Visual Analytics
Lecture week 11: Gaussian Processes

Ian T. Nabney

University of BRISTOL

Overview

- Understand fundamentals of Gaussian Processes (GPs) from weight-space and function viewpoints
- Able to train GPs
- Able to apply ARD to GPs

Further reading: Bishop section 6.3 and Nabney chapter 10.

From linear regression to Gaussian processes

- We have considered linear regression models of the form $y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x})$ in which $\mathbf{w}$ is a vector of parameters and $\phi(\mathbf{x})$ is a vector of fixed nonlinear basis functions that depend on the input vector $\mathbf{x}$.
- We showed that a prior distribution over $\mathbf{w}$ induced a corresponding prior distribution over functions $y(\mathbf{x}, \mathbf{w})$. Given a training data set, we then evaluated the posterior distribution over $\mathbf{w}$ and thereby obtained the corresponding posterior distribution over regression functions, which in turn (with the addition of noise) implies a predictive distribution $p(t|\mathbf{x})$ for new input vectors $\mathbf{x}$.
- In the Gaussian process viewpoint, we dispense with the parametric model and instead define a prior probability distribution over functions directly.
- At first sight, it might seem difficult to work with a distribution over the uncountably infinite space of functions. However, as we shall see, for a finite training set we only need to consider the values of the function at the discrete set of input values $\mathbf{x}_n$ corresponding to the training set and test set data points, and so in practice we can work in a finite space.

Derivation of GP prior

- Consider a prior distribution over $\mathbf{w}$ given by an isotropic Gaussian of the form

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

- For any given value of $\mathbf{w}$, this defines a particular function of $\mathbf{x}$ and the probability distribution over $\mathbf{w}$ defined by therefore induces a probability distribution over functions $y(\mathbf{x})$.
- In practice, we evaluate this function at specific values of $\mathbf{x}$, and we are therefore interested in the joint distribution of the function values $y(\mathbf{x}_1), \ldots, y(\mathbf{x}_N)$.
- $\boldsymbol{y}$ is a linear combination of Gaussian distributed variables given by the elements of $\mathbf{w}$ and hence is itself Gaussian. We therefore need only to find its mean and covariance, which are given by

$$\mathbb{E}[\boldsymbol{y}] = \boldsymbol{\Phi}\mathbb{E}[\mathbf{w}] = \mathbf{0} \tag{2}$$

$$\mathrm{cov}[\boldsymbol{y}] = \mathbb{E}\left[\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}\right] = \boldsymbol{\Phi}\mathbb{E}\left[\mathbf{w}\mathbf{w}^{\mathrm{T}}\right]\boldsymbol{\Phi}^{\mathrm{T}} = \frac{1}{\alpha}\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}} = \mathbf{K} \tag{3}$$

where $\mathbf{K}$ is the Gram matrix with elements $K_{nm} = k(\mathbf{x}_n, \mathbf{x}_m) = \frac{1}{\alpha}\phi(\mathbf{x}_n)^{\mathrm{T}}\phi(\mathbf{x}_m)$ and $k(\mathbf{x}, \mathbf{x}')$ is the kernel function.

University of BRISTOL

Definition of GP prior

- A Gaussian process is defined as a probability distribution over functions $y(\mathbf{x})$ such that the set of values of $y(\mathbf{x})$ evaluated at an arbitrary set of points $\mathbf{x}_1, \ldots, \mathbf{x}_N$ jointly have a Gaussian distribution.
- This is specified completely by the second-order statistics, namely the mean and the covariance.
- In most applications, we will not have any prior knowledge about the mean of $y(\mathbf{x})$ and so by symmetry we take it to be zero. This is equivalent to choosing the mean of the prior over weight values $p(\mathbf{w}|\alpha)$ to be zero in the basis function viewpoint.
- The specification of the Gaussian process is then completed by giving the covariance of $y(\mathbf{x})$ evaluated at any two values of $\mathbf{x}$, which is given by the kernel function

$$\mathbb{E}\left[y(\mathbf{x}_n)y(\mathbf{x}_m)\right] = k(\mathbf{x}_n, \mathbf{x}_m). \tag{4}$$

Kernel choices

- The squared exponential covariance function is the exponential of a weighted squared distance between points in $\mathbb{R}^d$:

$$C(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = v_0 \exp\left(-\frac{1}{2}\sum_{l=1}^{d} a_l (x_l^{(i)} - x_l^{(j)})^2\right) + b. \tag{5}$$

