Kriging-based black-box global optimization: analysis and new algorithms

Hossein Mohammadi

NNT : 2016LYSEM005

THÈSE

présentée par

Hossein MOHAMMADI

pour obtenir le grade de

Docteur de l'École Nationale Supérieure des Mines de Saint-Étienne

Spécialité : Mathématiques Appliquées

Kriging-based black-box global optimization: analysis and new algorithms

Optimisation Globale et processus Gaussiens: analyse et nouveaux algorithmes

soutenue à Saint-Étienne, le 11/04/2016

Membres du jury

Président:	Hervé Monod	Senior Researcher, INRA JOUY-EN-JOSAS
Rapporteurs:	Sonja Kuhnt	Professor, FH Dortmund
	David Ginsbourger	Senior Researcher at Idiap and Dozent at the University of Bern
Examinateur:	Nicolas Durrande	Assistant Professor , Mines Saint-Étienne
Directeur de thèse:	Rodolphe Le Riche	CNRS Senior Researcher, Mines Saint-Étienne
Co-encadrant:	Eric Touboul	Research Engineer, Mines Saint-Étienne

Spécialités doctorales	Responsables :	Spécialités doctorales	Responsables
SCIENCES ET GENIE DES MATERIAUX	K. Wolski Directeur de recherche	MATHEMATIQUES APPLIQUEES	O. Roustant, Maître-assistant
MECANIQUE ET INGENIERIE	S. Drapier, professeur	INFORMATIQUE	O. Boissier, Professeur
GENIE DES PROCEDES	F. Gruy, Maître de recherche	IMAGE, VISION, SIGNAL	JC. Pinoli, Professeur
SCIENCES DE LA TERRE	B. Guy, Directeur de recherche	GENIE INDUSTRIEL	A. Dolgui, Professeur
SCIENCES ET GENIE DE L'ENVIRONNEMENT	D. Graillot, Directeur de recherche	MICROELECTRONIQUE	S. Dauzere Peres, Professeur

EMSE : Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'État ou d'une HDR)

ABSI	Nabil	CR	Génie industriel	CMP
AVRIL	Stéphane	PR2	Mécanique et ingénierie	CIS
BALBO	Flavien	PR2	Informatique	FAYOL
BASSEREAU	Jean-François	PR	Sciences et génie des matériaux	SMS
BATTAIA-GUSCHINSKAYA	Olga	CR	Génie industriel	FAYOL
BATTON-HUBERT	Mireille	PR2	Sciences et génie de l'environnement	FAYOL
BERGER DOUCE	Sandrine	PR2	Sciences de gestion	FAYOL
BIGOT	Jean Pierre	MR(DR2)	Génie des Procédés	SPIN
BILAL	Essaid	DR	Sciences de la Terre	SPIN
BLAYAC	Sylvain	MA(MDC)	Microélectronique	CMP
BOISSIER	Olivier	PR1	Informatique	FAYOL
BONNEFOY	Olivier	MA(MDC)	Génie des Procédés	SPIN
BORBELY	Andras	MR(DR2)	Sciences et génie des matériaux	SMS
BOUCHER	Xavier	PR2	Génie Industriel	FAYOL
BRODHAG	Christian	DR	Sciences et génie de l'environnement	FAYOL
BRUCHON	Julien	MA(MDC)	Mécanique et ingénierie	SMS
BURLAT	Patrick	PR1	Génie Industriel	FAYOL
COURNIL	Michel	PR0	Génie des Procédés	DIR
DAUZERE-PERES	Stéphane	PR1	Génie Industriel	CMP
DEBAYLE	Johan	CR	Image Vision Signal	CIS
DELAFOSSE	David	PR0	Sciences et génie des matériaux	SMS
DELORME	Xavier	MA(MDC)	Génie industriel	FAYOL
DESRAYAUD	Christophe	PR1	Mécanique et ingénierie	SMS
DOLGUI	Alexandre	PR0	Génie Industriel	FAYOL
DRAPIER	Sylvain	PR1	Mécanique et ingénierie	SMS
FAVERGEON	Loïc	CR	Génie des Procédés	SPIN
FEILLET	Dominique	PR1	Génie Industriel	CMP
FRACZKIEWICZ	Anna	DR	Sciences et génie des matériaux	SMS
GARCIA	Daniel	MR(DR2)	Génie des Procédés	SPIN
GAVET	Yann	MA(MDC)	Image Vision Signal	CIS
GERINGER	Jean	MA(MDC)	Sciences et génie des matériaux	CIS
GOEURIOT	Dominique	DR	Sciences et génie des matériaux	SMS
GONDRAN	Natacha	MA(MDC)	Sciences et génie de l'environnement	FAYOL
GRAILLOT	Didier	DR	Sciences et génie de l'environnement	SPIN
GROSSEAU	Philippe	DR	Génie des Procédés	SPIN
GRUY	Frédéric	PR1	Génie des Procédés	SPIN
GUY	Bernard	DR	Sciences de la Terre	SPIN
HAN	Woo-Suck	MR	Mécanique et ingénierie	SMS
HERRI	Jean Michel	PR1	Génie des Procédés	SPIN
KERMOUCHE	Guillaume	PR2	Mécanique et Ingénierie	SMS
KLOCKER	Helmut	DR	Sciences et génie des matériaux	SMS
LAFOREST	Valérie	MR(DR2)	Sciences et génie de l'environnement	FAYOL
LERICHE	Rodolphe	CR	Mécanique et ingénierie	FAYOL
LI	Jean-Michel		Microélectronique	CMP
MALLIARAS	Georges	PR1	Microélectronique	CMP
MAURINE	Philippe	Ingénieur de recherche	Microélectronique	CMP
MOLIMARD	Jérôme	PR2	Mécanique et ingénierie	CIS
MONTHEILLET	Frank	DR	Sciences et génie des matériaux	SMS
MOUTTE	Jacques	CR	Génie des Procédés	SPIN
NEUBERT	Gilles	PR	Génie industriel	FAYOL
NIKOLOVSKI	Jean-Pierre	Ingénieur de recherche		CMP
NORTIER	Patrice	PR1		SPIN
OWENS	Rosin	MA(MDC)	Microélectronique	CMP
PICARD	Gauthier	MA(MDC)	Informatique	FAYOL
PIJOLAT	Christophe	PR0	Génie des Procédés	SPIN
PIJOLAT	Michèle	PR1	Génie des Procédés	SPIN
PINOLI	Jean Charles	PR0	Image Vision Signal	CIS
POURCHEZ	Jérémy	MR	Génie des Procédés	CIS
ROBISSON	Bruno	Ingénieur de recherche	Microélectronique	CMP
ROUSSY	Agnès	MA(MDC)	Génie industriel	CMP
ROUSTANT	Olivier	MA(MDC)	Mathématiques appliquées	FAYOL
ROUX	Christian	PR	Image Vision Signal	CIS
STOLARZ	Jacques	CR	Sciences et génie des matériaux	SMS
TRIA	Assia	Ingénieur de recherche	Microélectronique	CMP
VALDIVIESO	François	PR2	Sciences et génie des matériaux	SMS
VIRICELLE	Jean Paul	DR	Génie des Procédés	SPIN
WOLSKI	Krzystof	DR	Sciences et génie des matériaux	SMS
XIE	Xiaolan	PR1	Génie industriel	CIS
YUGMA	Gallian	CR	Génie industriel	CMP

