Machine Learning

Gaussian Processes

Wojciech Czech

January 9, 2024

Institute of Computer Science

Table of contents

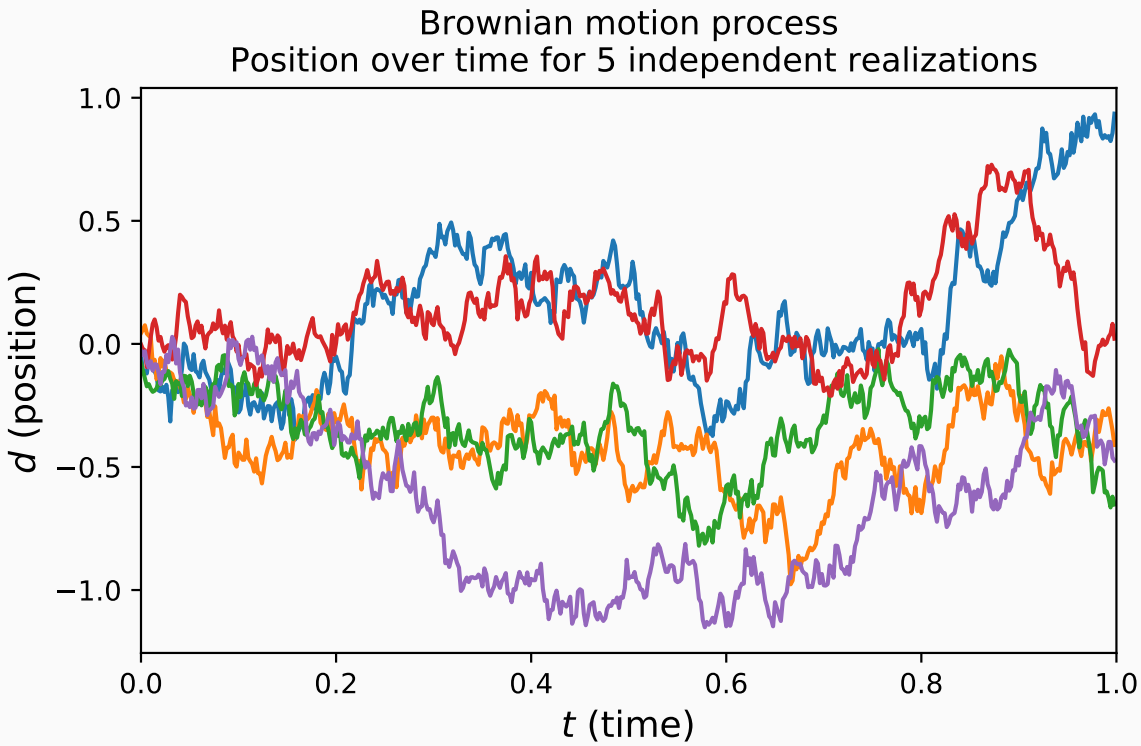
Stochastic processes

Brownian motion

- Random motion of particles suspended in the fluid
- Random walk of particles due to other particles randomly bumping into them

$$d(t + \Delta t) = d(t) + \Delta d \tag{1}$$

$$\Delta d \sim \mathcal{N}(0, \Delta t) \tag{2}$$



Stochastic processes - distributions over functions

- Stochastic processes describes systems randomly changing over time
- Trajectories (paths, realizations) of the stochastic process are different for the same starting point
- Every realization corresponds to a function
- Stochastic process can be understood as a distribution over functions

Gaussian processes

Gaussian processes

Gaussian process

Gaussian process is a Gaussian distribution over functions $f(\boldsymbol{x})$, defined by mean function $m(\boldsymbol{x})$ and positive definite covariance function (**kernel function**) $k(\boldsymbol{x}, \boldsymbol{x}')$,

$$f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')) \tag{3}$$

such that the set of values of $f(\boldsymbol{x})$ evaluated at an arbitrary set of points $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ jointly have Gaussian distribution:

$$\boldsymbol{y} = f(\boldsymbol{X}) \sim \mathcal{N}(m(\boldsymbol{X}), k(\boldsymbol{X}, \boldsymbol{X})) \tag{4}$$

with mean vector $m(\boldsymbol{X})$ and covariance matrix $k(\boldsymbol{X}, \boldsymbol{X})$.

Positive definite matrix

Positive definite matrix

A symmetric matrix $\boldsymbol{A}$ whose eigenvalues are positive is called *positive definite* ($\boldsymbol{z}^T\boldsymbol{A}\boldsymbol{z} > 0$ for $\boldsymbol{z} \in \mathbb{R}^{n\times 1}$ and $\boldsymbol{z} \neq \boldsymbol{0}$).

Kernel function as prior

- Sampling functions from Gaussian process requires defining mean and covariance (kernel) functions
- Covariance function models the joint variability of Gaussian process random variables and returns modeled covariance for each pair of inputs $k(\boldsymbol{x}, \boldsymbol{x}')$
- By choosing a specific kernel function k it is possible to set prior information on the distribution of $f(\boldsymbol{x})$

Sampling from prior

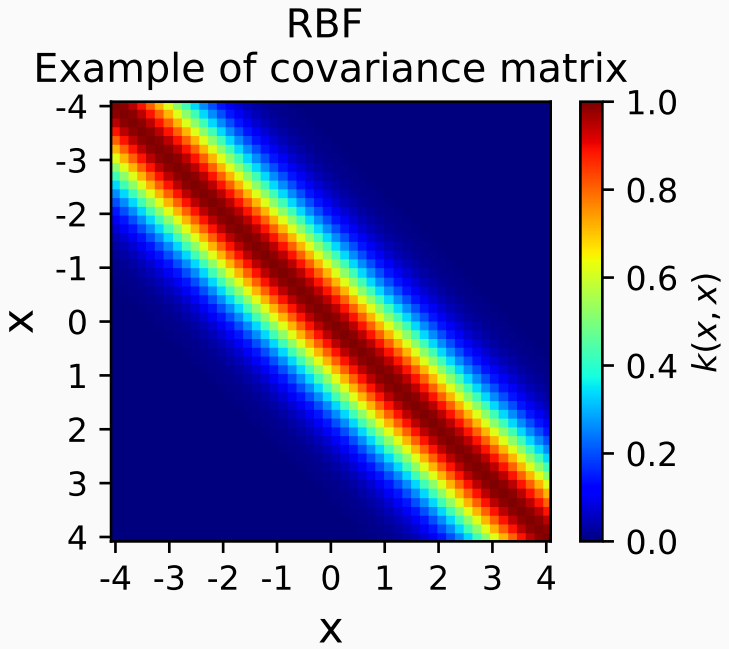
1. Sample function evaluations $\boldsymbol{y}$ from function $f(\boldsymbol{x})$ drawn from Gaussian process at arbitrary set of points $\boldsymbol{X}$

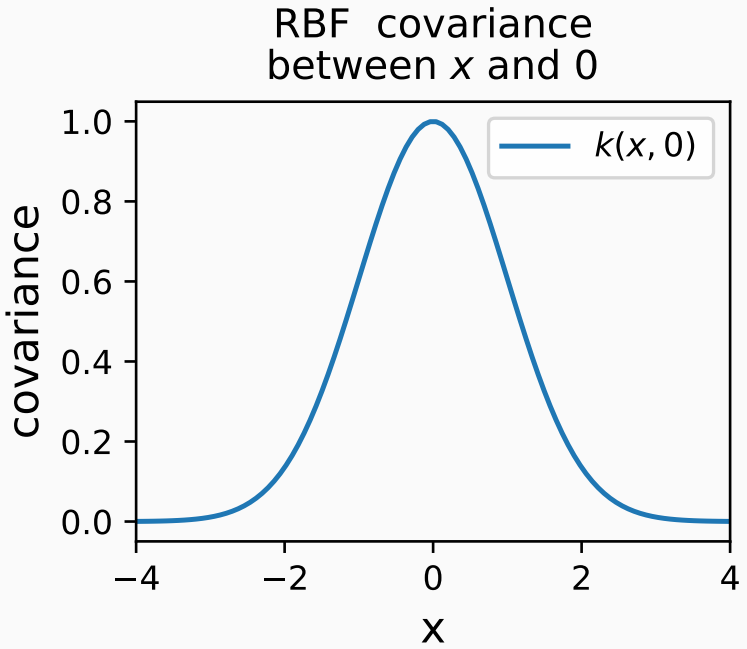
$$\boldsymbol{y} = f(\boldsymbol{X}) \tag{5}$$

2. Finite dimensional subset of the Gaussian process distribution results in Gaussian distribution $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu} = m(\boldsymbol{X})$ and $\boldsymbol{\Sigma} = k(\boldsymbol{X}, \boldsymbol{X})$

