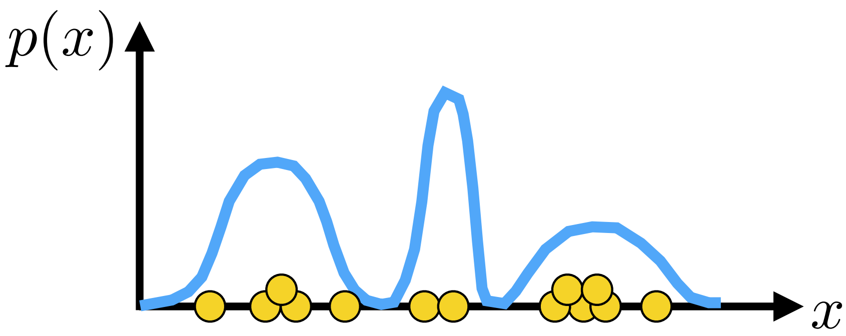
Nadia Chirkova

Gaussian processes

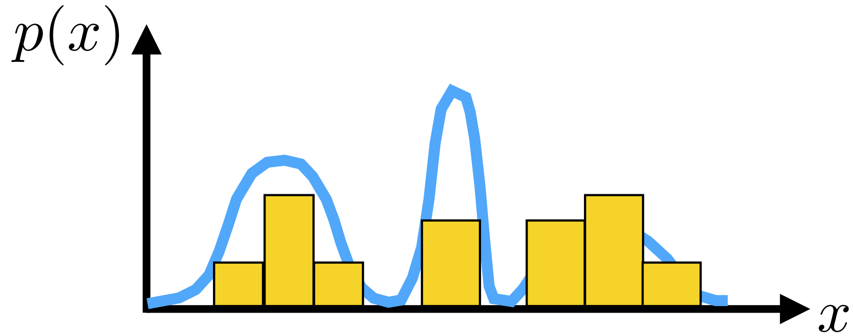
2021

Slides are partially based on lectures of Dmitry Vetrov, Dmitry Kropotov and Kirill Struminsky, deepbayes.ru/2018

Sampling points from a distribution



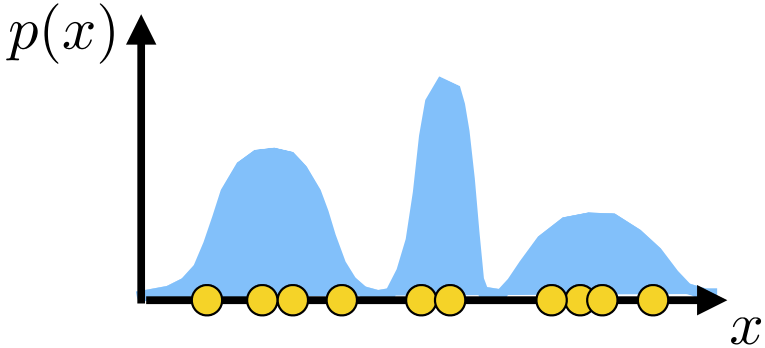
$x \sim p(x),\ x \in \mathbb{R}$



histogram approaches $p(x)$
when the number of points grows

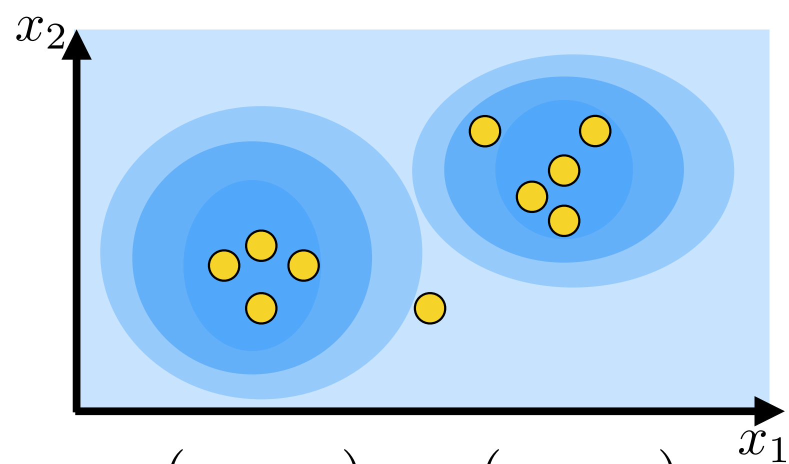
Sampling points from a multivariate distribution

1-dimensional distribution



$$x \sim p(x),\ x \in \mathbb{R}$$

2-dimensional distribution



$$(x_1, x_2) \sim p(x_1, x_2)$$
$$x \sim p(x),\ x \in \mathbb{R}^2$$

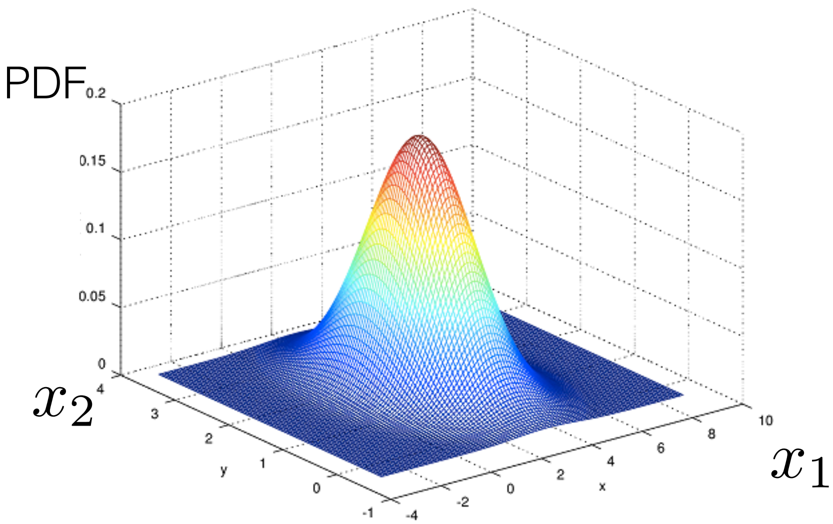
... n-dimensional distribution:

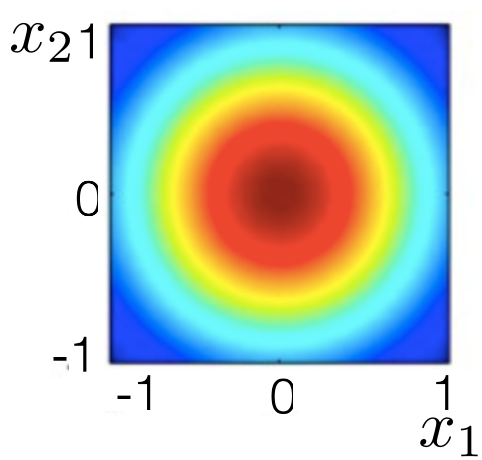
$$(x_1, \ldots, x_n) \sim p(x_1, \ldots, x_n)$$
$$x \sim p(x),\ x \in \mathbb{R}^n$$

Multivariate normal (Gaussian) distribution

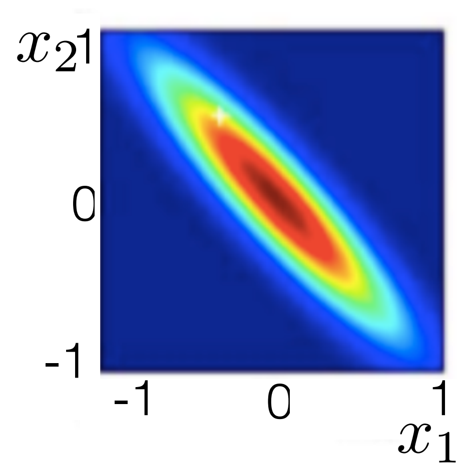
$$\mathcal{N}(\mu, \Sigma) = \frac{1}{\sqrt{\det(2\pi\Sigma)}} \exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)),$$

$$x \in \mathbb{R}^d$$
$$\mu \in \mathbb{R}^d$$
$$\Sigma \in \mathbb{R}^{d \times d}$$





diagonal Σ



non-diagonal Σ

Images from https://medium.com/ming-learns-thing/machine-learning-bayesian-linear-regression-f160c4eaef99, C. Bishop. Pattern Recognition and Machine Learning

Multivariate normal (Gaussian) distribution

$$\mathcal{N}(\mu, \Sigma) = \frac{1}{\sqrt{\det(2\pi\Sigma)}} \exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)),$$

$$x \in \mathbb{R}^d$$
$$\mu \in \mathbb{R}^d$$
$$\Sigma \in \mathbb{R}^{d \times d}$$

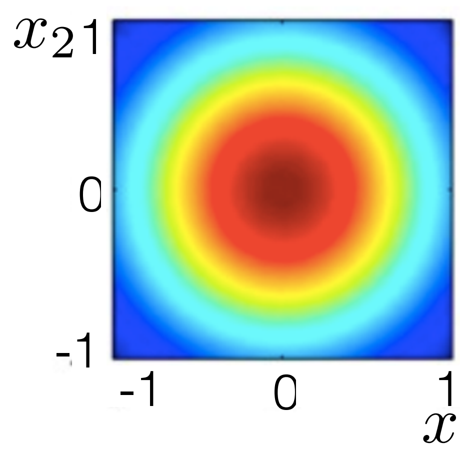
$\mathrm{Var}(x_1)$

$\mathrm{Cov}(x_1, x_2)$

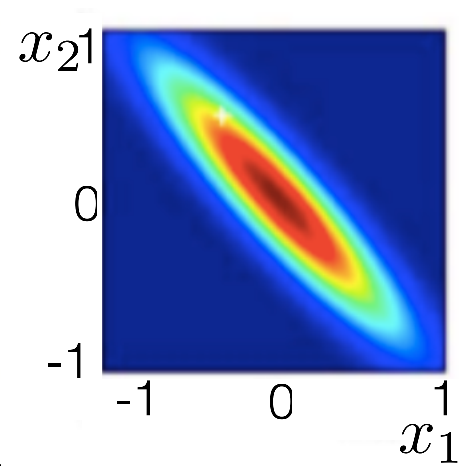
$$\Sigma \quad \boxed{\begin{matrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{12} & \Sigma_{22} \end{matrix}}$$

diagonal $\Sigma \Rightarrow$

independent $x_1, \ldots x_d$



diagonal Σ

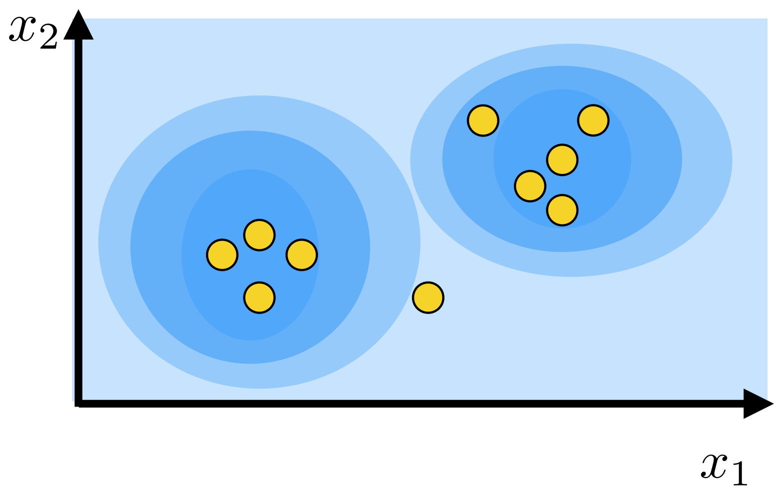


non-diagonal Σ

Images from https://medium.com/ming-learns-thing/machine-learning-bayesian-linear-regression-f160c4eaef99, C. Bishop. Pattern Recognition and Machine Learning

