Bayesian linear regression

Linear models for regression

Francesco Corona

Bayesian linear regression

In a maximum likelihood approach for setting parameters in a linear model for regression, we tune effective model complexity, the number of basis functions

- We control it based on the size of the data set

Adding a regularisation term to the log likelihood function means that the effective model complexity can be controlled by the regularisation coefficient

- The choice of the number and form of the basis functions is still important in determining the overall behaviour of the model

Bayesian linear regression (cont.)

This leaves the issue of setting appropriate model complexity for the problem

- It cannot be decided simply by maximising the likelihood function
- This always leads to excessively complex models and over-fitting

Remark

Independent hold-out data can be used to determine model complexity

- This can be both computationally expensive and wasteful of data

Bayesian linear regression (cont.)

We therefore turn to a Bayesian treatment of linear regression

- Avoids the over-fitting problem of maximum likelihood
- Leads to automatic methods of setting model complexity

We again focus on the case of a single target variable t

Outline

1. Bayesian linear regression
   - Parameter distribution
   - Predictive distribution
   - Equivalent kernel
2. Gaussian processes
   - Linear regression revisited
   - Gaussian processes for regression
   - Learning the hyper-parameters

Parameter distribution

Bayesian linear regression

Parameter distribution

The Bayesian treatment of linear regression starts by introducing a prior probability distribution over the model parameters $\mathbf{w}$[1]

The likelihood function $p(\mathtt{t}|\mathbf{w})$ is the exponential of a quadratic function of $\mathbf{w}$

$$p(\mathtt{t}|\mathbf{w}) = \prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{w}^T\phi(\mathbf{x}_n), \beta)$$

The corresponding conjugate prior is thus a Gaussian distribution of the form

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0) \tag{1}$$

- Mean $\mathbf{m}_0$ and covariance $\mathbf{S}_0$

[1] There also is the noise precision parameter β, we first assume it is a known constant

Parameter distribution (cont.)

The posterior distribution is $\propto$ to the product of likelihood function and prior

- Due to the choice of a conjugate prior, the posterior is Gaussian too[2][3]

$$p(\mathbf{w}|\mathtt{t}) \propto \Big(\prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{w}^T\phi(\mathbf{x}_n), \beta^{-1})\Big)\mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$$

$$\propto \exp\Big(-\frac{\beta}{2}(\mathtt{t} - \boldsymbol{\Phi})^T(\mathtt{t} - \boldsymbol{\Phi})\Big)\exp\Big(-\frac{1}{2}(\mathbf{w} - \mathbf{m}_0)^T\mathbf{S}_0^{-1}(\mathbf{w} - \mathbf{m}_0)\Big)$$

The posterior distribution can be thus written directly in the form

$$p(\mathbf{w}|\mathtt{t}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N) \tag{2}$$

$$\begin{aligned} \mathbf{m}_N &= \mathbf{S}_N(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta\boldsymbol{\Phi}^T\mathtt{t}) && (3) \\ \mathbf{S}_N^{-1} &= \mathbf{S}_0^{-1} - \beta\boldsymbol{\Phi}^T\boldsymbol{\Phi} && (4) \end{aligned}$$

[2] We derived something similar when discussing Bayes' theorem for Gaussian variables.

[3] This distribution is calculated by completing the square in the exponential and finding the normalisation coefficient using the result for a normalised Gaussian.

Parameter distribution (cont.)

Because the posterior distribution is Gaussian, mode and mean coincide

- The maximum posterior weight vector is given by $\mathbf{w}_{\text{MAP}} = \mathbf{m}_N$

If we consider an infinitely broad prior $\mathbf{S}_0 = \alpha^{-1}\mathbf{I}$ with $\alpha \to 0$, the mean $\mathbf{m}_N$ of the posterior distribution reduces to the maximum likelihood value

$$\mathbf{w}_{\text{ML}} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\mathtt{t}$$

Similarly, if $N = 0$, then again the posterior distribution reverts to the prior

Parameter distribution (cont.)

Consider a simple form of the Gaussian distribution, zero-mean isotropic

- Only a single precision parameter α characterises it

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I}) \tag{5}$$

The corresponding posterior distribution over $\mathbf{w}$ is $p(\mathbf{w}|\mathtt{t}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N)$

$$\mathbf{m}_N = \beta\mathbf{S}_N\boldsymbol{\Phi}^T\mathtt{t} \tag{6}$$

$$\mathbf{S}_N^{-1} = \alpha\mathbf{I} + \beta\boldsymbol{\Phi}^T\boldsymbol{\Phi} \tag{7}$$

Parameter distribution (cont.)

The log of the posterior distribution is given by the sum of the log likelihood and the log of the prior

- As a function of $\mathbf{w}$, it takes the form

$$\ln p(\mathbf{w}|\alpha) = -\frac{\beta}{2}\sum_{n=1}^{N}\Big(t_n - \mathbf{w}^T\phi(\mathbf{x}_n)\Big)^2 - \frac{\alpha}{2}\mathbf{w}^T\mathbf{w} + \text{const} \tag{8}$$

Maximisation of this posterior distribution with respect to $\mathbf{w}$ is equivalent to

$$\frac{1}{2}\sum_{n=1}^{N}\Big(t_n - \mathbf{w}^T\phi(\mathbf{x}_n)\Big)^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}, \quad \text{with } \lambda = \alpha/\beta$$

- the minimisation of the sum-of-squares error function
- with the addition of a quadratic regularisation term

Parameter distribution (cont.)

