Stochastic and deterministic algorithms for continuous black-box optimization

Wang, H.

CHAPTER 3

Kriging/Gaussian Process Regression

As nonparametric regression/interpolation methods, Kriging and Gaussian Process regression (GPR) (Stein, 1999; Rasmussen and Williams, 2006) are widely used as a (meta-)modeling tool in *Design and Analysis of Computer Experiments* (Sacks et al., 1989; Santner et al., 2003), Surrogate-assisted Evolutionary Algorithms (Emmerich, 2005; Jin, 2011), Global Optimization (Jones et al., 1998; Močkus, 2012) and Algorithm Configuration (Hutter et al., 2011; Bartz-Beielstein et al., 2005). Commonly, Kriging and GPR are used interchangeably in the literature due to the fact that they represent exactly the same estimator. However, they are motivated and derived differently and thus possess different assumptions and properties: Kriging is originated in geostatistics (Krige, 1951) while GPR is usually discussed in nonparametric Bayesian inference (van der Vaart and van Zanten, 2008). In this chapter, we shall compare these two methods conceptually and discuss to which extent they can be used interchangeably.

Moreover, it is well-known that Kriging/GPR suffers from the cubic time complexity and quadratic space complexity as the number of the data points increases. Several existing solutions to this issue are summarized and compared in this chapter. In addition, a novel solution framework, **Cluster Kriging** (CK), is proposed, in which the data set is divided into several folds and Kriging estimators constructed on each fold are combined in multiple ways. Similar to our argument on Kriging/GPR above, two parallel derivations of Cluster Kriging are presented: one approach taking the properties of Gaussian process (section 3.2.3) and the other one built on the theory of the *best linear unbiased prediction* (BLUP). In addition, the ability of reducing the time complexity is validated through experimental studies. To illustrate the usefulness of Cluster Kriging, it is then applied as the surrogate model in the efficient global optimization (EGO), aiming at reducing the running time of EGO without slowing down its convergence rate. The resulting CK-EGO algorithm is tested on some benchmark functions in Section 3.3.

3.1 General Discussion

The discussion begins with assumptions that are common in both Kriging and GPR. Consider a (noiseless) real-valued function of interest $f : \mathrm{S} \subset \mathbb{R}^d \to \mathbb{R}$. It could serve as an objective function in optimization or a response variable in meta-modeling. Without loss of generality, we assume space $L^2(\mathrm{S})$ for such functions. A real-valued stochastic process $Y = \{Y(\mathbf{x}) : \mathbf{x} \in \mathrm{S}\}$ is a collection of random variables indexed by a set S, where random variables

$$\forall \mathbf{x} \in \mathrm{S}, \quad Y(\mathbf{x}) : \Omega \to \mathbb{R},$$

are defined between the probability space $(\Omega, \mathscr{F}, \mathbb{P})$ and the measurable space $(\mathbb{R}, \mathscr{B})$ ($\mathscr{F}$ is the σ-algebra on Ω and $\mathscr{B}$ is the Borel algebra on the real line). In order to make clear arguments, it is convenient to define the stochastic process Y as a measurable function of two variables (Øksendal, 2003),

$$Y : \mathrm{S} \times \Omega \to \mathbb{R}, \quad (\mathbf{x}, \omega) \mapsto Y(\mathbf{x}, \omega).$$

Using this notation, for every point $\mathbf{x} \in \mathrm{S}$, $Y(\mathbf{x}, \cdot)$ denotes the random variable indexed by $\mathbf{x}$ and for every outcome $\omega \in \Omega$, $Y(\cdot, \omega) : \mathrm{S} \to \mathbb{R}$ is a real-valued function and is called **sample path/sample function** of process Y. In the following discussion, when the outcome ω is not explicitly given, we shall abbreviate $Y(\mathbf{x}, \cdot)$ as $Y(\mathbf{x})$.

The general assumption of Kriging/GPR is: f **is a sample function of** Y. Commonly, the stochastic process Y is specified by two components: a *deterministic* trend function t and a *centered* stochastic process Z:

$$Y(\mathbf{x}) = t(\mathbf{x}) + Z(\mathbf{x}). \tag{3.1}$$

In general, instead of specifying the distribution for Z, only the mean and the covariance structure are given: $\forall \mathbf{x}, \mathbf{x}' \in \mathrm{S}, \mathbb{E}Z(\mathbf{x}) = 0, \mathrm{Cov}\{Z(\mathbf{x}), Z(\mathbf{x}')\} = k(\mathbf{x}, \mathbf{x}')$. Note that $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a *positive-definite kernel*, called *covariance function.* Suppose the target function f is evaluated at n points[1]: $\mathbf{y} =$

[1] Those points are typically obtained via a design of experiment, e.g., Latin hypercube sampling.

$(f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n))^\top$. According to the assumption of Kriging, $\mathbf{y}$ also represents the realization of the random vector $\boldsymbol{\psi} = (Y(\mathbf{x}_1), Y(\mathbf{x}_2), \ldots, Y(\mathbf{x}_n))^\top$:

$$\exists \omega \in \Omega, \quad \mathbf{y} = \boldsymbol{\psi}(\omega) = (Y(\mathbf{x}_1, \omega), Y(\mathbf{x}_2, \omega), \ldots, Y(\mathbf{x}_n, \omega))^\top.$$

Then the task is to *approximate* the value $f(\mathbf{x})$ at an unobserved location $\mathbf{x}$ using vector $\mathbf{y}$. In the following, each component of process Y is specified. Normally, the trend function takes a parametric form. For example, it could be either a constant

$$t(\mathbf{x}) = \beta,$$

or the linear combination of a few basis functions,

$$t(\mathbf{x}) = \sum_{i=0}^{p} \beta_i b_i(\mathbf{x}) = \mathbf{b}(\mathbf{x})^\top \boldsymbol{\beta}, \quad b_0 = 1, \tag{3.2}$$

where $p + 1$ fixed basis functions b_i, abbreviated as $\mathbf{b} = (b_0, b_2, \ldots, b_p)^\top$, are typically specified by the user. Typically, the first or second order polynomial basis functions (Lophaven et al., 2002) are used. Depending on the form of the trend function and whether the coefficients β are known, Kriging methods are further categorized into *Simple Kriging*, *Ordinary Kriging* and *Universal Kriging* (Zimmerman et al., 1999; Stein, 1999). For detailed discussions on the history of Kriging variants, please see Cressie (2015, 1990). Those terms are clarified in Tab. 3.1. Note that, it is unnecessary to distinguish the constant and basis function because the former is special case of the latter when $p = 0$. Thus, we shall always refer to Eq. (3.2) for the trend function. For brevity, the trend component of all observations is denoted as:

Table 3.1: Taxonomy of Kriging methods.

	known β	β to estimate
Constant trend	Simple	Ordinary
Basis functions	None	Universal

$$\mathbf{t} = (t(\mathbf{x}_1), t(\mathbf{x}_2), \ldots, t(\mathbf{x}_n))^\top = \mathbf{B}\boldsymbol{\beta}, \quad \mathbf{B} = [\mathbf{b}(\mathbf{x}_1), \mathbf{b}(\mathbf{x}_2), \ldots, \mathbf{b}(\mathbf{x}_n)]^\top.$$

The covariance function is required to be a positive-definite kernel. A symmetric function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is positive definite (p.d.) if the following condition

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0 \tag{3.3}$$

holds for $\forall n \in \mathbb{N}, \forall \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in \mathrm{S}$ and $\forall c_1, c_2, \dots, c_n \in \mathbb{R}$. Some commonly used kernels include: Gaussian kernel, also known as radial basis functions (RBF) (Buhmann, 2003):

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\sum_{i=1}^{d} \frac{(x_i - x_i')^2}{2\theta_i^2}\right), \tag{3.4}$$

and the Matérn 3/2 kernel (Rasmussen and Williams, 2006):

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \prod_{i=1}^{d} \left(1 + \sqrt{3}\frac{h_i}{\theta_i}\right) \exp\left(-\sqrt{3}\frac{h_i}{\theta_i}\right), \quad h_i = |x_i - x_i'|. \tag{3.5}$$

Note that $\forall \mathbf{x} \in \mathrm{S}, k(\mathbf{x}, \mathbf{x}) = \sigma^2$. The parameters σ^2 and $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_d)^\top$ are the so-called *hyper-parameters* and are usually estimated from the data (please see the discussion on the likelihood function below). Throughout this thesis, the Matérn 3/2 kernel is applied to Kriging modeling as the Matérn family of kernels allows for accurate approximations of the local variation in the data (Stein, 1999). Please see Rasmussen and Williams (2006) for more kernel functions. As the kernel function governs the covariance structure, it is necessary to discuss the statistical properties of Z, when choosing the Matérn 3/2 kernel:

- Stationary: a stochastic process Z is called *weakly stationary* if for all $\mathbf{x}, \mathbf{x}'$ in its index set, the mean function is constant and the covariance only depends on $\mathbf{x} - \mathbf{x}'$, namely $\mathrm{Cov}\{Z(\mathbf{x}), Z(\mathbf{x}')\} = k(\mathbf{x} - \mathbf{x}', 0)$. This is a common assumption made on stochastic processes and it is assured by the Matérn 3/2 kernel.

- Isotropy: a stochastic process Z is called *weakly isotropic* if for all the locations of its index set, its mean function is constant and its covariance of $Z(\mathbf{x}), Z(\mathbf{x}')$ only depends on the Euclidean distance between the location, namely $\mathrm{Cov}\{Z(\mathbf{x}), Z(\mathbf{x}')\} = k(\|\mathbf{x} - \mathbf{x}'\|, 0)$. Intuitively, isotropy indicates that the process is rotation-invariant because $\|\mathbf{x} - \mathbf{x}'\| = \|\mathbf{R}(\mathbf{x} - \mathbf{x}')\|$ holds for any orthogonal matrix $\mathbf{R}$. It is straightforward to check that Matérn 3/2 kernel does not imply this property. In practice, the isotropy is too strong to assume on the data and thus non-isotropic kernels are suggested.

Lastly, some notations are introduced: the covariance matrix of $\mathbf{y}$ is written as:

$$\mathbf{K}(\sigma^2, \boldsymbol{\theta}) = \mathbb{E}\{(\mathbf{y}-\mathbf{t})(\mathbf{y}-\mathbf{t})^\top\} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \dots & k(\mathbf{x}_1, \mathbf{x}_n) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \dots & k(\mathbf{x}_2, \mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & k(\mathbf{x}_n, \mathbf{x}_2) & \dots & k(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix},$$

and its covariances with $Y(\mathbf{x})$ is denoted as $\mathbf{k}(\mathbf{x}) = (k(\mathbf{x}, \mathbf{x}_1), k(\mathbf{x}, \mathbf{x}_2), \dots, k(\mathbf{x}, \mathbf{x}_n))^\top$. The covariance matrix will be denoted as $\mathbf{K}$ for short in the following discussions. In addition, from the definition of positive definite kernel (Eq. (3.3)), it is straightforward to verify that $\mathbf{K}$ is a *positive semi-definite matrix*. Moreover, the singular case is ignored throughout this thesis and thus $\mathbf{K}$ is assumed to be a *positive-definite matrix*.

3.1.1 Best Linear Unbiased Predictor

Consider a finite collection of random variables: $\boldsymbol{\psi} = (Y(\mathbf{x}_1), Y(\mathbf{x}_2), \dots, Y(\mathbf{x}_n))^\top$. The basic idea is to construct a *nonparametric linear predictor* $\widehat{Y} = \boldsymbol{\alpha}^\top\boldsymbol{\psi} + \alpha_0$ to predict $Y(\mathbf{x})$. The *best predictor*[1] is chosen such that the following *risk function* (expected quadratic loss/mean squared error) is minimized:

$$\begin{aligned} R(\widehat{Y}, Y) &= \mathbb{E}\{\boldsymbol{\alpha}^\top\boldsymbol{\psi} + \alpha_0 - Y(\mathbf{x})\}^2 \\ &= \left(\mathbb{E}\{\boldsymbol{\alpha}^\top\boldsymbol{\psi} + \alpha_0 - Y(\mathbf{x})\}\right)^2 + \mathrm{Var}\{\boldsymbol{\alpha}^\top\boldsymbol{\psi} + \alpha_0 - Y(\mathbf{x})\} \end{aligned} \tag{3.6}$$

Note that, in this risk function the expectation is taken w.r.t. the joint distribution of $\boldsymbol{\psi}$ and $Y(\mathbf{x})$. In addition, a *linear unbiased predictor* (LUP) (Stein, 1999) is intended, which can be obtained by enforcing the following constraint:

$$\begin{aligned} \left(\mathbb{E}\{\boldsymbol{\alpha}^\top\boldsymbol{\psi} + \alpha_0 - Y(\mathbf{x})\}\right)^2 = 0 &\iff \forall\boldsymbol{\beta} \in \mathbb{R}^{p+1}\left(\boldsymbol{\alpha}^\top\mathbf{B}\boldsymbol{\beta} + \alpha_0 - \mathbf{b}^\top\boldsymbol{\beta} = 0\right) \\ &\iff \alpha_0 = 0 \wedge \mathbf{B}^\top\boldsymbol{\alpha} = \mathbf{b}. \end{aligned}$$

Therefore the existence of the LUP depends on the solution of the linear system $\mathbf{B}^\top\boldsymbol{\alpha} = \mathbf{b}$. We suppose the solution to this system exists for now ($\mathbf{b}$ is in the column space of $\mathbf{B}^\top$). As the bias in Eq. (3.6) is restricted to zero, only the variance

[1] Note that, after obtaining the best predictor $\widehat{Y}$, the best estimator $\hat{f}$ for f can be given by taking a sample function from $\widehat{Y}$, namely $\hat{f}(\cdot) = \widehat{Y}(\cdot, \omega)$ for some $\omega \in \Omega$. Please see Section 3.1.2 for more details.

term remains. Then, the task of finding the **best linear unbiased predictor** (BLUP) (Stein, 1999) becomes the minimization of the variance:

$$R(\widehat{Y}, Y) = \mathrm{Var}\{\boldsymbol{\alpha}^\top \boldsymbol{\psi} + \alpha_0 - Y(\mathbf{x})\} = \sigma^2 + \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} - 2\mathbf{k}^\top \boldsymbol{\alpha}.$$

This is a convex optimization task ($\mathbf{K}$ is positive definite) with equality constraints:

$$\begin{aligned} \underset{\boldsymbol{\alpha}\in\mathbb{R}^n}{\text{minimize}} \quad & \sigma^2 + \boldsymbol{\alpha}^\top \mathbf{K}\boldsymbol{\alpha} - 2\mathbf{k}^\top\boldsymbol{\alpha} \\ \text{subject to} \quad & \mathbf{B}^\top \boldsymbol{\alpha} = \mathbf{b}. \end{aligned} \tag{3.7}$$

This optimization problem can be solved using *Lagrange Multipliers.* The first order condition of optimality is (Boyd and Vandenberghe, 2004):

$$\begin{bmatrix} \mathbf{K} & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\mathbf{k} \\ \mathbf{b} \end{bmatrix},$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{p+1}$ is the *dual* variable and $\mathbf{O}$ represents the matrix of zeros. Solving this linear system, we have

$$\begin{aligned} \boldsymbol{\alpha}^* &= \mathbf{K}^{-1}(\mathbf{k} - \mathbf{B}\boldsymbol{\lambda}^*) \\ \boldsymbol{\lambda}^* &= \left(\mathbf{B}^\top \mathbf{K}^{-1}\mathbf{B}\right)^{-1} \left(\mathbf{B}^\top \mathbf{K}^{-1}\mathbf{k} - \mathbf{b}\right). \end{aligned}$$

Due to the convexity of this problem, $\boldsymbol{\alpha}^*$ is also sufficient to be the minimizer of Problem (3.7) (Nocedal and Wright, 2000). Plugging $\boldsymbol{\alpha}^*$ back, we have the **Kriging predictor**:

$$\widehat{Y} = \left[\mathbf{k} - \mathbf{B}\left(\mathbf{B}^\top \mathbf{K}^{-1}\mathbf{B}\right)^{-1}\left(\mathbf{B}^\top\mathbf{K}^{-1}\mathbf{k} - \mathbf{b}\right)\right]^\top \mathbf{K}^{-1}\boldsymbol{\psi}. \tag{3.8}$$