- The rational quadratic covariance is given by:

$$C(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = v_0 \left(1 + \sum_{l=1}^{d} a_l (x_l^{(i)} - x_l^{(j)})^2\right)^{-\nu} + b. \tag{6}$$

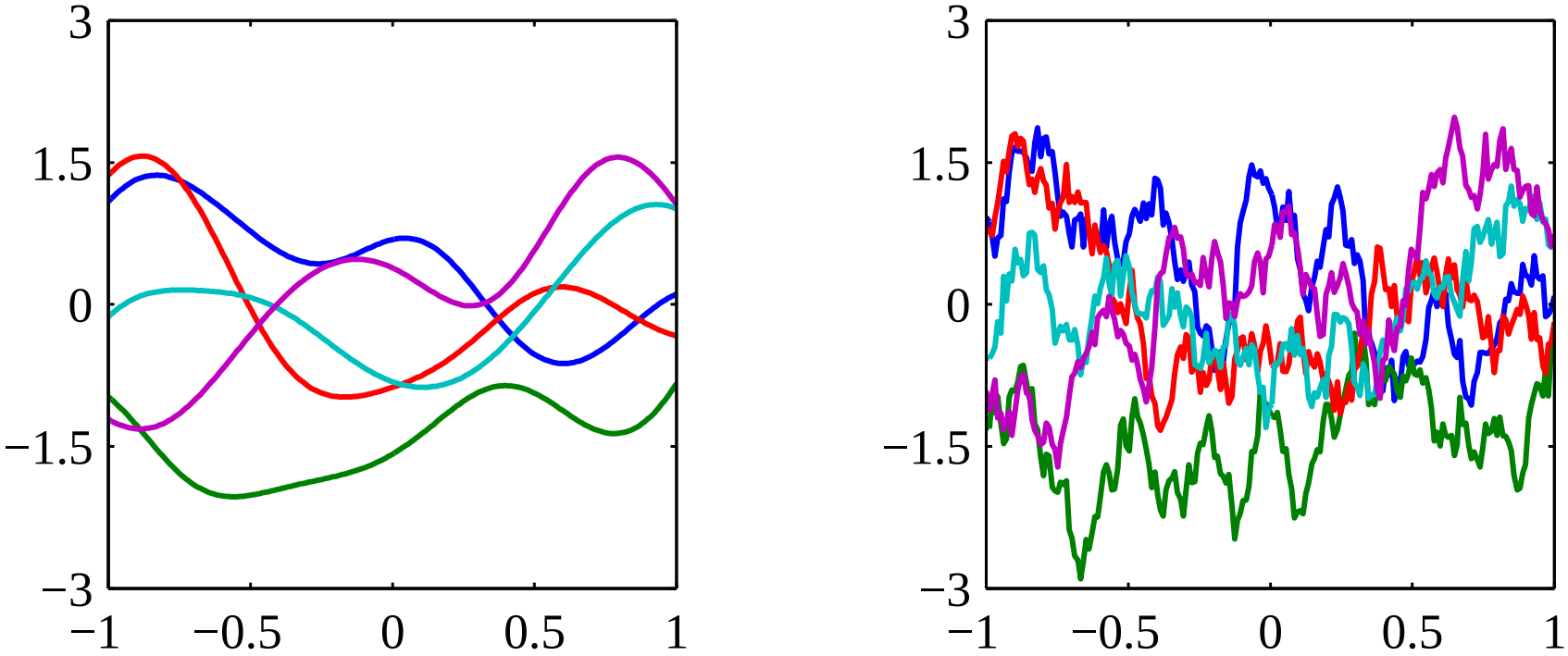
- The exponential kernel given by

$$k(x, x') = \exp\left(-\theta \left|x - x'\right|\right) \tag{7}$$

- scikit-learn contains several kernels for the GaussianProcessRegressor model[1]. These include squared exponential (called RBF), rational quadratic, Matérn, exp-sine, dot product. It also provides a framework that enables you to define your own kernels.

[1] https://scikit-learn.org/stable/modules/gaussian_process.html#gp-kernels

Sampling from a GP prior



Samples from Gaussian processes for a 'Gaussian' kernel (left) and an exponential kernel (right).

