MACHINE LEARNING

CHAPTER 4: KERNEL METHODS

KERNEL METHODS

1. Kernel function and dual representation
2. Construction kernel
3. Gaussian process regression

Kernel function

For a nonlinear feature space mapping $\phi(\mathbf{x})$, the *kernel function* is given by

$$k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^T \boldsymbol{\phi}(\mathbf{x}')$$

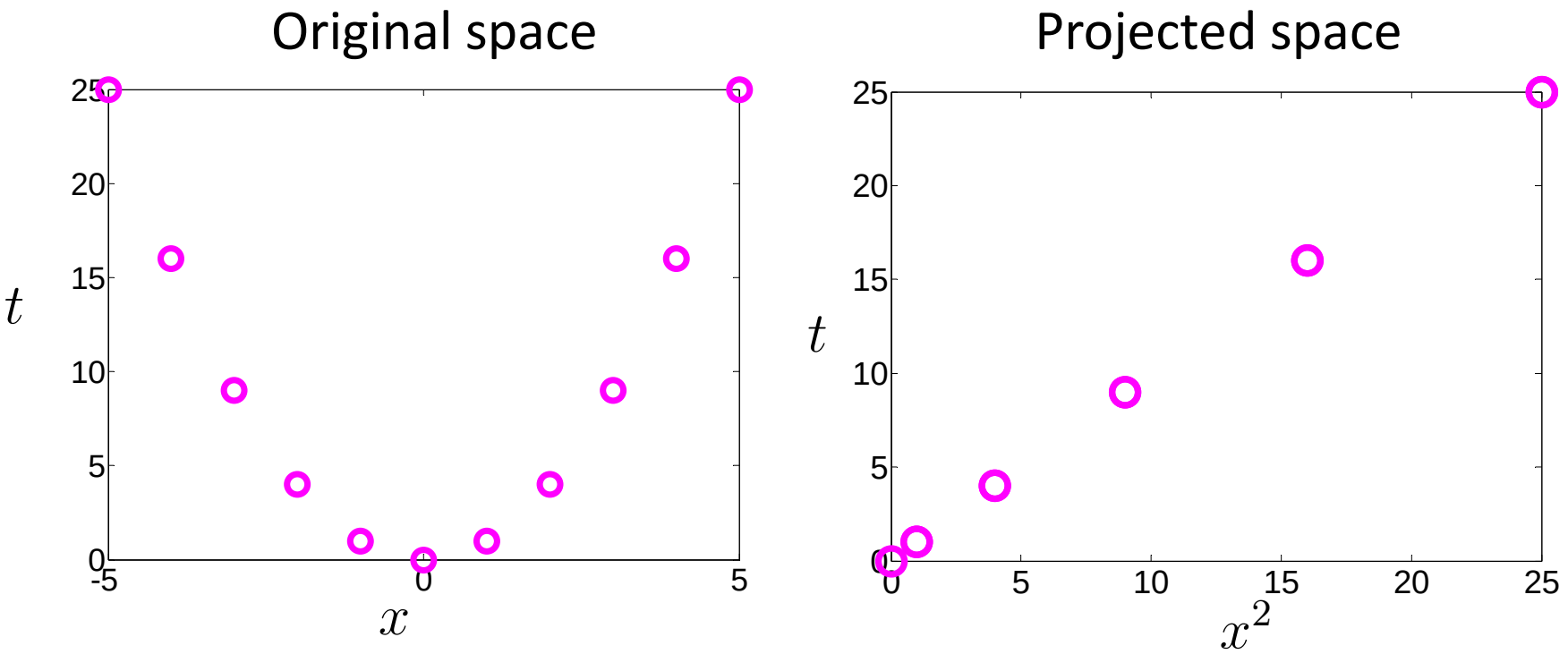
From this definition, we see that the kernel is a symmetric function.

Dual representation

A linear regression model whose parameters are determined by minimizing a regularized sum-of-squares error function given by

$$J(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

Why do we use kernel trick



Basis function: $\phi(x) = x^2$,

linear regression model: $y = \phi(x)$.

Why do we use kernel trick

If the projected feature has infinite dimensionality, it is impossible to learn the regression parameters **w**.

But the kernel function under this situation has an simple expression, such as Gaussian kennel

$$k\left(\mathbf{x}, \mathbf{x}'\right) = exp\left(-\|\mathbf{x} - \mathbf{x}'\|^2 / 2\sigma^2\right)$$

Dual problem is much more easy.

Dual representation

We suppose

$$\mathbf{w} = \sum_{n=1}^{N} a_n \boldsymbol{\phi}(\mathbf{x}_n) = \boldsymbol{\Phi}^T \mathbf{a}$$

where

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}.$$

Dual representation

Substitute $\mathbf{w} = \mathbf{\Phi}^T \mathbf{a}$ into $J(\mathbf{w})$, we obtain

$$J(\mathbf{a}) = \frac{1}{2}\mathbf{a}^{\mathrm{T}}\mathbf{\Phi}\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\mathbf{\Phi}^{\mathrm{T}}\mathbf{a} - \mathbf{a}^{\mathrm{T}}\mathbf{\Phi}\mathbf{\Phi}^{\mathrm{T}}\mathbf{t} + \frac{1}{2}\mathbf{t}^{\mathrm{T}}\mathbf{t} + \frac{\lambda}{2}\mathbf{a}^{\mathrm{T}}\mathbf{\Phi}\mathbf{\Phi}^{\mathrm{T}}\mathbf{a}$$

We define the Gram matrix $\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^T$, which is an $N \times N$ symmetric matrix with elements

$$K_{nm} = \boldsymbol{\phi}(\mathbf{x}_n)^T \boldsymbol{\phi}(\mathbf{x}_m) = k(\mathbf{x}_n, \mathbf{x}_m)$$

