Review

Tutorial on Bayesian Optimization

Loc Nguyen *

Posted Date: 16 March 2023

doi: 10.20944/preprints202303.0292.v1

Keywords: Bayesian optimization; Gaussian process regression; acquisition function; machine learning; reinforcement learning

Review

Tutorial on Bayesian Optimization

Loc Nguyen

Loc Nguyen's Academic Network, Vietnam Homepage: www.locnguyen.net; ng_phloc@yahoo.com

Abstract: Machine learning forks into three main branches such as supervised learning, unsupervised learning, and reinforcement learning where reinforcement learning is much potential to artificial intelligence (AI) applications because it solves real problems by progressive process in which possible solutions are improved and finetuned continuously. The progressive approach, which reflects ability of adaptation, is appropriate to the real world where most events occur and change continuously and unexpectedly. Moreover, data is getting too huge for supervised learning and unsupervised learning to draw valuable knowledge from such huge data at one time. Bayesian optimization (BO) models an optimization problem as a probabilistic form called surrogate model and then directly maximizes an acquisition function created from such surrogate model in order to maximize implicitly and indirectly the target function for finding out solution of the optimization problem. A popular surrogate model is Gaussian process regression model. The process of maximizing acquisition function is based on updating posterior probability of surrogate model repeatedly, which is improved after every iteration. Taking advantages of acquisition function or utility function is also common in decision theory but the semantic meaning behind BO is that BO solves problems by progressive and adaptive approach via updating surrogate model from a small piece of data at each time, according to ideology of reinforcement learning. Undoubtedly, BO is a reinforcement learning algorithm with many potential applications and thus it is surveyed in this research with attention to its mathematical ideas. Moreover, the solution of optimization problem is important to not only applied mathematics but also AI.

Keywords: Bayesian optimization; Gaussian process regression; acquisition function; machine learning; reinforcement learning

1. Introduction

Given target function $y = f(x)$, optimization problem is to find out extremizer x^* so that $f(x^*)$ gets extreme value $y = f(x^*)$. As a convention, $f(x)$ is scalar-by-vector function whose output (observed value or evaluated value) y is scalar and whose variable $\boldsymbol{x}$ is n-dimension vector. The extremizer x^* can be minimizer or maximizer so that $y^* = f(x^*)$ is minimum or maximum, respectively. Optimization problem is specified as follows:

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmin}} f(\boldsymbol{x})$$
$$\text{Or } \boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmax}} f(\boldsymbol{x}) \quad (1.1)$$

If the extremizer x^* is local minimizer or local maximizer, the optimization problem is local optimization problem where traditional methods such as Newton-Raphson and gradient descent are perfect solutions but they require $f(x)$ is totally convex for minimization (concave for maximization). The problem becomes much more complex if $f(x)$ is not totally convex (concave) which leads that x^* is global extremizer. This is the global optimization problem which is mentioned in this research. In literature, x^* is minimizer by default but in context of Bayesian optimization (BO) it is better to consider x^* as maximizer because BO mainly relates to probabilistic distributions whose peaks are concerned much. However, it is not serious because minimization is inverse of maximization, for example:

$$\underset{\boldsymbol{x}}{\operatorname{argmin}} f(\boldsymbol{x}) \sim \underset{\boldsymbol{x}}{\operatorname{argmax}}\big(-f(\boldsymbol{x})\big)$$

There are three approaches to solve (global) optimization problem such as analytic approach, probabilistic approach, and heuristic approach. Analytic approach applies purely mathematical tools into finding out optimizers such as approximation, cutting plane, branch and bound, and interval method, in which these methods focus on analytic essence of algebraic target function. Probabilistic approach considers looking for optimizers as random selection but such random selection is guided by some probabilistic model so as to reach an optimizer. Heuristic approach which is the most flexible one among three approaches tries to apply or imitate heuristic assumptions into searching for optimizers. It does not concern much mathematical reasonings because feasibility and effectiveness are most important. As usual, heuristic approach imitates natural activities, for example, particle swarm optimization (PSO) simulates how a flock of birds search for food. Evolutional algorithms like PSO and ant bee colony (ABC) which are inspired from biological activities are popular methods of heuristic algorithms. However, there are some implicit connections between heuristic approach (concretely, evolutional algorithms) and probabilistic approach that I mentioned in a research about minima distribution (Nguyen, 2022).

Bayesian optimization (BO) belongs to the probabilistic approach. It is based on Bayesian inference which considers parameter as random variable and updates posterior probability of parameter based on evidence and prior probability. Because BO does not impact directly on target function *f*($\boldsymbol{x}$), it must model *f*($\boldsymbol{x}$) as a probabilistic model and then define an acquisition function for such probabilistic model. BO solves the optimization problem by maximizing the acquisition function instead of maximizing the target function. Shortly, two main tasks of BO are:

1. Modeling f(x) by the probabilistic model called surrogate model (Shahriari, Swersky, Wang, Adams, & Freitas, 2016, pp. 149-150).
2. Defining the acquisition function for the surrogate model so that it is possible to maximize the acquisition function.

Posterior probability of surrogate model in BO is updated continuously along with maximizing acquisition function continuously until a maximizer of target function is reached. Following is pseudo code of BO.

Table 1.1. BO algorithm.

```
Model $f(\boldsymbol{x})$ by the surrogate model $S(f \mid \boldsymbol{D}_n)$ where sample $\boldsymbol{D}_n$ is a set of variable values $\boldsymbol{x}_i$.
Define the acquisition function $\alpha(\boldsymbol{x} \mid f, S)$ based on both $f(\boldsymbol{x})$ and $S(f \mid \boldsymbol{D}_n)$ whose variable is $\boldsymbol{x}$.
Initialize $n$.
Initialize randomly $\boldsymbol{D}_n = \{\boldsymbol{x}_1, \boldsymbol{x}_2,\ldots, \boldsymbol{x}_n\}$.

While maximizer $\boldsymbol{x}^*$ is not reached or the number of iterations is not many enough
Update posterior probability of $S(f \mid \boldsymbol{D}_n)$ with sample $\boldsymbol{D}_n$.
Determine acquisition function $\alpha(\boldsymbol{x} \mid f, S)$ with $S(f \mid \boldsymbol{D}_n)$.
Find $\boldsymbol{x}_{n+1}$ as a maximizer of $\alpha(\boldsymbol{x} \mid f, S)$ with regard to $\boldsymbol{x}$.
                    $\boldsymbol{x}_{n+1} = \underset{\boldsymbol{x}}{\operatorname{argmax}}\, \alpha(\boldsymbol{x}|f,S)$
(Checking whether $\boldsymbol{x}_{n+1}$ is maximizer $\boldsymbol{x}^*$).
Add $\boldsymbol{x}_{n+1}$ to sample $\boldsymbol{D}_n$.
                    $\boldsymbol{D}_n = \boldsymbol{D}_n \cup \{\boldsymbol{x}_{n+1}\}$
Increase $n = n + 1$.
End while
```

In the table above, there is a question about how to check if a given $\boldsymbol{x}_{n+1}$ is the global maximizer $\boldsymbol{x}^*$. The checking task here is implicitly executed by BO applications, for example, if two or more sequential iterations produce the same value $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n$ (or small enough deviation $|\boldsymbol{x}_{n+1} - \boldsymbol{x}_n|$) such value $\boldsymbol{x}_{n+1}$ can be the maximizer $\boldsymbol{x}^*$ or be considered as the maximizer $\boldsymbol{x}^*$ because BO algorithm improves $\boldsymbol{x}_{n+1}$ to be higher and higher by maximizing updated acquisition function after every iteration. Because the checking task is not essential task of BO, it is often not listed in BO literature. Besides, how to define acquisition function can decide which one among maximization or minimization is optimization problem.

If the surrogate model $S(f \mid \boldsymbol{D}_n)$ has explicit parameters related directly to $f(\boldsymbol{x})$, updating posterior probability of $S(f \mid \boldsymbol{D}_n)$ is indeed to update posterior probability of its explicit parameters and then, BO is called parametric BO (PBO). If the surrogate model $S(f \mid \boldsymbol{D}_n)$ has no explicit parameters or its parameters are not related directly to $f(\boldsymbol{x})$ then, BO is called nonparametric BO (NBO). As seen in BO algorithm, BO does not concern algebraic formulation of target function $f(\boldsymbol{x})$ because it only concerns output $y=f(\boldsymbol{x})$ and hence, $f(\boldsymbol{x})$ is a black box with regard to BO. In other words, $f(\boldsymbol{x})$ is an arbitrary mapping with subject to BO and so BO is a flexible solution for global optimization, which has many potential applications. For BO, especially NBO, $f(\boldsymbol{x})$ is a black box except its output $y = f(\boldsymbol{x})$ and so BO maximizes the acquisition function $\alpha(\boldsymbol{x} \mid f, S)$ instead of maximizing directly $f(\boldsymbol{x})$ because BO knows $\alpha(\boldsymbol{x} \mid f, S)$ which is created from $S(f \mid \boldsymbol{D}_n)$ that BO built up before. How to define acquisition function depends on which kind of BO is, for instance, PBO or NBO, and how to define surrogate model. BO and acquisition function will be mentioned in the next sections.

In general, the essence of BO is continuous improvement via updating posterior probability, which follows ideology of reinforcement learning (Shahriari, Swersky, Wang, Adams, & Freitas, 2016, p. 151). Therefore, it is possible to consider BO as a reinforcement learning technique. Before describing subjects related to Bayesian optimization (BO) in detail, it is necessary to mention conventions about mathematical notations. For instance, lowercase and un-bold letters like a, b, c, x, y, and z denote scalar whereas lowercase and bold letters like $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ denote vector. In some cases, uppercase and un-bold letters like A, B, C, X, Y, and Z can denote vector. Uppercase and bold / un-bold letters like A, B, C, $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, X, Y, Z, $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$ denote matrix. Variables can be denoted as x, y, z, $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$, X, Y, Z, $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$ whereas constants can be denoted as a, b, c, $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, A, B, C, $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$. Uppercase and bold / un-bold letters like X, Y, Z, $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$ can denote random variables. However, scalar, vector, matrix, variables, and random variables are stated explicitly in concrete cases. A vector is column vector by default if there is no additional explanation. Superscript "T" denotes transposition operator of vector and matrix, for example, given column vector $\boldsymbol{x}$ then the notation $\boldsymbol{x}^T$ is the row vector which is transposed vector of $\boldsymbol{x}$.

2. Bayesian Optimization

As aforementioned, Bayesian optimization (BO) solves the optimization problem by maximizing the acquisition function $\alpha(\boldsymbol{x} \mid f, S)$ which is derived from the surrogate model $S(f \mid \boldsymbol{D}_n)$ where $\boldsymbol{D}_n$ is the current sample. The surrogate model $S(f \mid \boldsymbol{D}_n)$, in turn, was a probabilistic representation of target function $f(\boldsymbol{x})$. The way to define surrogate model decides the kind of BO where there are two kinds of BO such as parametric BO (PBO) and nonparametric BO (NBO). PBO implies that parameters of the surrogate model $S(f \mid \boldsymbol{D}_n)$ were included explicitly in target function $f(\boldsymbol{x})$. Otherwise, the surrogate model $S(f \mid \boldsymbol{D}_n)$ of NBO has no explicit parameters or its parameters were not included explicitly in target function $f(\boldsymbol{x})$. Firstly, PBO is surveyed and then NPO is researched.

The simplest PBO inspired by a system consisting of r events X_k is known Bernoulli system, for example, a lottery machine or bandit system has r arms represented by r random variables X_k and probability of an arm k yielding lottery prize is $P(X_k=1)$ (Shahriari, Swersky, Wang, Adams, & Freitas, 2016, pp. 152-153). Such PBO is called Bernoulli PBO where each binary random variable X_k whose outcome is successful ($X_k=1$) or failed ($X_k=0$) follows Bernoulli distribution whose parameter is θ_k ($0 \leq \theta_i \leq 1$).

$$P(X_k|\theta_k) = \begin{cases} \theta_k \text{ if } X_k = 1 \\ 1 - \theta_k \text{ if } X_k = 0 \end{cases}$$

This means:

$$\theta_k = P(X_k = 1|\theta_k) \tag{2.1}$$

Note, the general notation $P(.)$ denotes probability function in both discrete case and continuous case. In Bayesian inference, parameter is considered as random variable which also follows distribution and so suppose each θ_k has prior probability $P(\theta_k)$ and hence, the probability $P(X_k=1)$ of event $X_k=1$ is expectation of Bernoulli distribution $P(X_k=1 \mid \theta_k)$ within the prior probability $P(\theta_k)$.

$$E(\theta_k) = P(X_k = 1) = \int_{\theta_k} P(X_k = 1|\theta_k)P(\theta_k)d\theta_k = \int_{\theta_k} \theta_k P(\theta_k)d\theta_k \tag{2.2}$$

Note, notation $E(.)$ denotes expectation of random variable. Going back to the example of lottery machine, $P(X_k=1)$ is the probability that an arm k yields lottery prize, which is called winning probability of arm k. A gambler does not know exactly which arm is the best one to win a prize and so he tries to choose an arm k having largest winning probability $P(X_k=1)$. Consequently, if he wins ($X_k=1$) then the winning probability $P(X_k=1)$ is increased and otherwise if he loses then $P(X_k=1)$ is decreased. In the next time, the gambler will continue to choose the arm whose winning probability is largest and so on. Suppose the gambler chooses arms through n times are considered as sample $\boldsymbol{D}_n = \{X_{v1}, X_{v2},\ldots, X_{vn}\}$ where each observation X_{vi} indicates that the gambler selects arm v_i at the i^{th} time with result X_{vi} (which equals to 0 or 1). Let D_k denote a sub-sample of $\boldsymbol{D}_n$ in which X_k is selected and let s_k and t_k be the numbers of $X_k=1$ and $X_k=0$, respectively, which are extracted from D_k. Suppose D_k obeys identically independent distribution (iid) criterion, likelihood function of D_k is:

$$P(D_k|\theta_k) = (\theta_k)^{s_k}(1-\theta_k)^{t_k}$$

Sample D_k here is called binomial sample when X_k follows Bernoulli distribution which leads to the event that the likelihood function of D_k above follows binomial distribution. Marginal probability of D_k within the prior probability $P(\theta_k)$ is:

$$P(D_k) = \int_{\theta_k} P(D_k|\theta_k)P(\theta_k)d\theta_k = E\left((\theta_k)^{s_k}(1-\theta_k)^{t_k}\right)$$

Posterior probability of θ_k given sample D_k according to Bayes' rule is:

$$P(\theta_k|D_k) = \frac{P(D_k|\theta_k)P(\theta_k)}{P(D_k)}$$

The probability $P(X_k=1 \mid D_k)$ of $X_k=1$ given D_k is.

$$E(\theta_k|D_k) = P(X_k = 1|D_k) = \int_{\theta_k} P(X_k = 1|\theta_k)P(\theta_k|D_k)d\theta_k = \int_{\theta_k} \theta_k P(\theta_k|D_k)d\theta_k \tag{2.4}$$

Obviously, for the example of lottery machine, the winning probability of arm k is expectation of parameter θ_k within its posterior probability given sample D_k. Therefore, given r parameters as parameter vector $\Theta = (\theta_1, \theta_2,\ldots, \theta_r)^T$, the target function of Bernoulli PBO is essentially an index function specified as follows:

$$f(\boldsymbol{x} = k) = f_k = E(\theta_k|D_k) \text{ where } k = \overline{1,r} \tag{2.5}$$

The optimization problem is specified as follows:

$$k^* = \underset{k}{\operatorname{argmax}} f_k = \underset{k}{\operatorname{argmax}} E(\theta_k|D_k)$$

Obviously, the target function is defined based on the posterior probability of θ_k which is now considered as surrogate model of Bernoulli PBO.

