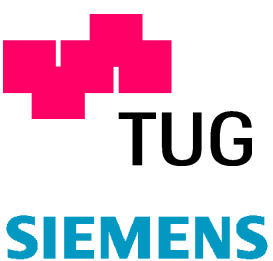
Some Aspects of Learning with Gaussian Processes

Anton Schwaighofer, May 2003



TU Graz, Institute for Theoretical Computer Science

Siemens Corporate Technology
Department of Neural Computation

Overview

- Gaussian processes basics
- Gaussian processes for Bayesian regression (GPR)
- The evidence procedure in GPR
- Choice of kernel functions
- Application areas

Gaussian Processes: Motivation

We wish to do (Bayesian) inference for predicting the value of some unknown function f on a test point $\boldsymbol{x}^*$, given noisy measurements (training data) $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$

Neural network Assume a neural network model with some parameters (weights) $\boldsymbol{w}$, set weights (train the network) to minimize the mean squared error on the training set

Gaussian processes Assume a prior distribution over the space of functions that possibly may have generated the training data. Compute the posterior distribution of functions $p\left(f \middle| \{\boldsymbol{x}_i, y_i\}\right)$ given the training data. Use the most likely function to make predictions

Gaussian processes are probably the simplest way of specifying a (non-trivial) prior over function space

Basics (1)

Stochastic process is a collection of random variables $\{F(\boldsymbol{x})|\boldsymbol{x} \in X\}$ indexed by a set X

- Signal processing: X is a time variable
- Here: $X = \mathbb{R}^D$ (locations in a D-dimensional space)

Stochastic processes are characterized by the joint distributions of finite subsets $\{F(\boldsymbol{x}_1), \ldots, F(\boldsymbol{x}_N)\}$

Gaussian process (GP): Any finite subset of F-variables has a joint multivariate Gaussian distribution. GP is specified by its mean function $\mu(\boldsymbol{x})$ (which we assume to be zero) and its covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$

$$k(\boldsymbol{x}, \boldsymbol{x}') = \mathrm{E}\left[F(\boldsymbol{x})F(\boldsymbol{x}')\right]$$

Basics (2)

For any set of locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ the associated random variables (functional values) $F(\boldsymbol{x}_1), \ldots, F(\boldsymbol{x}_N)$ are Gaussian distributed with mean $\boldsymbol{\mu} = 0$ and covariance matrix K

$$(F(\boldsymbol{x}_1), \ldots, F(\boldsymbol{x}_N)) \sim N(\mathbf{0}, K)$$
$$K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

$k(\cdot, \cdot)$ is the covariance function (kernel function) of the GP.

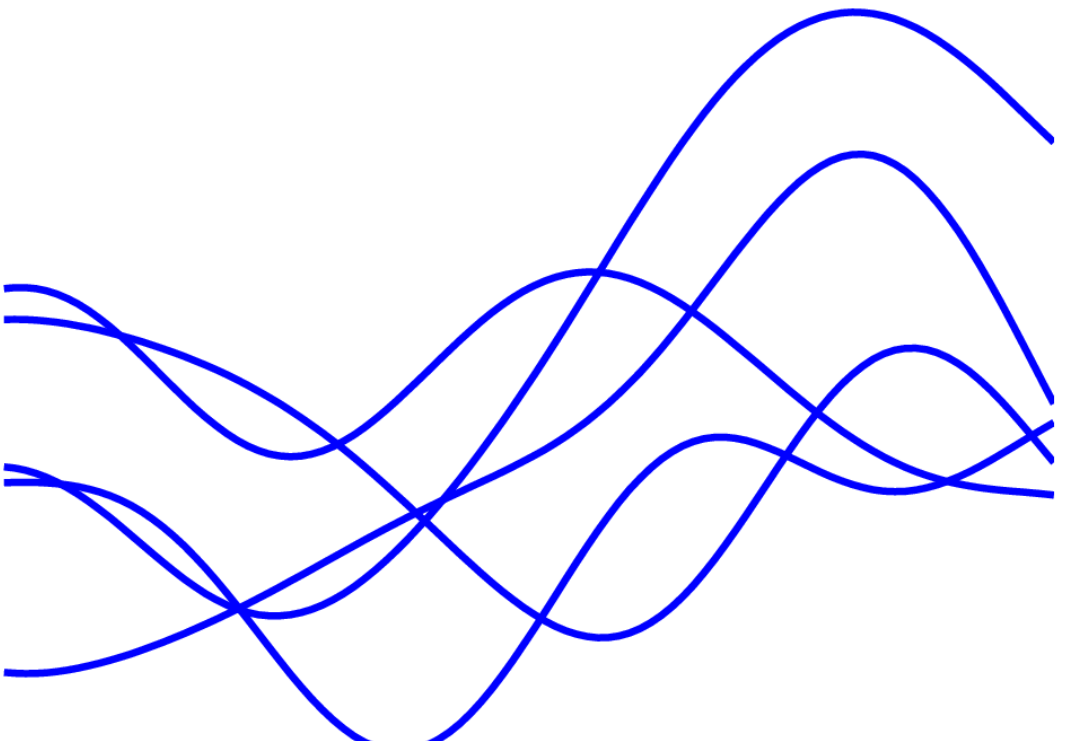
Typical choice in machine learning: $k(\cdot, \cdot)$ on data from $\mathbb{R}^D$ uses a parameterized formulation