To illustrate Bayesian learning in a linear basis function model, together with the sequential update of a posterior distribution, we consider plain line fitting

Consider a single input variable x, a single target variable t and linear model

$$y(x, \mathbf{w}) = w_0 + w_1 x$$

We generate a synthetic set of data from function $f(x, \mathbf{a}) = a_0 + a_1 x$

- with $a_0 = -0.3$ and $a_1 = 0.5$

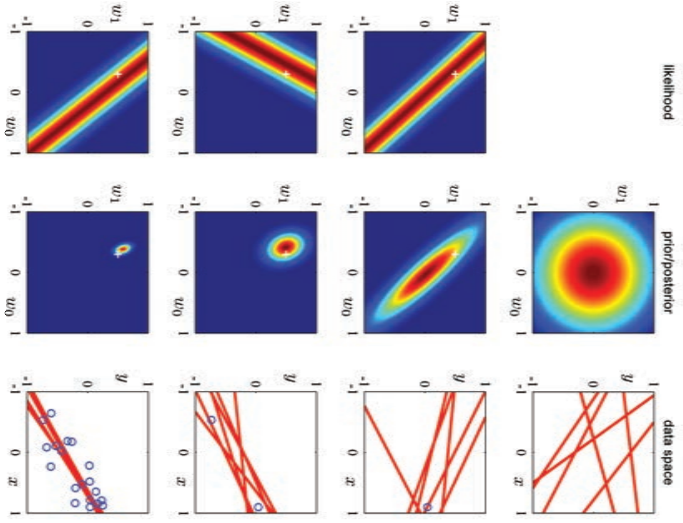
For a selection of input points $x_n \sim \mathcal{U}(-1, +1)$, we first evaluate $f(x_n, \mathbf{a})$ and then we add Gaussian noise $\varepsilon \sim \mathcal{N}(0, 0.2^2)$ to get the target values t_n

- The goal is to recover the values of a_0 and a_1 (thru w_0 and w_1)
- Under the assumption that the variance of the noise is known

$$\beta = \Big(\frac{1}{0.2}\Big)^2 = 25$$

- We fix $\alpha = 2.0$ in the Gaussian prior $p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I})$

Parameter distribution (cont.)



Because **w** is bi-dimensional, we can plot the prior and posterior distribution

Parameter distribution (cont.)

The plain Gaussian is not the only available form of prior over the parameters

- The Gaussian can be generalised

$$p(\mathbf{w}|\alpha) = \Big(\frac{q}{2}(\alpha/2)^{1/q}\frac{1}{\Gamma(1/q)}\Big)^M \exp\Big(-\frac{\alpha}{2}\sum_{j=0}^{M-1}|w_j|^q\Big) \qquad (9)$$

- It is not a conjugate prior to the likelihood function, unless $q = 2$

Finding the maximum of the posterior distribution over the parameters corresponds to the minimisation of a regularised error function

$$\frac{1}{2}\sum_{n=1}^{N}\Big(t_n - \mathbf{w}^T\phi(\mathbf{x}_n)\Big)^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|w_j|^q$$

Predictive distribution

Bayesian linear regression

Predictive distribution

In practice, we are not usually interested in the value of **w** itself

- We want to predictions of t for new values of **x**

This requires that we evaluate the **predictive distribution** defined by

$$p(t|\mathtt{t}, \alpha, \beta) = \int p(t|\mathbf{w}, \beta)p(\mathbf{w}|\mathtt{t}, \alpha, \beta)d\mathbf{w} \qquad (10)$$

where $\mathtt{t}$ is the vector of target values from the training set[4]

- The conditional distribution of the target is $p(t|\overset{\text{can be omitted}}{\cancel{\mathbf{x}}}, \mathbf{w}, \beta)$
- The posterior distribution of the weights is $p(\mathbf{w}|\mathtt{t}, \alpha, \beta)$

[4] We omit the corresponding input vectors $\mathtt{X}$ from the rhs of the conditioning to simplify notation

Predictive distribution (cont.)

Calculating the predictive distribution involves the convolution of Gaussians

$$p(t|\mathtt{t}, \alpha, \beta) = \int p(t|\mathbf{w}, \beta) p(\mathbf{w}|\mathtt{t}, \alpha, \beta) d\mathbf{w}$$

- The conditional distribution of the target

$$p(t|\mathbf{w}, \beta) = p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1}) \qquad \text{with} \begin{cases} y(\mathbf{x}, \mathbf{w}) = \phi(\mathbf{x})^T \mathbf{w} \\ \beta^{-1} \end{cases}$$

- The posterior distribution of the weights

$$p(\mathbf{w}|\mathtt{t}, \alpha, \beta) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N) \qquad \text{with} \begin{cases} \mathbf{m}_N = \mathbf{S}_N(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta \mathbf{\Phi}^T \mathtt{t}) \\ \mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} - \beta \mathbf{\Phi}^T \mathbf{\Phi} \end{cases}$$

The mean of the convolution is the sum of the mean of the two Gaussians, and the covariance of the convolution is the sum of their covariances

Predictive distribution (cont.)

