Meeting 1

📅Date: 02-01-2023
🕐Time: 13:13

Bipin Koirala

Table of Contents

1. Conditional Distribution of Multivariate Gaussian
2. Gaussian Process
    1. Gaussian Process Regression

Conditional Distribution of Multivariate Gaussian

Theorem

Let, $x \in \mathbb{R}^n$ and x_1, x_2 are subset of x s.t. $x_1 \in \mathbb{R}^{n_1}$ and $x \in \mathbb{R}^{n_2}$ with $n = n_1 + n_2$.

If $x \sim \mathcal{N}(\mu, \Sigma)$, then $x_1|x_2 \sim \mathcal{N}(\mu_{1|2}, \Sigma_{1|2})$.

with $\mu_{1|2} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2)$ and $\Sigma_{1|2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$

Here, without any loss of generality $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \right)$

Proof:
By construction; x_1 and x_2 are jointly Gaussian. Furthermore, Gaussian distributions are closed under marginalization and conditioning i.e.

$$\begin{aligned} x_1 &\sim \mathcal{N}(\mu_1, \Sigma_{11}) \\ x_2 &\sim \mathcal{N}(\mu_2, \Sigma_{22}) \end{aligned}$$

We have, $\mathbb{P}(x_1|x_2) = \frac{\mathbb{P}(x_1,x_2)}{\mathbb{P}(x_2)} = \frac{\mathcal{N}(x;\mu,\Sigma)}{\mathcal{N}(x_2;\mu_2,\Sigma_{22})}$

Note

PDF of Multivariate Normal Distribution:

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

Now,

$$\mathbb{P}(x_1|x_2) = \frac{1}{\sqrt{(2\pi)^{n-n_2}}} \sqrt{\frac{|\Sigma_{22}|}{|\Sigma|}} \exp\left[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu) + \frac{1}{2}(x_2-\mu_2)^T\Sigma_{22}^{-1}(x_2-\mu_2) \right] \tag{1}$$

Let; $\Sigma^{-1} = \begin{bmatrix} \Sigma^{11} & \Sigma^{12} \\ \Sigma^{21} & \Sigma^{22} \end{bmatrix}$ and since Σ^{-1} is symmetric matrix we have $(\Sigma^{21})^T = \Sigma^{12}$; the argument of exponential part in (1) becomes;

$$\begin{aligned} &= -\frac{1}{2}\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \right)^T \begin{bmatrix} \Sigma^{11} & \Sigma^{12} \\ \Sigma^{21} & \Sigma^{22} \end{bmatrix} \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \right) + \frac{1}{2}(x_2-\mu_2)^T\Sigma_{22}^{-1}(x_2-\mu_2) \\ &= -\frac{1}{2}[(x_1-\mu_1)^T \quad (x_2-\mu_2)^T] \begin{bmatrix} \Sigma^{11} & \Sigma^{12} \\ \Sigma^{21} & \Sigma^{22} \end{bmatrix} \begin{bmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{bmatrix} + \frac{1}{2}(x_2-\mu_2)^T\Sigma_{22}^{-1}(x_2-\mu_2) \\ &= -\frac{1}{2}\Big((x_1-\mu_1)^T\Sigma^{11}(x_1-\mu_1) + 2(x_1-\mu_1)^T\Sigma^{12}(x_2-\mu_2) + (x_2-\mu_2)^T\Sigma^{22}(x_2-\mu_2) \Big) \dots \\ &\dots + \frac{1}{2}(x_2-\mu_2)^T\Sigma_{22}^{-1}(x_2-\mu_2) \end{aligned}$$

Note

Inverse of a Block Matrix:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} (A-BD^{-1}C)^{-1} & -(A-BD^{-1}C)^{-1}BD^{-1} \\ -D^{-1}C(A-BD^{-1}C)^{-1} & D^{-1}+D^{-1}C(A-BD^{-1}C)^{-1}BD^{-1} \end{bmatrix}$$