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(\theta_0 - \sum_{d=1}^{D} \theta_d (x_d - x'_d)^2\right)$$

"squared exponential kernel"

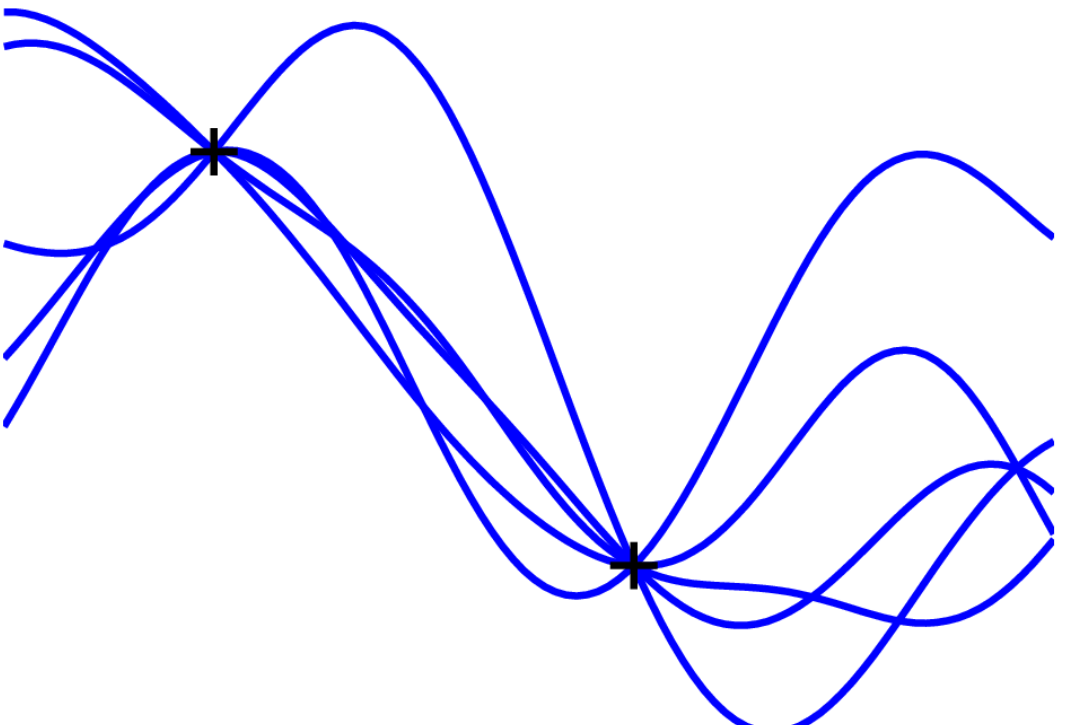
Gaussian Processes: Samples

5 samples of a GP with squared exponential kernel



Gaussian Processes: Samples

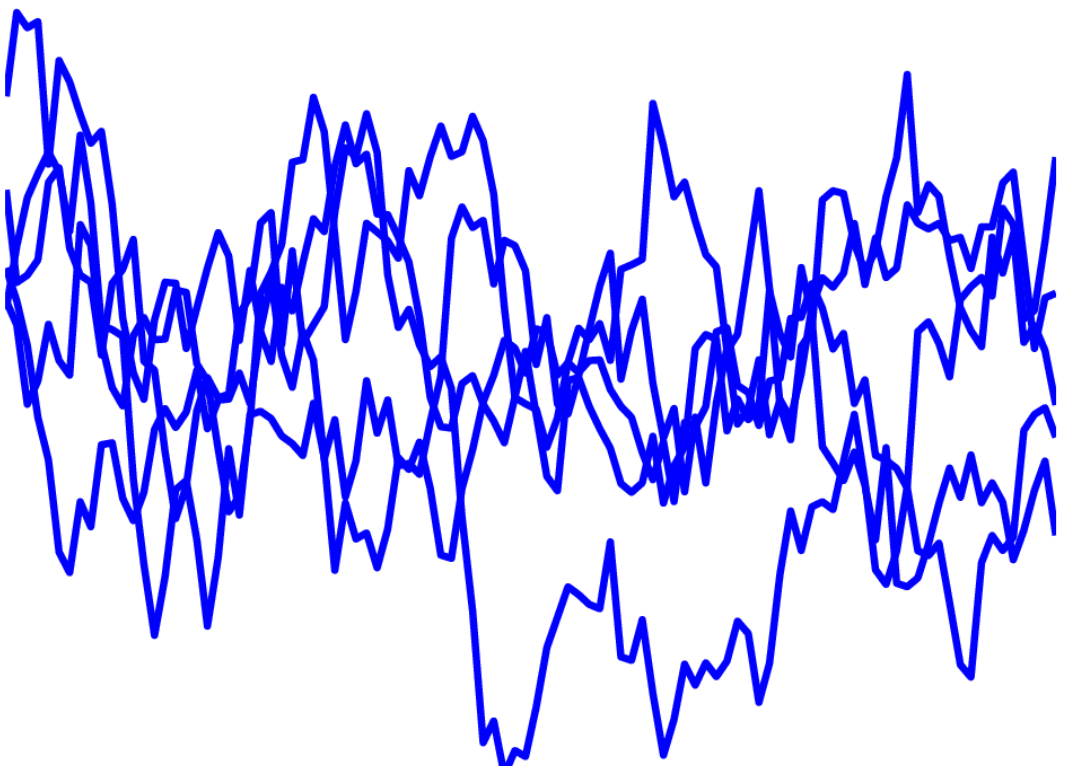
Prediction with Gaussian processes: 5 samples of the GP posterior when two training points are given