Dual representation

In terms of Gram matrix, the sum-of-squares error function can be rewritten as

$$J(\mathbf{a}) = \frac{1}{2}\mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{K}\mathbf{a} - \mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{t} + \frac{1}{2}\mathbf{t}^{\mathrm{T}}\mathbf{t} + \frac{\lambda}{2}\mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{a}.$$

Setting the gradient of J(**a**) with respect to zero, we obtain the following solution

$$\mathbf{a} = (\mathbf{K} + \lambda\mathbf{I}_N)^{-1}\,\mathbf{t}.$$

Dual representation

If we substitute this solution back to the linear regression model, the prediction function is

$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}) = \mathbf{a}^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{\phi}(\mathbf{x})$$

$$= \mathbf{k}(\mathbf{x})^{\mathrm{T}}\left(\mathbf{K} + \lambda\mathbf{I}_N\right)^{-1}\mathbf{t}$$

Where **k(x)** is an N dimensional vector with elements

$$k_n\left(\mathbf{x}\right) = k\left(\mathbf{x}_n, \mathbf{x}\right)$$

KERNEL METHODS

1. Kernel function and dual representation
2. Construction kernels
3. Gaussian process regression

Constructing kernels

a function $k(\mathbf{x}, \mathbf{x}')$ is a valid kernel

if the Gram matrix **K**, whose elements are given by $k(\mathbf{x}_n, \mathbf{x}_m)$, should be positive semidefinite for all possible choices of the set $\{\mathbf{x}_n\}$

Polynomial kernel $k(\mathbf{x}, \mathbf{x}') = \left(\mathbf{x}^T \mathbf{x}' + c\right)^2$

Gaussian kernel $k(\mathbf{x}, \mathbf{x}') = exp\left(-\|\mathbf{x} - \mathbf{x}'\|^2 / 2\sigma^2\right)$

Techniques for constructing new kernels

$$
\begin{aligned}
k(\mathbf{x}, \mathbf{x}') &= c k_1(\mathbf{x}, \mathbf{x}') \\
k(\mathbf{x}, \mathbf{x}') &= f(\mathbf{x}) k_1(\mathbf{x}, \mathbf{x}') f(\mathbf{x}') \\
k(\mathbf{x}, \mathbf{x}') &= q\left(k_1(\mathbf{x}, \mathbf{x}')\right) \\
k(\mathbf{x}, \mathbf{x}') &= \exp\left(k_1(\mathbf{x}, \mathbf{x}')\right) \\
k(\mathbf{x}, \mathbf{x}') &= k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}') \\
k(\mathbf{x}, \mathbf{x}') &= k_1(\mathbf{x}, \mathbf{x}') k_2(\mathbf{x}, \mathbf{x}') \\
k(\mathbf{x}, \mathbf{x}') &= k_3\left(\boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}')\right) \\
k(\mathbf{x}, \mathbf{x}') &= \mathbf{x}^{\mathrm{T}} \mathbf{A} \mathbf{x}' \\
k(\mathbf{x}, \mathbf{x}') &= k_a(\mathbf{x}_a, \mathbf{x}'_a) + k_b(\mathbf{x}_b, \mathbf{x}'_b) \\
k(\mathbf{x}, \mathbf{x}') &= k_a(\mathbf{x}_a, \mathbf{x}'_a) k_b(\mathbf{x}_b, \mathbf{x}'_b)
\end{aligned}
$$

KERNEL METHODS

1. Kernel function and dual representation
2. Construction kernels
3. Gaussian process regression

Gaussian process regression

A general regression model

$$t_n = y_n + \epsilon_n$$

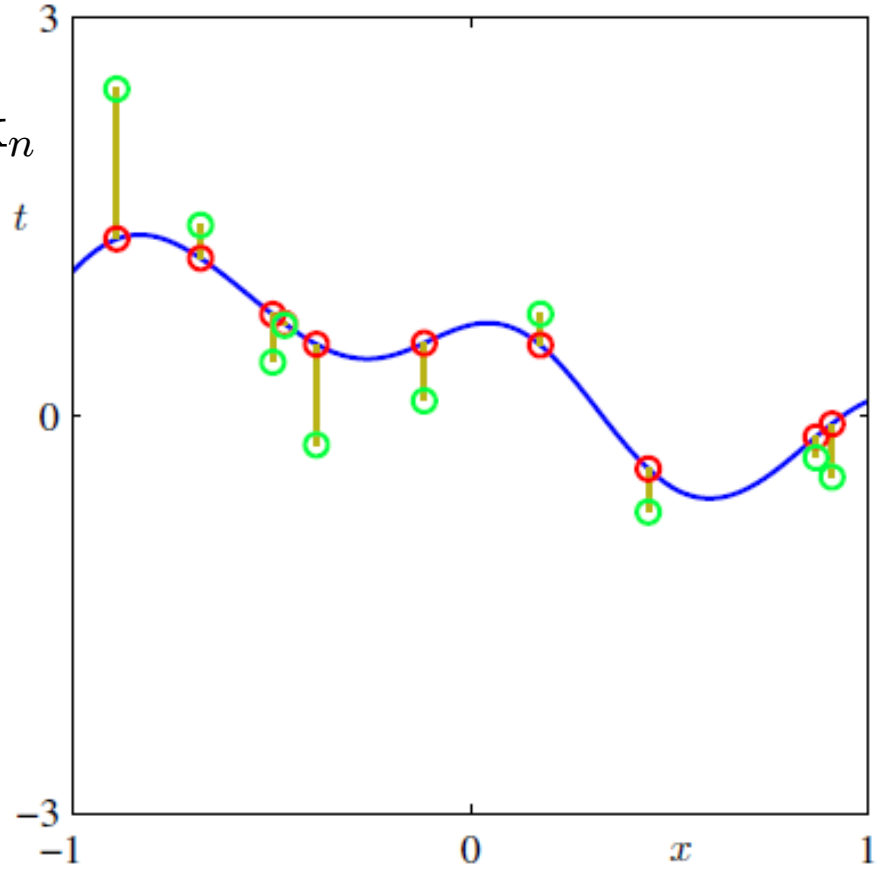
where $y_n = y(\mathbf{x}_n)$ is the output of regression model, and ϵ_n is a random noise variable. We suppose ϵ_n has a Gaussian distribution, so that