Using old results (Eq. 2.115, $\star$), the predictive distribution takes the form

$$p(t|\mathbf{x}, \mathtt{t}, \alpha, \beta) = \mathcal{N}(t|\mathbf{m}_N^T \phi(\mathbf{x}), \sigma_N^2(\mathbf{x})) \tag{11}$$

where the variance $\sigma_N^2(\mathbf{x})$ if the predictive distribution is

$$\sigma_N^2(\mathbf{x}) = \frac{1}{\beta} + \phi(\mathbf{x}) \mathbf{S}_N \phi(\mathbf{x}) \tag{12}$$

- the first term $1/\beta$ represents the noise on the data
- the second term reflects uncertainty associated with **w**

The noise process and the distribution of **w** are independent Gaussians

- their variances are additive

Predictive distribution (cont.)

As more points are observed, the posterior distribution becomes narrower $\star$

- As a consequence, it can be shown that $\sigma_{N+1}^2(\mathbf{x}) \leq \sigma_N^2(\mathbf{x})$

In the limit $N \to \infty$, second term in $\sigma_N^2(\mathbf{x}) = \dfrac{1}{\beta} + \phi(\mathbf{x}) \mathbf{S}_N \phi(\mathbf{x})$ goes to zero

- The variance of the predictive distribution arises solely from the additive noise governed by the parameter β

Predictive distribution (cont.)

Illustration of the predictive distribution for Bayesian linear regression

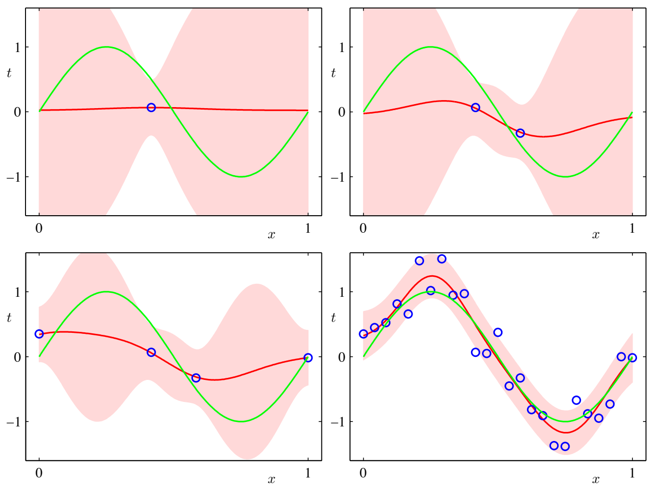
- The sinusoidal data with additive Gaussian noise

Model fitted to data, linear combination of 9 Gaussian basis functions

- Different datasets of different sizes
- $N = 1$, $N = 2$, $N = 4$ and $N = 25$

The red curve (one per N) is the mean of the Gaussian predictive distribution

- The red shaded region spans one standard deviation either side the mean



The predictive uncertainty (the variance) depends on x, it is smallest in the neighbourhood of the points and it decreases as more points are observed

Predictive distribution (cont.)

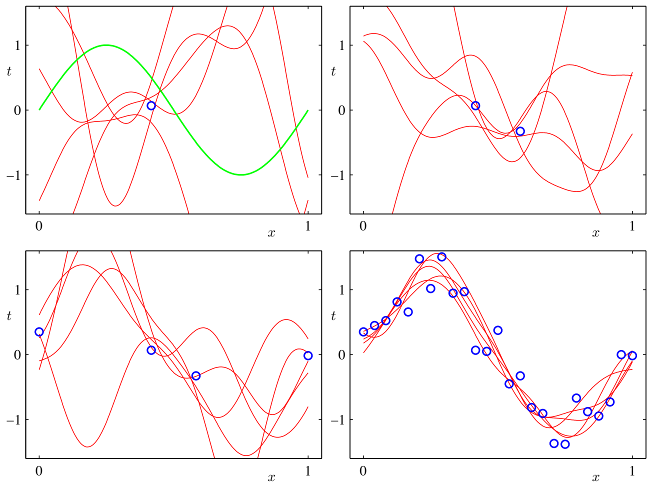
So far, we showed only the point-wise predictive variance as a function of x

In order to gain insight into the covariance between predictions at different values of x, we can draw samples from the posterior distribution over $\mathbf{w}$

- We have a probabilistic model and we can generate new data

Predictive distribution (cont.)

Plots of the functions $y(x, \mathbf{w})$, with sampled $\mathbf{w}$s from the posterior distribution



Predictive distribution (cont.)

If both $\mathbf{w}$ and β are treated as unknowns, we can introduce a conjugate prior distribution $p(\mathbf{w}, \beta)$ which will be given by a Gaussian-gamma distribution

- The resulting predictive distribution is a Student's t-distribution

Equivalent kernel

Bayesian linear regression

Equivalent kernel

The posterior mean solution $\mathbf{m}_N = \beta\mathbf{S}_N\boldsymbol{\Phi}^T\mathsf{t}$ for the linear basis function model has an interesting interpretation that sets the stage for kernel methods

Substituting $\mathbf{m}_N = \beta\mathbf{S}_N\boldsymbol{\Phi}^T\mathsf{t}$ into $y(\mathbf{x},\mathbf{w}) = \sum_{j=0}^{M-1} w_j\phi_j(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x})$, we get

$$y(\mathbf{x},\mathbf{m}_N) = \mathbf{m}_N^T\phi(\mathbf{x}) = \beta\phi(\mathbf{x})^T\mathbf{S}_N\boldsymbol{\Phi}^t\mathsf{t} = \sum_{n=1}^{N}\beta\phi(\mathbf{x})^T\mathbf{S}_N\phi(\mathbf{x}_n)t_n \qquad (13)$$

A new expression for the predictive distribution, where $\mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} - \beta\boldsymbol{\Phi}^T\boldsymbol{\Phi}$

- The mean of the predictive distribution at a point $\mathbf{x}$ is a linear combination of the training set target variables t_n

$$y(\mathbf{x},\mathbf{m}_N) = \sum_{n=1}^{N}\underbrace{\beta\phi(\mathbf{x})^T\mathbf{S}_N\phi(\mathbf{x}_n)}_{k(\mathbf{x},\mathbf{x}_n)}t_n$$

Equivalent kernel (cont.)

