Kernelized methods

Kairit Sirts

16.05.2014

Kernel function

- Kernel function is the inner product of the feature vectors:

$$K(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x})^T \phi(\mathbf{z})$$

- Kernel function can be constructed by either working out the inner product of the feature vectors
- or by combining Mercer's kernels.
- **Gram matrix** $\mathbf{K}$ is a $m \times m$ symmetric matrix with elements:

$$K_{ij} = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j)$$

Some popular kernel functions

- Linear kernel:

$$K(\mathbf{x}, \mathbf{z}) = \mathbf{x}^T \mathbf{z} + b$$

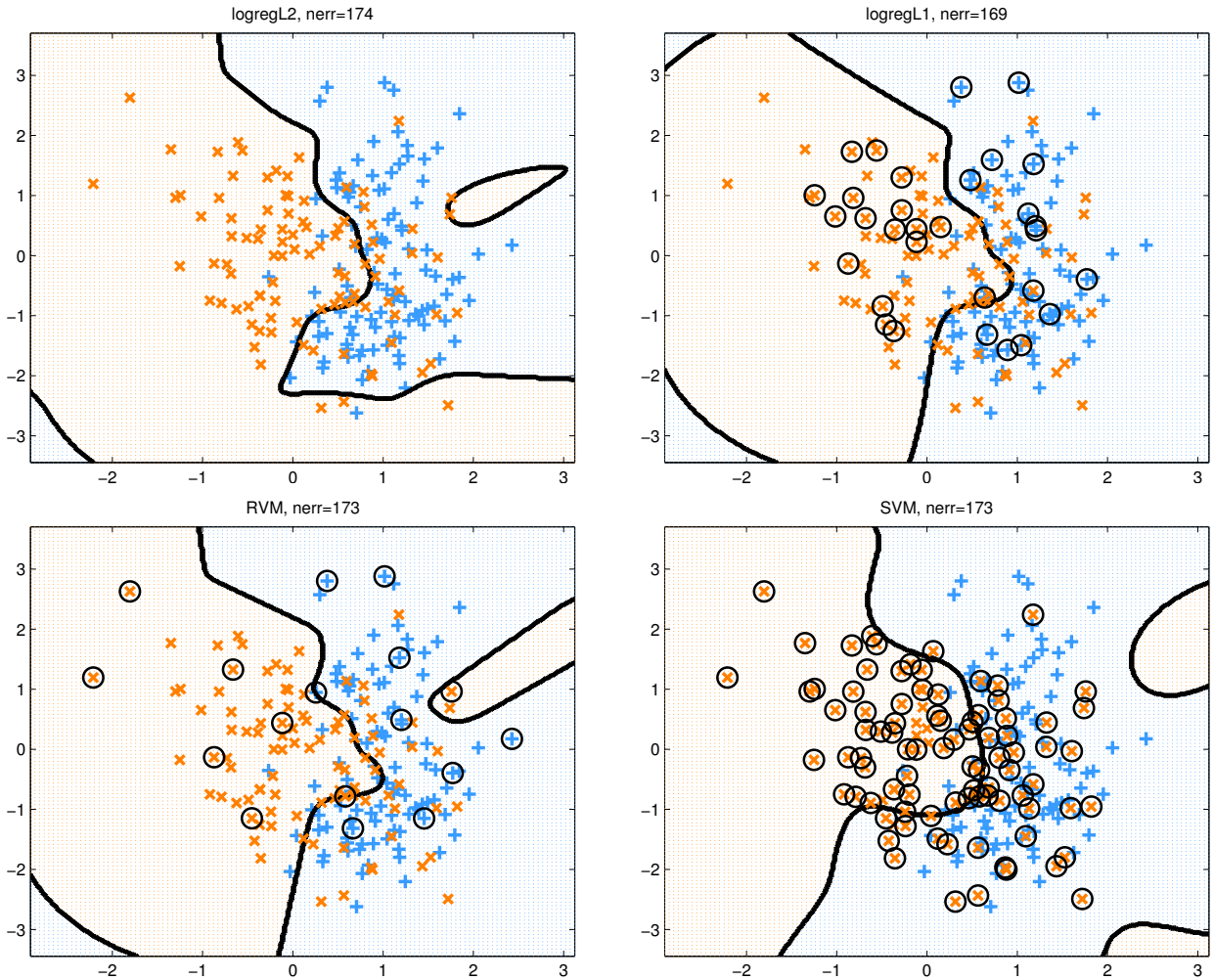
- Polynomial kernel:

$$K(\mathbf{x}, \mathbf{z}) = (r + \gamma \mathbf{x}^T \mathbf{z})^p$$

- Gaussian kernel:

$$K(\mathbf{x}, \mathbf{z}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\sigma^2}\right)$$

