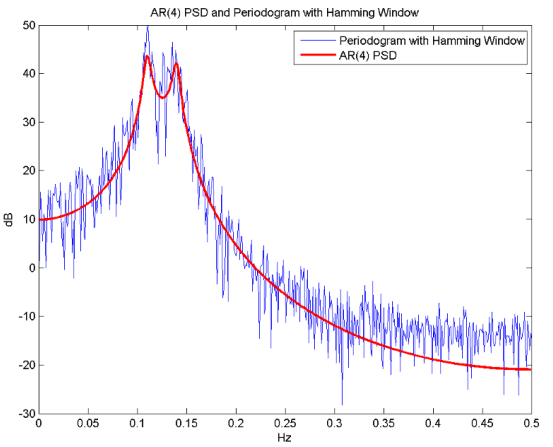


Bias and variability trade-off

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

Variability

How are the measured points related to the estimated model?



Given $P_{mes}(x_{mes},y_{mes})$ and $y = f(x)$, the error is measured as: $dist(y_{mes},f(x_{mes}))$, for instance Euclidean distance. It is associated to measurement error.

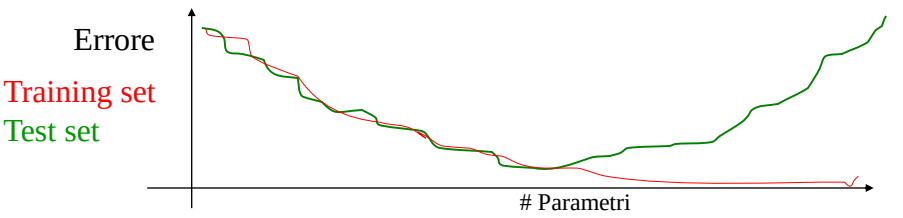
If variability goes to zero, bias increases and overfitting arises.
In a good model, variability tends to the statistics of the measurement noise.

Scelta empirica – cross-validation

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

Si vuole evitare che il modello si specializzi troppo sui pattern di training e non sia in grado di interpolare correttamente.

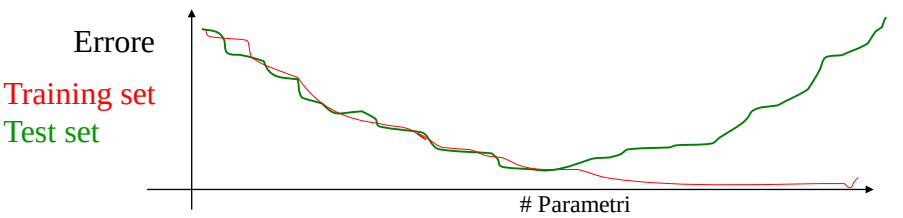
*Il numero di parametri viene aumentato fino a quando **entrambi** gli errori diminuiscono.*



Scelta teorica

Quale funzione costo minimizzo? Come posso inserire l'informazione di complessità nella funzione costo?

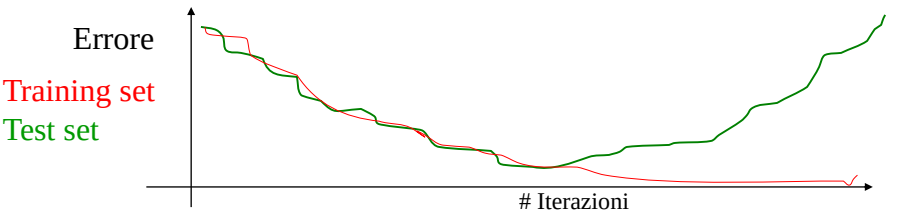
Penalizzo i modelli con tanti parametri. Regularization with Reproducible Hilbert Kernels as regularizers



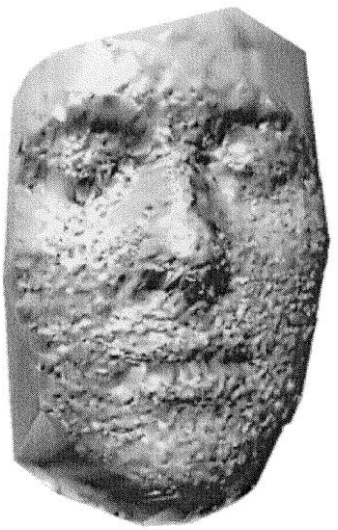
Altri approcci

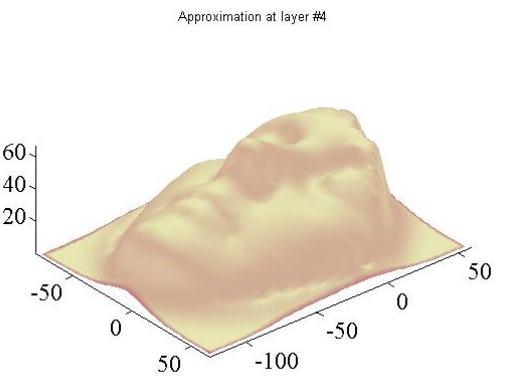
Semi-convergenza: non porto l'algoritmo fino alla convergenza nel punto di ottimo ma arresto le iterazioni prima.

Il modello non sarà perfettamente aderente ai dati, ma il residuo sarà tendenzialmente l'errore di misura.



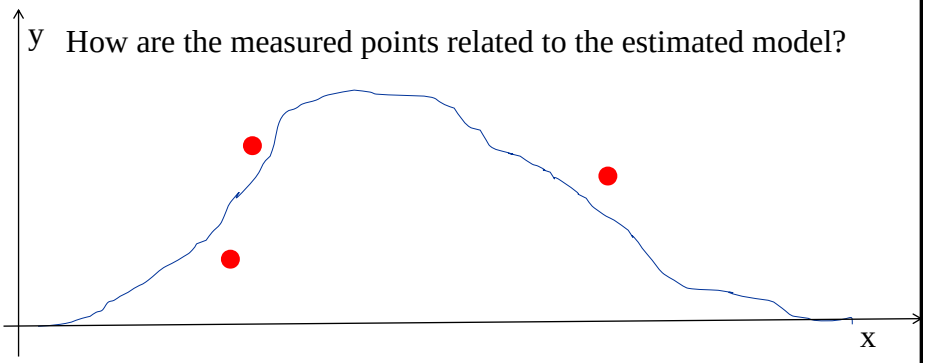
Problema dell'overfitting dovuto a sovraparametrizzazione





Quante unità?

Variability



How are the measured points related to the estimated model?

Given $P_{mes}(x_{mes},y_{mes})$ and $y = f(x)$, the error is measured as: $dist(y_{mes},f(x_{mes}))$, for instance Euclidean distance. It is associated to me

If variability goes to zero, bias increases and overfitting arises.
In a good model, variability tends to the statistics of the measurement

Riassunto

- Supervised learning: predictive regression.
- Regressione multi-scala
- Versione on-line
- Valutazione del modello