STA 414/2104

Statistical Methods for Machine Learning and Data Mining

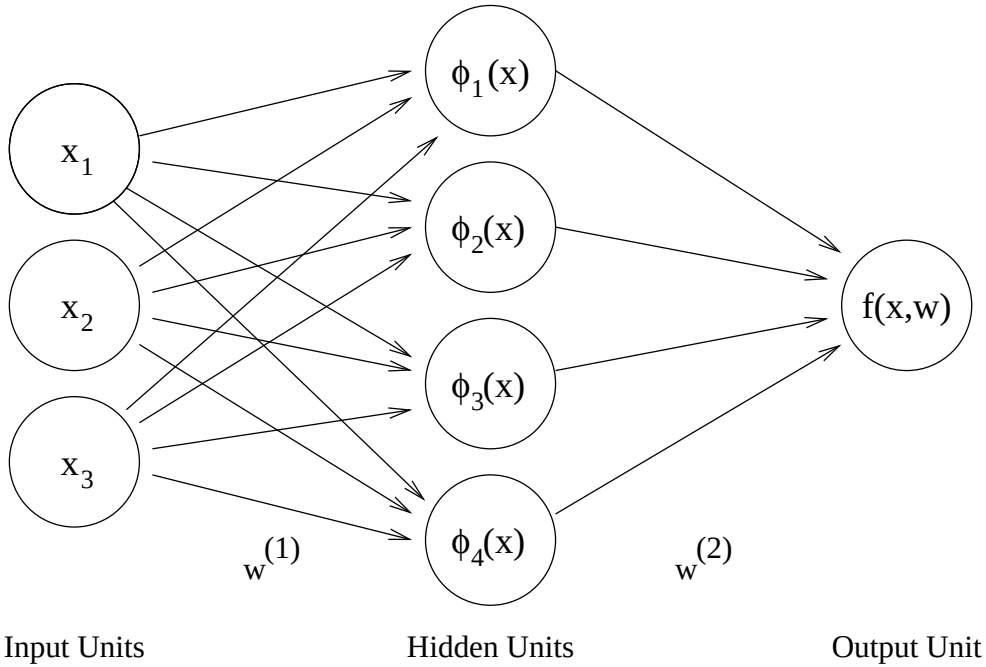
Radford M. Neal, University of Toronto, 2012

Week 7

Bayesian Neural Networks

A Prior Distribution for Network Weights

Recall the architecture of a multilayer perceptron network with one hidden layer:



$$f(x,w) \;=\; w_0^{(2)} \;+\; \sum_{j=1}^{m} w_j^{(2)} \phi_j(x,w), \qquad \phi_j(x,w) \;=\; h\Big(w_{0j}^{(1)} \;+\; \sum_{k=1}^{p} w_{kj}^{(1)} x_k\Big)$$

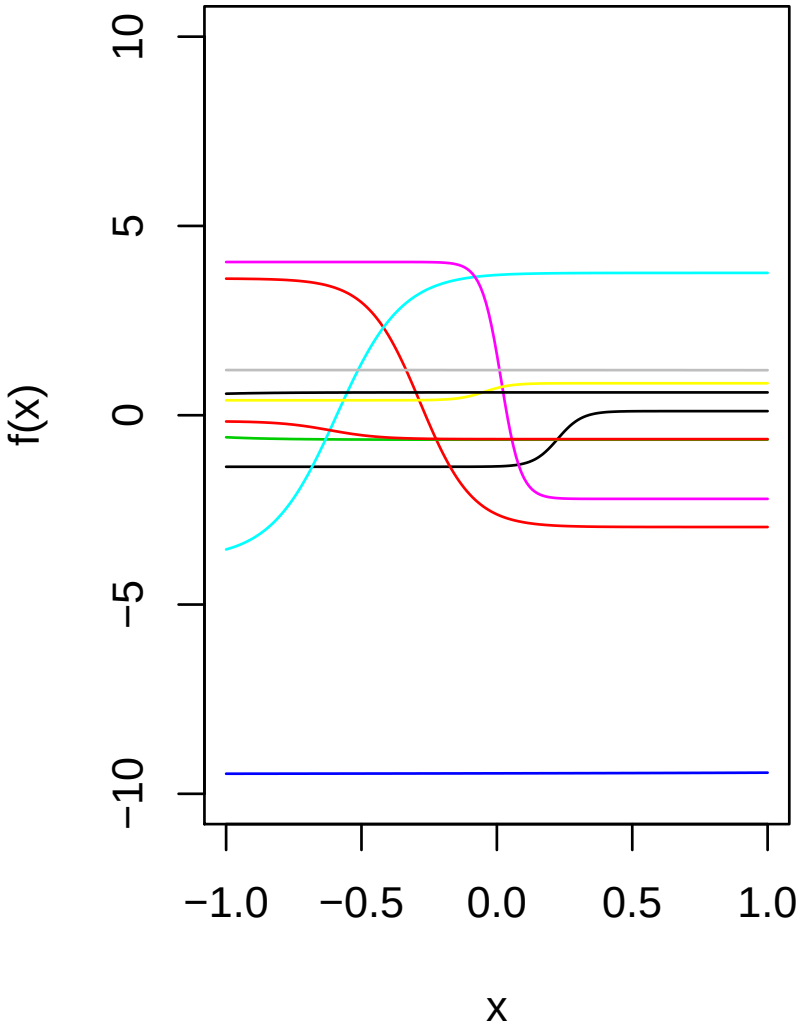
Rather than find parameters w by maximization, we can use a Bayesian method, in which we give a prior distribution to w. One possibility is independent normal priors, as follows:

$$w_{0j}^{(1)} \sim N(0, \sigma_0^{(1)}), \quad w_{kj}^{(1)} \sim N(0, \sigma^{(1)}), \quad w_0^{(2)} \sim N(0, \sigma_0^{(2)}), \quad w_j^{(2)} \sim N(0, \sigma^{(2)})$$

Samples From Priors Over Functions Defined by This Prior

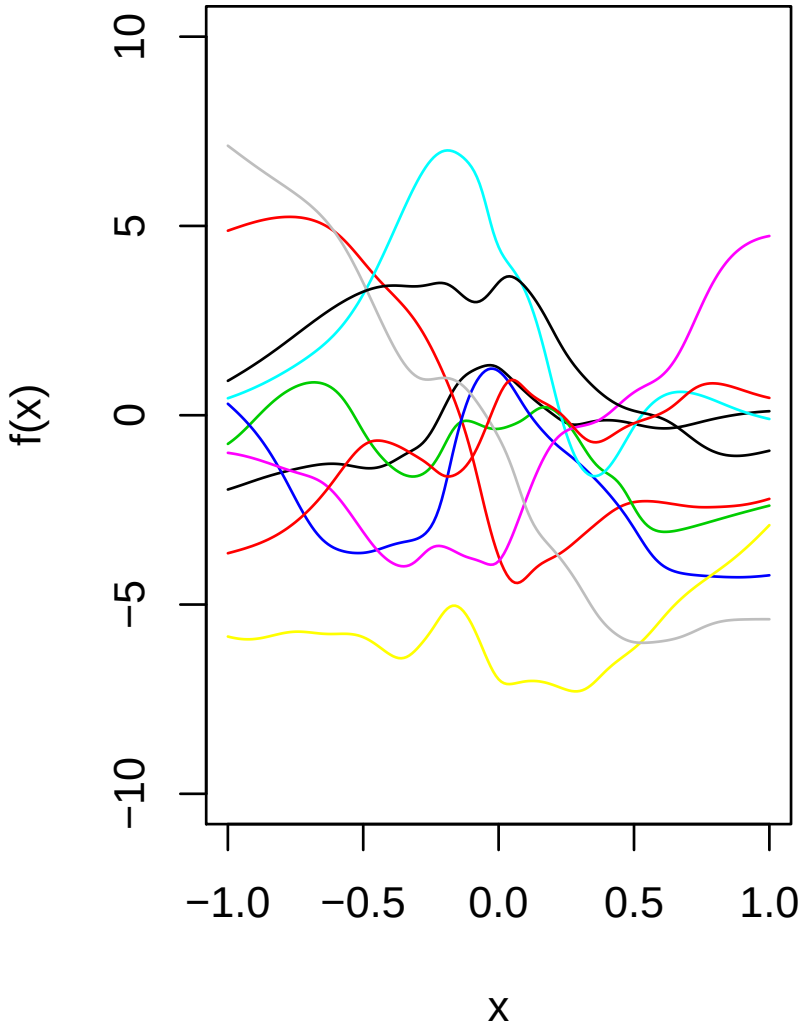
$m = 1$