ENISE : Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'État ou d'une HDR)

BERGHEAU	Jean-Michel	PU	Mécanique et Ingénierie	ENISE
BERTRAND	Philippe	MCF	Génie des procédés	ENISE
DUBUJET	Philippe	PU	Mécanique et Ingénierie	ENISE
FEULVARCH	Eric	MCF	Mécanique et Ingénierie	ENISE
FORTUNIER	Roland	PR	Sciences et Génie des matériaux	ENISE
GUSSAROV	Andrey	Enseignant contractuel	Génie des procédés	ENISE
HAMDI	Hédi	MCF	Mécanique et Ingénierie	ENISE
LYONNET	Patrick	PU	Mécanique et Ingénierie	ENISE
RECH	Joël	PU	Mécanique et Ingénierie	ENISE
SMUROV	Igor	PU	Mécanique et Ingénierie	ENISE
TOSCANO	Rosario	PU	Mécanique et Ingénierie	ENISE
ZAHOUANI	Hassan	PU	Mécanique et Ingénierie	ENISE

Mise à jour : 29/06/2015

Acknowledgments

First of all, I would like to thank my advisor Dr. Rodolphe Le Riche for his advice and guidance. I have learned a great deal about Gaussian processes and optimization from him.

I would like to express my gratitude to my co-advisor Eric Touboul. Working with Eric has been a wonderful experience and I have such good memories, even now that I am thinking of him I am smiling. We had very interesting discussions during the coffee breaks and he made me laugh.

I would like to extend my heartfelt gratitude and sincere appreciation to Christine Exbrayat. Christine! I don't know how to say thank you enough; I just say: "Je te suis profondément reconnaissant pour ce que tu as fait pour moi. Une chose est sure: je n'oublierais jamais".

I am grateful to all the people from DEMO department: Paolo, Mireille, Xavier Bay, Nicolas, Didier, Hassan, Mickael, Esperan and all the others. I also thank all my dear friends, especially my nice Iranian friends in Saint Etienne, for the good times.

Last but certainly not least, I have to express my very profound gratitude to my family for supporting me for everything.

Nomenclature

Abbreviations

CMA-ES, Covariance Matrix Adaptation Evolution Strategy

CV, Cross-validation

discr, model-data discrepancy

EGO, Efficient Global Optimization

EI, Expected Improvement

GP, Gaussian Process

ML, Maximum Likelihood

PI, Pseudoinverse

Greek symbols

τ^2, nugget value.

Δ, the difference between two likelihood functions.

$\delta(.,.)$, Kronecker delta.

ϵ, noise term.

κ, condition number of a matrix.

κ_{max}, maximum condition number after regularization.

λ_i, the ith largest eigenvalue of the covariance matrix.

$\mu(.)$, Gaussian process mean.

$\Phi(.)$, standard normal distribution function.

$\phi(.)$, standard normal density function.

ω_i, ith weight of a linear combination.

σ^2, process standard deviation, step-size.

σ^2, process variance.

$\mathbf{\Sigma}$, diagonal matrix made of covariance matrix eigenvalues.