Gaussian Processes: Samples

5 samples of a GP with Ornstein-Uhlenbeck kernel

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



Bayesian Regression with GPs (1)

We have observed samples from some unknown function f on locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, corrupted by Gaussian noise with variance σ^2:

$$
\begin{aligned}
y_i &= f(\boldsymbol{x}_i) + \epsilon_i, \quad i = 1, \ldots, N \\
\epsilon_i &\sim N(0, \sigma^2) \quad \text{Gaussian noise} \\
(f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_N)) &\sim N(0, K) \quad \text{Gaussian process prior}, K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j) \\
(y_1, \ldots, y_N) &\sim N(0, K + \sigma^2 I)
\end{aligned}
$$

We wish to predict at test point $\boldsymbol{x}^*$. From the GP property, we also know that

$$(y_1, \ldots, y_N, f(\boldsymbol{x}^*)) \sim N(0, K^*)$$

Everything is Gaussian: Conditional mean (prediction) is

$$
\begin{aligned}
p\big(f(\boldsymbol{x}^*)|y_1, \ldots, y_N, \theta\big) &= \frac{p\big(y_1, \ldots, y_N, f(\boldsymbol{x}^*)|\theta\big)}{p\big(y_1, \ldots, y_N|\theta\big)} \\
E(f(\boldsymbol{x}^*)|\mathcal{D}) &= \sum_{i=1}^{N} \alpha_i k(\boldsymbol{x}^*, \boldsymbol{x}_i) \\
\text{with } (K + \sigma^2 \mathbf{1})\boldsymbol{\alpha} &= \boldsymbol{y}
\end{aligned}
$$

Evidence maximization

We have so far assumed that the covariance function is given. GPR allows us to infer the most likely parameters of the kernel function from the training data.

Key quantity is the evidence: (marginal) log likelihood of the training data under the GP model with kernel hyperparameters $\boldsymbol{\theta}$:

$$\log p(\boldsymbol{y}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N, \boldsymbol{\theta}) = \\ -\frac{1}{2}\log\det(K + \sigma^2\mathbf{1}) - \frac{1}{2}\boldsymbol{y}^\top(K + \sigma^2\mathbf{1})^{-1}\boldsymbol{y} - \frac{N}{2}\log 2\pi \quad (1)$$

Integrate out Assume prior distributions on the hyperparameters $\boldsymbol{\theta}$ and integrate out with Monte Carlo method