$\sigma_0^{(1)} = 3,\ \sigma^{(1)} = 7,\ \sigma_0^{(2)} = 1,\ \sigma^{(2)} = 4$



$m = 100$

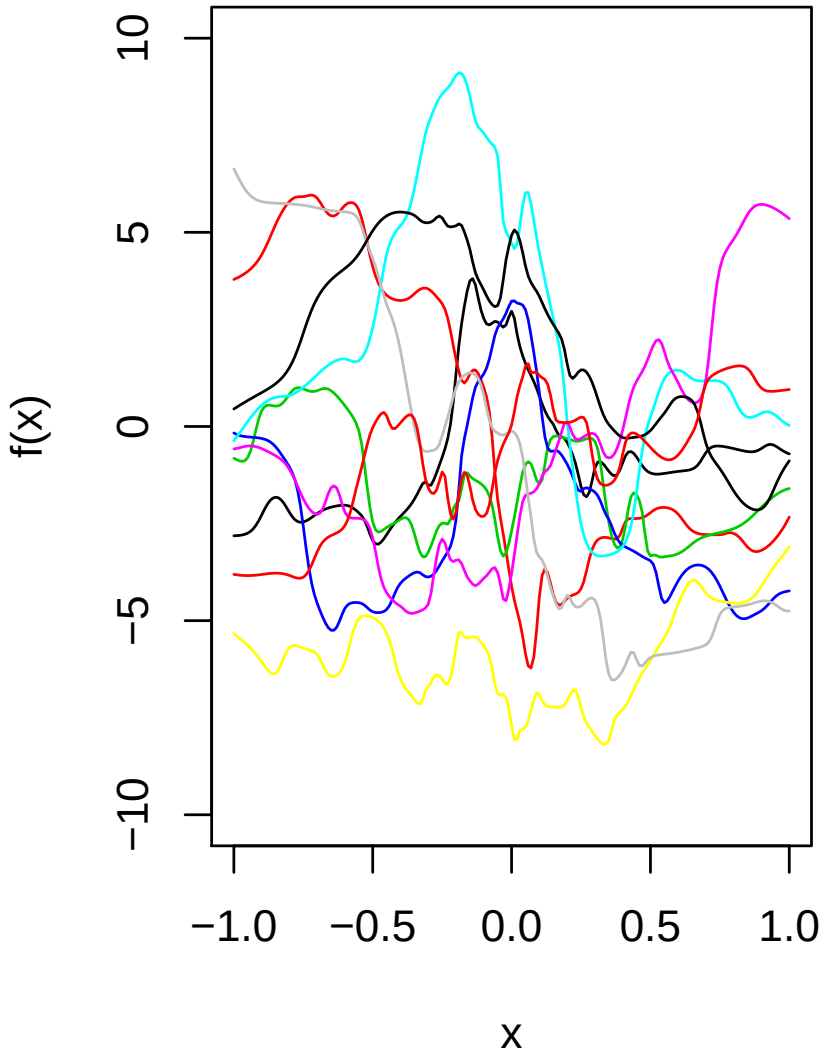
$\sigma_0^{(1)} = 3,\ \sigma^{(1)} = 7,\ \sigma_0^{(2)} = 1,\ \sigma^{(2)} = 4/\sqrt{m}$



More Samples From Priors Over Functions ($m = 100$)