$$p(t_n|y_n) = \mathcal{N}\left(t_n|y_n, \beta^{-1}\right)$$

Gaussian process regression

y_n is the output of the prediction function $y(\mathbf{x})$ at $\mathbf{x}_n$

$$t_n = y_n + \epsilon_n$$



Gaussian process regression

The joint distribution of target values $\mathbf{t} = (t_1, ..., t_N)^T$ conditioned on the values of **y** is

$$p(\mathbf{t}|\mathbf{y}) = \mathcal{N}\left(\mathbf{t}|\mathbf{y}, \beta^{-1}\mathbf{I}_N\right)$$

Gaussian process regression

Gaussian process regression suppose a Gaussian process prior on the outputs of regression model $\mathbf{y} = [y(\mathbf{x}_1), ..., y(\mathbf{x}_N)]^T$

$$p(\mathbf{y}|\mathbf{X}) = \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K})$$

The kernel function that determines **K** is typically chosen to express the property that, for points $\mathbf{x}_n$ and $\mathbf{x}_m$ that are similar, the corresponding values $y(\mathbf{x}_n)$ and $y(\mathbf{x}_m)$ will be strongly correlated than for dissimilar points.

Gaussian process regression

$$cov\left(y\left(\mathbf{x}_n\right), y\left(\mathbf{x}_m\right)\right) = k\left(\mathbf{x}_n, \mathbf{x}_m\right)$$

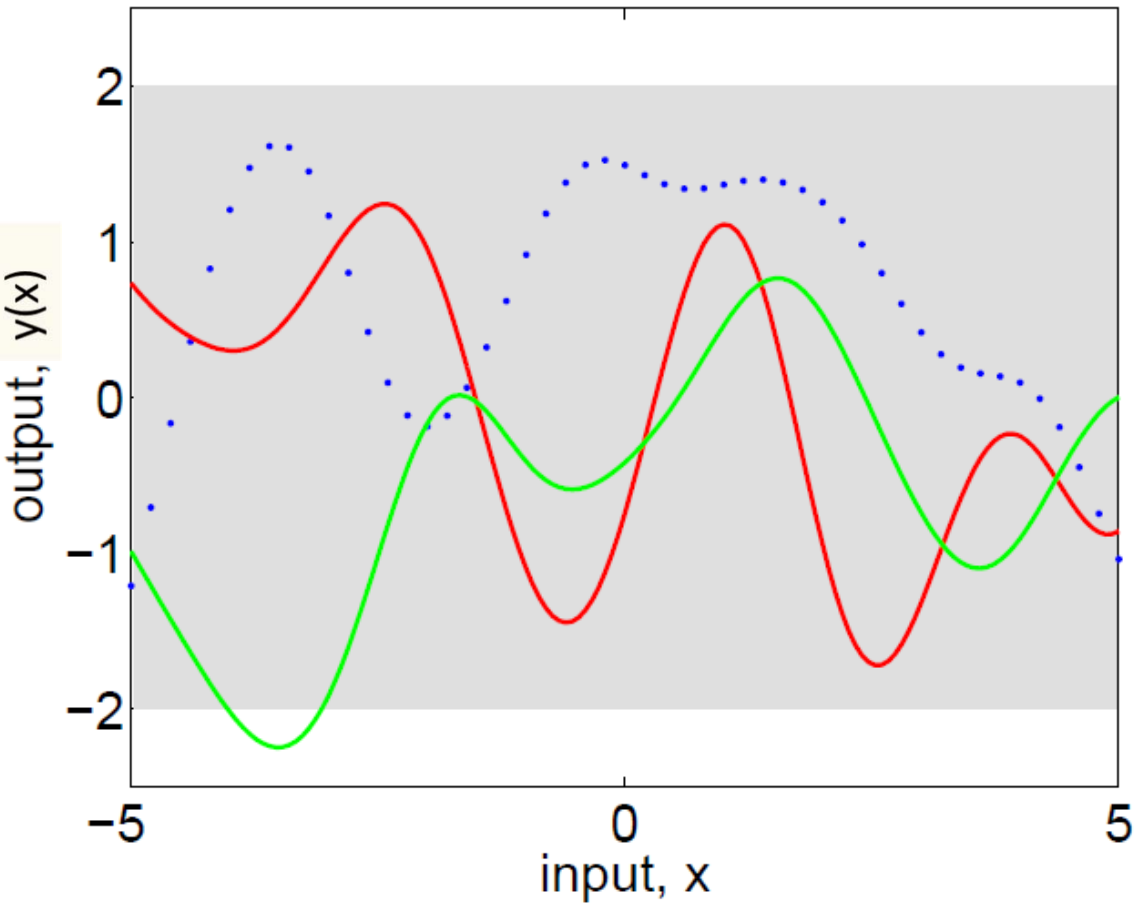
One widely used kernel function for Gaussian process is given by the exponential of a quadratic form,

$$k\left(\mathbf{x}_n, \mathbf{x}_m\right) = \theta_0 exp\left\{-\frac{\theta_1}{2}\left\|\mathbf{x}_n - \mathbf{x}_m\right\|^2\right\} + \theta_2 + \theta_3 \mathbf{x}_n^T \mathbf{x}_m$$