$$S(f|\boldsymbol{D}_n) \equiv P(\theta_k|D_k) \tag{2.6}$$

Bernoulli PBO is a PBO because surrogate model and target function share the same parameter θ_k. Acquisition function of Bernoulli PBO is the same to target function:

$$\alpha(k|f,S) = f_k = E(\theta_k|D_k) \tag{2.7}$$

The problem now is how to specify the posterior probability of θ_k. Suppose the prior probability of each θ_k follows beta distribution:

$$P(\theta_k) = \text{beta}(\theta_k|a_k, b_k) = \frac{\Gamma(a_k + b_k)}{\Gamma(a_k)\Gamma(b_k)}(\theta_k)^{a_k-1}(1-\theta_k)^{b_k-1} \tag{2.8}$$

There are two hyper parameters of beta distribution in which a_k indicates the number of successful outcomes and b_k indicates the number of failed outcomes in $a_k + b_k$ trials. Note, Γ(.) denotes gamma function (Neapolitan, 2003, p. 298) which is continuous form of factorial function.

$$\Gamma(x) = \int_0^{+\infty} t^{x-1}e^{-t}\text{d}t$$

Given prior beta distribution, the probability of X_i=1 is:

$$E(\theta_k) = P(X_k = 1) = \int_{\theta_k} \theta_k \text{beta}(\theta_k|a_k, b_k)d\theta_k = \frac{a_k}{a_k + b_k} \tag{2.9}$$

Given sub-sample D_k the posterior probability of each θ_k is:

$$P(\theta_k|D_k) = \text{beta}(\theta_k|a_k + s_k, b_k + t_k) = \frac{\Gamma(a_k + s_k + b_k + t_k)}{\Gamma(a_k + s_k)\Gamma(b_k + t_k)} * (\theta_k)^{a_k+s_k-1}(1-\theta_k)^{b_k+t_k-1} \tag{2.10}$$

Where s_k and t_k be the numbers of X_k=1 and X_k=0, respectively. Recall that such posterior probability is also called Bernoulli PBO. Because the prior probability $P(\theta_k)$ and the posterior probability $P(\theta_k \mid D_k)$ have the same form, they are called conjugate distributions and such conjugation is very important to Bayesian inference. The probability of X_k=1 given D_k which is also target function is:

$$E(\theta_k|D_k) = P(X_k = 1|D_k) = \int_{\theta_k} \theta_k P(\theta_k|D_k)d\theta_k = \frac{a_k + s_k}{a_k + s_k + b_k + t_k} \tag{2.11}$$

Formulation of surrogate model becomes neat because of the assumption of beta distribution with binomial sample. When prior probability $P(\theta_k)$ follows beta distribution and X_k follows Bernoulli distribution (so that likelihood function $P(D_k \mid \theta_k)$ follows binomial distribution with binomial sample D_k) then we obtain probabilistic conjugation in which the posterior probability $P(\theta_k \mid D_k)$ follows beta distribution. Going back to the example of lottery machine, it is easy to derive Bernoulli PBO with beta distribution as follows (Shahriari, Swersky, Wang, Adams, & Freitas, 2016, p. 153):

Table 2.1. Bernoulli PBO algorithm.

```
Initialize hyper parameters $a_k$ and $b_k$ of $r$ distributions beta($\theta_k \mid a_k, b_k$).

While the number of iterations is not many enough
Find $k_{n+1}$ as an index maximizer among $r$ distributions beta($\theta_k \mid a_k, b_k$) so that $E(\theta_k \mid D_k)$ is maximum.
    $k_{n+1} = \underset{k}{\text{argmax}}\, E(\theta_k|D_k) = \underset{k}{\text{argmax}} \frac{a_k + s_k}{a_k + s_k + b_k + t_k}$
If $X_{k_{n+1}} = 1$ then increasing $a_{k_{n+1}} = a_{k_{n+1}} + 1$.
Otherwise, if $X_{k_{n+1}} = 0$ then increasing $b_{k_{n+1}} = b_{k_{n+1}} + 1$.

Increase $n = n + 1$.
End while
```

Because the expectation $E(\theta_k \mid D_k)$ is essentially an estimate of θ_k given binomial sample according Bayesian statistics, the target function of Bernoulli PBO can be rewritten:

$$f_k = \hat{\theta}_k$$

Where,

$$\hat{\theta}_k = E(\theta_k | D_k)$$

This implies that target function in Bernoulli PBO is "identified" with probabilistic parameter and hence, for example, given lottery machine, each arm has no associated properties, which cannot be applied into more complex system. Therefore, suppose each arm k is associated with a real number vector X_k so that the target function is linear function of p-dimension X_k and vector parameter W as follows (Shahriari, Swersky, Wang, Adams, & Freitas, 2016, p. 154):

$$f(k) = f_k = W^T X_k \text{ where } k = \overline{1, r} \tag{2.12}$$

Note, the superscript "^{T}" denotes transposition operator of vector and matrix. There are r property vectors $\{X_1, X_2,\ldots, X_r\}$ when the number of arms is r. Suppose variable $Y = W^T X_k$ distributes normally with mean $W^T X_k$ and variance σ^2 as follows:

$$Y \sim \mathcal{N}(Y|W^T X_k, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(Y - W^T X_k)^2}{2\sigma^2}\right)$$

Note, the tilde sign "~" indicates probabilistic distribution of a random variable and notation $\mathcal{N}(.|.)$ denotes normal distribution. Essentially, this normal distribution is likelihood function:

$$P(Y|\Theta) = \mathcal{N}(Y|W^T X_k, \sigma^2)$$

Normal distribution is also called Gaussian distribution. According to Bayesian inference, suppose parameter W distributes normally in prior with mean μ_0 and covariance matrix Σ_0 as follows:

$$W \sim \mathcal{N}(W|\mu_0, \Sigma_0) = (2\pi)^{-\frac{p}{2}} |\Sigma_0|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(W - \mu_0)^T \Sigma_0^{-1} (W - \mu_0)\right)$$

The distribution above is multinormal distribution because W is random vector variable. Multinormal distribution is also called multivariate normal distribution or multivariate Gaussian distribution. Suppose parameter σ^2 follows inverse gamma distribution in prior with shape parameter α_0 and scale parameter β_0 as follows:

$$W \sim \mathrm{IG}(\sigma^2|\alpha_0, \beta_0) = \frac{\beta_0^{\alpha_0}}{\Gamma(\alpha_0)} (\sigma^2)^{-\alpha_0 - 1} \exp\left(-\frac{\beta_0}{\sigma^2}\right)$$

Therefore, the parameter of entire system like lottery machine is $\Theta = \theta = \{\mu, \Sigma, \alpha, \beta\}$ and the NBO for this parameter is called normal – inverse gamma NBO (NIG-NBO) because it combines multinormal distribution and inverse gamma distribution by product operator. The prior probability of Θ follows NIG distribution (Shahriari, Swersky, Wang, Adams, & Freitas, 2016, p. 154):

$$P(\Theta) = \mathcal{N}(W|\mu_0, \Sigma_0)\mathrm{IG}(\sigma^2|\alpha_0, \beta_0) \tag{2.13}$$

The prior parameter is $\Theta_0 = \{\mu_0, \Sigma_0, \alpha_0, \beta_0\}$. The sample $\boldsymbol{D}_n$ now is $\boldsymbol{D}_n = \boldsymbol{X} = (\boldsymbol{x}_1^T, \boldsymbol{x}_2^T,\ldots, \boldsymbol{x}_n^T)^T$ of size n where $\boldsymbol{x}_i$ is some X_k that the gambler selects at the i^{th} time, as follows:

$$\boldsymbol{D}_n = \boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}_1^T \\ \boldsymbol{x}_2^T \\ \vdots \\ \boldsymbol{x}_n^T \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix}, \boldsymbol{x}_i = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{ip} \end{pmatrix}$$

Let $\boldsymbol{y} = (y_1, y_2,\ldots, y_n)^T$ be the output vector of $\boldsymbol{X}$ with target function where $y_i = W^T \boldsymbol{x}_i$.

$$\boldsymbol{y} = \begin{pmatrix} y_1 = W^T \boldsymbol{x}_1 \\ y_2 = W^T \boldsymbol{x}_2 \\ \vdots \\ y_n = W^T \boldsymbol{x}_n \end{pmatrix}$$

Fortunately, the posterior probability of Θ given sample $\boldsymbol{D}_n$ is also a NIG distribution which is the probabilistic conjugation with the prior probability. Of course, such posterior probability is NIG-PBO surrogate model.

$$S(f|\boldsymbol{D}_n) \equiv P(\Theta|\boldsymbol{D}_n) = \mathcal{N}(W|\mu_n, \Sigma_n)\mathrm{IG}(\sigma^2|\alpha_n, \beta_n) \tag{2.14}$$

The posterior parameter is $\Theta_n = \{\mu_n, \Sigma_n, \alpha_n, \beta_n\}$. Fortunately, Θ_n was calculated from $\boldsymbol{D}_n$ in literature (Shahriari, Swersky, Wang, Adams, & Freitas, 2016, p. 154).

$$\begin{aligned}
\mu_n &= \Sigma_n(\Sigma_0^{-1}\mu_0 + \boldsymbol{X}^T\boldsymbol{y}) \\
\Sigma_n &= \Sigma_0^{-1} + \boldsymbol{X}^T\boldsymbol{X} \\
\alpha_n &= \alpha_0 + n/2 \\
\beta &= \beta_0 + \frac{1}{2}(\mu_0^T\Sigma_0^{-1}\mu_0 + y^Ty - \mu_n^T\Sigma_n^{-1}\mu_n)
\end{aligned} \tag{2.15}$$

Acquisition function of NIG-PBO is defined from target function and the posterior mean μ_n:

$$\alpha(k|f, S) = \alpha(k|\mu_n) = \mu_n^T X_k \tag{2.16}$$

Note, $\alpha(k \mid \mu_n)$ is a function of index k. When Θ_n is determined, it is easy to derive NIG-NBO algorithm.

Table 2.2. NIG-PBO algorithm.

```
Initialize prior parameters $\Theta_0 = \{\mu_0, \Sigma_0, \alpha_0, \beta_0\}$.

While the number of iterations is not many enough
Update posterior parameter $\Theta_n = \{\mu_n, \Sigma_n, \alpha_n, \beta_n\}$ from $\boldsymbol{X}=\boldsymbol{D}_n$.
    $\mu_n = \Sigma_n(\Sigma_0^{-1}\mu_0 + \boldsymbol{X}^T\boldsymbol{y})$
    $\Sigma_n = \Sigma_0^{-1} + \boldsymbol{X}^T\boldsymbol{X}$
    $\alpha_n = \alpha_0 + n/2$
    $\beta = \beta_0 + \frac{1}{2}(\mu_0^T\Sigma_0^{-1}\mu_0 + y^Ty - \mu_n^T\Sigma_n^{-1}\mu_n)$
Find $k_{n+1}$ as an index maximizer of the acquisition function $\alpha(k \mid \mu_n)$ among $r$ property vectors $X_k$.
    $k_{n+1} = \underset{k}{\mathrm{argmax}}\, \mu_n^T X_k$
Add $X_{k_{n+1}}$ to sample $\boldsymbol{D}_n$.
    $\boldsymbol{D}_n = \boldsymbol{D}_n \cup \{X_{k_{n+1}}\}$
Increase $n = n + 1$.
End while
```

When $f(\boldsymbol{x})$ does not have specific (explicit) aspect or property which becomes an explicit parameter of the surrogate model $S(f \mid D_n)$, the corresponding BO becomes nonparametric BO (NBO). The most popular technique to establish $S(f \mid D_n)$ for NBO is to use Gaussian process regression (GPR) for modeling $S(f \mid D_n)$. In other words, GPR is a surrogate model of NBO. This research focuses on NBO with GPR.

Because kernel function is very important to GPR when it is used to not only build up GPR but also make GPR line smoother. Kernel function measures similarity between two variables (two points), according to that, the closer the two variables are, the larger their kernel function is. Let $\Sigma(\boldsymbol{x}_i, \boldsymbol{x}_j)$ denote kernel function of two variables $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, for example, a popular kernel function is simple squared exponential function.

$$\Sigma_{SE}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \exp\left(-\frac{\left|\boldsymbol{x}_i - \boldsymbol{x}_j\right|^2}{2}\right)$$

In this research the notation |.| can denote absolute value of scalar, length (module) of vector, determinant of matrix, and cardinality of set. Concretely, kernel function $\Sigma(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is used to define covariance function in GPR. As a convention, let $\boldsymbol{x}_{i:j}$ where $i \leq j$ denote a sequential subset of $\boldsymbol{X}$ such that $\boldsymbol{x}_{i:j} = \{\boldsymbol{x}_i, \boldsymbol{x}_{i+1}, \ldots, \boldsymbol{x}_j\}$. Of course, we have $\boldsymbol{x}_{i:i} = \boldsymbol{x}_i$. Given two subsets $\boldsymbol{x}_{i:j}$ and $\boldsymbol{x}_{k:l}$, their covariance function is:

$$\Sigma\left(\boldsymbol{x}_{i:j}, \boldsymbol{x}_{k:l}\right) = \begin{pmatrix} \Sigma(\boldsymbol{x}_i, \boldsymbol{x}_k) & \Sigma(\boldsymbol{x}_i, \boldsymbol{x}_{k+1}) & \cdots & \Sigma(\boldsymbol{x}_i, \boldsymbol{x}_l) \\ \Sigma(\boldsymbol{x}_{i+1}, \boldsymbol{x}_k) & \Sigma(\boldsymbol{x}_{i+1}, X_{k+1}) & \cdots & \Sigma(\boldsymbol{x}_{i+1}, \boldsymbol{x}_l) \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma\left(\boldsymbol{x}_j, \boldsymbol{x}_k\right) & \Sigma\left(\boldsymbol{x}_j, \boldsymbol{x}_{k+1}\right) & \cdots & \Sigma\left(\boldsymbol{x}_j, \boldsymbol{x}_l\right) \end{pmatrix} \tag{2.17}$$

Output of a covariance function is a covariance matrix if such output is invertible and symmetric. For NBO, given sample $\boldsymbol{D}_n = \boldsymbol{x}_{1:n} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, the consequential sequence $y_{1:n} = f(\boldsymbol{x}_{1:n}) = f(\boldsymbol{D}_n) = \{y_1 = f(\boldsymbol{x}_1), y_2 = f(\boldsymbol{x}_2), \ldots, y_n = f(\boldsymbol{x}_n)\}$ is established to distribute normally with prior probability density function (prior PDF) as follows:

$$y_{1:n} | \boldsymbol{x}_{1:n} \sim f\left(y_{1:n} \middle| \boldsymbol{x}_{1:n}, \mu(.), \Sigma(.)\right) = \mathcal{N}\left(y_{1:n} \middle| \mu(y_{1:n}), \Sigma(\boldsymbol{x}_{1:n}, \boldsymbol{x}_{1:n})\right)$$

Where, in this research, notation $f(.|.)$ often denotes probability density function (PDF), which is not concrete function and notation $\mathcal{N}(.|.)$ denotes normal distribution, with note that the general notation $P(.)$ denotes probability function in both discrete case and continuous case whereas the term "PDF" focuses on probability function of continuous variable. Note, $\mu(.)$ and $\Sigma(.)$ are mean function and covariance function, respectively. In this special case, covariance function $\Sigma(.)$ is matrix function defined based on kernel function; exactly $\Sigma(.)$ is matrix-by-vector function if each $\boldsymbol{x}_i$ is vector.

$$\Sigma(\boldsymbol{x}_{1:n}, \boldsymbol{x}_{1:n}) = \begin{pmatrix} \Sigma(\boldsymbol{x}_1, \boldsymbol{x}_1) & \Sigma(\boldsymbol{x}_1, \boldsymbol{x}_2) & \cdots & \Sigma(\boldsymbol{x}_1, \boldsymbol{x}_n) \\ \Sigma(\boldsymbol{x}_2, \boldsymbol{x}_1) & \Sigma(\boldsymbol{x}_2, \boldsymbol{x}_2) & \cdots & \Sigma(\boldsymbol{x}_2, \boldsymbol{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma(\boldsymbol{x}_n, \boldsymbol{x}_1) & \Sigma(\boldsymbol{x}_n, \boldsymbol{x}_2) & \cdots & \Sigma(\boldsymbol{x}_n, \boldsymbol{x}_n) \end{pmatrix}$$

