Gaussian process covariance functions

Carl Edward Rasmussen

October 20th, 2016

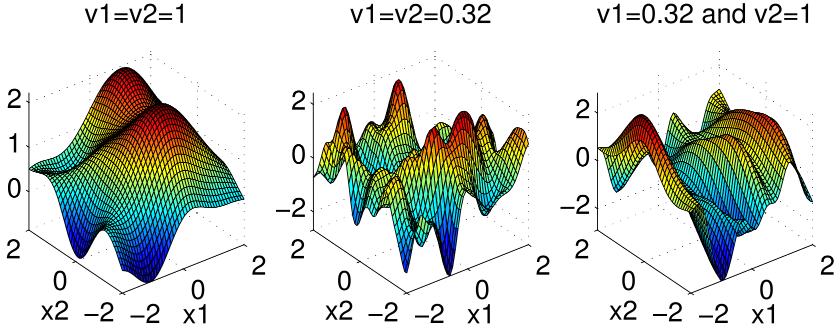
Key concepts

- chose covariance functions and use the marginal likelihood to
  - set hyperparameters
  - chose between different covariance functions
- covariance functions and hyperparameters can help interpret the data
- we illutrate a number of different covariance function families
  - stationary covariance functions: squared exponential, rational quadratic and Matérn forms
- many existing models are special cases of Gaussian processes
  - radial basis function networks (RBF)
  - splines
  - large neural networks
- combining existing simple covariance functions into more interesting ones

Model Selection, Hyperparameters, and ARD

We need to determine both the *form* and *parameters* of the covariance function.
We typically use a hierarchical model, where the parameters of the covariance are called hyperparameters.
A very useful idea is to use automatic relevance determination (ARD) covariance functions for feature/variable selection, e.g.:

$$k(\mathbf{x}, \mathbf{x}') = v_0^2 \exp\left(-\sum_{d=1}^{D} \frac{(x_d - x'_d)^2}{2v_d^2}\right), \qquad \text{hyperparameters } \theta = (v_0, v_1, \ldots, v_d, \sigma_n^2).$$



Rational quadratic covariance function

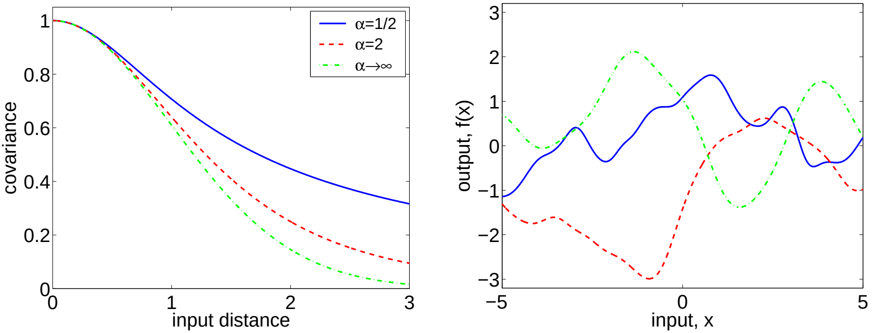
The *rational quadratic* (RQ) covariance function, where $r = x - x'$:

$$k_{RQ}(r) \;=\; \Big(1 + \frac{r^2}{2\alpha\ell^2}\Big)^{-\alpha}$$

with $\alpha,\ \ell > 0$ can be seen as a *scale mixture* (an infinite sum) of squared exponential (SE) covariance functions with different characteristic length-scales. Using $\tau = \ell^{-2}$ and $p(\tau|\alpha, \beta) \propto \tau^{\alpha-1}\exp(-\alpha\tau/\beta)$:

$$\begin{aligned} k_{RQ}(r) \;&=\; \int p(\tau|\alpha, \beta) k_{SE}(r|\tau) d\tau \\ &\propto\; \int \tau^{\alpha-1}\exp\Big(-\frac{\alpha\tau}{\beta}\Big)\exp\Big(-\frac{\tau r^2}{2}\Big) d\tau \;\propto\; \Big(1 + \frac{r^2}{2\alpha\ell^2}\Big)^{-\alpha}, \end{aligned}$$

Rational quadratic covariance function II



The limit $\alpha \to \infty$ of the RQ covariance function is the SE.