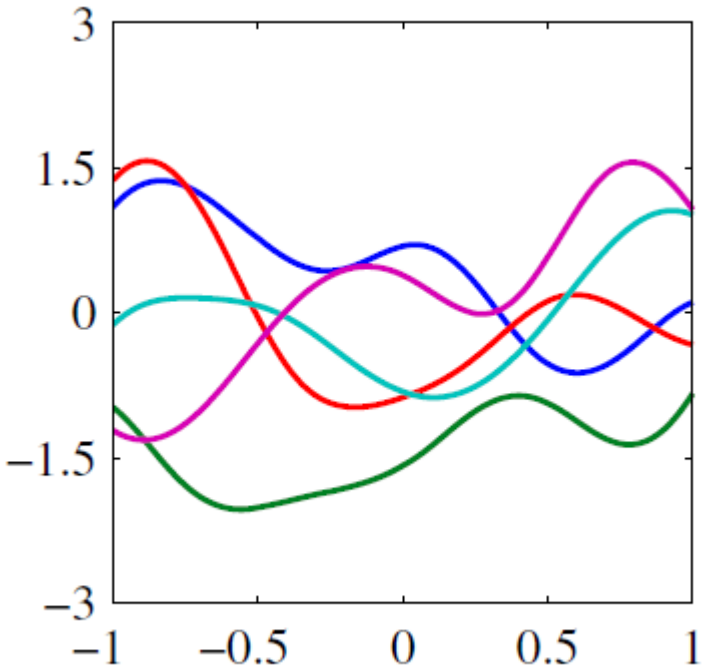
here $\boldsymbol{\theta} = \{\theta_0, \theta_1, \theta_2, \theta_3, \beta\}$

Samples from Gaussian process prior

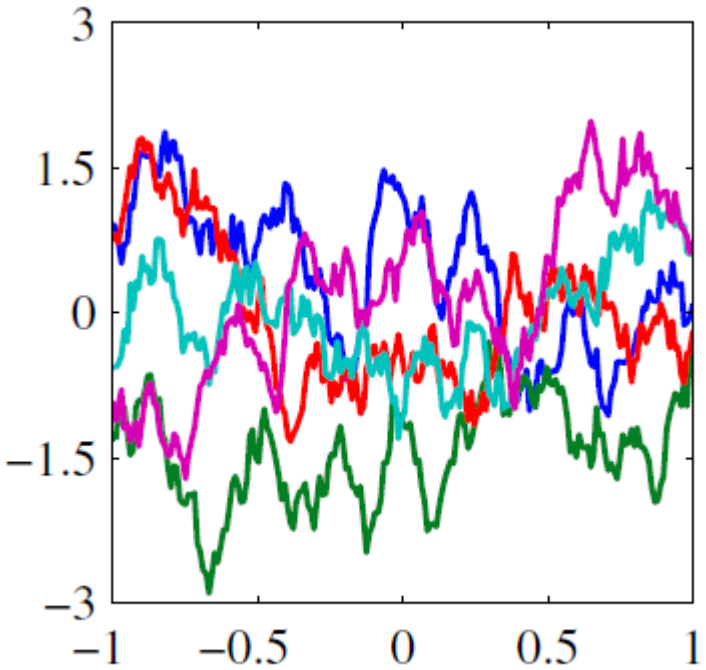
Three functions drawn at random form a Gaussian process prior, the dots indicate values of y actually generated; the two other functions have been drawn as lines by joining a large number of evaluated points



Samples from Gaussian process prior



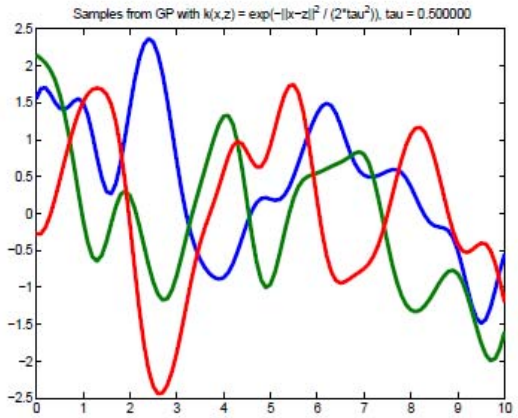
Gaussian kernel



Exponential kernel

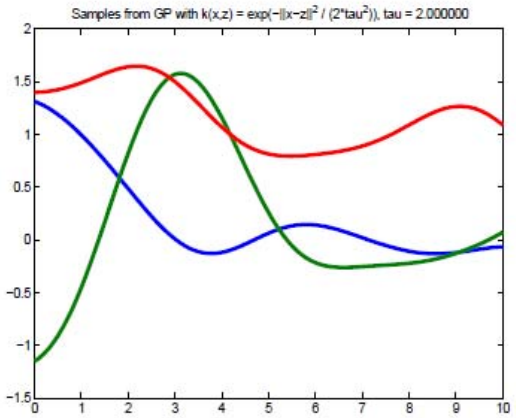
Samples from Gaussian process prior

Samples from a zero mean Gaussian process prior with covariance function of Gaussian kernel



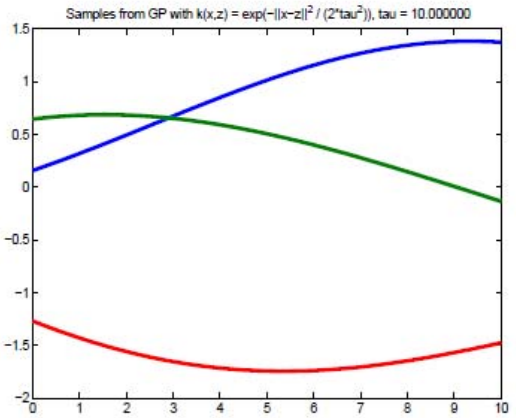
(a)

$\sigma = 0.5$



(b)

$\sigma = 2$



(c)

$\sigma = 10$

Samples from Gaussian process prior

As we know the Gaussian kernel is

$$k\left(\mathbf{x}_n, \mathbf{x}_m\right) = exp\left(-\|\mathbf{x}_n - \mathbf{x}_m\|^2 / 2\sigma^2\right)$$

The larger the hyper-parameter σ is, the more strongly correlated $y\left(\mathbf{x}_n\right)$ and $y\left(\mathbf{x}_m\right)$ are.