Example 1: RBF kernel

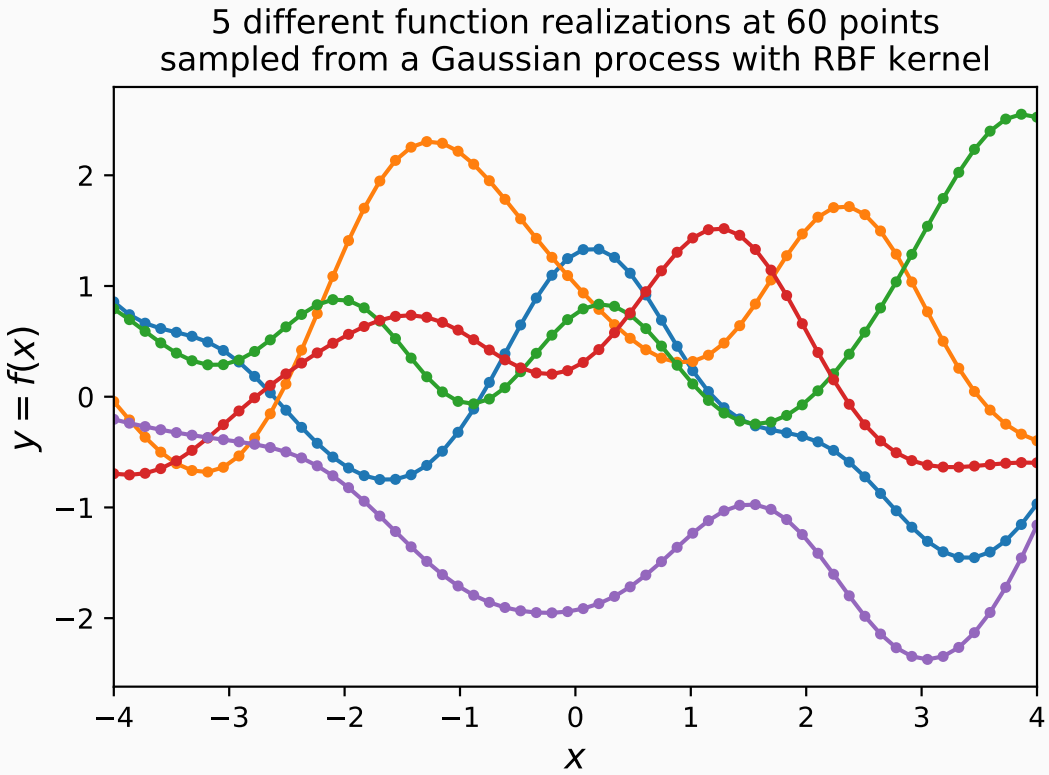
$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{1}{2\sigma^2}\left\|\boldsymbol{x} - \boldsymbol{x}'\right\|^2\right) \tag{6}$$



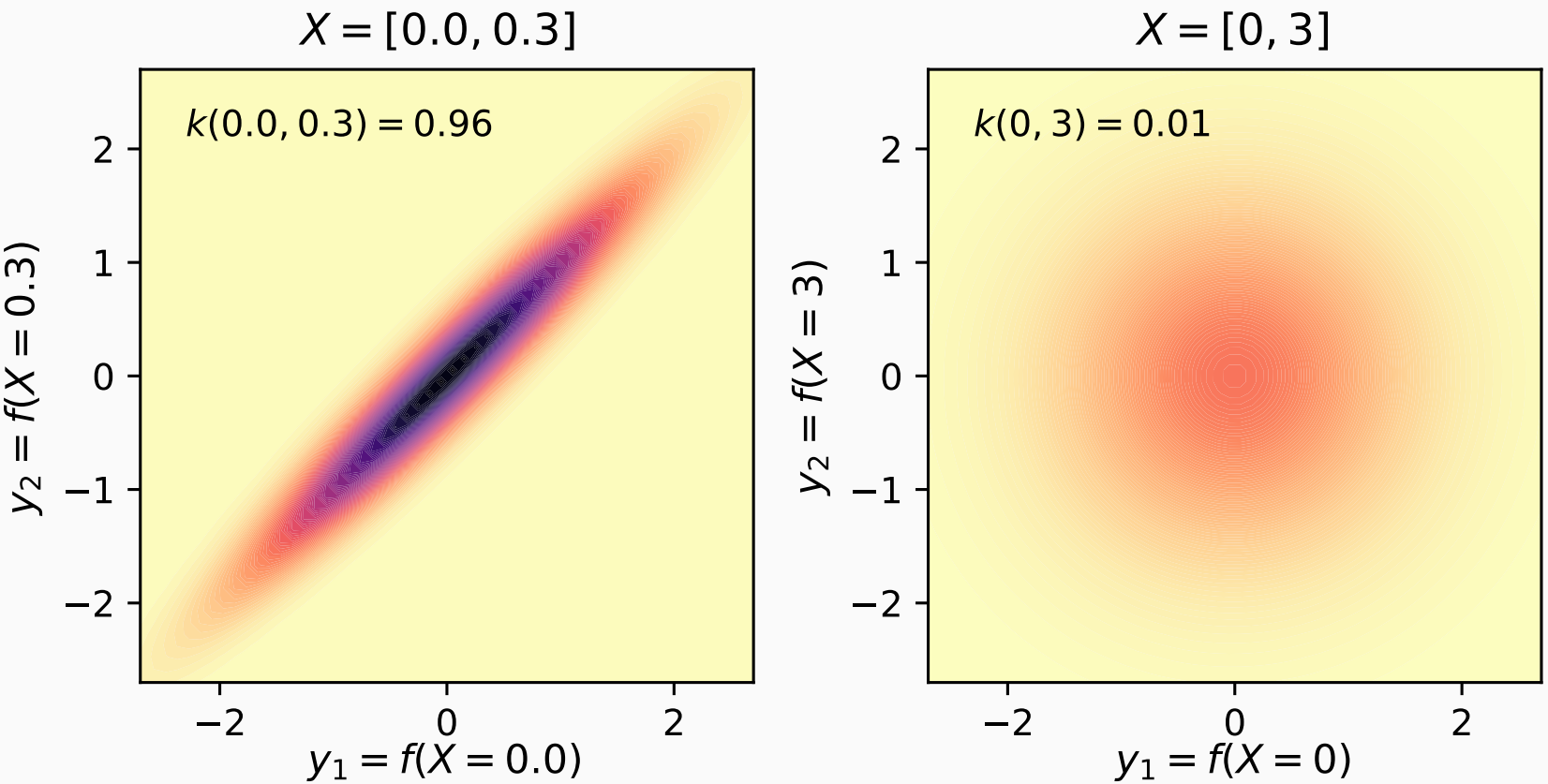


Example 1: Sampling from prior

1. Generate 60 equidistant samples from $[-4, 4]$: $\boldsymbol{X} = \{x_1, \ldots, x_{60}\}$
2. Calculate covariance matrix $= k(\boldsymbol{X}, \boldsymbol{X})$
3. Generate 5 samples (realizations) from 60-dimensional Gaussian distribution with covariance matrix $\boldsymbol{\Sigma}$ and zero mean $\mathbf{0}$