The function $k(\mathbf{x},\mathbf{x}‘)$ is known as the **smoother matrix** or **equivalent kernel**

$$k(\mathbf{x},\mathbf{x}‘) = \beta\phi(\mathbf{x})^T\mathbf{S}_N\phi(\mathbf{x}‘) \qquad (14)$$

Regression functions that make predictions by taking linear combinations of the target values t_n in the training set are known as **linear smoothers**

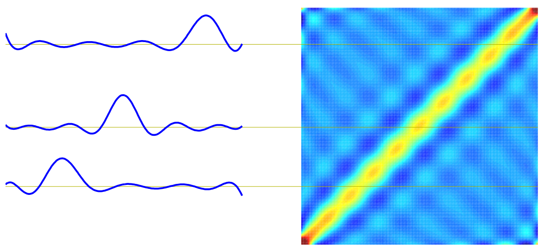
$$y(\mathbf{x},\mathbf{m}_N) = \sum_{n=1}^{N}k(\mathbf{x},\mathbf{x}_n)t_n \qquad (15)$$

The dependence on the input values $\mathbf{x}_n$ in the training set are through $\mathbf{S}_N$

Equivalent kernel (cont.)

The kernel functions $k(x,x')$ are collected in the smoother matrix

They can be plotted as a function of x' for different (3) values of x



Localised around x, so the mean $y(x,\mathbf{m}_N)$ of the predictive distribution at x

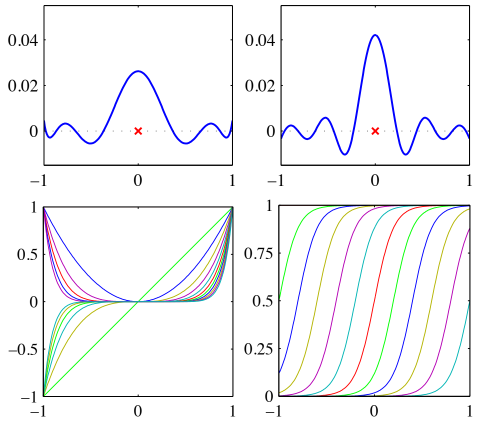
- is a weighted combination of the target values
- points close to x are given higher weight

Intuitively, local evidence is weighted more strongly that distant evidence

Equivalent kernel (cont.)

Examples of equivalent kernels $k(x, x')$ for $x = 0$ plotted as a function of x'

- Polynomial basis functions (left) and sigmoidal basis functions (right)



k is a localised function of x', though the corresponding basis function is not

Equivalent kernel (cont.)

Further insight into the role of the equivalent kernel can be obtained by considering the covariance between $y(\mathbf{x})$ and $y(\mathbf{x}')$, which is given[5] by

$$\begin{aligned} \text{cov}\Big[y(\mathbf{x}), y(\mathbf{x}')\Big] &= \text{cov}\Big[\phi(\mathbf{x})^T \mathbf{w}, \mathbf{w}^T \phi(\mathbf{x}')\Big] \\ &= \phi(\mathbf{x})^T \mathbf{S}_N \phi(\mathbf{x}') \\ &= \beta^{-1} k(\mathbf{x}, \mathbf{x}') \end{aligned} \tag{16}$$

- From the form of the equivalent kernel, we see that the predictive mean at nearby points will be highly correlated, whereas for more distant pairs of points the correlation will be smaller

[5] We used $p(\mathbf{w}|\mathtt{t}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N)$ and $k(\mathbf{x}, \mathbf{x}`) = \beta \phi(\mathbf{x})^T \mathbf{S}_N \phi(\mathbf{x}`)$

Equivalent kernel (cont.)

The formulation of linear regression in terms of a kernel function suggests an alternative approach to regression

Instead of introducing a set of basis functions, which implicitly determines an equivalent kernel, we can instead define a localised kernel directly and use this to make predictions for new input vectors $\mathbf{x}$, given the observed training set

Remark

This leads to a practical framework for regression with Gaussian processes

Equivalent kernel (cont.)

The effective kernel defines the weights by which the training set target values are combined in order to make a prediction at a new value of $\mathbf{x}$

It can be shown that these weights sum to one, in other words

$$\sum_{n=1}^{N} k(\mathbf{x}, \mathbf{x}') = 1, \quad \forall \mathbf{x} \tag{17}$$

It can also be shown that the kernel function can be written

$$k(\mathbf{x}, \mathbf{z}) = \psi(\mathbf{x})^T \psi(\mathbf{z}) \tag{18}$$

This is an inner product with respect to vector $\psi(\mathbf{x})$ of a set of nonlinear functions, with

$$\psi(\mathbf{x}) = \beta^{1/2} \mathbf{S}_N^{1/2} \phi(\mathbf{x})$$

Gaussian processes

Bayesian linear regression

Gaussian processes

We applied the concept of duality to a non-probabilistic model for regression

- We extend the role of kernels to probabilistic discriminative models

We considered linear regression models of the form $y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \phi(\mathbf{x})$

- $\mathbf{w}$ 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 evaluated the posterior distribution over $\mathbf{w}$

- To obtain a corresponding posterior distribution over regression functions

With noise, it implies a predictive distribution $p(t|\mathbf{x})$ for new inputs $\mathbf{x}$

Gaussian processes (cont.)