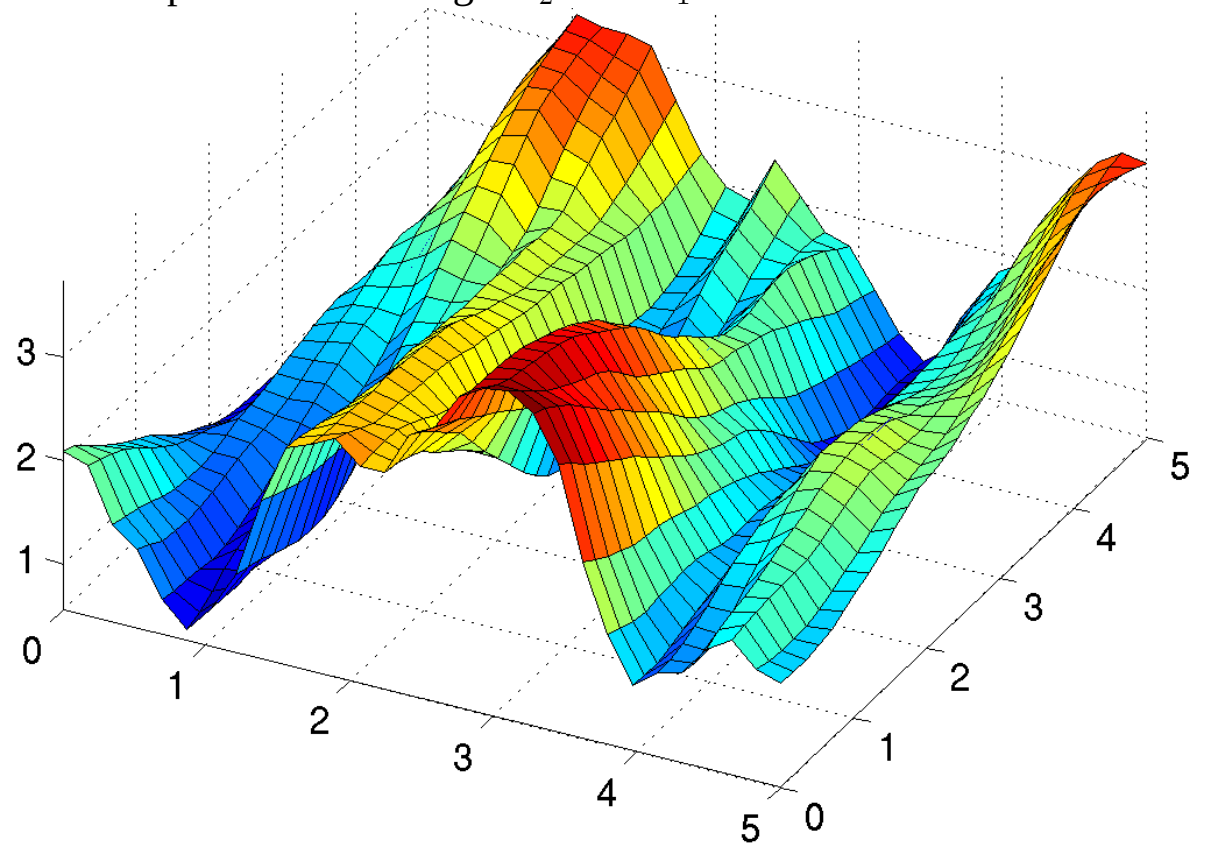
Maximization Maximize $\log p(\boldsymbol{y}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N, \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ (e.g. using conjugate gradient method)

GP with evidence maximization does implicit input pruning: Small θ_d in the kernel function indicate irrelevant inputs

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(\theta_0 - \sum_{d=1}^{D}\theta_d(x_d - x'_d)^2\right)$$

Samples of a GP With Anisotropic Kernel

Gaussian process with weight $\theta_2 = 25\theta_1$:



Bayesian Regression with GPs (2)

Cookbook: How to do Gaussian process regression with training data $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N, y_1, \ldots, y_N\}$

1. Assume a parameterized model for the covariance function $k_\theta(\cdot, \cdot)$ with parameters $\boldsymbol{\theta}$
2. Choose parameters $\boldsymbol{\theta}$ to maximize evidence: Choose the model that best explains the training data
3. Compute kernel matrix K, with $K_{ij} = k_\theta(\boldsymbol{x}_i, \boldsymbol{x}_j)$
4. Solve linear system $(K + \sigma^2 \mathbf{1})\boldsymbol{\alpha} = \boldsymbol{y}$
5. Predict value y^* for a test point $\boldsymbol{x}^*$ from $\sum_{i=1}^{N} \alpha_i k_\theta(\boldsymbol{x}^*, \boldsymbol{x}_i)$ (mean of the GP posterior)
6. Obtain error bars for y^* from the variance of the GP posterior

Relations to Other Methods

Regression with Gaussian Processes is closely related to

Kriging Best linear unbiased estimator (minimum variance), originally developed for spatial data in geostatistics

Infinite Neural Networks A 2-layer neural network with increasing number of hidden units converges to a Gaussian process predictor with a certain type of covariance function

Regularization Choice of kernel corresponds to a regularization operator (smoothness prior)

