Machine learning

Kernel regression and gaussian processes

Corso di Laurea Magistrale in Informatica

Università di Roma Tor Vergata

Prof. Giorgio Gambosi

a.a. 2021-2022

Metodi non parametrici: non combinano con dei coefficienti le feature. per poi fare la predizione.
Qui, la predizione si fa senza coefficienti ma combinando i valori target degli elementi del training set.

Kernel regression

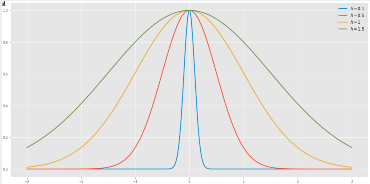
- In kernel regression methods, the target value corresponding to any item $\mathbf{x}$ is predicted by referring to items in the training set, and in particular to the items which are closer to $\mathbf{x}$.
- This is controlled by referring to a kernel function $\kappa_h(\mathbf{x})$, which is non zero only in an interval around 0
- h is the bandwidth of the kernel, which controls the width of $\kappa_h(\mathbf{x})$

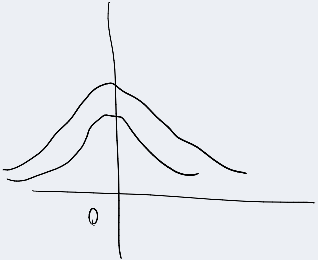
↳ intervallo ristretto.

A possible, common kernel, is the gaussian (or RBF) kernel

$$g(\mathbf{x}) = e^{-\frac{\|\mathbf{x}\|^2}{2h^2}}$$

più o meno stretto dipende dall' h.





$$p(t|\mathbf{x}) = \frac{p(t,\mathbf{x})}{p(\mathbf{x})} \rightarrow p(\mathbf{x},t)$$

In regression, we are interested in estimating the conditional expectation

$$f(\mathbf{x}) = E[t|\mathbf{x}] = \int t\,p(t|\mathbf{x})dt = \int t\frac{p(\mathbf{x},t)}{p(\mathbf{x})}dt = \frac{\int t\ p(\mathbf{x},t)dt}{p(\mathbf{x})} = \frac{\int t\ p(\mathbf{x},t)dt}{\int p(\mathbf{x},t)dt}$$

The joint distribution $p(\mathbf{x},t)$ is approximated by means of a kernel function as

$$p(\mathbf{x},t) \approx \frac{1}{n}\sum_{i=1}^{n} \kappa_h(\mathbf{x}-\mathbf{x}_i)\kappa_h(t-t_i)$$

misura la vicinanza del punto.

Non abbiamo fatto ipotesi di Gaussianità su $p(\mathbf{x}|t)$. La definiamo in modo diverso: l'argomento è la coppia (punto, target)

N.B.: non ci sono parametri.

This results into

$$f(\mathbf{x}) = \frac{\int t \frac{1}{n} \sum_{i=1}^{n} \kappa_t(\mathbf{x} - \mathbf{x}_i) \kappa_h(t - t_i) dt}{\int \frac{1}{n} \sum_{i=1}^{n} \kappa_h(\mathbf{x} - \mathbf{x}_i) \kappa_h(t - t_i) dt} = \frac{\sum_{i=1}^{n} \kappa_h(\mathbf{x} - \mathbf{x}_i) \int t \kappa_h(t - t_i) dy}{\sum_{i=1}^{n} \kappa_h(\mathbf{x} - \mathbf{x}_i) \int \kappa_h(t - t_i) dt}$$

If we assume that the kernel $\kappa(x)$ is a probability distribution with 0 mean, it results $\int \kappa_h(t - t_i) dt = 1$ and $\int t \kappa_h(t - t_i) dt = t_i$, we get

$$f(\mathbf{x}) = \frac{\sum_{i=1}^{n} \kappa_h(\mathbf{x} - \mathbf{x}_i) t_i}{\sum_{i=1}^{n} \kappa_h(\mathbf{x} - \mathbf{x}_i)}$$

Considera per ogni elemento il valore target pesato per questa funzione. Il peso è maggiore per vicinanza maggiore.

By setting

$$w_i(\mathbf{x}) = \frac{\kappa_h(\mathbf{x} - \mathbf{x}_i)}{\sum_{j=1}^{n} \kappa_h(\mathbf{x} - \mathbf{x}_j)}$$

we can write

$$f(\mathbf{x}) = \sum_{i=1}^{n} w_i(\mathbf{x}) t_i$$

that is, the predicted value is computed as a normalized linear combination of all target values, weighted by kernels (Nadaraya-Watson)

w_i dipende da x_i che da $\boldsymbol{x}$

Per la maggior parte dei punti $w \simeq 0$

Locally weighted regression

In Nadaraya-Watson model, the prediction is performed by means of a normalized weighted combination of constant values (target values in the training set).

Locally weighted regression (LOESS) improves that approach by referring to a weighted version of the sum of squared differences loss function used in regression.

