Gaussian Process Assignment

Professor: Kim Jin Young

Student: Nguyen Bui Ngoc Han - 228654

Introduction

A Glimpse into GPs:

Gaussian Process is a machine learning technique
GP works by modeling the underlying true function $y(x)$ as a realization of a Gaussian random process
Unlike traditional algorithms that focus on single-point estimates, GPs model the underlying function as a distribution, allowing them to quantify prediction uncertainty The key advantages of GPs are:

- **Flexibility**: handles complex, non-linear relationships
- **Interpretability**: model parameters offer insights into learned functions
- **Uncertainty quantification**: provides confidence estimates for predictions
- **Bayesian framework**: integrates prior knowledge and updates beliefs with new data

Application:

Apply in various domains, including:

- **Regression task**: time series prediction, stock prices...
- **Classification tasks**: spam filtering, document categorization...
- **Active learning**: efficient data acquisition for model improvement
- **Robotics and control**: designing optimal control strategies.
- **Scientific computing**: modeling complex physical phenomena.

GP Training and Testing Procedures

1. Gaussian Process Definition

A Gaussian Process is a collection of random variable $\{\mathbf{X}_i\}_{i=1}^{n}$, such that any subset/collection of these variable is jointly Gaussian

$$\mathbf{X}_i, \ldots, \mathbf{X}_j \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{1}$$

It its basic form, a Gaussian Process $f(.)$ is fully characterized by a mean μ, a variance σ^2, and a **kernel function** $K(x, x^*)$, such that a finite collection of $\mathbf{f} = [f(x_1), f(x_2), .., f(n)]$ follows a multivariate Gaussian distribution.

$$\mathbf{f} \sim \mathcal{N}(\mathbf{1}\mu, \sigma^2\mathbf{K}) \tag{2}$$

where $\mathbf{1}$ is a vector with n ones, and $\mathbf{K}$ is the correlation matrix, with its element $\mathbf{K}_{i,j} = K(x_i, x_j)$.

2. Gaussian Kernel Function

A one-dimension Gaussian kernel $K(x_i, x_j)$ is expressed as:

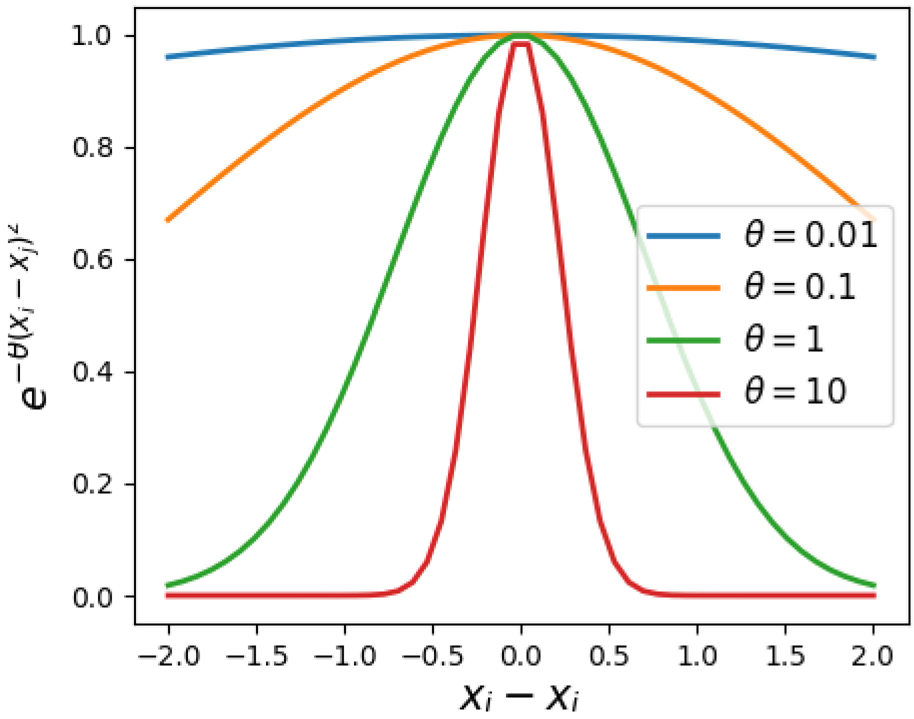
$$K(x_i, x_j) = e^{-\theta(x_i - x_j)^2} \tag{3}$$