Gaussian process regression

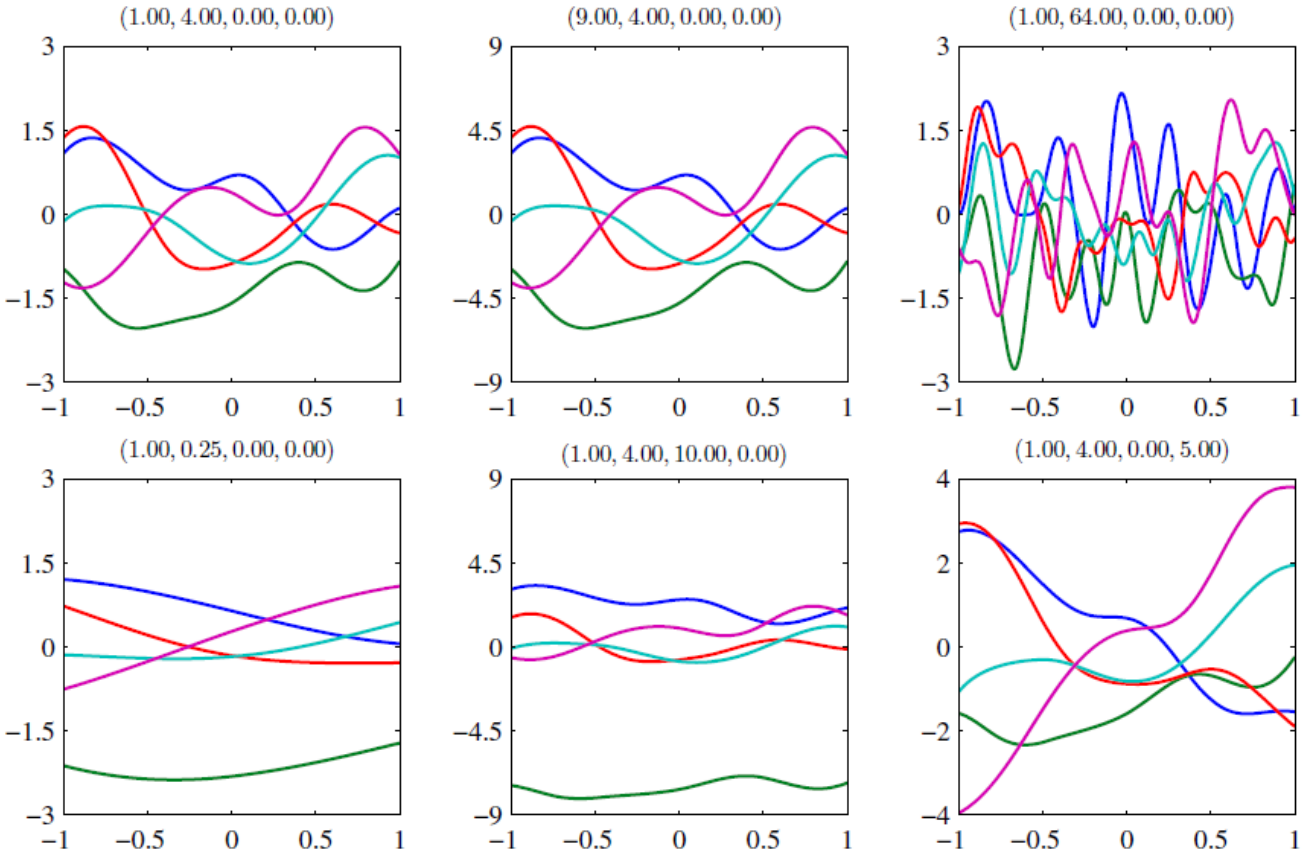


Figure 6.5 Samples from a Gaussian process prior defined by the covariance function (6.63). The title above each plot denotes $(\theta_0, \theta_1, \theta_2, \theta_3)$.

Gaussian process regression

The marginal distribution of target values

$$p\left(\mathbf{t}|\mathbf{X}\right) = \int p\left(\mathbf{t}|\mathbf{y}\right) p\left(\mathbf{y}|\mathbf{X}\right) d\mathbf{y} = \mathcal{N}\left(\mathbf{t}|\mathbf{0}, \mathbf{C}\right)$$

where the covariance matrix **C** has elements

$$C\left(\mathbf{x}_n, \mathbf{x}_m\right) = k\left(\mathbf{x}_n, \mathbf{x}_m\right) + \beta^{-1}\delta_{nm}$$

Gaussian process regression

The objective function of Gaussian process regression

$$\ln p\left(\mathbf{t}|\mathbf{X}\right) = -\frac{1}{2}\ln|\mathbf{C}| - \frac{1}{2}\mathbf{t}^T\mathbf{C}^{-1}\mathbf{t} - \frac{N}{2}\ln 2\pi$$

It is nonconvex.