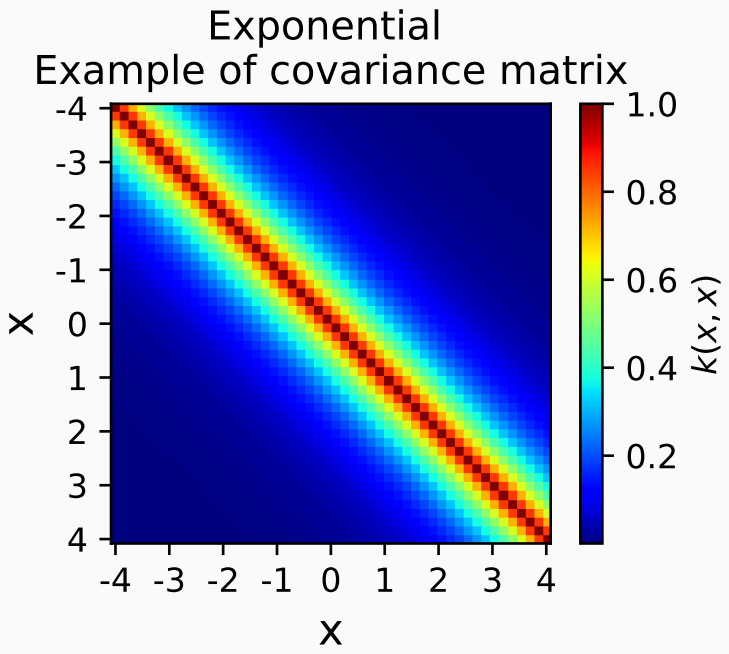


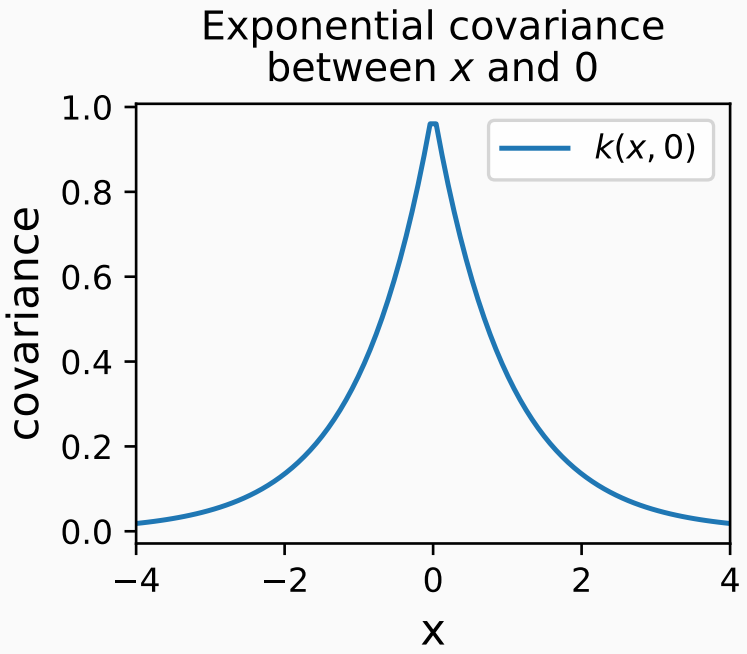
Example 1: 2D marginal distributions



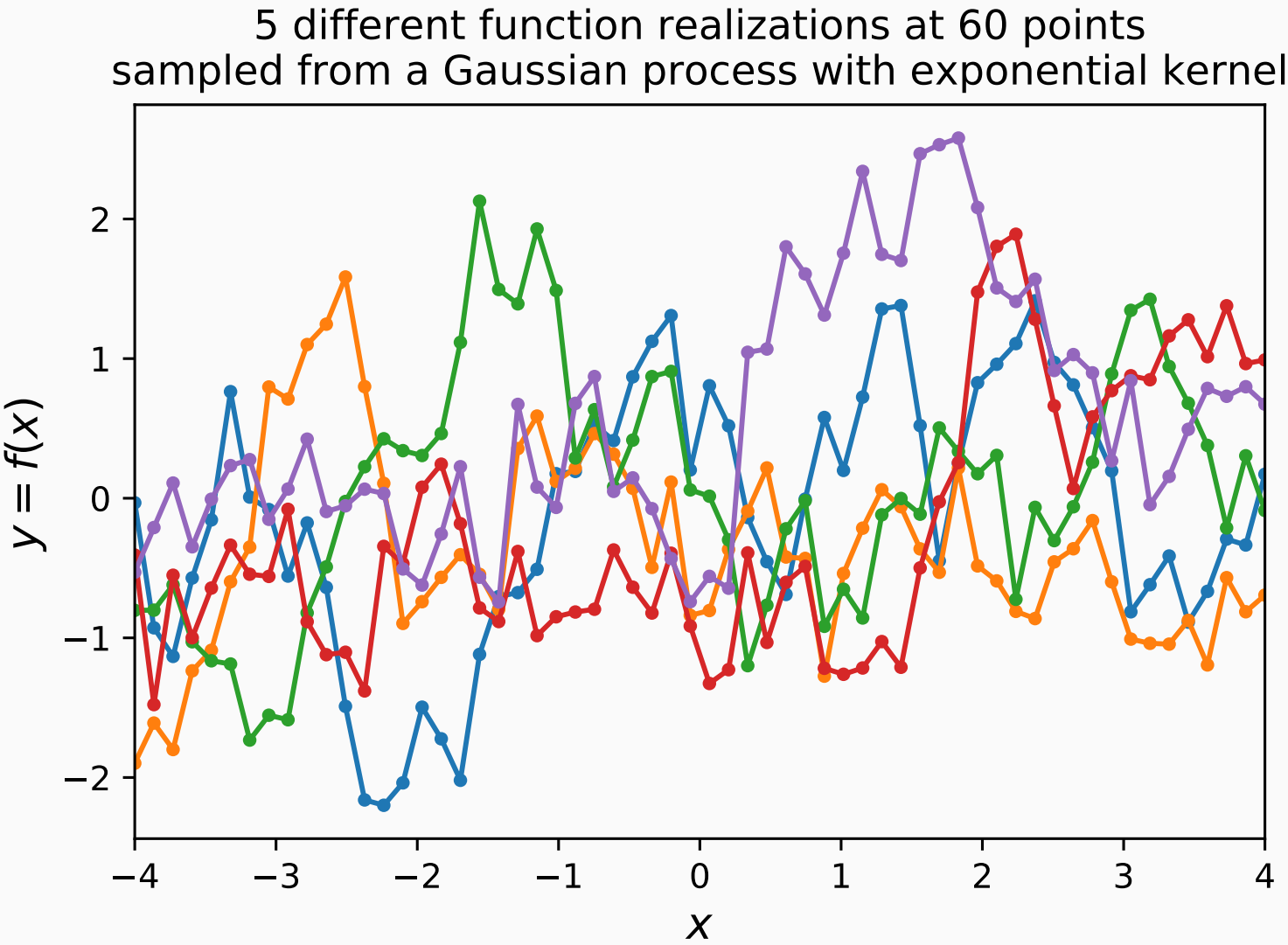
Example 2: Exponential kernel

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

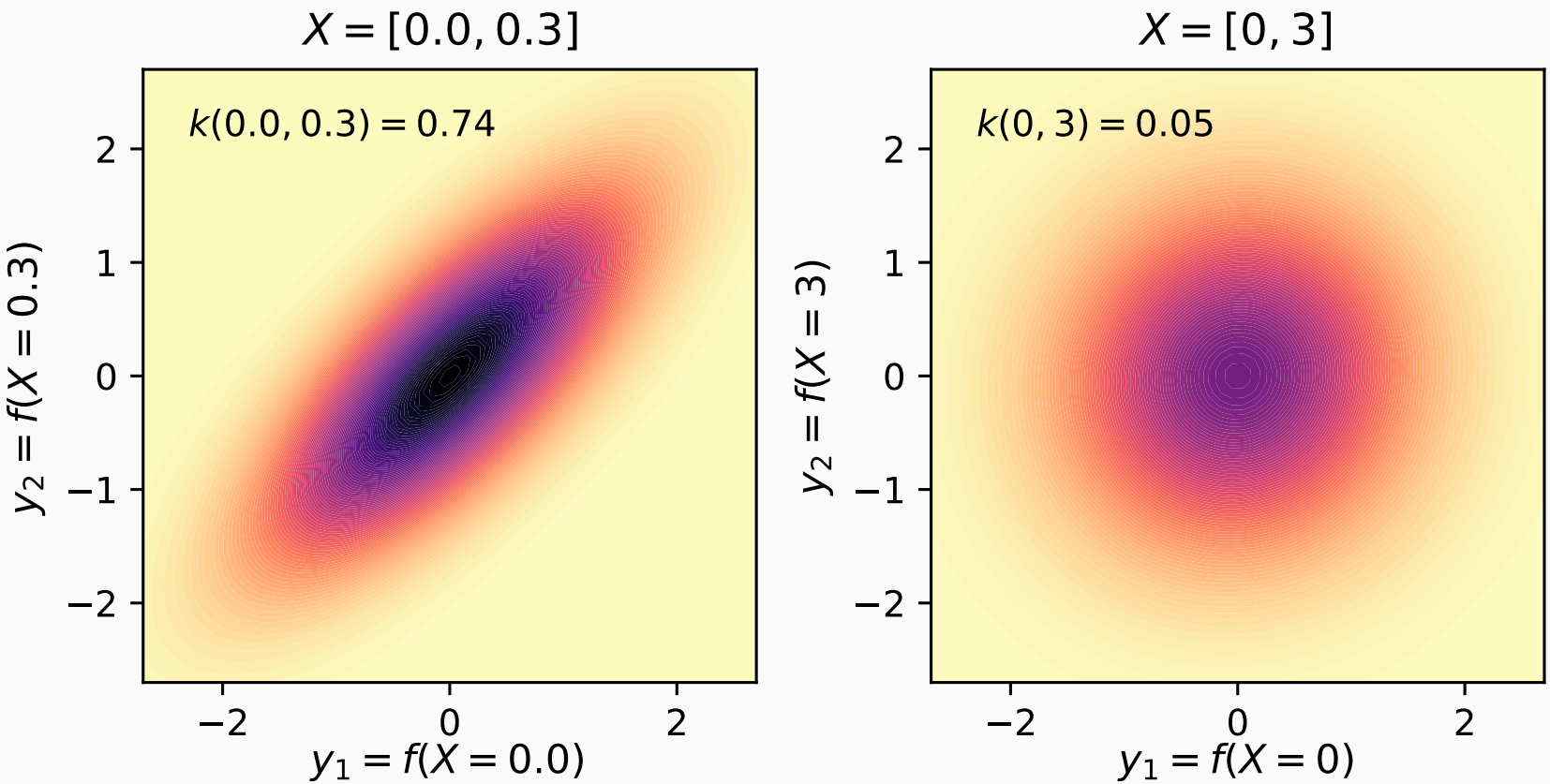




Example 2: Sampling from prior



Example 2: 2D marginal distributions



Gaussian processes for regression

Linear regression revisited

- Linear combination of fixed basis functions

$$y(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}) \tag{8}$$

- Prior distribution over $\boldsymbol{w}$

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\mathbf{0}, \alpha^{-1}\boldsymbol{I}) \tag{9}$$

- Probability distribution over $\boldsymbol{w}$ induces probability distribution over functions $y(\boldsymbol{w})$
- Joint probability of the function values: $\boldsymbol{y} = [y(\boldsymbol{x}_1), \ldots, y(\boldsymbol{x}_n)]^T$

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{w} \tag{10}$$

where $[\boldsymbol{\Phi}]_{i,j} = \phi_j(\boldsymbol{x}_i)$ - design matrix

Joint distribution over y and Gaussian process

$$\mathbb{E}[\boldsymbol{y}] = \boldsymbol{\Phi}\mathbb{E}[\boldsymbol{w}] = \mathbf{0} \tag{11}$$

$$\text{cov}[\boldsymbol{y}] = \mathbb{E}[\boldsymbol{y}\boldsymbol{y}^T] = \boldsymbol{\Phi}\mathbb{E}[\boldsymbol{w}\boldsymbol{w}^T]\boldsymbol{\Phi}^T = \frac{1}{\alpha}\boldsymbol{\Phi}\boldsymbol{\Phi}^T = \boldsymbol{K} \tag{12}$$

$$[\boldsymbol{K}]_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{1}{\alpha}\phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j) \tag{13}$$

$\boldsymbol{K}$ - Gram matrix

$k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ - kernel function vs. selecting basis

Prior distribution over $\boldsymbol{w}$ vs. prior distribution over functions $y(\boldsymbol{x})$

$$p(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y}|\mathbf{0}, \boldsymbol{K}) \tag{14}$$