If a value t has to be predicted for an item $\mathbf{x}$, a "local" version of the loss function is considered, with weight $\kappa_i(\mathbf{x})$.

$$L(\mathbf{x}) = \sum_{i=1}^{n} \kappa_i(\mathbf{x})(\mathbf{w}^T \overline{\mathbf{x}}_i - t_i)^2 = \sum_{i=1}^{n} \kappa_h(\mathbf{x} - \mathbf{x}_i)(\mathbf{w}^T \overline{\mathbf{x}}_i - t_i)^2$$

Weights $\kappa_i(\mathbf{x})$ are dependent from the "distance" between $\mathbf{x}$ and $\mathbf{x}_i$, as measured by the kernel function

$$\kappa_i(\mathbf{x}) = \kappa_h(\mathbf{x} - \mathbf{x}_i)$$

The minimization of this loss function

$$\hat{\mathbf{w}}(\mathbf{x}) = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{n} \kappa_i(\mathbf{x})(\mathbf{w}^T \overline{\mathbf{x}}_i - t_i)^2$$

has solution

$$\hat{\mathbf{w}}(\mathbf{x}) = (\overline{\mathbf{X}}^T \Psi(\mathbf{x}) \overline{\mathbf{X}})^{-1} \overline{\mathbf{X}}^T \Psi(\mathbf{x}) \mathbf{t}$$

where $\Psi(\mathbf{x})$ is a diagonal $n \times n$ matrix with $\Psi(\mathbf{x})_{ii} = \kappa_i(\mathbf{x})$.

The prediction is then performed as usual, as

$$y = \hat{\mathbf{w}}(\mathbf{x})^T \overline{\mathbf{x}}$$

l'approccio consiste nel definire una diversa f. di costo $\forall$ punto e minimizzare quella.

diversa in $\Psi(\mathbf{x})$

$\begin{pmatrix} k(x,x_1) & & 0 \\ & k(x,x_j) & \\ 0 & & \ddots \end{pmatrix}$

$\hat{w}(\mathbf{x})$ è il valore ottimo dei coefficienti.

La predizione è effettuata come sempre, ma con un altro punto devo ricalcolare tutto $\Rightarrow$ costa di più.

$\sum (w^T x_i - t_i)^2 \; w^* = (X^T X)^{-1} X^T t$

Local logistic regression

The same approach applied in the case of local regression can be applied for classification, by defining a weighted loss function to be minimized, with weights dependent from the item whose target must be predicted.

In this case, a weighted version of the cross entropy function is considered, which has to be maximized

$$L(\mathbf{x}) = \sum_{i=1}^{n} \kappa_h(\mathbf{x} - \mathbf{x}_i)(t_i \log p_i - (1 - t_i)\log(1 - p_i))$$

with $p_i = \sigma(\mathbf{w}^T \overline{\mathbf{x}}_i)$, as usual.

The loss function minimization can be performed, for example, by applying a suitable modification of the IRLS algorithm for logistic regression

Recap: some properties of Gaussian distribution

$\Sigma_{AA} = \begin{pmatrix} (x_1 - \mu)^2 & \cdots \\ \vdots & \ddots \end{pmatrix}$ Σ_{AB}: ha elem. di A come righe ed elem. di B come colonne: $\begin{pmatrix} \cdots & (x_{A,i} - x_{B,j})^2 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$

In order to introduce Gaussian processes and how they can be exploited for regression, let us first provide a short reminder on some properties of multivariate gaussian distributions.

Let $\mathbf{x} = (x_1, \dots, x_n)^T$ be a random vector with gaussian distribution $p(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ and let $\mathbf{x} = (\mathbf{x}_A, \mathbf{x}_B)$ be a partition of the components $\mathbf{x}$ such that:

- $\mathbf{x}_A = (x_1, \dots, x_r)^T$ ← Gaussiane
- $\mathbf{x}_B = (x_{r+1}, \dots, x_n)^T$

Una r-upla di un punto in uno spazio di dimensione n-r.

Then, the marginal densities $p(\mathbf{x}_A)$ and $p(\mathbf{x}_B)$ are both gaussian with means $\boldsymbol{\mu}_A, \boldsymbol{\mu}_B$ and covariance matrices Σ_A, Σ_B which can be derived from $\boldsymbol{\mu}, \Sigma$ by observing that

$$\boldsymbol{\mu} = (\boldsymbol{\mu}_A, \boldsymbol{\mu}_B)^T \qquad \Sigma = \begin{pmatrix} \Sigma_A & \Sigma_{AB} \\ \Sigma_{AB}^T & \Sigma_B \end{pmatrix}$$

È la distribuzione congiunta.

Recap: some properties of the Gaussian distribution

In the same situation, the conditional densities $p(\mathbf{x}_A|\mathbf{x}_B)$ and $p(\mathbf{x}_B|\mathbf{x}_A)$ are also gaussian with means

$$\boldsymbol{\mu}_{A|B} = \boldsymbol{\mu}_A + \Sigma_{AB}\Sigma_B^{-1}(\mathbf{x}_B - \boldsymbol{\mu}_B)$$
$$\boldsymbol{\mu}_{B|A} = \boldsymbol{\mu}_B + \Sigma_{BA}\Sigma_A^{-1}(\mathbf{x}_A - \boldsymbol{\mu}_A)$$

and covariance matrices

$$\Sigma_{A|B} = \Sigma_A - \Sigma_{AB}\Sigma_B^{-1}\Sigma_{BA}$$
$$\Sigma_{B|A} = \Sigma_B - \Sigma_{BA}\Sigma_A^{-1}\Sigma_{AB}$$

Gaussian processes

Facciamo tendere la dimensione a $+\infty$

- Multivariate gaussians on random vectors are useful for modeling finite collections of real-valued variables. They have nice analytical properties (see previous slides).
- Gaussian processes: extension of multivariate gaussians to infinite-sized collections of real-valued variables.
- We may think of gaussian processes as distributions not just over random vectors but over random real functions.