where θ is a kernel parameter that controls the correlation strength. Similarly, a m-dimensional Gaussian kernel is expressed as:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\big[\sum_{k=1}^{m} \theta_k (x_i^k - x_j^k)^2\big] \tag{4}$$

which is simply a series of multiplication of the one-dimensional Gaussian kernel for each feature. Here, we have the kernel parameter $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_m]$.

The impact of θ on the correlation strength



The length-scale hyperparameter θ controls how rapidly the unknown function f(x) varies with input x in the Gaussian process model. Large length-scales imply slow variation, so f(x) changes little even for distant inputs. This means x has low predictive power on f. Smaller length-scales indicate f changes rapidly with x, so nearby points can have very different outputs. Therefore, x has high predictive influence. Tuning this hyperparameter allows controlling the input sensitivity assumptions in the Gaussian process.

3. GP Model Training

Maximum likelihood estimation is used to derive μ, σ^2 and $\boldsymbol{\theta}$. The likelihood L of observing the labels $(y_1, y_2, \ldots, y_n)$ of the training instances $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$ is expressed as:

$$L(\boldsymbol{y}|\mu, \sigma^2, \boldsymbol{\theta}) = \frac{1}{\sqrt{(2\pi\sigma^2)^n|\boldsymbol{K}|}}\exp[-\frac{1}{2\sigma^2}(\boldsymbol{y}-\mathbf{1}\mu)^T\boldsymbol{K}^{-1}(\boldsymbol{y}-\mathbf{1}\mu)] \tag{5}$$

where $\boldsymbol{y} = [y_1, y_2, \ldots y_n]$ and $\boldsymbol{K}$ is the correlation matrix of the training instances. In practice, the logarithm of the likelihood L is maximized to avoid round-off error:

$$\ln(L) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2}\ln(|\boldsymbol{K}|) - \frac{1}{2\sigma^2}(\boldsymbol{y}-\mathbf{1}\mu)^T\boldsymbol{K}^{-1}(\boldsymbol{y}-\mathbf{1}\mu) \tag{6}$$

By setting the derivatives of $\ln(L)$ with respect to μ and σ^2 to zero, we can derive the analytical expressions for their optimum values:

$$\mu = (\mathbf{1}^T\boldsymbol{K}^{-1}\mathbf{1})^{-1}\mathbf{1}^T\boldsymbol{K}^{-1}\boldsymbol{y} \tag{7}$$

$$\sigma^2 = \frac{1}{n}(\boldsymbol{y}-\mathbf{1}\mu)^T\boldsymbol{K}^{-1}(\boldsymbol{y}-\mathbf{1}\mu) \tag{8}$$

For $\boldsymbol{\theta}$, its estimation requires solving an auxiliary optimization problem:

$$\boldsymbol{\theta} = \textcolor{red}{\backslash\text{argmax}}_\theta[\frac{n}{2}\ln(\sigma^2) - \frac{1}{2}\ln(|\boldsymbol{K}|)] \tag{9}$$

Equation (9) is obtained via substituting Equation (8) into Equation (6) and removing the constant term $-\frac{n}{2}\ln(2\pi)$.

4. GP Model Prediction

To predict f^* at $\boldsymbol{x}^*$ with a trained GP model, first of all, we write out the joint distribution of f^* and $\boldsymbol{y}$ (i.e., the observed labels of the training instances):

$$\begin{pmatrix} \boldsymbol{y} \\ f^* \end{pmatrix} \sim \mathcal{N}\left(\mu, \quad \sigma^2\begin{pmatrix} \boldsymbol{K} & \boldsymbol{k}^* \\ \boldsymbol{k}^{*T} & 1 \end{pmatrix}\right) \tag{10}$$

where $\boldsymbol{k}^*$ is a correlation vector between the testing and training instances, with its i-th element being $k_i^* = K(x^*, x_i)$.