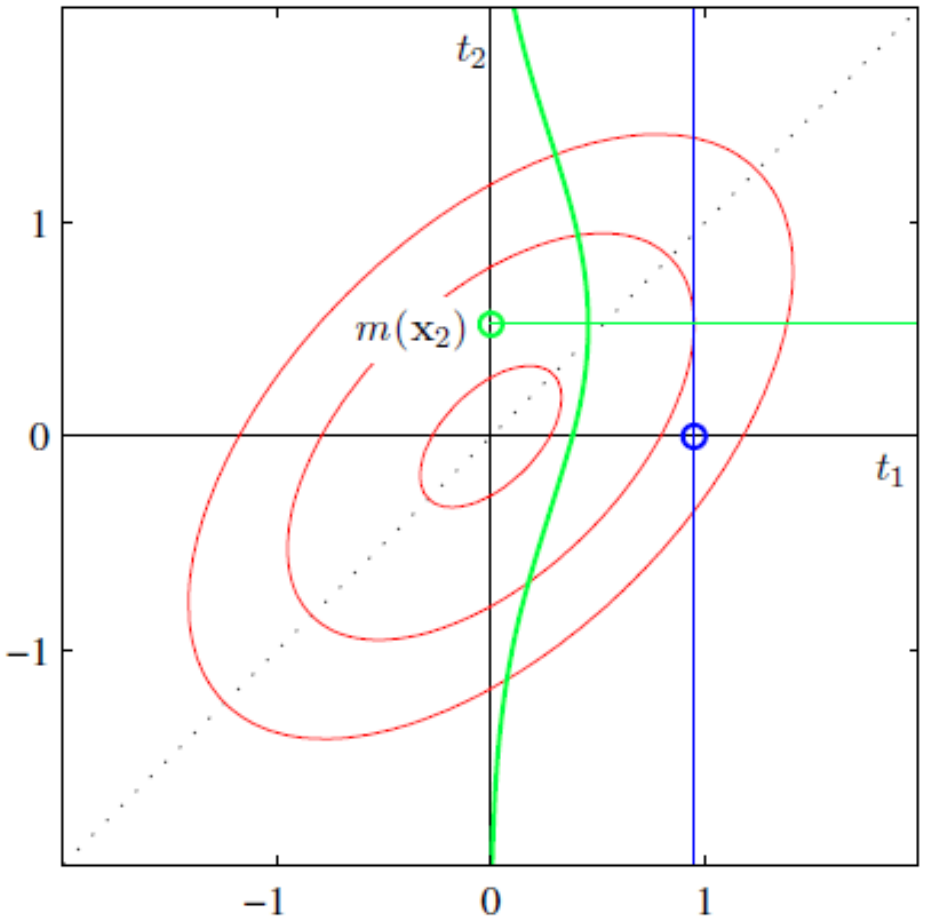
Gaussian process regression

To get the optimal hyper-parameters, we use gradient descent method.

$$\frac{\partial}{\partial \theta_i} \ln p\left(\mathbf{t}|\mathbf{X}\right) = -\frac{1}{2} Tr\left(\mathbf{C}^{-1}\frac{\partial \mathbf{C}}{\partial \theta_i}\right) + \frac{1}{2}\mathbf{t}^T\mathbf{C}^{-1}\frac{\partial \mathbf{C}}{\partial \theta_i}\mathbf{C}^{-1}\mathbf{t}$$

Gaussian process regression

Illustration of the mechanism of Gaussian process regression for the case of one training point and one test point, in which the red ellipses show contours of the joint distribution $p(t_1, t_2)$. Here t_1 is the training data point, and conditioning on the value of t_1, corresponding to the vertical blue line, we obtain $p(t_2|t_1)$ shown as a function of t_2 by the green curve.



Gaussian process regression

Given a new input vector $\mathbf{x}_{N+1}$, the joint distribution of target values is

$$p\left(\mathbf{t}_{N+1}|\mathbf{X}_{N+1}\right)=\mathcal{N}(\mathbf{t}_{N+1}|\mathbf{0},\mathbf{C}_{N+1})$$

where the covariance matrix is

$$\mathbf{C}_{N+1}=\begin{pmatrix}\mathbf{C}_N & \mathbf{k}\\ \mathbf{k}^{\mathrm{T}} & c\end{pmatrix}$$

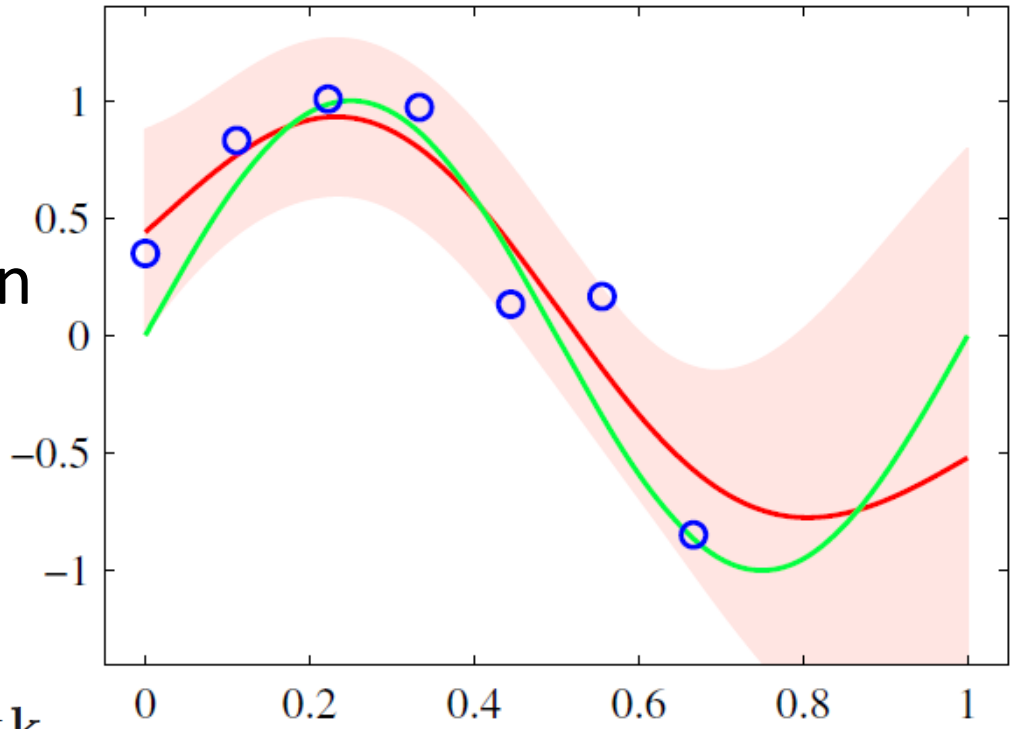
Gaussian process regression

The conditional distribution

$$p\left(t_{N+1}|\mathbf{X}, \mathbf{t}, \mathbf{x}_{N+1}\right)$$

is a Gaussian Distribution with mean and covariance given by

$$\begin{aligned} m(\mathbf{x}_{N+1}) &= \mathbf{k}^{\mathrm{T}}\mathbf{C}_N^{-1}\mathbf{t} \\ \sigma^2(\mathbf{x}_{N+1}) &= c - \mathbf{k}^{\mathrm{T}}\mathbf{C}_N^{-1}\mathbf{k} \end{aligned}$$