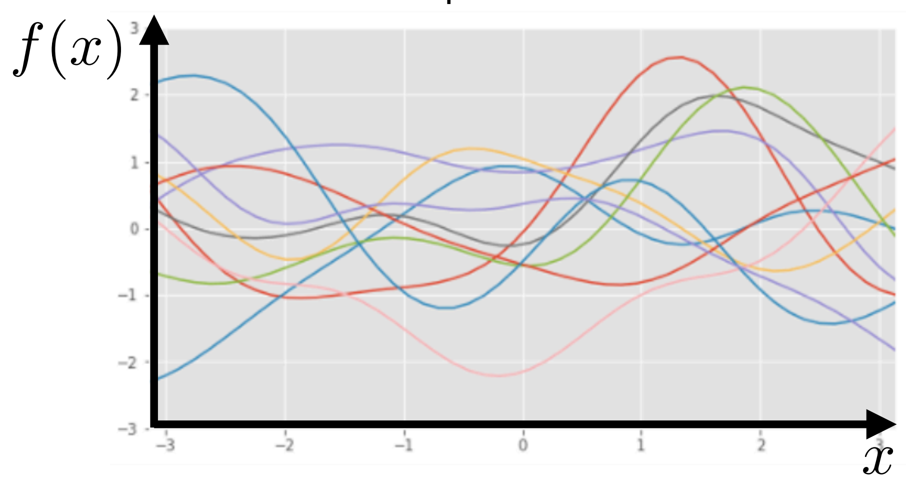
Sampling functions from a process

Multivariate distribution
— sample **points**



$$x \sim p(x)$$

Process
— sample **functions**



$$f(x) \sim GP(m(x), k(x, x'))$$

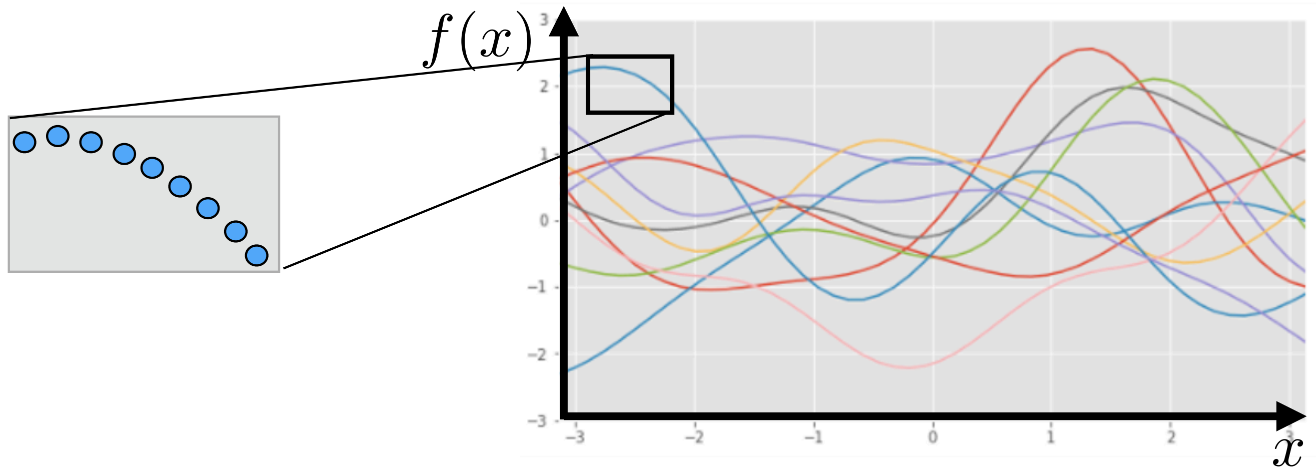
each line is a sample from the process

Right image from https://towardsdatascience.com/an-intuitive-guide-to-gaussian-processes-ec2f0b45c71d

Sampling functions from a process

When we plot
a function in python,
we define a function as
a sequence of points

Process
— sample **function**



$$f(x) \sim GP(m(x), k(x, x'))$$

each line is a sample from the process

Right image from https://towardsdatascience.com/an-intuitive-guide-to-gaussian-processes-ec2f0b45c71d

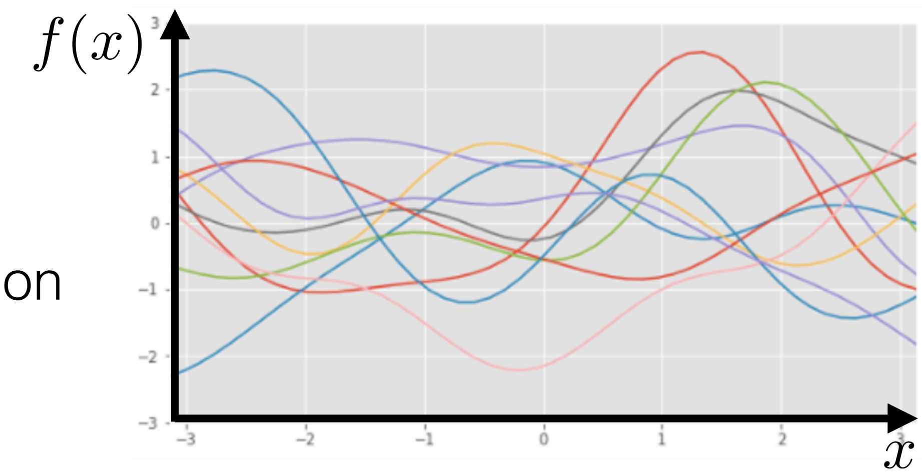
Gaussian process

$f(x) \sim GP(m(x), k(x, x'))$

$m(x)$ — mean function

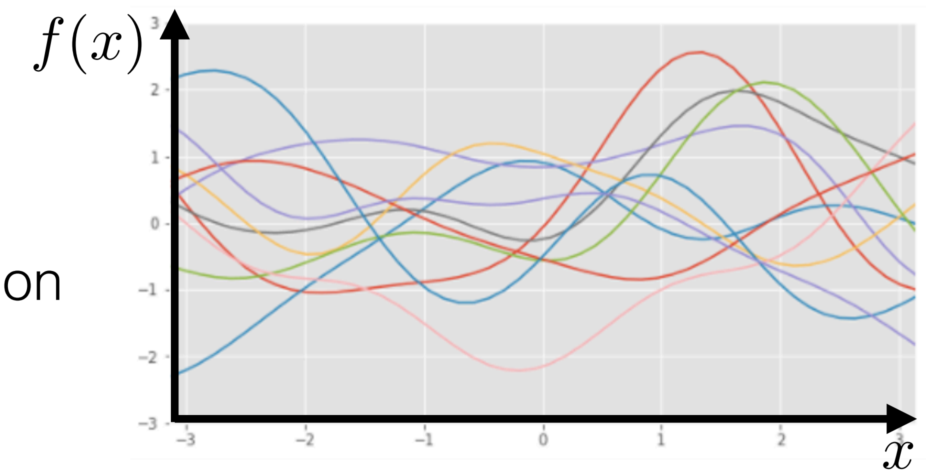
$k(x, x')$ — covariance (or kernel) function

(x may be a vector in general case)



Gaussian process

$f(x) \sim GP(m(x), k(x, x'))$



$m(x)$ — mean function

$k(x, x')$ — covariance (or kernel) function

(x may be a vector in general case)

Definition of Gaussian process:
every finite set of function values has a multivariate normal distribution

$$\forall n \quad \forall (x_1, \dots, x_n) \quad (f(x_1), \dots, f(x_n)) \sim \mathcal{N}(\mu, \Sigma)$$

$$\mu = \{m(x_i)\}_{i=1}^{n} \quad \Sigma = \{k(x_i, x_j)\}_{i,j=1}^{n,n}$$

Example 1: white noise

$$f(x) \sim GP(m(x), k(x, x'))$$

$$m(x) = 0$$

$$k(x, x') = \sigma^2 [x = x']$$

$[\text{condition}] = 1$ if condition is True else 0

Example 1: white noise

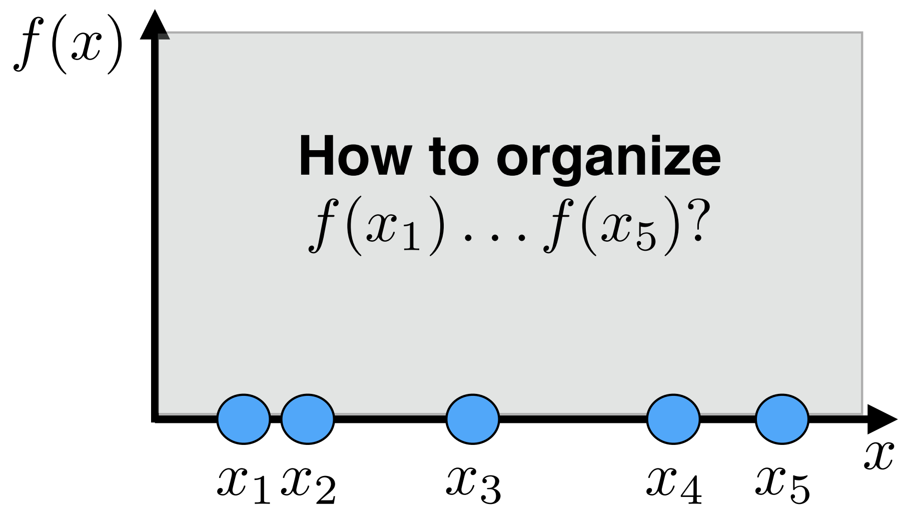
$f(x) \sim GP(m(x), k(x, x'))$

$m(x) = 0$

$k(x, x') = \sigma^2[x = x']$
$[\text{condition}] = 1$ if condition is True else 0

$(f(x_1), \ldots, f(x_n)) \sim \mathcal{N}(\mu, \Sigma)$

$\mu = 0 \qquad \Sigma = \sigma^2 I$



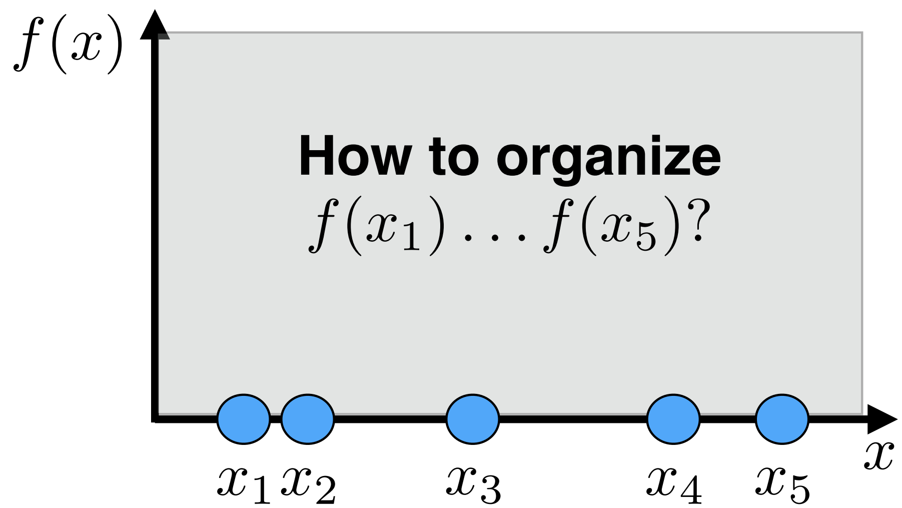
Example 1: white noise

$$f(x) \sim GP(m(x), k(x, x'))$$

$$m(x) = 0$$

$$k(x, x') = \sigma^2[x = x']$$

$[\text{condition}] = 1$ if condition is True else 0



$$(f(x_1), \dots, f(x_n)) \sim \mathcal{N}(\mu, \Sigma)$$

$$\mu = 0 \qquad \Sigma = \sigma^2 I$$

$$p(f(x_1), \dots, f(x_n)) = \prod_{i=1}^{n} \mathcal{N}(0, \sigma^2)$$

(all x are independent on each other)