In a second step, we derive the distribution of f^* conditioned on $\boldsymbol{y}$ from their joint distribution. This conditional distribution of f^* is written as $f^*|\boldsymbol{y} \sim \mathcal{N}(\mu^*, \Sigma^*)$, with

$$\mu^* = \mu + \boldsymbol{k}^{*T}\boldsymbol{K}^{-1}(\boldsymbol{y}-\mathbf{1}\mu) \tag{11}$$

$$\Sigma^* = \sigma^2(1 - \boldsymbol{k}^{*T}\boldsymbol{K}^{-1}\boldsymbol{k}^*) \tag{12}$$

$f^*|\boldsymbol{y} \sim \mathcal{N}(\mu^*, \Sigma^*)$ fully charactizes the GP predictions at $\boldsymbol{x}^*$

Mathematical Concept of Gaussian Process Regression (GPR)

1. Key Concepts of Gaussian Process Regression

For regression tasks, *a non-parametric*, *probabilistic machine learning model* called Gaussian Process (GP) regression is employed
A multivariate Gaussian distribution is assumed to produce the data points in GP regression, and the objective is to infer this distribution.

2. Mean and Covariance Functions

- **Mean function**: often a constant or a linear function
- **Covariance function**: common kernels include
    - **Squared exponential**: Smooth, infinitely differentiable functions
    - **Matern kernel**: Smoothness controlled by a hyperparameter, offering flexibility.
    - **Polynomial kernel**: Represents non-linear relationships.

3. Bayesian Framework:

Employs a Bayesian framework, incorporating *prior knowledge* about the underlying function through *the mean* and *kernel functions*.
As new data arrives, the model updates its beliefs by *updating the posterior* distribution of the function.

4. Key Formula:

- **Covariance Function**: $k(x_i, x_j) = exp(-||x_i - x_j||^2/2l^2)$
    - This is the squared exponential kernel, a common choice for smooth, continuous functions
    - l is length scale that controls the smoothness of the function
- **Marginal likelihood**: $\log p(y|X) = -\frac{1}{2}y^TK^{-1}y - \frac{1}{2}\log|K| - \frac{N}{2}\log(2\pi)$
    - Represents the likelihood of the observed data y given the input data X and the kernel matrix K
    - Maximizing the marginal likelihood estimates the model parameters
- **Predictive distribution**: $p(y^*|x^*, y, X) = N(\mu^*, {\sigma^*}^2)$
    - Mean prediction: $\mu^* = k(x^*, X)^TK^{-1}y$
    - Variance prediction: ${\sigma^*}^2 = k(x^*, x^*) - k(x^*, X)^TK^{-1}k(x^*, X)$

5. Hyperparameters

Key Hyperparameters

- **Mean function hyperparameters**: control the prior belief about the latent function
- **Kernel hyperparameters**: control the smoothness and complexity of the decision boundary, includes:
    - **Scale parameters**: determine the characteristic length scale of the kernel function, influencing how quickly the similarity between input points decays with distance.
    - **Noise variance**: accounts for inherent noise in the data, preventing overfitting.

Hyperparameter Tuning Techniques:

- **Maximum likelihood estimation**: maximize the likelihood of the observed data given the model

- **Cross-validation**: splits the data into training and validation sets, evaluating different hyperparameter values on the validation set to minimize the classification error.
- **Bayesian optimization**: utilizes a probabilistic framework to efficiently explore the hyperparameter space and find values that maximize a pre-defined objective function

Mathematical Concept of Gaussian Process Classification (GPC)

1. Key Concepts of Gaussian Process Classification

A Gaussian Process Extension for Classification Problems is called GPC.
Unlike regression, which predicts continuous outputs, GPC estimates the probability of a data point belonging to a specific class.
Any finite set of class labels follows a joint Gaussian distribution, allowing us to reason about the relationships between them and estimate their probabilities for new data points.