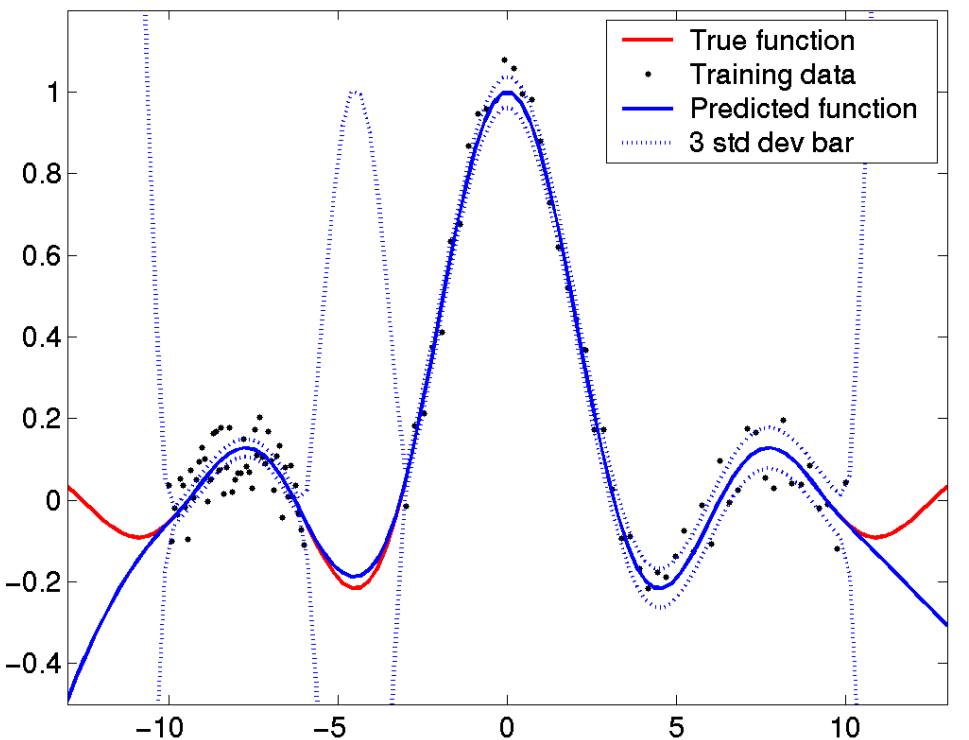
Overview

- Gaussian processes basics
- Gaussian processes for Bayesian regression (GPR)
- The evidence procedure in GPR
- Choice of kernel functions
- Application areas

Gaussian Processes versus RVM

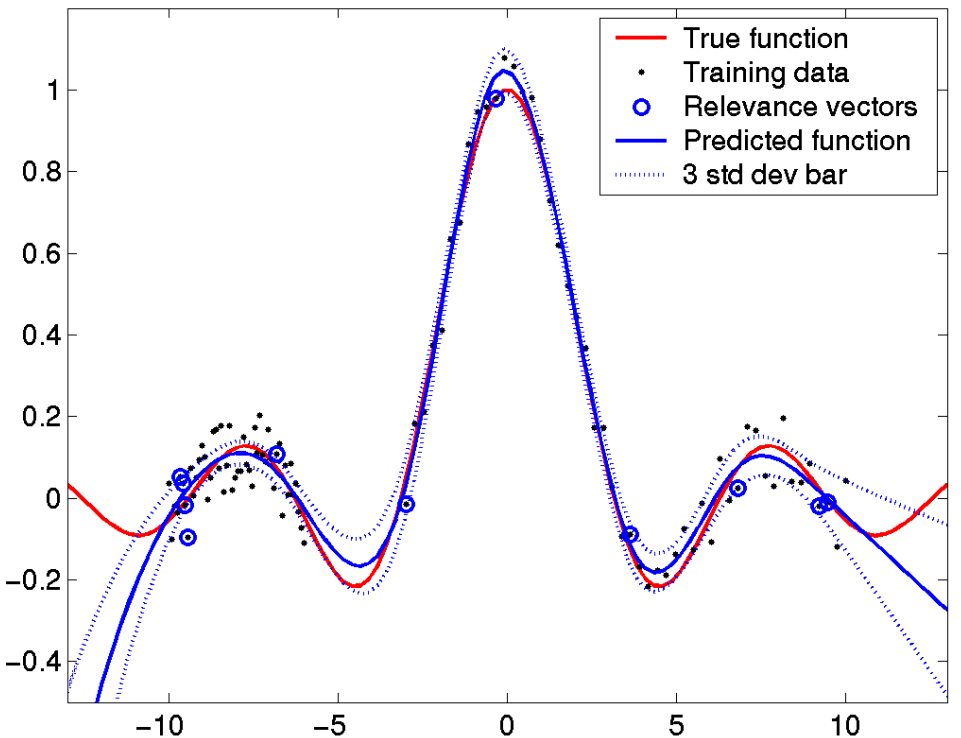
- Noisy sinc problem with regions of missing data
- Thin plate spline kernel function (as in Tipping's toy example)

True function and GP prediction with error bars:



Gaussian Processes versus RVM

True function and RVM prediction with error bars:



RVM is grossly over-optimistic about the variance!