To approximate the target function f, it is straightforward to take a sample function from $\widehat{Y}$:

$$\hat{f} = \mathbf{b}^\top \left[\left(\mathbf{B}^\top\mathbf{K}^{-1}\mathbf{B}\right)^{-1}\mathbf{B}^\top\mathbf{K}^{-1}\mathbf{y}\right] + \mathbf{k}^\top\mathbf{K}^{-1}\left\{\mathbf{y} - \mathbf{B}\left[\left(\mathbf{B}^\top\mathbf{K}^{-1}\mathbf{B}\right)^{-1}\mathbf{B}^\top\mathbf{K}^{-1}\mathbf{y}\right]\right\},$$

which is achieved by substituting the realization $\mathbf{y} = \boldsymbol{\psi}(\omega)$ into Eq. (3.8) and rearranging the terms. It is important to observe that $\hat{\boldsymbol{\beta}} := \left(\mathbf{B}^\top\mathbf{K}^{-1}\mathbf{B}\right)^{-1}\mathbf{B}^\top\mathbf{K}^{-1}\mathbf{y}$ is exactly the **Generalized Least Squares** (GLS) (Rao, Toutenburg, Shalabh, and Heumann, Rao et al.) estimate of $\boldsymbol{\beta}$ in the following sense. The trend function $t = \mathbf{b}^\top\boldsymbol{\beta}$ is treated as the regression function and Z is the stationary error process, whose second-order information (auto-covariance) is known. Then, the *best linear unbiased estimator* (BLUE) of $\boldsymbol{\beta}$ is $\hat{\boldsymbol{\beta}}$. Note that, 1) compared to Kriging, GLS considers $\mathbf{b}^\top\boldsymbol{\beta}$ as the predictor while in Kriging the counterpart is $\mathbf{b}^\top\boldsymbol{\beta} + Z$ and 2) the expression of $\hat{f}$ can also be derived in a much simpler way by first estimating $\boldsymbol{\beta}$ using the GLS formula and then predicting process Z on the residuals $\mathbf{y} - \mathbf{B}\hat{\boldsymbol{\beta}}$ (cf. Eq. (3.13)). However, this approach requires the complete specification of the auto-covariance/kernel and thus is erroneous when hyper-parameters σ^2 and $\boldsymbol{\theta}$ of the kernel function are subject to estimation. Taking the compact notation $\hat{\boldsymbol{\beta}}$, the function approximation is re-written as:

$$\hat{f} = \mathbf{b}^\top \hat{\boldsymbol{\beta}} + \mathbf{k}^\top \mathbf{K}^{-1} \left(\mathbf{y} - \mathbf{B}\hat{\boldsymbol{\beta}}\right). \tag{3.9}$$

In addition, it is also possible to give the covariance of the predictor:

$$\begin{aligned} &\operatorname{Cov}\left\{\widehat{Y}(\mathbf{x}), \widehat{Y}(\mathbf{x}')\right\} \\ &= \left[\mathbf{k} - \mathbf{B}\left(\mathbf{B}^\top\mathbf{K}^{-1}\mathbf{B}\right)^{-1}\left(\mathbf{B}^\top\mathbf{K}^{-1}\mathbf{k} - \mathbf{b}\right)\right]^\top \mathbf{K}^{-1}\left(\mathbf{K} + \mathbf{B}\mathbf{M}\mathbf{B}^\top\right)\mathbf{K}^{-1} \\ &\left[\mathbf{k}' - \mathbf{B}\left(\mathbf{B}^\top\mathbf{K}^{-1}\mathbf{B}\right)^{-1}\left(\mathbf{B}^\top\mathbf{K}^{-1}\mathbf{k}' - \mathbf{b}'\right)\right] - \mathbf{b}^\top\mathbf{M}\mathbf{b}', \end{aligned} \tag{3.10}$$

where $\mathbf{M} = \boldsymbol{\beta}\boldsymbol{\beta}^\top, \mathbf{b}' = \mathbf{b}(\mathbf{x}')$. Note that this covariance depends on the unknown parameter $\boldsymbol{\beta}$. When the kernel is completely specified, $\boldsymbol{\beta}$ can be substituted by its GLS estimate $\hat{\boldsymbol{\beta}}$. The minimal MSE of $\widehat{Y}$ can be obtained by putting $\boldsymbol{\alpha}^*$ back to Eq. (3.7) and it is called the **Kriging MSE**:

$$s^2 = \sigma^2 - \mathbf{k}^\top\mathbf{K}^{-1}\mathbf{k} + \left(\mathbf{b} - \mathbf{B}^\top\mathbf{K}^{-1}\mathbf{k}\right)^\top\left(\mathbf{B}^\top\mathbf{K}^{-1}\mathbf{B}\right)^{-1}\left(\mathbf{b} - \mathbf{B}^\top\mathbf{K}^{-1}\mathbf{k}\right). \tag{3.11}$$

Note that s^2 is the not the variance of the predictor $\widehat{Y}$. In addition, $s = \sqrt{s^2}$ shall be called *Kriging Root Mean Squared Error* (Kriging RMSE).

Remark. 1) In some literatures (den Hertog et al., 2006), s^2 is also called Kriging variance. When using this terminology, s^2 should not be confused with the stationary variance σ^2 of the process Z or the variance of the Kriging predictor. 2) It is important to point out that the MSE s^2 quantifies the uncertainty about predicting the stochastic process Y. It, however, does not directly measure the accuracy of the function approximation, namely to which degree $\hat{f}$ is close to f. To see how the approximation accuracy is related to s^2, please check Section 3.1.2.

The prediction residuals at different locations are correlated. It is possible to calculate the *covariance* among the residuals (please do not confuse with the covariance of the predictor defined in Eq. (3.10)):

$$\begin{aligned} &\operatorname{Cov}\left\{\left(\widehat{Y}(\mathbf{x}) - Y(\mathbf{x})\right)\left(\widehat{Y}(\mathbf{x}') - Y(\mathbf{x}')\right)\right\} \\ &= k(\mathbf{x}, \mathbf{x}') - \mathbf{k}^\top\mathbf{K}^{-1}\mathbf{k}' + (\mathbf{b} - \mathbf{B}^\top\mathbf{K}^{-1}\mathbf{k})^\top(\mathbf{B}^\top\mathbf{K}^{-1}\mathbf{B})^{-1}(\mathbf{b} - \mathbf{B}^\top\mathbf{K}^{-1}\mathbf{k}'), \end{aligned} \tag{3.12}$$

where $\mathbf{k} = \mathbf{k}(\mathbf{x}), \mathbf{k}' = \mathbf{k}(\mathbf{x}')$. It is straightforward to verify this covariance function is positive-definite.

Known trend function The discussion so far can be greatly simplified if the trend function is completely provided prior to the modeling. In principle, the trend effect can be subtracted from the random vector $\boldsymbol{\psi}$:

$$\boldsymbol{\psi}' = \boldsymbol{\psi} - \mathbf{B}\boldsymbol{\beta} = (Z(\mathbf{x}_1), Z(\mathbf{x}_2), \ldots, Z(\mathbf{x}_n))^\top$$

And the latent[1] realization of $\boldsymbol{\psi}'$ is: $\mathbf{z} = \mathbf{y} - \mathbf{B}\boldsymbol{\beta}$. It is then sufficient to search for the optimal linear predictor of Z. In addition, due to the stationarity assumption on Z, any linear predictor $\boldsymbol{\alpha}^\top\boldsymbol{\psi}'$ is unbiased. Therefore, the **best linear predictor** (BLP) (Stein, 1999) suffices for our aim. It is the minimizer of the unconstrained risk function (cf. Eq. (3.7)):

$$R(\boldsymbol{\alpha}^\top\boldsymbol{\psi}', Z) = \sigma^2 + \boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha} - 2\mathbf{k}^\top\boldsymbol{\alpha}.$$

The optimal coefficients are $\boldsymbol{\alpha}^* = \mathbf{K}^{-1}\mathbf{k}$ and the BLP of process Z is $\boldsymbol{\alpha}^{*\top}\boldsymbol{\psi}'$. The best linear predictor of Y is obtained by adding the trend function back to the BLP of Z:

$$\widehat{Y} = \mathbf{b}^\top\boldsymbol{\beta} + \mathbf{k}^\top\mathbf{K}^{-1}\left(\boldsymbol{\psi} - \mathbf{B}\boldsymbol{\beta}\right) \tag{3.13}$$

$$s^2 = \sigma^2 - \mathbf{k}^\top\mathbf{K}^{-1}\mathbf{k} \tag{3.14}$$

$$\mathrm{Var}\left\{\widehat{Y}(\mathbf{x})\right\} = \mathbf{k}^\top\mathbf{K}^{-1}\mathbf{k} \tag{3.15}$$

$$\mathrm{Cov}\left\{\widehat{Y}(\mathbf{x}), \widehat{Y}(\mathbf{x}')\right\} = \mathbf{k}^\top\mathbf{K}^{-1}\mathbf{k}' \tag{3.16}$$

The extreme of this treatment is to set $\boldsymbol{\beta}$ to zero and it is called *Simple Kriging*. In this case, $\widehat{Y} = \mathbf{k}^\top\mathbf{K}^{-1}\boldsymbol{\psi}$ and its variance and MSE are the same as Eq. (3.15) and (3.14).

[1] We use the term "latent" here as Z is not directly observable.

Discussion 1. It seems a daunting task to select an appropriate trend function. For local interpolations, theoretically it is known that BLPs exhibit the same performance asymptotically with BLUPs, even if the trend is zero ($\boldsymbol{\beta} = \mathbf{0}$) (Stein, 1999). For such modeling tasks, it is sufficient to set the trend function to zero. For the extrapolation, the Kriging estimator regresses back to the trend when the location is weakly correlated to most of the data points. Thus, choosing a proper trend function is necessary for the extrapolation purpose (Journel and Rossi, 1989). Generally, Universal Kriging is recommended for this scenario (Journel and Rossi, 1989) if no prior knowledge is available. However, thorough empirical/theoretical analyses are necessary before putting it as a conclusion. In addition, as will be shown later, the predictor of Simple Kriging is an element of the Hilbert space $\mathcal{H}$ induced by the kernel function. It would be interesting to investigate if polynomial trend functions can be fully expressed in $\mathcal{H}$. The incorporation of the trend function would be unnecessary if it is also an element of $\mathcal{H}$.

Noisy observations and Kriging nugget In practice, it is very likely that the observed response variable contains random measurement noises. Therefore it is, in general, helpful to consider the following *data generation process*:

$$\widetilde{Y} = Y + \varepsilon, \tag{3.17}$$

where $\{\varepsilon(\mathbf{x}) : \mathbf{x} \in \mathrm{S}\}$ is a white noise process (e.g., Gaussian white noise) that is independent from Y and has stationary variance $\sigma_n^2 < \infty$. Formally, ε is specified as:

$$\forall \mathbf{x}, \mathbf{x}' \in \mathrm{S}, \quad \mathbb{E}\varepsilon(\mathbf{x}) = 0, \quad \mathrm{Cov}\{\varepsilon(\mathbf{x}), \varepsilon(\mathbf{x}')\} = \sigma_n^2 \mathbb{1}_{\{\mathbf{x}\}}(\mathbf{x}'), \quad \varepsilon(\mathbf{x}) \perp\!\!\!\perp Y(\mathbf{x}').$$

Here $\mathbb{1}_{\{\mathbf{x}\}}$ is the characteristic function (or indicator function) and $\perp\!\!\!\perp$ denotes the statistical independence. It is important to point out that the goal is still to predict process Y. Under this setting, the task of predicting Y becomes a *nonparametric regression* task, in which the regression function $\hat{f}$ admits a nonparametric form. Again, consider the random vector $\widetilde{\boldsymbol{\psi}} = (\widetilde{Y}(\mathbf{x}_1), \widetilde{Y}(\mathbf{x}_2), \ldots, \widetilde{Y}(\mathbf{x}_n))^\top$ and its realizations $\widetilde{\mathbf{y}} = \widetilde{\boldsymbol{\psi}}(\omega), \omega \in \Omega$. For the sake of brevity, only simple Kriging ($\boldsymbol{\beta}$ is zero) is considered here. By minimizing the risk function (cf. Eq. (3.7)),

$$R(\widehat{Y}, Y) = \mathrm{Var}\{\boldsymbol{\alpha}^\top \widetilde{\boldsymbol{\psi}} + \alpha_0 - Y(\mathbf{x})\} = \sigma^2 + \boldsymbol{\alpha}^\top \left(\mathbf{K} + \sigma_n^2 \mathbf{I}\right) \boldsymbol{\alpha} - 2\mathbf{k}^\top \boldsymbol{\alpha},$$

the Kriging estimator under noisy observations is:

$$\widehat{Y} = \mathbf{k}^\top \left(\mathbf{K} + \sigma_n^2 \mathbf{I}\right)^{-1} \widetilde{\boldsymbol{\psi}}. \tag{3.18}$$

$$s^2 = \sigma^2 - \mathbf{k}^\top \left(\mathbf{K} + \sigma_n^2 \mathbf{I}\right)^{-1} \mathbf{k}. \tag{3.19}$$

The noise variance σ_n^2 is also known as the **Kriging nugget** or **nugget effect** (Cressie, 2015). Historically, the Kriging nugget is introduced with the so-called *Semivariogram* in the geostatistics literature. The semivariogram is defined as half the variance of the differences between observations at two locations: $\gamma(\mathbf{x}, \mathbf{x}') = \frac{1}{2}\mathbb{E}\{Y(\mathbf{x}) - Y(\mathbf{x}')\}^2$. As with the kernel function, the semivariogram is an alternative quantification of the auto-correlation (spatial dependency) on process Y. The nugget effect is defined to be the amount of the jump of the semivariogram at the origin (Fig. 3.1) and can be attributed to measurement errors due to the inherent imprecision in measurement devices. Consider the noisy process $\widetilde{Y}$, its semivariogram is,

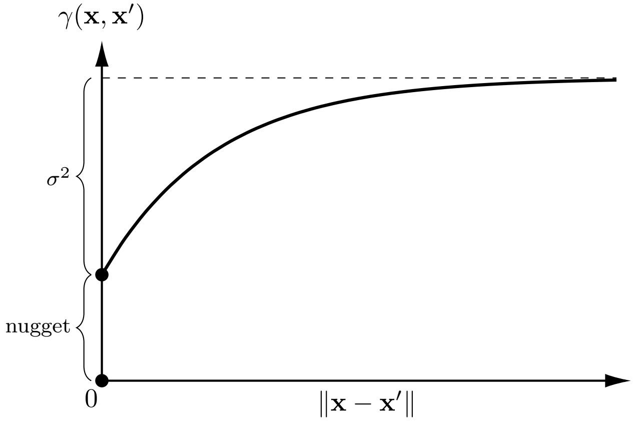


Figure 3.1: Illustration on the semivariogram and Kriging nugget.

$$\begin{aligned}\widetilde{\gamma}(\mathbf{x}, \mathbf{x}') &= \frac{1}{2}\mathbb{E}\left\{\widetilde{Y}(\mathbf{x}) - \widetilde{Y}(\mathbf{x}')\right\}^2 = \frac{1}{2}\mathbb{E}\left\{Y(\mathbf{x}) - Y(\mathbf{x}')\right\}^2 + \sigma_n^2 \\ &= \gamma(\mathbf{x}, \mathbf{x}') + \sigma_n^2.\end{aligned}$$

The semivariogram $\widetilde{\gamma}$ in the noisy case is translated upwards from the noiseless case γ by an amount of σ_n^2. The correspondence between the noise variance and Kriging nugget is clearly seen. Moreover, the nugget effect is sometimes referred as *nugget variance* (Webster and Oliver, 2007).