In the Gaussian process viewpoint, we dispense with the parametric model and instead define a prior probability distribution over functions directly

It is difficult to work with a distribution over the infinite space of functions

- 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

Linear regression revisited

Gaussian processes

Linear regression revisited

To illustrate the Gaussian process viewpoint, we consider linear regression

- We re-derive the predictive distribution
- In terms of distributions over functions $y(\mathbf{x}, \mathbf{w})$

Consider a model defined in terms of a linear combination of M fixed basis functions given by the elements of the vector $\phi(\mathbf{x}) = (\phi_0(\mathbf{x}), \ldots, \phi_{M-1}(\mathbf{x}))^T$

$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) \qquad (19)$$

where $\mathbf{x}$ is the input vector and $\mathbf{w}$ is the M-dimensional weight vector

Linear regression revisited (cont.)

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

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I}) \qquad (20)$$

governed by hyperparameter α, precision (inverse variance) of the distribution

- For any given value of $\mathbf{w}$, $y(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x})$ defines a function of $\mathbf{x}$

The probability distribution over $\mathbf{w}$ induces
a probability distribution over functions $y(\mathbf{x})$

We wish to evaluate this function at specific values of $\mathbf{x}$, say the training data

$$\mathbf{x}_1, \ldots, \mathbf{x}_N$$

Linear regression revisited (cont.)

We are interested in the joint distribution of function values $y(\mathbf{x}_1), \ldots, y(\mathbf{x}_N)$, which we denote by the vector y with elements $y_n = y(\mathbf{x}_n)$, for $n = 1, \ldots, N$

From $y(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x})$, this vector is given by

$$\mathrm{y} = \mathbf{\Phi}\mathbf{w} \qquad (21)$$

$$\mathbf{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix} : \text{Design matrix } (\Phi_{nk} = \phi_k(\mathbf{x}_n))$$

Linear regression revisited (cont.)

We find the probability distribution of y by seeing that y is a linear combo of Gaussian distributed variables, the elements of $\mathbf{w}$ and thus is itself Gaussian

- We need to find its mean and covariance

$$\mathbb{E}[\mathrm{y}] = \mathbf{\Phi}\mathbb{E}[\mathbf{w}] = \mathbf{0} \qquad (22)$$

$$\mathrm{cov}[\mathrm{y}] = \mathbb{E}[\mathrm{y}\mathrm{y}^T] = \mathbf{\Phi}\mathbb{E}[\mathbf{w}\mathbf{w}^T]\mathbf{\Phi}^T = \frac{1}{\alpha}\mathbf{\Phi}\mathbf{\Phi}^T = \mathbf{K} \qquad (23)$$

- where $\mathbf{K}$ is the Gram matrix with elements

$$k_{nm} = k(\mathbf{x}_n, \mathbf{x}_m) = \frac{1}{\alpha}\phi(\mathbf{x}_n)^T\phi(\mathbf{x}_m) \qquad (24)$$

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

Linear regression revisited (cont.)

Remark

- A Gaussian process is 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 $\{\mathbf{x}_n\}$ jointly have a Gaussian distribution

Linear regression revisited (cont.)

A key point about Gaussian processes is that the joint distribution over the N variables $y_1, \dots, y_N$ is specified completely by second-order statistics

- **Mean**: 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

- **Covariance**: The specification of the Gaussian process is completed by giving the covariance of $y(\mathbf{x})$ evaluated at any two values of $\mathbf{x}$

This is given by the kernel function $\mathbb{E}[y(\mathbf{x}_n)y(\mathbf{x})_m] = k(\mathbf{x}_n, \mathbf{x}_m)$

Linear regression revisited (cont.)

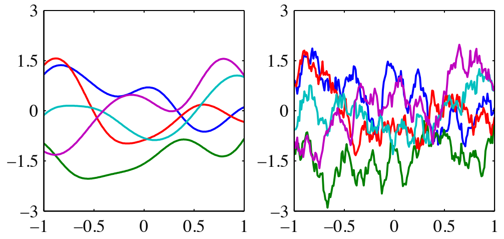
For the specific case of a Gaussian process defined by the linear regression model $y(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x})$ with a weight prior $p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I})$,

- the kernel function is given by $k_{nm} = k(\mathbf{x}_n, \mathbf{x}_m) = \dfrac{1}{\alpha}\phi(\mathbf{x}_n)^T\phi(\mathbf{x}_n)$

Linear regression revisited (cont.)

We can also define the kernel function directly, rather than indirectly

- By-pass the choice of basis functions
- Draw samples of functions from the GP



Gaussian processes for regression
Gaussian processes

Gaussian processes for regression

In order to apply Gaussian process models to the problem of regression, we need to take account of the noise on the observed target values

$$t_n = y_n + \varepsilon_n \tag{25}$$

where $y_n = y(\mathbf{x}_n)$ and ε_n is a random noise variable whose value is chosen independently for each observation n

We consider noise processes that have a Gaussian distribution, so that

$$p(t_n|y_n) = \mathcal{N}(t_n|y_n, \beta^{-1}) \tag{26}$$

where β is a hyperparameter representing for the precision of the noise

Gaussian processes for regression (cont.)