GPs for regression

- We need to take account of the noise on the observed target values: we shall assume that they have a Gaussian distribution, so that

$$p(t_n|y_n) = \mathcal{N}(t_n|y_n, \beta^{-1}) \tag{8}$$

- The joint distribution of the target values $\boldsymbol{t} = (t_1, \ldots, t_N)^{\mathrm{T}}$ conditioned on $\boldsymbol{y}$ is given by an isotropic Gaussian of the form

$$p(\boldsymbol{t}|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{t}|\boldsymbol{y}, \beta^{-1}\mathbf{I}_N) \tag{9}$$

- To find the marginal distribution $p(\boldsymbol{t})$, conditioned on the input values $\mathbf{x}_1, \ldots, \mathbf{x}_N$, we need to integrate over $\boldsymbol{y}$.
-

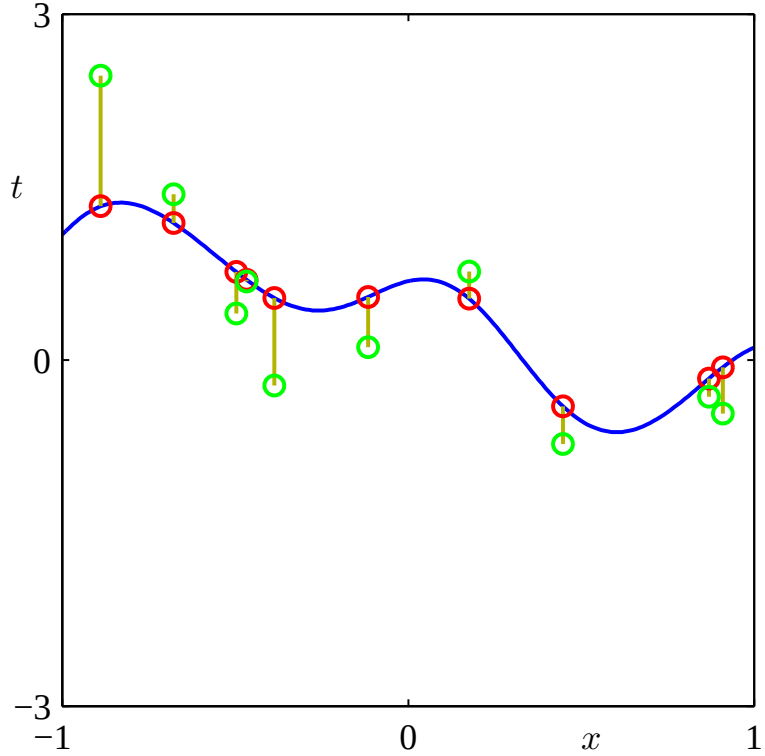
$$p(\boldsymbol{t}) = \int p(\boldsymbol{t}|\boldsymbol{y})p(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} = \mathcal{N}(\boldsymbol{t}|\mathbf{0}, \mathbf{C}) \tag{10}$$

where the covariance matrix $\mathbf{C}$ has elements

$$C(\mathbf{x}_n, \mathbf{x}_m) = k(\mathbf{x}_n, \mathbf{x}_m) + \beta^{-1}\delta_{nm}. \tag{11}$$