Note that, the nugget effect is useful even when no measurement error is present, e.g., in computer experiments. It can help relaxing the conditional number of the covariance matrix $\mathbf{K}$ when it gets ill-conditioned (Andrianakis and Challenor, 2012). Let $\widetilde{\mathbf{K}} = \mathbf{K} + \sigma_n^2 \mathbf{I}$ be the covariance matrix under the noisy assumption. It is then obvious that the eigenvalue $\tilde{\lambda}$ of $\widetilde{\mathbf{K}}$ admits the relation: $\tilde{\lambda} = \lambda + \sigma_n^2$, where λ is the eigenvalue of $\mathbf{K}$. Consequently, the condition number κ of $\widetilde{\mathbf{K}}$ is smaller than that of $\mathbf{K}$:

$$\kappa(\widetilde{\mathbf{K}}) = \frac{|\lambda_{\max} + \sigma_n^2|}{|\lambda_{\min} + \sigma_n^2|} < \frac{|\lambda_{\max}|}{|\lambda_{\min}|} = \kappa(\mathbf{K}).$$

Numerically, as the condition number increases, the covariance matrix becomes practically not invertible and therefore introducing the Kriging nugget can avoid numerical issues that are frequently encountered in the hyper-parameter estimation (Ababou et al., 1994).

3.1.2 Reproducing Kernel Hilbert Space

Here we shall take a different point of view on BLUP, namely from the Hilbert space associated with stochastic process Y. To simplify our discussions here, Y is assumed to have a zero mean and correspondingly the Kriging predictor is $\widehat{Y} = \mathbf{k}^\top \mathbf{K}^{-1} \boldsymbol{\psi}$ (obtained by setting $\boldsymbol{\beta}$ to zero in Eq. (3.13)). Moreover, the index set S is assumed to be a *separable space*. In addition, the covariance vector $\mathbf{k}$ is treated as a function from S to $\mathbb{R}^n$ and is denoted as $\mathbf{k}(\mathbf{x})$. More precisely, the approach posed in Section 3.1.1 is to predict a random variable $Y(\mathbf{x}, \cdot)$ using a linear combination of some other random variables on the process:

$$\widehat{Y}(\mathbf{x}, \cdot) = \sum_{i=1}^{n} \alpha_i(\mathbf{x}) Y(\mathbf{x}_i, \cdot), \quad n \in \mathbb{N}, \quad \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathrm{S}.$$

The optimal coefficient $\boldsymbol{\alpha}(\mathbf{x}) = \mathbf{K}^{-1}\mathbf{k}(\mathbf{x})$ is a function of $\mathbf{x}$ and is obtained by minimizing the risk function (Eq. (3.6)) as before. Note that the same prediction approach is applied for every $\mathbf{x} \in \mathrm{S}$, meaning that a predictor of the process Y is obtained:

$$\widehat{Y} = \left\{ \sum_{i=1}^{n} \alpha_i(\mathbf{x}) Y(\mathbf{x}_i, \cdot) : n \in \mathbb{N},\ \mathbf{x}, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathrm{S} \right\}. \tag{3.20}$$

We shall call $\widehat{Y}$ the **Kriging predictor**. The rationale is: $\widehat{Y}$ optimally predicts an unknown process Y only using partial information of its own. If we observe a sample function of Y partially[1], namely $\mathbf{y} = (Y(\mathbf{x}_1, \omega), Y(\mathbf{x}_2, \omega), \ldots, Y(\mathbf{x}_n, \omega))^\top$ for some $\omega \in \Omega$, it is possible to interpolate this sample function by plugging $\mathbf{y}$ into $\widehat{Y}$ (Eq. (3.20)):

$$\begin{aligned}\hat{f}(\cdot) = \widehat{Y}(\cdot, \omega) &= \sum_{i=1}^{n} \alpha_i(\cdot) Y(\mathbf{x}_i, \omega) = \mathbf{y}^\top \mathbf{K}^{-1} \mathbf{k}(\cdot) \\ &= \sum_{i=1}^{n} \xi_i k(\cdot, \mathbf{x}_i), \quad \boldsymbol{\xi} = \mathbf{K}^{-1}\mathbf{y}. \end{aligned} \tag{3.21}$$

Being a sample function from the predictor $\widehat{Y}$, $\hat{f}$ approximates f (cf. Eq. (3.9)). In short, we shall show that the function form of Eq. (3.21) is in the **Reproducing Kernel Hilbert Space** (RKHS) attached to process Y. Given a positive definite kernel $k(\cdot, \cdot)$ on S, there is a unique Hilbert space of functions: $\mathrm{S} \to \mathbb{R}$ for which k is a reproducing kernel (Moore-Aronszajn theorem (Aronszajn, 1950)). This space $\mathcal{H}$ is the completion of the following linear space $\mathcal{H}_0$:

$$\mathcal{H}_0 = \left\{ \sum_{i=1}^{n} c_i k(\cdot, \mathbf{x}_i) : n \in \mathbb{N},\ c_1, c_2, \ldots, c_n \in \mathbb{R},\ \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathrm{S} \right\}.$$

The completion is conducted with respect to the RKHS norm $\|\cdot\|_{\mathcal{H}}$ that is induced by the inner product,

$$\left\langle \sum_{i=1}^{m} c_i k(\cdot, \mathbf{x}_i), \sum_{j=1}^{n} c'_j k(\cdot, \mathbf{x}_j) \right\rangle_{\mathcal{H}} = \sum_{i=1}^{m} \sum_{j=1}^{n} c_i c'_j k(\mathbf{x}_i, \mathbf{x}_j). \tag{3.22}$$

The function in $\mathcal{H}$ has the form: $f(\cdot) = \sum_{i=1}^{\infty} a_i k(\cdot, \mathbf{x}_i)$, where $\sum_{i=1}^{\infty} a_i^2 k(\mathbf{x}_i, \mathbf{x}_i) < \infty$. It is then obvious to see that the Kriging estimator $\hat{f}(\cdot)$ is an element of $\mathcal{H}$. Equivalently, the space of estimator $\hat{f}$ is the set of all the sample functions of $\widehat{Y}$, namely $\left\{\widehat{Y}(\cdot, \omega) : \omega \in \Omega\right\} \subset \mathcal{H}$. The natural question is: how does the target function f related to $\hat{f}$ and $\mathcal{H}$ in general? Recall the assumption of Kriging: f is a realization of the process Y, or formally $f \in \mathrm{F}$, $\mathrm{F} \coloneqq \{Y(\cdot, \omega) : \omega \in \Omega\}$ (all sample functions of Y). Firstly, we will show that $\mathcal{H}$ is generally "smaller" than F. It is possible to construct a *surjection* from F to $\mathcal{H}$:

$$Y(\cdot, \omega) \mapsto \sum_{i=1}^{n} \tau_i Y(\mathbf{x}_i, \omega) k(\cdot, \mathbf{x}_i), \quad \tau_i \in \mathbb{R}.$$

[1] It is important to note that even countably many observations are partial information about the sample function because its domain S is separable.

It is obvious that for every function $\sum_{i=1}^{n} c_i k(\cdot, \mathbf{x}_i) \in \mathcal{H}$, there always exist $\tau_i \in \mathbb{R}$ and $\omega \in \Omega$ such that $\tau_i Y(\mathbf{x}_i, \omega) = c_i$. Thus, this mapping is surjective. However, it does not admit an inverse: $\{Y(\mathbf{x}_i, \omega)\}_i$ can be mapped back to infinitely many sample functions in F. Secondly, it is possible to quantify the difference between f and $\hat{f}$ using the supremum norm, $\left\|f - \hat{f}\right\|_\infty = \sup_{\mathbf{x} \in \mathrm{S}} \left\{\left|f(\mathbf{x}) - \hat{f}(\mathbf{x})\right|\right\}$. It is not hard to verify the following condition,

$$\left\|f - \hat{f}\right\|_\infty \leq \sup_{\omega \in \Omega} \left\{\left|Y(\cdot, \omega) - \widehat{Y}(\cdot, \omega)\right|\right\}.$$

However, it is not straightforward to build a linkage between $\left\|f - \hat{f}\right\|_\infty$ and the Kriging MSE s^2 (cf. Eq. (3.11)) based on this condition. Alternatively, such a relation can be established point-wisely on f.

Theorem 3.1 (Approximation Error Bound). *Let $\widehat{Y}$ be the BLUP of stochastic process Y. The MSE of $\widehat{Y}$ is $s^2 = \mathbb{E}\{Y(\mathbf{x}) - \widehat{Y}(\mathbf{x})\}^2$. Assume the target function $f : \mathrm{S} \to \mathbb{R}$ is a sample function of Y and it is approximated by a sample function of the BLUP: $\hat{f}(\cdot) = \widehat{Y}(\cdot, \omega), \omega \in \Omega$. Then for every point $\mathbf{x} \in \mathrm{S}$, the approximation error is bounded from above:*

$$\left|\hat{f}(\mathbf{x}) - f(\mathbf{x})\right| \leq \sqrt{\frac{s^2}{C}},$$

where

$$C = \int_{\mathbb{R}} \Pr\left(|Y(\mathbf{x})| > |f(\mathbf{x})| \,\middle|\, \widehat{Y}(\mathbf{x}) = u\right) p_{\widehat{Y}(\mathbf{x})}(u)\, \mathrm{d}u.$$

Proof. Define a random variable $R = |Y(\mathbf{x}) - \widehat{Y}(\mathbf{x})|$. $r = |f(\mathbf{x}) - \hat{f}(\mathbf{x})|$ is a realization of R. According to Markov's inequality, we have,

$$\Pr(R \geq r) \leq \frac{\mathbb{E}R^2}{r^2}. \tag{3.23}$$

Note that $s^2 = \mathbb{E}R^2$ and $r^2 = |f(\mathbf{x}) - \hat{f}(\mathbf{x})|^2$. Now, we expand the probability on the left-hand-side of the inequality:

$$\begin{aligned}\Pr(R \geq r) &= \int_{\mathbb{R}} \Pr\left(|Y(\mathbf{x}) - \hat{f}(\mathbf{x})| > |f(\mathbf{x}) - \hat{f}(\mathbf{x})| \,\middle|\, \widehat{Y}(\mathbf{x}) = u\right) p_{\widehat{Y}(\mathbf{x})}(u)\, \mathrm{d}u \\ &\geq \int_{\mathbb{R}} \Pr\left(|Y(\mathbf{x})| > |f(\mathbf{x})| \,\middle|\, \widehat{Y}(\mathbf{x}) = u\right) p_{\widehat{Y}(\mathbf{x})}(u)\, \mathrm{d}u.\end{aligned}$$

Combining this inequality with Eq. (3.23), we have $r^2 \leq s^2/C$. □

Remark. The linkage between the approximation error on f and the MSE of BLUPs is clearly seen from this theorem: reducing the MSE s^2 leads to a more precise function approximation, which is typically achieved by adding more data points/observations. The other factor C can be interpreted as the conditional probability $\Pr\left(|Y(\mathbf{x})| > |f(\mathbf{x})| \;\middle|\; \widehat{Y}(\mathbf{x}) = u\right)$, averaged over all possible predictions. Note that the smaller this conditional probability is (and thus the error bound is higher), it is less likely that $f(\mathbf{x})$ is a sample from $Y(\mathbf{x})$. This means the stochastic process Y tends to be **mis-specified** for f.

RKHS provides us another point of view on Kriging/GPR. In the following, it is shown that the same result in Eq. (3.13) can be obtained using the well-known *representer theorem* (Schölkopf et al., 2001), which gives the representer form of the solution to the regularized optimization problem in $\mathcal{H}$. We shall illustrate this theorem first and then build an alternative derivation of $\hat{f}$ based on it.

Theorem 3.2 (Representer Theorem)**.** *Let* S *be a nonempty set and* $k : \mathrm{S} \times \mathrm{S} \to \mathbb{R}$ *be a positive-definite kernel with corresponding reproducing kernel Hilbert space* $\mathcal{H}$*. The RKHS norm* $\|\cdot\|_{\mathcal{H}}$ *is induced from the inner product in Eq.* (3.22)*. Given a training sample* $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \in \mathrm{S} \times \mathbb{R}$*, a strictly monotonically increasing function* $g\colon [0, \infty) \to \mathbb{R}$*, and an arbitrary empirical risk function* R *of* $\{h(\mathbf{x}_i), y_i\}_{i=1}^n$ *and let* $h^* : \mathrm{S} \to \mathbb{R}$ *be the optimum of the regularized minimization problem:*

$$h^* = \operatorname*{arg\,min}_{h \in \mathcal{H}} R\left(\{h(\mathbf{x}_i), y_i\}_{i=1}^n\right) + g(\|h\|_{\mathcal{H}}),$$

then h^* *is represented as:*

$$h^*(\cdot) = \sum_{i=1}^n c_i k(\cdot, \mathbf{x}_i), \quad c_1, c_2, \ldots, c_n \in \mathbb{R}.$$

Proof. See Schölkopf et al. (2001). □

The representer form of h^* is exactly as Eq. (3.13), suggesting that the Kriging estimator can also be considered as the optimal function form that minimizes *empirical* risk. Furthermore, we will illustrate that under certain specifications, the Kriging coefficients in the noisy setting (Eq. (3.18)) can be obtained using this theorem.

Corollary 3.1. *Assume all the settings in Theorem 3.2, noisy observations* $\widetilde{\mathbf{y}}$ *generated from Eq.* (3.17) *and the following specifications: the empirical risk function* $R\left(\{\hat{f}(\mathbf{x}_i), \widetilde{y}_i\}_{i=1}^n\right) = \sum_{i=1}^n \left(\hat{f}(\mathbf{x}_i) - \widetilde{y}_i\right)^2$ *and* $g(\|\cdot\|_{\mathcal{H}}) = \sigma_n^2 \|\cdot\|_{\mathcal{H}}^2$*, then the coefficients* c_i *in Theorem 3.2 are given by* $\mathbf{c} = \left(\mathbf{K} + \sigma_n^2 \mathbf{I}\right)^{-1} \widetilde{\mathbf{y}}$ *and*

$$\hat{f}(\cdot) = \sum_{i=1}^n c_i k(\cdot, \mathbf{x}_i) = \widetilde{\mathbf{y}}^\top \left(\mathbf{K} + \sigma_n^2 \mathbf{I}\right)^{-1} \mathbf{k}(\cdot).$$

Proof. According to the representer theorem, $\hat{f}$ takes the form $\hat{f}(\cdot) = \sum_{i=1}^n c_i k(\cdot, \mathbf{x}_i)$. Then all predictions from $\hat{f}$ can be denoted as $(\hat{f}(\mathbf{x}_1), \ldots, \hat{f}(\mathbf{x}_n))^\top = \mathbf{Kc}$. Then the regularized minimization problem becomes:

$$\underset{\mathbf{c}\in\mathbb{R}^n}{\text{minimize}} \left\|\mathbf{Kc} - \widetilde{\mathbf{y}}\right\|^2 + \sigma_n^2 \mathbf{c}^\top \mathbf{Kc}. \tag{3.24}$$

The optimality condition of this problem is:

$$\frac{\partial}{\partial \mathbf{c}} \left(\left\|\mathbf{Kc} - \widetilde{\mathbf{y}}\right\|^2 + \sigma_n^2 \mathbf{c}^\top \mathbf{Kc} \right) = \mathbf{0}. \tag{3.25}$$

The solution of $\mathbf{c}$ results from this condition. □

This result can be interpreted as follows. Firstly, note that Problem (3.24) is equivalent to the following constrained convex optimization problem:

$$\begin{aligned} &\underset{\mathbf{c}\in\mathbb{R}^n}{\text{minimize}} && \left\|\mathbf{Kc} - \widetilde{\mathbf{y}}\right\|^2 \\ &\text{subject to} && \mathbf{c}^\top \mathbf{Kc} \leq t, \end{aligned} \tag{3.26}$$

where $t = \widetilde{\mathbf{y}}^\top \left(\mathbf{K} + \sigma_n^2 \mathbf{I}\right)^{-1} \mathbf{K} \left(\mathbf{K} + \sigma_n^2 \mathbf{I}\right)^{-1} \widetilde{\mathbf{y}}$. To see the equivalence, the *Karush-Kuhn-Tucker* conditions (KKT) (Boyd and Vandenberghe, 2004) of Problem (3.26) are,

$$\begin{aligned} \frac{\partial}{\partial \mathbf{c}} \left(\left\|\mathbf{Kc} - \widetilde{\mathbf{y}}\right\|^2 \right) + \eta \frac{\partial}{\partial \mathbf{c}} \left(\mathbf{c}^\top \mathbf{Kc} - t \right) &= \mathbf{0} \\ \eta \left(\mathbf{c}^\top \mathbf{Kc} - t \right) &= 0 \\ \mathbf{c}^\top \mathbf{Kc} - t &\leq 0 \\ \eta &\geq 0 \end{aligned} \tag{3.27}$$