Because the noise is independent for each point, the joint distribution of the target values $\mathtt{t} = (t_1, \dots, t_N)^T$ conditioned on the values of $y = (y_1, \dots, y_N)^T$ is given by an isotropic Gaussian of the form

$$p(\mathtt{t}|\mathtt{y}) = \mathcal{N}(\mathtt{t}|\mathtt{y}, \beta^{-1}\mathbf{I}_N) \tag{27}$$

From the definition of Gaussian process, the marginal distribution $p(\mathtt{y})$ is given by a Gaussian whose mean is zero and whose covariance is a Gram matrix $\mathbf{K}$

$$p(\mathtt{y}) = \mathcal{N}(\mathtt{y}|\mathbf{0}, \mathbf{K}) \tag{28}$$

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

Gaussian processes for regression (cont.)

In order to find the marginal distribution $p(\mathtt{t})$, conditioned on the input values $\mathbf{x}_1, \dots, \mathbf{x}_N$, we need to integrate $p(\mathtt{t}|\mathtt{y})$ over $\mathtt{y}$

$$p(\mathtt{t}) = \int p(\mathtt{t}|\mathtt{y})p(\mathtt{y})d\mathtt{y} = \mathcal{N}(\mathtt{t}|\mathbf{0}, \mathbf{C}) \tag{29}$$

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

$$C(\mathbf{x}_n, \mathbf{x}_m) = k(\mathbf{x}_n, \mathbf{x}_m) + \beta^{-1}\delta_{nm} \tag{30}$$

- δ_{nm} is a Kronecker delta (1 iff $n = m$, 0 otherwise)

The covariance matrix $\mathbf{C}$ reflects the fact that the two Gaussian sources of randomness (one associated with $y(\mathbf{x})$ and one to ε) are independent

- their covariances ($\mathbf{K}$ and $\beta^{-1}\mathbf{I}$) simply add

Gaussian processes for regression (cont.)

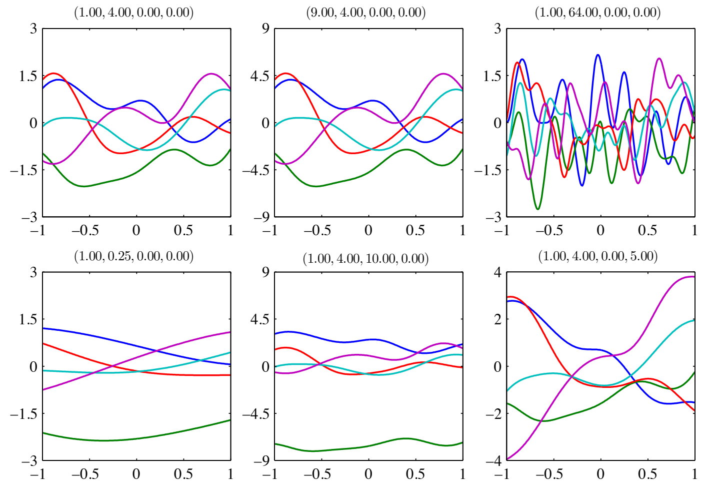
One widely used kernel function for Gaussian processes is the exponential of a quadratic form with the addition of constant and linear terms to give

$$k(\mathbf{x}_n, \mathbf{x}_m) = \theta_0 \exp\Big(-\frac{\theta_1}{2}||\mathbf{x}_n - \mathbf{x}_m||^2\Big) + \theta_2 + \theta_3 \mathbf{x}_n^T \mathbf{x}_m \qquad (31)$$

The term involving θ_3 corresponds to a parametric model that is a linear function of the input variables.

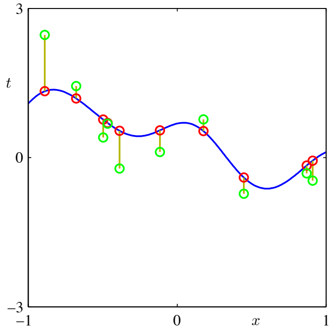
Gaussian processes for regression (cont.)

From a GP prior with covariance function $k(\mathbf{x}_n, \mathbf{x}_m)$, we can sample functions



Gaussian processes for regression (cont.)

A sampled function (blue line) drawn from the Gaussian process prior over functions is evaluated at a set of points $\{x_n\}$ to give points $\{y_n\}$ (red dots)



The corresponding values of $\{t_n\}$ (green dots) are obtained by adding independent Gaussian noise to each point in $\{y_n\}$

Gaussian processes for regression (cont.)