On the Choice of Kernel Function (1)

Widely used kernel function: Squared exponential (aka Gaussian, aka RBF kernel)

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(\theta_0 - \sum_{d=1}^{D} \theta_d (x_d - x'_d)^2\right) = \exp\left(\theta_0 - \|\boldsymbol{x} - \boldsymbol{x}'\|^2\right)$$

Criticism:

- Unreasonable smoothness assumption
- Variance of predictions tends to be underestimated

Alternatives: Matern kernel

$$k(\boldsymbol{x}, \boldsymbol{x}') = M_\nu(z) = \frac{2\left(\sqrt{\nu} z\right)^\nu}{\Gamma(\nu)} K_\nu(2\sqrt{\nu} z)$$

with $z = \|\boldsymbol{x} - \boldsymbol{x}'\|$. $\Gamma(\nu)$ is the Gamma function and $K_\nu(r)$ is the modified Bessel function of the second kind of degree ν.

On the Choice of Kernel Function (2)

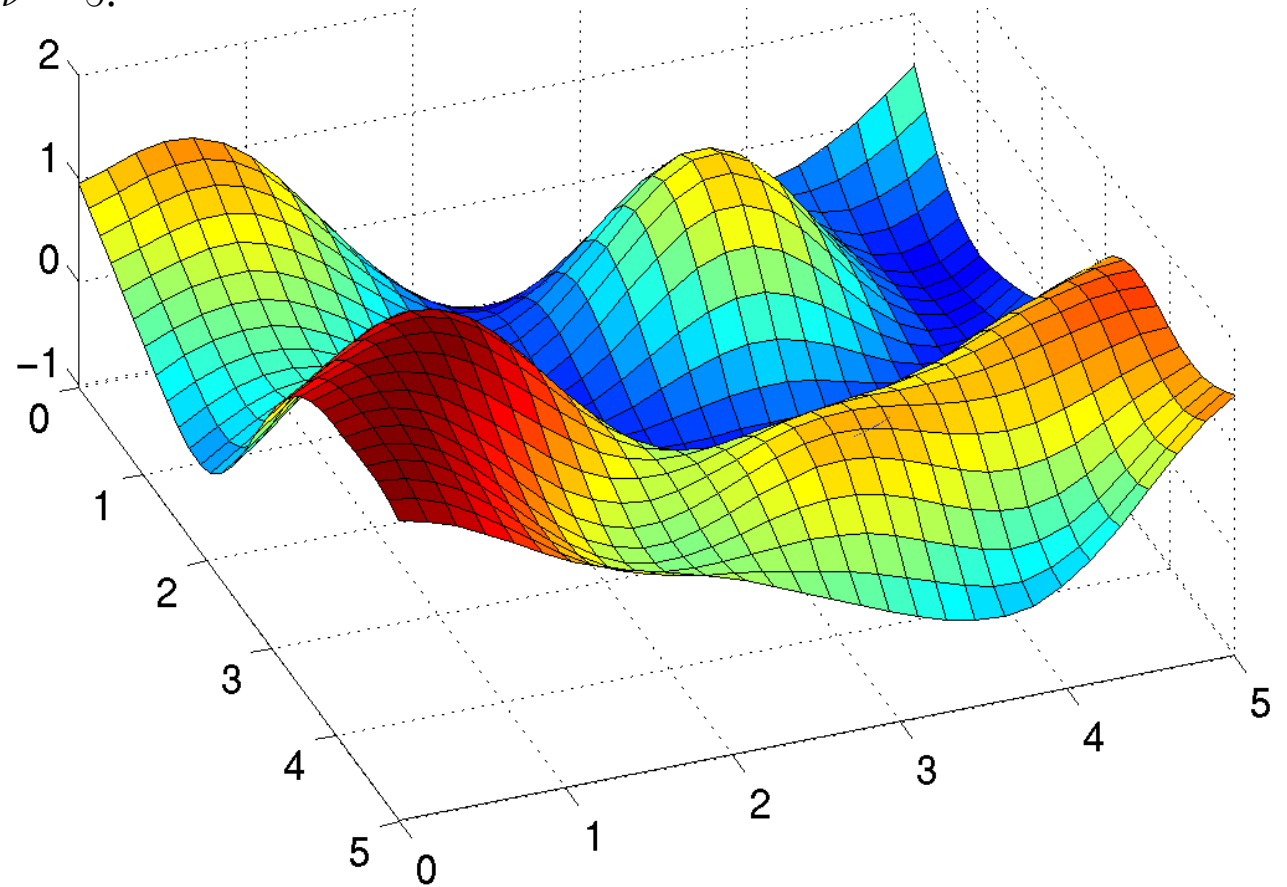
Matern kernel

$$k(\boldsymbol{x}, \boldsymbol{x}') = M_\nu(z) = \frac{2\left(\sqrt{\nu}z\right)^\nu}{\Gamma(\nu)} K_\nu(2\sqrt{\nu}z)$$

- With $\nu \to \infty$, the Matern kernel converges to the squared exponential kernel
- ν allows a continuous parameterization of the fractal dimension of the underlying process from smooth ($\nu \to \infty$) to rough
- Often, there is no basis for knowing the degree of smoothness of some process *a priori*. The Matern kernel allows to infer the smoothness from the data through the evidence procedure.
- Disadvantage: Computationally more intensive (Bessel and Gamma functions), numerically sensitive