2. Mathematical Framework

- **Mean and covariance functions**:
  - **Mean function**: constant, linear, or more complex functions depending on the problem.
  - **Covariance function**: squared exponential, Matern, polynomial kernels, each controlling the smoothness and complexity of the decision boundary.
- **Bayesian framework**:
  - **Likelihood function**: measures how likely the observed class labels are, given the latent function values and the chosen kernel function.
  - **Posterior distribution**: represents the updated belief about the latent function after incorporating the data

3. Key Formula

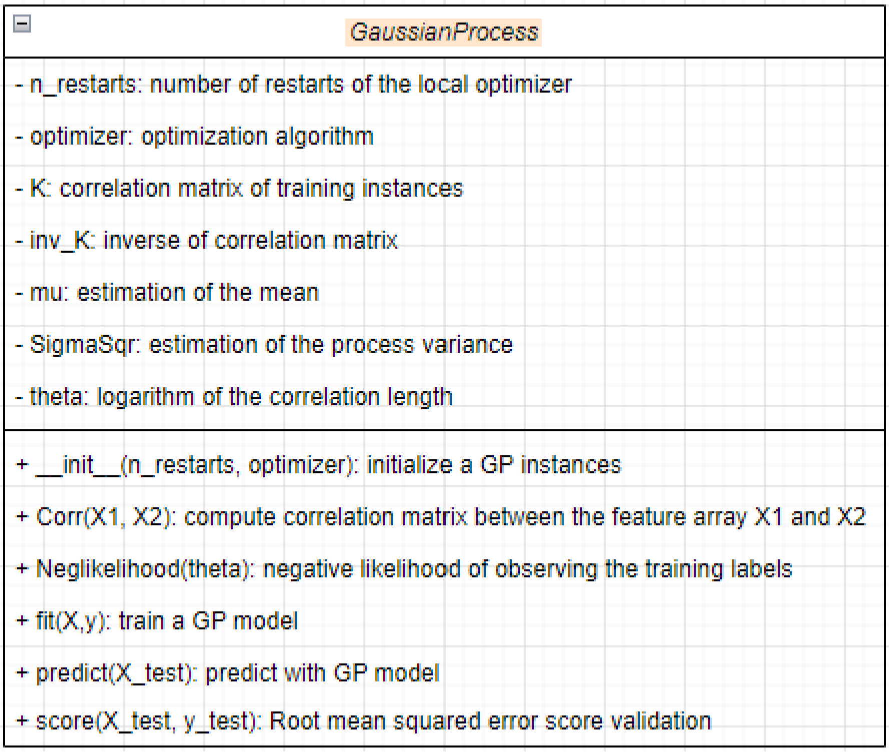
- **Posterior distribution**: $p(f|x, y) \propto N(\mu_f(x), K_f(x, x))$
  - Mean update: $\mu_f(x) = k(x, X)^T K^-1y$
  - Variance update: $K_f(x, x) = k(x, x) - k(x, X)^T K^-1k(x, X)$
  - $k(x_i, x_j)$ is the kernel function between input points x_i
  - X is the matrix of all training data points.
  - y is the vector of observed class labels.
  - K is the kernel matrix, containing pairwise kernel values for all training data points.
- **Predictive distribution**: Represents the class probability for a new data point x^*
  - $p(y^* = c|x^*, f) = \sigma(f(x^*))$
  - $\sigma(z)$ is the sigmoid function, mapping latent function values to class probabilities (0 for class 1, 1 for class 2).
  - $f(x^*)$ is the predicted latent function value for the new data point.