Our goal in regression is to make predictions of target variables for new inputs

- given a set of training data

Let us suppose that $\mathbf{t}_N = (t_1, \ldots, t_N)^T$ for input values $\mathbf{x}_1, \ldots, \mathbf{x}_N$, comprise the observed training data set and our goal is to predict the variable t_{N+1}

- for a new input vector $\mathbf{x}_{N+1}$

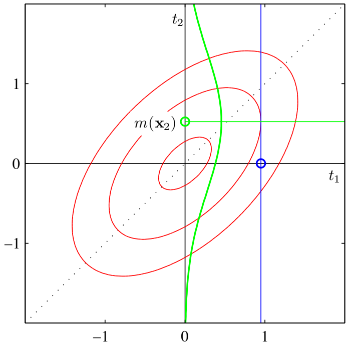
This requires that we evaluate the predictive distribution $p(t_{N+1}|\mathbf{t}_N)$

- This distribution is conditioned also on the variables $\mathbf{x}_1, \ldots, \mathbf{x}_N$ and $\mathbf{x}_{N+1}$

Gaussian processes for regression (cont.)

To find the conditional distribution $p(t_{N+1}|\mathtt{t}_N)$, we begin by writing down the joint distribution $p(\mathtt{t}_{N+1})$, $\mathtt{t}_{N+1}$ is the vector $\mathtt{t}_{N+1} = (t_1, \dots, t_N, t_{N+1})^T$

We then apply results from the Gaussian distribution to obtain the conditional



One training t_1 and one test point t_2

- Contours of the joint distribution $p(t_1, t_2)$

We condition on (fix) the value of t_1

- We obtain $p(t_2|t_1)$

The conditional distribution $p(t_{N+1}|\mathtt{t})$ will also be a Gaussian distribution

Gaussian processes for regression (cont.)

From $p(\mathtt{t}) = \int p(\mathtt{t}|\mathtt{y})p(\mathtt{y})d\mathtt{y}$, the joint distribution over $t_1, \dots, t_{N+1}$ is

$$p(\mathtt{t}_{N+1}) = \mathcal{N}(\mathtt{t}_{N+1}|\mathbf{0}, \mathbf{C}_{N+1}) \tag{32}$$

where $\mathbf{C}_{N+1}$ is a $(N+1) \times (N+1)$ covariance matrix with elements given by

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

Because this joint distribution is Gaussian, we can apply the results from the Gaussian distribution to characterise this conditional Gaussian distribution

Gaussian processes for regression (cont.)

Remembering that when two sets of variables are jointly Gaussian also the conditional distribution of one set conditioned is Gaussian

For an arbitrary vector $\mathbf{x}$ with Gaussian distribution $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$

- We first partitioned $\mathbf{x}$ into two disjoint subsets $\mathbf{x}_a$ and $\mathbf{x}_b$

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{pmatrix}$$

- We partitioned mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{pmatrix} \qquad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{pmatrix}$$

From these, we obtained the expressions for the mean and covariance of the conditional distribution $p(\mathbf{x}_a|\mathbf{x}_b)$ in terms of the partitioned covariance matrix

$$\begin{aligned} \boldsymbol{\mu}_{a|b} &= \boldsymbol{\mu}_a + \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}(\mathbf{x}_b - \boldsymbol{\mu}_b) \\ \boldsymbol{\Sigma}_{a|b} &= \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba} \end{aligned}$$

Gaussian processes for regression

We first partition the covariance matrix of the joint distribution

$$\mathbf{C}_{N+1} = \begin{pmatrix} \mathbf{C}_N & \mathbf{k} \\ \mathbf{k}^T & c \end{pmatrix} \tag{33}$$

- $\mathbf{C}_N$ is a $N \times N$ covariance matrix with elements given by

$$C(\mathbf{x}_n, \mathbf{x}_m) = k(\mathbf{x}_n, \mathbf{x}_m) + \beta^{-1}\delta_{nm}, \quad \text{with } n, m = 1, \dots, N$$

- Vector $\mathbf{k}$ has elements $k(\mathbf{x}_n, \mathbf{x}_{N+1})$ for $n = 1, \dots, N$
- Scalar $c = k(\mathbf{x}_{N+1}, \mathbf{x}_{N+1}) + \beta^{-1}$

Gaussian processes for regression (cont.)

Using expressions from the conditional Gaussian distribution on $p(t_n|\mathtt{t})$ yields

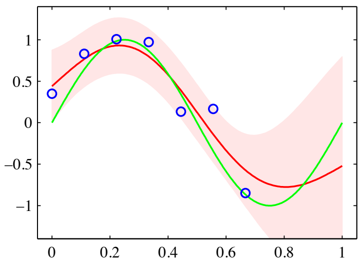
$$m(\mathbf{x}_{N+1}) = \mathbf{k}^T\mathbf{C}_N^{-1}\mathtt{t} \quad (34)$$

$$\sigma^2(\mathbf{x}_{N+1}) = c - \mathbf{k}^T\mathbf{C}_N^{-1}\mathbf{k} \quad (35)$$

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

Gaussian processes for regression (cont.)

Illustration of Gaussian process regression applied to the sinusoidal data set



Three right-most points were omitted

Green curve: The sine function from which data points (blue) are obtained by sampling and adding Gaussian noise

Red line: The mean of the Gaussian process predictive distribution

- ± two standard deviations

Note how the uncertainty increases in the region to the right of the data points

Gaussian processes for regression (cont.)