The conditions above are also necessary because Slater's condition (Slater, 2014) obviously holds on Problem (3.26). It is not hard to verify that any solution of condition (3.25) is also a solution of conditions (3.27) and vice versa. Consider the convex constraint $\mathbf{c}^\top \mathbf{Kc} \leq t$. The LHS of it is the RKHS norm $\left\|\hat{f}\right\|_{\mathcal{H}}$ of the estimator $\hat{f}(\cdot) = \sum_{i=1}^n c_i k(\cdot, \mathbf{x}_i)$. It means that "complexity" of the estimator $\hat{f}$ should be smaller than a threshold t when minimizing the empirical risk. To understand the choice of threshold t, please consider the prediction of the data generation process process $\widetilde{Y} = Y + \varepsilon$ using observations $\widetilde{\mathbf{y}}$ (instead of predicting Y). The Kriging estimator $\widetilde{f}_{\text{est}} \in \mathcal{H}$ is obtained by applying Eq. (3.13) to the overall process $\widetilde{Y}$, whose covariance function is $\widetilde{k}(\cdot,\cdot) = k(\cdot,\cdot) + \sigma_n^2 \mathbb{1}_{\{\cdot\}}(\cdot)$:

$$\widetilde{f}_{\text{est}}(\cdot) = \sum_{i=1}^n \widetilde{\alpha}_i \widetilde{k}(\cdot, \mathbf{x}_i), \quad \widetilde{\boldsymbol{\alpha}} = \left(\mathbf{K} + \sigma_n^2 \mathbf{I}\right)^{-1} \widetilde{\mathbf{y}}.$$

Note that $\widetilde{f}_{\text{est}}$ is an element of the RKHS $\widetilde{\mathcal{H}}$ induced by kernel $\widetilde{k}$ and its norm $\left\|\widetilde{f}_{\text{est}}\right\|_{\widetilde{\mathcal{H}}} = \widetilde{\mathbf{y}}^{\top}\left(\mathbf{K}+\sigma_n^2\mathbf{I}\right)^{-1}\widetilde{\mathbf{y}}$. It is clear that

$$
\begin{aligned}
\left\|\hat{f}\right\|_{\mathcal{H}} \leq t &= \widetilde{\mathbf{y}}^{\top}\left(\mathbf{K}+\sigma_n^2\mathbf{I}\right)^{-1}\mathbf{K}\left(\mathbf{K}+\sigma_n^2\mathbf{I}\right)^{-1}\widetilde{\mathbf{y}} \\
&\leq \widetilde{\mathbf{y}}^{\top}\left(\mathbf{K}+\sigma_n^2\mathbf{I}\right)^{-1}\widetilde{\mathbf{y}} \\
&= \left\|\widetilde{f}_{\text{est}}\right\|_{\widetilde{\mathcal{H}}},
\end{aligned}
$$

which means the estimator of the component f from the noisy function $\widetilde{f}$ should not be more complex than the estimator of $\widetilde{f}$. In summary, the Kriging estimator (Eq. (3.18)) under noisy observations is the solution to the following problem:

$$
\begin{aligned}
\underset{\hat{f}\in\mathcal{H}}{\text{minimize}} \quad & \sum_{i=1}^{n}\left(\hat{f}(\mathbf{x}_i)-\widetilde{y}_i\right)^2 \\
\text{subject to} \quad & \left\|\hat{f}\right\|_{\mathcal{H}} \leq \left\|\widetilde{f}_{\text{est}}\right\|_{\widetilde{\mathcal{H}}}.
\end{aligned}
$$

3.1.3 Bayesian Inference

Known trend function It is possible to give an alternative derivation of the Kriging estimator, using *Bayesian statistics.* Consider again the random vector $\boldsymbol{\psi} = (Y(\mathbf{x}_1), Y(\mathbf{x}_2), \ldots, Y(\mathbf{x}_n))^{\top}, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathrm{S}$ and its realization $\mathbf{y} = \boldsymbol{\psi}(\omega), \omega \in \Omega$. Bayesian inference requires the specification of the prior distribution on Y and the likelihood $p(\boldsymbol{\psi} \mid Y(\mathbf{x}))$. When the trend function is assumed to be known, the *posterior* of $Y(\mathbf{x})$ is

$$
p(Y(\mathbf{x}) \mid \boldsymbol{\psi}) = \frac{p(\boldsymbol{\psi} \mid Y(\mathbf{x}))p(Y(\mathbf{x}))}{p(\boldsymbol{\psi})}. \tag{3.28}
$$

Note that, this posterior probability is the conditional probability $p(Y(\mathbf{x}) \mid \boldsymbol{\psi})$ due to the fact that $Y(\mathbf{x})$ and $\boldsymbol{\psi}$ are taken from the same stochastic process Y in our setting. In some other processes, this conditional probability is even given in their definitions (e.g., Markov process). However, Eq. (3.28) gives a plausible rationale on using the conditional probability for the prediction and attaches a Bayesian interpretation to the Kriging predictor. The most common choice (and perhaps the most natural) on the prior distribution is Gaussian: the stochastic process Y is assumed to be a *Gaussian Process.* In addition to the first- (mean) and second-order (covariance) specifications (Section 3.1), the Gaussian process prior on Y prescribes that random vector $\boldsymbol{\psi}$ is a multivariate Gaussian (See Appendix A for its definition). The following notation is used for a Gaussian process prior with kernel function k:

$$Y \sim t + \mathcal{GP}(0, k(\cdot, \cdot)),$$

where t is the trend function defined in Eq. (3.2) and it is called the *prior mean* function in this section. Note that, trend t is deliberately separated from the centered Gaussian Process $\mathcal{GP}(0, k(\cdot, \cdot))$ because t could admit a stochastic form and the addition of those two terms might not be Gaussian. Recall the basis expansion trend $t = \mathbf{b}^\top \boldsymbol{\beta}$ and $\boldsymbol{\beta}$ is known. It is then straightforward that $Y(\mathbf{x}) \sim \mathcal{N}(\mathbf{b}^\top \boldsymbol{\beta}, \sigma^2)$ and $\boldsymbol{\psi} \sim \boldsymbol{\mathcal{N}}(\mathbf{B}\boldsymbol{\beta}, \mathbf{K})$. Moreover, $Y(\mathbf{x})$ and $\boldsymbol{\psi}$ are jointly Gaussian:

$$\begin{bmatrix} Y(\mathbf{x}) \\ \boldsymbol{\psi} \end{bmatrix} \sim \boldsymbol{\mathcal{N}} \left(\begin{bmatrix} \mathbf{b}^\top \boldsymbol{\beta} \\ \mathbf{B}\boldsymbol{\beta} \end{bmatrix}, \begin{bmatrix} \sigma^2 & \mathbf{k}^\top \\ \mathbf{k} & \mathbf{K} \end{bmatrix} \right).$$

Recall the definition of the covariance vector $\mathbf{k}$ in Section 3.1. Directly applying the conditioning formula (Eq. (A.4)), the conditional distribution $p(Y(\mathbf{x}) \mid \boldsymbol{\psi})$ can be specified

$$Y(\mathbf{x}) \mid \boldsymbol{\psi} \sim \mathbf{b}^\top \boldsymbol{\beta} + \mathbf{k}^\top \mathbf{K}^{-1} (\boldsymbol{\psi} - \mathbf{B}\boldsymbol{\beta}) + \boldsymbol{\mathcal{N}} \left(0, \sigma^2 - \mathbf{k}^\top \mathbf{K}^{-1} \mathbf{k}\right). \tag{3.29}$$

Given this conditional distribution, it is obvious that the best unbiased predictor of Y is the conditional mean, i.e., $\widehat{Y} = \mathbf{b}^\top \boldsymbol{\beta} + \mathbf{k}^\top \mathbf{K}^{-1} (\boldsymbol{\psi} - \mathbf{B}\boldsymbol{\beta})$. The MSE of $\widehat{Y}$ is $s^2 = \mathbb{E}\{\widehat{Y} - Y\}^2 = \sigma^2 - \mathbf{k}^\top \mathbf{K}^{-1} \mathbf{k}$, which is also the conditional variance. Now, as the target function f is assumed to be a sample function of Y and we have observed some values on the target function, the approximation $\hat{f}$ is obtained by replacing $\boldsymbol{\psi}$ by its realization in $\widehat{Y}$ (cf. Eq. (3.21)):

$$\hat{f}(\cdot) = \widehat{Y}(\cdot, \omega) = \mathbf{b}^\top \boldsymbol{\beta} + \mathbf{k}^\top \mathbf{K}^{-1} (\mathbf{y} - \mathbf{B}\boldsymbol{\beta}), \quad \omega \in \Omega.$$

Note that those terms are exactly the same as the Kriging BLP estimator (cf. Eq. (3.13)). Note that, in terms of Bayesian statistics, the posterior mean in Eq. (3.29) can also be considered as a *Maximum a Posterior Probability* (MAP) estimate because the mode coincides with the mean in Gaussian distributions. This result is commonly referred to as **Gaussian Process Regression** (GPR) in the machine learning field (Rasmussen and Williams, 2006). Consequently, the Kriging MSE s^2 is also called **GPR variance** in this thesis.

Remark. In the standard treatment of GPR, there is no need to use the stochastic process Y because the prior Gaussian process is directly imposed on the target function f. In this section, process Y is taken to keep the consistence with the discussion on BLUP/BLP (Section 3.1.1).

Moreover, a posterior Gaussian process is implied by Eq. (3.29), whose mean function is $\widehat{Y}$. To see the covariance structure of the posterior process, consider two locations $\mathbf{x}_1, \mathbf{x}_2 \in \mathrm{S}$ in the query:

$$\begin{bmatrix} Y(\mathbf{x}_1) \\ Y(\mathbf{x}_2) \\ \boldsymbol{\psi} \end{bmatrix} \sim \boldsymbol{\mathcal{N}} \left(\begin{bmatrix} \mathbf{b}_1^\top \boldsymbol{\beta} \\ \mathbf{b}_2^\top \boldsymbol{\beta} \\ \mathbf{B}\boldsymbol{\beta} \end{bmatrix}, \begin{bmatrix} \sigma^2 & k(\mathbf{x}_1, \mathbf{x}_2) & \mathbf{k}_1^\top \\ k(\mathbf{x}_2, \mathbf{x}_1) & \sigma^2 & \mathbf{k}_2^\top \\ \mathbf{k}_1 & \mathbf{k}_2 & \mathbf{K} \end{bmatrix} \right),$$

in which $\mathbf{b}_1 = \mathbf{b}(\mathbf{x}_1), \mathbf{b}_2 = \mathbf{b}(\mathbf{x}_2)$ and $\mathbf{k}_1 = \mathbf{k}(\mathbf{x}_1), \mathbf{k}_2 = \mathbf{k}(\mathbf{x}_2)$. Conditioning on $\boldsymbol{\psi}$ again, we obtain:

$$\begin{bmatrix} Y(\mathbf{x}_1) \\ Y(\mathbf{x}_2) \end{bmatrix} \Bigg| \, \boldsymbol{\psi} \sim \begin{bmatrix} \mathbf{b}_1^\top \boldsymbol{\beta} + \mathbf{k}_1^\top \mathbf{K}^{-1} (\boldsymbol{\psi} - \mathbf{B}\boldsymbol{\beta}) \\ \mathbf{b}_2^\top \boldsymbol{\beta} + \mathbf{k}_2^\top \mathbf{K}^{-1} (\boldsymbol{\psi} - \mathbf{B}\boldsymbol{\beta}) \end{bmatrix}$$
$$+ \boldsymbol{\mathcal{N}} \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma^2 - \mathbf{k}_1^\top \mathbf{K}^{-1} \mathbf{k}_1 & k(\mathbf{x}_1, \mathbf{x}_2) - \mathbf{k}_1^\top \mathbf{K}^{-1} \mathbf{k}_2 \\ k(\mathbf{x}_2, \mathbf{x}_1) - \mathbf{k}_2^\top \mathbf{K}^{-1} \mathbf{k}_1 & \sigma^2 - \mathbf{k}_2^\top \mathbf{K}^{-1} \mathbf{k}_2 \end{bmatrix} \right).$$

In this posterior formulation, it is clear to see that the covariance at two arbitrary locations is expressed in the cross-term of the posterior covariance matrix. Consequently, we give the posterior mean (trend) $\widehat{Y}$ and posterior kernel k':

$$\widehat{Y}(\mathbf{x}) \coloneqq \mathbb{E}\{Y(\mathbf{x}) \mid \boldsymbol{\psi}\} = \mathbf{b}^\top \boldsymbol{\beta} + \mathbf{k}^\top \mathbf{K}^{-1} (\boldsymbol{\psi} - \mathbf{B}\boldsymbol{\beta}) \tag{3.30}$$

$$k'(\mathbf{x}, \mathbf{x}') \coloneqq \mathrm{Cov}\{Y(\mathbf{x}), Y(\mathbf{x}') \mid \boldsymbol{\psi}\} = k(\mathbf{x}, \mathbf{x}') - \mathbf{k}^\top \mathbf{K}^{-1} \mathbf{k}' \tag{3.31}$$

It is straightforward to show that k' is a stationary positive-definite kernel.

Unknown trend function When $\boldsymbol{\beta} \in \mathbb{R}^{p+1}$ is subject to estimation, the most common approach is to use hierarchical Bayesian inference by providing a prior on $\boldsymbol{\beta}$. For example, the Gaussian prior is assumed again $\boldsymbol{\beta} \sim \boldsymbol{\mathcal{N}}(\boldsymbol{\zeta}, \boldsymbol{\Sigma})$, with $\boldsymbol{\beta} \perp\!\!\!\perp Y$. It is important to note that when randomness on $\boldsymbol{\beta}$ is introduced the process Y is not necessarily Gaussian any longer. However, the conditional distribution/process on $\boldsymbol{\beta}$, e.g., $p(Y(\mathbf{x}) \mid \boldsymbol{\beta}, \boldsymbol{\psi})$ is still Gaussian. The posterior distribution of $\boldsymbol{\beta}$ is (Stein, 1999),

$$\begin{aligned} p(\boldsymbol{\beta} \mid \boldsymbol{\psi}) &= \frac{p(\boldsymbol{\psi} \mid \boldsymbol{\beta}) p(\boldsymbol{\beta})}{\int_{\mathbb{R}^{p+1}} p(\boldsymbol{\psi} \mid \boldsymbol{\beta}) p(\boldsymbol{\beta}) \, \mathrm{d}\boldsymbol{\beta}} \\ &= (2\pi)^{-\frac{p+1}{2}} \det(\boldsymbol{\Sigma}')^{\frac{1}{2}} \exp\left(-\frac{1}{2} (\boldsymbol{\beta} - \boldsymbol{\zeta}')^\top \boldsymbol{\Sigma}'^{-1} (\boldsymbol{\beta} - \boldsymbol{\zeta}') \right), \end{aligned}$$

where the posterior mean $\boldsymbol{\zeta}'$ and covariance $\boldsymbol{\Sigma}'$ are give below:

$$\boldsymbol{\zeta}' = \boldsymbol{\Sigma}' \left(\mathbf{B}^\top \mathbf{K}^{-1} \boldsymbol{\psi} + \boldsymbol{\Sigma}^{-1} \right), \quad \boldsymbol{\Sigma}' = \left(\mathbf{B}^\top \mathbf{K}^{-1} \mathbf{B} + \boldsymbol{\Sigma}^{-1} \right)^{-1}.$$

Note that the conditional distribution $p(Y(\mathbf{x}) \mid \boldsymbol{\psi})$ is obtained by marginalizing $\boldsymbol{\beta}$ out,

$$p(Y(\mathbf{x}) \mid \boldsymbol{\psi}) = \int_{\mathbb{R}^{p+1}} p(Y(\mathbf{x}) \mid \boldsymbol{\beta}, \boldsymbol{\psi}) p(\boldsymbol{\beta} \mid \boldsymbol{\psi}) \, \mathrm{d}\boldsymbol{\beta}.$$

