IEE575 - Applied Stochastic Operations Research Models
Lab 1

Tanmay Khandait
tkhandai@asu.edu

April 7, 2025

Hello, World!

- Hi! I'm **Tanmay Khandait**, your TA for the lab sessions.
- I'm a PhD student working on machine learning, especially probabilistic models.
- Ask questions anytime—I'm here to help!
- Labs can be chill but insightful—let's make GPs less mysterious together.
- You can reach me via , or catch me during office hours.
  I will let you all know the time in the announcements :).

Logistics

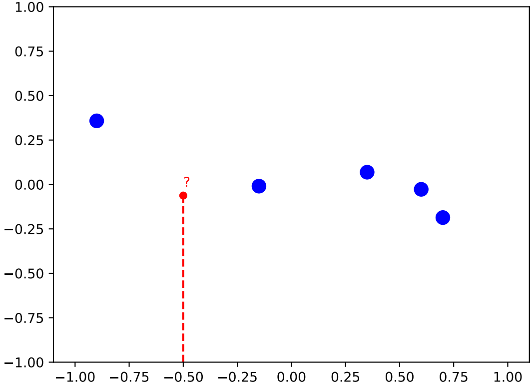
- April 07 (Monday): Introduction to Python, Foundations of Gaussian Process (GP) Regressors
- April 14 (Monday): Deep Dive into GP Regressors, Starting Implementation in Python
- April 16 (Wednesday): Coding Gaussian Process Regressors
- April 21 (Monday): Bayesian Optimization (BO) Concepts, Integrating GP within the BO Framework
- April 23 (Wednesday): Coding Bayesian Optimization, Practical Applications of BO
- April 28 (Monday): Practical Application of BO, Lab Exam Prep
- April 30 (Wednesday): Lab Exam Day

Logistics

- Bring Your Laptop + Charger to Every Session
- Set Up Python/Google Colab
    - Install Python (with NumPy, Matplotlib, scikit-learn) **or**
    - Use Google Colab: colab.research.google.com
    - All the codes will be available in the modules and the github repository https://github.com/cpslab-asu/IEE575_Spring2025
- We will be following this tutorial for the Gaussian Process Regressors.
    - Gaussian Process Tutorial — arxiv.org/abs/2009.10862
- **Mandatory Attendance**
    - Engage in discussions, ask questions, and collaborate!
    - Labs build toward the exam — attendance is 10% of your module grade.

Motivation

Consider the following regression problem:
What would be the output be at the point in red?



Motivation

Consider the following regression problem:
What would be the output be at the point in red?

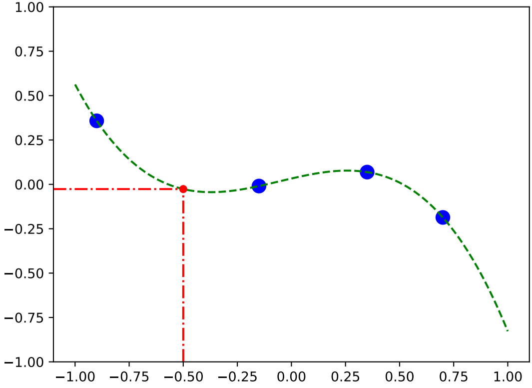


Figure: Here is an illustration of using non-linear regression.

Motivation

Consider the following regression problem:
What would be the output be at the point in red?

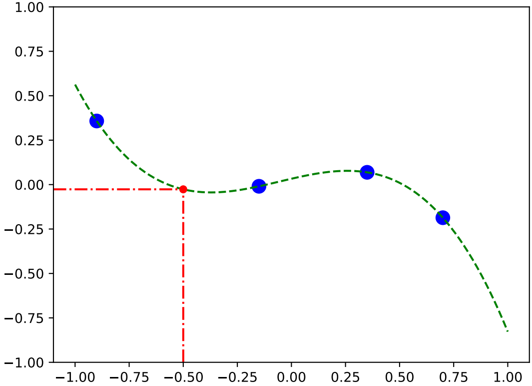


Figure: Here is an illustration of using non-linear regression.

However, in addition to this, we also need uncertainty estimates.

Motivation

Can we do this using what we know about Gaussian Distributions?

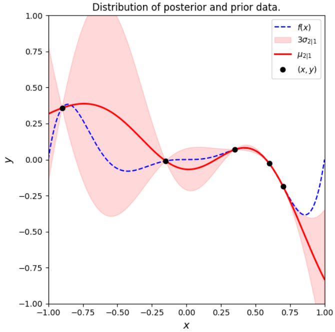


Figure: Here is an illustration of using non-linear regression.

Plan for the next 2 sessions

Today:

- Introduction to Python.
- Revisit Univariate and Multivariate Gaussian Distributions.

Introduction to Python

Univariate Gaussian Distribtution

- A random variable X is normally distributed with mean μ and variance σ^2 if it has the probability density function of X as:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

.