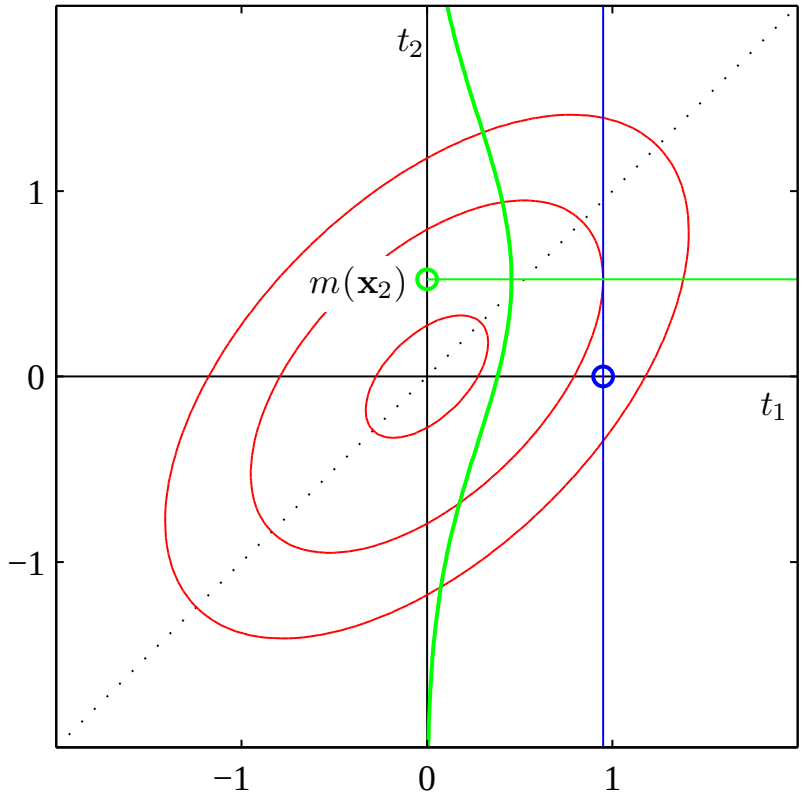
Sampling data from a GP

- Illustration of the sampling of data points $\{t_n\}$ from a Gaussian process. item The blue curve shows a sample function from the Gaussian process prior over functions, and the red points show the values of y_n obtained by evaluating the function at a set of input values $\{x_n\}$.
- The corresponding values of $\{t_n\}$, shown in green, are obtained by adding independent Gaussian noise to each of the $\{y_n\}$.



Making predictions

- Suppose that $\boldsymbol{t}_N = (t_1, \ldots, t_N)^{\mathrm{T}}$ and $\mathbf{x}_1, \ldots, \mathbf{x}_N$, comprise the training set, and our goal is to predict the target variable t_{N+1} for a new input vector $\mathbf{x}_{N+1}$.
- This requires that we evaluate the predictive distribution $p(t_{N+1}|\boldsymbol{t}_N)$.



GP regression for the case of one training point t_1 and one test point t_2. Red ellipses show contours of the joint distribution $p(t_1, t_2)$. Conditioning on the value of t_1 (vertical blue line), we obtain $p(t_2|t_1)$ shown as a function of t_2 by the green curve.

Predictive equations

- The joint distribution over $t_1, \ldots, t_{N+1}$ is

$$p(\boldsymbol{t}_{N+1}) = \mathcal{N}(\boldsymbol{t}_{N+1}|\mathbf{0}, \mathbf{C}_{N+1}) \tag{12}$$

  where $\mathbf{C}_{N+1}$ is an $(N+1) \times (N+1)$ covariance matrix with elements given by (11).
- Because this joint distribution is Gaussian, we can by partitioning the covariance matrix

$$\mathbf{C}_{N+1} = \begin{pmatrix} \mathbf{C}_N & \mathbf{k} \\ \mathbf{k}^{\mathrm{T}} & c \end{pmatrix} \tag{13}$$

  where $\mathbf{C}_N$ is the $N \times N$ covariance matrix with elements given by (11) for $n, m = 1, \ldots, N$, the vector $\mathbf{k}$ has elements $k(\mathbf{x}_n, \mathbf{x}_{N+1})$ for $n = 1, \ldots, N$, and the scalar $c = k(\mathbf{x}_{N+1}, \mathbf{x}_{N+1}) + \beta^{-1}$.
- Then the conditional distribution $p(t_{N+1}|\boldsymbol{t})$ is a Gaussian distribution with mean and covariance given by

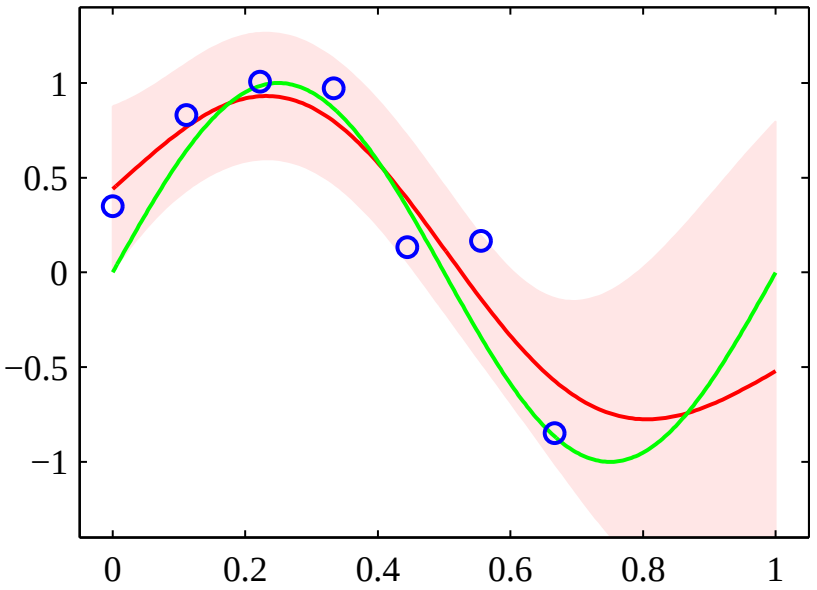
$$m(\mathbf{x}_{N+1}) = \mathbf{k}^{\mathrm{T}} \mathbf{C}_N^{-1} \boldsymbol{t} \tag{14}$$