Of course, each element $\Sigma(\boldsymbol{x}_i, \boldsymbol{x}_j)$ of covariance function $\Sigma(\boldsymbol{x}_{1:n}, \boldsymbol{x}_{1:n})$ is kernel function. Recall that

$$\Sigma(\boldsymbol{x}, \boldsymbol{x}_{1:n}) = \begin{pmatrix} \Sigma(\boldsymbol{x}, \boldsymbol{x}_1) \\ \Sigma(\boldsymbol{x}, \boldsymbol{x}_2) \\ \vdots \\ \Sigma(\boldsymbol{x}, \boldsymbol{x}_n) \end{pmatrix}^T, \Sigma(\boldsymbol{x}_{1:n}, \boldsymbol{x}) = \begin{pmatrix} \Sigma(\boldsymbol{x}_1, \boldsymbol{x}) \\ \Sigma(\boldsymbol{x}_2, \boldsymbol{x}) \\ \vdots \\ \Sigma(\boldsymbol{x}_n, \boldsymbol{x}) \end{pmatrix}$$

It is popular to define $\mu(y_{1:n})$ based on $\mu(y)$ separately:

$$\mu(y_{1:n}) = \begin{pmatrix} \mu(y_1) \\ \mu(y_2) \\ \vdots \\ \mu(y_n) \end{pmatrix}$$

Moreover, for BO, mean function $\mu(.)$ is often set to be zero as follows:

$$\mu(y) = 0 \text{ so that } \mu(y_{1:n}) = \begin{pmatrix} \mu(y_1) = 0 \\ \mu(y_2) = 0 \\ \vdots \\ \mu(y_n) = 0 \end{pmatrix} = \boldsymbol{0}^T$$

Given variable $\boldsymbol{x}$, GPR surrogate model is represented by the posterior PDF of $y = f(\boldsymbol{x})$ given $y_{1:n}$, $\boldsymbol{x}_{1:n}$, and $\boldsymbol{x}$ as follows:

$$y | y_{1:n}, \boldsymbol{x}_{1:n}, \boldsymbol{x} \sim f\left(y \middle| y_{1:n}, \boldsymbol{x}_{1:n}, \boldsymbol{x}, \mu(.), \Sigma(.)\right) = \mathcal{N}\left(y \middle| \mu(y | y_{1:n}, \boldsymbol{x}_{1:n}, \boldsymbol{x}), \sigma^2(y | \boldsymbol{x}_{1:n}, \boldsymbol{x})\right)$$

This posterior PDF is derived from interesting properties of normal distribution which will be mentioned in the next section. Note that $\mu(y | y_{1:n}, \boldsymbol{x}_{1:n}, \boldsymbol{x})$ and $\sigma^2(y | \boldsymbol{x}_{1:n}, \boldsymbol{x})$ are mean and variance of the multinormal posterior PDF of y given $y_{1:n}$, $\boldsymbol{x}_{1:n}$, and $\boldsymbol{x}$, respectively.

$$\mu(y|y_{1:n}, \boldsymbol{x}_{1:n}, \boldsymbol{x}) = \mu(y) + \Sigma(\boldsymbol{x}, \boldsymbol{x}_{1:n})\big(\Sigma(\boldsymbol{x}_{1:n}, \boldsymbol{x}_{1:n})\big)^{-1}\big(y_{1:n} - m(y_{1:n})\big)$$

$$\sigma^2(y|\boldsymbol{x}_{1:n}, \boldsymbol{x}) = \Sigma(\boldsymbol{x}, \boldsymbol{x}) - \Sigma(\boldsymbol{x}, \boldsymbol{x}_{1:n})\big(\Sigma(\boldsymbol{x}_{1:n}, \boldsymbol{x}_{1:n})\big)^{-1}\Sigma(\boldsymbol{x}_{1:n}, \boldsymbol{x})$$

Note, $(\Sigma(\boldsymbol{x}_{1:n}, \boldsymbol{x}_{1:n}))^{-1}$ denotes inverse of covariance matrix (output of covariance function) $\Sigma(\boldsymbol{x}_{1:n}, \boldsymbol{x}_{1:n})$. Please pay attention that $m(y_{1:n})$ is a realistic mean which is recurrently determined by previously known $\mu(y \mid y_{1:n0}, \boldsymbol{x}_{1:n0}, \boldsymbol{x})$:

$$m(y_n) = \mu\big(y|y_{1:n_0+n-1}, \boldsymbol{x}_{1:n_0+n-1}, \boldsymbol{x}_n\big)$$

The variance $\sigma^2(y \mid \boldsymbol{x}_{1:n}, \boldsymbol{x})$ is function of only $\boldsymbol{x}$ and so, in practice, mean function $\mu(y)$ is set to be zero so that the mean function $\mu(y \mid y_{1:n}, \boldsymbol{x}_{1:n}, \boldsymbol{x})$ is also function of only $\boldsymbol{x}$ too, as follows (Frazier, 2018, p. 4):

$$\mu_n(\boldsymbol{x}) = \mu(y|y_{1:n}, \boldsymbol{x}_{1:n}, \boldsymbol{x}) = \Sigma(\boldsymbol{x}, \boldsymbol{x}_{1:n})\big(\Sigma(\boldsymbol{x}_{1:n}, \boldsymbol{x}_{1:n})\big)^{-1}\big(y_{1:n} - m(y_{1:n})\big)$$

$$\sigma_n^2(\boldsymbol{x}) = \sigma^2(y|\boldsymbol{x}_{1:n}, \boldsymbol{x}) = \Sigma(\boldsymbol{x}, \boldsymbol{x}) - \Sigma(\boldsymbol{x}, \boldsymbol{x}_{1:n})\big(\Sigma(\boldsymbol{x}_{1:n}, \boldsymbol{x}_{1:n})\big)^{-1}\Sigma(\boldsymbol{x}_{1:n}, \boldsymbol{x})$$

Please pay attention that $m(y_{1:n})$ is a realistic mean which is recurrently determined by previously known $\mu_{n-1}(\boldsymbol{x})$:

$$m(y_n) = \mu_{n-1}(\boldsymbol{x}_n) \tag{2.18}$$

The event that both $\mu_n(\boldsymbol{x})$ and $\sigma_n^2(\boldsymbol{x})$ are functions of only $\boldsymbol{x}$ is necessary to determine acquisition function of BO later with note that $\boldsymbol{x}_{1:n}$ and $y_{1:n}$ were known. GPR surrogate model is rewritten:

$$y|\boldsymbol{x}_{1:n}, \boldsymbol{x} \sim f\big(y|\boldsymbol{x}_{1:n}, \mu(.), \Sigma(.)\big) = \mathcal{N}\big(y|\mu_n(\boldsymbol{x}), \sigma_n^2(\boldsymbol{x})\big) = S(f|\boldsymbol{D}_n) \tag{2.19}$$

Where,

$$\begin{gathered}\mu_n(\boldsymbol{x}) = \Sigma(\boldsymbol{x}, \boldsymbol{x}_{1:n})\big(\Sigma(\boldsymbol{x}_{1:n}, \boldsymbol{x}_{1:n})\big)^{-1}\big(y_{1:n} - m(y_{1:n})\big)\\ \sigma_n^2(\boldsymbol{x}) = \Sigma(\boldsymbol{x}, \boldsymbol{x}) - \Sigma(\boldsymbol{x}, \boldsymbol{x}_{1:n})\big(\Sigma(\boldsymbol{x}_{1:n}, \boldsymbol{x}_{1:n})\big)^{-1}\Sigma(\boldsymbol{x}_{1:n}, \boldsymbol{x})\end{gathered} \tag{2.20}$$

Therefore, the acquisition function $\alpha(\boldsymbol{x} \mid f, S)$ which will be based on $\mu_n(\boldsymbol{x})$ and $\sigma_n^2(\boldsymbol{x})$ is denoted as follows:

$$\alpha(\boldsymbol{x}|f, S) = \alpha\big(\boldsymbol{x}|\mu_n(\boldsymbol{x}), \sigma_n^2(\boldsymbol{x})\big)$$

Indeed, GPR is represented by the two parameters $\mu_n(\boldsymbol{x})$ and $\sigma_n^2(\boldsymbol{x})$ but such parameters are not included in the target function $f(\boldsymbol{x})$ and so this is a NBO. Given acquisition function $\alpha(\boldsymbol{x} \mid \mu_n(\boldsymbol{x}), \sigma_n^2(\boldsymbol{x}))$ based on $\mu_n(\boldsymbol{x})$ and $\sigma_n^2(\boldsymbol{x})$, and also known sample $\boldsymbol{D}_n = \boldsymbol{x}_{1:n} = \{\boldsymbol{x}_1, \boldsymbol{x}_2,\ldots, \boldsymbol{x}_n\}$, let $\boldsymbol{x}_{n+1}$ be a maximizer of $\alpha(\boldsymbol{x} \mid \mu_n(\boldsymbol{x}), \sigma_n^2(\boldsymbol{x}))$ with regard to $\boldsymbol{x}$ and hence $\boldsymbol{x}_{n+1}$ will be updated continuously after every iteration until it reaches the entire maximizer $\boldsymbol{x}^*$ of $f(\boldsymbol{x})$.

$$\boldsymbol{x}_{n+1} = \underset{\boldsymbol{x}}{\operatorname{argmax}}\, \alpha\big(\boldsymbol{x}|\mu_n(\boldsymbol{x}), \sigma_n^2(\boldsymbol{x})\big) \tag{2.21}$$

As a result, the pseudo code of NBO with GPR is finetuned as follows:

Table 2.3. NBO algorithm with GPR.

```
Initialize randomly $\boldsymbol{D}_n = \boldsymbol{x}_{1:n} = \{\boldsymbol{x}_1, \boldsymbol{x}_2,\ldots, \boldsymbol{x}_n\}$.
Initialize $m(y_{1:n}) = \mu(y_{1:n})$.

While maximizer $\boldsymbol{x}^*$ is not reached or the number of iterations is not many enough
Update posterior mean $\mu_n(\boldsymbol{x})$ and variance $\sigma_n^2(\boldsymbol{x})$ of GPR with sample $\boldsymbol{D}_n$ as follows:
        $\mu_n(\boldsymbol{x}) = \Sigma(\boldsymbol{x}, \boldsymbol{x}_{1:n})\big(\Sigma(\boldsymbol{x}_{1:n}, \boldsymbol{x}_{1:n})\big)^{-1}\big(y_{1:n} - m(y_{1:n})\big)$
```

```
    $\sigma_n^2(\boldsymbol{x}) = \Sigma(\boldsymbol{x}, \boldsymbol{x}) - \Sigma(\boldsymbol{x}, \boldsymbol{x}_{1:n})\big(\Sigma(\boldsymbol{x}_{1:n}, \boldsymbol{x}_{1:n})\big)^{-1}\Sigma(\boldsymbol{x}_{1:n}, \boldsymbol{x})$
Determine acquisition function $\alpha(\boldsymbol{x} \mid \mu_n(\boldsymbol{x}), \sigma_n^2(\boldsymbol{x}))$ based on $\mu_n(\boldsymbol{x})$ and $\sigma_n^2(\boldsymbol{x})$ of GPR.
Find $\boldsymbol{x}_{n+1}$ as a maximizer of $\alpha(\boldsymbol{x} \mid \mu_n(\boldsymbol{x}), \sigma_n^2(\boldsymbol{x}))$ with regard to $\boldsymbol{x}$.
    $\boldsymbol{x}_{n+1} = \underset{\boldsymbol{x}}{\operatorname{argmax}}\, \alpha\big(\boldsymbol{x} \big| \mu_n(\boldsymbol{x}), \sigma_n^2(\boldsymbol{x})\big)$
Set
    $m(y_{n+1}) = \mu_n(\boldsymbol{x}_{n+1})$
Add $\boldsymbol{x}_{n+1}$ to sample $\boldsymbol{D}_n$.
    $\boldsymbol{D}_n = \boldsymbol{D}_n \cup \{\boldsymbol{x}_{n+1}\} = \boldsymbol{x}_{i:(n+1)}$
Increase $n = n + 1$.
End while
```

In general, there are two important tasks of NBO in which the first one is to determine the posterior mean $\mu_n(\boldsymbol{x})$ and variance $\sigma_n^2(\boldsymbol{x})$ of GPR. The second one is to specify the acquisition function $\alpha(\boldsymbol{x} \mid \mu_n(\boldsymbol{x}), \sigma_n^2(\boldsymbol{x}))$ which is described shortly here. Detailed description of GPR and acquisition function will be mentioned in the next sections. Note that $\alpha(\boldsymbol{x} \mid \mu_n(\boldsymbol{x}), \sigma_n^2(\boldsymbol{x}))$ is function of $\boldsymbol{x}$. Recall that BO maximizes the acquisition function $\alpha(\boldsymbol{x} \mid f, S)$ so as to search for maximizer $\boldsymbol{x}^*$ because target function $f(\boldsymbol{x})$ is assumed to be a black box for BO and BO creates previously surrogate model from which acquisition function is derived later. Acquisition function is especially important to NBO because NBO does not know $f(\boldsymbol{x})$ and parameters of NBO surrogate model are not relevant to $f(\boldsymbol{x})$. Acquisition function may not be strict with PBO but it is very strict with NBO. Moreover, finding maximizer of acquisition function should be cheaper than finding maximizer of target function $f(\boldsymbol{x})$ so that researchers in optimization domain will pay more attention to BO. There are some acquisition functions, for example, probability of improvement, expected improvement, entropy search, and upper confidence bound but expected improvement (EI) is the most popular one. EI is mentioned here and other ones are described in the next section.

Given sample $\boldsymbol{D}_n = \boldsymbol{x}_{1:n} = \{\boldsymbol{x}_1, \boldsymbol{x}_2,\ldots, \boldsymbol{x}_n\}$ and its evaluations $y_{1:n} = \{y_1, y_2,\ldots, y_n\}$, let $y_n^* = \max\{y_1, y_2,\ldots, y_n\}$ be the temporal maximum at current iteration of NBO algorithm. EI acquisition function is determined as follows:

$$\alpha_{\mathrm{EI}}(\boldsymbol{x}) = \alpha\big(\boldsymbol{x} \big| \mu_n(\boldsymbol{x}), \sigma_n^2(\boldsymbol{x})\big) = (\mu_n(\boldsymbol{x}) - y_n^*)\Phi\left(\frac{\mu_n(\boldsymbol{x}) - y_n^*}{\sigma_n(\boldsymbol{x})}\right) + \sigma_n(\boldsymbol{x})\phi(y_n^*)$$

Where $\sigma_n(\boldsymbol{x})$ is called standard deviation of y given $y_{1:n}$, $\boldsymbol{x}_{1:n}$, and $\boldsymbol{x}$.

$$\sigma_n(\boldsymbol{x}) = \sqrt{\sigma_n(\boldsymbol{x})}$$

The most important here is that $\phi(.)$ and $\Phi(.)$ are standard normal PDF and standard normal cumulative distribution function (standard normal CDF). Standard normal distribution is also called standard Gaussian distribution.

$$\phi(z) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{z^2}{2}\right)$$

$$\Phi(z_0) = \phi(z < z_0) = \phi(z \leq z_0) = \int_{-\infty}^{z_0} \phi(z)dz$$

Mean and variance of standard normal distribution are 0 and 1, respectively. Moreover, values of standard normal CDF were evaluated in practice. As a result, the maximizer $\boldsymbol{x}_{n+1}$ of EI acquisition function in NBO algorithm at a current iteration is determined as follows:

$$\boldsymbol{x}_{n+1} = \underset{\boldsymbol{x}}{\operatorname{argmax}}\, \alpha_{\mathrm{EI}}(\boldsymbol{x}) = \underset{\boldsymbol{x}}{\operatorname{argmax}}\left((\mu_n(\boldsymbol{x}) - y_n^*)\Phi\left(\frac{\mu_n(\boldsymbol{x}) - y_n^*}{\sigma_n(\boldsymbol{x})}\right) + \sigma_n(\boldsymbol{x})\phi(y_n^*)\right) \tag{2.22}$$

Recall that defining and maximizing acquisition function is one of two most important tasks in BO whereas the other one is to build up surrogate model. The essential criterion is that maximizing acquisition function like EI should be cheaper than maximizing analytic target function $f(\boldsymbol{x})$ by analytic methods. Anyhow, BO is always significant because $f(\boldsymbol{x})$ can be arbitrary or have no analytic formulation for BO.