From the above note and we can get the corresponding expressions for each entries of Σ^{-1}. Plugging these expressions back to (1) yields the following:

$$\begin{aligned} \mathbb{P}(x_1|x_2) = \frac{1}{\sqrt{(2\pi)^{n_1}}}\sqrt{\frac{|\Sigma_{22}|}{|\Sigma|}} \exp\Bigg[-\frac{1}{2}\Big(& (x_1-\mu_1)^T(\Sigma_{11}-\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})^{-1}(x_1-\mu_1) \\ & -2(x_1-\mu_1)^T(\Sigma_{11}-\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})^{-1}\Sigma_{12}\Sigma_{22}^{-1}(x_2-\mu_2) \\ & +(x_2-\mu_2)^T[\Sigma_{22}^{-1}+\Sigma_{22}^{-1}\Sigma_{21}(\Sigma_{11}-\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})^{-1}](x_2-\mu_2)\Big) \\ & +\frac{1}{2}(x_2-\mu_2)^T\Sigma_{22}^{-1}(x_2-\mu_2)\Bigg] \end{aligned} \tag{2}$$

Note

Determinant of a Block Matrix:

$\begin{vmatrix} A & B \\ C & D \end{vmatrix} = |D| \cdot |A-BD^{-1}C|$

Hence;

$$|\Sigma| = |\Sigma_{22}| \cdot |\Sigma_{11}-\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}| \tag{3}$$

Upon re-arranging the terms from (2) and using the fact (3), we get:

$$\begin{aligned} \mathbb{P}(x_1|x_2) &= \frac{1}{\sqrt{(2\pi)^{n_1}}}|\Sigma_{11}-\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}|^{-1/2} \exp\left\{ -\frac{1}{2}\Big[x_1-\big(\mu_1+\Sigma_{12}\Sigma_{22}^{-1}(x_2-\mu_2)\big)\Big]^T(\Sigma_{11}-\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})^{-1}\Big[x_1-\big(\mu_1+\Sigma_{12}\Sigma_{22}^{-1}(x_2-\mu_2)\big)\Big]\right\} \\ &= \frac{1}{\sqrt{(2\pi)^{n_1}}}|\Sigma_{1|2}^{-1/2}| \exp\left\{-\frac{1}{2}(x_1-\mu_{1|2})^T\Sigma_{1|2}^{-1}(x_1-\mu_{1|2})\right\} \end{aligned}$$

$$\therefore x_1|x_2 \sim \mathcal{N}(\mu_{1|2}, \Sigma_{1|2})$$

Corollary:

$\mathbb{P}(x_2|x_1) = \frac{1}{\sqrt{(2\pi)^{n_2}}}|\Sigma_{2|1}^{-1/2}| \exp\left\{-\frac{1}{2}(x_2-\mu_{2|1})^T\Sigma_{2|1}^{-1}(x_2-\mu_{2|1})\right\}$

Gaussian Process

A Gaussian Process, GP in short, is a (potentially infinite) collection of random variables (RVs) such that the joint distribution of every finite subset of RVs is a Multivariate Gaussian.

$$f \sim GP(\mu, k)$$

where $\mu(x)$ and $\kappa(x, x')$ are the mean and covariance of f respectively.

To model the predictive distribution, we use a GP prior: $\mathbb{P}(f|x) \sim \mathcal{N}(\mu, \Sigma)$ and condition it on the training data $\mathcal{D}$ to model the joint distribution $f(X)$ and it prediction at test data $f(X')$.

Gaussian Process Regression

Without any loss of generality and before observing the training labels, we assume that the labels are drawn from the zero-mean prior Gaussian distribution i.e.