This marginalization can be interpreted as averaging $p(Y(\mathbf{x}) \mid \boldsymbol{\beta}, \boldsymbol{\psi})$ over the posterior of $\boldsymbol{\beta}$. Without giving the details on the derivation, the posterior mean and kernel are expressed as follows (Omre, 1987; O'Hagan and Kingman, 1978):

$$\widehat{Y}(\mathbf{x}) = \left(\mathbf{b} - \mathbf{B}^\top \mathbf{K}^{-1}\mathbf{k}\right)^\top \boldsymbol{\zeta}' + \mathbf{k}^\top \mathbf{K}^{-1}\boldsymbol{\psi} \tag{3.32}$$

$$k'(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') - \mathbf{k}^\top \mathbf{K}^{-1}\mathbf{k}' + \left(\mathbf{b} - \mathbf{B}^\top \mathbf{K}^{-1}\mathbf{k}\right)^\top \boldsymbol{\Sigma}' \left(\mathbf{b} - \mathbf{B}^\top \mathbf{K}^{-1}\mathbf{k}'\right) \tag{3.33}$$

The formula above depends on the choice of prior parameter $\boldsymbol{\zeta}, \boldsymbol{\Sigma}$. Consider the limit $\boldsymbol{\Sigma} \to \mathbf{O}$ (matrix of zeros), meaning $\boldsymbol{\beta}$ becomes more and more *non-informative* because the prior is increasingly flat everywhere. Then posterior mean and covariance matrix of $\boldsymbol{\beta}$ have the following convergence,

$$\boldsymbol{\zeta}' \to \left(\mathbf{B}^\top \mathbf{K}^{-1}\mathbf{B}\right)^{-1} \mathbf{B}^\top \mathbf{K}^{-1}\boldsymbol{\psi}, \quad \boldsymbol{\Sigma}' \to \left(\mathbf{B}^\top \mathbf{K}^{-1}\mathbf{B}\right)^{-1}.$$

Consequently, the posterior mean and kernel converges to the Kriging predictor (BLUP) and covariance (cf. Eq (3.9) and (3.12)):

$$\widehat{Y}(\mathbf{x}) \to \left[\mathbf{k} - \mathbf{B}\left(\mathbf{B}^\top \mathbf{K}^{-1}\mathbf{B}\right)^{-1}\left(\mathbf{B}^\top \mathbf{K}^{-1}\mathbf{k} - \mathbf{b}\right)\right]^\top \mathbf{K}^{-1}\boldsymbol{\psi}$$

$$k'(\mathbf{x}, \mathbf{x}') \to k(\mathbf{x}, \mathbf{x}') - \mathbf{k}^\top \mathbf{K}^{-1}\mathbf{k}' + \left(\mathbf{b} - \mathbf{B}^\top \mathbf{K}^{-1}\mathbf{k}\right)^\top \left(\mathbf{B}^\top \mathbf{K}^{-1}\mathbf{B}\right)^{-1} \left(\mathbf{b} - \mathbf{B}^\top \mathbf{K}^{-1}\mathbf{k}'\right)$$

Because limiting the posterior mean results in the same expression as the Kriging predictor, we shall treat the Kriging predictor and the posterior mean interchangeably in this thesis.

3.1.4 Differentiation

The Kriging predictor and MSE play a central role in Efficient Global Optimization and their derivatives are frequently used in such algorithms. Thus, the gradients of the Kriging predictor (Eq. (3.9)) and MSE (Eq. (3.11)) w.r.t. the index variable are given below (using the *denominator layout*):

$$\frac{\partial \hat{f}}{\partial \mathbf{x}} = \frac{\partial \mathbf{b}}{\partial \mathbf{x}}\hat{\boldsymbol{\beta}} + \frac{\partial \mathbf{k}}{\partial \mathbf{x}}\mathbf{K}^{-1}(\mathbf{y} - \mathbf{B}\hat{\boldsymbol{\beta}}) \tag{3.34}$$

$$\frac{\partial s^2}{\partial \mathbf{x}} = 2\Big[\left(\frac{\partial \mathbf{k}}{\partial \mathbf{x}}\mathbf{K}^{-1}\mathbf{B} - \frac{\partial \mathbf{b}}{\partial \mathbf{x}}\right)(\mathbf{B}^\top \mathbf{K}^{-1}\mathbf{B})^{-1}\left(\mathbf{B}^\top \mathbf{K}^{-1}\mathbf{k} - \mathbf{b}\right) - \frac{\partial \mathbf{k}}{\partial \mathbf{x}}\mathbf{K}^{-1}\mathbf{k}\Big], \tag{3.35}$$

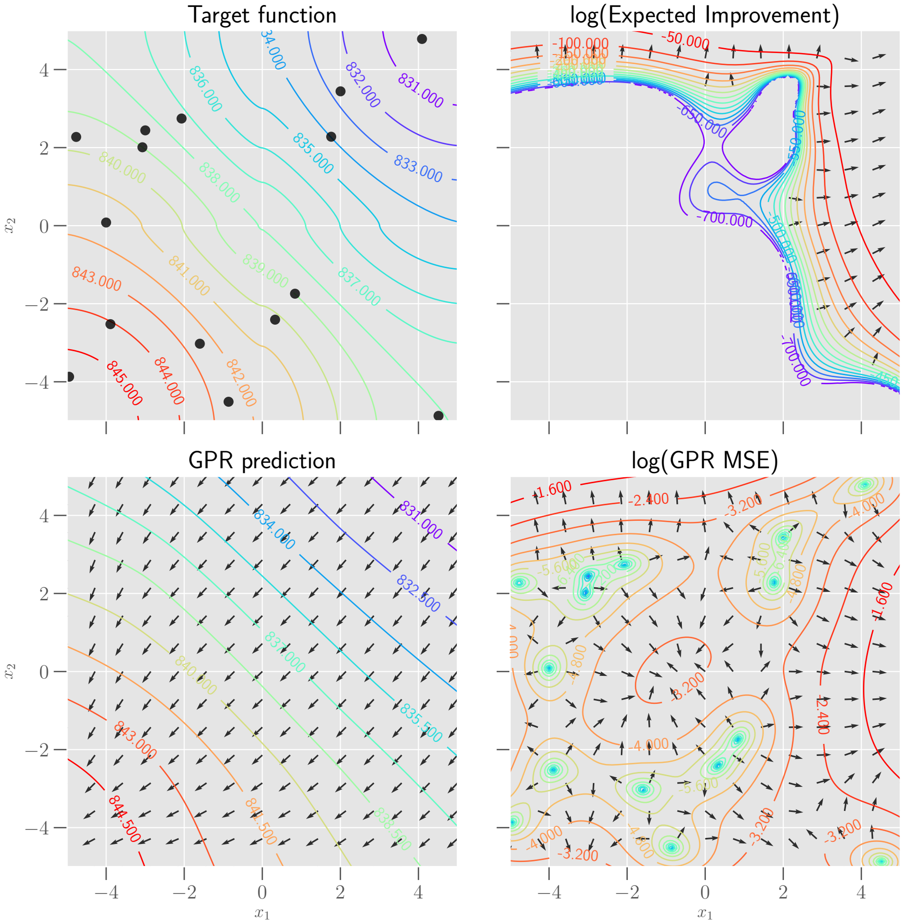


Figure 3.2: On the 2-D *Schwefel function* (**top-left**), several gradient fields and contour lines are depicted for the Kriging/GPR prediction (**bottom-left**), the Kriging/GPR MSE (**bottom-right**) and the so-called *Expected Improvement* criterion (**top-right**) defined on the Kriging prediction and MSE (cf. Eq. (4.5)). Ordinary Kriging with the Matérn 3/2 kernel is chosen for this illustration, which is trained on 15 uniformly generated locations (black dots in the **top-left** plot).

where

$$\frac{\partial \mathbf{k}}{\partial \mathbf{x}} = \left[\frac{\partial k(\mathbf{x}, \mathbf{x}_1)}{\partial \mathbf{x}}, \frac{\partial k(\mathbf{x}, \mathbf{x}_2)}{\partial \mathbf{x}}, \ldots, \frac{\partial k(\mathbf{x}, \mathbf{x}_n)}{\partial \mathbf{x}}\right].$$

For the Matérn 3/2 kernel (Eq. (3.5)), this derivative is given as:

$$\frac{\partial k(\mathbf{x}, \mathbf{x}')}{\partial x_i} = (-1)^s \frac{3\sigma^2 h_i}{\theta_i^2} \exp\left(-\sqrt{3}\frac{h_i}{\theta_i}\right), \quad h_i = |x_i - x_i'|, \quad s = \mathbb{1}_{[x_i', \infty)}(x_i).$$

In addition, in Fig. 3.2, the gradient calculation here is visualized on a 2-D *Schwefel function.*

3.2 Cluster Kriging

Despite the theoretically sound development of the Kriging model, it suffers from several issues when applied to large data sets. The major bottleneck is the high time and memory complexity of the model fitting process: The inverse of the covariance matrix $\mathbf{K}^{-1}$ needs to be computed for both the posterior mean and variance (Eq. (3.9) and (3.11)), which has roughly $O(n^3)$ time complexity (n is the number of data points)[1]. In addition, the likelihood function of hyper-parameters $\sigma, \boldsymbol{\theta}$ is expressed through $\mathbf{K}^{-1}(\sigma^2, \boldsymbol{\theta})$. In the *Maximum Likelihood Estimation* (MLE), $\mathbf{K}^{-1}$ needs to be calculated for each likelihood value, resulting in a $O(n^3)$ computational cost per hyper-parameter evaluation. Even if efficient numerical optimizers are used in MLE, e.g., the quasi-Newton method (Bonnans et al., 2006), this computational overhead is still extremely high for a large data set. This bottleneck hinders the practical usage of Kriging/GPR. Various attempts have been made to relax the computational complexity issue of Kriging (Rasmussen and Williams, 2006). The historical approaches on this topic are categorized as follows.

Subset Methods The first category of approximation algorithms uses only a subset of the complete data set to approximate a full Kriging model. The idea behind these methods is to get a realistic representation of the complete data set by taking only a small portion of the data points. The main issue with the subset approximation approach is to select a representative subset of the data set. Two major subset approximation algorithms are:

[1] There are asymptotically faster algorithms for matrix inversion, e.g., Strassen algorithm $O(n^{2.807})$ and Stothers $O(d^{2.373})$, but their practical performance is worse than some methods with $O(n^3)$ time complexity.

- *Subset of Data* (SoD) (Lawrence, 2004) is a naive approach in reducing complexity by taking a subset of $m < n$ data points. The points are usually taken at random. The obvious disadvantage of such an approach is that possible valuable information is lost in the process. Taking a representative subset of data points is a non-trivial task.

- *Subset of Regressors* (SoR) (Silverman, 1985) approximates Kriging by a linear combination of kernel functions on a set of basis points. The basis points are linearly weighted to construct the predictor. The choice of the basis points does influence the final outcome significantly. As noted also in Quiñonero-Candela and Rasmussen (2005), there are only m (number of basis points) degrees of freedom in the model because the model degenerates, which might be too restrictive.

Approximation using Sparsity In the second category, the sparsity of the covariance kernel is exploited for the approximation. Most of algorithms in this category also use a subset of the data as in the subset approximation method.

- *Sparse On-Line Gaussian Processes* (OGP) (Csató and Opper, 2002) uses a Bayesian on-line algorithm, together with a sequential construction of a subsample of the data that specifies the prediction of the GP model. The idea behind constructing a subsample of basis vectors is very similar to the Fully Independent Training Conditional mentioned below. The advantage of OGP is that additional data points can be added to the OGP model without always completely retraining the model.

- *Gaussian Markov Random Fields* (Hartman and Hössjer, 2008) uses an approximation of the covariance matrix with a sparse precision matrix. It uses *Gaussian Markov Random Fields* (GMRF) on a reasonably dense grid to exploit the computational benefits of a Markov field while keeping the formula of Kriging weights. This method reduces the complexity for simple and ordinary Kriging, but might not always be efficient with universal Kriging.

- *Fully Independent Training Conditional* (FITC) (Naish-Guzman and Holden, 2007; Snelson and Ghahramani, 2005) uses a more sophisticated likelihood approximation with a richer covariance structure. It is a non-degenerate version of the *SoR* algorithm. By providing a set of basis points (Pseudo inputs), the model is fitted and validated on the training data. As with *SoR* the choice of basis points is a problem and it is usually either a subset of the training data or a uniform distribution over the input space.

Divide and Conquer Methods In this category, the time complexity issue is relaxed by partitioning a big data set into several smaller subsets (or clusters) and then constructing a Kriging/GPR model on each subset. Because such a partitioning is usually obtained via clustering techniques, the subset and the model trained on them only capture local properties of the target function. Despite of the construction of local models, typically a *global predictor* is obtained by combining the local Kriging/GPR models. In this thesis, a novel divide and conquer method, called **Cluster Kriging** is proposed.

- *Bayesian Committee Machines* (BCM) (Tresp, 2000) is an algorithm similar to the ones we propose, but developed from a completely different perspective. The basic motivation is to divide a huge training set into several relatively small subsets and then construct GPR models on each subset. The benefit of this approach is that the training time on each subset is satisfactory and the training task can be easily parallelized. After training, the prediction is made by a weighted combination of estimations from all the GPR models. In addition, the batch prediction is enabled to speed up the computation even further. However, when using independent hyper-parameters for each GPR model or some GPR models are badly fitted, BCM yields unsatisfactory performance in terms of accuracy.

- *Cluster Kriging* (CK) (van Stein, Wang, Kowalczyk, Emmerich, and Bäck, 2016) combines multiple local Kriging/GPR predictors that are constructed on several partitions of the data set, where the partitions are obtained from clustering algorithms. Loosely speaking, if the whole data set is partitioned into clusters of similar sizes, Cluster Kriging will reduce the time complexity by a factor of q^2 (where q is the number of clusters), resulting in n^3/q^2, if Kriging estimators are fitted sequentially. When exploiting q CPUs in parallel, the time complexity will be further reduced to n^3/q^3. Ideally, when scaling up q to be a linear function of n, the time complexity is reduced to a linear term of n and even becomes a constant in the parallelization mode. However, in practice, such a setting on q is not suggested because it is necessary to keep enough data points in each cluster, to ensure each local Kriging model is well-fitted. To estimate $f(\mathbf{x})$ at an unobserved data point $\mathbf{x}$,

> each Kriging estimator provides a (local) estimation $\hat{f}$ and it is proposed to either combine all the Kriging estimations or select the most proper Kriging estimations for $f(\mathbf{x})$. There are many options for the data partitioning, e.g., *K-means* (MacQueen et al., 1967) and *Gaussian mixture models* (Reynolds, 2009) (GMM), and the Kriging model on clusters can also be combined in different manners. By varying the options in each step of the Cluster Kriging, many algorithms can be generated. Four of them will be explained in the next section. In this section, the options in each step of the algorithms are introduced step-by-step.

Several other attempts have been made to divide the Kriging model in submodels (Chen and Ren, 2009; Nguyen-Tuong et al., 2009). In Chen and Ren (2009), a *Bagging* (Breiman, 1996) method is proposed to increase the robustness of the Kriging algorithm, rather than speeding up the algorithm's training time. In Nguyen-Tuong et al. (2009), a partitioning method is introduced to separate the data points into local Kriging models and combine the different models using a distance metric.

All of these approximation algorithms have their advantages and disadvantages and they are compared to our newly proposed Cluster Kriging algorithms. For the empirical study, three commonly applied algorithms: *SoD*, *FITC* and *BCM* are selected to compare with the proposed approaches in this thesis.

3.2.1 Clustering

Given some data points $\mathrm{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \subset \mathrm{S}$ and corresponding response values $\mathbf{y} = (f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n))^\top$, the first step in Cluster Kriging is to cluster the data set $(\mathrm{X}, \mathbf{y})$ into several smaller subsets. In general, the goal is to obtain a set $\mathcal{P}$ containing q clusters on the input data set X.

$$\mathcal{P} = \{\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_q\}, \quad \text{where} \bigcup_{i=1}^{q} \mathrm{X}_i = \mathrm{X}. \tag{3.36}$$

As with the partition on X, the response values $\mathbf{y}$ are also grouped: $\mathbf{y} = (\mathbf{y}_1^\top, \mathbf{y}_2^\top, \ldots, \mathbf{y}_q^\top)^\top$. The clustering can be done in many ways, with the most simple and feasible approach being random clustering. For our framework, however, we introduce three more sophisticated partitioning methods that are used in the experiments later on.