Gaussian process for regression

1. Use Gaussian process defined by kernel function as prior
2. Generate posterior distribution given training data
3. Use posterior distribution to predict the expected value and probability of the output variable y given input variables

Regression problem with noise

- Training data
  a) Target values: $\boldsymbol{t}_n = [t_1, \ldots, t_n]^T$
  b) Input values: $\boldsymbol{X}_n = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$
- Noise on observed target values

$$t_i = y_i + \epsilon_i \tag{15}$$

  where $y_i = f(\boldsymbol{x}_i)$, $\boldsymbol{y}_n = [y_1, \ldots, y_n]^T$
- Noise has Gaussian distribution

$$p(t_i|y_i) = \mathcal{N}(t_i|y_i, \beta^{-1}) \tag{16}$$

- Joint distribution on data points (see [1])

$$p(\boldsymbol{t}_n|\boldsymbol{y}_n) = \mathcal{N}(\boldsymbol{t}_n|\boldsymbol{y}_n, \beta^{-1}\boldsymbol{I}_n) \tag{17}$$

Predictive distribution

- Model of the joint distribution over sets of data points

$$p(\boldsymbol{t}) = \int p(\boldsymbol{t}|\boldsymbol{y})p(\boldsymbol{y})d\boldsymbol{y} = \mathcal{N}(\boldsymbol{t}|\boldsymbol{0}, \boldsymbol{C}) \tag{18}$$

$$[\boldsymbol{C}]_{i,j} = C(\boldsymbol{x}_i, \boldsymbol{x}_j) = k(\boldsymbol{x}_i, \boldsymbol{x}_j) + \beta^{-1}\delta_{ij} \tag{19}$$

- Prediction problem: provide target value t_{n+1} for a new input vector $\boldsymbol{x}_{n+1}$
- Find predictive distribution
  $p(t_{n+1}|\boldsymbol{t}_n, \boldsymbol{X}_n, \boldsymbol{x}_{n+1}) = p(t_{n+1}|\boldsymbol{t}_n, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n+1})$

Step 1: Joint distribution $p(\boldsymbol{t}_{n+1})$

$$p(\boldsymbol{t}_{n+1}) = p(t_1, \ldots, t_{n+1}) = \mathcal{N}(\boldsymbol{t}_{n+1}|\boldsymbol{0}, \boldsymbol{C}_{n+1}) \tag{20}$$

$\boldsymbol{C}_{n+1} \in \mathbb{R}^{(n+1)\times(n+1)}$ - covariance matrix

$$C(\boldsymbol{x}_i, \boldsymbol{x}_j) = k(\boldsymbol{x}_i, \boldsymbol{x}_j) + \beta^{-1}\delta_{ij} \tag{21}$$

$$\boldsymbol{C}_{n+1} = \begin{bmatrix} \boldsymbol{C}_n & \boldsymbol{k} \\ \boldsymbol{k}^T & c \end{bmatrix} \tag{22}$$

$$c = k(\boldsymbol{x}_{n+1}, \boldsymbol{x}_{n+1}) + \beta^{-1} \tag{23}$$

$$\boldsymbol{k} = [k(\boldsymbol{x}_1, \boldsymbol{x}_{n+1}), \ldots, k(\boldsymbol{x}_n, \boldsymbol{x}_{n+1})]^T \tag{24}$$

Step 2: Conditional distribution $p(\boldsymbol{t}_{n+1}|\boldsymbol{t}_n)$

$$p(t_{n+1}|\boldsymbol{t}_n, \boldsymbol{X}_n, \boldsymbol{x}_{n+1}) = \mathcal{N}(t_{n+1}|m(\boldsymbol{x}_{n+1}), \sigma^2(\boldsymbol{x}_{n+1})) \tag{25}$$

Partitioned Gaussian distributions

Two sets of variables

If two sets of variables are jointly Gaussian, then conditional distribution of one set conditioned on the other is also Gaussian. Marginal distribution of either set is also Gaussian.

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_a \\ \boldsymbol{x}_b \end{bmatrix}, \boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{bmatrix} \tag{26}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{bmatrix}, \boldsymbol{\Sigma} = \boldsymbol{\Sigma}^T, \boldsymbol{\Sigma}_{ab} = \boldsymbol{\Sigma}_{ba}^T \tag{27}$$

Partitioned Gaussians

Conditional distribution

$$p(\boldsymbol{x}_a|\boldsymbol{x}_b) = \mathcal{N}(\boldsymbol{x}_a|\boldsymbol{\mu}_{a|b}, \boldsymbol{\Sigma}_{a|b}) \tag{28}$$

$$\boldsymbol{\mu}_{a|b} = \boldsymbol{\mu}_a + \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}(\boldsymbol{x}_b - \boldsymbol{\mu}_b) \tag{29}$$

$$\boldsymbol{\Sigma}_{a|b} = \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba} \tag{30}$$

Predictive distribution - solution

$$p(t_{n+1}|\boldsymbol{t}_n, \boldsymbol{X}_n, \boldsymbol{x}_{n+1}) = \mathcal{N}(t_{n+1}|m(\boldsymbol{x}_{n+1}), \sigma^2(\boldsymbol{x}_{n+1})) \tag{31}$$

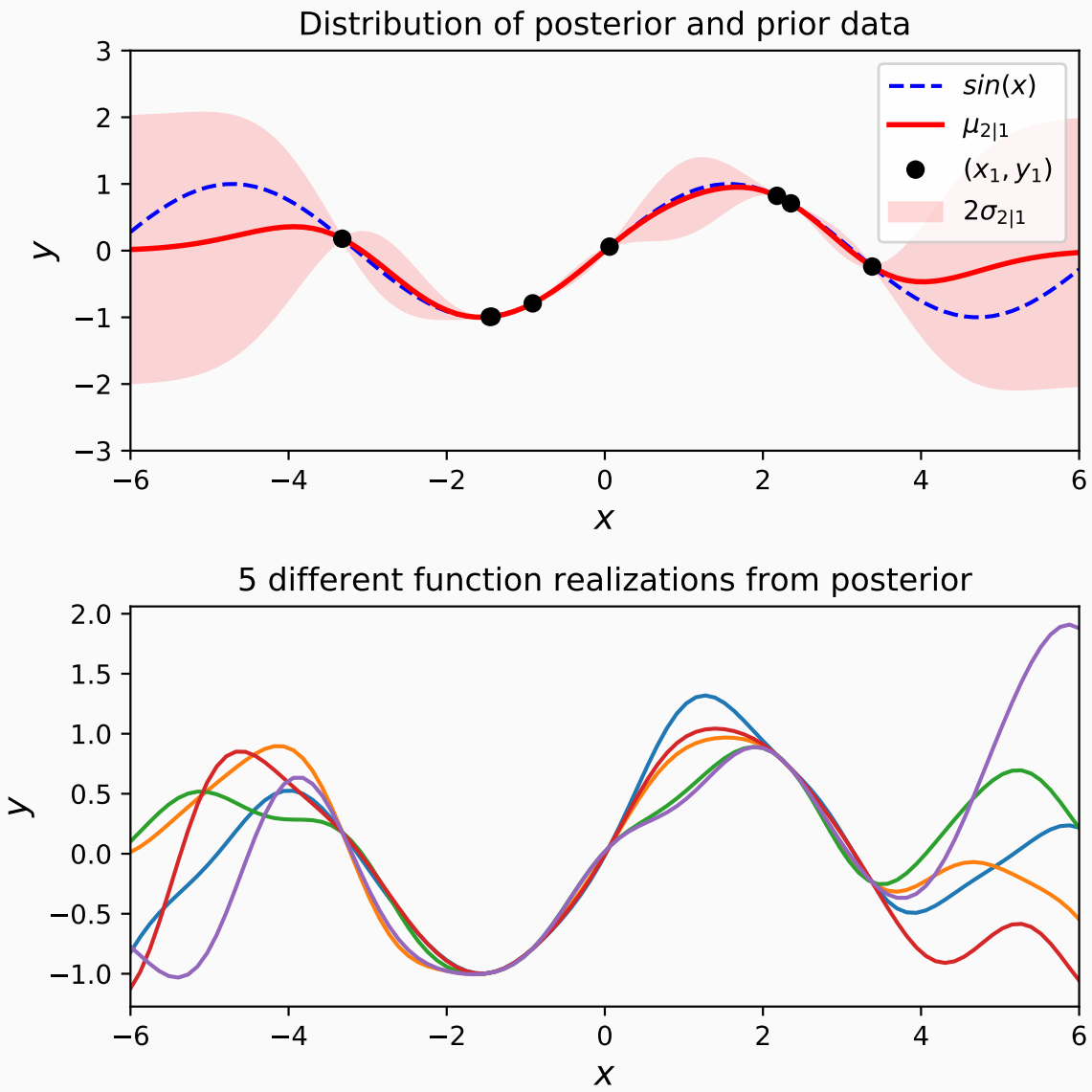
$$m(\boldsymbol{x}_{n+1}) = \boldsymbol{k}^T \boldsymbol{C}_n^{-1} \boldsymbol{t}_n \tag{32}$$

$$\sigma^2(\boldsymbol{x}_{n+1}) = c - \boldsymbol{k}^T \boldsymbol{C}_n^{-1} \boldsymbol{k}_n \tag{33}$$

