Gaussian Processes

Nipun Batra

December 23, 2023

IIT Gandhinagar

Gaussian Distribution

1D Gaussian Scatter Plot

1D Gaussian Histogram

Varying 1D Gaussian Variance

Bi-variate Gaussian

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} a & \rho \\ \rho & b \end{pmatrix} \right)$$

ρ denotes the correlation between X_1 and X_2.

b denotes the variance of X_1.

a denotes the variance of X_2.

Sampling Bi-variate Gaussian - 1

Here the covariance between the two random variables is positive.

Sampling Bi-variate Gaussian - 2

Here the covariance between the two random variables is negative.

Sampling Bi-variate Gaussian - 3

The two random variables are uncorrelated.

Surface Plots Bi-variate Gaussian - 1

Surface Plots Sampling Bi-variate Gaussian - 2

Visualizing samples from 2D Gaussian

Cov = 0.1 — Random State - 1

One can notice, increasing the ρ (or the covariance) between X_1 and X_2 results in the realizations of X_1 and X_2 to increasingly move together.

iter1¡ 6

Cov = 0.7 — Random State - 2

One can notice, increasing the ρ (or the covariance) between X_1 and X_2 results in the realizations of X_1 and X_2 to increasingly move together.

Conditional Bi-variate Distribution

Conditional Bi-variate Distribution

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \right)$$

The conditional expectation of X_2 given X_1 is:
$\mathrm{E}(X_2 \mid X_1 = x_1) = \rho x_1$

and the conditional variance is: $\mathrm{var}(X_2 \mid X_1 = x_1) = 1 - \rho^2$

iter1¡ 6

Conditional bi-variate — Cov = 0.1 — Random State - 2

Notice that upon fixing the first random variable, the variance of the second random variable X_2 is a function of the covariance (ρ) between the two random variables.

iter1¡ 6

Conditional bi-variate — Cov = 0.7 — Random State - 2

Notice that upon fixing the first random variable, the variance of the second random variable X_2 is a function of the covariance (ρ) between the two random variables.

5D Multivariate

iter1¡ 6

Multivariate Gaussian Sample — Random State - 2

From the visualisation above we can see that:

- Since X_1 and X_2 are highly correlated, they move up and down together
- but, X_1 and X_5 have low correlation, thus, they can seem to wiggle almost independently of each other.

Conditional Multivariate Distribution

Conditional Multivariate Distribution Definition[1]

If N-dimensional $\mathbf{x}$ is partitioned as follows

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_A \\ \mathbf{x}_B \end{bmatrix} \text{ with sizes } \begin{bmatrix} q \times 1 \\ (N-q) \times 1 \end{bmatrix}$$

and accordingly μ and $\mathbf{\Sigma}$ are partitioned as follows

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_A \\ \boldsymbol{\mu}_B \end{bmatrix} \text{ with sizes } \begin{bmatrix} q \times 1 \\ (N-q) \times 1 \end{bmatrix}$$

$$\mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma}_{AA} & \mathbf{\Sigma}_{AB} \\ \mathbf{\Sigma}_{BA} & \mathbf{\Sigma}_{BB} \end{bmatrix} \text{ with sizes } \begin{bmatrix} q \times q & q \times (N-q) \\ (N-q) \times q & (N-q) \times (N-q) \end{bmatrix}$$

[1] https://en.wikipedia.org/wiki/Multivariate_normal_distribution#Conditional_distributions

then the distribution of $\mathbf{x}_A$ conditional on $\mathbf{x}_B = \mathbf{b}$ is multivariate normal $(\mathbf{x}_A|\mathbf{x}_B = b) \sim \mathcal{N}(\bar{\mu}, \bar{\mathbf{\Sigma}})$

$$\bar{\boldsymbol{\mu}} = \boldsymbol{\mu}_A + \mathbf{\Sigma}_{AB}\mathbf{\Sigma}_{BB}^{-1}\left(\mathbf{B} - \boldsymbol{\mu}_B\right)$$

and covariance matrix

$$\overline{\mathbf{\Sigma}} = \mathbf{\Sigma}_{AA} - \mathbf{\Sigma}_{AB}\mathbf{\Sigma}_{BB}^{-1}\mathbf{\Sigma}_{BA}.$$

iter1¡ 6

Conditional Multivariate Distribution — Random State - 2

From the visualisation above we can see that:

- Since the covariance between X_4 and X_5 is high, X_4 moves such that it's value is similar to the fixed value of X_5.
- Since covariance between X_1 and X_5 is low, thus the realizations of X_1 are uncorrelated to the fixed value of X_5.