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} \sim \mathcal{N}(0, \Sigma)$$

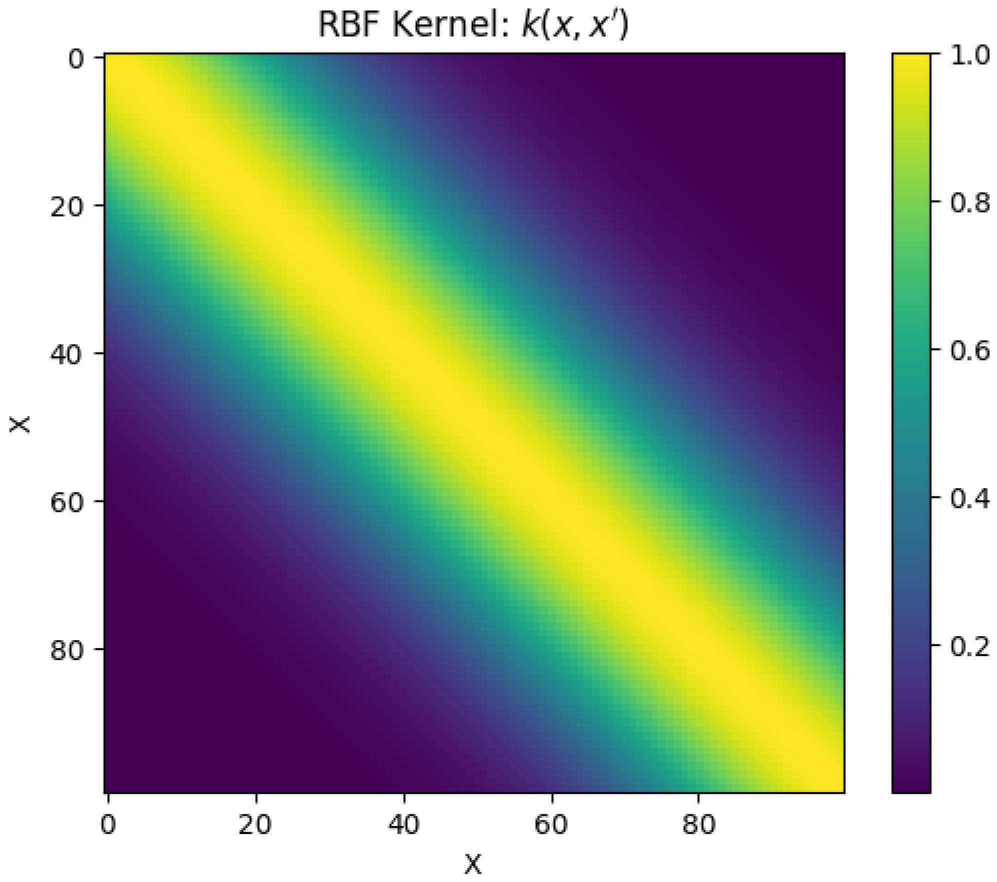
Let $y_2, y_3, \ldots, y_t$ be training points and $y_{t+1}, y_{t+2}, \ldots, y_n$ be test points. Then the covariance matrix Σ is a block matrix as shown below.