η, tolerance of pseudoinverse.

θ_i, characteristic length-scale in dimension i.

Latin symbols

$\mathbf{c}$, vector of covariances between a new point and the design points $\mathbf{X}$.

$\mathbf{C}$, covariance matrix.

$\mathbf{C}^i$, ith column of $\mathbf{C}$.

$\mathbf{e}_i$, ith unit vector.

$f : \mathbb{R}^d \to \mathbb{R}$, true function, to be predicted.

$\mathbf{H}$, Hessian matrix.

$\mathbf{I}$, identity matrix.

K, kernel or covariance function.

$m(.)$, kriging mean.

$\mathbf{m}$, mean of a multivariate normal distribution.

n, number of design points.

N, number of redundant points.

$\mathbf{P}_{Im}$, orthogonal projection matrix onto the image space of a matrix (typically $\mathbf{C}$).

$\mathbf{P}_{Nul}$, orthogonal projection matrix onto the null space of a matrix (typically $\mathbf{C}$).

$\mathbf{R}$, correlation matrix.

r, rank of the matrix $\mathbf{C}$.

$s^2(.)$, kriging variance.

s_i^2, variance of response values at i-th repeated point.

$\mathbf{V}$, column matrix of eigenvectors of $\mathbf{C}$ associated to strictly positive eigenvalues.

$\mathbf{W}$, column matrix of eigenvectors of $\mathbf{C}$ associated to zero eigenvalues.

$\mathbf{X}$, matrix of design points.

$Y(.)$, Gaussian process.

$\mathbf{y}$, vector of response or output values.

$\overline{y_i}$, mean of response values at i-th repeated point.

Contents

Chapter 1

Introduction to black-box optimization

1.1 Problem statement

Optimization (alternatively, mathematical programming) is a field of mathematics that studies the problem of finding the best choice(s) among a set of candidate choices. Such decisions appear in many domains like biology, economics, geophysics, or mechanics and are thus at the core of many challenges in our society. For instance, *worst-case analysis* in engineering design is done by solving a mathematical programming problem. Here the problem is to find the *worst-case* values of design parameters in order to conservatively check the performance of a safety-critical system. It is then possible to decide whether the system is safe or reliable with respect to the parameter variations [BV04].

A single objective optimization problem can be formulated in the following way

$$\min_{\mathbf{x}\in\mathcal{S}} f(\mathbf{x}), \tag{1}$$

where $f : \mathcal{S} \longmapsto \mathbb{R}$ is the objective function (or fitness function) and $\mathcal{S}$ is the search space (or solution space). Sometimes the problem is to find a point with the highest function value. To do so, it is just enough to minimize the negative of the objective function $\min_{\mathbf{x}\in\mathcal{S}} \; -f(\mathbf{x})$.

The solution(s) of the above problem denoted by $\mathbf{x}^*$ might be local or global. It is said that $\mathbf{x}^*$ is a local minimum if $f(\mathbf{x}) \geq f(\mathbf{x}^*)$ for all $\mathbf{x}$ in a neighborhood of $\mathbf{x}^*$. If the relation $f(\mathbf{x}) \geq f(\mathbf{x}^*)$ holds for all $\mathbf{x}$ in the search space $\mathcal{S}$, then $\mathbf{x}^*$ is a global minimum. The function illustrated in Figure 1.1 has several local optima (multimodal function) but the one denoted by x^* is the global optimum. In this thesis, we are interested in finding the

global optimum (minimum) of functions.

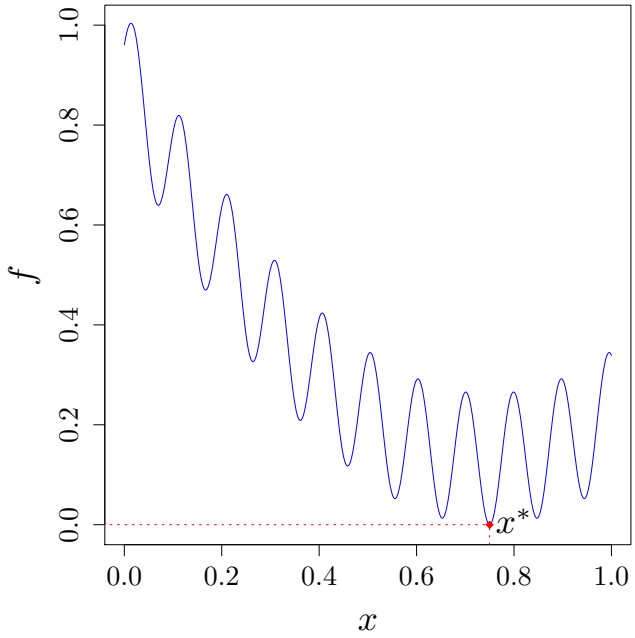


Figure 1.1: Illustration of a multimodal function with several local optima and one global minimum shown by x^*.

When the search space is a subset of $\mathbb{R}^d$, i.e., $\mathcal{S} \subseteq \mathbb{R}^d$, the optimization problem is called continuous. The other types of optimization problems with respect to the set $\mathcal{S}$ are called integer or mixed integer programming. In integer programming $\mathcal{S}$ is finite or countable while a mixed integer programming, as the name indicate, is a mixture of continuous and integer programming. Throughout this work we only deal with continuous problems.