3. Gaussian Process Regression

Nonparametric BO is based on Gaussian process regression (GPR) which, in turn, is based on Gaussian process. Therefore, we start the description of GPR with a concept of Gaussian process. Given a random process $\boldsymbol{X} = (X_1, X_2,\ldots, X_n)$ over n timepoints in which each X_t where t belongs to $\{1, 2,\ldots, n\}$ is random variable then, $\boldsymbol{X}$ is called *Gaussian random process* or Gaussian process (GP) in brief if and only if for any finite index set $\{t_1, t_2,\ldots, t_k\}$ of $\{1, 2,\ldots, n\}$ where t_j belongs to $\{1, 2,\ldots, n\}$, the subset $\left(X_{t_1}, X_{t_2}, \ldots, X_{t_k}\right)$ considered as t_k-dimension random vector $\left(X_{t_1}, X_{t_2}, \ldots, X_{t_k}\right)^T$ follows multinormal distribution known as multivariate Gaussian distribution. Note, each X_{t_1} represents one dimension of the k-dimension random vector variable $\left(X_{t_1}, X_{t_2}, \ldots, X_{t_k}\right)^T$. Moreover, please pay attention that any combination of $X_{t_1}, X_{t_2}, \ldots, X_{t_k}$ follows multinormal distribution too. Without loss of generality, we denote the random process as random variable $\boldsymbol{X} = (X_1, X_2,\ldots, X_n)^T$ where $\{t_1, t_2,\ldots, t_k\} = \{1, 2,\ldots, n\}$ obeying multinormal distribution also called multivariate normal distribution or multivariate Gaussian distribution as follows:

$$\boldsymbol{X} = (X_1, X_2, \ldots, X_n)^T \sim \mathcal{N}\big(\boldsymbol{X} \big| \mu(\boldsymbol{X}), \Sigma(\boldsymbol{X})\big) \tag{3.1}$$

Where $\mu(\boldsymbol{X})$ and $\Sigma(\boldsymbol{X})$ are mean function and covariance function of $\boldsymbol{X}$, respectively. Note, the superscript "^{T}" denotes transposition operator of vector and matrix whereas the tilde sign "~" indicates probabilistic distribution of a random variable. GP is known as infinite multinormal distribution which is the generalized form of n-dimension multinormal distribution because n can approach positive infinity $n = +\infty$. Let $f(\boldsymbol{X} \mid \mu(\boldsymbol{X}), \Sigma(\boldsymbol{X})$ be probability density function (PDF) of $\boldsymbol{X}$ when $\boldsymbol{X}$ is continuous, we have:

$$f\big(\boldsymbol{X} \big| \mu(\boldsymbol{X}), \Sigma(\boldsymbol{X})\big) = \mathcal{N}\big(\boldsymbol{X} \big| \mu(\boldsymbol{X}), \Sigma(\boldsymbol{X})\big)$$

Where mean $\mu(\boldsymbol{X})$ and covariance $\Sigma(\boldsymbol{X})$ are functions of $\boldsymbol{X}$, respectively. Note, in this research, notation $f(.\mid.)$ often denotes PDF, which is not concrete function, and notation $\mathcal{N}(.\,|.)$ denotes normal distribution. Probability function, distribution function, cumulative distribution function (CDF), and PDF are terms which can be exchangeable in some cases, but the term "PDF" focuses on probability function of continuous variable whereas the general notation $P(.)$ denotes probability function in both discrete case and continuous case.

In literature, $\mu(\boldsymbol{X})$ is assumed to be zero as $\mu(\boldsymbol{X}) = \boldsymbol{0}^T$ for convenience but it can be defined as the random process $\boldsymbol{X}$ itself, $\mu(\boldsymbol{X}) = \boldsymbol{X}$. Besides, $\mu(\boldsymbol{X})$ can be customized according to concrete applications, for example it can be constant as $\mu(\boldsymbol{X}) = \boldsymbol{\mu}$. However, it is better to set $\mu(\boldsymbol{X}) = \boldsymbol{0}^T$ for my opinion. As a convention in Gaussian process, output of a mean function is a mean and so $\mu(\boldsymbol{X})$ also denotes the theoretical mean of $\boldsymbol{X}$. In general case, $\mu(\boldsymbol{X})$ is vector function of combination of X_1, X_3,…, and X_n (s) belong to $\boldsymbol{X}$ but, as a convention, each $\mu(X_i)$ depends only X_i and moreover, $\mu(X_i)$ has the same formulation of every X_i. Therefore, $\mu(\boldsymbol{X})$ should be identified with $\mu(X_i)$ or $\mu(X)$ where X denotes any X_i belonging to $\boldsymbol{X}$.

$$\mu(\boldsymbol{X}) \text{ should be } \begin{pmatrix} \mu(X_1) \\ \mu(X_2) \\ \vdots \\ \mu(X_n) \end{pmatrix}$$

Covariance function $\Sigma(\boldsymbol{X})$ measures the correlation between random variables when the random process "moves" them, in which the closer the given two random variables are, the larger their covariance. The two most important properties based on covariance function $\Sigma(\boldsymbol{X})$ of random process are stationarity and isotropy among four basic properties: stationarity, isotropy, smoothness, and periodicity. Stationarity implies that the PDF $f(\boldsymbol{X} \mid \mu(\boldsymbol{X})$ of random process $\boldsymbol{X}$ will not be changed when the process is moved in time, for example, if new random variable X_{n+1} raises to be added then means and covariances of old (previous) variables X_i where $1 \leq i \leq n$ in $\boldsymbol{X}$ will not be changed. It is proved that if GP $\boldsymbol{X}$ satisfies stationarity, $\Sigma(\boldsymbol{X})$ will depend only on the deviation $X_i–X_j$ but the inversed statement is not asserted. However, if $\Sigma(\boldsymbol{X})$ depends only on the Euclidean distance $|X_i–X_j|$ then, GP $\boldsymbol{X}$ will satisfy isotropy. If $\boldsymbol{X}$ satisfies both stationarity and isotropy, $\boldsymbol{X}$ is called homogeneous process. In cases where each element of matrix function $\Sigma(\boldsymbol{X})$ depends on only X_i and X_j like stationarity case and isotropy, it will result out a following matrix.

$$\Sigma(\boldsymbol{X}) = \Sigma(\boldsymbol{X}, \boldsymbol{X}) = \begin{pmatrix} \Sigma(X_1, X_1) & \Sigma(X_1, X_2) & \cdots & \Sigma(X_1, X_n) \\ \Sigma(X_2, X_1) & \Sigma(X_2, X_2) & \cdots & \Sigma(X_2, X_n) \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma(X_n, X_1) & \Sigma(X_n, X_2) & \cdots & \Sigma(X_n, X_n) \end{pmatrix} \quad (3.2)$$

In these cases, covariance function $\Sigma(\boldsymbol{X})$ can be "identified" with its element function $\Sigma(X_i, X_j)$ when the formulation of $\Sigma(X_i, X_j)$ is not changed formally and hence, $\Sigma(X_i, X_j)$ is called *kernel function*. Please pay attention that the term "covariance function" is slightly different from the term "kernel function". When referring to only two variables X_i and X_j, they are the same. When referring to one or two sets of variables such as $\boldsymbol{X}$ and $\boldsymbol{X}^*$, the notations $\Sigma(\boldsymbol{X}, \boldsymbol{X}^*)$, $\Sigma(\boldsymbol{X}, \boldsymbol{X})$, or $\Sigma(\boldsymbol{X})$ mentions covariance function as matrix function whose elements are defined by kernel function if each element depends on only two variables like X_i and X_j. Exactly $\Sigma(.)$ is matrix-by-vector function if each X_i is vector. Output of a covariance function is a covariance matrix if such output is invertible and symmetric; in this case, covariance function is "identified" with covariance matrix. For explanation, suppose each X_i belonging to $\boldsymbol{X}$ is scalar (so that $\boldsymbol{X}$ is vector) and the output of covariance function $\Sigma(\boldsymbol{X})$ is covariance matrix, the multinormal PDF of $\boldsymbol{X}$ if formulated as follows:

$$\begin{aligned} f\left(\boldsymbol{X} \middle| \mu(\boldsymbol{X}), \Sigma(\boldsymbol{X})\right) &= \mathcal{N}\left(\boldsymbol{X} \middle| \mu(\boldsymbol{X}), \Sigma(\boldsymbol{X})\right) \\ &= (2\pi)^{-\frac{n}{2}} |\Sigma(\boldsymbol{X})|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\left(\boldsymbol{X} - \mu(\boldsymbol{X})\right)^T \Sigma(\boldsymbol{X})^{-1} \left(\boldsymbol{X} - \mu(\boldsymbol{X})\right)\right) \end{aligned}$$

However, each X_i in GPR is arbitrary, such as scalar, vector, and matrix. Note, $\Sigma(\boldsymbol{X})^{-1}$ denotes inverse of covariance matrix $\Sigma(\boldsymbol{X})$. Kernel function is not only essential to define covariance function of Gaussian process but also used to make GPR line smoother. The following are some kernel functions.

$$\text{Linear: } \Sigma_L(X_i, X_j) = X_i^T X_j$$

$$\text{Squared exponential: } \Sigma_{SE}(X_i, X_j) = \exp\left(-\frac{|X_i - X_j|^2}{2l^2}\right)$$

Where l is the characteristic length-scale of the process which reinforces similarity of X_i and X_j. In this research the notation $|.|$ denotes absolute value of scalar, length (module) of vector, determinant of matrix, and cardinality of set. As a convention, let $X_{i:j}$ where $i \leq j$ denote a sequential subset of $\boldsymbol{X}$ such that $X_{i:j} = \{X_i, X_{i+1}, \ldots, X_j\}$. Of course, we have $X_{i:i} = X_i$. In general case, $X_{i:j}$ is arbitrary subset of distinguish variables. Let $\Sigma(X_i, X_{j:k})$ denote a row-vector covariance function of X_i and $X_{j:k}$ as follows:

$$\Sigma(X_i, X_{j:k}) = \left(\Sigma(X_i, X_j), \Sigma(X_i, X_{j+1}), \ldots, \Sigma(X_i, X_k)\right) = \begin{pmatrix} \Sigma(X_i, X_j) \\ \Sigma(X_i, X_{j+1}) \\ \vdots \\ \Sigma(X_i, X_k) \end{pmatrix}^T$$

Let $\Sigma(X_{j:k}, X_i)$ denote a column-vector covariance function of $X_{j:k}$ and X_i as follows:

$$\Sigma(X_{j:k}, X_i) = \begin{pmatrix} \Sigma(X_j, X_i) \\ \Sigma(X_{j+1}, X_i) \\ \vdots \\ \Sigma(X_k, X_i) \end{pmatrix}$$

As a convention, let $\Sigma(X_{i:j}, X_{k:l})$ denote a covariance function of $X_{i:j}$ and $X_{k:l}$ as follows:

$$\Sigma\left(X_{i:j}, X_{k:l}\right) = \begin{pmatrix} \Sigma(X_i, X_k) & \Sigma(X_i, X_{k+1}) & \cdots & \Sigma(X_i, X_l) \\ \Sigma(X_{i+1}, X_k) & \Sigma(X_{i+1}, X_{k+1}) & \cdots & \Sigma(X_{i+1}, X_l) \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma\left(X_j, X_k\right) & \Sigma\left(X_j, X_{k+1}\right) & \cdots & \Sigma\left(X_j, X_l\right) \end{pmatrix} \quad (3.3)$$

Obviously, the output of $\Sigma(X_{i:j}, X_{k:l})$ cannot be a covariance matrix if it is not squared. Note, $\Sigma(X_{i:j}, X_{k:l})$ is a partition of $\Sigma(\boldsymbol{X})$ and we have $\Sigma(\boldsymbol{X}) = \Sigma(X_{i:n}, X_{i:n})$. If denoting $\boldsymbol{X}_1 = (X_i, X_{i+1},\ldots, X_j)^T$ and $\boldsymbol{X}_2 = (X_k, X_{k+1},\ldots, X_l)^T$, we denote:

$$\begin{gathered} \Sigma(\boldsymbol{X}_1, \boldsymbol{X}_2) = \Sigma\left(X_{i:j}, X_{k:l}\right) \\ \Sigma(\boldsymbol{X}_1, \boldsymbol{X}_1) = \Sigma(\boldsymbol{X}_1) = \Sigma\left(X_{i:j}, X_{i:j}\right) = \Sigma\left(X_{i:j}\right) \end{gathered} \quad (3.4)$$

Gaussian process repression (GPR) is based on Gaussian process (GP) when there is a target function which attaches to each X where X represents any X_i in $\boldsymbol{X}$ such that $Y = f(X)$. Please distinguish the target function f from the formal notations $f(.|.)$ of probability density function (PDF). Of course, Y or Y_i is also random variable and we also have $Y_i = f(X_i)$. Besides, in context of regression model, the target function $f(X)$ is not a formal function with arithmetic operators and exactly, it is a mapping between X and Y. For example, sample $\{\boldsymbol{X}, \boldsymbol{Y}\}$ has two paired datasets $\boldsymbol{X}$ and $\boldsymbol{Y}$ in which for every X_i belonging to $\boldsymbol{X}$ there is a Y_i belong to $\boldsymbol{Y}$ and hence, the equation $Y = f(X)$ only indicates such mapping. GPR model tries to represent or draw a regressive PDF of $\boldsymbol{Y}$ from its previous ones (which will be explained later) and $\boldsymbol{X}$. Suppose we had a GP $\boldsymbol{X} = (X_1, X_2,\ldots, X_n)^T$ and target function $\boldsymbol{Y} = f(\boldsymbol{X})$, assuming that the prior PDF of $\boldsymbol{Y} = (Y_1, Y_2,\ldots, Y_n)^T = f(\boldsymbol{X})$ given $\boldsymbol{X}$ is also derived from the multinormal PDF of $\boldsymbol{X}$.

$$\boldsymbol{Y}|\boldsymbol{X} \leftarrow f\left(\boldsymbol{X}\middle|\mu(\boldsymbol{X}), \Sigma(\boldsymbol{X})\right) = \mathcal{N}\left(\boldsymbol{X}\middle|\mu(\boldsymbol{X}), \Sigma(\boldsymbol{X})\right)$$

Therefore, we have:

$$\boldsymbol{Y}|\boldsymbol{X} \sim f\left(\boldsymbol{Y}\middle|\boldsymbol{X}, \mu(.), \Sigma(.)\right) = \mathcal{N}\left(\boldsymbol{Y}\middle|\mu(\boldsymbol{Y}), \Sigma(\boldsymbol{X}, \boldsymbol{X})\right) \quad (3.5)$$

Where $\mu(.)$ and $\Sigma(.)$ denote mean function and covariance function in formality, respectively, and $\Sigma(\boldsymbol{X}, \boldsymbol{X}) = \Sigma(\boldsymbol{X})$. The equation above implies that the prior PDF of GP $\boldsymbol{Y}$ is initialized with the PDF of GP $\boldsymbol{X}$. Mean vector and covariance matrix of $\boldsymbol{Y}$ are $\mu(\boldsymbol{Y})$ and $\Sigma(\boldsymbol{X}, \boldsymbol{X})$ *) within the PDF $f(\boldsymbol{Y} \mid \boldsymbol{X}, \mu(.), \Sigma(.))$, respectively. The mean function $\mu(\boldsymbol{Y})$ is redefined here with variable $\boldsymbol{Y}$ but it should be the same to the mean function $\mu(\boldsymbol{X})$ in formulation if $\mu(\boldsymbol{X})$ can be extended with the lower/higher dimensional space. For instance, if $\mu(\boldsymbol{X})$ is defined as itself $\mu(\boldsymbol{X}) = \boldsymbol{X}$ then, $\mu(\boldsymbol{Y})$ will be defined as itself $\mu(\boldsymbol{Y}) = \boldsymbol{Y} = (Y_1, Y_2,\ldots, Y_n)^T$. In literature, for simplicity, $\mu(\boldsymbol{Y})$ is set to be zero as $\mu(\boldsymbol{Y}) = \boldsymbol{0}^T$. It is also better to set $\mu(\boldsymbol{Y}) = \boldsymbol{0}^T$.