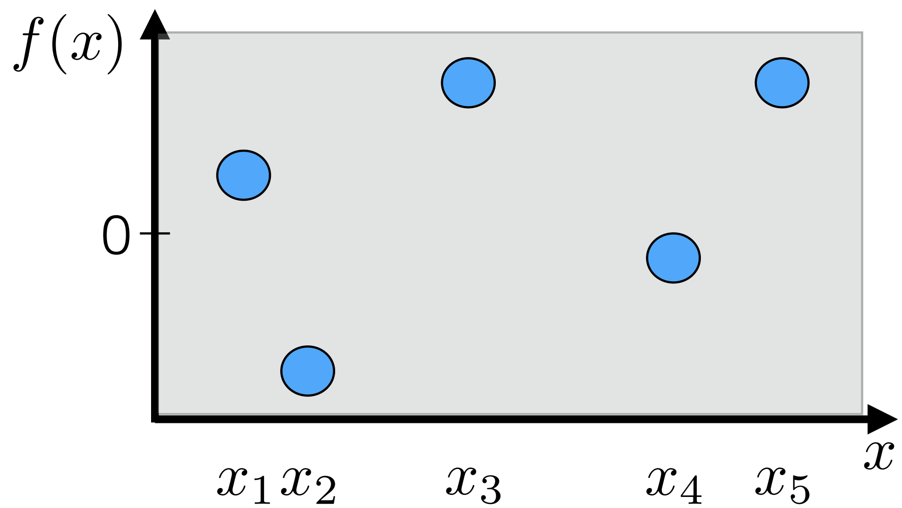
Example 1: white noise

$f(x) \sim GP(m(x), k(x,x'))$

$m(x) = 0$

$k(x,x') = \sigma^2[x = x']$

[condition] = 1 if condition is True else 0



$(f(x_1), \ldots, f(x_n)) \sim \mathcal{N}(\mu, \Sigma)$

$\mu = 0 \qquad \Sigma = \sigma^2 I$

$p(f(x_1), \ldots, f(x_n)) = \prod_{i=1}^{n} \mathcal{N}(0, \sigma^2)$

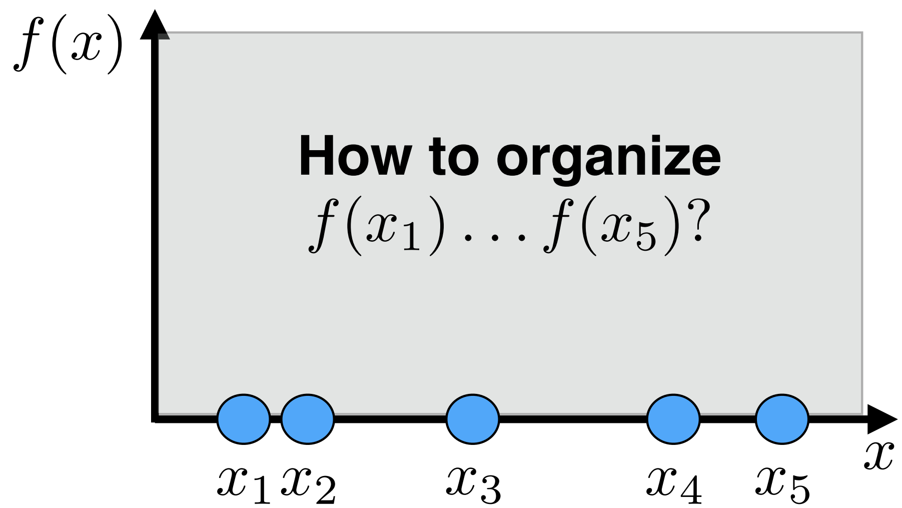
(all x are independent on each other)

for any x
$f(x)$ is sampled
independently

Example 2: constant function

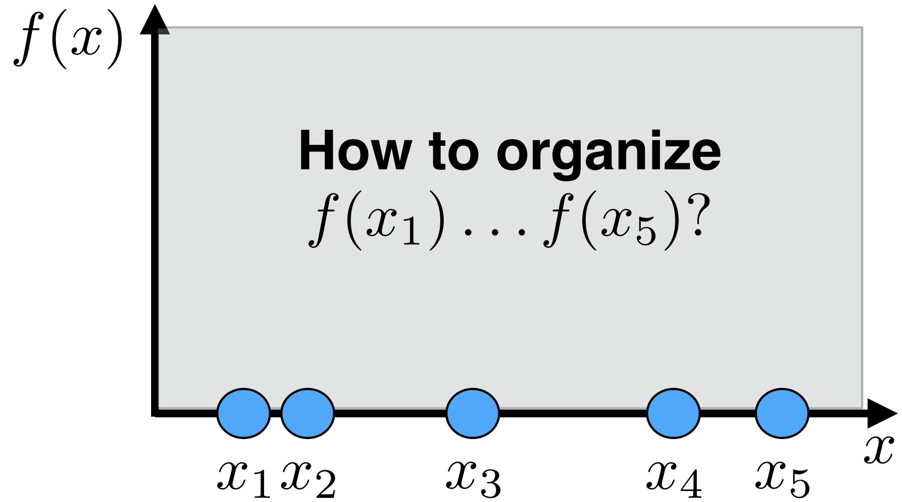
$$f(x) \sim GP(m(x), k(x, x'))$$
$$m(x) = 0$$
$$k(x, x') = \mathrm{C}$$

$$(f(x_1), \ldots, f(x_n)) \sim \mathcal{N}(\mu, \Sigma)$$
$$\mu = 0 \qquad \Sigma = \{C\}_{i,j=1}^{n,n}$$



Example 2: constant function

$$f(x) \sim GP(m(x), k(x, x'))$$
$$m(x) = 0$$
$$k(x, x') = \mathrm{C}$$

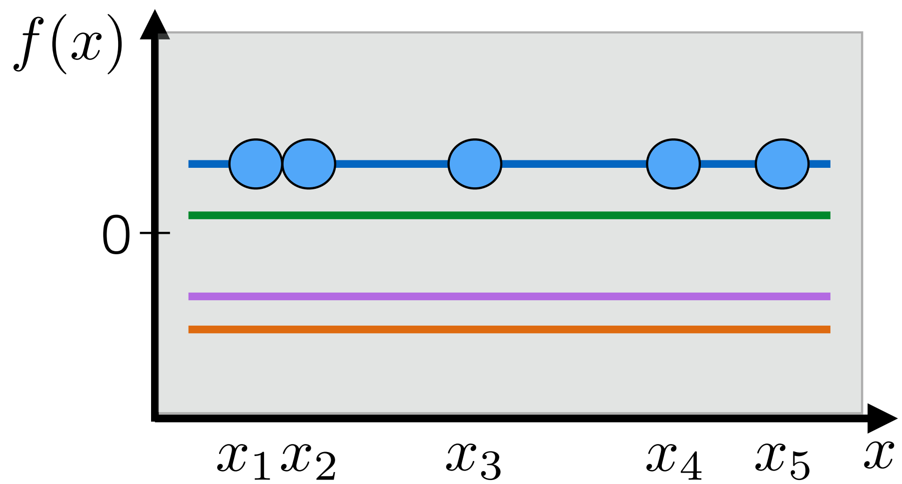


$$(f(x_1), \dots, f(x_n)) \sim \mathcal{N}(\mu, \Sigma)$$
$$\mu = 0 \qquad \Sigma = \{C\}_{i,j=1}^{n,n}$$

$$\left.\begin{array}{l} \forall i \neq j \\ \mathrm{Corr}(f(x_i), f(x_j)) = \dfrac{\mathrm{Cov}(f(x_i), f(x_j))}{\sqrt{\mathrm{Var}(f(x_i))\mathrm{Var}(f(x_j))}} = \dfrac{C}{\sqrt{C}^2} = 1 \\ \mathrm{Var}(f(x_i)) = \mathrm{Var}(f(x_j)) = C, \quad \mathbb{E}f(x_i) = \mathbb{E}f(x_j) = 0 \end{array}\right] \Rightarrow f(x_i) = f(x_j)$$

Example 2: constant function

$$f(x) \sim GP(m(x), k(x, x'))$$
$$m(x) = 0$$
$$k(x, x') = \mathrm{C}$$



$$(f(x_1), \ldots, f(x_n)) \sim \mathcal{N}(\mu, \Sigma)$$
$$\mu = 0 \qquad \Sigma = \{C\}_{i,j=1}^{n,n}$$

$$\left.\begin{aligned} &\forall i \neq j \\ &\mathrm{Corr}(f(x_i), f(x_j)) = \frac{\mathrm{Cov}(f(x_i), f(x_j))}{\sqrt{\mathrm{Var}(f(x_i))\mathrm{Var}(f(x_j))}} = \frac{C}{\sqrt{C}^2} = 1 \\ &\mathrm{Var}(f(x_i)) = \mathrm{Var}(f(x_j)) = C, \quad \mathbb{E}f(x_i) = \mathbb{E}f(x_j) = 0 \end{aligned}\right] \Rightarrow f(x_i) = f(x_j)$$

Example 3: RBF-kernel

$$f(x) \sim GP(m(x), k(x, x'))$$

$$m(x) = 0$$

$$k(x, x') = \sigma^2 \exp\left(-\frac{(x - x')^2}{2\ell^2}\right)$$

$$(f(x_1), \ldots, f(x_n)) \sim \mathcal{N}(\mu, \Sigma)$$

$$\mu = 0 \qquad \Sigma = \{k(x_i, x_j)\}_{i,j=1}^{n,n}$$

Example 3: RBF-kernel

$$f(x) \sim GP(m(x), k(x, x'))$$
$$m(x) = 0$$
$$k(x, x') = \sigma^2 \exp\left(-\frac{(x - x')^2}{2\ell^2}\right)$$

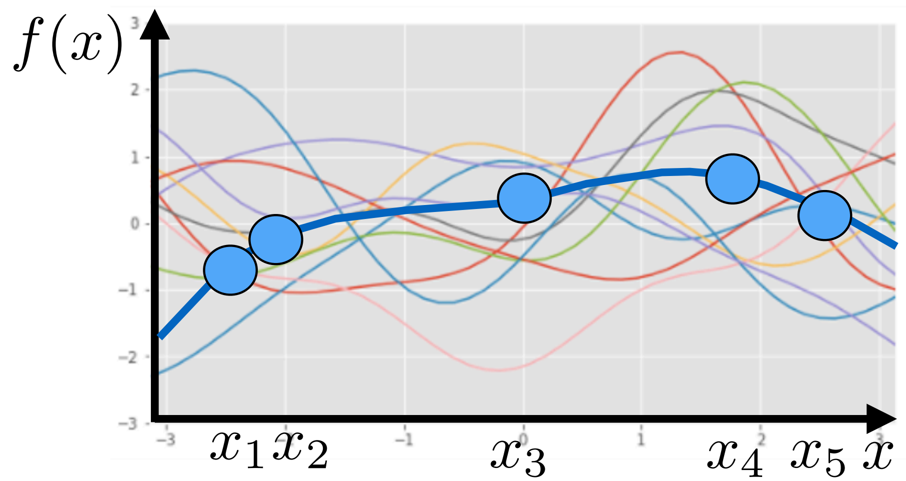
$$(f(x_1), \ldots, f(x_n)) \sim \mathcal{N}(\mu, \Sigma)$$
$$\mu = 0 \qquad \Sigma = \{k(x_i, x_j)\}_{i,j=1}^{n,n}$$

$$\text{if } \|x_i - x_j\| \approx 0 \quad \Rightarrow \quad \Sigma_{ij} \approx \sigma^2 = \Sigma_{ii} = \Sigma_{jj} \quad \Rightarrow \quad f(x_i) \approx f(x_j)$$

$$\text{if } \|x_i - x_j\| \gg 0 \quad \Rightarrow \quad \Sigma_{ij} \approx 0, \;\; f(x_i) \text{ and } f(x_j) \text{ are not correlated}$$

Example 3: RBF-kernel

$$f(x) \sim GP(m(x), k(x, x'))$$
$$m(x) = 0$$
$$k(x, x') = \sigma^2 \exp\left(-\frac{(x - x')^2}{2\ell^2}\right)$$



$$(f(x_1), \ldots, f(x_n)) \sim \mathcal{N}(\mu, \Sigma)$$
$$\mu = 0 \qquad \Sigma = \{k(x_i, x_j)\}_{i,j=1}^{n,n}$$