The search space is mathematically defined by constraints. In general, there are three types of constraints:

1. *Box-constraints*: $\mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u$, in which $\mathbf{x}_l$ and $\mathbf{x}_u$ are the vectors of the lower and upper bounds.

2. *Equality constraints*: $h_i(\mathbf{x}) = 0\ , \quad i = 1, \ldots, k,$

3. *Inequality constraints*: $g_j(\mathbf{x}) \leq 0\ , \quad j = 1, \ldots, l,$

where $h_i(\mathbf{x})$ and $g_j(\mathbf{x})$ are functions that map the elements of $\mathcal{S}$ into $\mathbb{R}$. If the search space is not limited by any constraints, the problem is called unconstrained optimization.

The optimization problems we consider in this work have box-constraints. So, the search space is *compact* because it is closed (containing all its limit points) and bounded. With the assumptions that $\mathcal{S}$ is continuous and compact the existence of a global minimum and a global maximum is guaranteed based on the Weierstrass theorem [Rud76].

Weierstrass Theorem Let $\mathcal{S}$ be a compact subset of $\mathbb{R}^d$ and the function f be continuous on $\mathcal{S}$. Then there exists $\mathbf{x}_1$ and $\mathbf{x}_2$ in $\mathcal{S}$ such that $f(\mathbf{x}_1) \leq f(\mathbf{x}) \leq f(\mathbf{x}_2)\ \forall \mathbf{x} \in \mathcal{S}$. $\mathbf{x}_1$ is a point of global minimum and $\mathbf{x}_2$ is a point of global maximum of f.

Optimization problems can also be classified based on the type of objective function and constraints. If the objective and all constraint functions of a mathematical programming are linear with respect to $\mathbf{x}$ it is called linear optimization. Obviously, in nonlinear optimization the objective or constraint functions are not linear. Another important class of optimization problems is convex optimization (versus non-convex), in which the objective and all constraint functions are convex. In convex optimization problems any locally optimal point is globally optimal. Interested readers are referred to [BV04] for further information about convex optimization. It is worthy to note that linear optimization is convex whereas the convexity of a nonlinear optimization has to be tested.

1.2 Context: Black-box optimization of expensive-to-evaluate functions

Today, numerical simulations are powerful tools to model complex phenomena because they are typically faster and less expensive than the physical experiments. In order to find the best input parameters of a simulator, black-box optimization, which is also referred to as direct-search or derivative-free optimization, is a (large) family of methods of choice. Indeed, it frequently happens in practice that the function to be optimized is given as an executable code only. Even when the source code is available, it is often preferable to handle the problem as a black-box problem: as the difficulties/complexities of real-world problems are generally unknown a priori, designing a problem- specific optimization technique is often (technically/financially) infeasible.

In this work we focus on the optimization of black-box functions which are computa-

tionally expensive. It means that each function evaluation takes from a few minutes to a few hours and therefore the number of function evaluations is limited. High-fidelity computer simulations such as computational fluid dynamics (CFD) or finite elements analysis (FEA) are examples of such time-consuming black-box functions.

When optimizing such expensive functions, it is critical to converge towards the global optimum with the least possible number of function evaluations. Accordingly, algorithms such as Genetic algorithms [Hol75] are not efficient enough in this setting because they require several thousands of function calls.

One approach to alleviate the computation cost and speed-up the optimization procedure is to approximate the underlying function with a simpler model, known as a surrogate model or a metamodel. The surrogate models are usually defined everywhere in the design space and built on statistical grounds from a finite set of model input-outputs [FSK08, Jon01, QHS+05]. Kriging [Cre93], polynomial regression [RPD98], artificial neural network (ANN) [Bis95], and support vector regression [SS04] are common surrogate models. The principle of surrogate based optimization is to substitute a part of the calls to the expensive simulations by calls to the less computationally intensive (once it is built) surrogate.

All surrogate-based optimization methods share the following steps [FK09]:

1. Choose the design variables, i.e., the variables to be optimized over.
2. Start with an initial design of experiments (DoE) and evaluate the objective function at the selected points.
3. Construct the surrogate model on the data points.
4. Search for a new design point(s) in the space of design variables.
5. Add the new data point(s) to the available sampled points and update the surrogate model.
6. Iterate through steps 3 and 5 until a stopping criterion is met.

In step (2), the DoE is often space filling because the true function is generally unknown a priori. To achieve this property, one should create the initial samples according to a sampling plan like Latin hypercubes [MBC00]. Other sampling schemes include factorial,

fractional factorial and central composite designs [Mon01], and low discrepancy sequences (such as Halton and Sobol sequences [Hal60, Sob67]).

The *size of the initial DoE* is an algorithm's setting that reflects typical choices to be made in black-box optimization. If it is small, the surrogate model may not be able to satisfactorily represent the true function and the approximation could be misleading. This is especially the case for the multimodal landscapes. Conversely, starting the optimization with a large number of sample points may waste expensive function evaluations by naively filling non-optimal regions of the search space. As a "rules of thumb" it is recommended by [JSW98] in the framework of kriging-based optimization that the size of initial DoE should be linear in the number of dimensions d, more precisely $10 \times d$. Reference [SLK05] suggests that a safe choice for initial DoE is about 35% of the total computational budget.