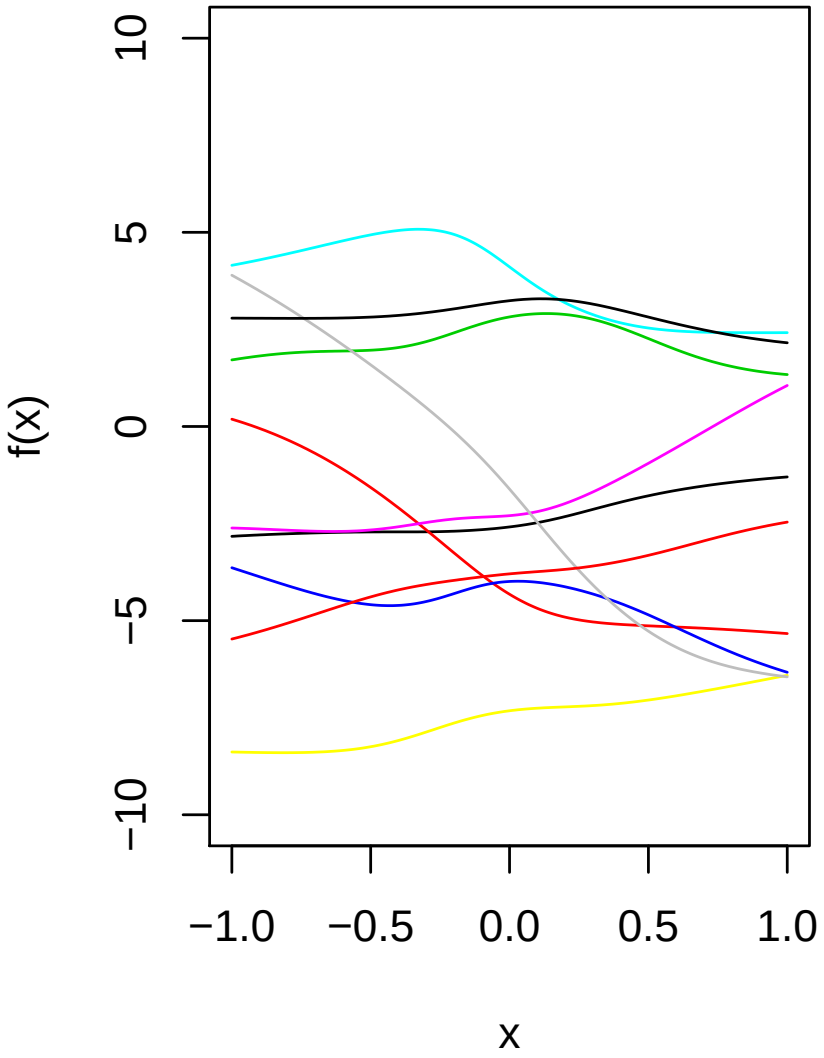
$\sigma_0^{(1)} = 3 \times 5$, $\sigma^{(1)} = 7 \times 5$

$\sigma_0^{(2)} = 1$, $\sigma^{(2)} = 4/\sqrt{m}$



$\sigma_0^{(1)} = 3/5$, $\sigma^{(1)} = 7/5$

$\sigma_0^{(2)} = 4$, $\sigma^{(2)} = 4/\sqrt{m}$



Properties of this Prior for a Multilayer Perceptron Network

- The standard deviation for $w^{(1)}$ determines how "wiggly" the function is, when m is large.
- The standard deviation for $w_0^{(1)}$ determines the range over which such wiggles occur.
- The standard deviation for $w^{(2)}$ determines the scale variation in the function.
- To keep this variation the same as we increase m, we need to decrease the standard deviation for $w^{(2)}$ by the factor $\sqrt{m}$.
- The standard deviation for $w_0^{(2)}$ determines how large a vertical offset the function might have.
- When m is large, the Central Limit Theorem tells us that the prior distribution for $f(x)$, with x some fixed input, approaches a Gaussian distribution.
- In fact, the joint distribution for $f(x_1)$, $f(x_2)$, $f(x_3)$, etc. approaches a multivariate Gaussian distribution, for any set of inputs x_1, x_2, x_3, ...

OK, Nice Prior, But What About the Posterior...?

To actually use a Bayesian neural network, we have to make predictions for test cases based on the posterior distribution.

This requires sophisticated Markov chain Monte Carlo methods, which work quite well, but are beyond the scope of this course.

But the way that the prior over values of the function approaches a multivariate Gaussian as $m \to \infty$ suggests an alternative approach, which we'll look at next...

Gaussian Process Models

Bayesian Linear Basis Function Model

Recall the linear basis function model, which we can write as

$$y \sim N(\Phi\beta, \sigma^2 I)$$

where here,

- y is the vector of observed targets
- β is the vector of regression coefficients
- σ^2 is the "noise" variance
- Φ is the matrix of basis function values in the training cases.