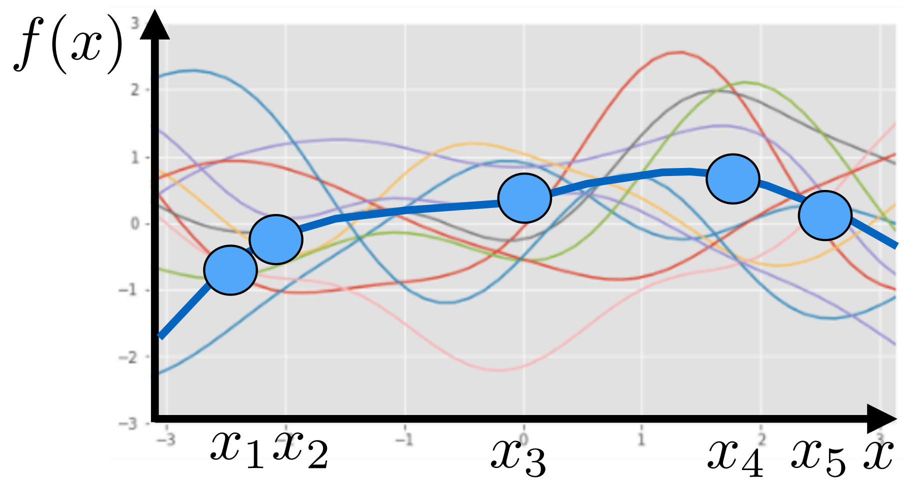
$$\text{if } \|x_i - x_j\| \approx 0 \quad \Rightarrow \quad \Sigma_{ij} \approx \sigma^2 = \Sigma_{ii} = \Sigma_{jj} \quad \Rightarrow \quad f(x_i) \approx f(x_j)$$

$$\text{if } \|x_i - x_j\| \gg 0 \quad \Rightarrow \quad \Sigma_{ij} \approx 0,\ \ f(x_i) \text{ and } f(x_j) \text{ are not correlated}$$

Example 3: RBF-kernel

$$f(x) \sim GP(m(x), k(x, x'))$$
$$m(x) = 0$$
$$k(x, x') = \sigma^2 \exp\left(-\frac{(x - x')^2}{2\ell^2}\right)$$

$$(f(x_1), \ldots, f(x_n)) \sim \mathcal{N}(\mu, \Sigma)$$
$$\mu = 0 \qquad \Sigma = \{k(x_i, x_j)\}_{i,j=1}^{n,n}$$



σ^2 defines the "height" of the function
ℓ^2 defines the frequency of fluctuations

Example 3: RBF-kernel

ℓ^2 defines the frequency of fluctuations

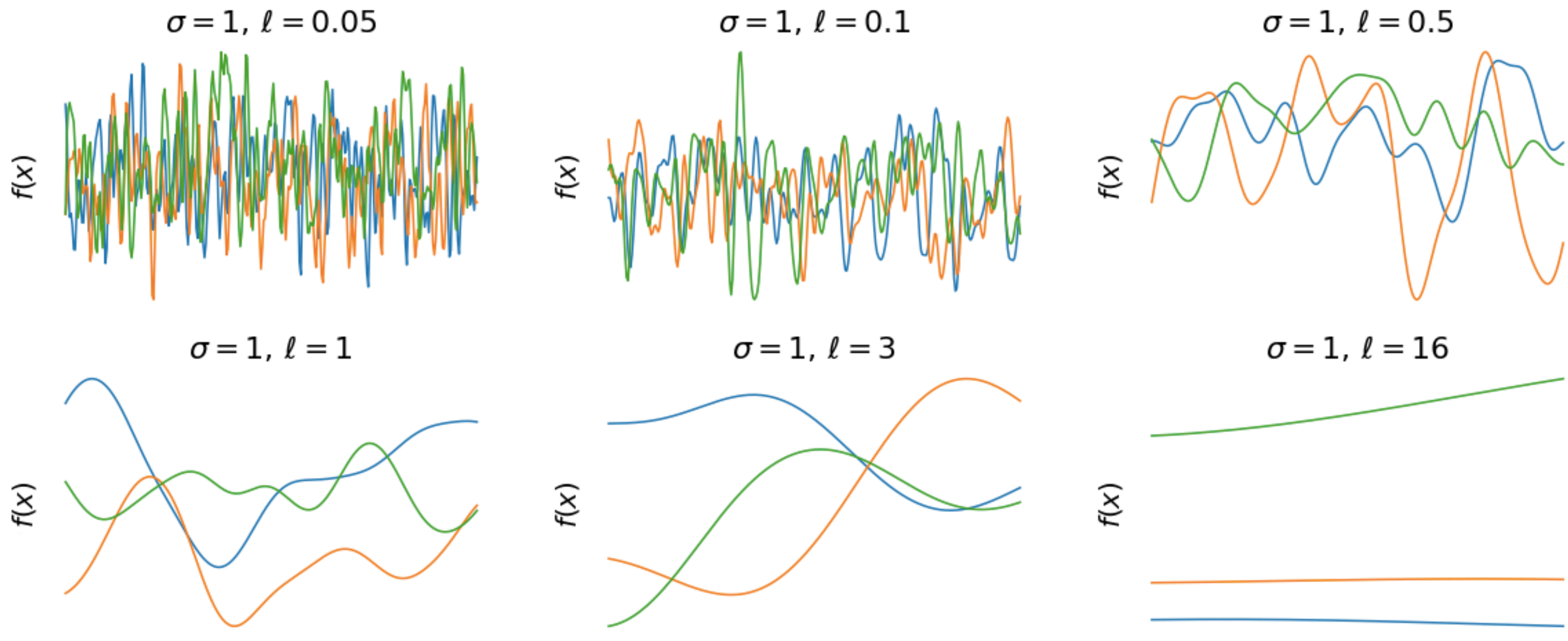


Image from https://relguzman.blogspot.com/2018/03/gaussian-processes-explained.html

More kernels

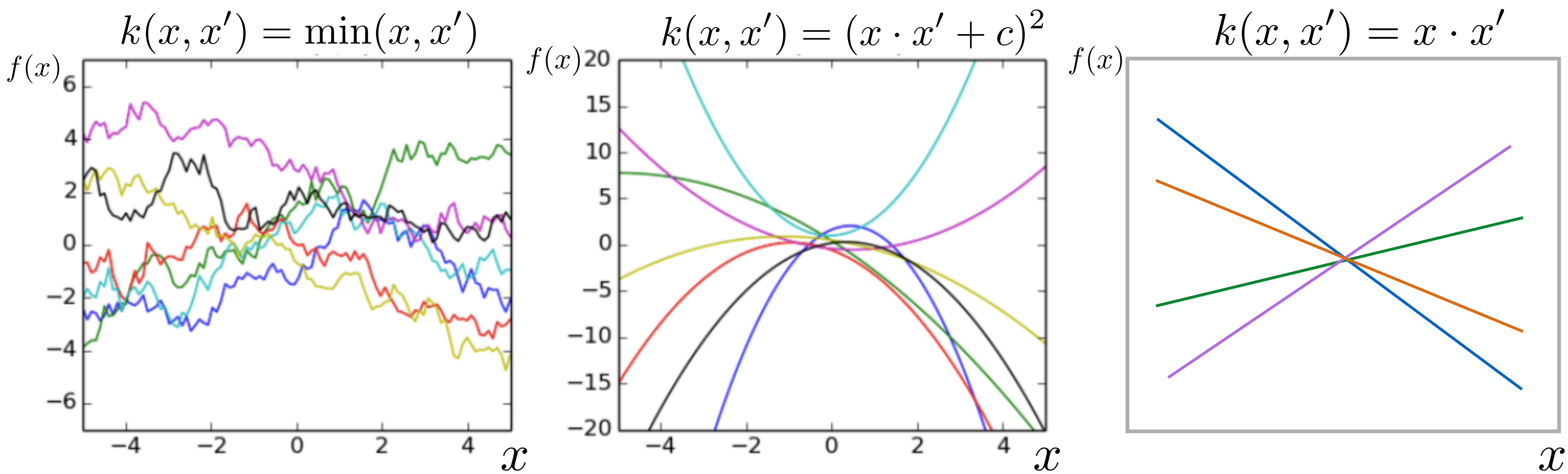
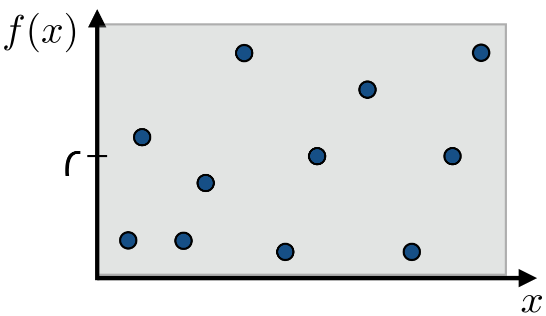
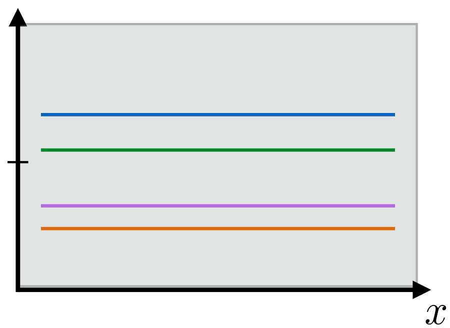


Image from https://www.kaggle.com/residentmario/gaussian-process-regression-and-classification

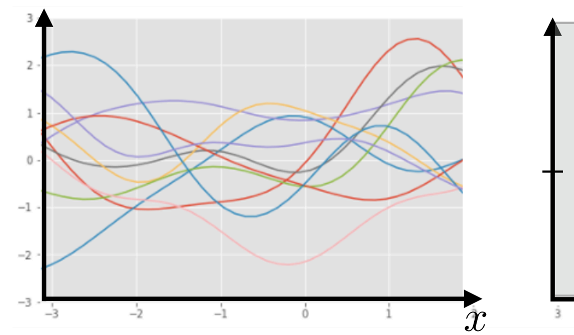
Sum-kernel



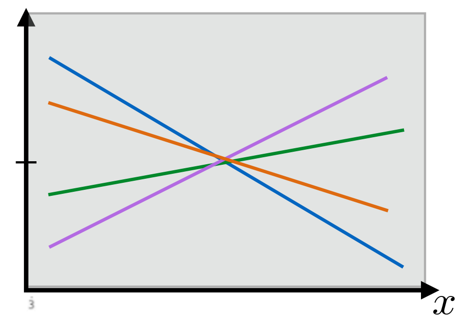
$$k(x,x') = \sigma^2[x = x']$$



$$k(x,x') = \mathrm{C}$$



$$k(x,x') = \sigma^2 \exp\left(-\frac{(x-x')^2}{2r^2}\right)$$

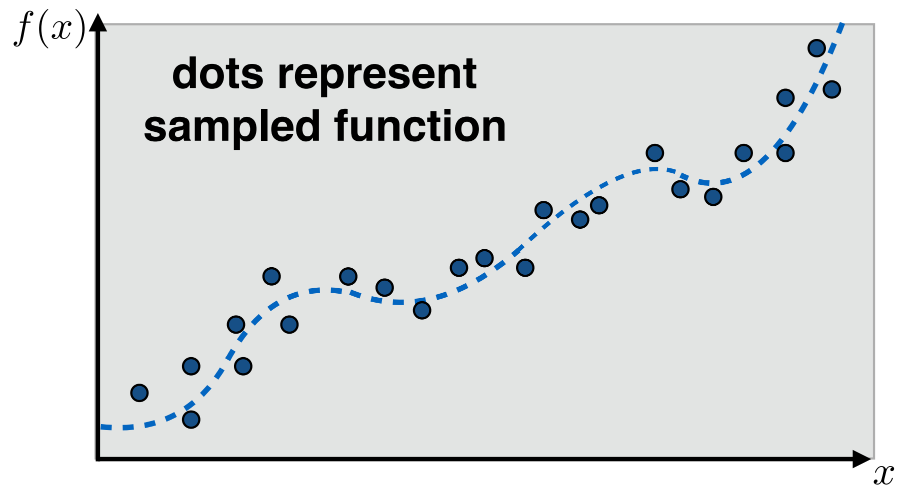


$$k(x,x') = x \cdot x'$$

Sum-kernel (multidimensional case):