The only restriction on the kernel function is that the covariance matrix $C(\mathbf{x}_n, \mathbf{x}_m) = k(\mathbf{x}_n, \mathbf{x}_m) + \beta^{-1}\delta_{nm}$ must be positive definite

- If λ_i is an eigenvalue of $\mathbf{K}$, then the associated eigenvalue of $\mathbf{C}$ will be $\lambda_i + \beta^{-1}$

It is therefore sufficient that the kernel matrix $k(\mathbf{x}_n, \mathbf{x}_m)$ be positive semidefinite for any pair of points $\mathbf{x}_n$ and $\mathbf{x}_m$, so that $\lambda_i \geq 0$

- any eigenvalue λ_i that is zero will still give rise to a positive eigenvalue for $\mathbf{C}$ because $\beta > 0$

This is the same restriction on the kernel function discussed earlier

- We can exploit all of the techniques to construct suitable kernels

Gaussian processes for regression (cont.)

Mean $m(\mathbf{x}_{N+1}) = \mathbf{k}^T\mathbf{C}_N^{-1}\mathbf{t}$ of the predictive distribution is a function of $\mathbf{x}_{N+1}$

$$m(\mathbf{x}_{N+1}) = \sum_{n=1}^{N} a_n k(\mathbf{x}_N, \mathbf{x}_{N+1}), \quad a_n \text{ is the } n\text{-th component of } \mathbf{C}_N^{-1}\mathtt{t} \quad (36)$$

For a kernel function $k(\mathbf{x}_n, \mathbf{x}_m)$ depends only on the distance $||\mathbf{x_n} - \mathbf{x}_m||$, we obtain an expansion in radial basis functions

Gaussian processes for regression (cont.)

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

The results above define the predictive distribution for Gaussian process for regression with an arbitrary kernel function $k(\mathbf{x}_n, \mathbf{x}_m)$

In the particular case in which the kernel function $k(\mathbf{x}_n, \mathbf{x}_m)$ is defined in terms of a finite set of basis functions, we can derive the results for linear regression

- from a Gaussian process view point ($\star$)

For such models, we can therefore obtain the predictive distribution either

- by taking a parameter space viewpoint and using linear regression results
- by taking a function space viewpoint and using the GP model result

Gaussian processes for regression (cont.)

The central computational operation in GP is the inversion of a $N \times N$ matrix

- Standard methods require $\mathcal{O}(N^3)$ computations

In the basis function model, we have to invert a matrix $\mathbf{S}_N$ of size $M \times M$

- $\mathcal{O}(M^3)$ computational complexity

For both, matrix inversion must be performed once for the given training set

Remark

For each new test point, both require a vector-matrix multiply, which has cost $\mathcal{O}(N^2)$ for Gaussian process models and $\mathcal{O}(M^2)$ for linear basis models

If the number M of basis functions is smaller than the number N of points, it is computationally more efficient to work in the basis function framework

Learning the hyper-parameters

Gaussian processes

Learning the hyper-parameters

Predictions of a GP model depends partly on the choice of covariance function

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 data

These parameters govern such things as length scale of correlations and the precision of noise, they are hyper-parameters in a standard parametric model

Learning the hyper-parameters (cont.)

Remark

Techniques for learning the hyper-parameters are based on the evaluation of the likelihood function $p(\mathtt{t}|\boldsymbol{\Theta})$ where $\boldsymbol{\Theta}$ are the hyper-parameters of the GP

The simplest approach is to make a point estimate of $\boldsymbol{\Theta}$ by maximising the log likelihood function (e.g., by gradient-based optimisation algorithms as CG)

Learning the hyper-parameters (cont.)

The log likelihood function for a Gaussian process regression model is evaluated using the standard form for a multivariate Gaussian distro

$$\ln p(\mathtt{t}|\boldsymbol{\Theta}) = -\frac{1}{2}\mathrm{Tr}|\mathbf{C}_N| - \frac{1}{2}\mathtt{t}^T\mathbf{C}_N^{-1}\mathtt{t} - \frac{N}{2}\ln(2\pi) \qquad (37)$$

For nonlinear optimisation, we also need the gradient of the log likelihood function with respect to the parameter vector $\boldsymbol{\Theta}$

$$\frac{\partial \ln p(\mathtt{t}|\boldsymbol{\Theta})}{\partial \theta_i} = -\frac{1}{2}\mathrm{Tr}\Big(\mathbf{C}_N^{-1}\frac{\partial \mathbf{C}_N}{\partial \theta_i}\Big) + \frac{1}{2}\mathtt{t}^T\mathbf{C}_N^{-1}\frac{\partial \mathbf{C}_N}{\partial \theta_i}\mathbf{C}_N^{-1}\mathtt{t} \qquad (38)$$

In general, $\ln p(\mathtt{t}|\boldsymbol{\Theta})$ is a non-convex function, it might have multiple maxima

Learning the hyper-parameters (cont.)

Optionally, we could introduce a prior over $\boldsymbol{\Theta}$ and maximise the log posterior

- Again, by using gradient-based methods

Remark

In a fully Bayesian treatment, we would need to evaluate marginals over $\boldsymbol{\Theta}$ weighted by the product of the prior $p(\boldsymbol{\Theta})$ and the likelihood function $p(\mathtt{t}|\boldsymbol{\Theta})$

- In general, however, exact marginalisation will be intractable
- We must resort to approximations

Learning the hyper-parameters (cont.)

The Gaussian process regression model gives a predictive distribution whose mean and variance are functions of the input vector $\mathbf{x}$

However, we have assumed that the contribution to the predictive variance arising from the additive noise, governed by the parameter β, is a constant

For hetero-scedastic problems, the noise variance itself will also depend on $\mathbf{x}$

To model this, we can extend the Gaussian process framework by introducing a second Gaussian process to represent the dependence of β on the input $\mathbf{x}$