Matérn covariance functions

Stationary covariance functions can be based on the Matérn form:

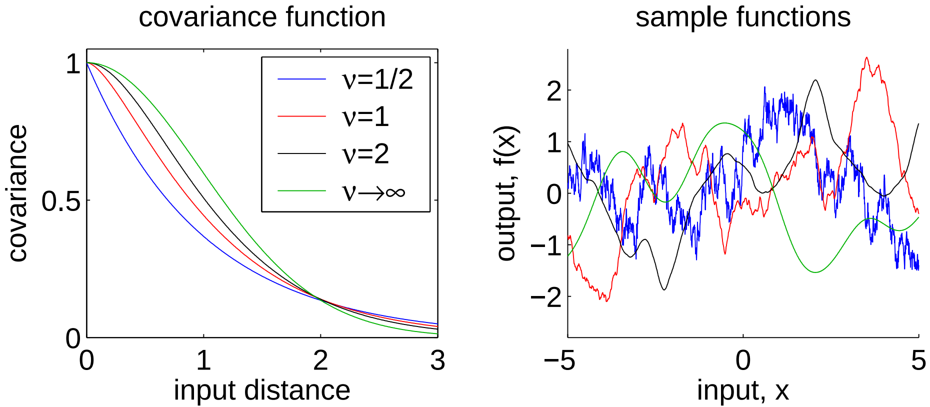
$$k(\mathbf{x}, \mathbf{x}') = \frac{1}{\Gamma(\nu)2^{\nu-1}}\Big[\frac{\sqrt{2\nu}}{\ell}|\mathbf{x}-\mathbf{x}'|\Big]^{\nu} K_{\nu}\Big(\frac{\sqrt{2\nu}}{\ell}|\mathbf{x}-\mathbf{x}'|\Big),$$

where K_ν is the modified Bessel function of second kind of order ν, and ℓ is the characteristic length scale.
Sample functions from Matérn forms are $\lfloor \nu - 1 \rfloor$ times differentiable. Thus, the hyperparameter ν can control the degree of smoothness
Special cases:

- $k_{\nu=1/2}(r) = \exp(-\frac{r}{\ell})$: Laplacian covariance function, Brownian motion (Ornstein-Uhlenbeck)
- $k_{\nu=3/2}(r) = \big(1 + \frac{\sqrt{3}r}{\ell}\big)\exp\big(-\frac{\sqrt{3}r}{\ell}\big)$ (once differentiable)
- $k_{\nu=5/2}(r) = \big(1 + \frac{\sqrt{5}r}{\ell} + \frac{5r^2}{3\ell^2}\big)\exp\big(-\frac{\sqrt{5}r}{\ell}\big)$ (twice differentiable)
- $k_{\nu\to\infty} = \exp(-\frac{r^2}{2\ell^2})$: smooth (infinitely differentiable)

Matérn covariance functions II

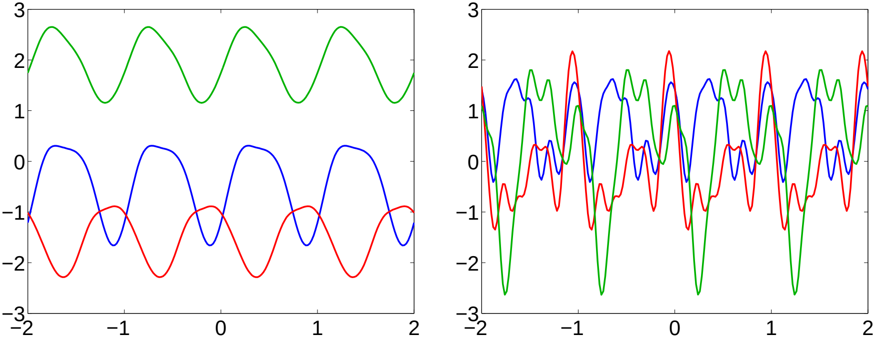
Univariate Matérn covariance function with unit characteristic length scale and unit variance:



Periodic, smooth functions

To create a distribution over periodic functions of x, we can first map the inputs to $u = (\sin(x), \cos(x))^\top$, and then measure distances in the u space. Combined with the SE covariance function, which characteristic length scale ℓ, we get:

$$k_{\text{periodic}}(x, x') = \exp(-2\sin^2(\pi(x - x'))/\ell^2)$$



Three functions drawn at random; left $\ell > 1$, and right $\ell < 1$.