Hard Clustering Hard clustering splits the data into k smaller *disjoint* data sets: $\mathrm{X}_i \bigcap \mathrm{X}_j = \emptyset$ $(i \neq j)$. This can be achieved by various methods, for instance the K-means algorithm (MacQueen et al., 1967). K-means clustering minimizes the within-cluster sum of squares, that is expressed as:

$$\arg\min_{\mathcal{P}} \sum_{i=1}^{q} \sum_{\mathbf{x} \in \mathrm{X}_i} \left\| \mathbf{x} - \boldsymbol{\mu}_i \right\|^2 , \tag{3.37}$$

where $\boldsymbol{\mu}_i \in \mathbb{R}^d$ is the centroid of cluster i and is calculated as the mean of the points in X_i. The evaluation of the within-cluster sum of squares takes $O(nqd)$ execution time.

Soft Clustering Instead of using a hard clustering approach, a fuzzy clustering algorithm can be used to introduce slight overlap between the various smaller data sets, which might increase the final model accuracy. To incorporate fuzzy clustering, instead of directly applying cluster labels, the probabilities that a point belongs to a cluster are calculated (Eq. (3.39)). This probability is called the membership value of a point to a cluster. With ν a user defined setting that defines the overlap, $\lceil \nu n/q \rceil$ number of points with the highest membership values are assigned for each cluster. Here ν is set between 1 (no overlap) and 2 (completely overlapping clusters).

In principle, any fuzzy clustering algorithm can be used for the partitioning. In this thesis the *Fuzzy C-means* (FCM) (Dunn, 1973) clustering algorithm and the *Gaussian Mixture Models* (GMM) (Reynolds, 2009) are used. FCM is a clustering algorithm very similar to the well known *K-means*. The algorithm differs from K-means in that it has additional membership coefficients and a fuzzifier. The membership coefficients of a given point give the degrees that this point belongs to each cluster. These coefficients are normalized so they sum up to one. The algorithm can be fitted on a given data set and returns the coefficients for each data point to each cluster. The number of clusters is a user defined parameter. Fuzzy C-means optimizes the objective function given in Eq. (3.38) iteratively. In each iteration, the membership coefficients of each point being in the clusters are computed using Eq. (3.39). Subsequently, the centroid of each cluster $\boldsymbol{\mu}_j$ is computed as the center of mass of all data points, taking the membership coefficients as weights. The objective of fuzzy C-means is to find a set of centroids that minimizes the following function:

$$\sum_{i=1}^{n}\sum_{j=1}^{q} w_{ij}^{m} \left\| \mathbf{x}_i - \boldsymbol{\mu}_j \right\|^2 , \tag{3.38}$$

where w_{ij} are the membership values (see Eq. 3.39) and m is the so-called fuzzifier ($m = 2$ in this thesis). The fuzzifier determines the level of cluster fuzziness as follows:

$$w_{ij}^{m} = \frac{1}{\sum_{k=1}^{q} \left(\frac{\left\| \mathbf{x}_i - \boldsymbol{\mu}_j \right\|}{\left\| \mathbf{x}_i - \boldsymbol{\mu}_k \right\|} \right)^{\frac{2}{m-1}}} \tag{3.39}$$

The other fuzzy clustering procedure used is the Gaussian Mixture Models. GMM are used together with the *expectation-maximization* (EM) (Sundberg, 1974) algorithm for fitting the Gaussian models. The mixture models are fitted on the training data and later used in the weighted combination of the Kriging models by estimating cluster membership probabilities of the unseen data points. The advantage of this clustering technique is that it is fairly robust and that the number of clusters can be specified by the user. For the GMM method one could use the full covariance matrix whenever the dimensionality of the input data is small. However, when working with high dimensional data a diagonal covariance matrix can be used instead. The time complexity of GMM depends on the underlying EM algorithm. In each iteration of EM, it takes $O(nq)$ operations to re-estimate the model parameters.

Regression Tree Partitioning The third method used is the partitioning by use of a Regression Tree (Breiman et al., 1984) on the complete training set. The regression tree splits the data set recursively at the best splitting point using the variance reduction criterion. Each leaf node of the Regression Tree represents a cluster of data points. The number of leaves (or the number of records per leave) can be set by the user. By reducing the variance in each leaf node and therefore the variance in each data set, the Kriging models can be fitted to the local data sets much better as will be presented later on. The time complexity of using a Regression Tree for the partitioning is $O(n)$, given that the depth of the tree or the number of leaf nodes is set by the user.

The partitioning done by the regression tree depends on the splitting criterion. For a faster execution of the Cluster Kriging algorithm we could choose to use a splitting criterion that splits the data set in each node evenly, balancing the load for each of the local Kriging models attached to the leafs. From emprical experience we know that splitting using the standard variance reduction function generally results in better performing models than using such an evenly splitting criterion. This is likely due to the fact that data sets with a lower variance can be more easily fitted by a Kriging model.

3.2.2 Modeling

Technically, modeling the function f using Kriging/GPR implies using the stochastic process $\{Y(\mathbf{x}) : \mathbf{x} \in \mathrm{S}\}$ (cf. Eq. (3.1)) as the *statistical model* of f. Under this setting, the response values $\mathbf{y}$ are treated as the observations from Y. After partitioning the data set into several clusters, Kriging/GPR models are fitted on each of the smaller data sets. Consider the random vector $\boldsymbol{\psi} = (Y(\mathbf{x}_1), Y(\mathbf{x}_2), \ldots, Y(\mathbf{x}_n))^\top$. It is also partitioned according to the clustering on X: $\boldsymbol{\psi} = (\boldsymbol{\psi}_1^\top, \boldsymbol{\psi}_2^\top, \ldots, \boldsymbol{\psi}_q^\top)^\top$. For simplicity we assume the kernel functions used on each cluster to be the same and *Ordinary Kriging* is used in each cluster. Typically, each cluster only captures the local information about f and thus the Kriging model on each cluster shall be called *local Kriging/GPR model.* On each cluster, the (local) posterior distribution of the $Y(\mathbf{x})$ is:

$$Y(\mathbf{x}) \mid \boldsymbol{\psi}_i = \mathbf{y}_i \sim \boldsymbol{\mathcal{N}}\left(\hat{f}_i(\mathbf{x}), s_i^2(\mathbf{x})\right), \quad i = 1, 2, \ldots, q, \tag{3.40}$$

where $\hat{f}_i$ and s_i^2 are the Kriging estimator and MSE in Eq. (3.9) and (3.11) except that the observations $\mathbf{y}_i$ is now only a fraction of the whole observations $\mathbf{y}$. Note that training the Kriging estimator can be easily parallelized, which gives an additional speedup to Cluster Kriging. Another benefit of building each model separately, is that each model has usually a much better local fit than a single global Kriging model would obtain.

3.2.3 Cluster Kriging Predictor

For the prediction, several approaches are proposed in the following. Depending on the partitioning method used before, the simplest approach to predict the unseen data point is by using a single local model. When the partitions are overlapping a combination of the different local models into one global model is then required.

Single Cluster Predictor The simplest method is to pick just one local Kriging model for each data point and use this local model for the prediction. This does require the partitioning used to create partitions based on locality like K-means clustering or a regression tree. First the partitioning method is used to predict which cluster the new data point belongs to, then the Kriging model trained using this particular cluster is used to predict the mean and variance at the new data point. In case of the Regression Tree procedure, the targets are predicted from new unseen data points by first deciding which model needs to be used, using the Regression Tree. The target is then predicted using the specific Kriging model assigned to the leaf node. The main advantage of this method is that there is no combination of different predictions and only one of the local Kriging models needs to provide a prediction. This results in a significant speed-up for the prediction task. Disadvantages of this method are 1) a potential inability of capturing the global trend of the target function and 2) artificial discontinuities at the boundary of partitions. In Fig. 3.3 (top row), we visualize a Cluster Kriging model using regression trees, in which the intersections between the different local models are marked by black dashed lines. It can be observed that the edges of the local models are not completely matching, meaning that the predictions near the border are not as smooth as they would be in a global Kriging model. It can also be observed that the area covered by each cluster is not the same, which is due to the splitting criterion of the regression tree. While the splitting criterion could be chosen in such a way that it balances the cluster sizes, using variance reduction as the splitting criterion generally gives better fitted local models.

Superposition of Posterior Processes Instead of using single model predictions, the multiple local models can be combined into one global model using various combination procedures. Some additional assumptions are necessary to give the following derivation. Assume an independent Gaussian process prior on each cluster:

$$\forall i \neq j \in \{1, 2, \dots, q\}, \quad Y_i \perp\!\!\!\perp Y_j, \quad Y_i \sim t + \mathcal{GP}(0, k(\cdot, \cdot)).$$

After clustering (e.g., *K-means*) the data set $(\mathrm{X}, \mathbf{y})$, independent posterior Gaussian processes Y'_i are obtained on each cluster:

$$Y'_i := Y_i \mid \boldsymbol{\psi}_i \sim \mathcal{GP}(\hat{f}_i, k'_i(\cdot, \cdot)),$$

where the posterior mean $\hat{f}_i$ and kernel k'_i are defined in Eq. (3.32) and (3.33). In this sense, it is possible to construct a "global" Gaussian process as the superposition of all posterior Gaussian processes. In addition, a weighting scheme $\{w_i\}_i$ is used to model how much "belief" should be put on each posterior process. Using positive weights whose sum is one, the posterior process is:

$$\mathcal{Y} := \sum_{i=1}^{q} w_i Y'_i \sim \mathcal{GP}\left(\sum_{i=1}^{q} w_i \hat{f}_i, \sum_{i=1}^{q} w_i^2 k'_i(\cdot,\cdot)\right),$$

The posterior kernel is derived as follows: consider the covariance between $\mathcal{Y}(\mathbf{x}_1)$ and $\mathcal{Y}(\mathbf{x}_2)$:

$$\begin{aligned}\operatorname{Cov}\left\{\sum_{i=1}^{q} w_i Y'_i(\mathbf{x}_1), \sum_{j=1}^{q} w_j Y'_j(\mathbf{x}_2)\right\} &= \sum_{i=1}^{q}\sum_{j=1}^{q} w_i w_j \operatorname{Cov}\left\{Y'_i(\mathbf{x}_1), Y'_j(\mathbf{x}_2)\right\} \\ &= \sum_{i=1}^{q} w_i^2 k'_i(\mathbf{x}_1, \mathbf{x}_2).\end{aligned}$$

At an unobserved point $\mathbf{x}$, the following predictive distribution is obtained,

$$\mathcal{Y}(\mathbf{x}) \sim \mathcal{N}\left(\sum_{i=1}^{q} w_i \hat{f}_i(\mathbf{x}), \sum_{i=1}^{q} w_i^2 s_i^2(\mathbf{x})\right), \tag{3.41}$$

where $s_i^2(\mathbf{x}) = k'_i(\mathbf{x}, \mathbf{x})$. The best linear unbiased predictor of $\mathcal{Y}$ is its mean function: $\widehat{\mathcal{Y}} = \sum_{i=1}^{q} w_i \hat{f}_i$ and its MSE is the variance $\sum_{i=1}^{q} w_i^2 s_i^2$. Note that the predictor and its MSE depend on the choice of weights. The optimal predictor is defined in the sense that the MSE is minimized with respect to the weight (van Stein, Wang, Kowalczyk, Bäck, and Emmerich, 2015):

$$\begin{aligned} \underset{\{w_1,\ldots,w_q\}}{\text{minimize}} \quad & \sum_{i=1}^{q} w_i^2 s_i^2(\mathbf{x}) \\ \text{subject to} \quad & \sum_{i=1}^{q} w_i = 1, \quad w_i \geq 0, \quad i = 1, \ldots, q. \end{aligned}$$

This convex optimization problem can be solved by using Lagrange Multipliers, resulting in:

$$w_i^*(\mathbf{x}) = \frac{1/s_i^2(\mathbf{x})}{\sum_{j=1}^{q} 1/s_j^2(\mathbf{x})}. \tag{3.42}$$

The optimal weights are then used to construct the optimal predictor, which is the inner product of the model predictions with the optimal weights.

Mixture of Posterior Processes As an alternative to the linear predictor given in Eq. (3.41) that arises from the superposition of posterior processes, we formulate another linear predictor here, resulting from the *mixture* of posterior processes. Firstly, the combination weights are motivated a bit differently: for the GMM and other soft clustering approaches, the membership probabilities can be used for unseen records to define the weights for the combination of predictions. For instance, given a point $\mathbf{x}$, the weights are defined as,

$$w_i := \Pr(C = i \mid \mathbf{x}), \quad i = 1, \ldots, q, \tag{3.43}$$

where C is the cluster indicator variable ranging from 1 to q. Note that those weights can be given by the clustering algorithm or obtained by an optimization procedure (see below). Secondly, instead of considering an independent Gaussian process prior for each cluster, *a single and global Gaussian process prior is assumed for all clusters.* By applying the total probability with respect to the cluster indicator variable C, the conditional density of Y over $\boldsymbol{\psi}$ is (van Stein, Wang, Kowalczyk, Emmerich, and Bäck, 2016):

$$\begin{aligned} p(Y(\mathbf{x}) \mid \boldsymbol{\psi} = \mathbf{y}) &= \sum_{i=1}^{q} p(Y(\mathbf{x}), C = i \mid \boldsymbol{\psi} = \mathbf{y}, \mathbf{x}) \\ &= \sum_{i=1}^{q} p(Y(\mathbf{x}) \mid C = i, \boldsymbol{\psi} = \mathbf{y}) \Pr(C = i \mid \mathbf{x}) \\ &\approx \sum_{i=1}^{q} p(Y(\mathbf{x}) \mid \boldsymbol{\psi}_i = \mathbf{y}_i) \Pr(C = i \mid \mathbf{x}). \end{aligned} \tag{3.44}$$

Note that we approximate the density $p(Y(\mathbf{x}) \mid C = i, \boldsymbol{\psi} = \mathbf{y})$ by $p(Y(\mathbf{x}) \mid \boldsymbol{\psi}_i = \mathbf{y}_i)$. Such an approximation is accurate when the amount of the overlap between clusters is small. In Eq. (3.44), the first term within the summation is the posterior density obtained from cluster i. The second term represents the probability that data point $\mathbf{x}$ belonging to a cluster, which is the weight in Eq. (3.43). Consequently, the overall predictive density $p(Y(\mathbf{x}) \mid \boldsymbol{\psi} = \mathbf{y})$ comes from the *mixture of posterior processes.* According to statistical decision theory (Hastie et al., 2009), the best predictor of Y when knowing the conditional density of Y on $\mathbf{y}$ is the conditional expectation, i.e.,

$$
\begin{aligned}
\mathbb{E}\left\{Y(\mathbf{x}) \mid \boldsymbol{\psi}=\mathbf{y}\right\} &= \int_{-\infty}^{\infty} y \sum_{i=1}^{q} p(Y(\mathbf{y}) \mid \boldsymbol{\psi}_i = \mathbf{y}_i) \Pr(C=i \mid \mathbf{x}) \,\mathrm{d}y \\
&= \sum_{i=1}^{q} \Pr(C=i \mid \mathbf{x}) \mathbb{E}\left\{Y(\mathbf{x}) \mid \boldsymbol{\psi}_i = \mathbf{y}_i\right\} \\
&= \sum_{i=1}^{q} w_i \hat{f}_i(\mathbf{x}). \qquad (3.45)
\end{aligned}
$$

In contrast to Eq. (3.41), the predictor above is also a linear combination of Kriging predictors from all clusters. However, the differences are 1) the predictive density $p(Y(\mathbf{x}) \mid \boldsymbol{\psi} = \mathbf{y})$ is no longer Gaussian, 2) the weights in Eq. (3.41) are resulted from an optimization procedure while the weights in Eq. (3.45) are either given directly by the clustering algorithm or obtained from the optimization. To optimize the weights, please consider the MSE of this predictor, which is the variance of the mixture of posterior processes:

$$
\begin{aligned}
&\operatorname{Var}\left\{Y(\mathbf{x}) \mid \boldsymbol{\psi}=\mathbf{y}\right\} \\
&= \mathbb{E}\left\{Y(\mathbf{x})^2 \mid \boldsymbol{\psi}=\mathbf{y}\right\} - \left(\mathbb{E}\left\{Y(\mathbf{x}) \mid \boldsymbol{\psi}=\mathbf{y}\right\}\right)^2 \\
&= \sum_{i=1}^{q} w_i \left(\operatorname{Var}\{Y(\mathbf{x}) \mid \boldsymbol{\psi}_i = \mathbf{y}_i\} + (\mathbb{E}\{Y(\mathbf{x}) \mid \boldsymbol{\psi}_i = \mathbf{y}_i\})^2\right) - (\mathbb{E}\{Y(\mathbf{x}) \mid \boldsymbol{\psi} = \mathbf{y}\})^2 \\
&= \sum_{i=1}^{q} w_i \left(s_i^2(\mathbf{x}) + \hat{f}_i^2(\mathbf{x})\right) - \left(\sum_{i=1}^{q} w_i \hat{f}_i(\mathbf{x})\right)^2. \qquad (3.46)
\end{aligned}
$$

Again, the weights are considered optimal in the sense that the MSE is minimized:

$$
\begin{aligned}
\underset{\{w_1,\ldots,w_q\}}{\text{minimize}} \quad & \operatorname{Var}\left\{Y(\mathbf{x}) \mid \boldsymbol{\psi}=\mathbf{y}\right\} \\
\text{subject to} \quad & \sum_{i=1}^{q} w_i = 1, \quad w_i \geq 0, \quad i = 1, \ldots, q.
\end{aligned}
$$

Cluster Kriging Variants By choosing different methods for the clustering and prediction, various Cluster Kriging variants are instantiated:

- *Optimally Weighted Cluster Kriging* (OWCK) uses a *K-means* clustering algorithm for the partitioning and the superposition of posterior processes to construct the predictor.