Probability distributions over functions with finite domains

Diverse funzioni che assegnano ai • i • • etc....

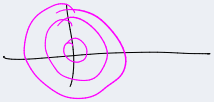
Let us first consider the case of functions defined over finite vectors.

- Let $\chi = (\mathbf{x}_1, \dots, \mathbf{x}_m)$ be any finite vector, and let $\mathscr{H}$ be the set of functions $f : \chi \mapsto \mathbb{R}$: f assigns a value $f(\mathbf{x}_i)$ to each $\mathbf{x}_i \in \chi$
  - A function $f \in \mathscr{H}$ can be described by the vector $(f(\mathbf{x}_1), \dots, f(\mathbf{x}_m))$
  - Any vector $(y_1, \dots, y_m)$ can be seen as the description of a function $f \in \mathscr{H}$ such that $f(\mathbf{x}_i) = y_i$
  - The set $\mathscr{H}$ is then in 1-to-1 correspondence with the set of vectors in $\mathbf{R}^m$
- A probability distribution $p(\mathbf{x}), \mathbf{x} \in \mathbf{R}^m$ over m-dimensional real vectors is also a distribution $p(f), f \in \mathscr{H}$ over functions from $\mathbf{R}^m$ to $\mathbb{R}$

Ogni funzione f ha una definizione finita (è definita su un insieme finito, ne elenco i valori).
$\Rightarrow$ Sono rappresentabili con vettori.

Gaussian distributions over functions with finite domains

Da una distribuzione così "pesco" (sampling) funzioni di un certo tipo (vicine al centro) cosa più facilitata.



Assume that $p(\mathbf{x})$ (or, equivalently, $p(f)$) is a (multivariate, m-dimensional) Gaussian distribution centered on $\mathbf{0}$ and with diagonal covariance $\sigma^2\mathbf{I}$, that is

$$p(f|\sigma^2) = \mathcal{N}(f|\mathbf{0}, \sigma^2\mathbf{I}) = \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{f(\mathbf{x}_i)^2}{2\sigma^2}}$$

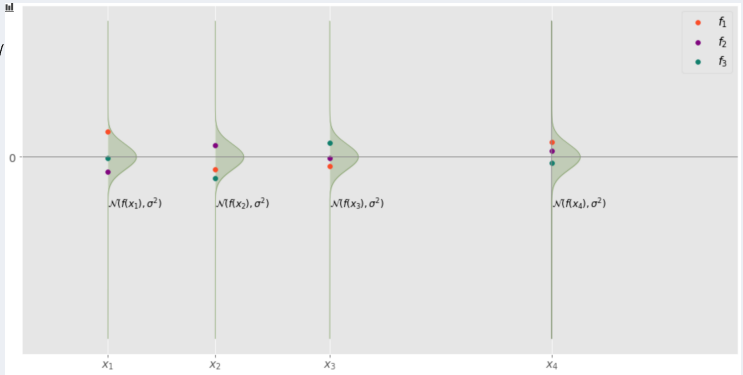
prodotto di m univariate centrate su 0.

- This is equivalent to assuming that each function value $f(\mathbf{x}_i)$ has normal distribution with mean 0 and variance σ^2, and that items are independent
- A dependence between function values at different points could be modeled through a non-diagonal covariance matrix

Gaussian distributions over functions with finite domains

Essendo la matrice di cov. diagonale, posso pensare che siano indipendenti.

Ma posso pensare ad un livello di cov. che varia in base alla vicinanza dei punti (Kernel reg.)



Ogni funzione è estratta a caso secondo il valore che assume sui punti in base alla distribuzione Gaussiana.

We may consider $p(f|\sigma^2)$ as a prior distribution of functions, with respect to the observation of the value t_j actually taken by any variable $\mathbf{x}_j$, $1 \leq j \leq m$.

- Assume now that for some subset $\mathbf{X} = \{\mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_k}\}$ of component indices, the corresponding targets $\mathbf{t} = \{t_{i_1}, \dots, t_{i_k}\}$ are available.
- the posterior distribution $p(f|\mathbf{X}, \mathbf{t})$ of functions (wrt to $\mathbf{X}, \mathbf{t}$) can be defined and derived according to Bayes' rule, provided a likelihood model is defined

$$p(\mathbf{X}, \mathbf{t}|f) = \prod_{\mathbf{x}_i \in \mathbf{X}} p(\mathbf{x}_i, t_i|f) = \prod_{\mathbf{x}_i \in \mathbf{X}} p(t_i|\mathbf{x}_i, f) p(\mathbf{x}_i|f) \propto \prod_{\mathbf{x}_i \in \mathbf{X}} p(t_i|\mathbf{x}_i, f)$$

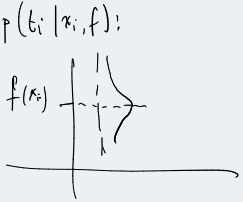
$p(x_i)$ è uniforme quindi non la consideriamo.