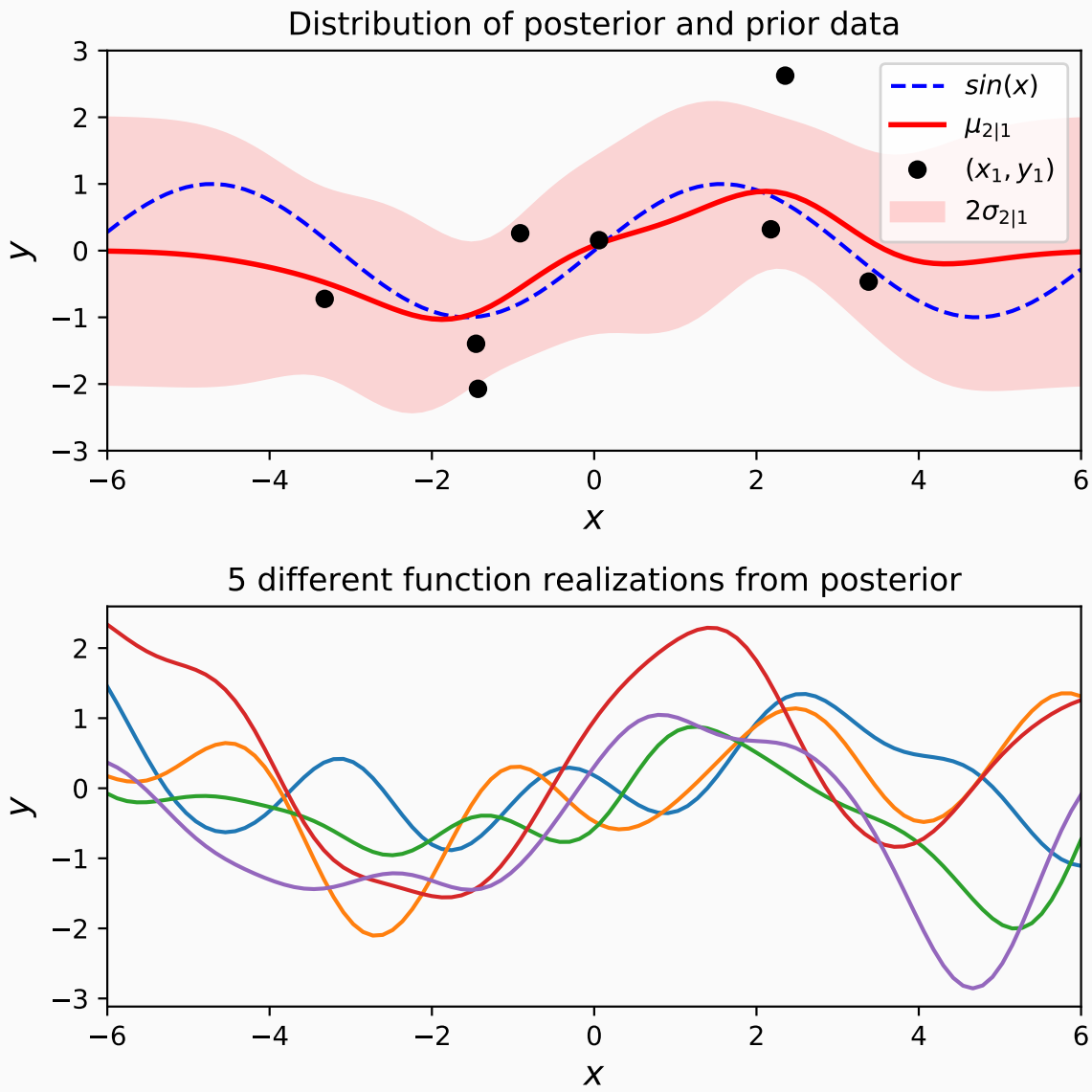
Gaussian process regression - algorithm

1. Collect training points
2. Define testing points in posterior (e.g. uniformly spaced to capture function)
3. Compute posterior mean and variance using Equation 32 and Equation 33
4. Compute standard deviations at tests points
5. Plot posterior distribution and samples from posterior (see [2])

Example 3: Gaussian process regression - noiseless distribution



Example 4: Gaussian process regression - noisy observations



Gaussian process kernels

Valid kernel

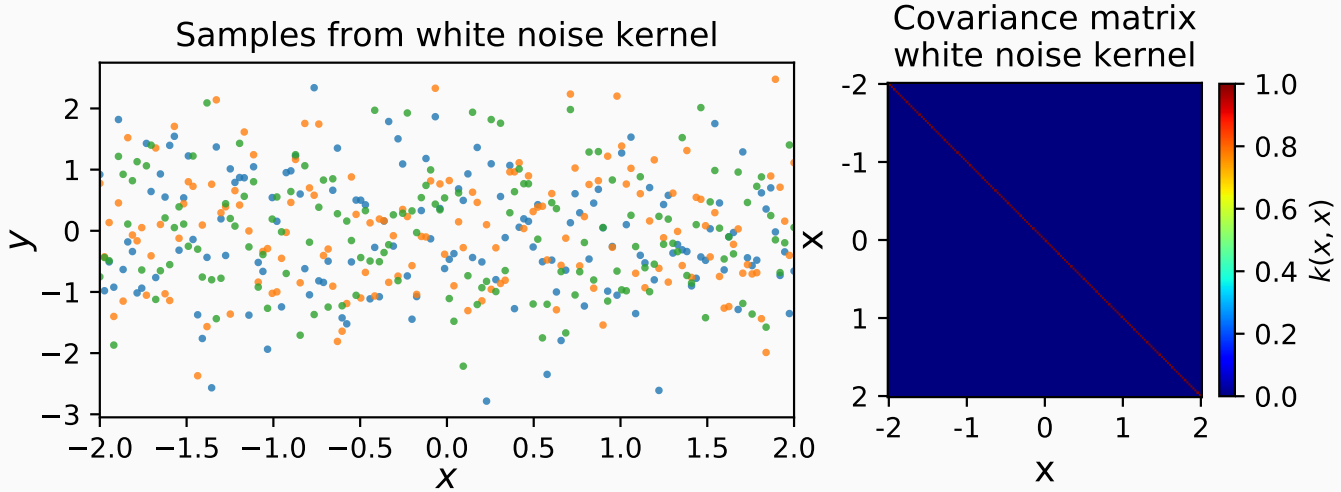
$$\Sigma = k(\boldsymbol{X}, \boldsymbol{X}) \tag{34}$$

- Positive definite $\boldsymbol{x}^T \Sigma \boldsymbol{x} > 0$ ($\boldsymbol{x}$ is non-zero vector)
- Σ is symmetric
- Σ is invertible

White noise kernel

$$k(\boldsymbol{x}_a, \boldsymbol{x}_b) = \beta^{-1} \boldsymbol{I}_n \tag{35}$$

- β - precision of the noise, $\beta^{-1} = \sigma^2$
- $\boldsymbol{I}_n$ - identity matrix
- Diagonal covariance matrix (noise is uncorrelated)



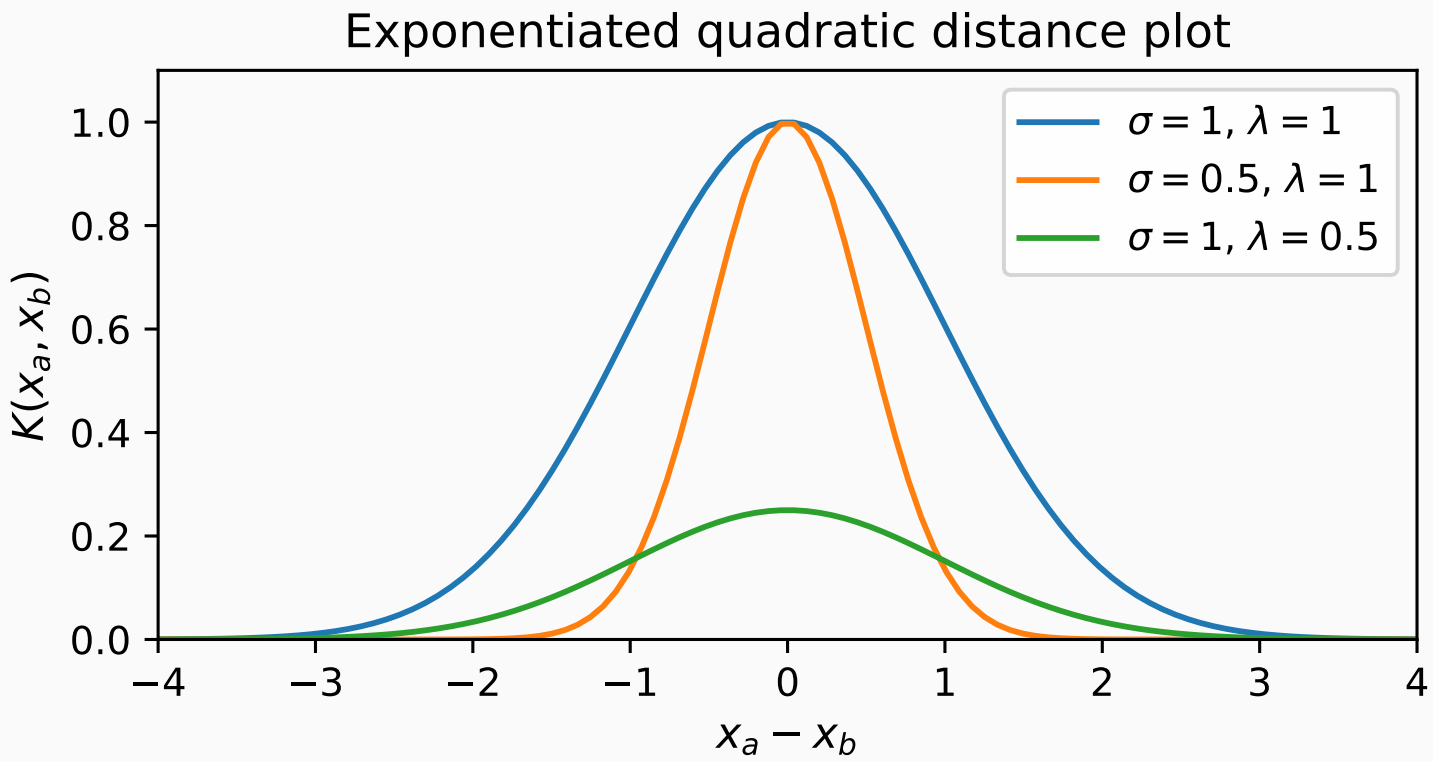
Exponential quadratic kernel

$$k(\boldsymbol{x}_a, \boldsymbol{x}_b) = \lambda^2 \exp\left(-\frac{\|\boldsymbol{x}_a - \boldsymbol{x}_b\|^2}{2\sigma^2}\right) \tag{36}$$