Now suppose we randomize a new set of variables $\boldsymbol{X}^* = (X_{n+1}, X_{n+2},\ldots, X_{n+k})^T$ and obtain their evaluated values $\boldsymbol{Y}^* = f(\boldsymbol{X}^*) = (Y_{n+1}, Y_{n+2},\ldots, Y_{n+k})^T$. Similarly, mean vector and covariance matrix of $\boldsymbol{Y}^*$ are $\mu(\boldsymbol{Y}^*)$ and $\Sigma(\boldsymbol{X}^*, \boldsymbol{X}^*)$ within the PDF $f(\boldsymbol{Y}^* \mid \boldsymbol{X}^*, \mu(.), \Sigma(.))$, respectively. Of course, the mean function $\mu(.)$ for both $\boldsymbol{Y}$ and $\boldsymbol{Y}^*$ is the same. Consequently, the joint PDF of $\boldsymbol{Y}$ and $\boldsymbol{Y}^*$ which is of course multinormal distribution is denoted as follows (Wang, 2022, p. 11):

$$f\left(\boldsymbol{Y}, \boldsymbol{Y}^*\middle|\boldsymbol{X}, \boldsymbol{X}^*, \mu(.), \Sigma(.)\right) = \mathcal{N}\left(\boldsymbol{Y}\middle|\begin{pmatrix}\mu(\boldsymbol{Y})\\ \mu(\boldsymbol{Y}^*)\end{pmatrix}, \begin{pmatrix}\Sigma(\boldsymbol{X}, \boldsymbol{X}) & \Sigma(\boldsymbol{X}, \boldsymbol{X}^*)\\ \Sigma(\boldsymbol{X}^*, \boldsymbol{X}) & \Sigma(\boldsymbol{X}^*, \boldsymbol{X}^*)\end{pmatrix}\right) \quad (3.6)$$

Where mean vector and covariance matrix of the joint PDF of $\boldsymbol{Y}$ and $\boldsymbol{Y}^*$ are $\begin{pmatrix}\mu(\boldsymbol{Y})\\ \mu(\boldsymbol{Y}^*)\end{pmatrix}$ and $\begin{pmatrix}\Sigma(\boldsymbol{X}, \boldsymbol{X}) & \Sigma(\boldsymbol{X}, \boldsymbol{X}^*)\\ \Sigma(\boldsymbol{X}^*, \boldsymbol{X}) & \Sigma(\boldsymbol{X}^*, \boldsymbol{X}^*)\end{pmatrix}$, respectively. The equation above is one of interesting properties of multinormal distribution. As a result, Gaussian process regression (GPR) model of dependent variable Y is conditional PDF of $\boldsymbol{Y}^*$ given $\boldsymbol{Y}$, $\boldsymbol{X}$, and $\boldsymbol{X}^*$ such as $f(\boldsymbol{Y}^* \mid \boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{X}^*, \mu(.), \Sigma(.))$ which is also predictive PDF of $\boldsymbol{Y}^*$ because it can be used to predict next occurrence of Y. From interesting properties of multinormal distribution, it is easy to drawn GPR from the joint PDF $f(\boldsymbol{Y}, \boldsymbol{Y}^* \mid \boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{X}^*, \mu(.), \Sigma(.))$ as follows:

$$\boldsymbol{Y}^*|\boldsymbol{Y},\boldsymbol{X},\boldsymbol{X}^* \sim f\big(\boldsymbol{Y}^*\big|\boldsymbol{Y},\boldsymbol{X},\boldsymbol{X}^*,\mu(.),\Sigma(.)\big) = \mathcal{N}\big(\boldsymbol{Y}\big|\mu(\boldsymbol{Y}^*|\boldsymbol{Y},\boldsymbol{X},\boldsymbol{X}^*),\Sigma(\boldsymbol{Y}^*|\boldsymbol{X},\boldsymbol{X}^*)\big) \tag{3.7}$$

Where mean vector $\mu(\boldsymbol{Y}^* \mid \boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{X}^*)$ and covariance matrix $\Sigma(\boldsymbol{Y}^* \mid \boldsymbol{X}, \boldsymbol{X}^*)$ are:

$$(\boldsymbol{Y}^*|\boldsymbol{Y},\boldsymbol{X},\boldsymbol{X}^*) = \mu(\boldsymbol{Y}^*) + \Sigma(\boldsymbol{X}^*,\boldsymbol{X})\big(\Sigma(\boldsymbol{X},\boldsymbol{X})\big)^{-1}\big(\boldsymbol{Y} - \mu(\boldsymbol{Y})\big)$$

$$\Sigma(\boldsymbol{Y}^*|\boldsymbol{X},\boldsymbol{X}^*) = \Sigma(\boldsymbol{X}^*,\boldsymbol{X}^*) - \Sigma(\boldsymbol{X}^*,\boldsymbol{X})\big(\Sigma(\boldsymbol{X},\boldsymbol{X})\big)^{-1}\Sigma(\boldsymbol{X},\boldsymbol{X}^*)$$

When $\boldsymbol{Y}^*$ is variable then, $\mu(\boldsymbol{Y}^*)$ is still mean function but $\mu(\boldsymbol{Y})$ was determined and hence, for making clear this vagueness, let $m(\boldsymbol{Y})$ replace $\mu(\boldsymbol{Y})$ so that mean vector $\mu(\boldsymbol{Y}^* \mid \boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{X}^*)$ and covariance matrix $\Sigma(\boldsymbol{Y}^* \mid \boldsymbol{X}, \boldsymbol{X}^*)$ are rewritten as follows:

$$\begin{gathered}\mu(\boldsymbol{Y}^*|\boldsymbol{Y},\boldsymbol{X},\boldsymbol{X}^*) = \mu(\boldsymbol{Y}^*) + \Sigma(\boldsymbol{X}^*,\boldsymbol{X})\big(\Sigma(\boldsymbol{X},\boldsymbol{X})\big)^{-1}\big(\boldsymbol{Y} - m(\boldsymbol{Y})\big)\\ \Sigma(\boldsymbol{Y}^*|\boldsymbol{X},\boldsymbol{X}^*) = \Sigma(\boldsymbol{X}^*,\boldsymbol{X}^*) - \Sigma(\boldsymbol{X}^*,\boldsymbol{X})\big(\Sigma(\boldsymbol{X},\boldsymbol{X})\big)^{-1}\Sigma(\boldsymbol{X},\boldsymbol{X}^*)\end{gathered} \tag{3.8}$$

The quantity $m(\boldsymbol{Y})$ is a realistic mean. At the beginning of any GPR algorithm, it is defined as the same to the mean function $\mu(.)$, for example, $m(\boldsymbol{Y}) = \mu(\boldsymbol{Y}) = \boldsymbol{0}^T$, $m(\boldsymbol{Y}) = \mu(\boldsymbol{Y}) = \boldsymbol{Y}$, etc., but, at the later phase of any GPR algorithm, it is recurrently determined by previously known $\mu(\boldsymbol{Y} \mid Y_{1:n0}, X_{1:n0}, \boldsymbol{X})$ as follows:

$$m(\boldsymbol{Y}) = \mu\big(\boldsymbol{Y}\big|Y_{1:n_0},X_{1:n_0},\boldsymbol{X}\big) = \begin{pmatrix}\mu\big(\boldsymbol{Y}\big|Y_{1:n_0},X_{1:n_0},X_1\big)\\ \mu\big(\boldsymbol{Y}\big|Y_{1:n_0},X_{1:n_0},X_2\big)\\ \vdots\\ \mu\big(\boldsymbol{Y}\big|Y_{1:n_0},X_{1:n_0},X_n\big)\end{pmatrix} \tag{3.9}$$

Obviously, the GPR of $\boldsymbol{Y}^*$ distributes normally with mean vector $\mu(\boldsymbol{Y}^* \mid \boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{X}^*)$ and covariance matrix $\Sigma(\boldsymbol{Y}^* \mid \boldsymbol{X}, \boldsymbol{X}^*)$. Indeed, the GPR $f(\boldsymbol{Y}^* \mid \boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{X}^*, \mu(.), \Sigma(.))$ is posterior PDF of $\boldsymbol{Y}$ whose prior PDF is $f(\boldsymbol{Y} \mid \boldsymbol{X}, \mu(.), \Sigma(.))$. Initializing such prior PDF by the PDF of $\boldsymbol{X}$ as $f(\boldsymbol{Y} \mid \boldsymbol{X}, \mu(.), \Sigma(.)) = f(\boldsymbol{X} \mid \mu(.), \Sigma(.))$ is not totally correct but implicit biases will be decreased after the posterior PDF $f(\boldsymbol{Y}^* \mid \boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{X}^*, \mu(.), \Sigma(.))$ is updated. We have following summary:

$$\begin{gathered}\text{Prior: } \boldsymbol{Y}|\boldsymbol{X} \sim \mathcal{N}\big(\boldsymbol{Y}\big|\mu(\boldsymbol{X}),\Sigma(\boldsymbol{X})\big)\\ \text{Posterior (GPR): } \boldsymbol{Y}^*|\boldsymbol{Y},\boldsymbol{X},\boldsymbol{X}^* \sim \mathcal{N}\big(\boldsymbol{Y}\big|\mu(\boldsymbol{Y}^*|\boldsymbol{Y},\boldsymbol{X},\boldsymbol{X}^*),\Sigma(\boldsymbol{Y}^*|\boldsymbol{X},\boldsymbol{X}^*)\big)\end{gathered} \tag{3.10}$$

Although the GPR of $\boldsymbol{Y}^*$ depends on $\boldsymbol{Y}$, $\boldsymbol{X}$, and $\boldsymbol{X}^*$, the main semantic meaning of regression model here is mentioned mainly that $\boldsymbol{Y}^*$ is determined based on its previous one $\boldsymbol{Y}$ when both of them are assumed to be based on X via the PDF of $\boldsymbol{Y}$ as $f(\boldsymbol{Y} \mid \boldsymbol{X}, \mu(.), \Sigma(.))$ and the PDF of $\boldsymbol{Y}^*$ as $f(\boldsymbol{Y}^* \mid \boldsymbol{X}^*, \mu(.), \Sigma(.))$. This implies that X is the intermediate point for the probabilistic dependence of $\boldsymbol{Y}^*$ on $\boldsymbol{Y}$.

If $\boldsymbol{X}^*$ and $\boldsymbol{Y}^*$ are 1-element vectors such that $\boldsymbol{X}^* = X_{n+1}$ and $\boldsymbol{Y}^* = Y_{n+1}$ and let $\boldsymbol{x}_{1:n} = \boldsymbol{X}$, $\boldsymbol{x} = \boldsymbol{X}^*$, $y_{1:n} = \boldsymbol{Y}$, and $y = \boldsymbol{Y}^*$, the GPR of $\boldsymbol{Y}^*$ which is now GPR of y becomes:

$$y|y_{1:n},\boldsymbol{x}_{1:n},\boldsymbol{x} \sim \mathcal{N}\big(y\big|\mu(y|y_{1:n},\boldsymbol{x}_{1:n},\boldsymbol{x}),\sigma^2(y|\boldsymbol{x}_{1:n},\boldsymbol{x})\big) \tag{3.11}$$

Where $\mu(y \mid y_{1:n}, \boldsymbol{x}_{1:n}, \boldsymbol{x})$ and $\sigma^2(y \mid \boldsymbol{x}_{1:n}, \boldsymbol{x})$ are mean and variance of the posterior PDF of y given $y_{1:n}$, $\boldsymbol{x}_{1:n}$, and $\boldsymbol{x}$.

$$\begin{gathered}\mu(y|y_{1:n},\boldsymbol{x}_{1:n},\boldsymbol{x}) = \mu(y) + \Sigma(\boldsymbol{x},\boldsymbol{x}_{1:n})\big(\Sigma(\boldsymbol{x}_{1:n},\boldsymbol{x}_{1:n})\big)^{-1}\big(y_{1:n} - m(y_{1:n})\big)\\ \sigma^2(y|\boldsymbol{x}_{1:n},\boldsymbol{x}) = \Sigma(\boldsymbol{x},\boldsymbol{x}) - \Sigma(\boldsymbol{x},\boldsymbol{x}_{1:n})\big(\Sigma(\boldsymbol{x}_{1:n},\boldsymbol{x}_{1:n})\big)^{-1}\Sigma(\boldsymbol{x}_{1:n},\boldsymbol{x})\end{gathered} \tag{3.12}$$

The equation above with single variables $\boldsymbol{x}$ and y (single posterior processes) is popular in BO, especially $\boldsymbol{x}$ is vector and y is scalar although $\boldsymbol{x}$ and y can be arbitrary such as scalar, vector, and matrix with note that high dimensional spaces require tensor products for example. Please pay attention that $m(y_{1:n})$ in GPR algorithm is the realistic mean which is recurrently determined by previously known $\mu(y \mid y_{1:n-1}, \boldsymbol{x}_{1:n-1}, \boldsymbol{x})$:

$$m(y_n) = \mu\big(y\big|y_{1:n_0+n-1},\boldsymbol{x}_{1:n_0+n-1},\boldsymbol{x}_n\big) \tag{3.13}$$

Note,

$$\Sigma(\boldsymbol{x}_{1:n},\boldsymbol{x}_{1:n}) = \begin{pmatrix} \Sigma(\boldsymbol{x}_1,\boldsymbol{x}_1) & \Sigma(\boldsymbol{x}_1,\boldsymbol{x}_2) & \cdots & \Sigma(\boldsymbol{x}_1,\boldsymbol{x}_n) \\ \Sigma(\boldsymbol{x}_2,\boldsymbol{x}_1) & \Sigma(\boldsymbol{x}_2,\boldsymbol{x}_2) & \cdots & \Sigma(\boldsymbol{x}_2,\boldsymbol{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma(\boldsymbol{x}_n,\boldsymbol{x}_1) & \Sigma(\boldsymbol{x}_n,\boldsymbol{x}_2) & \cdots & \Sigma(\boldsymbol{x}_n,\boldsymbol{x}_n) \end{pmatrix}$$

$$\Sigma(\boldsymbol{x},\boldsymbol{x}_{1:n}) = \begin{pmatrix} \Sigma(\boldsymbol{x},\boldsymbol{x}_1) \\ \Sigma(\boldsymbol{x},\boldsymbol{x}_2) \\ \vdots \\ \Sigma(\boldsymbol{x},\boldsymbol{x}_n) \end{pmatrix}^T, \Sigma(\boldsymbol{x}_{1:n},\boldsymbol{x}) = \begin{pmatrix} \Sigma(\boldsymbol{x}_1,\boldsymbol{x}) \\ \Sigma(\boldsymbol{x}_2,\boldsymbol{x}) \\ \vdots \\ \Sigma(\boldsymbol{x}_n,\boldsymbol{x}) \end{pmatrix}$$

Suppose only $\boldsymbol{x}$ is considered variable whereas y, $y_{1:n}$, are $\boldsymbol{x}_{1:n}$ are known then, mean function $\mu(.)$ is set to be zero as $\mu(y) = 0$ so that both the mean $\mu(y \mid y_{1:n}, \boldsymbol{x}_{1:n}, \boldsymbol{x})$ and the variance $\sigma^2(y \mid \boldsymbol{x}_{1:n}, \boldsymbol{x})$ are functions of only $\boldsymbol{x}$ as follows:

$$\mu_n(\boldsymbol{x}) = \mu(y|y_{1:n},\boldsymbol{x}_{1:n},\boldsymbol{x}) = \Sigma(\boldsymbol{x},\boldsymbol{x}_{1:n})\big(\Sigma(\boldsymbol{x}_{1:n},\boldsymbol{x}_{1:n})\big)^{-1}\big(y_{1:n} - m(y_{1:n})\big)$$

$$\sigma_n^2(\boldsymbol{x}) = \sigma^2(y|\boldsymbol{x}_{1:n},\boldsymbol{x}) = \Sigma(\boldsymbol{x},\boldsymbol{x}) - \Sigma(\boldsymbol{x},\boldsymbol{x}_{1:n})\big(\Sigma(\boldsymbol{x}_{1:n},\boldsymbol{x}_{1:n})\big)^{-1}\Sigma(\boldsymbol{x}_{1:n},\boldsymbol{x})$$

Recall that $m(y_{1:n})$ is the realistic mean which is recurrently determined by previously known $\mu_{n-1}(\boldsymbol{x})$:

$$m(y_n) = \mu_{n-1}(\boldsymbol{x}_n)$$

The event that $\mu_n(\boldsymbol{x})$ and $\sigma_n^2(\boldsymbol{x})$ are functions of only $\boldsymbol{x}$ is necessary to define acquisition function for optimization task in BO.