- for example, we could assume the usual likelihood $p(t|\mathbf{x}, f, \beta) = \mathcal{N}(t|f(\mathbf{x}), \beta)$, which implies

$$p(\mathbf{X}, \mathbf{t}|f, \beta) \propto \prod_{\mathbf{x}_i \in \mathbf{X}} \mathcal{N}(t_i|f(\mathbf{x}_i), \beta)$$

- the posterior distribution then would be

$$p(f|\mathbf{X}, \mathbf{t}, \beta, \sigma^2) \propto \prod_{\mathbf{x}_i \in \mathbf{X}} \mathcal{N}(t_i|f(\mathbf{x}_i), \beta) p(f|\sigma^2)$$



Since both the prior and the posterior distributions of f are gaussian, the predictive distribution

$$p(t|\mathbf{x}, \mathbf{X}, \mathbf{t}, \beta, \sigma^2) = \int p(t|\mathbf{x}, f, \beta) p(f|\mathbf{X}, \mathbf{t}, \beta, \sigma^2) df$$

is itself a gaussian.

precisione

passo attributo alla precisione

That would the case also in the more general case when some dependancy between function points is assumed. In this case, a general covariance matrix is defined for the prior distribution

$$p(f|\Sigma) = \mathcal{N}(f|\mathbf{0}, \Sigma)$$

Cosa succede se assumiamo che i valori assunti siano dipendenti fra loro? Vogliamo metterci in una situazione dove dal target vogliamo prevedere un valore non noto.
Media 0, ma c'è un effetto di covarianza

Gaussian distributions over functions with infinite domains

Abbiamo fissato m punti discreti, poniamo che m cresca molto: copriamo la retta $\mathbb{R}$ non molto sempre più fitto. Al limite $\to$ copro tutto $\mathbb{R}$, quindi ho funzioni sui reali. È una proiezione di quanto detto a ∞.

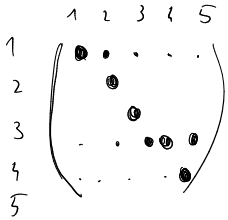
- In the case of infinite χ, we have to deal with an infinite collection of random variables.
- In this case, the role of multidimensional distributions is covered by stochastic processes.
  - A *stochastic process* is a collection of random variables, $\{f(\mathbf{x}) : \mathbf{x} \in \chi\}$, indexed by elements from some set χ, known as the index set.
- A Gaussian process is a stochastic process such that for any finite subset $\mathbf{x}_1, \ldots, \mathbf{x}_n$ of χ, the function values $f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)$ have joint multivariate Gaussian distribution

La distribuzione è Gaussiana, quindi preso un insieme finito $\in \mathbb{R}$ la restrizione all'insieme è Gaussiana.

- In order to define a Gaussian process, both a mean and a covariance function must be defined.
  - a mean function $m : \mathbb{R}^d \mapsto \mathbb{R}$ mapping each point $\mathbf{x}_i \in \chi$ to the expectation
    $$m(\mathbf{x}_i) = E_f[f(\mathbf{x}_i)]$$
    of $f(x)$ over all functions f
  - a covariance function $\kappa : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ mapping each pair of variables $(\mathbf{x}_i, \mathbf{x}_j) \in \chi^2$ to the covariance
    $$\kappa(\mathbf{x}_i, \mathbf{x}_j) = E_f[(f(\mathbf{x}_i) - m(\mathbf{x}_i))^2 (f(\mathbf{x}_j) - m(\mathbf{x}_j))^2]$$
    ⟶ da la dipendenza fra x_i ed x_j
    of $f(x_j)$ and $f(x_j)$ over all functions f.

Considereremo:
- media 0
- matrice di covarianza dove la cov. fra due punti è determinata da una funzione di distanza $\kappa(x_i, x_j)$

Il valore di cov:



Si abbassa man mano che mi allontano

(è la rappresentazione nel notebook).

L'idea è la stessa di prima: passavamo da una distribuzione a priori ad una a posteriori che ci permetteva di fare delle previsioni nei punti dell'insieme di cui non conoscevamo il target.
Qui avviene la stessa cosa.

- The covariance function κ is assumed to be a positive definite (Mercer) kernel.
- This means that for any set of distinct points $\mathbf{x}_1, \dots, \mathbf{x}_n$ it must be

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j \kappa(\mathbf{x}_i, \mathbf{x}_j) > 0$$

  for any choice of the constants $c_1, \dots, c_n$ such that not all c_i are equal to 0.

No ?