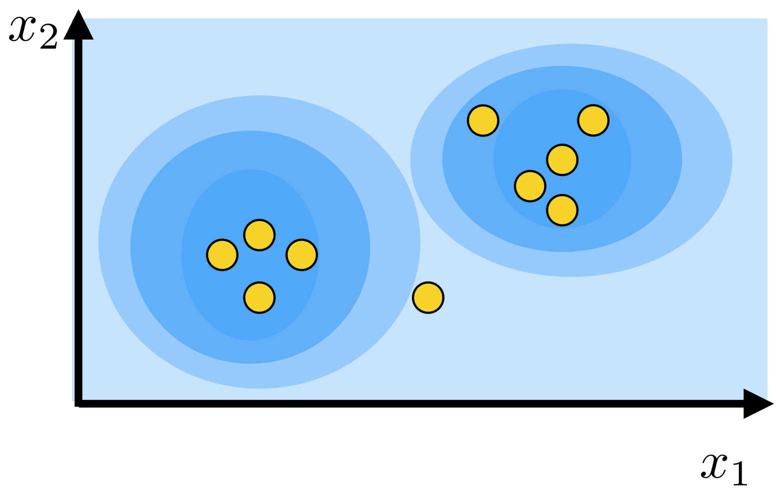
$$k(x,x') = x^T x' + \sigma_1^2 \exp\left(-\frac{||x-x'||^2}{2\ell^2}\right) + \sigma_2^2[x = x'] + \sigma_3^2$$

$$x, x' \in \mathbb{R}^d, d - \text{number of features}$$



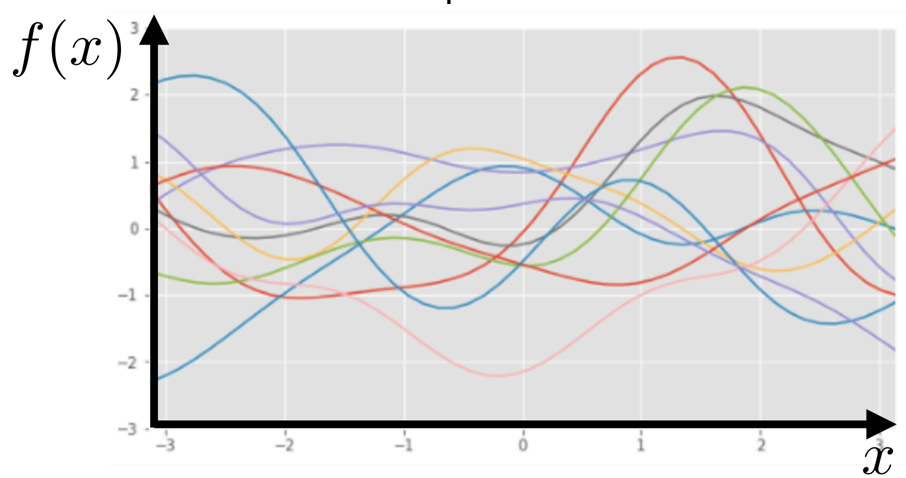
Sampling functions from a process

Multivariate distribution
— sample **points**



$$x \sim p(x)$$

Process
— sample **functions**

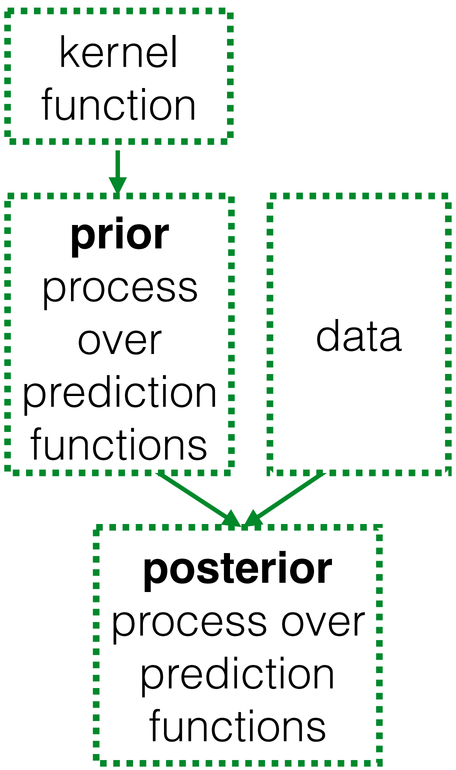


$$f(x) \sim GP(m(x), k(x, x'))$$

each line is a sample from the process

Right image from https://towardsdatascience.com/an-intuitive-guide-to-gaussian-processes-ec2f0b45c71d

Gaussian processes for regression



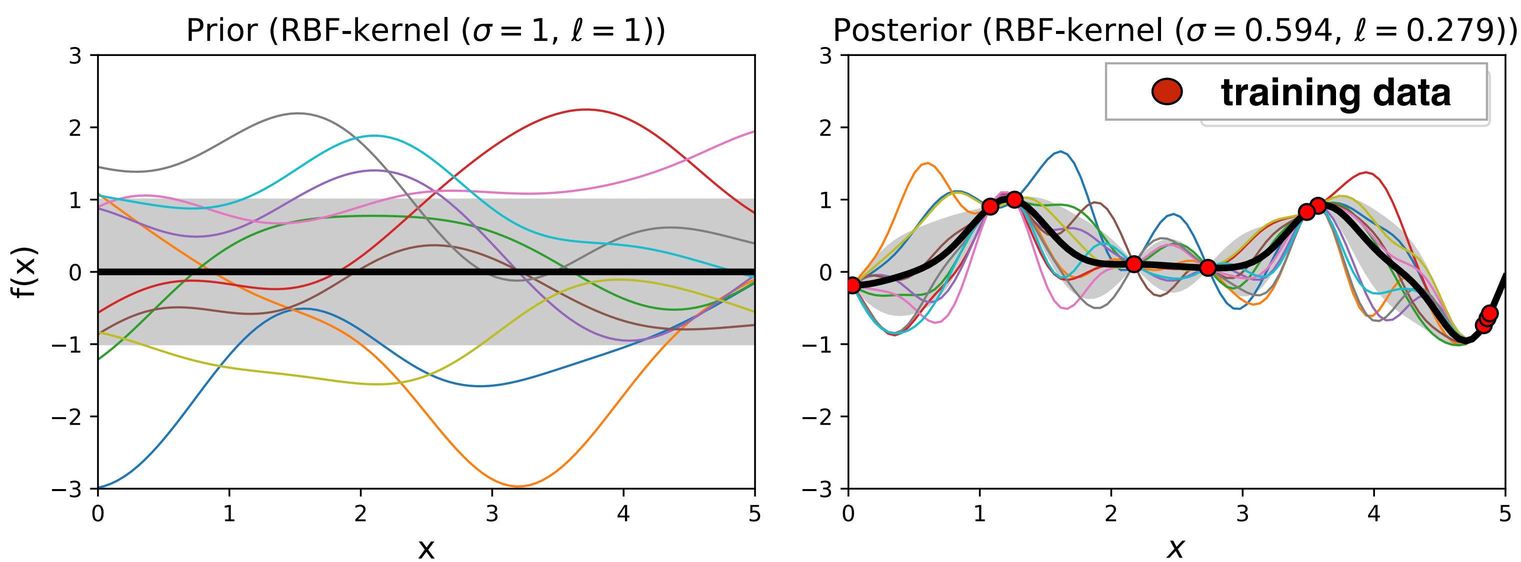
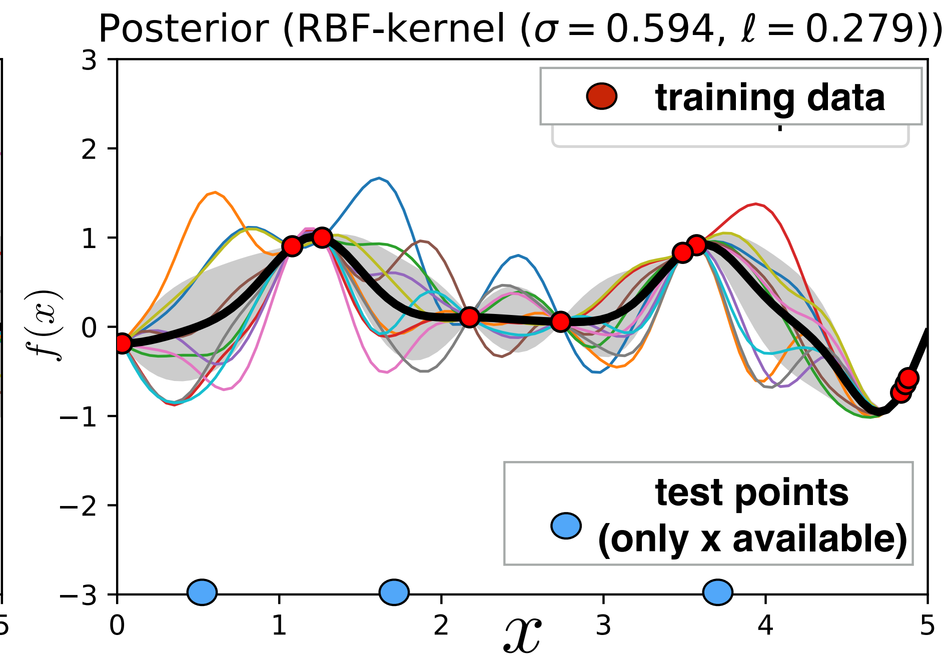


Image source: https://scikit-learn.org/stable/modules/gaussian_process.html

Gaussian processes for regression



Given: (1) training data and
(2) prior Gaussian process over
prediction functions $a(x)$

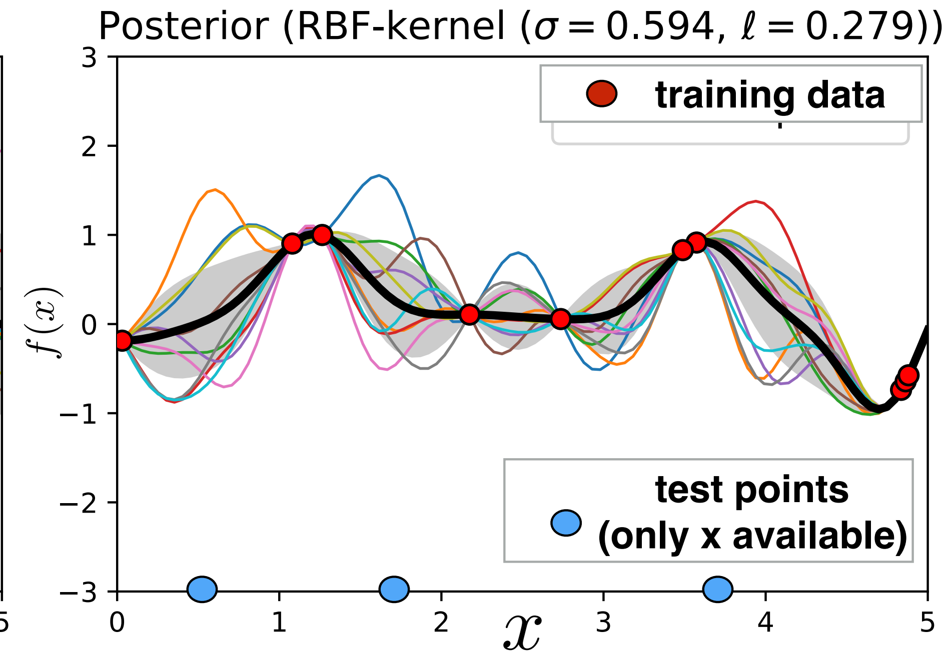
Training data :