Examples of non-linear classification with SVM



Kernelized Linear Regression

- Recall that the cost function is given as:

$$J(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{m}(\mathbf{w}^T\phi(\mathbf{x}_i) - y_i)^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

- Set the gradient with respect to $\mathbf{w}$ to zero and express $\mathbf{w}$:

$$\mathbf{w} = -\frac{1}{\lambda}\sum_{i=1}^{m}(\mathbf{w}^T\phi(\mathbf{x}_i) - y_i)\phi(\mathbf{x}_i) = \sum_{i=1}^{m} a_i\phi(\mathbf{x}_i) = \Phi^T\mathbf{a}$$

- Φ is the design matrix containing feature vectors
- $\mathbf{a}$ is the vector $(a_1, \ldots, a_m)^T)$ where

$$a_i = -\frac{1}{\lambda}(\mathbf{w}^T\phi(\mathbf{x}_i) - y_i)$$

Dual for kernelized linear regression

- Substitute $\mathbf{w} = \Phi^T \mathbf{a}$ back to the cost function:

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

- Gram matrix is defined here by $\mathbf{K} = \Phi\Phi^T$
- Because the Gram matrix entries are inner products of feature vectors it defines a kernel function.
- In terms of the Gram matrix, the cost function can be written as:

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

Kernelized linear regression

- $\mathbf{w}^T\phi(\mathbf{x})$ still exists in the definition of a_i-s
- Substitute $\mathbf{w} = \Phi^T\mathbf{a}$ and solve for $\mathbf{a}$:

$$\mathbf{a} = (\mathbf{K} + \lambda\mathbf{I}_m)^{-1}\mathbf{y}$$

- Substitute it to the model:

$$h(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x}) = \mathbf{a}^T\Phi\phi(\mathbf{x}) = \mathbf{k}(\mathbf{x})^T(\mathbf{K} + \lambda\mathbf{I}_m)^{-1}\mathbf{y}$$

- $\mathbf{k}(\mathbf{x})$ is a vector with elements: $k_i(\mathbf{x}) = K(\mathbf{x}_i, \mathbf{x})$

Some remarks

- Solving the kernelized linear regression model requires inverting the matrix of size $m \times m$, where m is the number of training items
- Solving the linear regression model in primal form required inverting the matrix of size $n \times n$, where n is the number of features
- Typically m is much larger than n, so in that sense the kernelized version does not seem to be very useful
- However, in dual form, the feature vectors are only expressed via the kernel function and this enables to work in very high dimensional feature spaces

Probabilistic linear regression

- Notation:

inputs:	$\mathbf{x}_i$
outputs:	y_i
predictions:	$h_i = h(\mathbf{x}_i) = \mathbf{w}^T \phi(\mathbf{x}_i)$

- Predictions are given as usual:

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

- Weight vector is given a Gaussian prior:

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \sigma^2 \mathbf{I})$$

- For each different value of $\mathbf{w}$ there is a different function $h(\mathbf{x})$
- $p(\mathbf{w})$ also induces probability distribution over functions $h(\mathbf{x})$

Probabilistic linear regression

- We wish to evaluate the function $h(\mathbf{x})$ at the training points $\mathbf{x}_1, \ldots, \mathbf{x}_m$

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

- $\mathbf{h}$ is the vector with elements $h_i = h(\mathbf{x}_i)$
- We are interested in the probability distribution over $\mathbf{h}$
- Note that the linear combination of independent Gaussian random variables is also Gaussian
- Each component in $\mathbf{h}$ is a linear combination of the Gaussian components from $\mathbf{w}$
- Therefore $\mathbf{h}$ is also a Gaussian

Distribution of $\mathbf{h}$

- As $\mathbf{h}$ is a Gaussian we need to find its mean and covariance:

$$\mathbb{E}[\mathbf{h}] = \Phi\mathbb{E}[\mathbf{w}] = \mathbf{0}$$

- Covariance in general can be computed as:

$$\text{cov}[\mathbf{x}, \mathbf{y}] = \mathbb{E}_{\mathbf{x},\mathbf{y}}[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{y}^T - \mathbb{E}[\mathbf{y}^T])]$$