20 Dimensional Multivariate

The above heatmap shows that there is high covariance value between nearby points, but close to zero or zero covariance otherwise.

iter1¡ 6

Multivariate (20D) Distribution Samples — Random State - 2

From the animation above, we can see different family of functions of mean zero across 20 points. Notice that nearby points are more correlated (making the curve smooth) than the points farther away.

Learning from Data

Adding new Data Points

Now we want to update the model with new data. We find the functional value at the points X_1, X_2, X_6, X_{11}.

We will be using the equations at the start of section Conditional multivariate distribution, to update or Guassian Process, in the wake of new data points.

Updated Covariance Matrix

Notice that the variance of the points near the newly added data points seem to have reduced.

iter1¡ 6

Conditional Multivariate (20D) Distribution Samples — Random State - 2

From the animation above, we can see points near the added data points (red) seem to have a much lower variance compared to points far off.

Multivariate (20D) Posterior

We can easily see the reduced variance in this plot.

Kernels!

Defining Squared Exponential Kernel

We will now dive a bit into kernels. These are functions that are used to generate the covariance matrix.

Below we have defined, what is known as the Squared Exponential Kernel[2].
We have 2 parameters the define this kernel.

- σ is the scale of variance.
- l is the influence of the point to the neighbouring points

$$k(x_1, x_2) = \sigma^2 \exp\left(-\frac{(x_i - x_j)^2}{2l^2}\right)$$

[2] http://evelinag.com/Ariadne/covarianceFunctions.html

Varying l with $\sigma = 1$

As it can be seen with the plots above, using a smaller l means the function is much more smoother. Using a larger l, as in the case when $l = 1$, we see the covariance between two points that are far off, falls to zero.

Varying σ with $l = 0.1$

As it can be seen with the plots above, a small s keeps the variance and covariance values small. Whereas, a bigger s leads to higher values of variance and covariance. One thing to notice is that we are talking about covariance, not correlation, s doesn't affect the correlation between random variables.

Varying kernel parameters on 20D Guassian with conditioning

The big dark circles in the above plot denote the fixed points on which the GP is conditioned. Furthermore, notice the translucent areas, denoting the variance and the smoothness of the function denoting the correlation between random variables.

We have used the squared exponential kernel introduced earlier in this example.

Varying kernel parameters on 20D Guassian with conditioning

We can notice the increase in the variance of each of the random variables by increasing the value of s parameter of the kernel.

Varying kernel parameters on 20D Guassian with conditioning

Increasing the value of l reduces the correlation between nearby points. We, therefore, see that the resulting curve is rougher.

Varying kernel parameters on 20D Guassian with conditioning

Keeping a huge value of l results in the conditioned random variables to be highly uncorrelated with the fixed points, as can be seen above. Further, the conditioned points just move around the mean, which is set to zero in this example.

Formalization of Gaussian Processes

Gaussian Processes

A Gaussian process is fully specified by a mean function $m(\mathbf{x})$ and covariance function $K(\mathbf{x}, \mathbf{x}')$

$$f(\mathbf{x}) \sim GP(m(\mathbf{x}), K(\mathbf{x}, \mathbf{x}'))$$

where $\mathbf{x}$ is a vector and f is a real-valued function, i.e. $f : \mathbb{R}^n \to \mathbb{R}$.

Noiseless GPs

We will first consider the case of noiseless GPs. This case assumes that for a particular $\mathbf{x}$, we would always receive a single functional value $f(\mathbf{x})$, i.e., there is no inherent noise in our function.

Noiseless GPs

Given train data:

$$D = (\mathbf{x_i}, y_i), i = 1 : N$$

Noiseless GPs

Given a test set $\mathbf{X}_*$ of size $N_* \times d$ containing N_* points in $\mathbf{R}_d$, we want to predict function outputs y_*.

We can write:

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{y}_* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu \\ \mu_* \end{pmatrix}, \begin{pmatrix} \mathbf{K} & \mathbf{K}_* \\ \mathbf{K}_*^T & \mathbf{K}_{**} \end{pmatrix} \right)$$

where:

$$\mathbf{K} = Ker(\mathbf{X}, \mathbf{X}) \in \mathbb{R}^{N \times N}$$
$$\mathbf{K}_* = Ker(\mathbf{X}, \mathbf{X}_*) \in \mathbb{R}^{N \times N_*}$$
$$\mathbf{K}_{**} = Ker(\mathbf{X}_*, \mathbf{X}_*) \in \mathbb{R}^{N_* \times N_*}$$

Noiseless GPs

Conditioning Gaussian results into another Gaussian. We get the following mean and covariance matrix postconditioning.