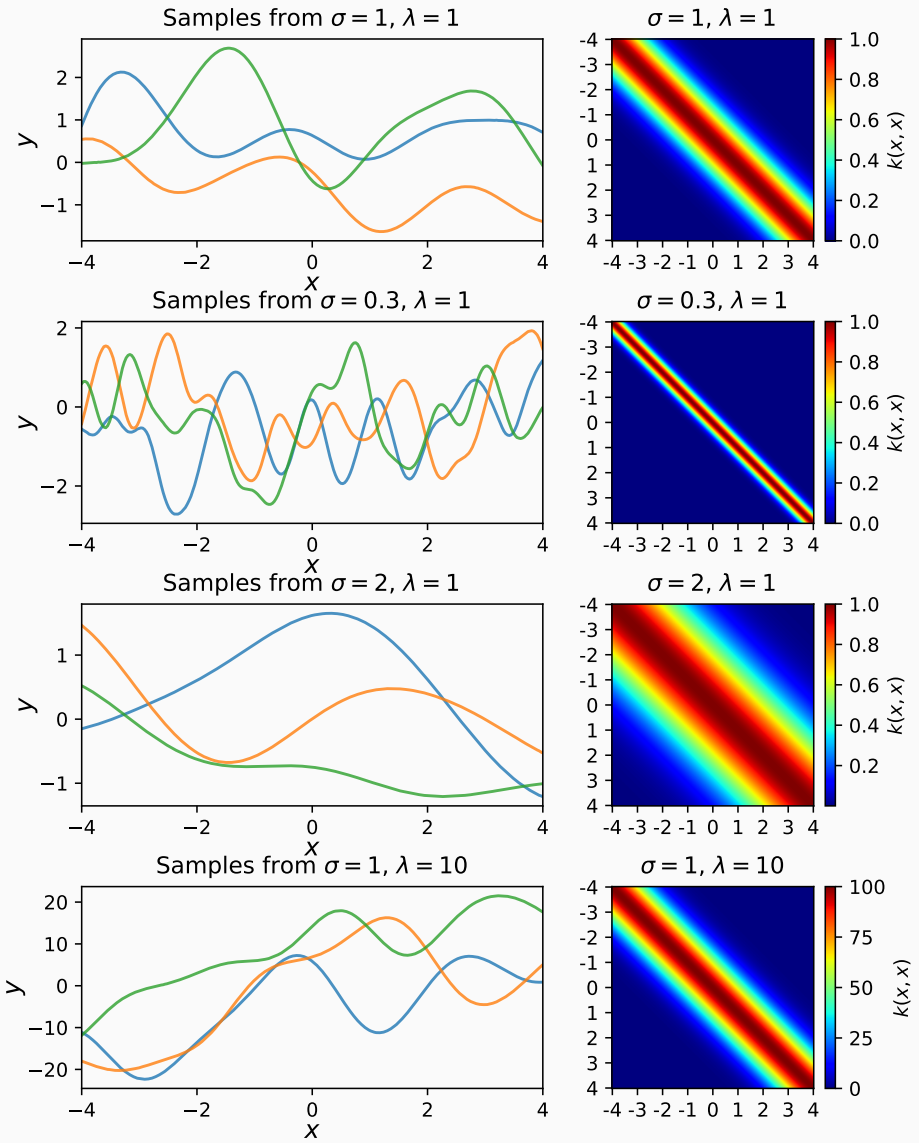
- λ^2 - overall variance (affects maximum value of the covariance)
- σ - length scale (the spread of the covariance)
- Smooth prior on functions sampled from Gaussian process

Exponential quadratic kernel - distance

$$x_a = 0$$



Exponential quadratic kernel - samples



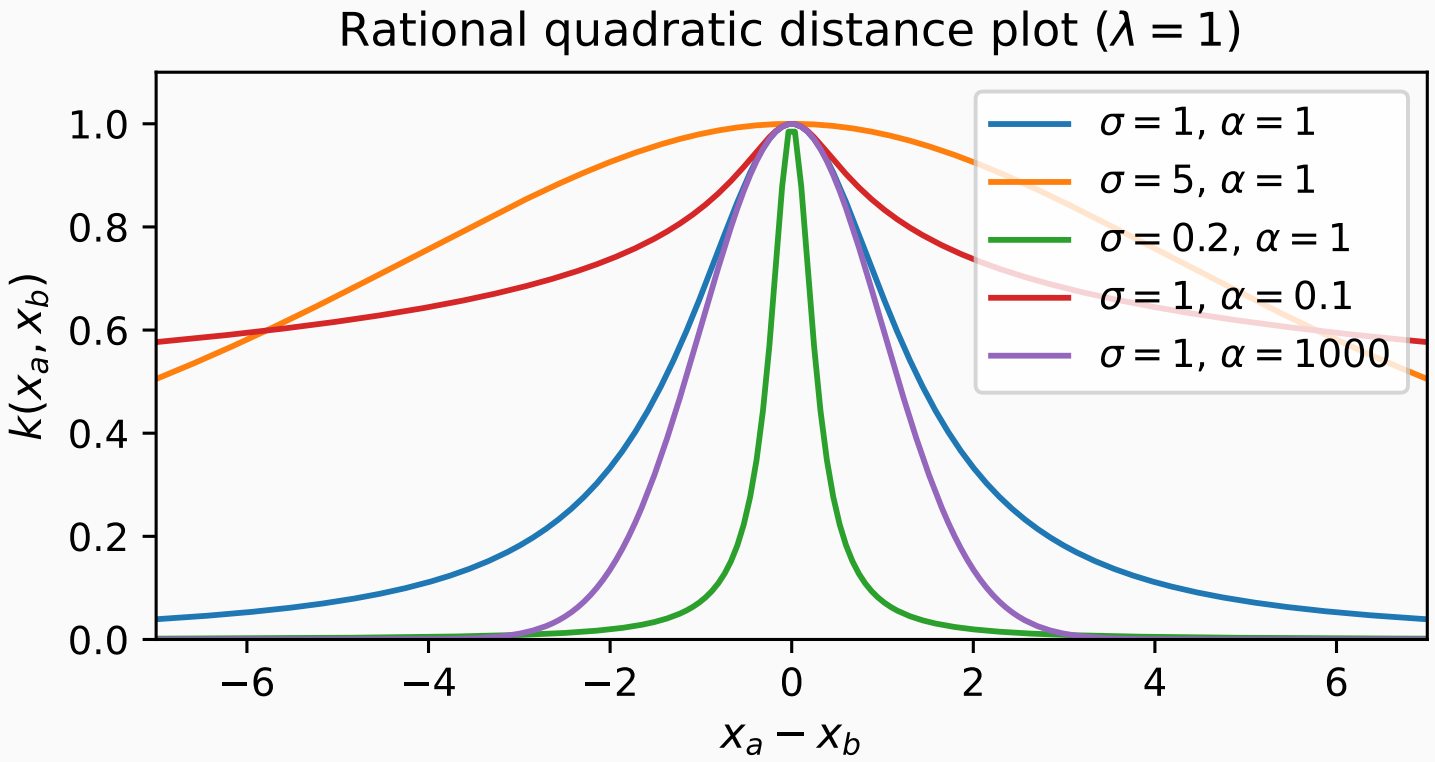
Rational quadratic kernel

$$k(\boldsymbol{x}_a, \boldsymbol{x}_b) = \lambda^2 \left(1 + \frac{\|\boldsymbol{x}_a - \boldsymbol{x}_b\|^2}{2\alpha\sigma^2}\right)^{-\alpha} \tag{37}$$