- Equivalently, the square Gram matrix G defined as

$$G = \begin{pmatrix} \kappa(\mathbf{x}_1, \mathbf{x}_1) & \kappa(\mathbf{x}_1, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_1, \mathbf{x}_n) \\ \kappa(\mathbf{x}_2, \mathbf{x}_1) & \kappa(\mathbf{x}_2, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_2, \mathbf{x}_n) \\ \cdots & \cdots & \cdots & \cdots \\ \kappa(\mathbf{x}_n, \mathbf{x}_1) & \kappa(\mathbf{x}_n, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_n, \mathbf{x}_n) \end{pmatrix}$$

  must have positive eigenvalues.
- A collection of positive definite kernels is known in the literature and can be constructed by applying suitable rules.

Given a gaussian process $p(f) = \mathcal{GP}(m, \kappa)$, then for any set of items $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$, the distribution of $f(\mathbf{x}_1), \dots, f(\mathbf{x}_n)$ is a gaussian

$$(f(\mathbf{x}_1), \dots, f(\mathbf{x}_n)) \sim \mathcal{N}(\boldsymbol{\mu}(\mathbf{X}) | \Sigma(\mathbf{X}))$$

where

- $\boldsymbol{\mu}(\mathbf{X}) = (m(\mathbf{x}_1), \dots, m(\mathbf{x}_n))^T$
- $\Sigma(\mathbf{X})$ is the Gram matrix wrt $\mathbf{x}_1, \dots, \mathbf{x}_n$ of a kernel function $\kappa(\mathbf{x}, \mathbf{x}')$

→ mean in practice is 0

The mean vector - at least initially, with no information from data - is usually assumed to be $\mathbf{0}$: different processes are then characterized only by their covariance kernel κ.

Sampling functions from gaussian processes

- For any finite subset $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ of χ it is possible to sample from $p(f)$ the values of $f(\mathbf{x}_1), \dots, f(\mathbf{x}_m)$ by sampling from $\mathcal{N}(f|\mathbf{0}, \Sigma(\mathbf{X}))$, where, as stated before

$$\boldsymbol{\mu}(\mathbf{X})_i = m(\mathbf{x}_i)$$
$$\Sigma(\mathbf{X})_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$$

RBF kernel

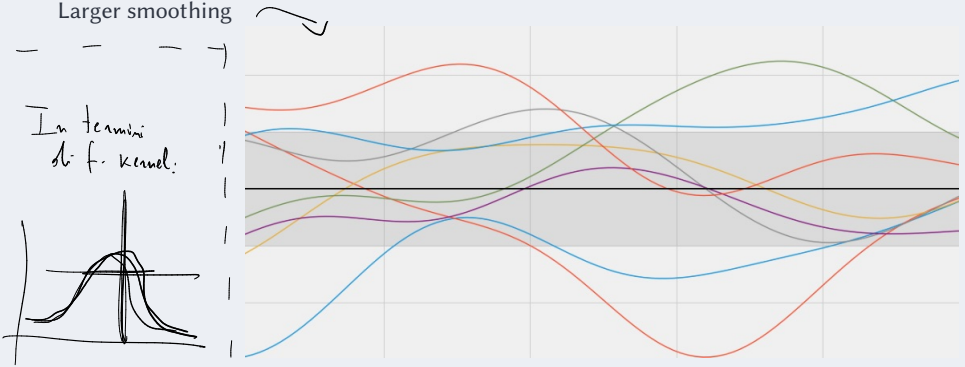
- One of the most applied kernel is the RBF kernel

$$\kappa(\mathbf{x}_1, \mathbf{x}_2) = \sigma^2 e^{-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{2\tau^2}}$$

  which tends to assign higher covariance between $f(\mathbf{x}_1)$ and $f(\mathbf{x}_2)$ if $\mathbf{x}_1$ and $\mathbf{x}_2$ are nearby points.
- Functions drawn from a Gaussian process with RBF kernel tend to be smooth (values computed for nearby points tend to have similar values). Smoothing is larger for larger τ.

Il valore assunto dalla funzione in un punto dipende dai valori vicini → ho funzioni smooth

Larger smoothing

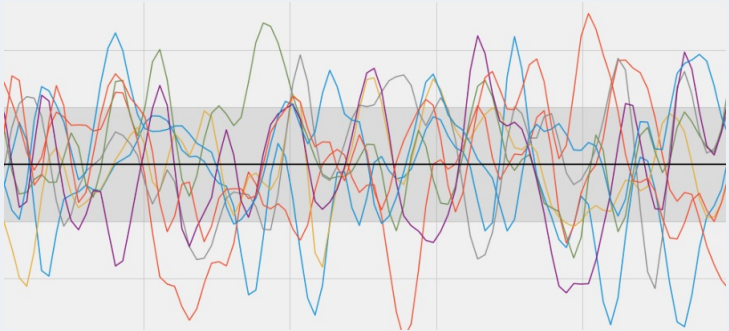


In termini di f. kernel:

più è largo e più il valore assunto in un punto è determinato da quelli intorno. Il caso indipendente è l'impulso centrale.