Suppose that our prior for β is $N(0, S_0)$. This is a conjugate prior, with the posterior for β also being normal.

For the moment, we regard σ^2 and S_0 as known.

Prior Distribution of Responses for a Linear Basis Function Model with Gaussian Noise and Gaussian Prior

When m, the number of basis functions, is greater than n, the number of observations in our training set, it is computationally attractive to shift focus from the parameters β_j for $j = 0, \ldots, m-1$ (collectively written β) to the observed responses, y_i for $i = 1, \ldots, n$ (collectively written y).

We need to find the prior distribution of y implied by the prior distribution of β.

If the prior distribution of β is Gaussian, the prior of y will also be Gaussian, since $y = \Phi\beta + e$ is a linear function of jointly Gaussian variables.

If the prior for β has mean zero, so will the prior for y.

If the prior covariance of β is S_0, the prior covariance of y will be $\sigma^2 I + \Phi S_0 \Phi^T$. If the β_j are independent in the prior, with the variance of β_j being ω_j^2, then

$$\text{Cov}(y_i, y_{i'}) \;=\; \sigma^2 \delta_{i,i'} \;+\; \sum_{j=0}^{m-1} \omega_j^2 \phi_j(x_i) \phi_j(x_{i'})$$

where $\delta_{i,i'} = 1$ if $i = i'$ and zero otherwise.

Predicting Directly Using the Prior for Responses

In similar fashion, we can find the prior covariance between responses in any two cases, whether they be training cases or future test cases.

Let C be the $n \times n$ covariance matrix of all the responses, $y_1, \ldots, y_n$, in the training set. For some test case with input x_*, let k be the vector of covariances of the response for the test case, y_*, with the responses for training cases. Finally, let v be the variance of the test response (covariance of y_* with itself).

As before, we assume prior means of zero (from a prior mean of zero for β).

We can now make predictions directly, without further reference to the β parameters, by finding the predictive density

$$P(y_* \mid y_1, \ldots, y_n)$$

Since conditional distributions from multivariate Gaussians are Gaussian, this predictive distribution will be Gaussian, fully specified by mean and variance.

Applying the general formulas for Gaussian conditional distributions, we get

$$E(y_* \mid y_1, \ldots, y_n) \;=\; k^T C^{-1} y, \qquad \mathrm{Var}(y_* \mid y_1, \ldots, y_n) \;=\; v \;-\; k^T C^{-1} k$$

This takes $O(n^3 + n^2 m)$ time to compute, versus $O(m^3 + nm^2)$ for previous method.

Marginal Likelihood Directly from the Prior for the Responses

When σ, ω, and perhaps some parameters of the ϕ functions are not known, we may wish to estimate or sample them based on the marginal likelihood given the observed responses, y.

We saw how to do this before, working with the posterior distribution of β, in $O(m^3+nm^2)$ time.

Working directly with the covariances of the responses, the marginal likelihood is just the Gaussian prior probability density for the responses. So the log marginal likelihood is

$$-\frac{n}{2}\log(2\pi) \;-\; \frac{1}{2}\log(|C|) \;-\; \frac{1}{2}y^T C^{-1} y$$

This takes $O(n^3+n^2m)$ time to compute.

For both prediction and marginal likelihood, which method is faster depends on the relative magnitudes of n and m. When m is sufficiently bigger than n, it's better to work directly with the responses, integrating away β.

Letting the Number of Basis Functions go to Infinity

When working directly with the responses, the basis functions and the prior for the β_j are used only to find the covariance between the responses in two cases, which we can write as

$$\text{Cov}(y_i, y_{i'}) \quad = \quad \sigma^2 \delta_{i,i'} \; + \; K(x_i, x_{i'})$$

where K is the noise-free covariance function:

$$K(x, x') \quad = \quad \sum_{j=0}^{m-1} \omega_j^2 \phi_j(x) \phi_j(x')$$

If our choice of ω_j and ϕ_j for $j = 1, 2, 3, \ldots$ is such that the sum above reaches a finite limit as $m \to \infty$, the model with infinite m makes sense.