4. Hyperparameters

As the same in GPR

Simple implementation [GPR]

Gaussian Process class diagram



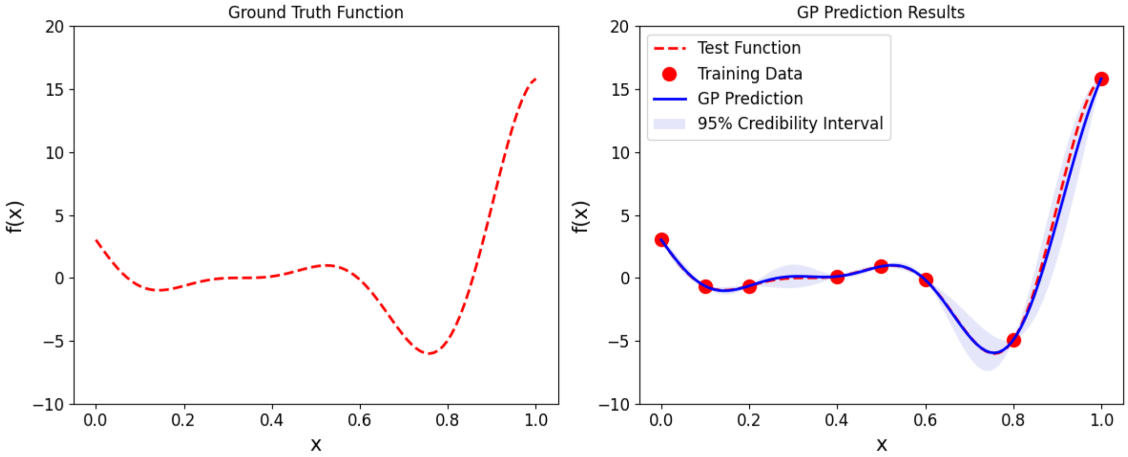
GaussianProcess

- n_restarts: number of restarts of the local optimizer
- optimizer: optimization algorithm
- K: correlation matrix of training instances
- inv_K: inverse of correlation matrix
- mu: estimation of the mean
- SigmaSqr: estimation of the process variance
- theta: logarithm of the correlation length

+ __init__(n_restarts, optimizer): initialize a GP instances
+ Corr(X1, X2): compute correlation matrix between the feature array X1 and X2
+ Neglikelihood(theta): negative likelihood of observing the training labels
+ fit(X,y): train a GP model
+ predict(X_test): predict with GP model
+ score(X_test, y_test): Root mean squared error score validation

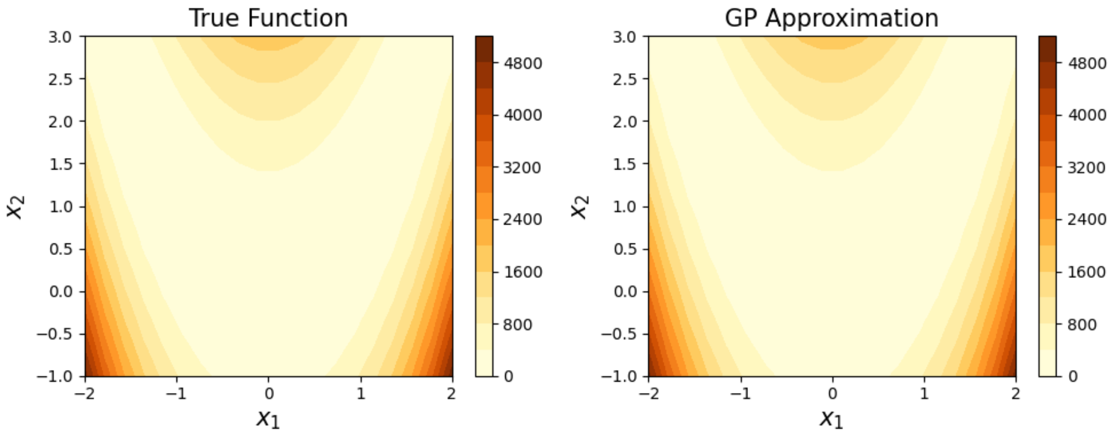
1D function test\ Test RMSE: 6.269640

$$y = (6x-2)^2\cos(12x-4) \quad x \in [0,1]$$



2D function test\ Test RMSE: 9.603220

$$y = (1-x_1)^2 + 100(x_2 - x_1^2)^2 \quad x_1 \in [-2,2], x_2 \in [-1,3]$$


True Function
GP Approximation
x_1
x_2
−2
−1
0
1
2
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
800
1600
2400
3200
4000
4800