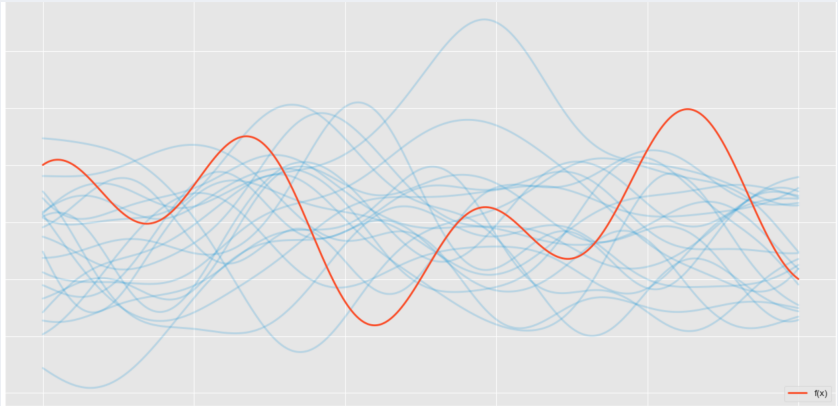
intorno più largo, meno dx numero.

Smaller smoothing



Gaussian process regression: no noise

- By the gaussian process definition, $\mathbf{f}$ is distributed as a multivariate gaussian such that the mean of any value $\mathbf{x}$ is $m(\mathbf{x})$ and the covariance of any pair $\mathbf{x}, \mathbf{x}'$ is $\kappa(\mathbf{x}, \mathbf{x}')$
- as a consequence, for any finite set of points $\mathbf{X}$, we have that $\mathbf{f}(\mathbf{X})$ is distributed as a multivariate gaussian with mean $\boldsymbol{\mu}(\mathbf{X})$ defined as $\boldsymbol{\mu}(\mathbf{X})_i = m(\mathbf{x}_i)$ and covariance matrix $\Sigma(\mathbf{X})$, defined as $\Sigma(\mathbf{X})_{i,j} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$



Abbiamo quindi il prior, conosciamo il valore che la funzione deve assumere nel punto per un certo insieme dei punti.

Ciò che conta qui è la matrice di covarianza: la regressione è prevedere il valore assunto in un punto sapendo il valore che veniva assunto nei punti precedenti e questo è dato dalla matrice di covarianza.

Gaussian process regression: no noise

- Let us now assume that for a set of points $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^T$ the corresponding values $\mathbf{t} = (t_1, \dots, t_n)^T$ are known
- that is, we assume that a training set $\mathbf{X}, \mathbf{t}$ is available such that the target values in the training set correspond exactly to the function value $t_i = f(\mathbf{x}_i)$. Note that in the probabilistic model of regression this is not true, since a (gaussian) error is assumed

$X_1, \dots, X_n \sim f(X)$.

test set.

In general, for any new set of points $\mathbf{X}^* = (\mathbf{x}_1^*, \dots, \mathbf{x}_m^*)^T$, the joint distribution of $(\mathbf{f}(\mathbf{X}), \mathbf{f}(\mathbf{X}^*))$ is a multivariate gaussian distribution with mean $\boldsymbol{\mu}(\mathbf{X}, \mathbf{X}^*)$ and covariance $\Sigma(\mathbf{X}, \mathbf{X}^*)$

- $\boldsymbol{\mu}(\mathbf{X}, \mathbf{X}^*) = (\boldsymbol{\mu}(\mathbf{X}), \boldsymbol{\mu}(\mathbf{X}^*))^T$
- $\Sigma(\mathbf{X}, \mathbf{X}^*) = \begin{pmatrix} \Sigma(\mathbf{X}) & \Sigma(\mathbf{X}^*, \mathbf{X}) \\ \Sigma(\mathbf{X}^*, \mathbf{X})^T & \Sigma(\mathbf{X}^*) \end{pmatrix}$ ← è nota

Ci chiediamo come sia fatto X^*

$(f(X), f(X^*))$ è multivariata poiché stiamo considerando processi gaussiani.

where $\Sigma(\mathbf{X}^*, \mathbf{X}) = \begin{pmatrix} \kappa(\mathbf{x}_1^*, \mathbf{x}_1) & \kappa(\mathbf{x}_1^*, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_1^*, \mathbf{x}_n) \\ \kappa(\mathbf{x}_2^*, \mathbf{x}_1) & \kappa(\mathbf{x}_2^*, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_2^*, \mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ \kappa(\mathbf{x}_m^*, \mathbf{x}_1) & \kappa(\mathbf{x}_m^*, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_m^*, \mathbf{x}_n) \end{pmatrix}$

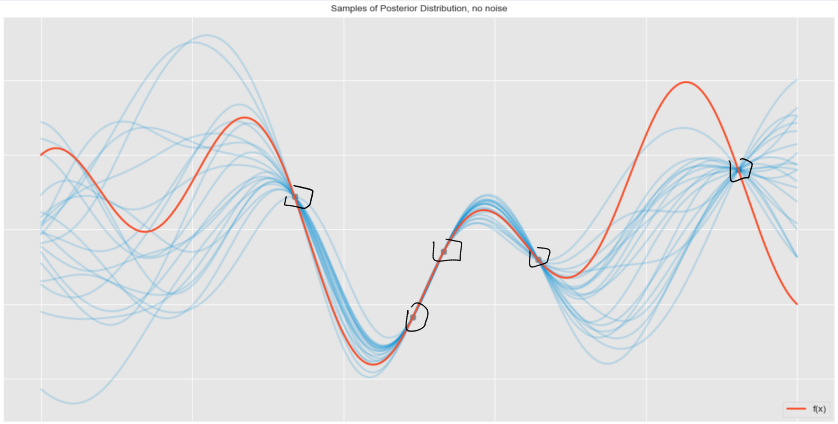
matrice di matrici covarianza: ho tutte le coppie.