If there's a formula to compute this infinite sum, we can implement this model with infinite m. If time to compute $K(x, x')$ is linear in the number of inputs, p, computing the marginal likelihood or a prediction will take $O(n^3 + n^2 p)$ time.

[When predicting for many test cases, each additional test case takes $O(np)$ time for just the predictive mean, and $O(n^2 + np)$ time if we also want the variance.]

An Infinite Basis Function Model with Sines and Cosines

With one input, let's use as basis functions $\phi_0(x) = 1$, and for $h = 1, 2, 3, \ldots$

$$\phi_{2h-1}(x) \;=\; \sin(f_h x), \qquad \phi_{2h}(x) \;=\; \cos(f_h x)$$

where each f_h is independently drawn from the $N(0, \rho^2)$ distribution.

For $j = 1, \ldots, m-1$, we'll let

$$\omega_j^2 \;=\; \frac{\eta^2}{(m-1)/2}$$

We now look at the limit as $m \to \infty$ of

$$\begin{aligned}
K(x, x') &= \omega_0^2 + \sum_{j=1}^{m-1} \omega_j^2 \phi_j(x) \phi_j(x') \\
&= \omega_0^2 + \sum_{h=1}^{(m-1)/2} \frac{\eta^2}{(m-1)/2} \Big[\sin(f_h x) \sin(f_h x') + \cos(f_h x) \cos(f_h x') \Big] \\
&= \omega_0^2 + \eta^2 \frac{1}{(m-1)/2} \sum_{h=1}^{(m-1)/2} \Big[\sin(f_h x) \sin(f_h x') + \cos(f_h x) \cos(f_h x') \Big]
\end{aligned}$$

The average of $(m-1)/2$ terms above approaches an integral as $m \to \infty$.

Covariance Function for the Model with Sines and Cosines

We can now find the covariance function as $m \to \infty$:

$$
\begin{aligned}
K(x,x') &= \omega_0^2 + \eta^2 \frac{1}{(M-1)/2} \sum_{h=1}^{(m-1)/2} \Big[\sin(f_h x)\sin(f_h x') + \cos(f_h x)\cos(f_h x') \Big] \\
&\to \omega_0^2 + \eta^2 \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}\rho} \exp\Big(-\frac{f^2}{2\rho^2}\Big) \Big[\sin(fx)\sin(fx') + \cos(fx)\cos(fx') \Big] \, df \\
&= \omega_0^2 + \eta^2 \frac{1}{\sqrt{2\pi}\rho} \int_{-\infty}^{+\infty} \exp\Big(-\frac{f^2}{2\rho^2}\Big) \cos(f(x-x')) \, df \\
&= \omega_0^2 + \eta^2 \frac{1}{\sqrt{2\pi}\rho} \Big[\sqrt{2\pi}\rho \, \exp(-\rho^2(x-x')^2/2) \Big] \\
&= \omega_0^2 + \eta^2 \exp(-\rho^2(x-x')^2/2)
\end{aligned}
$$

This is simple to compute, so it's easy to use the model with infinite m.

From Linear Basis Function Models to Gaussian Processes

We see that a linear basis function model with a Gaussian prior for the coefficients defines a Gaussian prior distribution for any set of observed or unobserved responses.

If we fix all means to zero, this Gaussian prior distribution is determined by the covariances between responses, which we saw could sometimes be computed even when the number of basis functions is infinite.

But why bother?

We can just *start* with a function that defines the covariances between responses. As long this function always produces positive-definite covariance matrices, we can use it to infer unobserved responses from observed responses.

[Actually, we might bother with the original basis functions approach if that was the simplest way of expressing our prior beliefs, but often the covariances themselves have more intuitive meaning.]

Defining a Gaussian Process Model

We'll model the response y_i associated with covariate vector x_i as being equal to some function, f, of x_i plus Gaussian noise: $y_i = f(x_i) + e_i$, with $e_i \sim N(0, \sigma^2)$.

We can define the "noise-free" covariances in terms of a function $K(x, x')$, as:

$$\text{Cov}(f(x_i), f(x_{i'})) = K(x_i, x_{i'})$$

I'll always assume that the prior means of all the y_i are zero, so this is enough to specify a multivariate Gaussian distribution for the value of the function at any set of x_i's.