- Thus, the covariance of the $\mathbf{h}$ is:

$$\begin{aligned}\text{cov}[\mathbf{h}] &= \mathbb{E}[(\mathbf{h} - \mathbb{E}[\mathbf{h}])(\mathbf{h}^T - \mathbb{E}[\mathbf{h}^T])] \\ &= \mathbb{E}[\mathbf{h}\mathbf{h}^T] = \Phi\mathbb{E}[\mathbf{w}\mathbf{w}^T]\Phi^T = \sigma^2\Phi\Phi^T = \mathbf{K}\end{aligned}$$

- $\mathbf{K}$ is the gram matrix with elements:

$$K_{ij} = K(\mathbf{x}_i, \mathbf{x}_k) = \sigma^2\phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$$

Gaussian processes

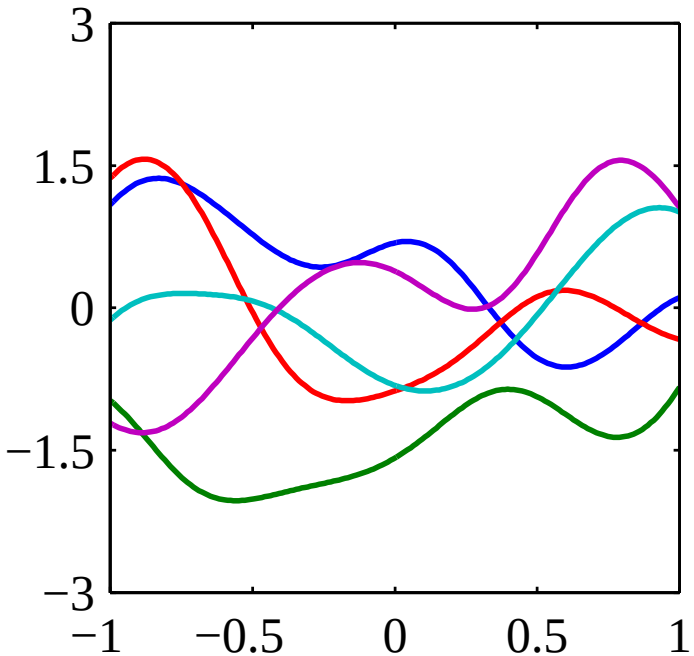
- The presented model is an example of a Gaussian process
- A Gaussian process is defined as a probability distribution over functions $h(\mathbf{x})$ such that the joint distribution over a set of values evaluated at arbitrary points $\mathbf{x}_1, \ldots, \mathbf{x}_m$ is a Gaussian
- A key property of the Gaussian processes that the joint distribution over a set of m values is completely specified by the mean vector and the co-variance matrix
- Usually, there is no prior information about the mean and so it's set to zero
- The co-variance is given by the kernel function:

$$\mathbb{E}[h(\mathbf{x}_i)h(\mathbf{x}_j)] = K(\mathbf{x}_i, \mathbf{x}_j)$$

Example of Gaussian processes

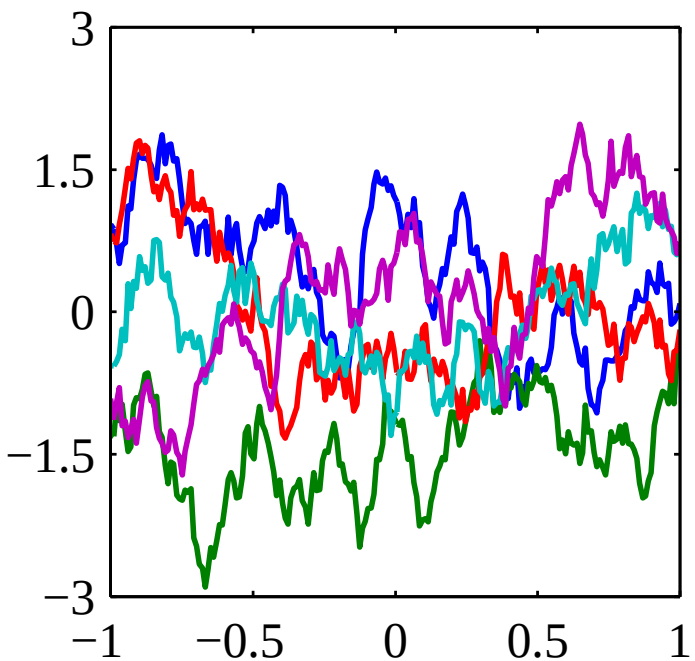
Gaussian kernel:

$$K(\mathbf{x}, \mathbf{z}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\sigma}\right)$$