Gaussian process regression

$$m(\mathbf{x}_{N+1}) = \mathbf{k}^{\mathrm{T}} \mathbf{C}_N^{-1} \mathbf{t}$$

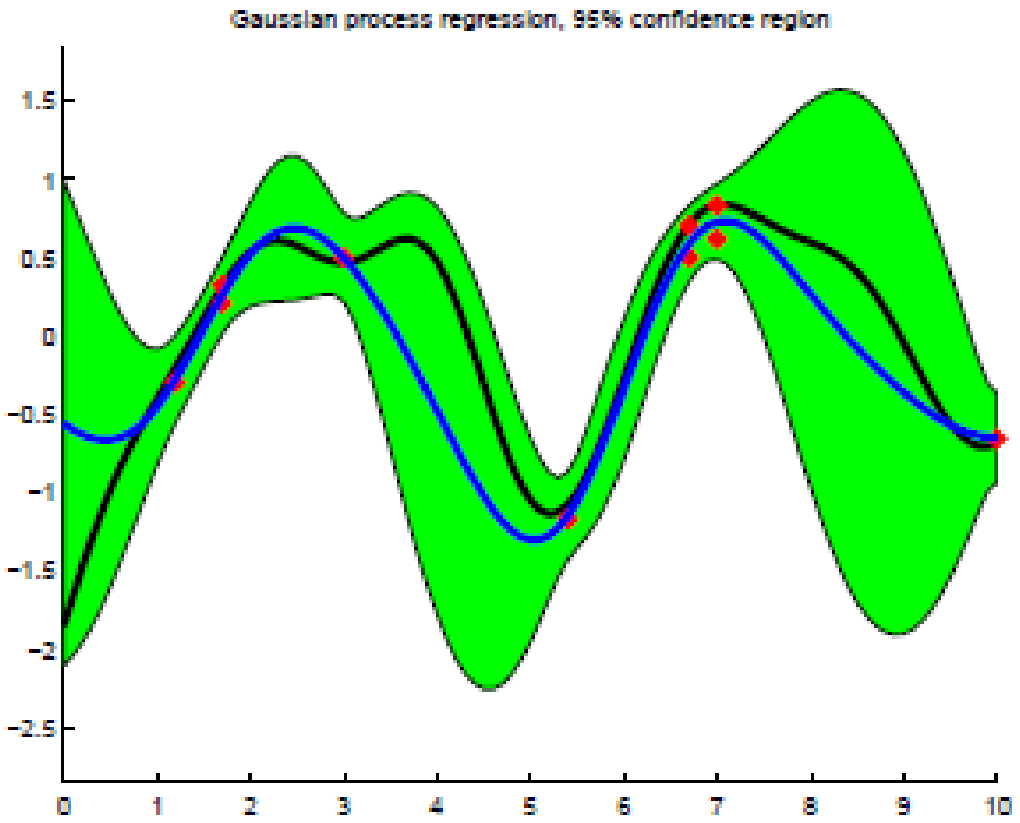
The mean of the predictive distribution can be written as a function of $\mathbf{x}_{N+1}$

$$m\left(\mathbf{x}_{N+1}\right) = \sum_{n=1}^{N} a_n k\left(\mathbf{x}_n, \mathbf{x}_{N+1}\right)$$

where a_n is the nth component of $\mathbf{C}_N^{-1}\mathbf{t}$.