Spline models

One dimensional minimization problem: find the function $f(x)$ which minimizes:

$$\sum_{i=1}^{c}(f(x^{(i)}) - y^{(i)})^2 + \lambda \int_0^1 (f''(x))^2 dx,$$

where $0 < x^{(i)} < x^{(i+1)} < 1, \ \forall i = 1, \ldots, n-1$, has as solution the Natural Smoothing Cubic Spline: first order polynomials when $x \in [0; x^{(1)}]$ and when $x \in [x^{(n)}; 1]$ and a cubic polynomical in each $x \in [x^{(i)}; x^{(i+1)}], \ \forall i = 1, \ldots, n-1$, joined to have continuous second derivatives at the knots.
The identical function is also the mean of a Gaussian process: Consider the class a functions given by:

$$f(x) = \alpha + \beta x + \lim_{n \to \infty} \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} \gamma_i (x - \frac{i}{n})_+, \qquad \text{where } (x)_+ = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

with Gaussian priors:

$$\alpha \sim \mathcal{N}(0, \xi), \quad \beta \sim \mathcal{N}(0, \xi), \quad \gamma_i \sim \mathcal{N}(0, \Gamma), \ \forall i = 0, \ldots, n-1.$$

The covariance function becomes:

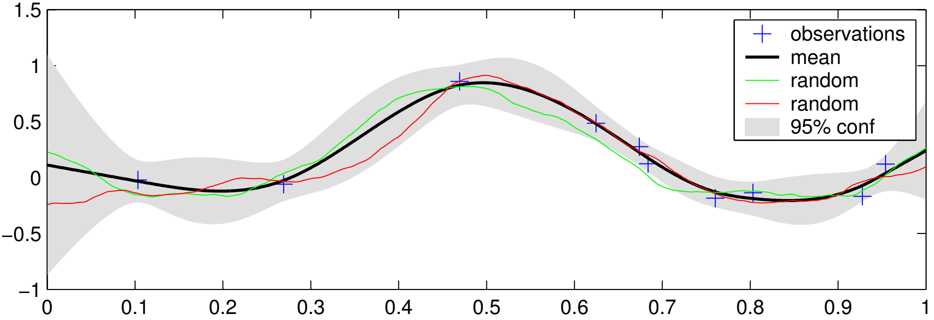
$$
\begin{aligned}
k(x, x') &= \xi + xx'\xi + \Gamma \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} (x - \frac{i}{n})_+ \; (x' - \frac{i}{n})_+ \\
&= \xi + xx'\xi + \Gamma \int_0^1 (x - u)_+ \; (x' - u)_+ du \\
&= \xi + xx'\xi + \Gamma\Big(\frac{1}{2}|x - x'| \min(x, x')^2 + \frac{1}{3} \min(x, x')^3\Big).
\end{aligned}
$$

In the limit $\xi \to \infty$ and $\lambda = \sigma_n^2/\Gamma$ the posterior mean becomes the natrual cubic spline.
We can thus find the hyperparameters σ^2 and Γ (and thereby λ) by maximising the marginal likelihood in the usual way.
Defining $h(x) = (1, x)^\top$ the posterior predictions with mean and variance:

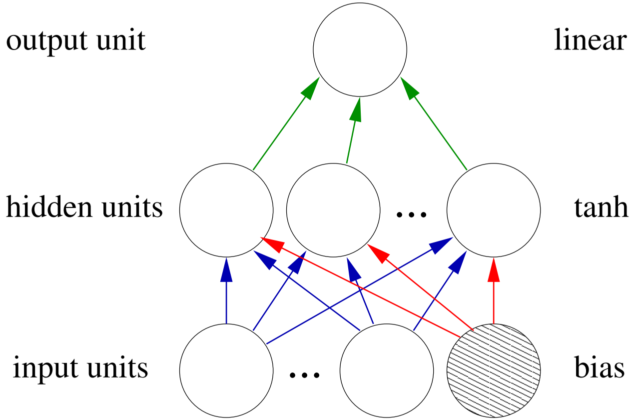
$$
\begin{aligned}
\tilde{\mu}(X_*) &= H(X_*)^\top \beta + K(X, X_*)[K(X, X) + \sigma_n^2 I]^{-1}(\mathbf{y} - H(X)^\top \beta) \\
\tilde{\Sigma}(x_*) &= \Sigma(X_*) + R(X, X_*)^\top A(X)^{-1} R(X, X_*) \\
\beta &= A(X)^{-1} H(X)[K + \sigma_n^2 I]^{-1}\mathbf{y}, \quad A(X) = H(X)[K(X, X) + \sigma_n^2 I]^{-1} H(X)^\top \\
R(X, X_*) &= H(X_*) - H(X)[K + \sigma_n^2 I]^{-1} K(X, X_*)
\end{aligned}
$$