Exponential kernel:

$$K(\mathbf{x}, \mathbf{z}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{z}\|}{2\sigma}\right)$$



Gaussian processes for regression

- Targets include noise:

$$y_i = h_i + \epsilon_i \qquad \epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

- Thus we can express:

$$P(y_i|h_i) = \mathcal{N}(y_i|h_i, \sigma^2)$$

- Because we assume that the noise terms are independent:

$$P(\mathbf{y}|\mathbf{h}) = \mathcal{N}(\mathbf{y}|\mathbf{h}, \sigma^2\mathbf{I}_m)$$

Gaussian processes for regression

- According to the definition of the Gaussian processes:

$$P(\mathbf{h}) = \mathcal{N}(\mathbf{h}|\mathbf{0}, \mathbf{K})$$

- The choice of the kernel function depends on the application and should be chosen such that for similar points $\mathbf{x}_i$ and $\mathbf{x}_j$ the values h_i and h_j would be more correlated that for dissimilar points.
- For obtaining the marginal probability distribution over outputs we have to integrate over hypotheses:

$$P(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{h})p(\mathbf{h})d\mathbf{h} = \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{C})$$

- Covariance matrix $\mathbf{C}$ elements are:

$$C(\mathbf{x}_i, \mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j) + \sigma^2 \mathbb{I}(i = j)$$

Digression: Finding the marginal Gaussian

- Given a marginal Gaussian for $\mathbf{x}$:

$$P(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- and a conditional Gaussian of $\mathbf{y}$ given $\mathbf{x}$:

$$P(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{x} + \mathbf{b}, \mathbf{S})$$

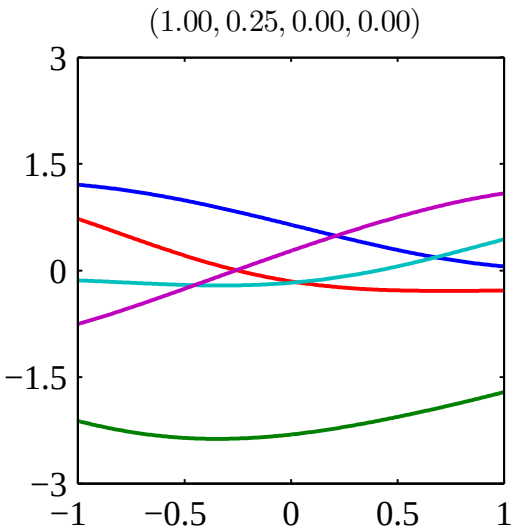
- Then the marginal distribution of $\mathbf{y}$ is Gaussian and can be found as:

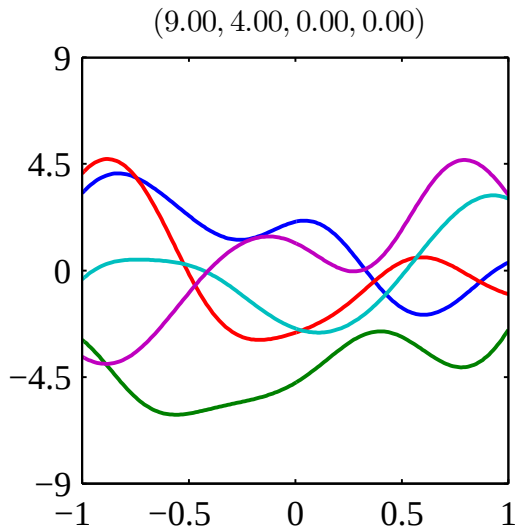
$$P(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{S} + \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^T)$$