GPR can be executed continuously with new $\boldsymbol{X}^*$ while the old $\boldsymbol{X}^*$ is incorporated into larger GP $\boldsymbol{X}$. In practice, such continuous execution is implemented as iterative algorithm whose each iteration has two following steps:

Table 3.1. GPR algorithm.

```
Step 1:
Take new $\boldsymbol{X}^*$
$\mu(\boldsymbol{Y}^*|\boldsymbol{Y},\boldsymbol{X},\boldsymbol{X}^*) = \mu(\boldsymbol{Y}^*) + \Sigma(\boldsymbol{X}^*,\boldsymbol{X})\big(\Sigma(\boldsymbol{X},\boldsymbol{X})\big)^{-1}\big(\boldsymbol{Y} - m(\boldsymbol{Y})\big)$
$\Sigma(\boldsymbol{Y}^*|\boldsymbol{X},\boldsymbol{X}^*) = \Sigma(\boldsymbol{X}^*,\boldsymbol{X}^*) - \Sigma(\boldsymbol{X}^*,\boldsymbol{X})\big(\Sigma(\boldsymbol{X},\boldsymbol{X})\big)^{-1}\Sigma(\boldsymbol{X},\boldsymbol{X}^*)$
Step 2:
$m(\boldsymbol{Y}) = \mu(\boldsymbol{Y}^*|\boldsymbol{Y},\boldsymbol{X},\boldsymbol{X}^*)$
$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{X} \\ \boldsymbol{X}^* \end{pmatrix}$
```

In step 1, the covariance matrix $\Sigma(X, X)$ is not recomputed entirely because some of its elements $\Sigma(X_i, X_j)$ were determined before.

Because GP $\boldsymbol{X}$ will get huge when the iterative algorithm repeats many iterations and $\boldsymbol{X}^*$ is incorporated into $\boldsymbol{X}$ over and over again, it is possible to apply first-order Markov property so that each iteration remembers only one previous X. Therefore, we can assign $\boldsymbol{X}^*$ to $\boldsymbol{X}$ as $\boldsymbol{X} = \boldsymbol{X}^*$ in step 2 so that the iterative algorithm runs faster and saves more computational resources as follows:

Table 3.2. GPR algorithm with first-order Markov property.

Step 1:
Take new $\boldsymbol{X}^*$
$\mu(\boldsymbol{Y}^*\vert\boldsymbol{Y},\boldsymbol{X},\boldsymbol{X}^*) = \mu(\boldsymbol{Y}^*) + \Sigma(\boldsymbol{X}^*,\boldsymbol{X})\big(\Sigma(\boldsymbol{X},\boldsymbol{X})\big)^{-1}\big(\boldsymbol{Y} - m(\boldsymbol{Y})\big)$
$\Sigma(\boldsymbol{Y}^*\vert\boldsymbol{X},\boldsymbol{X}^*) = \Sigma(\boldsymbol{X}^*,\boldsymbol{X}^*) - \Sigma(\boldsymbol{X}^*,\boldsymbol{X})\big(\Sigma(\boldsymbol{X},\boldsymbol{X})\big)^{-1}\Sigma(\boldsymbol{X},\boldsymbol{X}^*)$
Step 2:
$m(\boldsymbol{Y}) = \mu(\boldsymbol{Y}^*\vert\boldsymbol{Y},\boldsymbol{X},\boldsymbol{X}^*)$
$\boldsymbol{X} = \boldsymbol{X}^*$

However, this advantage of first-order Markov property cannot be applied into BO (by defining how to take new $\boldsymbol{X}^*$ with acquisition functions) because it is necessary to look entire X domain forward and backward to find out potential global optimizer. The use of Markov property will be appropriate to tasks of prediction and estimation in which entire time series are split into many smaller time windows where it is necessary to remember w windows with w-order Markov property. Shortly, the assignment $\boldsymbol{X} = \boldsymbol{X}^*$ is suitable to forward modeling.

GPR focuses mainly on the regressive (posterior) covariance matrix $\Sigma(\boldsymbol{Y}^* \mid \boldsymbol{X}, \boldsymbol{X}^*)$ via kernel function $\Sigma(X_i, X_j)$ because adjusting the regressive (posterior) mean vector $\mu(\boldsymbol{Y}^* \mid \boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{X}^*)$ via adjusting the mean function $\mu(\boldsymbol{Y})$ is unnecessary due to:

$$\mu(\boldsymbol{Y}^*|\boldsymbol{Y},\boldsymbol{X},\boldsymbol{X}^*) = \mu(\boldsymbol{Y}^*) + \Sigma(\boldsymbol{X}^*,\boldsymbol{X})\big(\Sigma(\boldsymbol{X},\boldsymbol{X})\big)^{-1}\big(\boldsymbol{Y} - m(\boldsymbol{Y})\big)$$

As usual, mean function $\mu(\boldsymbol{Y}^*)$ are set to be zero as $\mu(\boldsymbol{Y}^*) = \boldsymbol{0}^T$ so that

$$\mu(\boldsymbol{Y}^*|\boldsymbol{Y},\boldsymbol{X},\boldsymbol{X}^*) = \Sigma(\boldsymbol{X}^*,\boldsymbol{X})\big(\Sigma(\boldsymbol{X},\boldsymbol{X})\big)^{-1}\big(\boldsymbol{Y} - m(\boldsymbol{Y})\big)$$

Which is indeed an arithmetic regression of $\boldsymbol{Y}^*$ on $\boldsymbol{Y}$, $\boldsymbol{X}$, and $\boldsymbol{X}^*$ in addition to the main semantic meaning that the posterior PDF of $\boldsymbol{Y}^*$ given $\boldsymbol{Y}$ is the main regression. However, it is still better to define exactly the mean function $\mu(\boldsymbol{Y})$ in cases of predicting more precisely confident intervals of Y based on X because the equation above:

$$\mu(\boldsymbol{Y}^*|\boldsymbol{Y},\boldsymbol{X},\boldsymbol{X}^*) = \Sigma(\boldsymbol{X}^*,\boldsymbol{X})\big(\Sigma(\boldsymbol{X},\boldsymbol{X})\big)^{-1}\big(\boldsymbol{Y} - m(\boldsymbol{Y})\big)$$

Which implies variation of Y according to variation of X from the origin. It is not a real value of Y. By another way, if setting $\mu(\boldsymbol{Y}^*) = \boldsymbol{Y}^*$ (also deriving $\mu(\boldsymbol{Y}) = \boldsymbol{Y}$) so that

$$\mu(\boldsymbol{Y}^*|\boldsymbol{Y},\boldsymbol{X},\boldsymbol{X}^*) = \boldsymbol{Y}^* + \Sigma(\boldsymbol{X}^*,\boldsymbol{X})\big(\Sigma(\boldsymbol{X},\boldsymbol{X})\big)^{-1}\big(\boldsymbol{Y} - m(\boldsymbol{Y})\big)$$

That is not a regression function, which is impossible to predict Y^* based on X^* if Y^* is unknown. Therefore, I propose a technique based on linear regression model to define $\mu(\boldsymbol{Y}^*)$ with constraint that each element $\mu(Y^*)$ of $\mu(\boldsymbol{Y}^*)$ depend only on Y^*, for instance:

$$\mu(\boldsymbol{Y}^*) = \begin{pmatrix} \mu(Y_{n+1}) \\ \mu(Y_{n+2}) \\ \vdots \\ \mu(Y_{n+k}) \end{pmatrix}$$

Note, Y^* represents any Y_i belonging to the subprocess $\boldsymbol{Y}^* = (Y_{n+1}, Y_{n+2},\ldots, Y_{n+k})^T$ and X^* represents any X_i belonging to the subprocess $\boldsymbol{X}^* = (X_{n+1}, X_{n+2},\ldots, X_{n+k})^T$. Let $\varphi(X)$ be the transformation function which transforms X space into Y space such that $\varphi(.)$ is invertible.

$$\begin{gathered} Y = \varphi(X), Y^* = \varphi(X^*) \\ X = \varphi^{-1}(Y), X^* = \varphi^{-1}(Y^*) \end{gathered} \quad (3.14)$$

Therefore, $\varphi(.)$ should be defined with constraint that X space and Y space have the same dimension for information preservation. The simplest form of $\varphi(.)$ is identity function.

$$\varphi(X) = X$$

Let Z^* be a regressive estimate of X^*:

$$Z^* = A^*X^* \tag{3.15}$$

Where A^* is regressive coefficient. How to calculate A^* from sample $\{\boldsymbol{X}^*, \varphi^{-1}(\boldsymbol{Y}^*)\}$ will be described later. Please pay attention that this linear regression is totally different from the regression meaning of GPR via posterior PDF. Let $\hat{Y}^*$ be an estimate of Y^* from X^* with association of linear regression model and transformation function $\varphi(.)$.

$$\hat{Y}^* = \varphi(Z^*) = \varphi(A^*X^*) \tag{3.16}$$

Obviously, we have:

$$Z^* = \varphi^{-1}(\hat{Y}^*)$$

Let $|\boldsymbol{X}|$, $|\boldsymbol{X}^*|$, $|\boldsymbol{Y}|$, and $|\boldsymbol{Y}^*|$ be cardinalities (also dimensions) of $\boldsymbol{X}$, $\boldsymbol{X}^*$, $\boldsymbol{Y}$, and $\boldsymbol{Y}^*$, respectively. Of course, we have $|\boldsymbol{X}| = |\boldsymbol{Y}|$ and $|\boldsymbol{X}^*| = |\boldsymbol{Y}^*|$, for example, we also have $|\boldsymbol{X}| = |\boldsymbol{Y}| = n$ and $|\boldsymbol{X}^*| = |\boldsymbol{Y}^*| = k$. Concentration ratio (CR) of a subprocess which is point density of such subprocess is defined by the ratio of cardinality of such subprocess to the cardinality of entire process, for example, CR of $\boldsymbol{X}^*$ over $\{\boldsymbol{X}, \boldsymbol{X}^*\}$ is:

$$cr(\boldsymbol{X}^*|\boldsymbol{X},\boldsymbol{X}^*) = \frac{|\boldsymbol{X}^*|}{|\boldsymbol{X}| + |\boldsymbol{X}^*|} \tag{3.17}$$

Suppose A also being the regressive coefficient estimated from sample $\{\boldsymbol{X}, \varphi^{-1}(\boldsymbol{Y})\}$ was determined before, the mean function $\mu^*(X^*)$ is proposed as a weighted estimation with concentration ratios of $\boldsymbol{X}$ and $\boldsymbol{X}^*$ as follows:

$$\mu^*(X^*) = \varphi\left(\frac{|\boldsymbol{X}|}{|\boldsymbol{X}| + |\boldsymbol{X}^*|}AX^* + \frac{|\boldsymbol{X}^*|}{|\boldsymbol{X}| + |\boldsymbol{X}^*|}A^*X^*\right) \tag{3.18}$$

As a result, the two steps at each iteration of the iterative algorithm for estimating GPR are finetuned as follows:

Table 3.3. GPR algorithm with linear regression.

```
Step 1:
Take new $\boldsymbol{X}^*$
Calculate $A^*$ from sample $\{\boldsymbol{X}^*, \varphi^{-1}(\boldsymbol{Y}^*)\}$
$\mu(\boldsymbol{Y}^*|\boldsymbol{X},\boldsymbol{X}^*) = \mu^*(\boldsymbol{X}^*) + \Sigma(\boldsymbol{X}^*,\boldsymbol{X})\big(\Sigma(\boldsymbol{X},\boldsymbol{X})\big)^{-1}\big(\boldsymbol{Y} - m(\boldsymbol{Y})\big)$
$\Sigma(\boldsymbol{Y}^*|\boldsymbol{X},\boldsymbol{X}^*) = \Sigma(\boldsymbol{X}^*,\boldsymbol{X}^*) - \Sigma(\boldsymbol{X}^*,\boldsymbol{X})\big(\Sigma(\boldsymbol{X},\boldsymbol{X})\big)^{-1}\Sigma(\boldsymbol{X},\boldsymbol{X}^*)$
Step 2:
$m(\boldsymbol{Y}) = \mu(\boldsymbol{Y}^*|\boldsymbol{X},\boldsymbol{X}^*)$
$A = A^*$
$\boldsymbol{X} = \begin{pmatrix}\boldsymbol{X}\\ \boldsymbol{X}^*\end{pmatrix}$ or $\boldsymbol{X} = \boldsymbol{X}^*$
```

Where $\mu^*(\boldsymbol{X}^*)$ is determined based on $\mu^*(X^*)$ as follows:

$$\mu^*(\boldsymbol{X}^*) = \begin{pmatrix}\mu^*(X_{n+1})\\ \mu^*(X_{n+2})\\ \vdots\\ \mu^*(X_{n+k})\end{pmatrix}$$

Note, the vector $\varphi^{-1}(\boldsymbol{Y}^*)$ is determined:

$$\varphi^{-1}(\boldsymbol{Y}^*) = \begin{pmatrix} \varphi^{-1}(Y_{n+1}) \\ \varphi^{-1}(Y_{n+2}) \\ \vdots \\ \varphi^{-1}(Y_{n+k}) \end{pmatrix}$$

Both mean vector $\mu(\boldsymbol{Y}^* \mid \boldsymbol{X}, \boldsymbol{X}^*)$ and covariance matrix $\Sigma(\boldsymbol{Y}^* \mid \boldsymbol{X}, \boldsymbol{X}^*)$ of GPR with linear regression are free from $\boldsymbol{Y}$ because they are totally based on only $\boldsymbol{X}$ and $\boldsymbol{X}^*$.