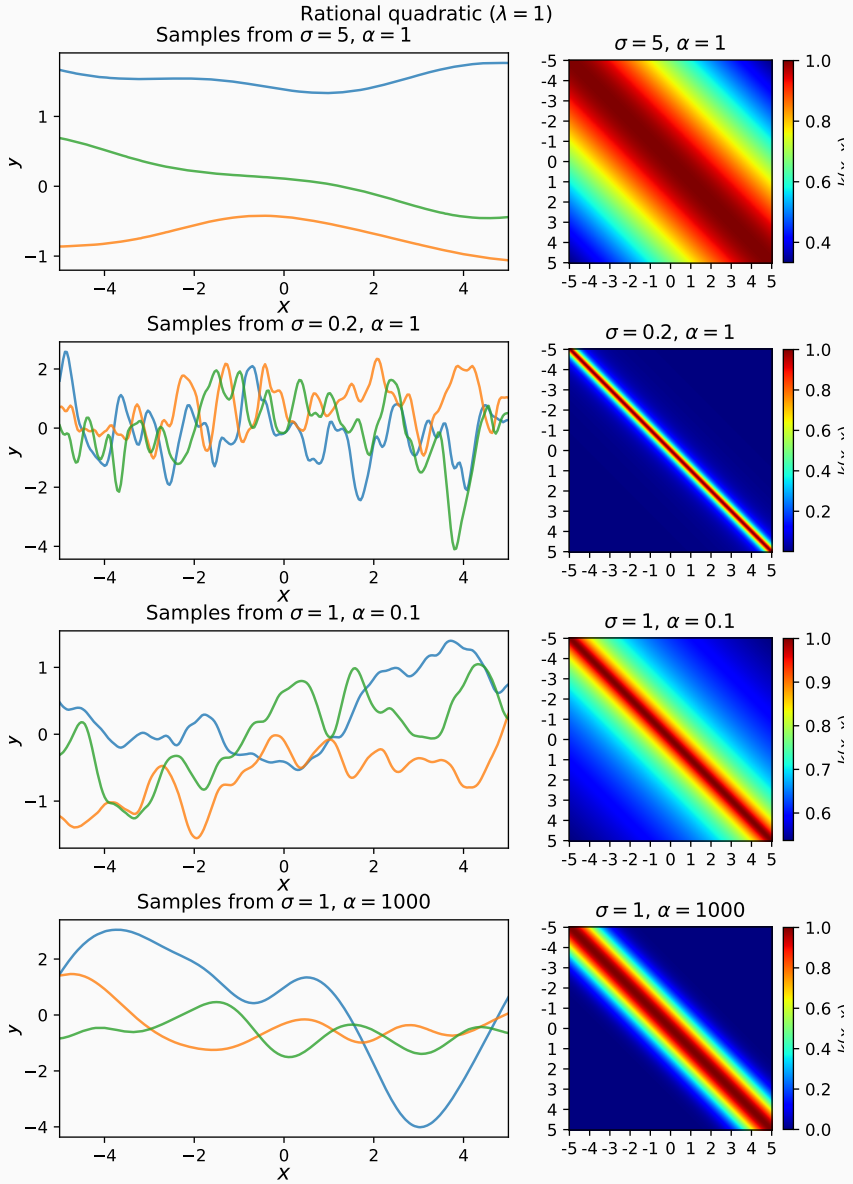
- λ^2 - overall variance, σ - length scale
- α - scale mixture ($\alpha > 0$)
- Infinite sum of different exponentiated quadratic kernels with different length scales and with α determining the weighting between different length scales

Rational quadratic kernel - distance

$$x_a = 0$$



Rational quadratic kernel - samples

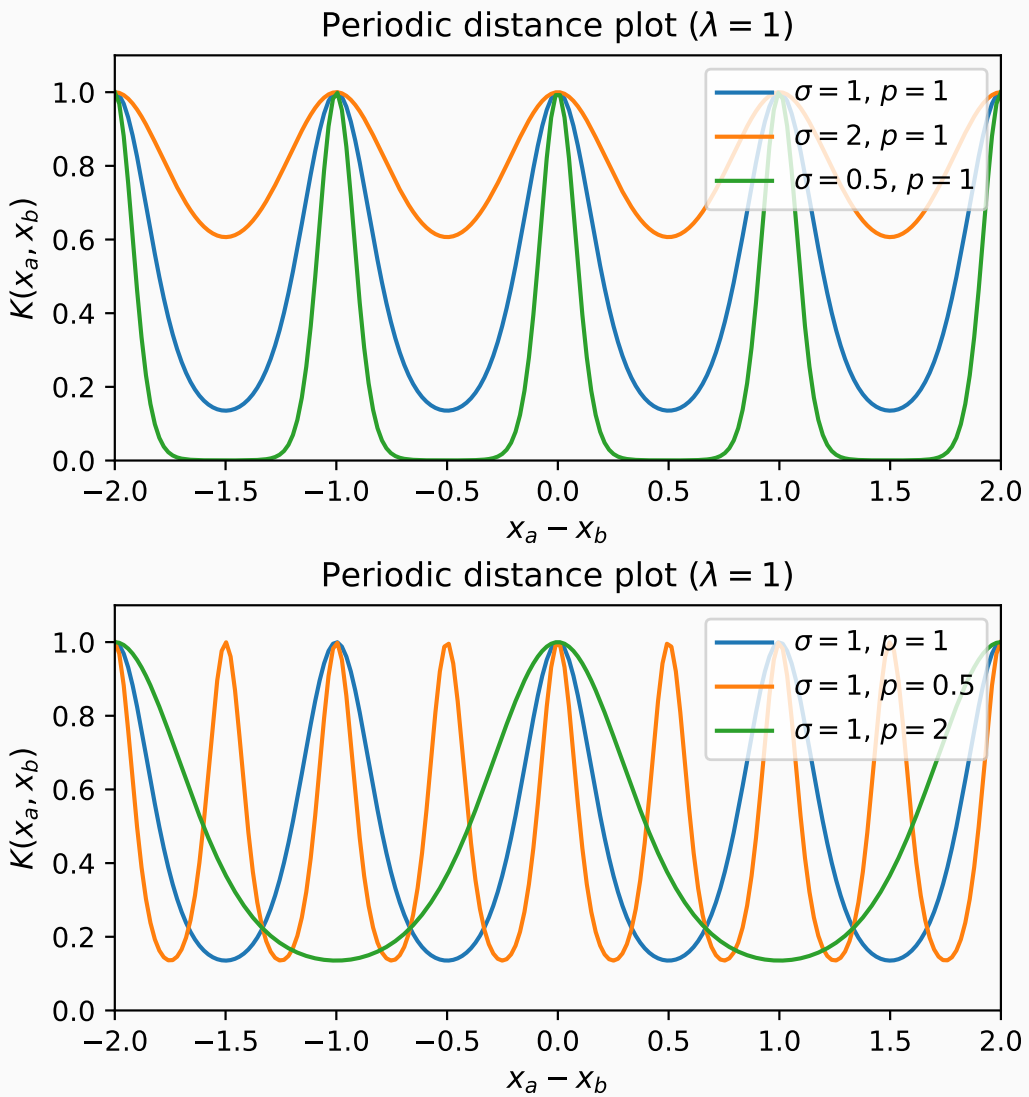


Periodic kernel

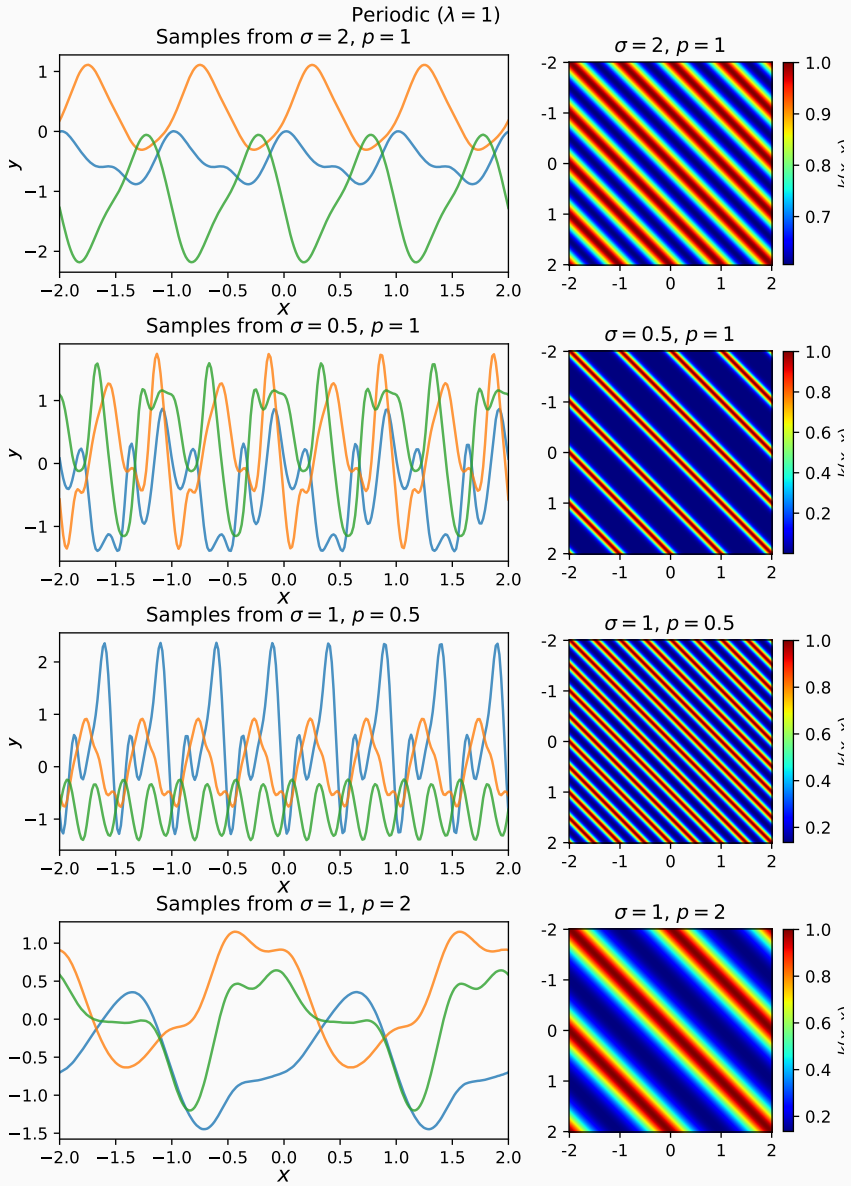
$$k(\boldsymbol{x}_a, \boldsymbol{x}_b) = \lambda^2 \exp\left(-\frac{2}{\sigma^2}\sin^2\left(\pi\frac{\|\boldsymbol{x}_a - \boldsymbol{x}_b\|}{p}\right)\right) \tag{38}$$