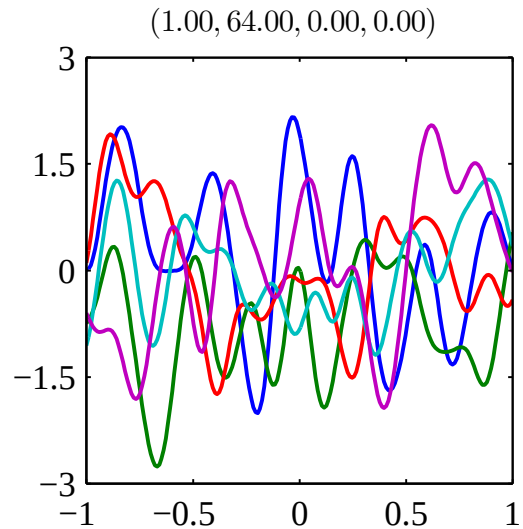
Example kernel for Gaussian process regression

- A widely used kernel function for Gaussian process regression is:

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

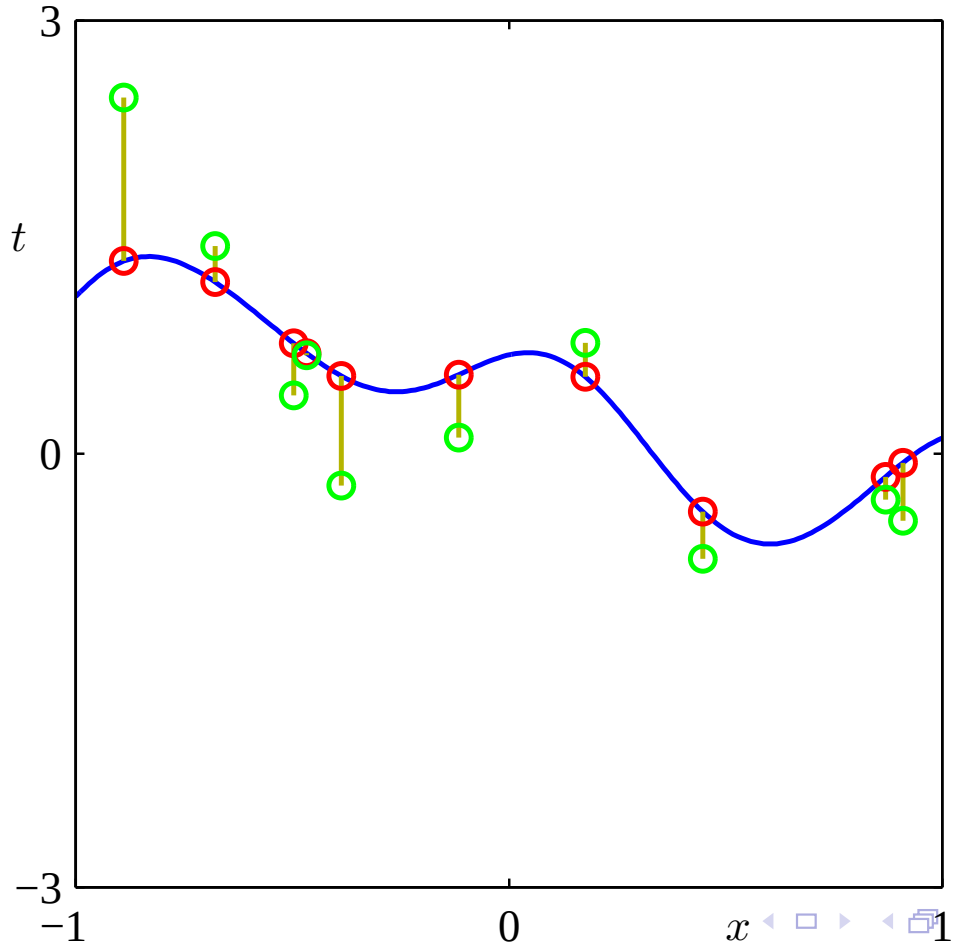






Example draw from a Gaussian process prior

- Blue line is the sampled function
- Red points show the values of h_i evaluated on a set of points
- Green points are the noisy observations at the same set of points



Predictions in Gaussian process regression

- Assume we have m training data points with observed labels
- Our goal is to make prediction y_{m+1} for a new datapoint $\mathbf{x}_{m+1}$
- For that we have to find the predictive distribution:

$$P(y_{m+1}|\mathbf{y})$$

- We begin from the joint distribution:

$$P(\mathbf{y}_{m+1}) = P(\mathbf{y}, y_{m+1}) = \mathcal{N}(\mathbf{y}_{m+1}|\mathbf{0}, \mathbf{C}_{m+1})$$