$$p(\mathbf{y}_*|\mathbf{X}_*, \mathbf{X}, \mathbf{y}) \sim \mathcal{N}(\mu', \Sigma')$$

where:

$$\mu' = \mu_* + \mathbf{K}_*^T \mathbf{K}^{-1}(\mathbf{x} - \mu)$$
$$\Sigma' = \mathbf{K}_{**} - \mathbf{K}_*^T \mathbf{K}^{-1} \mathbf{K}_*$$

GP Updates - Cholesky Decomposition

If we notice the last slide, we used a matrix inversion while trying to find the updated covariance matrix. In practical settings, matrix inversions are usually avoided due to multiple reasons. Some of them being:

1. Numerically unstable
2. Computationally heavy

In some cases where we can avoid matrix inversion, therefore it is preferred to do so.

GP Updates - Cholesky Decomposition

$\mathbf{K}$ is a semi positive definite matrix. Such matrices can be decomposed using cholesky decomposition. Which can be written as:

$$\mathbf{A} = \mathbf{L}\mathbf{L}^T$$

where: $\mathbf{L}$ is a lower triangular matrix with real and positive diagonal entries.

We can thus re-write the posterior mean and covariance as:

$$p(y_*|X_*, X, y) \sim \mathcal{N}(\mu', \Sigma')$$
$$K = LL^T$$

GP Updates - Cholesky Decomposition

We are now going to use the $\backslash$ as follows: if $\mathbf{A} = \mathbf{B}$, then $= \mathbf{B}n\mathbf{A}$.
We now have:

$$\alpha = \mathbf{K}^{-1}(\mathbf{x} - \mu)$$

$$\text{or, } \alpha = \mathbf{L}\mathbf{L}^{T^{-1}}(\mathbf{x} - \mu)$$

$$\text{or, } \alpha = \mathbf{L}^{-T}\mathbf{L}^{-1}(\mathbf{x} - \mu)$$

$$\text{Let, } \mathbf{K}^{-1}(\mathbf{x} - \mu) = \beta$$

$$\text{Thus, } \mathbf{L}^{-T}\mathbf{L}^{-1}(\mathbf{x} - \mu) = \beta$$

$$\text{Let, } \mathbf{L}^{-1}(\mathbf{x} - \mu) = \gamma$$

$$\text{Thus, } \mathbf{L}\gamma = \mathbf{x} - \mu$$

$$\text{Thus, } \gamma = \mathbf{L} \setminus (\mathbf{x} - \mu)$$

$$\text{Thus, } \alpha = \mathbf{L}^{T} \setminus (\mathbf{L} \setminus (\mathbf{x} - \mu))$$

We avoided matrix inversion by instead solving a system of eq.

GP Updates - Cholesky Decomposition

Thus, we can find the posterior mean as:

$$\mu' = \mu_* + \mathbf{K}_*^T \alpha$$

We also know that

$$\Sigma' = \mathbf{K}_{**} - \mathbf{K}_*^T \mathbf{K}^{-1} \mathbf{K}_*$$

Let us now define

$$\mathbf{v} = \mathbf{L} \setminus \mathbf{K}_*$$

$$\text{or, } \mathbf{v} = \mathbf{L}^{-1} \mathbf{K}_*$$

$$\text{Thus, } \mathbf{v}^T = \mathbf{K}_*^T \mathbf{L}^{-T}$$

$$\text{Thus, } \mathbf{v}^T \mathbf{v} = \mathbf{K}_*^T \mathbf{L}^{-T} \mathbf{L}^{-1} \mathbf{K}_*$$

$$\text{Thus, } \mathbf{v}^T \mathbf{v} = \mathbf{K}_*^T \mathbf{K}^{-1} \mathbf{K}_* = \mathbf{K}_{**} - \Sigma'$$

Noiseless GPs

Initially, we had assumed that the functional evaluations were free of noise. That is.

$$y_i = f(\mathbf{x}_i)$$

But there can be cases where we have noise in the observed data as well.

$$y_i = f(\mathbf{x}_i) + \epsilon$$
$$\epsilon \sim \mathcal{N}(0, \sigma_y^2)$$

This adds uncertainty to the training points as well. We can counter this issue by updating the covariance matrix $\mathbf{K}$ in the following way.

$$\mathbf{K}_y = \sigma_y^2 \mathbf{I_n} + \mathbf{K}$$

One can see the differences between the two in the above plots. The left one is of a noisy GP, where even after adding the data points, the uncertainty doesn't go to zero, whereas the uncertainty reaches zero for the noiseless GP.