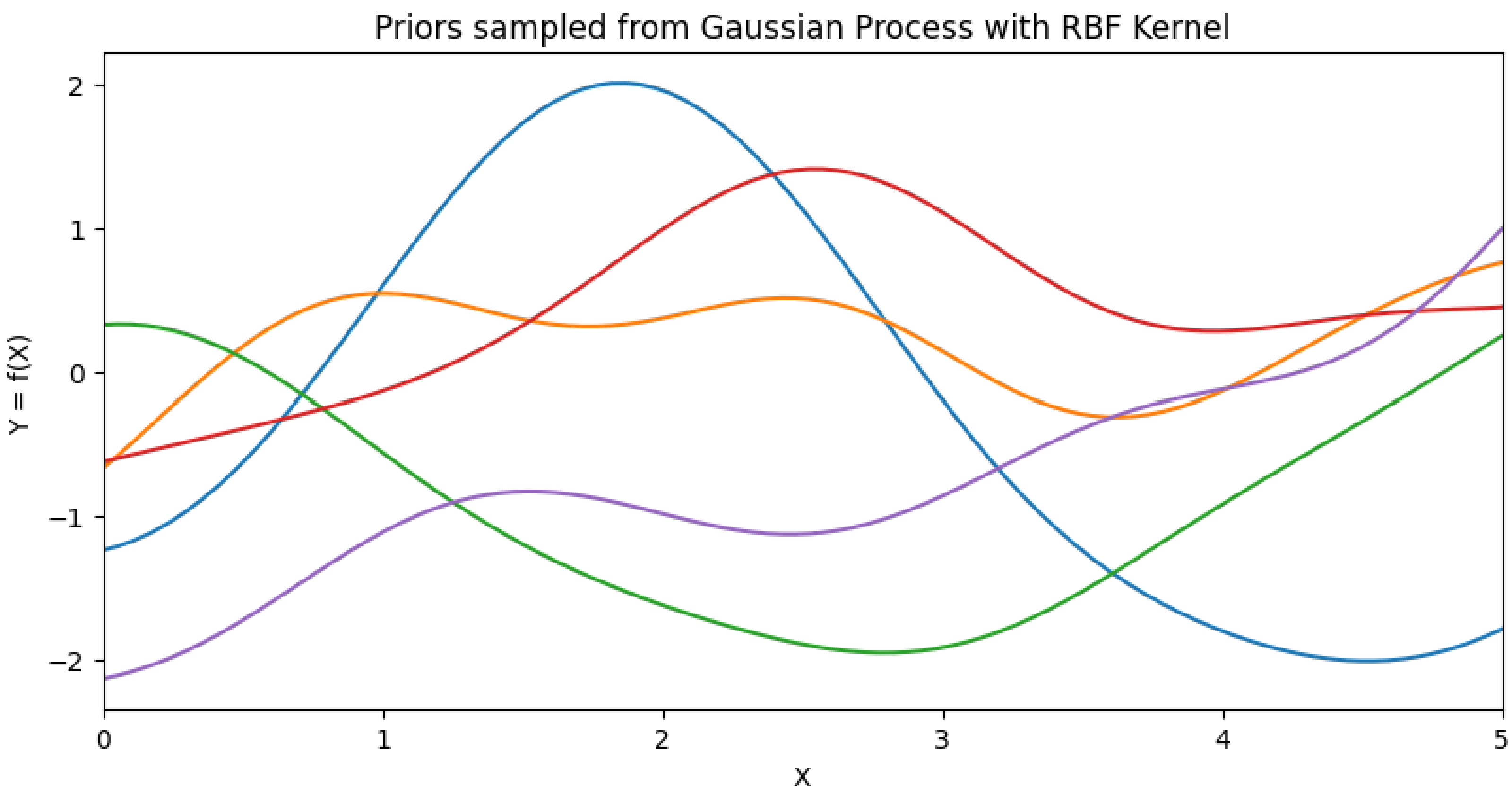
$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$$

where $\Sigma_{11} = \mathcal{K}(x_1, x_1)$ and so on. Also, x_1, x_2 are train points and test points respectively.
Most commonly used kernel is $Radial\ Basis\ Function$ (RBF):