  - In this expression, you see the squared difference between the variable x and its mean, μ.
  - This value will be minimized when x is equal to μ.
  - The quantity $-\frac{x-\mu^2}{\sigma^2}$ will take its largest value when x is equal to μ or likewise since the exponential function is a monotone function, the normal density takes a maximum value when x is equal to μ.
  - The variance σ^2 defines the spread of the distribution about that maximum. If it is large, then the spread is going to be large, otherwise, if the value is small, then the spread will be small.
- If X is random variable that follows a normal distribution with mean μ and variance σ^2, then we will denote it as $X \sim \mathcal{N}(\mu, \sigma^2)$.

If you are not familiar, play around with this link:
`https://demonstrations.wolfram.com/TheNormalDistribution/`

- The multivariate normal distribution of a k-dimensional random vector $\boldsymbol{X} = (X_1, X_2, ..., X_k)^T$ is written as $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$.
- The probability density function is given as follows:

$$f(\boldsymbol{x}) = \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

  where,
  - $\boldsymbol{\mu} = E[\boldsymbol{X}] = (E[X_1], E[X_2], ..., E[X_k])^T$
  - $\Sigma_{i,j} = E[(X_i - \mu_i)(X_j - \mu_j)] = Cov[X_i, X_j]$

Fun fact: correctly constructed covariance matrices are always symmetric and positive semi-definite. And thus invertible.
But let's look at the covariance matrix in detail.

Covariance Matrix I

Let us look at the covariance matrix in detail: $\Sigma_{i,j} = E[(X_i - \mu_i)(X_j - \mu_j)] = Cov[X_i, X_j]$

$$\Sigma_{k\times k} = \begin{bmatrix} \sigma_1^2 & cov(x_1, x_2)^2 & \dots & cov(x_1, x_{k-1}) & cov(x_1, x_k) \\ cov(x_2, x_1)^2 & \sigma_2^2 & \dots & cov(x_2, x_{k-1}) & cov(x_2, x_k) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ cov(x_{k-1}, x_1)^2 & cov(x_{k-1}, x_2)^2 & \dots & \sigma_{k-1}^2 & cov(x_{k-1}, x_k)^2 \\ cov(x_k, x_1)^2 & cov(x_k, x_2)^2 & \dots & cov(x_k, x_{k-1})^2 & \sigma_k^2 \end{bmatrix}$$

- The diagonal elements of the matrix contain the variances of the variables.
- The off-diagonal elements contain the covariance between all possible pairs of variables.

> **Question:** What happens when the off-diagonal elements are 0? What does it mean?

Let us look at a bivariate Gaussian Distribution for different values of the covariance matrix.

Covariance Matrix II

$$\boldsymbol{\mu}_{2\times 1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma_{2\times 2} = \begin{bmatrix} 1 & 0.0 \\ 0.0 & 1 \end{bmatrix}$$

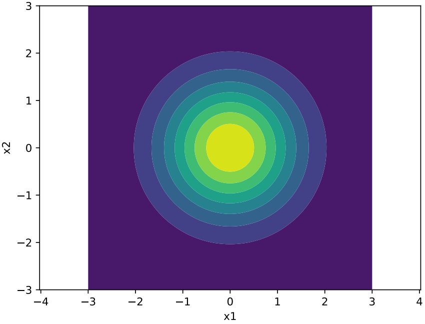


Figure: Distribution of Points

Covariance Matrix III

$$\boldsymbol{\mu}_{2\times 1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma_{2\times 2} = \begin{bmatrix} 1 & 0.7 \\ 0.7 & 1 \end{bmatrix}$$

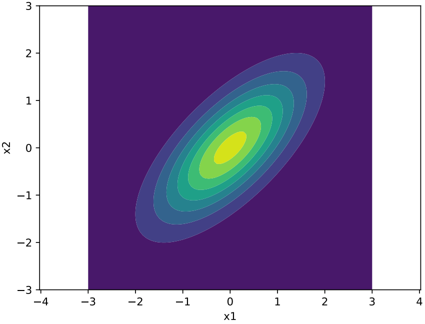


Figure: Distribution of Points

$$\boldsymbol{\mu}_{2\times 1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma_{2\times 2} = \begin{bmatrix} 1 & -0.7 \\ -0.7 & 1 \end{bmatrix}$$

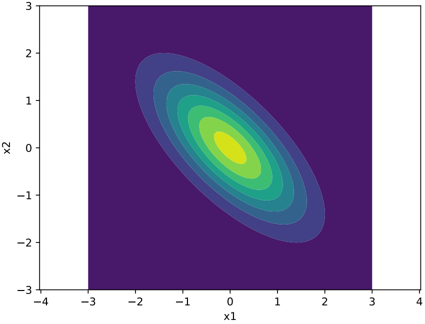


Figure: Distribution of Points