The posterior distribution of $\mathbf{y} = \mathbf{f}(\mathbf{X}^*)$, given $\mathbf{X}, \mathbf{t}$ can be derived by gaussian distribution properties, and turns out to be a m-dimensional gaussian distribution with mean and covariance defined as

- $\boldsymbol{\mu}_p^* = \mu(\mathbf{y}|\mathbf{X}, \mathbf{t}) = \boldsymbol{\mu}(\mathbf{X}^*) + \Sigma(\mathbf{x}, \mathbf{X})\Sigma(\mathbf{X})^{-1}(\mathbf{t} - \boldsymbol{\mu}(\mathbf{X}))$
- $\Sigma_p^* = \Sigma(\mathbf{X}^*) - \Sigma(\mathbf{x}, \mathbf{X})\Sigma(\mathbf{X})^{-1}\Sigma(\mathbf{x}, \mathbf{X})^T$

Manca la distribuzione dei valori assunti dalla funzione sul test set: è ancora una Gaussiana con media e matrice di cov. calcolate. Ho di nuovo distribuzione di funzioni.

Gaussian process regression: no noise



□ : f(x) è l'ignoto.

Le funzioni generate passano per forza per i punti noti.

Ho una distribuzione di funzioni.

—— : funzione da cui sono stati generati i punti del train set

Gaussian process regression: no noise

In particular, for a single test point $\mathbf{x}$, the joint distribution of $(\mathbf{t}, \mathbf{f}(\mathbf{x}))$ is a multivariate gaussian distribution with mean $\boldsymbol{\mu}(\mathbf{X}, \mathbf{x})$ and covariance $\Sigma(\mathbf{X}, \mathbf{x})$

- $\boldsymbol{\mu}(\mathbf{X}, \mathbf{x}) = (\boldsymbol{\mu}(\mathbf{X}), \mu(\mathbf{x}))^T$
- $\Sigma(\mathbf{X}, \mathbf{x}) = \begin{pmatrix} \Sigma(\mathbf{X}) & \Sigma(\mathbf{x}, \mathbf{X}) \\ \Sigma(\mathbf{x}, \mathbf{X})^T & \Sigma(\mathbf{x}, \mathbf{x}) \end{pmatrix}$

where $\Sigma(\mathbf{x}, \mathbf{X}) = (\kappa(\mathbf{x}, \mathbf{x}_1), \kappa(\mathbf{x}, \mathbf{x}_2), \dots, \kappa(\mathbf{x}_n, \mathbf{x}_n))^T$ and $\Sigma(\mathbf{x}, \mathbf{x}) = \kappa(\mathbf{x}, \mathbf{x})$

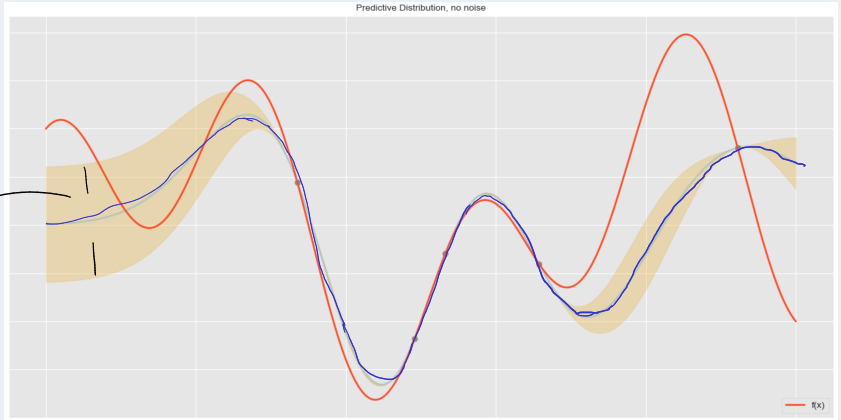
Inoltre ora gli elementi di cui non conosciamo il target nella distribuzione stessa.

Deriva da: (1

As a consequence, the predictive distribution of $y = f(\mathbf{x})$ is

$$m_p(y|\mathbf{X}, \mathbf{f}) = m(\mathbf{x}) + \Sigma(\mathbf{x}, \mathbf{X})\Sigma(\mathbf{X})^{-1}(\mathbf{t} - \boldsymbol{\mu}(\mathbf{X}))$$
$$\sigma^2 = \Sigma_p(\mathbf{x}, \mathbf{x}) = \kappa(\mathbf{x}, \mathbf{x}) - \Sigma(\mathbf{x}, \mathbf{X})\Sigma(\mathbf{X})^{-1}\Sigma(\mathbf{x}, \mathbf{X})^T$$

— : distribuzione predittiva



contorni dei punti noti, aumenta la fascia di incertezza

Il modo in cui cambia l'incertezza dipende dalla funzione kernel: più la varianza è stretta, più salirà il livello di incertezza.