$X^{tr} = \{x_i^{tr}\}_{i=1}^N,\ x_i^{tr} \in \mathbb{R}^d$ — input data

$Y^{tr} = \{y_i^{tr}\}_{i=1}^N,\ y_i^{tr} \in \mathbb{R}$ — targets

N – number of objects, d – number of features

Gaussian processes for regression



Training data:

$$X^{tr} = \{x_i^{tr}\}_{i=1}^{N},\ x_i^{tr} \in \mathbb{R}^d \text{ — input data}$$

$$Y^{tr} = \{y_i^{tr}\}_{i=1}^{N},\ y_i^{tr} \in \mathbb{R} \text{ — targets}$$

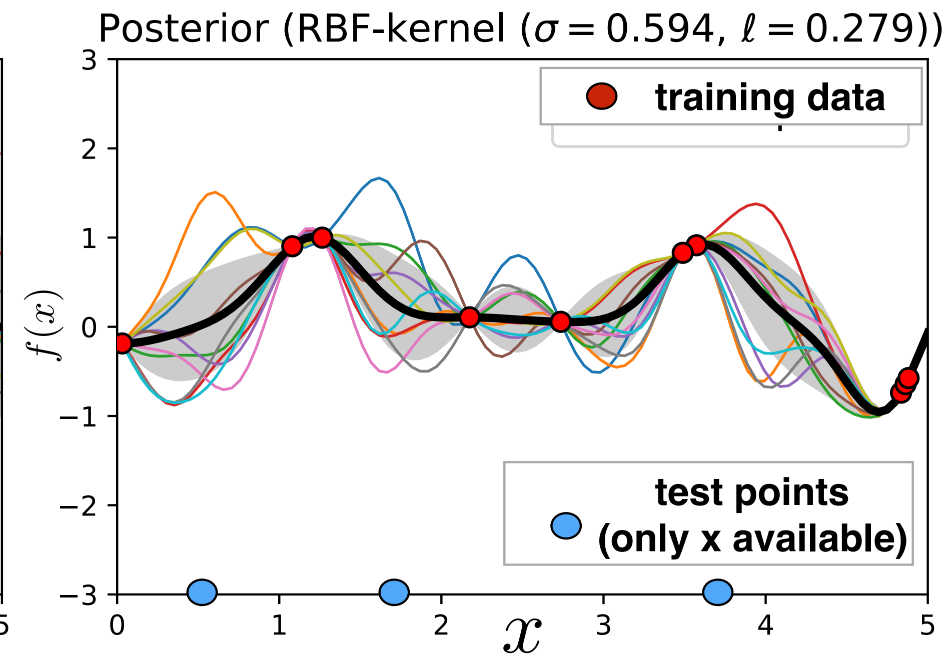
N – number of objects, d – number of features

Test points (any set of points):

$$X^{te} = \{x_i^{te}\}_{i=1}^{M},\ x_i^{te} \in \mathbb{R}^d$$

Given: (1) training data and (2) prior Gaussian process over prediction functions $a(x)$

Gaussian processes for regression



Given: (1) training data and (2) prior Gaussian process over prediction functions $a(x)$

Training data :

$$X^{tr} = \{x_i^{tr}\}_{i=1}^N, \; x_i^{tr} \in \mathbb{R}^d \text{ — input data}$$

$$Y^{tr} = \{y_i^{tr}\}_{i=1}^N, \; y_i^{tr} \in \mathbb{R} \text{ — targets}$$

N – number of objects, d – number of features

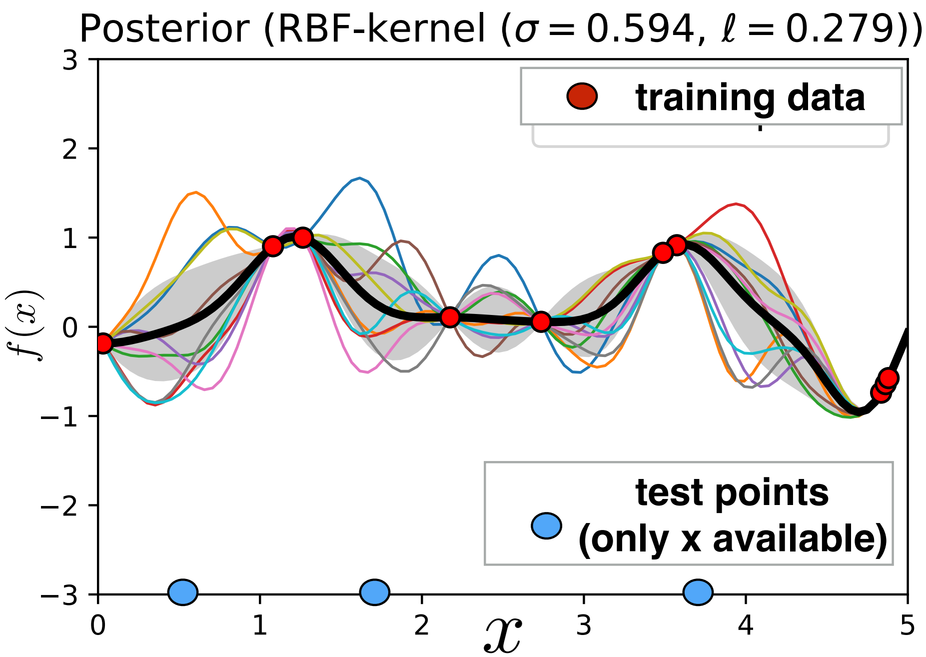
Test points (any set of points):

$$X^{te} = \{x_i^{te}\}_{i=1}^M, \; x_i^{te} \in \mathbb{R}^d$$

Find:

$$p(a(x_1^{te}), \ldots, a(x_M^{te})) - ?$$ **p( )**

Conditioning in multivariate normal distribution

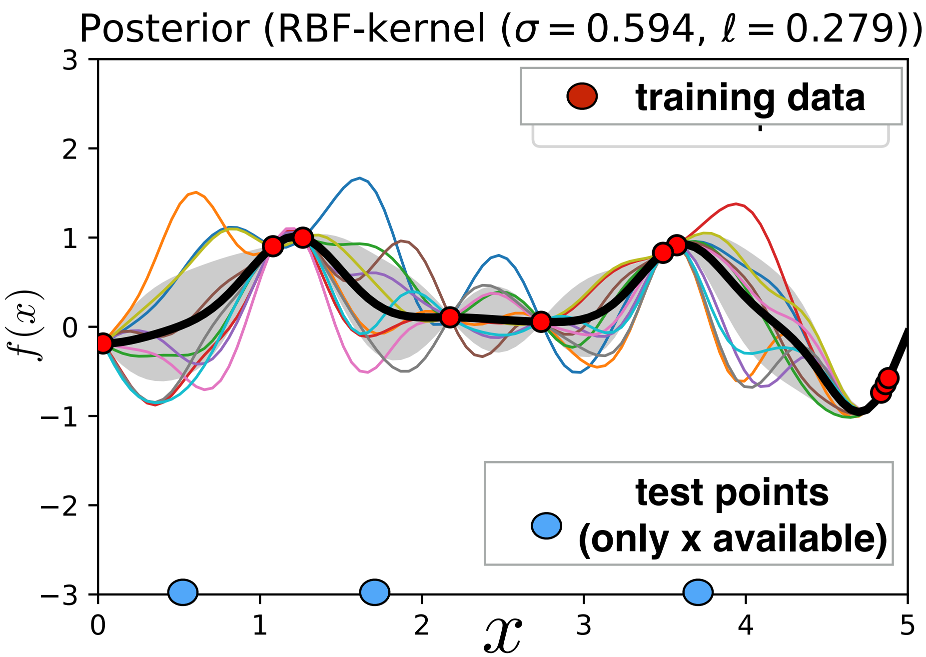


Definition of Gaussian process:
every finite set of function values
has a multivariate normal distribution

$$\forall n \quad \forall (x_1, \ldots, x_n) \quad (a(x_1), \ldots, a(x_n)) \sim \mathcal{N}(\mu, \Sigma)$$

$$\mu = \{m(x_i)\}_{i=1}^{n} \quad \Sigma = \{k(x_i, x_j)\}_{i,j=1}^{n,n}$$

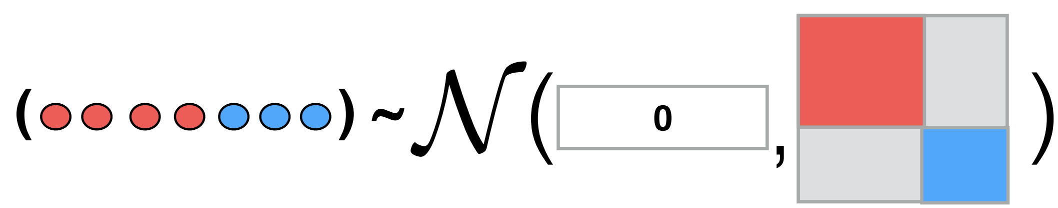
Conditioning in multivariate normal distribution



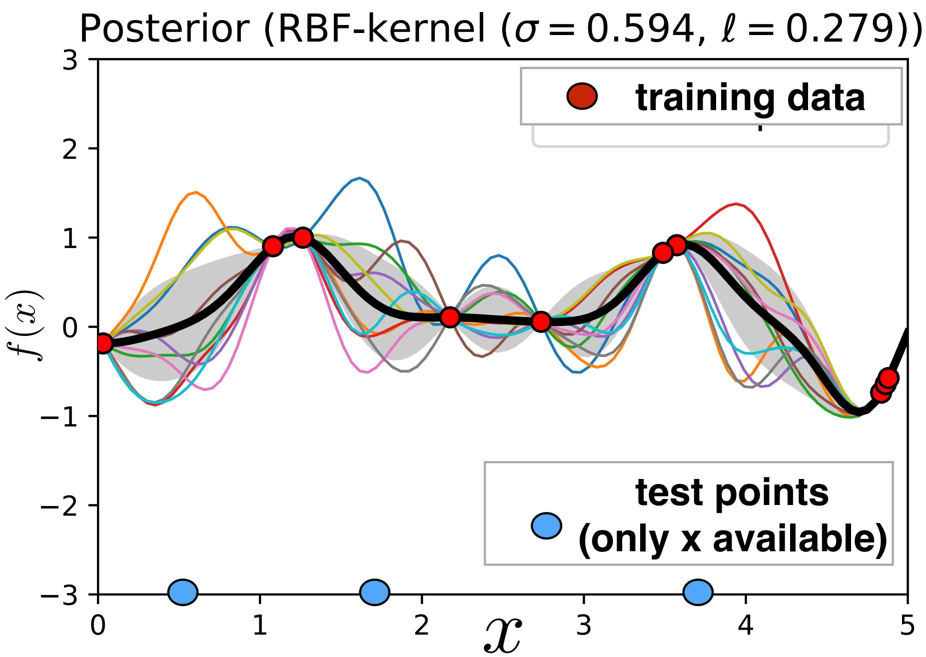
Definition of Gaussian process:
every finite set of function values
has a multivariate normal distribution

$$\forall n \quad \forall (x_1, \ldots, x_n) \quad (a(x_1), \ldots, a(x_n)) \sim \mathcal{N}(\mu, \Sigma)$$

$$\mu = \{m(x_i)\}_{i=1}^{n} \quad \Sigma = \{k(x_i, x_j)\}_{i,j=1}^{n,n}$$



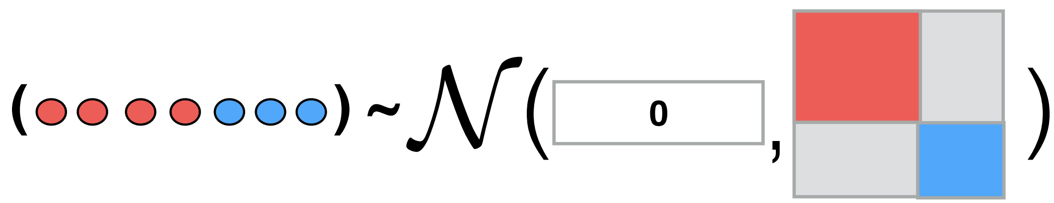
Conditioning in multivariate normal distribution