Since this gives us a prior for $f(x)$ at an arbitrarily large set of x's, it effectively gives us a prior for the function f itself.

The covariances for actual (noisy) observations will be

$$\text{Cov}(y_i, y_{i'}) = \sigma^2 \delta_{i,i'} + K(x_i, x_{i'})$$

The Covariance Function

We can choose the noise-free covariance function, $K(x, x')$, to be anything we want, **provided** that it produces positive definite (or at least positive semi-definite) covariance matrices for the y_i's with any allowed set of x_i's.

It's not easy to determine whether some arbitrary $K(x, x')$ will produce positive definite covariance matrices. But there are some well-known classes of valid covariance functions.

Furthermore, the *sum* of two valid covariance functions, K_1 and K_2, is also a valid covariance function. It can be seen as the covariance function for the sum of two functions, one drawn from the Gaussian process with covariance K_1 and the other drawn independently from the Gaussian process with covariance K_2.

The *product* of two covariance functions is also a valid covariance function (though this isn't so obvious).

Constant and Linear Covariance Functions

The *constant* covariance function:

$$K(x, x') \quad = \quad \sigma_0^2$$

can be derived from a model in which the function is an unknown constant: $f(x) = \mu$, with $\mu \sim N(0, \sigma_0^2)$.

The *linear* covariance function:

$$K(x, x') \quad = \quad \sigma_1^2 \, x \, x'$$

comes from a simple linear regression model, $f(x) = \beta x$ with $\beta \sim N(0, \sigma_1^2)$.

If we have two covariates, we can add linear covariances based on each, plus a constant covariance, to get

$$K(x, x') \quad = \quad \sigma_0^2 + \sigma_1^2 \, x_1 \, x_1' + \sigma_2^2 \, x_2 \, x_2'$$

(Note that subscripts on x here select covariates, not cases.)

Stationary Covariance Functions

A *stationary* covariance function can be written as $K(x, x') = K(x - x')$. It is translationally invariant, since only the difference $x - x'$ matters.

Typically, the covariance goes down with increasing distance of x and x'. One class of valid covariance functions of this form is

$$K(x, x') \quad = \quad \gamma^2 \exp(-\rho^2 ||x - x'||^r)$$

where $||x||$ is the Euclidean norm. This is valid for any $r \in (0, 2]$. It's clearly rotationally symmetric.

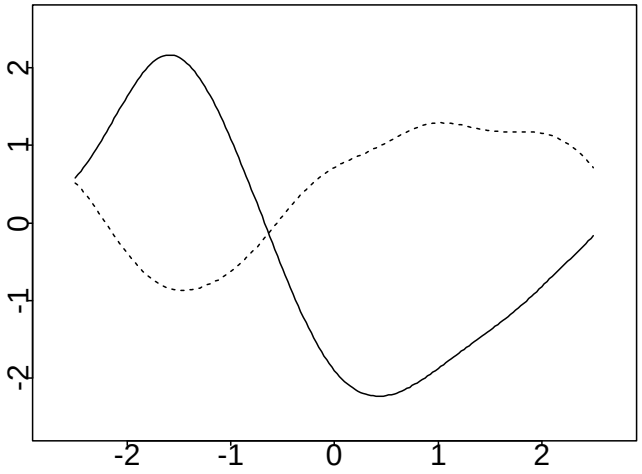
One can compare the above to the following:

$$K(x, x') \quad = \quad \gamma^2 \exp\left(-\rho^2 \sum_{j=1}^{p} |x_j - x'_j|^r\right)$$

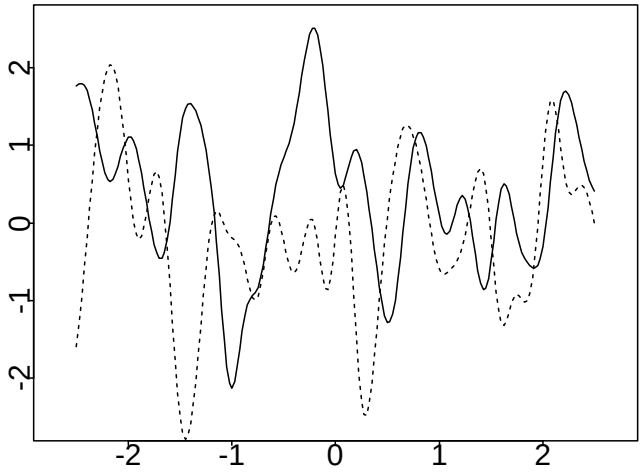
They're the same when $p = 1$ or $r = 2$, but they otherwise are different.