Gaussian process regression: no noise

In this case, an interpolation of the given values has been performed: $f(\mathbf{x}_i) = t_i$ for all possible functions, sampled from $f(\mathbf{x}|\mathbf{X}, \mathbf{f})$.

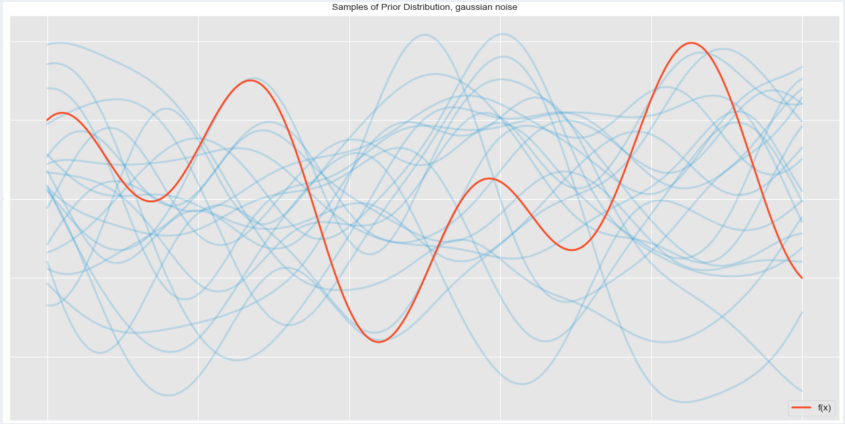
It results, in fact, for all $\mathbf{x}_i \in \mathbf{X}$,

$$m(\mathbf{x}_i|\mathbf{X}, \mathbf{f}) = t_i$$
$$\sigma^2 = 0$$

NO?

Gaussian process regression: gaussian noise

- $\mathbf{f}$ is now distributed as a multivariate gaussian with known mean $\boldsymbol{\mu}(\mathbf{X}) = (m(\mathbf{x}_i), \dots, m(\mathbf{x}_n))^T$ and covariance matrix $\hat{\Sigma}(\mathbf{X}) = \Sigma(\mathbf{X}) + \sigma_f^2\mathbf{I}$, defined by κ and σ_f^2



- Let us now assume that a training set $\mathbf{X}, \mathbf{t}$ is available such that the target values in the training set correspond approximately to the function value $t_i = f(\mathbf{x}_i) + \varepsilon$.
- In this case, for any new set of points $\mathbf{X}^*$, the joint distribution of $(\mathbf{t}, \mathbf{f}(\mathbf{X}^*))$ is a multivariate gaussian distribution with mean $\boldsymbol{\mu}(\mathbf{X}, \mathbf{X}^*)$ and covariance $\Sigma(\mathbf{X}, \mathbf{X}^*)$
  - $\boldsymbol{\mu}(\mathbf{X}, \mathbf{X}^*) = (\boldsymbol{\mu}(\mathbf{X}), \boldsymbol{\mu}(\mathbf{X}^*))^T$
  - $\Sigma(\mathbf{X}, \mathbf{X}^*) = \begin{pmatrix} \hat{\Sigma}(\mathbf{X}) & \Sigma(\mathbf{X}^*, \mathbf{X}) \\ \Sigma(\mathbf{X}^*, \mathbf{X})^T & \Sigma(\mathbf{X}^*) \end{pmatrix}$

Componente di incertezza più un dal rumore.

where in particular $\hat{\Sigma}(\mathbf{X}) = \begin{pmatrix} \kappa(\mathbf{x}_1, \mathbf{x}_1) + \sigma_f^2 & \kappa(\mathbf{x}_1, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_1, \mathbf{x}_n) \\ \kappa(\mathbf{x}_2, \mathbf{x}_1) & \kappa(\mathbf{x}_2, \mathbf{x}_2) + \sigma_f^2 & \cdots & \kappa(\mathbf{x}_2, \mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ \kappa(\mathbf{x}_n, \mathbf{x}_1) & \kappa(\mathbf{x}_n, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_n, \mathbf{x}_n) + \sigma_f^2 \end{pmatrix}$

Gaussian process regression: gaussian noise

The posterior distribution of $\mathbf{y} = \mathbf{f}(\mathbf{X}^*)$, given $\mathbf{X}, \mathbf{X}^*, \mathbf{t}$ can be again derived by the gaussian distribution properties, and turns out again to be a gaussian distribution with mean and covariance defined as

- $\boldsymbol{\mu}_p^* = \boldsymbol{\mu}(\mathbf{X}^*) + \Sigma(\mathbf{x}, \mathbf{X})\hat{\Sigma}(\mathbf{X})^{-1}(\mathbf{t} - \boldsymbol{\mu}(\mathbf{X}))$
- $\Sigma^* = \Sigma(\mathbf{X}^*) - \Sigma(\mathbf{x}, \mathbf{X})\hat{\Sigma}(\mathbf{X})^{-1}\Sigma(\mathbf{x}, \mathbf{X})^T$

funzioni estratte dalla distribuzione a posteriori: i valori di training non sono = target, c'è rumore. Le funzioni passano + vicine al valore target.