- λ^2 - overall variance (λ is amplitude), σ - the length scale
- p - the period (distance between repetitions)
- Allows for modeling periodic functions

Periodic kernel - distance



Periodic kernel - samples



Combining kernels by multiplication

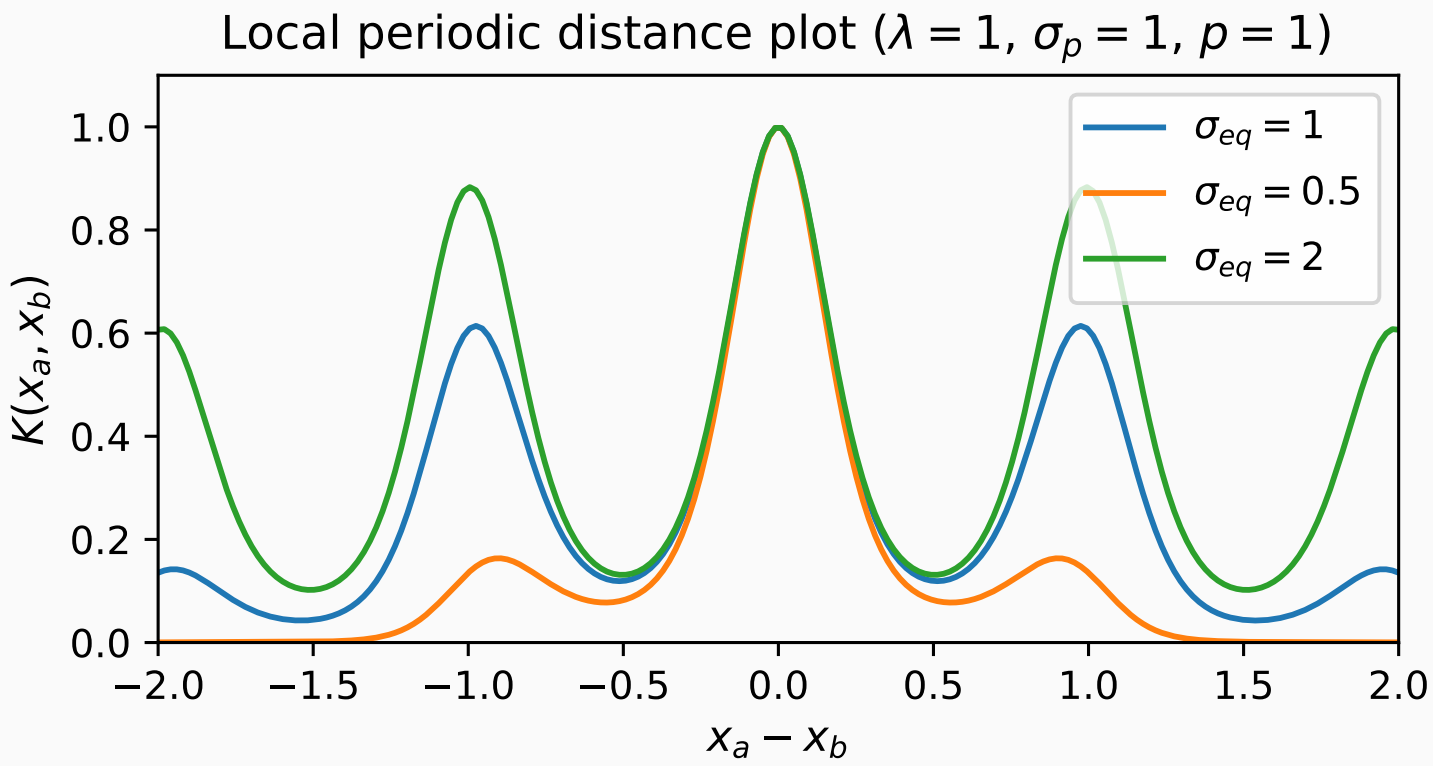
- Multiplying kernels is an element wise multiplication of their corresponding covariance matrices (AND operator)
- Allows for modeling functions that are only locally periodic - the shape of the repeating part of the function can change over time

Local periodic kernel

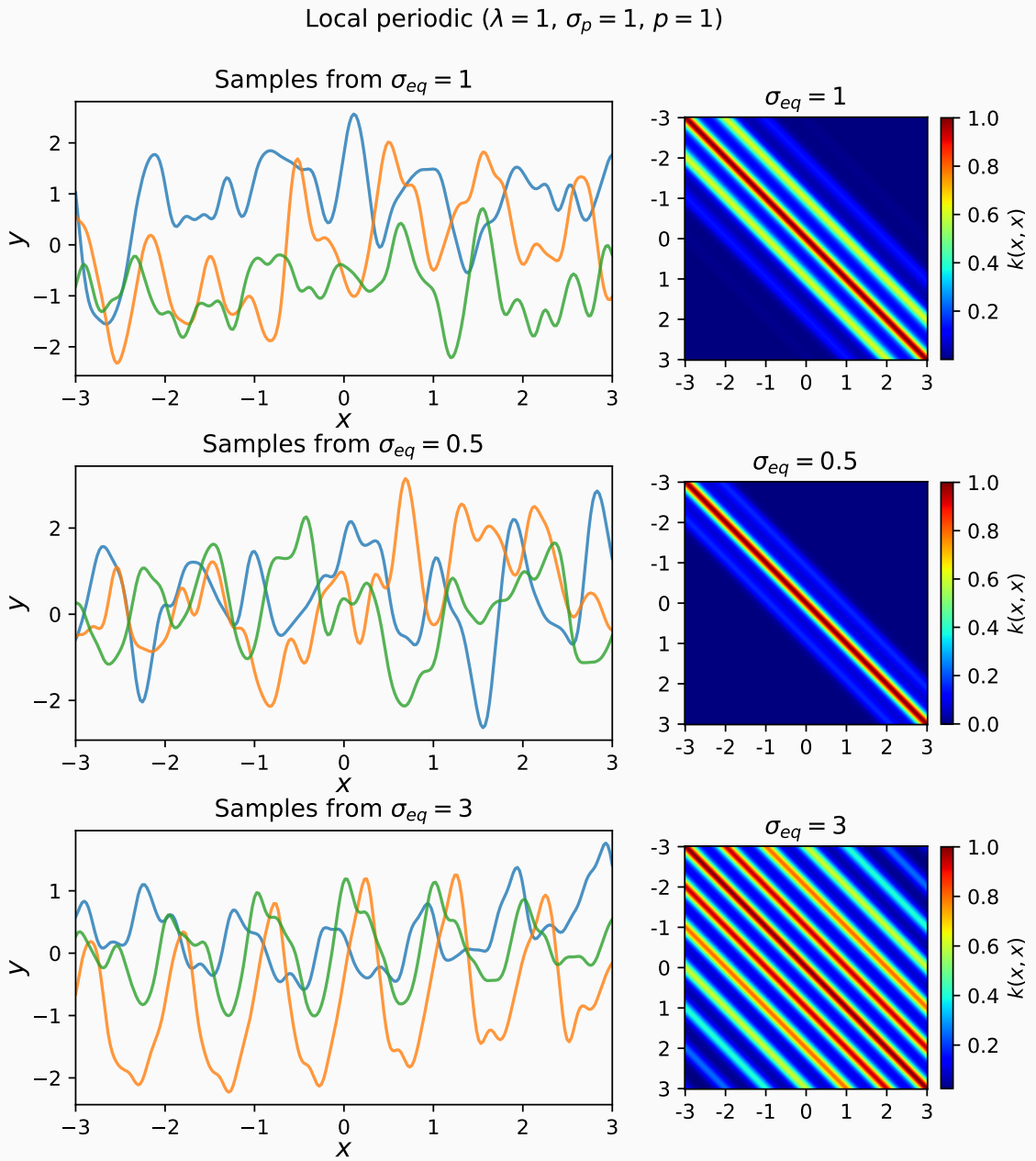
$$k(\boldsymbol{x}_a, \boldsymbol{x}_b) = \lambda^2 \exp\left(-\frac{2}{\sigma_p^2}\sin^2\left(\pi\frac{\|\boldsymbol{x}_a - \boldsymbol{x}_b\|}{p}\right)\right)\exp\left(-\frac{\|\boldsymbol{x}_a - \boldsymbol{x}_b\|^2}{2\sigma_{eq}^2}\right) \tag{39}$$

- λ^2 - overall variance (λ is amplitude), σ_p - the length scale of the periodic function, σ_{eq} - the length scale of the exponentiated quadratic function
- p - the period (distance between repetitions)
- Allows the periods to vary over longer distances

Local periodic kernel - distance



Local periodic kernel - samples



Combining kernels by addition

- Kernels can be combined by adding them together
- Adding kernels is an element wise addition of their corresponding covariance matrices (OR operator)

Fitting Gaussian process kernel

Create kernel combined by addition to reflect different characteristics of the data:

- Long term smooth change
- Seasonality
- Short term irregularities
- Medium term irregularities
- White noise

References


C. M. Bishop.
***Pattern Recognition and Machine Learning*.**
Springer, 2006.


P. Roelants.
***Understanding Gaussian processes*, Accessed November, 2020.**
Available at `https://peterroelants.github.io/posts/gaussian-process-tutorial/`.