For interval estimation of Y^* with given X^*, suppose GPR algorithm finished obtaining regression coefficient A after some iterations on $\boldsymbol{X}$, the mean function $\mu(Y^*)$ is reduced with X^* as follows:

$$\mu^*(X^*) = \varphi(AX^*)$$

As a result, the confident interval of Y^* which is the pair $\{\mu(Y^* \mid \boldsymbol{X}, X^*), \Sigma(Y^* \mid \boldsymbol{X}, X^*)\}$ is determined as follows:

$$\begin{gathered} \mu(Y^*|\boldsymbol{X}, X^*) = \varphi(AX^*) + \Sigma(X^*, \boldsymbol{X})\big(\Sigma(\boldsymbol{X}, \boldsymbol{X})\big)^{-1}\big(\boldsymbol{Y} - m(\boldsymbol{Y})\big) \\ \Sigma(Y^*|\boldsymbol{X}, X^*) = \Sigma(X^*, X^*) - \Sigma(X^*, \boldsymbol{X})\big(\Sigma(\boldsymbol{X}, \boldsymbol{X})\big)^{-1}\Sigma(\boldsymbol{X}, X^*) \end{gathered} \tag{3.19}$$

Recall that $m(\boldsymbol{Y})$ is a realistic mean vector which is recurrently determined by previously known $\mu(\boldsymbol{Y} \mid X_{1:n0}, \boldsymbol{X})$:

$$m(\boldsymbol{Y}) = \mu\big(\boldsymbol{Y}\big|X_{1:n_0}, \boldsymbol{X}\big) = \begin{pmatrix} \mu\big(\boldsymbol{Y}\big|X_{1:n_0}, X_1\big) \\ \mu\big(\boldsymbol{Y}\big|X_{1:n_0}, X_2\big) \\ \vdots \\ \mu\big(\boldsymbol{Y}\big|X_{1:n_0}, X_n\big) \end{pmatrix}$$

Because the GPR algorithm here defines the mean function $\mu(Y^*)$ based on multiple linear regression (MLR) model, it is necessary to describe MLR in short. Given a dependent random vector variable $Z = (z_1, z_2,\ldots, z_m)^T$ and an independent random variable $X = (1, x_1, x_2,\ldots, x_n)^T$, MLR tries to establish linear relationship between $Z = (z_1, z_2,\ldots, z_m)^T$ and X so that Z is sum of a linear combination of X and an random error vector $\boldsymbol{\varepsilon}$.

$$Z = AX + \boldsymbol{\varepsilon} \tag{3.20}$$

As a convention, X are called regressor and Z is called responsor whereas $A = (\alpha_0, \alpha_1, \alpha_2,\ldots, \alpha_q)^T$ is mx$(n+1)$ regressive coefficient matrix.

$$A = \begin{pmatrix} a_{00} & a_{11} & a_{12} & \cdots & a_{1n} \\ a_{10} & a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m0} & a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \tag{3.21}$$

Suppose $\boldsymbol{\varepsilon}$ distributes normally with mean vector $\mathbf{0}^T$ and covariance matrix Σ then Z distributes normally with mean vector AX and covariance matrix Σ due to:

$$\begin{gathered} E(Z) = E(AX + \boldsymbol{\varepsilon}) = AX \\ V(Z) = V(AX + \boldsymbol{\varepsilon}) = \Sigma \end{gathered} \tag{3.22}$$

Note, $E(.)$ and $V(.)$ denote theoretical expectation and variance, respectively and Σ is mxm invertible matrix. This implies that the PDF of random variable Z is:

$$\begin{aligned} Z|X \sim f(Z|X, A, \Sigma) &= \mathcal{N}(Z|AX, \Sigma) \\ &= (2\pi)^{-\frac{m}{2}}|\Sigma|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(Z - AX)^T\Sigma^{-1}(Z - AX)\right) \end{aligned}$$

MLR of Z given X is built from sample $\{\boldsymbol{X}, \mathbf{Z}\}$ of size N in form of data matrix as follows:

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}_1^T \\ \boldsymbol{x}_2^T \\ \vdots \\ \boldsymbol{x}_N^T \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1n} \\ 1 & x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & x_{N2} & \cdots & x_{Nn} \end{pmatrix}, \boldsymbol{x}_i = \begin{pmatrix} 1 \\ x_{i1} \\ x_{i2} \\ \vdots \\ x_{in} \end{pmatrix}$$
$$\boldsymbol{Z} = \begin{pmatrix} \boldsymbol{z}_1^T \\ \boldsymbol{z}_2^T \\ \vdots \\ \boldsymbol{z}_N^T \end{pmatrix} = \begin{pmatrix} z_{11} & z_{12} & \cdots & z_{1m} \\ z_{21} & z_{22} & \cdots & z_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ z_{N1} & z_{N2} & \cdots & z_{Nm} \end{pmatrix}, \boldsymbol{z}_i = \begin{pmatrix} z_{i1} \\ z_{i2} \\ \vdots \\ z_{im} \end{pmatrix} \tag{3.23}$$

Therefore, $\boldsymbol{x}_i$ and $\mathbf{z}_i$ is the i^{th} instances of regressor X and responsor Z at the i^{th} row of matrix ($\boldsymbol{X}$, $\mathbf{Z}$). As a convention, we can connect datasets $\boldsymbol{X}$ and $\boldsymbol{Z}$ of MLR here with the GP $\boldsymbol{X}^*$ and the set $\varphi^{-1}(\boldsymbol{Y}^*)$ aforementioned, respectively if removing the first column of values 1 from datasets $\boldsymbol{X}$.

$$\begin{gathered} N \sim k \\ \boldsymbol{x}_i \sim X_{n+i} \\ \boldsymbol{X} \sim \boldsymbol{X}^* \\ \boldsymbol{z}_i \sim \varphi^{-1}(Y_{n+i}) \\ \boldsymbol{Z} \sim \varphi^{-1}(\boldsymbol{Y}^*) \end{gathered} \tag{3.24}$$

The essence of MLR is to estimate the regressive coefficient matrix A and the covariance matrix Σ. By applying maximum likelihood estimation (MLE) method, we obtain estimates $\hat{A}$ and $\hat{\Sigma}$ of A and Σ.

$$\begin{gathered} \hat{A} = \boldsymbol{Z}^T\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1} \\ \hat{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{z}_i - \hat{A}\boldsymbol{x}_i)(\boldsymbol{z}_i - \hat{A}\boldsymbol{x}_i)^T \end{gathered} \tag{3.25}$$

4. Acquisition Functions

Recall that Bayesian optimization (BO) maximizes the acquisition function $\alpha(\boldsymbol{x} \mid f, S)$ so as to search for maximizer $\boldsymbol{x}^*$. Especially, nonparametric BO (NBO) based on Gaussian process regression (GPR) requires support of acquisition function to guide movement of $\boldsymbol{x}$ in the search space so as to reach maximizer $\boldsymbol{x}^*$. Moreover, acquisition function of NBO is created from GPR surrogate model; concretely, it is defined with two essential parameters such as posterior mean $\mu_n(\boldsymbol{x})$ and variance $\sigma_n^2(\boldsymbol{x})$ of GPR.

$$\mu_n(\boldsymbol{x}) = \Sigma(\boldsymbol{x}, \boldsymbol{x}_{1:n})\big(\Sigma(\boldsymbol{x}_{1:n}, \boldsymbol{x}_{1:n})\big)^{-1}\big(y_{1:n} - m(y_{1:n})\big)$$

$$\sigma_n^2(\boldsymbol{x}) = \Sigma(\boldsymbol{x}, \boldsymbol{x}) - \Sigma(\boldsymbol{x}, \boldsymbol{x}_{1:n})\big(\Sigma(\boldsymbol{x}_{1:n}, \boldsymbol{x}_{1:n})\big)^{-1}\Sigma(\boldsymbol{x}_{1:n}, \boldsymbol{x})$$

Note that $m(y_{1:n})$ is a realistic mean which is recurrently determined by previously known $\mu_{n-1}(\boldsymbol{x})$:

$$m(y_n) = \mu_{n-1}(\boldsymbol{x}_n)$$

This is the reason that acquisition function of NBO is denoted as of $\alpha(\boldsymbol{x} \mid \mu_n(\boldsymbol{x}), \sigma_n^2(\boldsymbol{x}))$ in NBO. Therefore, this section focuses on NBO acquisition function. GPR surrogate model of NBO is represented by the posterior PDF of $y = f(\boldsymbol{x})$ given $y_{1:n}$, $\boldsymbol{x}_{1:n}$, and $\boldsymbol{x}$ as follows:

$$y|y_{1:n}, \boldsymbol{x}_{1:n}, \boldsymbol{x} \sim f\big(y\big|y_{1:n}, \boldsymbol{x}_{1:n}, \boldsymbol{x}, \mu(.), \Sigma(.)\big) = \mathcal{N}\big(y\big|\mu_n(\boldsymbol{x}), \sigma_n^2(\boldsymbol{x})\big)$$

Both mean $\mu_n(\boldsymbol{x})$ and variance $\sigma_n^2(\boldsymbol{x})$ are functions of only $\boldsymbol{x}$ because $y_{1:n}$ and $\boldsymbol{x}_{1:n}$ were known, which is necessary to determine acquisition function which is function of $\boldsymbol{x}$ too. Note, in this research, notation $f(.|.)$ often denotes PDF, which is not concrete function and notation $\mathcal{N}(.|.)$ denotes normal distribution. The normal PDF of $y = f(\boldsymbol{x})$ given $y_{1:n}$, $\boldsymbol{x}_{1:n}$, and $\boldsymbol{x}$ is formulated as follows:

$$y|y_{1:n},\boldsymbol{x}_{1:n},\boldsymbol{x} \sim \mathcal{N}\left(y\middle|\mu_n(\boldsymbol{x}),\sigma_n^2(\boldsymbol{x})\right) = \frac{1}{\sqrt{2\pi\sigma_n^2(\boldsymbol{x})}}\exp\left(-\frac{\left(y-\mu_n(\boldsymbol{x})\right)^2}{2\sigma_n^2(\boldsymbol{x})}\right)$$

This PDF should be standardized into standard normal PDF $\phi(z)$ because values of standard normal cumulative distribution function (standard normal CDF) $\Phi(z)$ were evaluated in practice. Standard normal distribution is also called standard Gaussian distribution. If variable y is standardized into variable z such that $z = \frac{y-\mu_n(\boldsymbol{x})}{\sigma_n(\boldsymbol{x})}$ then, distribution of y is equivalent to distribution of z via its PDF $\phi(z)$ also its CDF $\Phi(z)$.

$$z = \frac{y-\mu_n(\boldsymbol{x})}{\sigma_n(\boldsymbol{x})}$$
$$\phi(z) = \mathcal{N}(z|0,1) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{z^2}{2}\right) \quad (4.1)$$
$$\Phi(z_0) = \phi(z < z_0) = \int_{-\infty}^{z_0} \phi(z)dz$$

Where the quantity $\sigma_n(\boldsymbol{x}) = \sqrt{\sigma_n^2(\boldsymbol{x})}$ is called standard deviation of y given $y_{1:n}$, $\boldsymbol{x}_{1:n}$, and $\boldsymbol{x}$. Note, standard normal PDF has mean 0 and variance 1. We have some important properties of standard normal distribution.

$$\phi(z < z_0) = \phi(z \leq z_0)$$
$$\phi(z > z_0) = \phi(z \geq z_0)$$
$$\Phi'(z) = \phi(z)$$
$$\Phi(-z_0) = 1 - \Phi(z_0) = \phi(z > z_0) = \int_{z_0}^{+\infty} \phi(z)dz$$

If the PDF of y is standardized, the statement that y distributes normally given $y_{1:n}$, $\boldsymbol{x}_{1:n}$, and $\boldsymbol{x}$ with mean $\mu_n(\boldsymbol{x})$ and variance $\sigma_n^2(\boldsymbol{x})$ will be equivalent to the statement that the distribution of z given $y_{1:n}$, $\boldsymbol{x}_{1:n}$, and $\boldsymbol{x}$ is standard normal distribution where:

$$z = \frac{y-\mu_n(\boldsymbol{x})}{\sigma_n(\boldsymbol{x})}$$

Of course, we have:

$$y = \mu_n(\boldsymbol{x}) + \sigma_n(\boldsymbol{x})z \quad (4.2)$$

Recall that the acquisition function $\alpha(\boldsymbol{x} \mid f, S)$ based on mean $\mu_n(\boldsymbol{x})$ and variance $\sigma_n^2(\boldsymbol{x})$ of GPR surrogate model is denoted as follows, which implies that it is function of $\boldsymbol{x}$.

$$\alpha(\boldsymbol{x}|f,S) = \alpha\left(\boldsymbol{x}\middle|\mu_n(\boldsymbol{x}),\sigma_n^2(\boldsymbol{x})\right)$$

Recall, given acquisition function $\alpha(\boldsymbol{x} \mid \mu_n(\boldsymbol{x}), \sigma_n^2(\boldsymbol{x}))$ based on $\mu_n(\boldsymbol{x})$ and $\sigma_n^2(\boldsymbol{x})$, and also known sample $\boldsymbol{D}_n = \boldsymbol{x}_{1:n} = \{\boldsymbol{x}_1, \boldsymbol{x}_2,\ldots, \boldsymbol{x}_n\}$, let $\boldsymbol{x}_{n+1}$ be a maximizer of $\alpha(\boldsymbol{x} \mid \mu_n(\boldsymbol{x}), \sigma_n^2(\boldsymbol{x}))$ in NBO algorithm at a current iteration.

$$\boldsymbol{x}_{n+1} = \underset{\boldsymbol{x}}{\operatorname{argmax}}\, \alpha\left(\boldsymbol{x}\middle|\mu_n(\boldsymbol{x}),\sigma_n^2(\boldsymbol{x})\right)$$

There are some acquisition functions, for example, probability of improvement, expected improvement, entropy search, and upper confidence bound. Let $\boldsymbol{x}^*$ and $y^* = f(\boldsymbol{x}^*)$ be the maximizer and its respective maximum value of target function to which all acquisitions aim. Both probability of improvement (PI) technique and expected improvement (EI) technique belong to improvement approach in which their acquisition functions, which are also utility functions in decision theory, are expectations of a predefined respective improvement function but their definitions of improvement function are different. For instance, the improvement function denoted $I(\boldsymbol{x})$ sets the deviation between the next value $y = f(x)$ and the current and temporal maximum $y_n^* = f(x_n^*)$ with expectation that y will be larger than y_n^* for searching realistic maximizer x^*; otherwise, $I(x)$ is 0.

$$I(\boldsymbol{x}) \equiv \max(y - y^*, 0) = \max(f(\boldsymbol{x}) - y_n^*, 0)$$

Note, given sample $\boldsymbol{D}_n = \boldsymbol{x}_{1:n} = \{\boldsymbol{x}_1, \boldsymbol{x}_2,\ldots, \boldsymbol{x}_n\}$ and its evaluations $y_{1:n} = \{y_1, y_2,\ldots, y_n\}$, we have $y_n^* = \max\{y_1, y_2,\ldots, y_n\}$ which is the temporal maximum at current iteration of NBO algorithm. When the PDF of y is standardized, $y = \mu_n(\boldsymbol{x}) + \sigma_n(\boldsymbol{x})z$ then the improvement function is rewritten:

$$I(\boldsymbol{x}) = \max(\mu_n(\boldsymbol{x}) + \sigma_n(\boldsymbol{x})z - y_n^*, 0) \tag{4.3}$$

Probability of improvement (PI) technique defines acquisition function as the probability of $I(\boldsymbol{x}) > 0$, which means that the probability of the event that next candidate of target maximizer is larger than the old one must be maximized (Shahriari, Swersky, Wang, Adams, & Freitas, 2016, p. 160).

$$\alpha_{\text{PI}}(\boldsymbol{x}) = \alpha\left(\boldsymbol{x}\middle|\mu_n(\boldsymbol{x}), \sigma_n^2(\boldsymbol{x})\right) = P(I(\boldsymbol{x}) > 0)$$

Note, notation $P(.)$ denotes probability. As a result, PI acquisition function is determined as follows:

$$\alpha_{\text{PI}}(\boldsymbol{x}) = \Phi\left(\frac{\mu_n(\boldsymbol{x}) - y_n^*}{\sigma_n(\boldsymbol{x})}\right) \tag{4.4}$$