In step (4), the strategy used for selecting a new infill sample should be an appropriate trade-off between local exploitation of promising basins of attraction and global exploration of landscape. During local search, the vicinity of promising points is examined to achieve with low risk further improvement in the objective function value. The global component of the search contributes not only to reduce the model uncertainty where there is less information but also to escape from local optima. For more details see Section 2.3.2.

1.3 Classification and review of black-box optimizers

In black-box optimization, as the name implies, we do not have access to the analytical formula of the objective function. The only available information to seek for the optimum is the function value for a given input variable vector. Moreover, in this thesis, in agreement with most practical situations, we will assume that derivative information is not available or practically impossible to obtain. This is the case when the objective function is, for example, expensive to evaluate or noisy. Thus, the algorithms developed for solving black-box optimization problems must rely only on the objective function evaluations.

Many black-box optimization algorithms exist in the literature. They can be classified based on different criteria [RS13]:

(i) *Direct* vs. *model-based*: In direct algorithms the search direction is determined by computing objective function values directly. For instance, at each iteration of the Nelder-Mead method [NM65] a simplex is formed and the objective function cal-

culated at its vertices. Future simplexes are shaped from the order of the objective functions values. When the function evaluations are costly or time-consuming, model-based algorithms are usually preferred. In model-based algorithms a surrogate model of the objective function is constructed, $\hat{f}$. Then, the surrogate model is used to guide the optimization of the true function. Any mismatch between f and $\hat{f}$ is assumed to be the model error and not noise [RS13]. The Efficient Global Optimization (EGO) [JSW98] is one of the most important model-based global optimization algorithms. EGO uses a conditional Gaussian process to approximate the true function f from a set of observations and sequentially adds new points which maximize the "expected improvement" in f. Detailed explanations about EGO are provided in Chapter 2.

(ii) *Local* search vs. *global* search: Local optimization algorithms start from an initial point and iteratively move in a neighborhood around the current best point [HS04] until a stopping criterion is met. Global search algorithms opportunistically explore the whole volume of $\mathcal{S}$ in order to locate the global optimum. An example is provided by the *DIRECT* algorithm, which comes from the shortening of the phrase "DIviding RECTangles", by Jones et al. [JPS93], where rectangles pave the entire search space and are divided as new values of the objective function are calculated. Since the search domain of local algorithms is smaller, they are usually faster than global ones. A local optimizer can be transformed into global if it is repeated several times from different initial points [LLR04].

(iii) *Stochastic* vs. *deterministic*: Stochastic algorithms use transitions in probability (whether explicit or not) to search while deterministic algorithm always reproduce the same opportunist steps. The final solutions obtained by a stochastic method change when the run is repeated.The randomness of stochastic methods makes them capable of escaping local optima areas. But this is at the price of slowing down the optimization procedure.

Lipschitzian-based partitioning techniques (DIRECT, [JPS93]) are deterministic algorithms while evolution strategies such as Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [HO01] are stochastic methods. CMA-ES will be explained with more details in Chapter 4.

We now briefly describe three black-box (derivative free, continuous) optimization al-

gorithms different in their principles from EGO and CMA-ES which will receive more attention in Chapters 2 and 4, respectively. There exist many other useful black-box algorithms like e.g., Cross-entropy Optimization or EMNA (Estimation of Multivariate Normal Algorithm) [dBKMR05, SD98], Simulated Annealing [KGV83], ... Describing all these methods is out of the scope of this manuscript.

Nelder-Mead simplex method. The Nelder-Mead method or downhill simplex method was first proposed by John Nelder and Roger Mead [NM65]. It is a deterministic and local direct search algorithm. This method is a heuristic method, i.e., there is no guarantee that the algorithm converges to stationary points. A larger, related class of algorithms are the pattern search methods [AJ02] for which convergence to stationary points on discontinuous functions is established but which are, arguably, slower and less utilized. Note that pattern search and Nelder Mead algorithms perform local searches (although they can visit many basin of attraction when the size of their pattern is large enough). The Nelder-Mead method starts with a set of points that form a simplex. In an n-dimensional space, a simplex is a convex hull of $n+1$ points. A line segment on a line and a triangle on a plane are examples of a simplex in 1 and 2 dimensional spaces, respectively.

At each iteration, the objective function is evaluated at the vertices of the simplex and they are ranked from best to worst. The worst corner is replaced by another vertex whose new function value is calculated and a new simplex is formed. The candidate vertex is obtained by transforming the worst corner through the centroid of the remaining n points. The main transformation operations are *reflection*, *expansion* and *contraction* by which the simplex can move towards the optimum. See Figure (1.2) for more details.