$$\sigma^2(\mathbf{x}_{N+1}) = c - \mathbf{k}^{\mathrm{T}} \mathbf{C}_N^{-1} \mathbf{k}. \tag{15}$$

- Because the vector $\mathbf{k}$ is a function of the test point input value $\mathbf{x}_{N+1}$, we see that the predictive distribution is a Gaussian whose mean and variance both depend on $\mathbf{x}_{N+1}$.

GP regression example

- Gaussian process regression applied to a sinusoidal data set in some of the right-most data points have been omitted.
- The green curve shows the sinusoidal function from which the data points, shown in blue, are obtained by sampling and addition of Gaussian noise.
- The red line shows the mean of the Gaussian process predictive distribution, and the shaded region corresponds to plus and minus two standard deviations.
- Notice how the uncertainty increases in the region to the right of the data points.



Learning hyperparameters

- In practice, rather than fixing the covariance function, we may prefer to use a parametric family of functions and then infer the parameter values from the data.
- We evaluate the likelihood function $p(\boldsymbol{t}|\boldsymbol{\theta})$ where $\boldsymbol{\theta}$ denotes the hyperparameters of the Gaussian process model.
- The simplest approach is to make a point estimate of $\boldsymbol{\theta}$ by maximizing the log likelihood function. This is equivalent to the type 2 evidence procedure for linear regression models. Maximization of the log likelihood can be done using efficient gradient-based optimization algorithms such as conjugate gradients.
- The log likelihood function for a Gaussian process regression model is easily evaluated

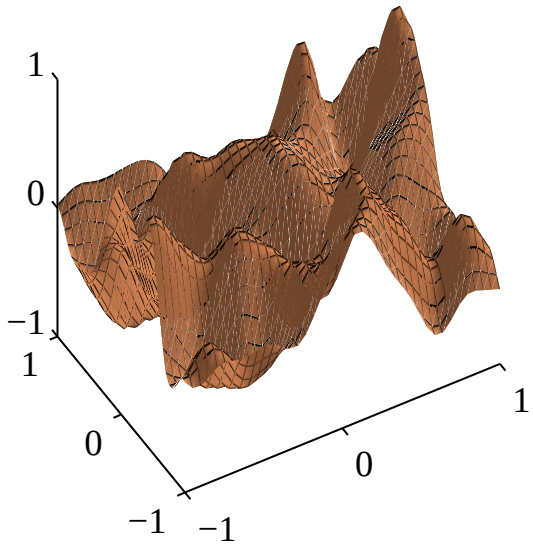
$$\ln p(\boldsymbol{t}|\boldsymbol{\theta}) = -\frac{1}{2}\ln|\mathbf{C}_N| - \frac{1}{2}\boldsymbol{t}^{\mathrm{T}}\mathbf{C}_N^{-1}\boldsymbol{t} - \frac{N}{2}\ln(2\pi). \tag{16}$$

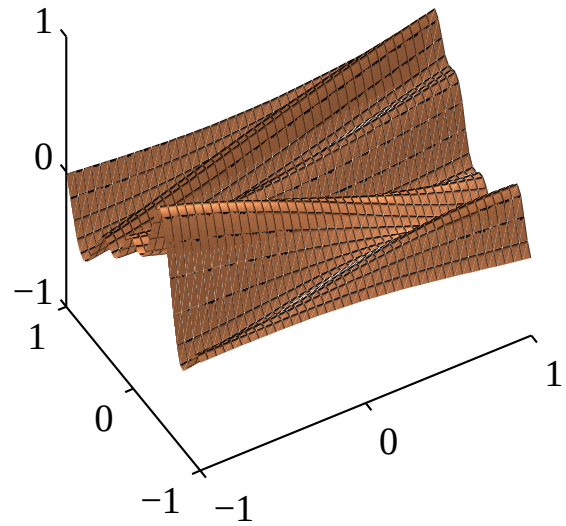
- For nonlinear optimization, we also need the gradient of the log likelihood function with respect to $\boldsymbol{\theta}$. We assume that evaluation of the derivatives of $\mathbf{C}_N$ is straightforward. We obtain