- *Optimally Weighted Fuzzy Cluster Kriging* (OWFCK) is similar to OWCK except that *K-means* is replaced by *Fuzzy C-means.*

- *Gaussian Mixture Model Cluster Kriging* (GMMCK) uses Gaussian Mixture Models to partition the data into q overlapping clusters and the membership probabilities are used as the combination weights. The mixture of posterior Gaussian processes (Eq. (3.45)) is used for the prediction.

- *Model Tree Cluster Kriging* (MTCK) uses a regression tree to partition the data in the objective space. The tree is generated from the root node by recursively splitting the training data using the target variable and the variance reduction criterion. Once a node contains less than the minimum samples needed to split or the node contains only one record, the splitting stops and the node is called a leaf. To control the number of clusters, the user can set the maximum number of leaves or the minimum leaf size. Next, each leaf node is assigned a unique index and each record belonging to the leaf is assigned to this index. For each leaf, a Kriging predictor is built using only those records assigned to this leaf. For the prediction, the regression tree decides which Kriging predictor should be used.

3.2.4 Experiments

A broad variety of experiments is conducted to compare Optimally Weighted Cluster Kriging and its Fuzzy and Model Tree variants, to a wide set of other Kriging approximation algorithms. The algorithms included in the test are: *Bayesian Committee Machines*, both with shared parameters (BCM sh.) and with individual parameters (BCM), *Subset of Data* (SoD), *Fully Independent Training Conditional* (FITC), *Optimally Weighted Cluster Kriging* (OWCK) using K-means clustering, Fuzzy Cluster Kriging using Fuzzy C-means (OWFCK), Fuzzy Cluster Kriging with Gaussian Mixture Models (GMMCK) and finally Model Tree Cluster Kriging (MTCK). The algorithms are evaluated on three different data sets from the *UCI machine learning repository* (Bache and Lichman, 2013):

- *Concrete Strength* (Yeh, 1998) is a data set with 1030 records, 8 attributes and one target attribute. The task is to predict the strength of concrete.

- *Combined Cycle Power Plant* (CCPP) (Kaya et al., 2012) is a data set of 9568 records, 3 attributes and one target attribute. The target is the hourly electrical energy output and the task is to predict this target.

- *SARCOS* (Vijayakumar et al., 2005) is a data set from *gaussianprocess.org* with a training set of 44484 records, 21 attributes and 7 target attributes. The task is to predict the joint torques of an anthropomorphic robot arm. All 21 attributes are used as training data but only the 1^{st} target attribute is used as target. The data set comes with a predefined test set of 4449 records.

For the *Concrete Strength* data set and all synthetic data sets: FITC is set to a range of inducing points starting from 32 and increasing in powers of 2 to 512. SoD is set to the same range as FITC but for SoD this means the number of data points. BCM, both shared and non-shared versions and all *Cluster Kriging* variants are set to a range from 2 to 32 clusters, increasing with powers of 2. For the *Combined Cycle Power Plant* data set: FITC is set to a range of inducing points starting from 64 and increasing in powers of 2 to 1024. *SoD* is set to a range from 256 to 4092 data points. BCM, both shared and non-shared versions and all *Cluster Kriging* variants are set to a range from 4 to 64 clusters. Finally, for the *SARCOS* data set, the range of FITC's inducing points stays the same as for the CCPP data set, for SoD the range is from 512 to 8184 data points, and for all cluster based algorithms and the model tree variant, the range is set from 8 to 128 clusters.

In addition, 8 synthetic data sets with each 10000 records, 20 attributes and one target attribute are used. The synthetic data sets are generated on common benchmark functions: *Ackley*, *Schaffer*, *Schwefel*, *Rastrigin*, *H1*, *Rosenbrock*, *Himmelblau* and *Diffpow*. The implementations of those functions are taken from the *Deap* Python Package (Fortin et al., 2012).

Hyper-parameter Optimization As mentioned before, Ordinary Kriging is chosen for all the clusters throughout this thesis. For each local Ordinary Kriging model, its constant trend β is estimated using the GLS (Generalized Least Squares) formula (Eq. (3.9)). Consequently, the so-called profile log-likelihood is adopted to estimate the hyper-parameter. In each local Kriging model, hyper-parameters $\sigma^2, \boldsymbol{\theta}$ of the kernel function are optimized using the Maximum Likelihood Estimation (MLE) method. As for the choice of numerical optimization algorithm, we use a quasi-Newton method (BFGS) (Fletcher, 2013) with restarting heuristic.

Table 3.2: Average R^2 score per data set for each algorithm

Data set	SOD	OWCK	GMMCK	OWFCK	FITC	BCM	BCM sh.	MTCK
concrete	0.784	0.826	0.839	0.696	0.675	−81.888	−242.459	**0.851**
CCPP	0.948	0.937	0.968	0.916	0.890	0.220	−24.602	**0.968**
sarcos	0.964	0.894	0.996	0.570	0.941	−627.280	0.448	**0.999**
ackley	0.952	0.957	0.951	0.954	0.260	0.921	−0.039	**0.981**
schaffer	0.321	0.388	0.369	0.406	0.208	0.452	−0.050	**0.672**
schwefel	0.990	0.973	0.977	0.947	0.006	0.969	−0.043	**0.999**
rast	0.973	0.947	0.948	0.932	0.322	0.914	−0.043	**0.998**
h1	0.676	−0.082	0.527	−1.125	0.165	0.657	−0.046	**0.977**
rosenbrock	0.999	0.997	0.997	0.981	0.000	0.994	−0.050	**1.000**
himmelblau	0.997	0.995	0.995	0.981	0.291	0.994	−0.044	**1.000**
diffpow	0.995	0.991	0.991	0.975	0.001	−0.001	−0.001	**1.000**

Table 3.3: Average MSLL score per data set for each algorithm

Data set	SOD	OWCK	GMMCK	OWFCK	FITC	BCM	BCM sh.	MTCK
concrete	−0.837	−0.946	−1.100	−0.692	−0.629	18.590	68.013	**−1.140**
CCPP	−0.089	−1.438	−1.525	−1.109	−1.165	7.826	69.346	−1.193
sarcos	−1.926	−1.371	−3.147	−0.302	−1.463	780.090	507.721	**−3.429**
ackley	−1.622	−1.516	−1.517	−1.462	−0.104	7.352	13.010	**−2.012**
schaffer	0.477	−0.073	0.081	−0.091	−0.107	16.872	11.707	**−0.514**
schwefel	−2.554	−2.013	−2.162	−1.944	−0.002	−0.144	12.034	**−3.278**
rast	−2.179	−1.686	−1.807	−1.642	−0.193	4.554	11.590	**−2.901**
h1	−0.766	−0.276	−0.540	−0.060	−0.059	9.018	17.393	**−1.967**
rosenbrock	−3.479	−2.915	−3.074	−2.738	high*	0.612	18.575	**−4.054**
himmelblau	−3.204	−2.646	−2.790	−2.553	−0.193	−1.422	12.826	**−3.739**
diffpow	−3.020	−2.548	−2.666	−2.438	high*	high*	high*	**−3.744**

Table 3.4: Average SMSE score per data set for each algorithm

Data set	SOD	OWCK	GMM-CK	FCM-CK	FITC	BCM	BCM sh.	MTCK
concrete	0.216	0.174	0.161	0.304	0.325	82.888	243.459	**0.149**
CCPP	0.052	0.063	0.032	0.084	0.110	0.780	25.602	**0.032**
sarcos	0.036	0.106	0.004	0.430	0.059	628.280	0.552	**0.001**
ackley	0.048	0.043	0.049	0.046	0.740	0.079	1.039	**0.019**
schaffer	0.679	0.612	0.631	0.594	0.792	0.548	1.050	**0.328**
schwefel	0.010	0.027	0.023	0.053	0.994	0.031	1.043	**0.001**
rast	0.027	0.053	0.052	0.068	0.678	0.086	1.043	**0.002**
h1	0.324	1.082	0.473	2.125	0.835	0.343	1.046	**0.023**
rosenbrock	0.001	0.003	0.003	0.019	1.000	0.006	1.050	**0.000**
himmelblau	0.003	0.005	0.005	0.019	0.709	0.006	1.044	**0.000**
diffpow	0.005	0.009	0.009	0.025	0.999	1.001	1.001	**0.000**

Whenever the fuzzy clustering algorithm is applied, the overlap rate ν is set to 10%, which is chosen based empirical investigations: although higher percentages (above 10%) usually increase the accuracy marginally, it also brings additional computational costs as each cluster becomes larger. For the Model Tree variant, the number of leaves is enforced by setting a minimum number of data points per leaf and an optional maximum number of leaves.

Quality Measurements The quality of the experiments is estimated with the help of 5-fold cross validation, except for the *SARCOS* data set, which uses its predefined test set. The experiments are performed in a test framework similar to the framework proposed in (Chalupka et al., 2013), i.e., several quality measurements are used to evaluate the performance of each algorithm. The *Coefficient of determination* R^2 score, *Mean Standardized Log Loss* (MSLL) (Rasmussen and Williams, 2006) and the *Standardized Mean Squared Error* (SMSE) are measured for each test run. The *Mean Standardized Log Loss* is a measurement that takes both the prediction and MSE (estimated by the model) into account, penalizing inaccurate predictions that have small estimated MSEs. For MSLL and SMSE lower scores are better, for R^2, 1.0 is the best possible score meaning a perfect fit and everything lower is worse.

Results On real-world data sets *Concrete Strength*, *CCPP* and *SARCOS*, the experiment results are summarized in the following tables. Two performance measures, time and accuracy (x and y axis respectively) are shown. The R^2 scores of each data set per algorithm, averaged over all folds, are shown in Table 3.2. The MSLL scores are provided in Tab. 3.3 and the SMSE scores in Tab. 3.4. The best results for each data set are indicated in bold face.

3.3 Cluster Kriging and EGO

When applying the EGO algorithm to a large initial data set (e.g., in the experiment design), typically the Kriging model is re-trained in every iteration and the CPU time spent on the hyper-parameter re-estimation becomes computationally infeasible. To relax this bottleneck, it is proposed to use the Cluster Kriging algorithm in an EGO algorithm. Specifically, the following three Cluster Kriging variants shall be used:

- *Cluster Kriging* (OWCK)
- *Gaussian Mixture Model Cluster Kriging* (GMMCK)
- *Model Tree Cluster Kriging* (MTCK)

When choosing the MTCK variant, it brings several other advantages than the time complexity reduction. 1) The search space is recursively divided into smaller hypercubes, in a manner that the variance of the target value on each node is greedily reduced. Such a reduced the variance of target values in each cluster potentially facilitates the numerical stability in the model training, because the covariance matrix $\mathbf{K}$ tends to become singular when the target value varies abruptly in the local scale. 2) For the infill criterion, multi-modality is artificially created as a by-product of applying the MTCK variant. Intuitively, as independent Kriging/GPR models are trained on each tree partitions, the prediction MSE increases rapidly around the boundary of the partition. Potentially, this behavior results in local optimality of the infill criterion on each partition. Using this artefact, multiple distinct and potentially well-performing points can be proposed for the evaluation. Essentially, this is an alternative approach to the *infill criterion parallelization* problem stated in Section 4.5. Our argument is visually validated in Fig. 3.3. Here 500 data points are sampled on the 2-D Ackley function using the *Halton sequence* (Niederreiter, 1992). It is important to observe that the prediction MSE shows basins of attraction on each partition. Consequently, the expected improvement criterion also exhibits basins of attraction on each partition and thus is highly multi-modal. For each partition, the local maximum of EI is indicated by the red star symbol.

3.3.1 The algorithm

Although various complexity reduction (or approximation) methods exist for Kriging (for instance, FITC (Naish-Guzman and Holden, 2007; Snelson and Ghahramani, 2005) and Bayesian Committee Machines (Tresp, 2000)), we state that Cluster Kriging is more suitable for the EGO algorithm for the following reasons (Wang et al., 2017):

1. Kriging predictors (posterior processes in Eq. (3.40)) on each cluster can be trained in parallel, which yields an additional linear speedup in practice.

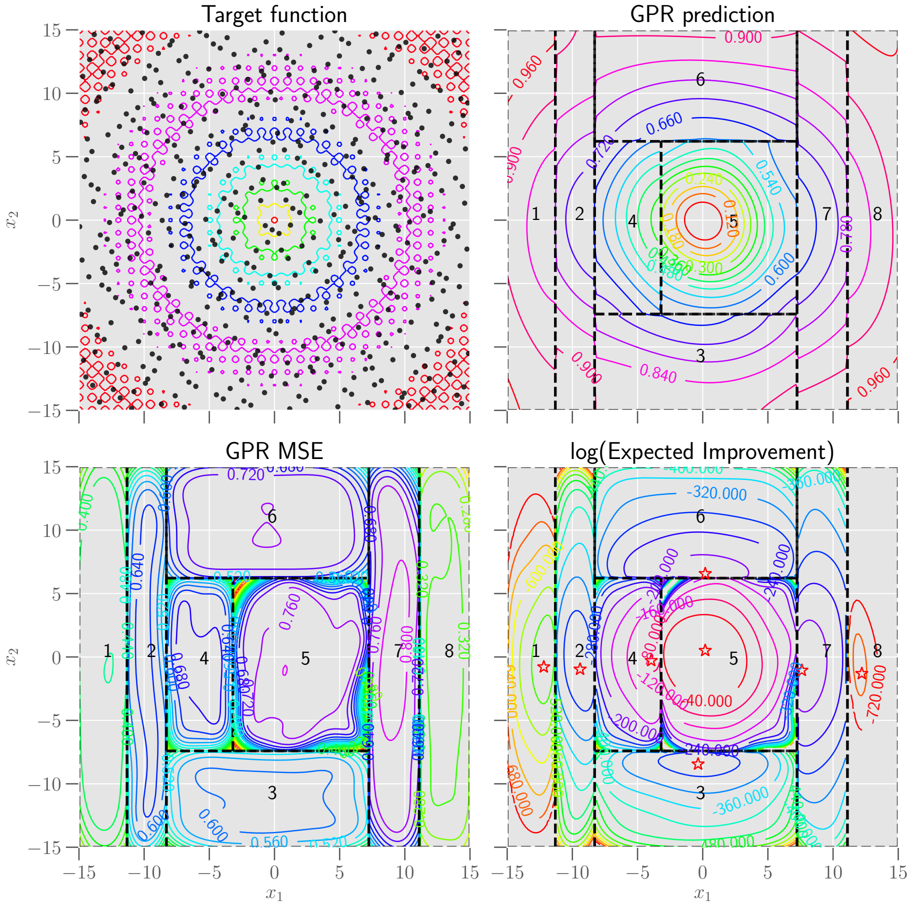


Figure 3.3: On the 2-D *Ackley function* (**top-left**), 500 random points (black dots in the top-left plot) are generated using the *Halton sequence* (Niederreiter, 1992). A Model Tree Cluster Kriging (MTCK) with the Gaussian kernel is trained on the data point, where the underlying tree clusters are indicated by dashed lines (except the top-left plot). Contour lines are depicted for the Kriging/GPR prediction (**top-right**), the Kriging/GPR MSE (**bottom-left**) and the so-called *Expected Improvement* (EI) criterion (**bottom-right**) defined on the Kriging prediction and MSE (cf. Eq. (4.5)). Multiple local maxima (☆ in the bottom-right plot) of EI are obtained by conducting the quasi-Newton search on each cluster.