- $\mathbf{C}_{m+1}$ is a $(m+1) \times (m+1)$ co-variance matrix with entries:

$$C(\mathbf{x}_i, \mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j) + \sigma^2 \mathbb{I}(i = j)$$

Predictive distribution

- Partitition the co-variance matrix:

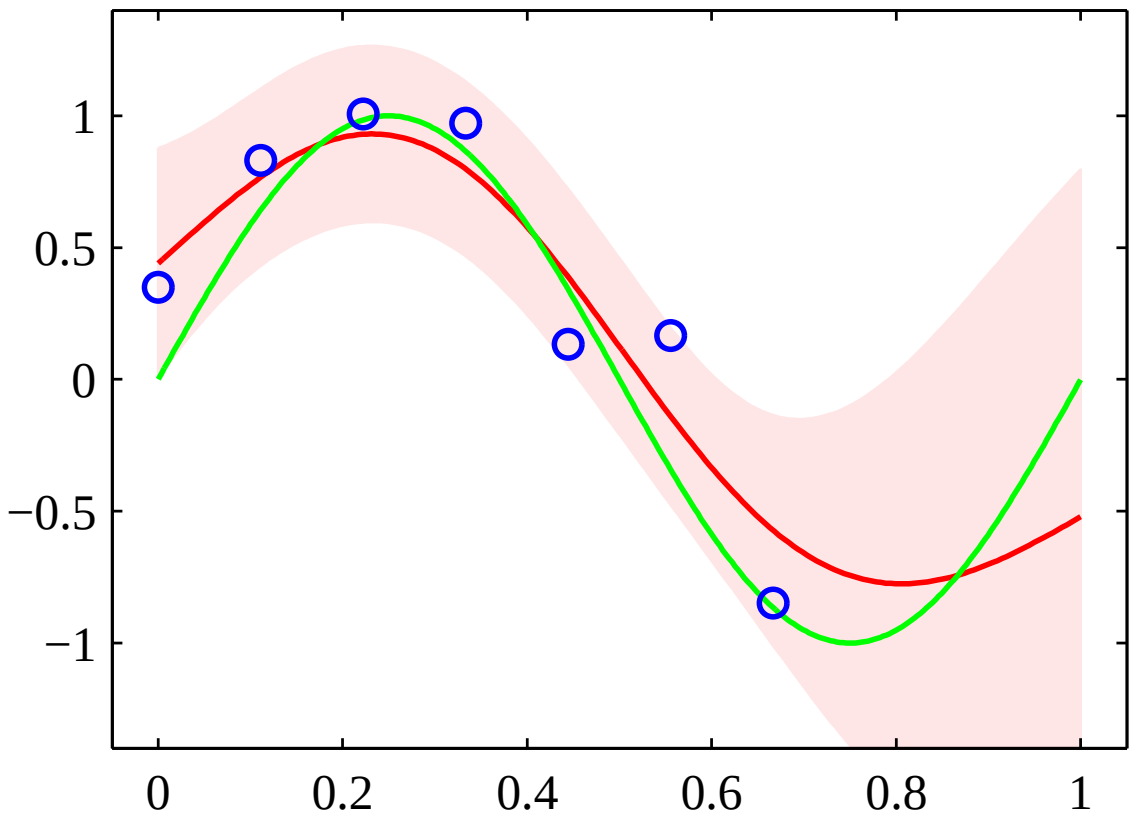
$$\mathbf{C}_{m+1} = \begin{pmatrix} \mathbf{C}_m & \mathbf{k} \\ \mathbf{k}^T & c \end{pmatrix}$$

- $\mathbf{C}_m$ is the covariance matrix for $\mathbf{y}$
- $\mathbf{k}$ is a vector with elements $K(\mathbf{x}_i, \mathbf{x}_{m+1})$ for $i = 1, \ldots, m$
- c is a number $c = K(\mathbf{x}_{m+1}, \mathbf{x}_{m+1}) + \sigma^2$
- Using some properties of the Gaussians, we get the mean and covariance of the predictive distribution:

$$\boldsymbol{\mu}(\mathbf{x}_{m+1}) = \mathbf{k}^T \mathbf{C}_m^{-1} \mathbf{y}$$
$$\sigma^2(\mathbf{x}_{m+1}) = c - \mathbf{k}^T \mathbf{C}_m^{-1} \mathbf{k}$$