Cubic Splines, Example

Although this is not the fastest way to compute splines, it offers a principled way of finding hyperparameters, and uncertainties on predictions.
Note also, that although the posterior mean is smooth (piecewise cubic), posterior sample functions are not.



Feed Forward Neural Networks



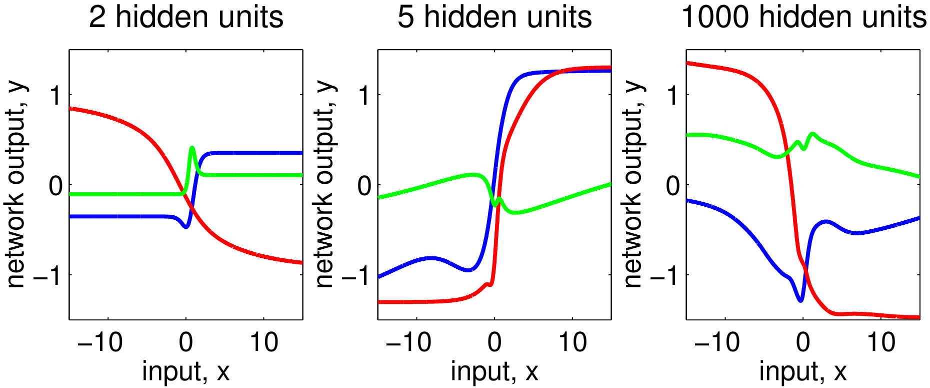
Weight groups:
output weights
input–hidden
bias–hidden

A feed forward neural network implements the function:

$$f(x) = \sum_{i=1}^{H} v_i \tanh\Big(\sum_j u_{ij} x_j + b_j\Big)$$

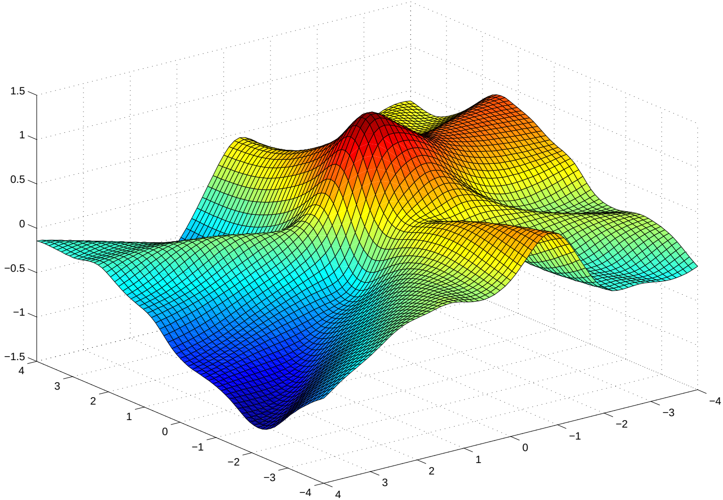
Limits of Large Neural Networks

Sample random neural network weights from the (Gaussian) prior.



Note: The prior on the neural network weights *induces* a prior over functions.

Function drawn at random from a Neural Network covariance function



$$k(x, x') = \frac{2}{\pi} \arcsin\Big(\frac{2x^\top \Sigma x'}{\sqrt{(1 + x^\top \Sigma x)(1 + 2x'^\top \Sigma x')}}\Big).$$

Composite covariance functions

We've seen many examples of covariance functions.

Covariance functions have to be possitive definite.

One way of building covariance functions is by composing simpler ones in various ways

- sums of covariance functions $k(x, x') = k_1(x, x') + k_2(x, x')$
- products $k(x, x') = k_1(x, x') \times k_2(x, x')$
- other combinations: $g(x)k(x, x')g(x')$
- etc.

The gpml toolbox supports such constructions.