We can see that the second form is valid since it's a product of covariance functions of the first form.

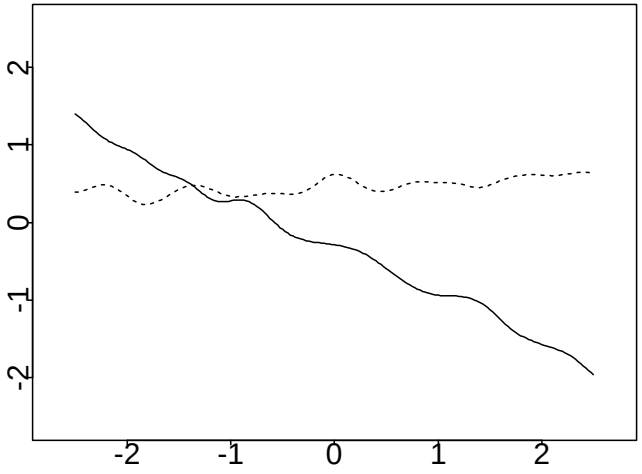
Functions From Some Gaussian Process Priors



$\exp\left(-(x-x')^2\right)$

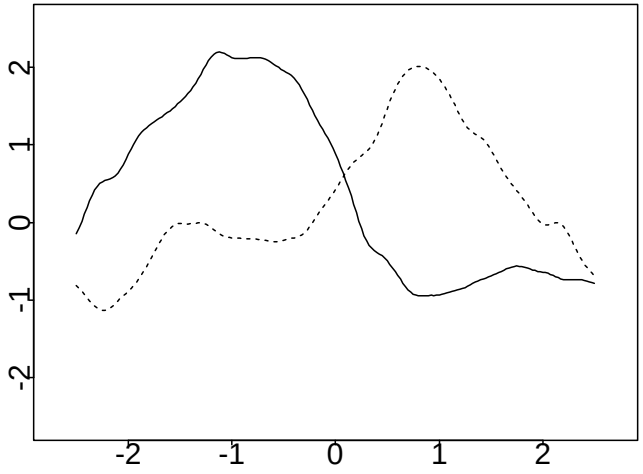


$\exp\left(-5^2(x-x')^2\right)$



$1 + xx'$

$+ 0.1^2 \exp\left(-3^2(x-x')^2\right)$



$\exp\left(-(x-x')^2\right)$

$+ 0.1^2 \exp\left(-5^2(x-x')^2\right)$

Predictions with Gaussian Process Models

If we know the covariance function, and the noise variance, predictions for test cases with a Gaussian process model can be done with straightforward matrix operations.

As we saw before, if y is the vector of responses in the n training cases, and y_* is the response for a test case, the conditional distribution of y_* given y will be Gaussian, with mean and variance given by

$$E(y_* \,|\, y_1, \ldots, y_n) \;=\; k^T C^{-1} y, \qquad \mathrm{Var}(y_* \,|\, y_1, \ldots, y_n) \;=\; v \;-\; k^T C^{-1} k$$

Here C is the $n \times n$ covariance matrix of the responses, in the training set, k is the vector of covariances of the response for the test case with the responses for training cases, and v is the variance of the test response (covariance of y_* with itself).

If $K(x, x')$ takes $O(p)$ time to compute, then C^{-1} will take $O(pn^2 + n^3)$ time to compute, after which a prediction for each test case takes $O(np)$ time for just the mean, plus $O(n^2)$ time if the variance is also required.

When the Covariance Function Isn't Known

In practice, the covariance function usually has some unknown parameters — such as the scale parameters γ and ρ in the exponential covariance function. The noise variance is also typically not known.

The covariance matrix of responses, C, needed for prediction, will depend on these unknown parameters.

One could find the maximum likelihood estimates for the unknown parameters, and then use these single values for prediction.

The full Bayesian approach is to average predictions over the posterior distribution for the unknown parameters, probably using Markov chain Monte Carlo methods.