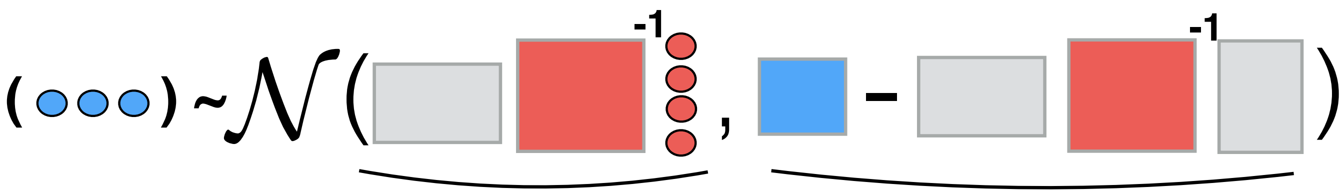
Definition of Gaussian process:
every finite set of function values
has a multivariate normal distribution

$$\forall n \quad \forall(x_1, \ldots, x_n) \quad (a(x_1), \ldots, a(x_n)) \sim \mathcal{N}(\mu, \Sigma)$$

$$\mu = \{m(x_i)\}_{i=1}^{n} \quad \Sigma = \{k(x_i, x_j)\}_{i,j=1}^{n,n}$$



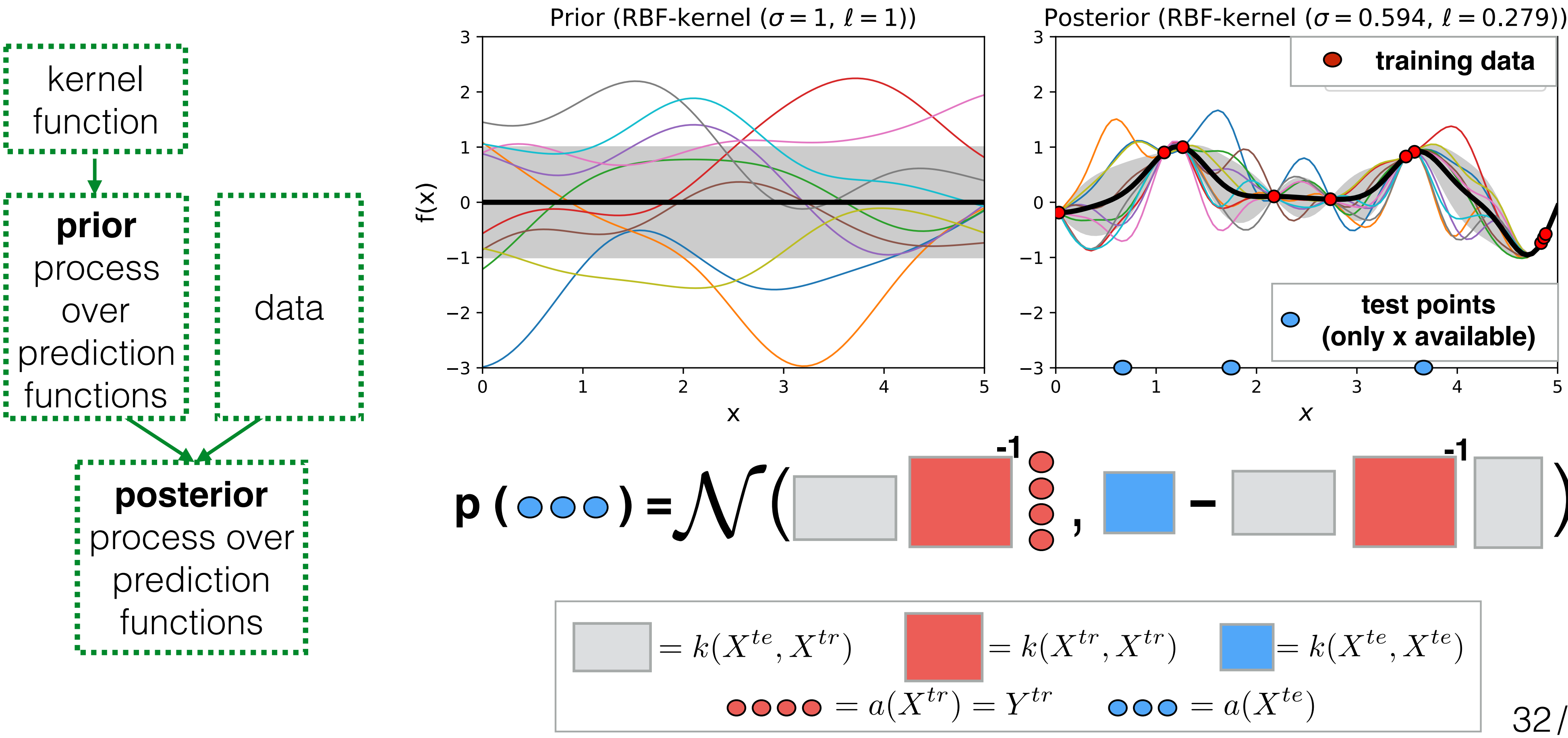
According to properties of normal distribution:



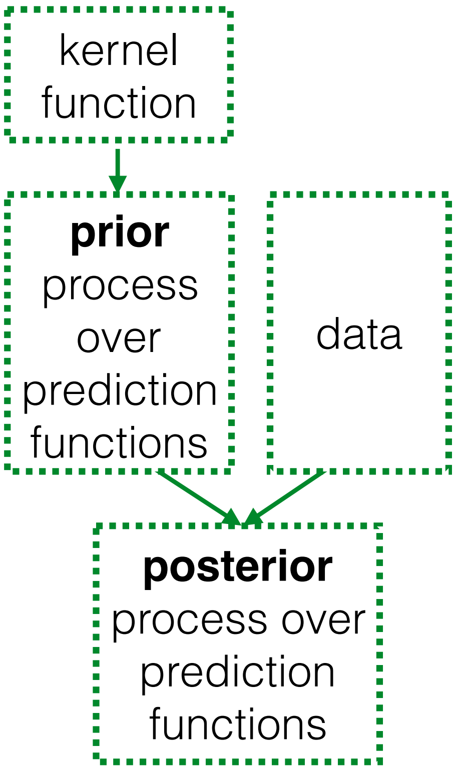
defines a new mean function

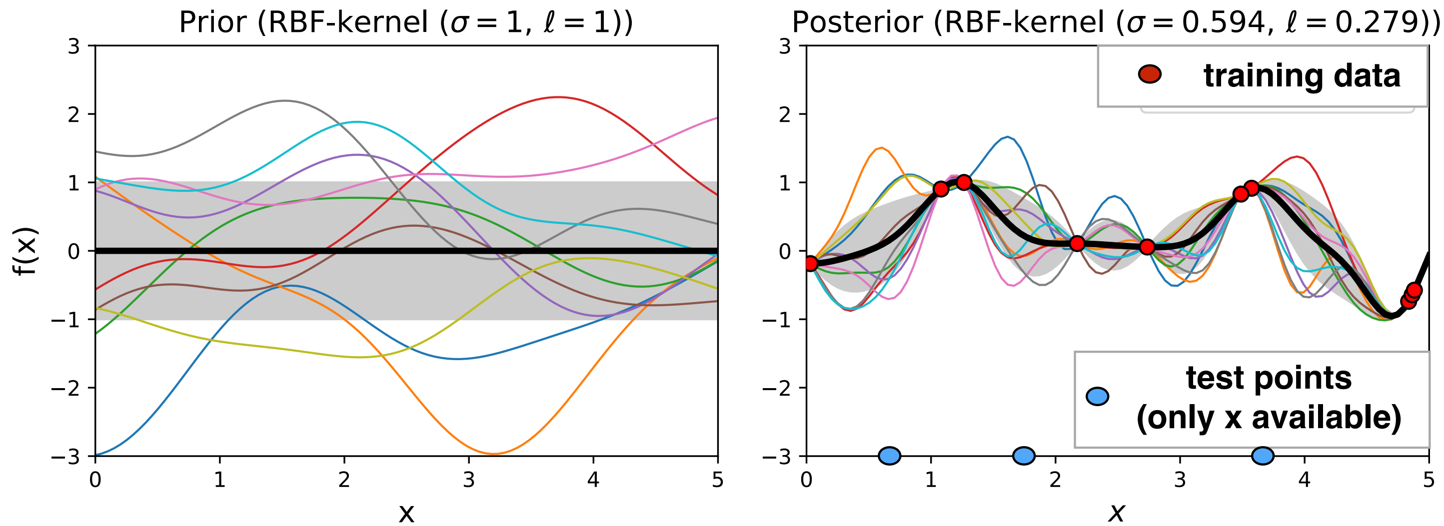
defines a new covariance function

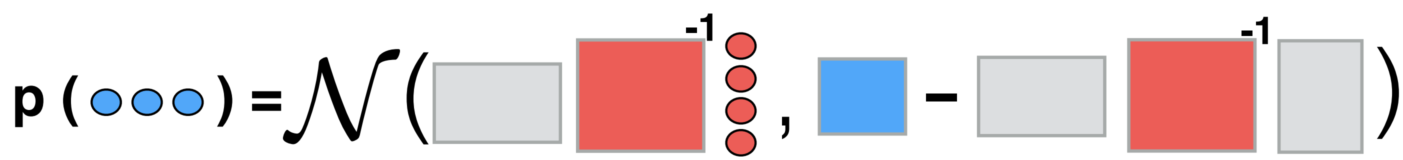
Gaussian processes for regression



Gaussian processes for regression

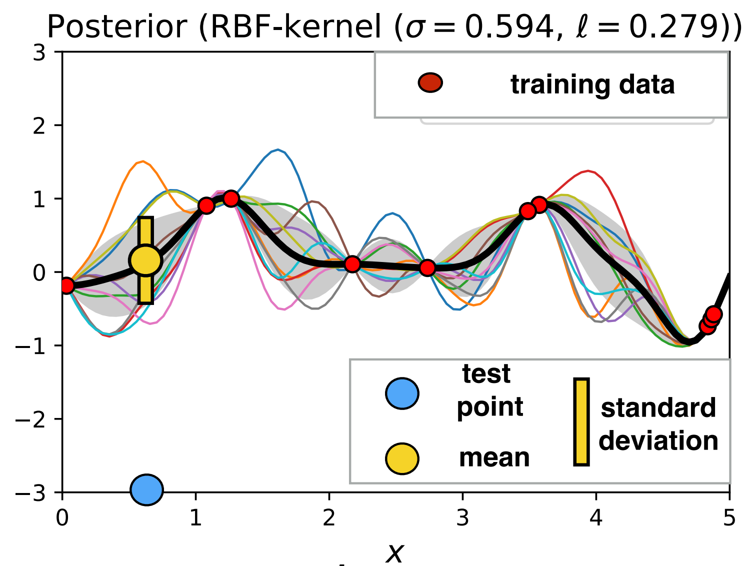




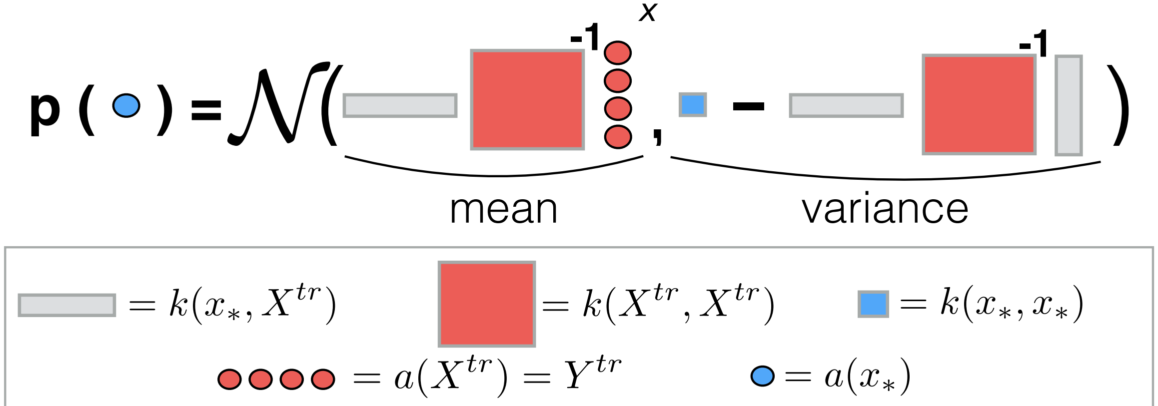


Training? Prediction?