Trust regions methods. Trust regions methods [Pow02, Pow09] are model-based local search algorithms. The procedure is such that a surrogate model (often a quadratic $Q(\mathbf{x})$) of the objective function around the current k-th iterate, $\mathbf{x}_k$, is constructed. Since the surrogate model may not well represent the objective function in regions far away from $\mathbf{x}_k$, the search for the optimum of the model is restricted to a "trusted" region around $\mathbf{x}_k$. This region is usually a ball defined as

$$\forall\, \mathbf{x} \in \mathbb{R}^n \;: \|\mathbf{x} - \mathbf{x}_k\| \leq \Delta_k, \tag{2}$$

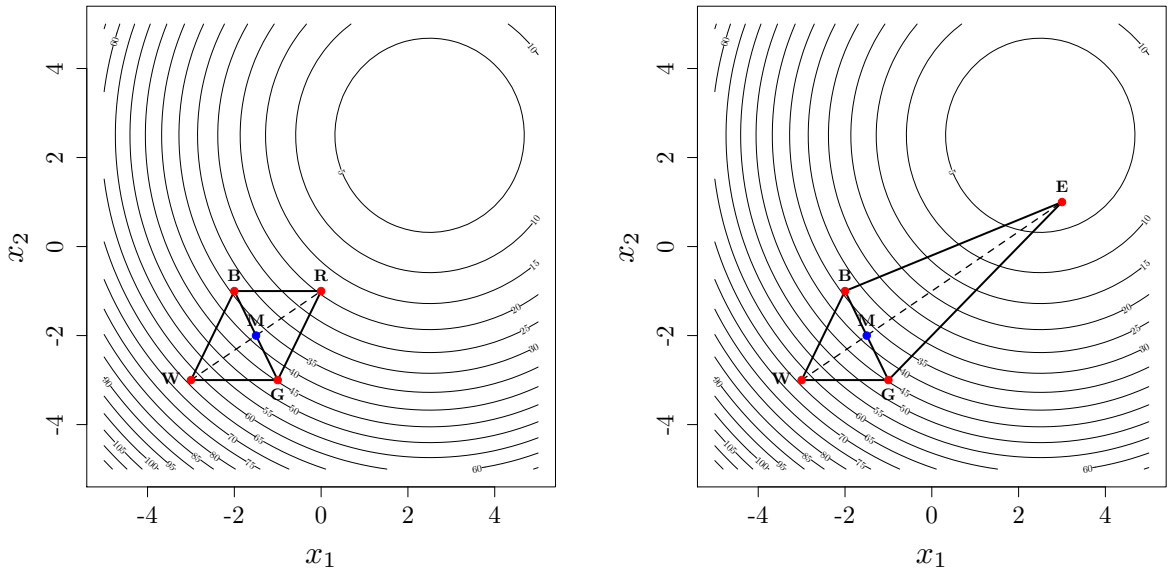


Figure 1.2: A 2-dimensional illustration of the Nelder-Mead method. The function to be optimized is Sphere function with a minimum at point $(2.5, 2.5)$. The triangle BGW is the initial simplex where the rank of the vertices are: B (best), G (good) and W (worst). M is the centroid of B and G. Left: the next simplex is BGR in which R is obtained by reflection operation. Right: the next simplex is BGE where R is obtained through expansion operation.

in which Δ_k is the radius of the ball. The next query point, $\mathbf{x}_{k+1}$, is where the model reaches its optimum within the trust region. Then, the actual and predicted improvements, denoted by Δf_{actual} and $\Delta f_{predict}$, are calculated:

$$\Delta f_{actual} = f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}), \tag{3}$$

$$\Delta f_{predict} = Q_k(\mathbf{x}_k) - Q_k(\mathbf{x}_{k+1}). \tag{4}$$

Finally, the trust region size is adapted based on the ratio $\rho_k = \frac{\Delta f_{actual}}{\Delta f_{predict}}$ as follows [RKL14]

- If $\rho_k \geq \frac{3}{4}$, the model is in good agreement with the objective function and we can expand the trust region size in the next iteration.
- If $\rho_k \leq \frac{1}{4}$, the trust region size should be shrinked in the next iteration.
- Otherwise, the size of the trust region is good and it is better to keep it.

Lipschitzian-based partitioning techniques. Here, we describe Shubert's algorithm [Shu72] which uses the information obtained from Lipschitz continuity of functions to seek the optimum. In mathematical analysis, the function f is called Lipschitz-continuous if there exist a positive constant $L > 0$ such that the inequality

$$|f(\mathbf{x}) - f(\mathbf{x}')| \leq L\,|\mathbf{x} - \mathbf{x}'|\,, \tag{5}$$

holds for all $\mathbf{x}$ and $\mathbf{x}'$ in the domain of f. Suppose that the function f defined in $[a, b]$ is Lipschitz-continuous.
According to Equation (5), f must satisfy the following two inequalities

$$\begin{aligned} f(x) &\geq f(a) - L(x - a) \\ f(x) &\geq f(b) + L(x - b). \end{aligned} \tag{6}$$

for every $x \in [a, b]$. The lines corresponding to the above two inequalities form a V-shape beneath f as it is shown in Figure 1.3. The point of intersection of the two lines, $(x_1, f(x_1))$ in the figure, is considered as the first estimate of the function's optimum. The algorithm continues by performing the same procedure on the intervals $[a, x_1]$ and $[x_1, b]$, dividing next the one with the lower function value at the intersection. The DIRECT algorithm [JPS93] is an extension of the Schubert algorithm to many dimensions where all the possible Lipschitz constants L are accounted for.