$$\frac{\partial}{\partial\theta_i}\ln p(\boldsymbol{t}|\boldsymbol{\theta}) = -\frac{1}{2}\mathrm{Tr}\left(\mathbf{C}_N^{-1}\frac{\partial\mathbf{C}_N}{\partial\theta_i}\right) + \frac{1}{2}\boldsymbol{t}^{\mathrm{T}}\mathbf{C}_N^{-1}\frac{\partial\mathbf{C}_N}{\partial\theta_i}\mathbf{C}_N^{-1}\boldsymbol{t}. \tag{17}$$

Sampling the GP-ARD prior

- Maximum likelihood can be used to determine a value for the correlation length-scale parameter in a Gaussian process.
- This technique can be extended by incorporating a separate parameter for each input variable.
- Optimization of these parameters by maximum likelihood allows the relative importance of different inputs to be inferred from the data.

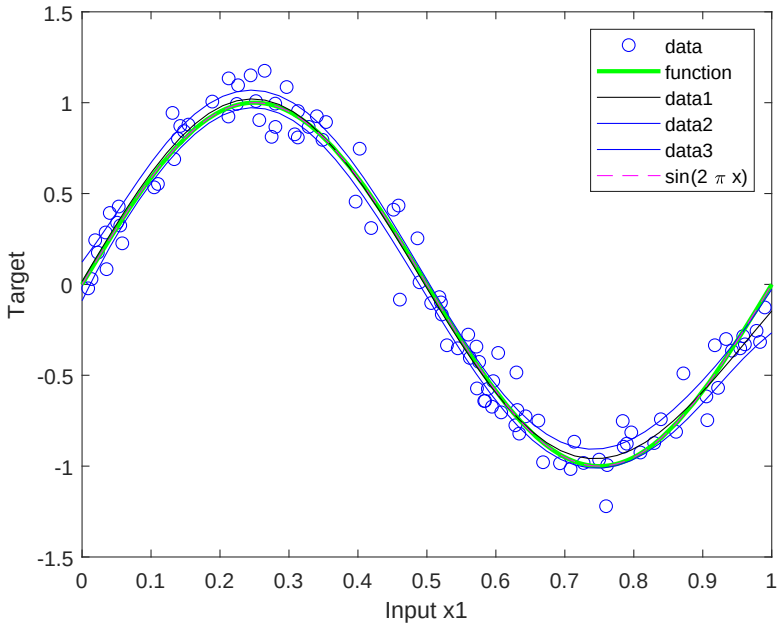




GP ARD regression

- We see that the inverse lengthscale associated with input x_1 is large, that of x_2 has an intermediate value and the variance of weights associated with x_3 is small.
- This implies that the Gaussian Process is giving greatest emphasis to x_1 and least emphasis to x_3, with intermediate emphasis on x_2 in the covariance function.

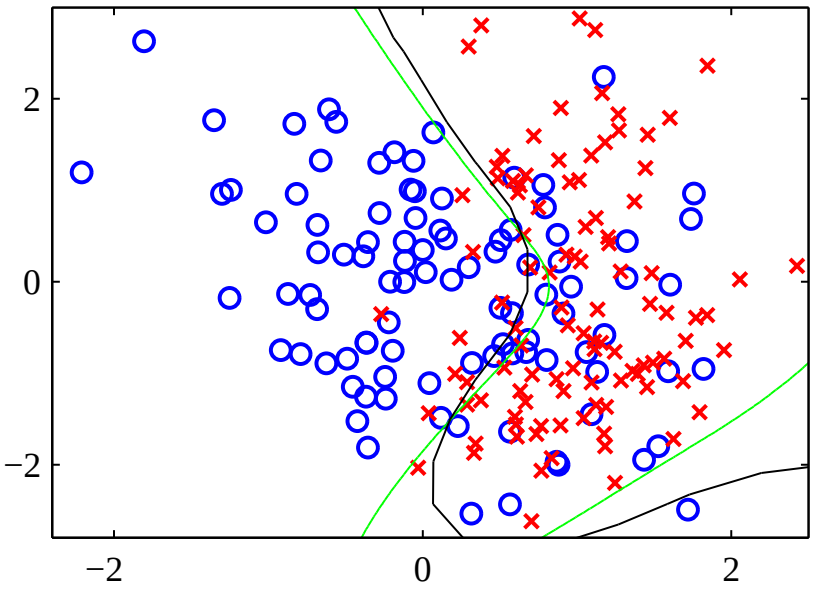
Input 1	8.959124
Input 2	0.056850
Input 3	0.000457