2. After a new candidate solution is found via the optimization on the infill criterion, the hyper-parameters of Kriging need to be re-estimated. Taking the cluster information into account, it is proposed to only re-estimate the hyper-parameters on the clusters that this new solution belongs to. This operation leads to additional speedup in model training, as in the best scenario, only one local Kriging predictor is re-trained.

3. The infill criterion, e.g., the expected improvement is still well-defined over the Cluster Kriging because either the Gaussian posterior process (Eq. (3.41)) or the mean and variance function (Eq. (3.45) and (3.46)) are available.

The resulting algorithm is presented in Alg. 7. Note that, the training of the initial Kriging models can be parallelized (line 4). The counter c is used to keep track of the number of the recently evaluated data points. When c is bigger than 10% of the size of the data set, the clustering procedure is performed again to balance the size of clusters.

The commonly used infill criteria, e.g., Expected Improvement (Eq. (4.5)) remain well-defined on all the variants of Cluster Kriging in the following sense. Usually infill criteria are defined over the posterior process and take the Gaussian assumption on it. Some Cluster Kriging variants, e.g., superposition of posterior processes (Eq. (3.41)) admit a *Gaussian* posterior. For the others, e.g., mixture of the posterior processes (Eq. (3.45)), we argue that although the posterior is not Gaussian any longer, it is accurate enough to use the first- and second-order structure of the posterior for infill criteria.

For the optimization of the infill criterion (line 8 in Alg. 7), it is possible to exploit fast black-box optimization algorithms, for instance the well-known *Covariance Matrix Adaptation Evolution Strategy* (CMA-ES) (Hansen, 2006; Hansen and Ostermeier, 2001), because the evaluation of the infill criterion is not expensive compared to the Kriging fitting procedure. However, as most of infill criteria have a closed-form, it is straightforward to explore the gradient field of infill criteria. And the global optimization can be conducted by applying a quasi-Newton method with random restarts. To align with existing work (Roustant et al., 2012) on using gradient-based optimization techniques for the infill criteria, we give the gradient of the predictor and MSE in Cluster Kriging, as they are required to differentiate most of infill criteria. For the superposition of posterior processes (Eq. (3.41)), the

Algorithm 7 Cluster Kriging assisted Efficient Global Optimization

```
procedure CK-EGO$(\mathrm{X}, \mathbf{y}, f, q, \mathscr{A})$                ▷ $q$: number of clusters
    $\{\mathrm{X}_i, \mathbf{y}_i\}_{i=1}^{q} \leftarrow$ CLUSTERING$(\mathrm{X}, \mathbf{y}, q)$
    for $i = 1 \rightarrow q$ do
        $Y \mid \mathbf{y}_i \sim \mathcal{N}\left(\hat{f}_i, s_i^2\right)$        ▷ Train the Kriging predictor on each cluster
    end for
    $c \leftarrow 0$
    while the stop criteria are not fulfilled do
        $\mathbf{x}^* \leftarrow \arg\max_{\mathbf{x} \in \mathrm{S}} \mathscr{A}(\mathbf{x})$                ▷ Maximize the infill criterion
        $y^* \leftarrow f(\mathbf{x}^*)$                                              ▷ Evaluation
        $c \leftarrow c + 1$
        if $c/|\mathrm{X}| > 10\%$ then                                   ▷ $|\mathrm{X}|$: cardinality of X
            $\mathrm{X}, \mathbf{y} \leftarrow$ MERGE$(\{\mathrm{X}_i, \mathbf{y}_i\}_{i=1}^{q})$
            $\{\mathrm{X}_i, \mathbf{y}_i\}_{i=1}^{q} \leftarrow$ CLUSTERING$(\mathrm{X}, \mathbf{y}, q)$                ▷ Re-clustering
            for $i = 1 \rightarrow q$ do
                $Y \mid \mathbf{y}_i \sim \mathcal{N}\left(\hat{f}_i, s_i^2\right)$
            end for
            $c \leftarrow 0$
        else
            for every cluster $i$ that $\mathbf{x}^*$ belongs to do
                $\mathrm{X}_i \leftarrow \mathrm{X}_i \cup \{\mathbf{x}^*\},\ \mathbf{y}_i \leftarrow (\mathbf{y}_i^\top, y^*)^\top$          ▷ Extend the data set
                $Y \mid \mathbf{y}_i \sim \mathcal{N}\left(\hat{f}_i, s_i^2\right)$          ▷ Re-train the predictor on cluster $i$
            end for
        end if
    end while
    return $\mathbf{x}^*$
end procedure
```

gradients of its predictor and MSE are (cf. Eq. (3.41) and (3.42)):

$$\frac{\partial \hat{f}}{\partial \mathbf{x}} = \sum_{i=1}^{q} \left(w_i \frac{\partial \hat{f}_i}{\partial \mathbf{x}} + \hat{f}_i \frac{\partial w_i}{\partial \mathbf{x}} \right)$$

$$\frac{\partial s^2}{\partial \mathbf{x}} = \sum_{i=1}^{q} \left(w_i^2 \frac{\partial s_i^2}{\partial \mathbf{x}} + 2 w_i s_i^2 \frac{\partial w_i}{\partial \mathbf{x}} \right)$$

$$\frac{\partial w_i}{\partial \mathbf{x}} = \sum_{i=1}^{q} \left(\frac{1}{s_i^4 M} \frac{\partial s_i^2}{\partial \mathbf{x}} + \frac{1}{s_i^2 M^2} \sum_{i=1}^{q} \frac{1}{s_i^4} \frac{\partial s_i^2}{\partial \mathbf{x}} \right), \quad M = \sum_{j=1}^{q} \left(s_j^2 \right)^{-1}$$

The gradient of the Kriging predictor and MSE on each cluster, $\partial \hat{f}_i / \partial \mathbf{x}, \partial s_i^2 / \partial \mathbf{x}$, are given in Eq. (3.34) and (3.35). For the mixture of posterior processes, its gradient can be obtained in a similar way. This is omitted here for the sake of simplicity.

In addition, for the Tree-based local Kriging models (MTCK), it is shown (Fig. **??**) that each cluster (leaf node) can be treated as a sub-problem in the infill criteria optimization. Therefore, it might be more efficient to conduct independent searches in each leaf region of the Regression Tree and choose the best point from all these sub-problems.

3.3.2 Experiments

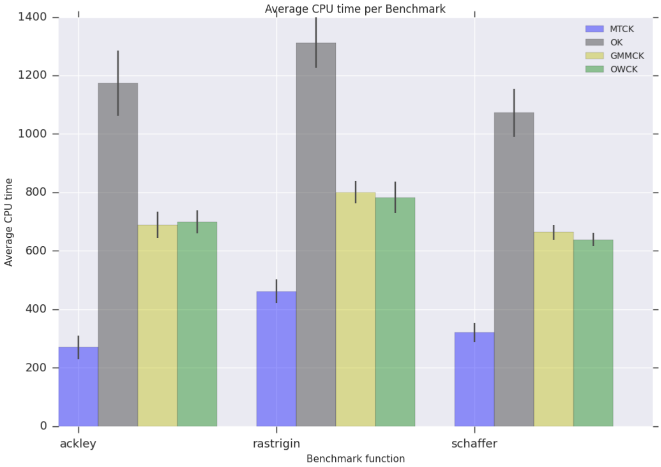
Several experiments are conducted to show both the empirical time complexity and convergence rate of the proposed Cluster Kriging based EGO, including all the variants of Cluster Kriging discussed in Section 3.2.3. The performance of the proposed algorithm is compared to the original EGO that uses *Ordinary Kriging* (OK). For our experiments, the benchmark functions chosen are *Ackley*, *Rastrigin* and *Schaffer*. These functions are chosen because they are used often in optimization experiments, are highly multi modal, and are of a relatively high complexity.

The algorithms under comparisons are: EGO with Ordinary Kriging (OK), Tree-based local Kriging models (MTCK), Superposition of Kriging models (OWCK) and the mixture of Kriging models (GMMCK). Each of the Cluster Kriging variants uses 5 clusters. Both execution time and convergence rate are being measured with a fixed set of EGO iterations and optimization budget. The convergence is measured by taking the absolute error between the real optimum of the benchmark functions and the found optimum for each iteration of EGO. Each EGO run performs 10 iterations for the three benchmark functions in two dimensions. Three different initial sample sizes (500, 1000 and 5000) are used to train the surrogate models, in order to illustrate the growth of CPU time required per algorithm, when the size of the data available increases. For each different experimental setup, the average time and distance to the optimum is recorded over 20 runs with different random seed ($[0, 20]$).

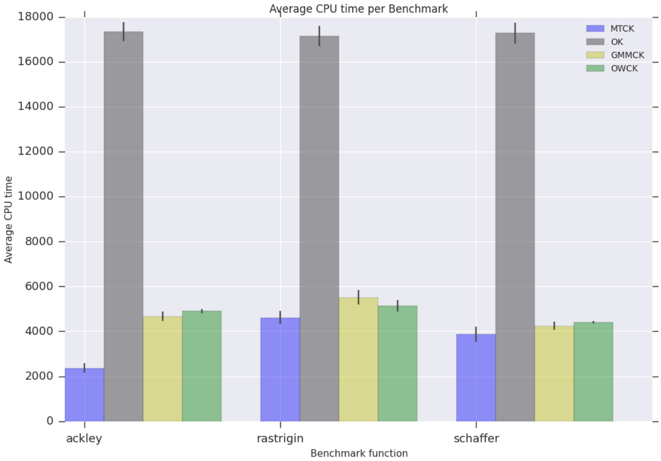
Results In Fig. 3.5 it can be observed that the Cluster Kriging based EGO variants perform very similar to OK, depending on the target function; a specific variant even outperforms Ordinary Kriging. Due to the relative large variance in the results it is difficult to judge which algorithm performs better. However, in terms of the CPU time (Fig. 3.4), it can be observed that Cluster Kriging and in particular MTCK takes only a fraction of the time that Ordinary Kriging requires. Using a sample size of 500 points this difference is mainly due to the re-fitting of only one local model at a time. This can be seen by comparing MTCK with GMMCK and OWCK, since all three Cluster Kriging variants use the same number of local models and only MTCK uses an adaptive local model strategy. When the number of points increases to 1000 and even 5000, the difference between the three Cluster Kriging variants decreases but the difference with Ordinary Kriging becomes enormous. This shows that using EGO with Ordinary Kriging quickly becomes infeasible when the number of data points grows.

3.4 Summary

This chapter addresses three aspects of the Kriging/GPR method. Firstly, a unified view of Kriging/GPR is provided: the same formulation can be derived independently from the theory of optimal linear predictors, Bayesian statistics and the optimization in Reproducing Kernel Hilbert Spaces (RKHS). We try to link those three approaches together and give a conceptual comparison among them. Secondly, the time complexity bottleneck of Kriging is discussed in detail, where several novel methods (Cluster Kriging variants) are derived, aiming at reducing the training time and increase the model quality. Finally, the proposed Cluster Kriging method is combined with the EGO algorithm to demonstrate its usefulness in improving the existing optimization algorithm.



(a) CPU time, sample size 1000



(b) CPU time, sample size 5000

Figure 3.4: Average CPU time in seconds per benchmark function for varying sample sizes.

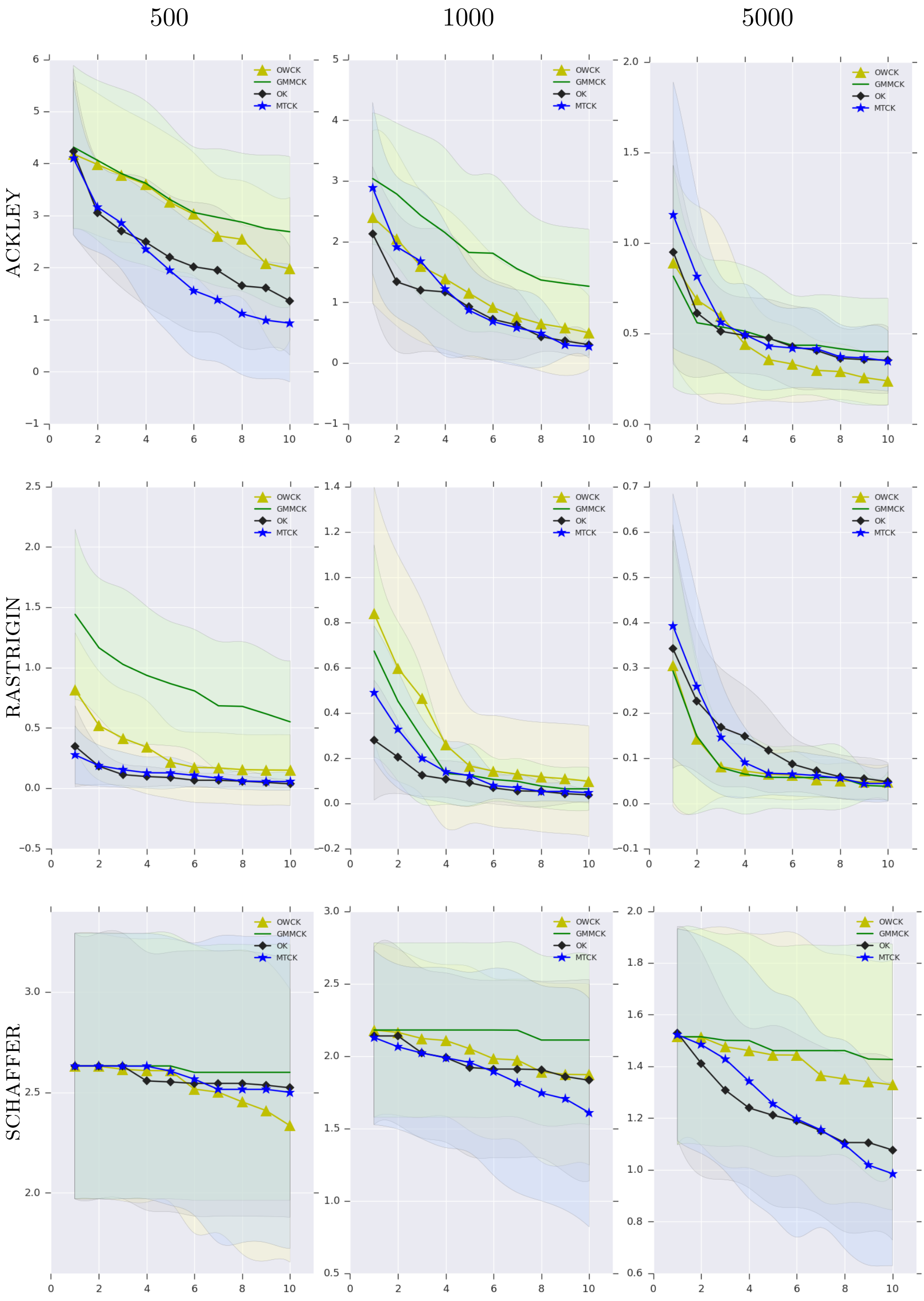


Figure 3.5: Average convergence of the absolute error of three benchmark functions in two dimensions, with varying training sample sizes n and 10 iterations of EGO. Shown is the average over 20 runs (lines) and one standard deviation (shaded areas). Legend: ◆: Ordinary Kriging, ▲: OWCK, ★: MTCK, –: GMMCK.