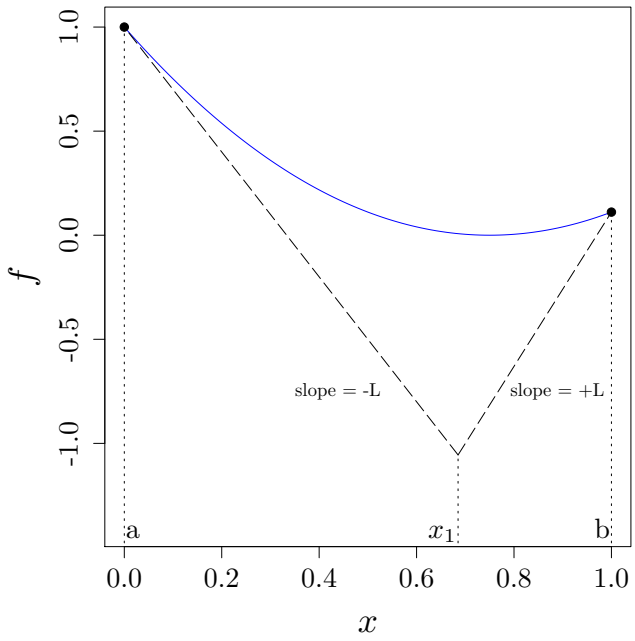


Figure 1.3: First iteration of Shubert's algorithm. The function f, blue line, is Lipschitz-continuous in $[a, b]$ with constant L. The dashed lines are correspond to the inequalities defined in (6) and x_1 is the intersection. x_1 is the first estimate of the minimum of f.

1.4 Motivations

The Efficient Global Optimization (EGO) algorithm is a mathematically well funded and often used method for the unconstrained global optimization of expensive-to-calculate functions. It relies on a Gaussian process model of the true function (and will be presented in details in Chapter 2). Since it was proposed in 1998 [JSW98], many alternative versions of the method have been investigated [Jon01, VVW09, GLRC10, CH14] that have moved forward towards new infill sampling criteria (see Section 2.3.2) and new capabilities like parallel optimization. However, many aspects of the classical EGO method are not well understood or still deserve improvements for practical applications. This PhD thesis addresses the following questions:

– How to deal with ill-conditioning in Gaussian Processes? Optimization with Gaussian Processes systemically leads to ill-conditioning of the covariance matrix which must be inverted as optimization iterates gather in tight clusters at high performance

regions of the design space. After providing a new algebraic comparison of two classical regularization methods, we propose a novel approach to overcome the degeneracy of the covariance matrix in GPs.

– How do CMA-ES and EGO algorithms, two state-of-the-art global optimization approaches, compare? The EGO which is commonly used for the optimization of expensive-to-evaluate functions lacks the accurate convergence to the optimum exhibited by CMA-ES. After comparing the two methods, our contribution to improve the convergence of EGO is to combine it with CMA-ES in a warm start approach.

– In the field of optimization with surrogate, a general and important question is which surrogate most efficiently leads the search to the optimum? This question, transposed to EGO method, translates to: what is the effect of the GP parameters on the convergence of the EGO algorithm? Because the parameters are usually learned by statistical estimations such as maximum likelihood or cross-validation error, this question has not really been investigated. Yet, previous works in other fields have observed that the best surrogate for an optimization algorithm is not the one that corresponds the most closely to the true function [Los13, OZL06]. In this thesis, we carefully analyze the effect of the GP parameters on the EGO performance. To the best of our knowledge, there is no such a comprehensive study in the literature.

– In the EGO algorithm the GP parameters are estimated by statistical methods such as maximum likelihood or cross-validation. It is still an open question to know if these statistical approaches are the most appropriate in the context of optimization. Indeed, at the beginning of the search, very little is known about the true function and it is not clear that statistical approaches are the most appropriate. Furthermore, as stated above, there is no agreement in the optimization with surrogate literature that the best surrogate should match the true function (which is the purpose of statistical learning). To investigate this question, we propose and study a new learning method in which the parameters of the GP are adapted solely based on the optimization convergence, without maximum likelihood or cross-validation.

1.5 Thesis outline

Chapter 1 introduces the field of continuous black-box optimization and reviews some algorithms developed to tackle such problems. It proceeds with the research objectives of this dissertation.

Chapter 2 is an introduction to the Gaussian processes, kriging model and EGO which is the algorithm mainly used in this work. It provides the necessary materials for the next chapters.

Chapter 3 contributes to a better theoretical and practical understanding of the impact of regularization strategies on GP regression. Differences between pseudo-inverse and nugget regularizations are mathematically proven. A new regularization approach based on a new distribution-wise GP is presented. Practical guidelines for choosing a regularization strategy in GP regression ensue.

Chapter 4 first presents the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) which is regarded as the state-of-the-art unconstrained continuous optimization algorithm. Then, the search principles of EGO and CMA-ES are compared. Finally, a new algorithm called EGO-CMA is introduced which has advantages of both algorithms.

Chapter 5 theoretically and empirically analyzes the effect of length-scale covariance parameters and nugget on the design of experiments generated by EGO and the associated optimization performance.

Chapter 6 proposes a new self-adaptive EGO where the parameters of the GP are directly learned from their contribution to the optimization.

Chapter 7 concludes the dissertation and proposes potential extensions for future research.