Proof,

$$\alpha_{\text{PI}}(\boldsymbol{x}) = P(I(\boldsymbol{x}) > 0) = P(\mu_n(\boldsymbol{x}) + \sigma_n(\boldsymbol{x})z - y_n^* > 0)$$
$$= P\left(z > \frac{y_n^* - \mu_n(\boldsymbol{x})}{\sigma_n(\boldsymbol{x})}\right) = \phi\left(z > \frac{y_n^* - \mu_n(\boldsymbol{x})}{\sigma_n(\boldsymbol{x})}\right) = \Phi\left(\frac{\mu_n(\boldsymbol{x}) - y_n^*}{\sigma_n(\boldsymbol{x})}\right)$$
$$\big(\text{Due to } \phi(z > z_0) = \Phi(-z_0)\big)\blacksquare$$

Therefore, the maximizer $\boldsymbol{x}_{n+1}$ of PI acquisition function in NBO algorithm at a current iteration is determined as follows:

$$\boldsymbol{x}_{n+1} = \underset{\boldsymbol{x}}{\operatorname{argmax}}\, \Phi\left(\frac{\mu_n(\boldsymbol{x}) - y_n^*}{\sigma_n(\boldsymbol{x})}\right)$$

Expected improvement (EI) technique defines acquisition function as expectation of $I(\boldsymbol{x})$, which means that the mean of the event that next candidate of target maximizer is larger than the old one must be maximized.

$$\alpha_{\text{EI}}(\boldsymbol{x}) = \alpha\left(\boldsymbol{x}\middle|\mu_n(\boldsymbol{x}), \sigma_n^2(\boldsymbol{x})\right) = \int_{-\infty}^{+\infty} I(\boldsymbol{x})\phi(z)dz$$

$$= \int_{-\infty}^{+\infty} \max(\mu_n(\boldsymbol{x}) + \sigma_n(\boldsymbol{x})z - y_n^*, 0)\phi(z)dz$$

Note, such expectation of $I(\boldsymbol{x})$ is determined with the standard PDF $\phi(z)$ given $y_{1:n}$, $\boldsymbol{x}_{1:n}$, and $\boldsymbol{x}$. As a result, EI acquisition function is determined as follows (Frazier, 2018, p. 7):

$$\alpha_{\text{EI}}(\boldsymbol{x}) = (\mu_n(\boldsymbol{x}) - y_n^*)\Phi\left(\frac{\mu_n(\boldsymbol{x}) - y_n^*}{\sigma_n(\boldsymbol{x})}\right) + \sigma_n(\boldsymbol{x})\phi(y_n^*) \tag{4.5}$$

Proof (Kamperis, 2021),

$$\alpha_{\text{EI}}(\boldsymbol{x}) = \int_{-\infty}^{+\infty} \max(\mu_n(\boldsymbol{x}) + \sigma_n(\boldsymbol{x})z - y_n^*, 0)\phi(z)dz$$

$$= \int_{-\infty}^{y_n^*} \max(\mu_n(\boldsymbol{x}) + \sigma_n(\boldsymbol{x})z - y_n^*, 0)\phi(z)dz$$

$$+ \int_{y_n^*}^{+\infty} \max(\mu_n(\boldsymbol{x}) + \sigma_n(\boldsymbol{x})z - y_n^*, 0)\phi(z)dz$$

$$= \int_{y_n^*}^{+\infty} (\mu_n(\boldsymbol{x}) + \sigma_n(\boldsymbol{x})z - y_n^*)\phi(z)dz$$

$$\big(\text{Due to } \max(\mu_n(\boldsymbol{x}) + \sigma_n(\boldsymbol{x})z - y_n^*, 0)\big)$$

$$= (\mu_n(\boldsymbol{x}) - y_n^*) \int_{y_n^*}^{+\infty} \phi(z)dz + \sigma_n(\boldsymbol{x}) \int_{y_n^*}^{+\infty} z\phi(z)dz$$

$$= (\mu_n(\boldsymbol{x}) - y_n^*)\Phi\left(\frac{\mu_n(\boldsymbol{x}) - y_n^*}{\sigma_n(\boldsymbol{x})}\right) + \sigma_n(\boldsymbol{x}) \int_{y_n^*}^{+\infty} z\phi(z)dz$$

$$\left(\text{Due to } \int_{y_n^*}^{+\infty} \phi(z)dz = \Phi(-z) = \Phi\left(\frac{\mu_n(\boldsymbol{x}) - y_n^*}{\sigma_n(\boldsymbol{x})}\right)\right)$$

$$= (\mu_n(\boldsymbol{x}) - y_n^*)\Phi\left(\frac{\mu_n(\boldsymbol{x}) - y_n^*}{\sigma_n(\boldsymbol{x})}\right) + \frac{\sigma_n(\boldsymbol{x})}{\sqrt{2\pi}} \int_{y_n^*}^{+\infty} z\mathrm{exp}\left(-\frac{z^2}{2}\right)dz$$

$$= (\mu_n(\boldsymbol{x}) - y_n^*)\Phi\left(\frac{\mu_n(\boldsymbol{x}) - y_n^*}{\sigma_n(\boldsymbol{x})}\right) - \frac{\sigma_n(\boldsymbol{x})}{\sqrt{2\pi}} \int_{y_n^*}^{+\infty} d\left(\exp\left(-\frac{z^2}{2}\right)\right)$$

$$= (\mu_n(\boldsymbol{x}) - y_n^*)\Phi\left(\frac{\mu_n(\boldsymbol{x}) - y_n^*}{\sigma_n(\boldsymbol{x})}\right) - \frac{\sigma_n(\boldsymbol{x})}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right)\Big|_{y_n^*}^{+\infty}$$

$$= (\mu_n(\boldsymbol{x}) - y_n^*)\Phi\left(\frac{\mu_n(\boldsymbol{x}) - y_n^*}{\sigma_n(\boldsymbol{x})}\right) + \frac{\sigma_n(\boldsymbol{x})}{\sqrt{2\pi}} \exp\left(-\frac{(y_n^*)^2}{2}\right)$$

$$= (\mu_n(\boldsymbol{x}) - y_n^*)\Phi\left(\frac{\mu_n(\boldsymbol{x}) - y_n^*}{\sigma_n(\boldsymbol{x})}\right) + \sigma_n(\boldsymbol{x})\phi(y_n^*)$$

$$\left(\text{Due to } \phi(y_n^*) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y_n^*)^2}{2}\right)\right) \blacksquare$$

Therefore, the maximizer $\boldsymbol{x}_{n+1}$ of EI acquisition function in NBO algorithm at a current iteration is determined as follows:

$$\boldsymbol{x}_{n+1} = \underset{\boldsymbol{x}}{\mathrm{argmax}}\left((\mu_n(\boldsymbol{x}) - y_n^*)\Phi\left(\frac{\mu_n(\boldsymbol{x}) - y_n^*}{\sigma_n(\boldsymbol{x})}\right) + \sigma_n(\boldsymbol{x})\phi(y_n^*)\right)$$

An important clue of NBO is that maximizing acquisition function should be cheaper than maximizing target function $f(x)$ if $f(x)$ is analytic function which is appropriate to analytic approach. Fortunately, it is not prerequisite to find out global maximizer of acquisition function because essentially NBO is a reinforcement algorithm whose solution candidate for x^* is improved

doi:10.20944/preprints202303.0292.v1

progressively and moreover, acquisition function itself makes tradeoff between exploitation (convergence in searching maximizer) and exploration (searching global maximizer) for NBO. Therefore, it is possible to apply traditional mathematical methods such as Newton-Raphson and gradient descent for obtaining the local maximizer $\boldsymbol{x}_{n+1}$ of acquisition function. It is only necessary to check first-order derivative known as gradient of acquisition function when traditional methods like Newton-Raphson and gradient descent require existence of gradient. Because PI is simpler than EI, how to maximize PI is mentioned here as an example of acquisition maximization.

$$\boldsymbol{x}_{n+1} = \underset{\boldsymbol{x}}{\operatorname{argmax}}\, \alpha_{\text{PI}}(\boldsymbol{x}) = \underset{\boldsymbol{x}}{\operatorname{argmax}}\, \Phi\left(\frac{\mu_n(\boldsymbol{x}) - y_n^*}{\sigma_n(\boldsymbol{x})}\right)$$

Now we only need to determine the gradient of $\alpha_{\text{PI}}(\boldsymbol{x})$ with regard to variable $\boldsymbol{x}$. Let $\Sigma'(\boldsymbol{x}_i, \boldsymbol{x})$ be gradient (first-order derivative) of $\Sigma(\boldsymbol{x}_i, \boldsymbol{x})$ with regard to $\boldsymbol{x}$ given known $\boldsymbol{x}_i = (x_{i1}, x_{i2},\ldots, x_{in})^T$ and unknown $\boldsymbol{x} = (x_1, x_2,\ldots, x_n)^T$. As a convention, $\Sigma'(\boldsymbol{x}_i, \boldsymbol{x})$ is row vector. If covariance function produces symmetric matrix then $\Sigma'(\boldsymbol{x}_i, \boldsymbol{x}) = \Sigma'(\boldsymbol{x}, \boldsymbol{x}_i)$. For example, given simple squared exponential kernel function $\Sigma_{SE}(\boldsymbol{x}, \boldsymbol{x}_i) = \exp\left(-\frac{|\boldsymbol{x}-\boldsymbol{x}_i|^2}{2}\right)$, its gradient with regard to $\boldsymbol{x}$ is:

$$\Sigma'(\boldsymbol{x}, \boldsymbol{x}_i) = \frac{d}{d\boldsymbol{x}}\left(\exp\left(-\frac{|\boldsymbol{x}-\boldsymbol{x}_i|^2}{2}\right)\right) = \frac{d}{d\boldsymbol{x}}\left(\exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{x}_i)^T(\boldsymbol{x}-\boldsymbol{x}_i)}{2}\right)\right)$$
$$= -(\boldsymbol{x}-\boldsymbol{x}_i)^T \exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{x}_i)^T(\boldsymbol{x}-\boldsymbol{x}_i)}{2}\right)$$

Let $\Sigma'(\boldsymbol{x}_{1:n}, \boldsymbol{x})$ be gradient of $\Sigma(\boldsymbol{x}_{1:n}, \boldsymbol{x})$, essentially it is a matrix but written as column vector whose each element is row vector:

$$\Sigma'(\boldsymbol{x}_{1:n}, \boldsymbol{x}) = \begin{pmatrix} \Sigma'(\boldsymbol{x}_1, \boldsymbol{x}) \\ \Sigma'(\boldsymbol{x}_2, \boldsymbol{x}) \\ \vdots \\ \Sigma'(\boldsymbol{x}_n, \boldsymbol{x}) \end{pmatrix}$$

If covariance function produces symmetric matrix, we have:

$$\Sigma'(\boldsymbol{x}, \boldsymbol{x}_{1:n}) = \left(\Sigma'(\boldsymbol{x}_{1:n}, \boldsymbol{x})\right)^T$$

Gradients of $\mu_n(\boldsymbol{x})$ and $\sigma_n^2(\boldsymbol{x})$ with regard to $\boldsymbol{x}$ are:

$$\mu_n'(\boldsymbol{x}) = \left(y_{1:n} - m(y_{1:n})\right)^T \left(\Sigma(\boldsymbol{x}_{1:n}, \boldsymbol{x}_{1:n})\right)^{-1} \Sigma'(\boldsymbol{x}_{1:n}, \boldsymbol{x})$$
$$\left(\sigma_n^2(\boldsymbol{x})\right)' = \Sigma'(\boldsymbol{x}, \boldsymbol{x}) - 2\Sigma(\boldsymbol{x}, \boldsymbol{x}_{1:n})\left(\Sigma(\boldsymbol{x}_{1:n}, \boldsymbol{x}_{1:n})\right)^{-1} \Sigma'(\boldsymbol{x}_{1:n}, \boldsymbol{x}) \tag{4.6}$$

Note, $\Sigma(\boldsymbol{x}_{1:n}, \boldsymbol{x}_{1:n})$ is invertible and symmetric. Because of $\sigma_n(\boldsymbol{x}) = \sqrt{\sigma_n^2(\boldsymbol{x})}$, we have gradient of $\sigma_n(\boldsymbol{x})$ as follows:

$$\sigma_n'(\boldsymbol{x}) = \frac{\left(\sigma_n^2(\boldsymbol{x})\right)'}{2\sqrt{\sigma_n^2(\boldsymbol{x})}}$$

Let $z_n(\boldsymbol{x}) = (y_n^* - \mu_n(\boldsymbol{x})) / \sigma_n(\boldsymbol{x})$ as function of $\boldsymbol{x}$, its gradient with regard to $\boldsymbol{x}$ is:

$$z_n'(\boldsymbol{x}) = -\frac{\mu_n'(\boldsymbol{x})\sigma_n(\boldsymbol{x}) + \left(y_n^* - \mu_n(\boldsymbol{x})\right)\sigma_n'(\boldsymbol{x})}{\sigma_n^2(\boldsymbol{x})} \tag{4.7}$$

Gradient of PI is totally determined as follows:

$$\alpha_{\text{PI}}'(\boldsymbol{x}) = \Phi'\left(\frac{\mu_n(\boldsymbol{x}) - y_n^*}{\sigma_n(\boldsymbol{x})}\right) = -\phi\left(z_n(\boldsymbol{x})\right)z_n'(\boldsymbol{x}) \tag{4.8}$$

Where,

$$z_n(\boldsymbol{x}) = \frac{y_n^* - \mu_n(\boldsymbol{x})}{\sigma_n(\boldsymbol{x})}$$

It is possible to apply traditional methods like Newton-Raphson and gradient descent to find out local maximizer $\boldsymbol{x}_{n+1}$ of PI because PI gradient is determined given $\boldsymbol{x}$. Similarly, it is easy to obtain gradient of EI acquisition function because gradients of $\mu_n(\boldsymbol{x})$ and $\sigma_n^2(\boldsymbol{x})$ are most important, which are the base to calculate gradients of PI and EI.

5. Conclusions

This research focuses on nonparametric Bayesian optimization (NBO) although there are interesting aspects of parametric Bayesian optimization (PBO). Strictness is emphasized in some researches but other ones need only reasonableness, especially applied researches or heuristic researches. Although NBO belongs to applied mathematics, its mathematical base is strict even its outlook of surrogate model and acquisition function maximization is seemly tricky. Therefore, its costing price is that NBO is, in turn, based on traditional convex optimization methods like Newton-Raphson and gradient descent to maximize acquisition function, which means that NBO requires other knowledge outside its circle while heuristic methods like particle swarm optimization (PSO) and ant bee colony (ABC) do not require mathematical constraints. Anyhow, NBO is a significant solution of global optimization problem because, within requisite assumption of too strict mathematical constraints, the progress of applied mathematics will be hesitative. Therefore, the main problem here is effectiveness of NBO in balancing exploration and exploitation for search global optimizer. Heuristic methods like PSO are proved their simplicity and feasibility when they do not focus on complicated mathematical tools although there are some researches trying to apply mathematical theories into explaining them. For human research, whether thinking via theory is prior to natural imitation or vice versa is a big question whose answers can be dependent on individuals. This question is equivalent to the other question that which one among logicality and imagination is important.

References

1. Frazier, P. I. (2018, July 8). A Tutorial on Bayesian Optimization. *arXiv*. doi:10.48550/arXiv.1807.02811
2. Kamperis, S. (2021, June 11). *Acquisition functions in Bayesian Optimization*. Retrieved from Stathis Kamperis's GitHub page: https://ekamperi.github.io/machine%20learning/2021/06/11/acquisition-functions.html
3. Neapolitan, R. E. (2003). *Learning Bayesian Networks.* Upper Saddle River, New Jersey, USA: Prentice Hall.
4. Nguyen, L. (2022, June 27). A short study on minima distribution. *Preprints*. doi:10.20944/preprints202206.0361.v1
5. Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & Freitas, N. d. (2016, January). Taking the Human Out of the Loop: A Review of Bayesian Optimization. (J. H. Trussell, Ed.) *Proceedings of the IEEE, 104*(1), 148 - 175. doi:10.1109/JPROC.2015.2494218
6. Wang, J. (2022, April 18). An Intuitive Tutorial to Gaussian Processes Regression. *arXiv*. doi:10.48550/arXiv.2009.10862