$$k(x,x') = \sigma^2 \, e^{\left(-\frac{||x-x'||^2}{2\,l^2}\right)}$$

```
import numpy as np
import matplotlib.pyplot as plt
import scipy

def kernel(x, xp):
        '''k(x,x') = sigma^2 exp(-0.5*length^2*|x-x'|^2)'''
        σ = 1
        length = 1
        sq_norm = scipy.spatial.distance.cdist(x, xp, 'sqeuclidean')
        return σ**2 * np.exp(-0.5*sq_norm*length**2)

# Sample from Gaussian Process Distribution
pts = 100 # number of points in each function
n = 5 # number of functions to sample


# Independent Variable Samples
X = np.linspace(0,5, pts)
X = X.reshape(-1,1)
Σ = kernel(X,X)
fx = np.random.multivariate_normal(mean = np.zeros(pts), cov = Σ, size = n)



plt.title('RBF Kernel: $k(x,x\')$')
plt.imshow(Σ, cmap = 'viridis')
plt.colorbar()
plt.xlabel('X')
plt.ylabel('X')
plt.show()


plt.figure(figsize=(8,4))
for i in range(n):
        plt.plot(X, fx[i])

plt.tight_layout()
plt.xlim(0,5)
plt.xlabel('X')
plt.ylabel('Y = f(X)')
plt.title('Priors sampled from Gaussian Process with RBF Kernel')
plt.show()
```





Now, posterior is obtained using the formula:

$$\mathbb{P}(y_2|y_1, X1, X2) = \mathcal{N}(\mu_{2|1}, \Sigma_{2|1})$$

where; $\mu_{2|1} = \mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(y_1 - 0)$ and $\Sigma_{2|1} = \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}$
And,

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} f(x_1) \\ f(x_2) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \right)$$

Furthermore, if we have a noisy observation data X_1 it can be approximately modeled by taking $\Sigma_{11} = k(x_1, x_1) + \sigma_\epsilon^2 I$

```
def posterior(X1, y1, X2, kernel, noise = None):
'''
Compute posterior mean and covariance i.e. mu_(2|1) and cov_(2|1)
y1 = f(x1)
'''

        Σ11 = kernel(X1, X1)
        if noise is not None:
                err = (noise**2) * np.eye(Σ11.shape[0])
                Σ11 += err

        Σ22 = kernel(X2, X2)
        Σ12 = kernel(X1, X2)

        sol = scipy.linalg.solve(Σ11, Σ12, assume_a = 'pos').T
        #μ1 = np.mean(X1)
        μ1 = 0 # assume prior mean is 0
        μ2 = np.mean(X2)
        μ = μ2 + sol @ (y1 - μ1)
        Σ = Σ22 - (sol @ Σ12)

        return μ, Σ
```

```
# Define the true function
f_sin = lambda x: (np.sin(x)).flatten()
n1 = 10 # number of points to condition on (training points)
n2 = 70 # number of points in posterior (test points)
ny = 5 # number of functions that will be sampled from posterior

# Sample observations
X1 = np.random.uniform(-4, 4, size = (n1, 1))
y1 = f_sin(X1)

# Predict points at uniform spacing to capture funciton
X2 = np.linspace(-6, 6, n2).reshape(-1,1)

# Compute posterior mean and covariance
μ2, Σ2 = posterior(X1, y1, X2, kernel = kernel, noise = 0.2)

# Compute standard deviation at test points to be plotted
σ2 = np.sqrt(np.diag(Σ2))
```

```
# Draw some samples from the posterior
y2 = np.random.multivariate_normal(mean = μ2, cov = Σ2, size = ny)

plt.figure(figsize=(10,5))
plt.plot(X2, f_sin(X2), 'b--',label = '$sin(x)$')
plt.scatter(X1, y1, color = 'red',label = '($x_1, y_1$)')
plt.plot(X2, μ2, color = 'red', label = '$\mu_{2|1}$')
plt.fill_between(X2.flatten(), μ2 - σ2, μ2 + σ2, color = 'blue', alpha = 0.1, label = '$\pm \sigma$')
plt.plot()
plt.legend()
plt.xlim(-6,6)
plt.title('Posterior Distribution')
plt.grid()
plt.show()

plt.figure(figsize=(10,5))
plt.title('Sampling from Posterior $\mathbb{P}(x_2|x_1)$')
plt.plot(X2, y2.T)
plt.xlim(-6,6)
plt.grid()
plt.show()
```