Training and prediction in GP for regression

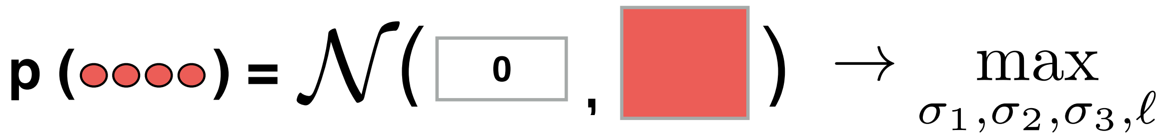


Prediction:



Training and prediction in GP for regression

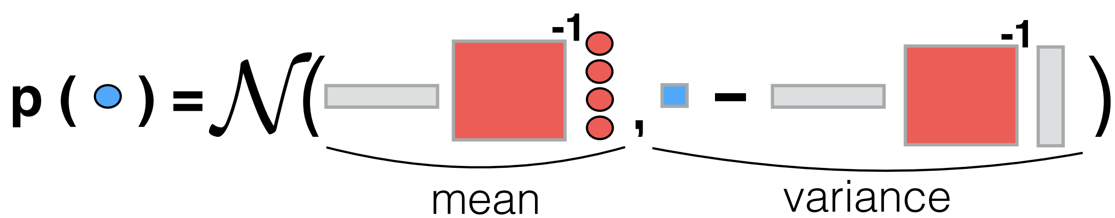
Training:

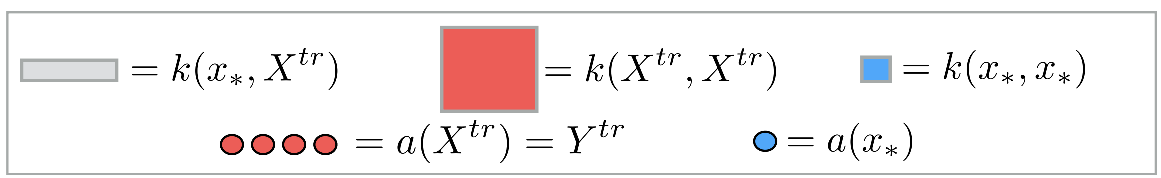


$$k(x,x') = x^T x' + \sigma_1^2 \exp\left(-\frac{||x-x'||^2}{2\ell^2}\right) + \sigma_2^2[x=x'] + \sigma_3^2$$

parameters of the kernel (covariance) function

Prediction:



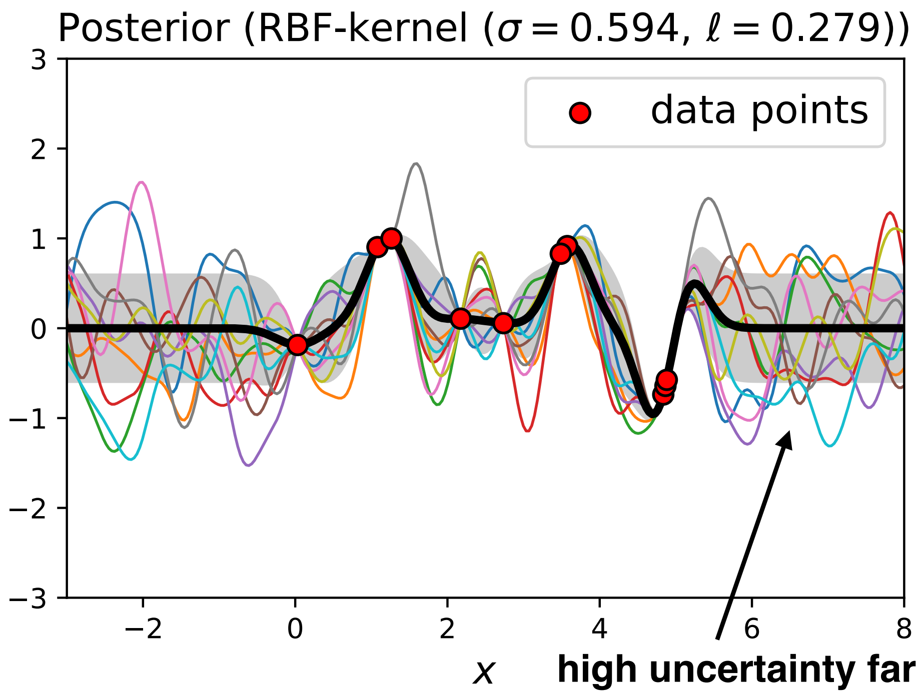


Parametric vs non-parametric models

	Parametric models:	**Non-parametric models:**
Prediction:	$a(x)$ – function of x and parameters θ	$a(x)$ – function of x and training data
Training:	finding θ based on training data	none (or tuning a small number of parameters)
Examples:	decision trees (Bayesian) linear regression	kNN Gaussian processes

Pros and cons of Gaussian processes

+ uncertainty estimation

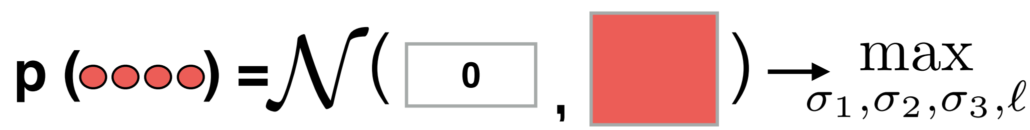


high uncertainty far from training data

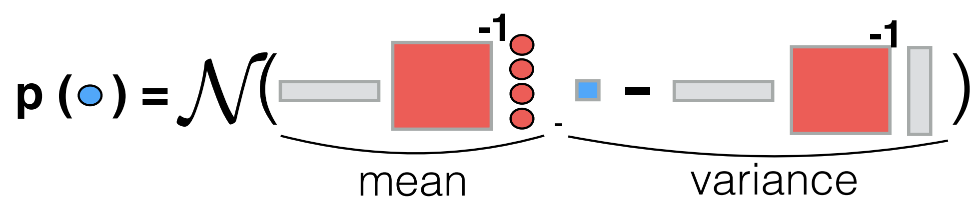
— kernel (covariance) function?

— slow computation

Training: O(N³)



Prediction: O(N) — mean, O(N²) — std



N — number of training objects

Summary

- Gaussian process is a ``distribution'' over *functions*
- Regression with Gaussian Process generalizes kNN in a probabilistic manner
- Gaussian Processes provide reliable uncertainty estimates but require careful choice of kernel function and are slow in training and testing

Questions

Exercise

- Consider we are given the following training data (1 feature):

x	y
-1.5	1
0.5	3
0.7	2.5

We use zero mean function and RBF-kernel: $k(x, x') = 0.5 \exp\left(-\frac{(x - x')^2}{2}\right)$

- What prediction will we make for a new object $x_\star = 0?$ for $x_\star = 3?$

Formulas:

$$p(a(x_*)) = \mathcal{N}(m_*, \sigma_*) \quad m_* = k_*^T K^{-1} Y, \quad \sigma_*^2 = k_{**} - k_*^T K^{-1} k_*$$

$$k_{**} = k(x_*, x_*), \quad k_* = \{k(x_i, x_*)\}_{i=1}^N, \quad K = \{k(x_i, x_j)\}_{i,j=1}^{N,N}, \quad Y = \{y_i\}_{i=1}^N$$

Exercise

x	y
-1.5	1
0.5	3
0.7	2.5

$$K = 0.5 \cdot \begin{bmatrix} 1 & \exp\left(-\frac{(-1.5-0.5)^2}{2}\right) & \exp\left(-\frac{(-1.5-0.7)^2}{2}\right) \\ \exp\left(-\frac{(-1.5-0.5)^2}{2}\right) & 1 & \exp\left(-\frac{(0.5-0.7)^2}{2}\right) \\ \exp\left(-\frac{(-1.5-0.7)^2}{2}\right) & \exp\left(-\frac{(0.5-0.7)^2}{2}\right) & 1 \end{bmatrix}$$

Formulas:

$$p(a(x_*)) = \mathcal{N}(m_*, \sigma_*) \quad m_* = k_*^T K^{-1} Y, \quad \sigma_*^2 = k_{**} - k_*^T K^{-1} k_*$$

$$k_{**} = k(x_*, x_*), \quad k_* = \{k(x_i, x_*)\}_{i=1}^N, \quad K = \{k(x_i, x_j)\}_{i,j=1}^{N,N}, \quad Y = \{y_i\}_{i=1}^N$$

Exercise

x	y
-1.5	1
0.5	3
0.7	2.5

$$K = 0.5 \cdot \begin{bmatrix} 1 & \exp\left(-\frac{(-1.5-0.5)^2}{2}\right) & \exp\left(-\frac{(-1.5-0.7)^2}{2}\right) \\ \exp\left(-\frac{(-1.5-0.5)^2}{2}\right) & 1 & \exp\left(-\frac{(0.5-0.7)^2}{2}\right) \\ \exp\left(-\frac{(-1.5-0.7)^2}{2}\right) & \exp\left(-\frac{(0.5-0.7)^2}{2}\right) & 1 \end{bmatrix}$$

For $x_* = 0:$

$$k_* = 0.5 \cdot \begin{bmatrix} \exp\left(-\frac{(0+1.5)^2}{2}\right) & \exp\left(-\frac{(0-0.5)^2}{2}\right) & \exp\left(-\frac{(0-0.7)^2}{2}\right) \end{bmatrix}$$

$$k_{**} = \begin{bmatrix} 0.5 \end{bmatrix}$$

Formulas:

$$p(a(x_*)) = \mathcal{N}(m_*, \sigma_*) \quad m_* = k_*^T K^{-1} Y, \quad \sigma_*^2 = k_{**} - k_*^T K^{-1} k_*$$

$$k_{**} = k(x_*, x_*), \quad k_* = \{k(x_i, x_*)\}_{i=1}^N, \quad K = \{k(x_i, x_j)\}_{i,j=1}^{N,N}, \quad Y = \{y_i\}_{i=1}^N$$

Exercise

x	y
-1.5	1
0.5	3
0.7	2.5

$$K = 0.5 \cdot \begin{bmatrix} 1 & \exp\left(-\frac{(-1.5-0.5)^2}{2}\right) & \exp\left(-\frac{(-1.5-0.7)^2}{2}\right) \\ \exp\left(-\frac{(-1.5-0.5)^2}{2}\right) & 1 & \exp\left(-\frac{(0.5-0.7)^2}{2}\right) \\ \exp\left(-\frac{(-1.5-0.7)^2}{2}\right) & \exp\left(-\frac{(0.5-0.7)^2}{2}\right) & 1 \end{bmatrix}$$

For $x_* = 0:$ $\qquad k_* = 0.5 \cdot \begin{bmatrix} \exp\left(-\frac{(0+1.5)^2}{2}\right) & \exp\left(-\frac{(0-0.5)^2}{2}\right) & \exp\left(-\frac{(0-0.7)^2}{2}\right) \end{bmatrix} \qquad k_{**} = \begin{bmatrix} 0.5 \end{bmatrix}$

$$\mu_* = \begin{bmatrix} 0.162 & 0.441 & 0.391 \end{bmatrix} \begin{bmatrix} 0.5 & 0.067 & 0.044 \\ 0.067 & 0.5 & 0.490 \\ 0.044 & 0.490 & 0.5 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 3 \\ 2.5 \end{bmatrix} \qquad \sigma_*^2 = 0.5 - \begin{bmatrix} 0.162 & 0.441 & 0.391 \end{bmatrix} \begin{bmatrix} 0.5 & 0.067 & 0.044 \\ 0.067 & 0.5 & 0.490 \\ 0.044 & 0.490 & 0.5 \end{bmatrix}^{-1} \begin{bmatrix} 0.162 \\ 0.441 \\ 0.391 \end{bmatrix}$$

Formulas:

$$p(a(x_*)) = \mathcal{N}(m_*, \sigma_*) \quad m_* = k_*^T K^{-1} Y, \quad \sigma_*^2 = k_{**} - k_*^T K^{-1} k_*$$

$$k_{**} = k(x_*, x_*), \quad k_* = \{k(x_i, x_*)\}_{i=1}^N, \quad K = \{k(x_i, x_j)\}_{i,j=1}^{N,N}, \quad Y = \{y_i\}_{i=1}^N$$