GP for classification

- In a probabilistic approach to classification, our goal is to model the posterior probabilities of the target variable for a new input vector, given a set of training data.
- We can adapt Gaussian processes to classification problems by transforming the output of the Gaussian process using an appropriate nonlinear activation function (such as a logistic sigmoid).
- The integral to compute $p(t_{N+1} = 1|\boldsymbol{t}_N)$ is analytically intractable, and so may be approximated using sampling methods.
- Alternatively, we can consider techniques based on an analytical approximation.
- Three different approaches to obtaining a Gaussian approximation have been considered.
  1. Variational inference makes use of the local variational bound on the logistic sigmoid. This allows the product of sigmoid functions to be approximated by a product of Gaussians thereby allowing the marginalization over $\mathbf{a}_N$ to be performed analytically.
  2. Expectation propagation can give good results.
  3. The Laplace approximation can be applied.

GP classification example



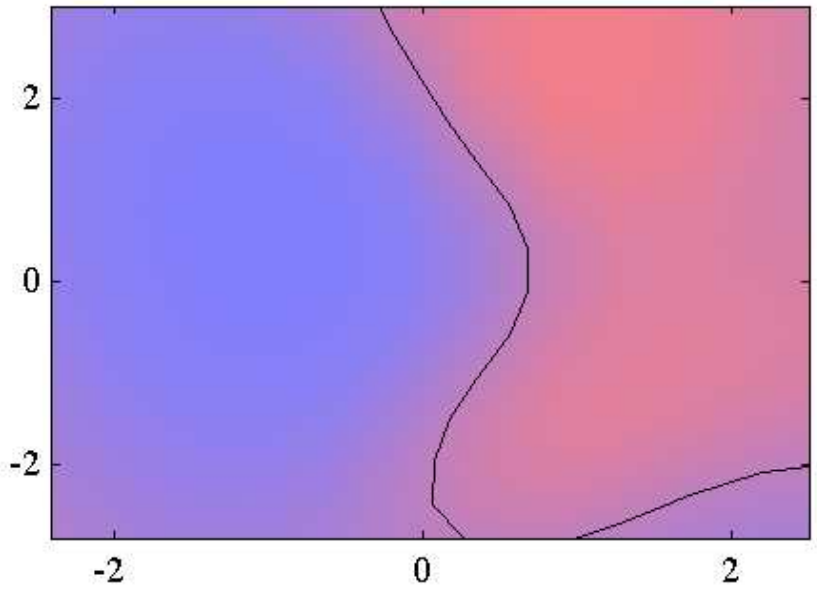


Figure shows the data on the left together with the optimal decision boundary from the true distribution in green, and the decision boundary from the Gaussian process classifier in black. On the right is the predicted posterior probability for the blue and red classes together with the Gaussian process decision boundary.

Practical issues

- The central computational operation in using Gaussian processes involves the inversion of a matrix of size $N \times N$, for which standard methods require $O(N^3)$ computations.
- By contrast, in the basis function model we have to invert a matrix $\mathbf{S}_N$ of size $M \times M$, which has $O(M^3)$ computational complexity.
- For each new test point, both methods require a vector-matrix multiply, which has cost $O(N^2)$ in the Gaussian process case and $O(M^2)$ for the linear basis function model.
- If the number M of basis functions is smaller than the number N of data points, it will be computationally more efficient to work in the basis function framework. However, an advantage of a Gaussian processes viewpoint is that we can consider covariance functions that can only be expressed in terms of an infinite number of basis functions.
- For large training data sets the direct application of Gaussian process methods can become infeasible, and so a range of approximation schemes have been developed that have better scaling with training set size than the exact approach.

Overview

- Understand fundamentals of Gaussian Processes (GPs) from weight-space and function viewpoints
- Able to train GPs
- Able to apply ARD to GPs

Further reading: Bishop section 6.3 and Nabney chapter 10.