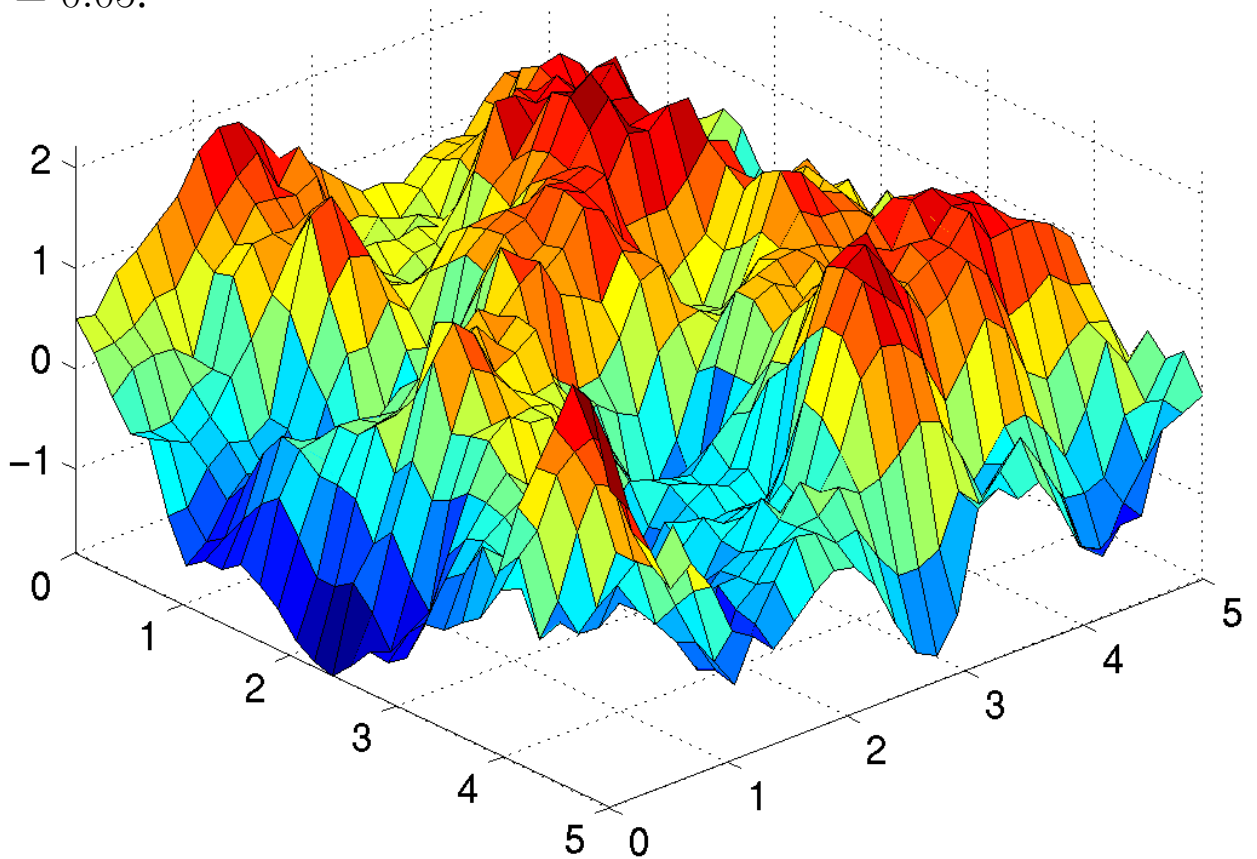
GP with Matern Kernel: Samples (1)

$\nu = 5$:



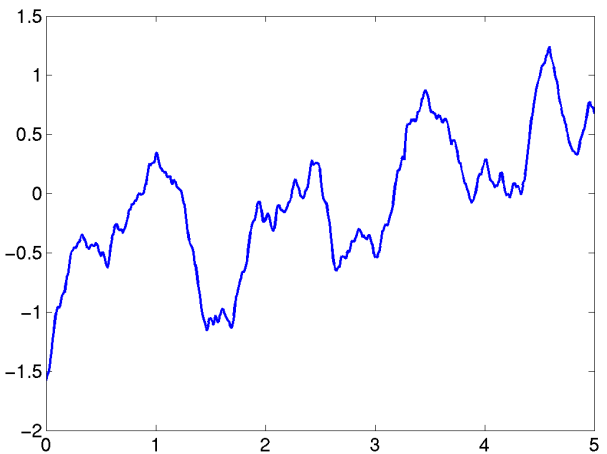
GP with Matern Kernel: Samples (2)

$\nu = 0.05$:

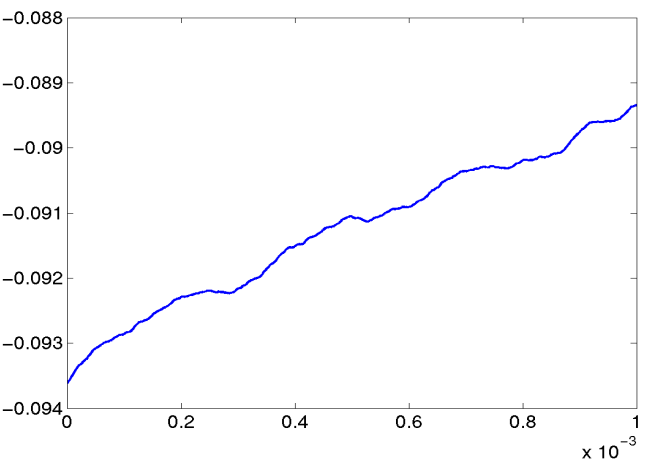


GP with Matern Kernel: Samples (3)

Normal scale



Zoom in



- Sample paths become increasingly rough when $\nu \to 0$
- Random structure visible at all scales

Application areas (1)

- Regression and classification problems
- Modelling of dynamic systems
- Neural responses (finding the most likely stimulus to a pool of spiking neurons)
- Learning kernel functions (Platt, MSR: Kernels for music playlists)

Advantages:

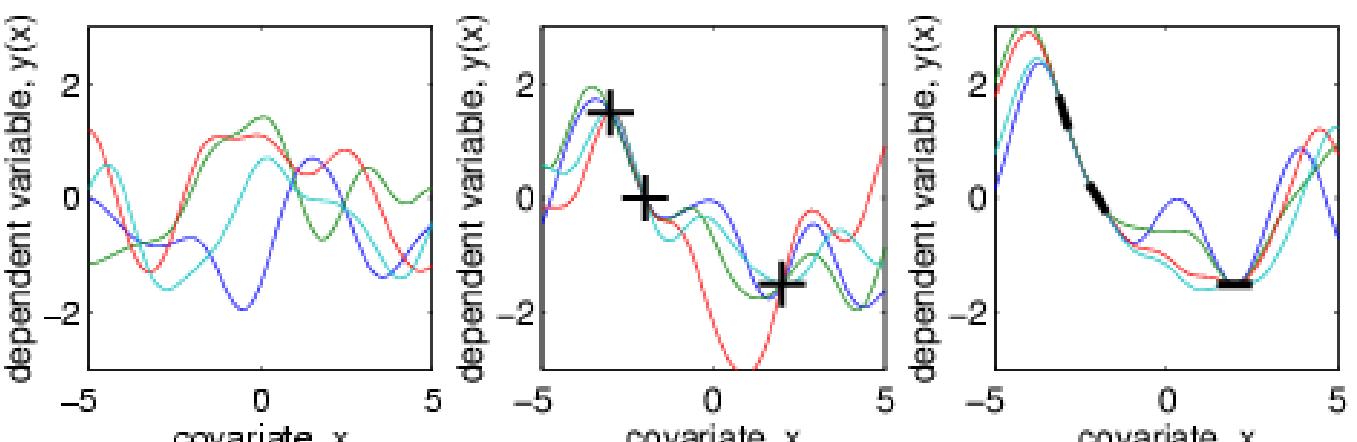
- Tractable exact Bayesian inference: Many of the distributions of interest are again Gaussian
- Kernel functions allow working on arbitrary data types (as long as you can define a kernel on them)

Application Areas (2): Dynamic Systems

- Derivative of a Gaussian process is again a Gaussian process:

$$\operatorname{cov}\left(f(\boldsymbol{x}), \frac{\partial f(\boldsymbol{x}')}{\partial x_i}\right) = \frac{\partial}{\partial x_i} \operatorname{cov}(f(\boldsymbol{x}), f(\boldsymbol{x}'))$$

- Derivative observations can be incorporated into a GP model in a straightforward way, leading to reduced predictive variance
- Application: Build an accurate model of complex probability distributions from few samples, for increasing speed of Monte Carlo simulations (Rasmussen, Bayesian Statistics vol. 7)



Application Areas (3)

Regression problems on non-vectorial data

- Gaussian processes only require knowledge of kernel evaluations $k(\boldsymbol{x}, \boldsymbol{x}')$
- We can work with any type of data we can formulate a kernel for (similar to Support Vector Machines)
- Application: Gaussian processes for generating music playlists from discrete features (Platt, NIPS*2001)

Conclusions

- Gaussian processes are an elegant and easy-to-use way for doing nonlinear regression
- GPs have shown excellent performance on many regression and classification task
- General feature of kernel systems: handling of arbitrary data types (as long as you can define a kernel = covariance function for them)

Active areas of research:

- Kernel design, in particular for applications with non-vectorial data