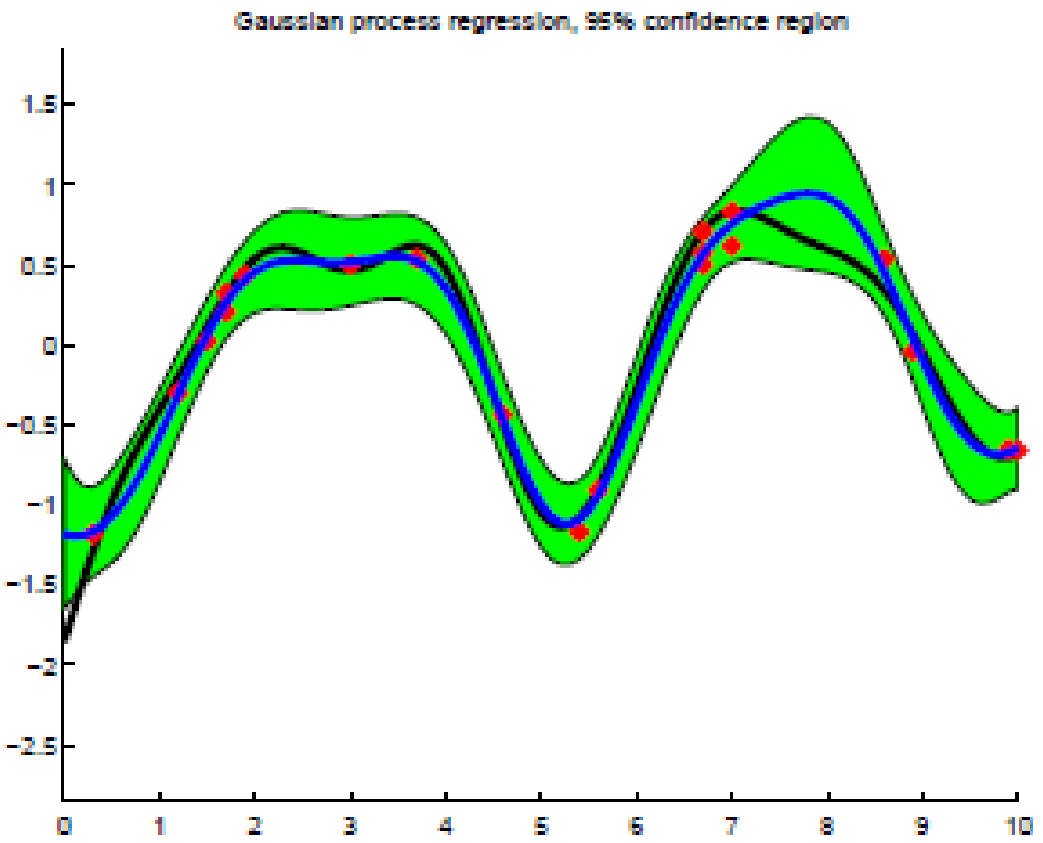
Gaussian process regression

The number of training examples is 10, noise level is $\sigma = 1$



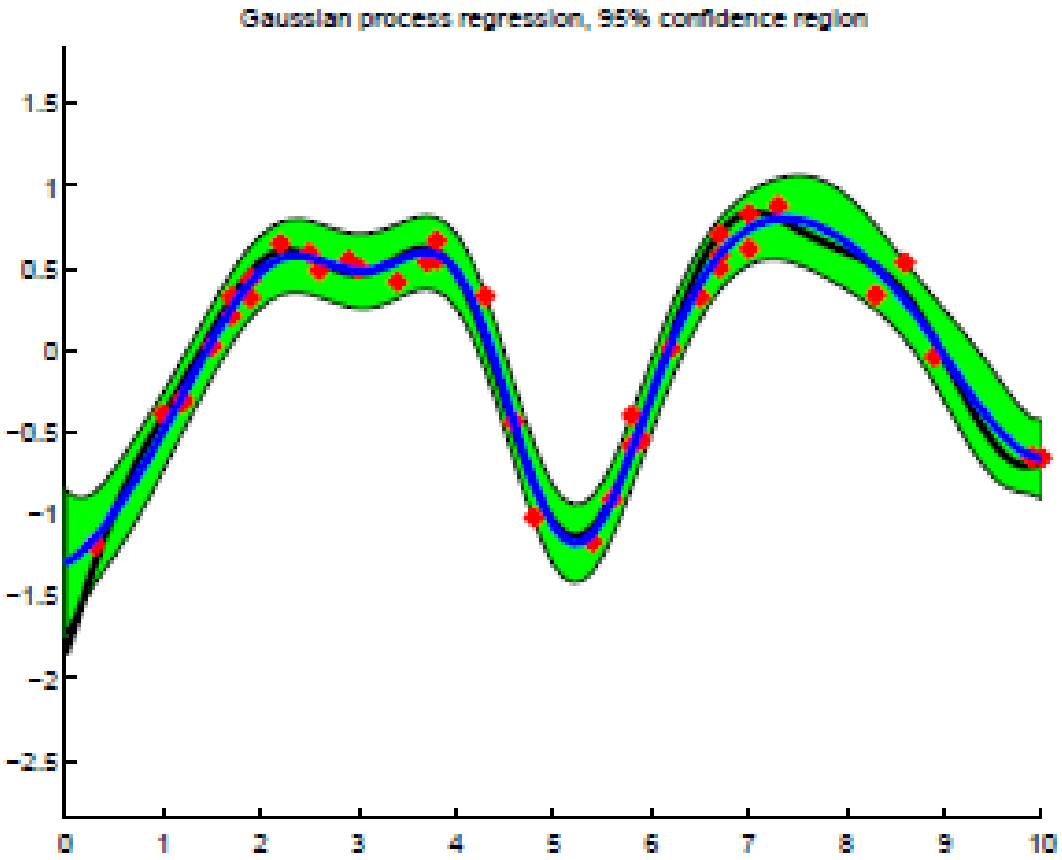
Gaussian process regression

The number of training examples is 20, noise level is $\sigma = 1$



Gaussian process regression

The number of training examples is 40, noise level is $\sigma = 1$



Linear regression revisited

Consider the linear model

$$y(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})$$

A prior distribution over **w** given by an isotropic Gaussian of the form

$$p(\mathbf{w}) = \mathcal{N}\left(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I}\right)$$

governed by the hyper-parameter α.

Linear regression revisited

Let **y** denote model outputs $\mathbf{y} = [y(\mathbf{x}_1), ..., y(\mathbf{x}_N)]^T$, we have

$$\mathbf{y} = \mathbf{\Phi}\mathbf{w}$$

where

$$\mathbf{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}$$

Linear regression revisited

y is a linear transformation of **x**, therefore **y** is Gaussian.

$$\begin{aligned} \mathbb{E}[\mathbf{y}] &= \mathbf{\Phi}\mathbb{E}[\mathbf{w}] = \mathbf{0} \\ \mathrm{cov}[\mathbf{y}] &= \mathbb{E}\left[\mathbf{y}\mathbf{y}^{\mathrm{T}}\right] = \mathbf{\Phi}\mathbb{E}\left[\mathbf{w}\mathbf{w}^{\mathrm{T}}\right]\mathbf{\Phi}^{\mathrm{T}} = \frac{1}{\alpha}\mathbf{\Phi}\mathbf{\Phi}^{\mathrm{T}} = \mathbf{K} \end{aligned}$$

where **K** is the gram matrix with elements

$$K_{nm} = \boldsymbol{\phi}\left(\mathbf{x}_n\right)^T \boldsymbol{\phi}\left(\mathbf{x}_m\right) = k\left(\mathbf{x}_n, \mathbf{x}_m\right)$$

Linear regression revisited

This model provides us with a particular example of Gaussian process.

The set of values of y(**x**) evaluated at an arbitrary set of point $\mathbf{x}_1, \ldots, \mathbf{x}_N$ jointly have a Gaussian distribution.