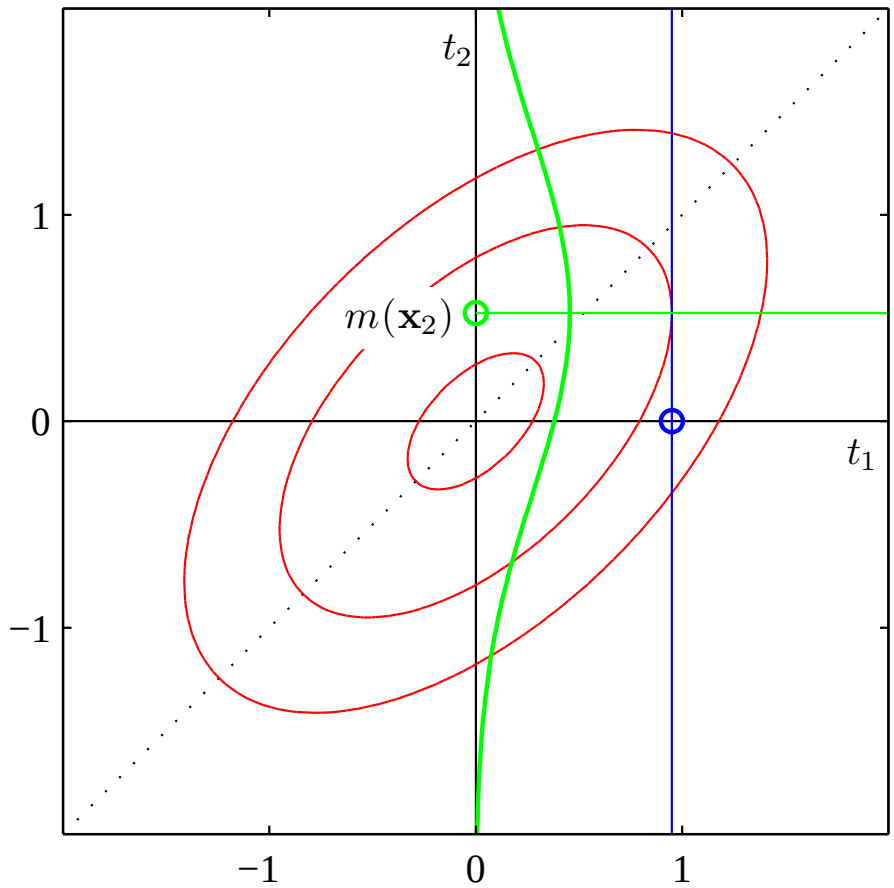
Gaussian process regression prediction: example

- Green is the underlying function
- Blue points are the noisy observations
- Red line is the mean of the Gaussian process predictive distribution
- Shaded region corresponds to ± 2 standard deviations



Gaussian process regression prediction: example

- Example with one training point and one test point
- Red lines show the contours of $p(t_1, t_2)$
- Green line show the predictive distribution $p(t_2|t_1)$



Mean of the predictive distribution

- The mean of the predictive distribution for the point $\mathbf{x}_{m+1}$ is:

$$\boldsymbol{\mu}(\mathbf{x}_{m+1}) = \sum_{i=1}^{m} a_i K(\mathbf{x}_i, \mathbf{x}_{m+1})$$

- where a_i is the i-th component of $\mathbf{C}_m^{-1}\mathbf{y}$

Remarks

- Gaussian processes regression requires the inversion of a matrix of size $m \times m$
- This inversion must be done only once for a given training set
- For each new test point we must do a vector-matrix multiplication
- If the feature vector dimensionality n is finite and it is smaller than m, it will be more efficient to perform Bayesian linear regression in the original parameter space
- The advantage of Gaussian processes is the ability to use covariance functions that can only expressed in infinite-dimensional feature space
- For large training sets, the direct application of Gaussian processes can be infeasible, so one must use some approximation scheme

Learning the hyperparameters

- Hyperparameters are the parameters in the covariance function (kernel)
- One can do a grid-search on development set
- Better way is to estimate the hyperparameter values from the data
- Point estimates can be obtained by maximizing the log-likelihood $p(\mathbf{y}|\boldsymbol{\theta})$ using gradient-based optimization techniques
- We can also introduce prior to the parameters and maximize the log-posterior