Chapter 2

EGO: using Gaussian processes as surrogates

2.1 Gaussian processes

Gaussian processes define a probability distribution over functions. They can be seen as generalization of the multivariate normal distribution to infinitely many dimensions where the input vector $\mathbf{x}$ plays the role of index. Formally, a GP is a collection of random variables, any finite number of which have a multivariate Gaussian distribution [RW05].

A GP is fully determined by a mean and a covariance function (or *kernel*). Consider a Gaussian process Y with mean and covariance function denoted by $\mu(.)$ and $K(.,.)$, respectively. They are defined as:

$$Y \sim \mathcal{GP}(\mu, K): \quad \mu(\mathbf{x}) = \mathrm{E}(Y(\mathbf{x})) \ , \ \ K(\mathbf{x}, \mathbf{x}') = \mathrm{Cov}(Y(\mathbf{x}), Y(\mathbf{x}')). \tag{1}$$

By definition, for any $n \in \mathbb{N}$ and any set $\mathbf{X} = (\mathbf{x}^1, \ldots, \mathbf{x}^n)$ of input vectors, the associated n-dimensional vector $\mathbf{Y} = (Y(\mathbf{x}^1), ..., Y(\mathbf{x}^n))$ has a multivariate Gaussian distribution:

$$\mathbf{Y} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{C}) \ , \ \boldsymbol{\mu} = \begin{bmatrix} \mu(\mathbf{x}^1) \\ \vdots \\ \mu(\mathbf{x}^n) \end{bmatrix} \ , \ \mathbf{C} = \begin{bmatrix} K(\mathbf{x}^1, \mathbf{x}^1) & \ldots & K(\mathbf{x}^1, \mathbf{x}^n) \\ \vdots & \ddots & \vdots \\ K(\mathbf{x}^n, \mathbf{x}^1) & \ldots & K(\mathbf{x}^n, \mathbf{x}^n) \end{bmatrix}.$$

Now we wish to calculate the probability distribution at some new points $\{\mathbf{X}_*, \mathbf{Y}_*\}$, while $\{\mathbf{X}, \mathbf{Y}\}$ is given. The vector $\mathbf{Y}_*$ has the following normal distribution:

$$Y_* \sim \mathcal{N}(\boldsymbol{\mu}_*, \mathbf{C}_{**} = K(\mathbf{X}_*, \mathbf{X}_*)). \tag{2}$$

Accordingly, the joint probability distribution of $\mathbf{Y}$ and $\mathbf{Y}_*$ is

$$\begin{bmatrix} \mathbf{Y} \\ \mathbf{Y}_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}_* \end{bmatrix}, \begin{bmatrix} \mathbf{C} & \mathbf{C}_* \\ \mathbf{C}_*^\top & \mathbf{C}_{**} \end{bmatrix} \right), \tag{3}$$

where $\mathbf{C}_* = K(\mathbf{X}, \mathbf{X}_*)$. Finally, the conditional distribution of $\mathbf{Y}_*$ given $\mathbf{Y}$ can be expressed as (see e.g.,[BCdF09])

$$p\left(\mathbf{Y}_* | \mathbf{Y}\right) = \mathcal{N}\left(\boldsymbol{\mu}_* + \mathbf{C}_*^\top \mathbf{C}^{-1}(\mathbf{Y} - \boldsymbol{\mu}), \mathbf{C}_{**} - \mathbf{C}_*^\top \mathbf{C}^{-1} \mathbf{C}_*\right). \tag{4}$$

As can be seen, various quantities can be obtained from the normal distribution properties. This makes GPs an important tool in statistical learning.

The structure of a GP's sample path such as smoothness and periodicity is determined by its kernel. Since covariance functions are closed under addition and multiplication, it is possible to create sophisticated structures through combining them [DNR11, DGR12]. Covariance functions are positive definite symmetric functions and hence, the associated covariance matrices are positive semidefinite: $v^\top \mathbf{C} v \geq 0 \;\; \forall v \in \mathbb{R}^n$. Table 2.1 presents some well-known covariance functions in which σ^2 is the process variance. In the expression for the Matérn kernel, Γ is the Gamma function and H_ν is the modified Bessel function of the second kind of order ν.

Table 2.1: Some covariance functions used in GP modeling

Covariance function	Expression
Matérn	$\frac{\sigma^2 2^{1-\nu}}{\Gamma(\nu)} \left[\frac{\sqrt{2\nu}}{\theta}\|\mathbf{x}-\mathbf{x}'\|\right]^\nu H_\nu\left(\frac{\sqrt{2\nu}}{\theta}\|\mathbf{x}-\mathbf{x}'\|\right)$
Squared exponential	$\sigma^2 \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\theta^2}\right)$
Exponential	$\sigma^2 \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|}{\theta}\right)$
Power exponential	$\sigma^2 \exp\left(-\left(\frac{\|\mathbf{x}-\mathbf{x}'\|}{\theta}\right)^{\theta'}\right),\ 0 < \theta' \leq 2$

The covariance functions introduced in Table 2.1 are called *stationary* because their values depend only on the Euclidean distance between their input vectors. Note that a

stationary covariance function is translation invariant. A stationary covariance function is said *isotropic* (or homogeneous) [Gen02], if it depends only on the Euclidean norm between $\mathbf{x}$ and $\mathbf{x}'$, i.e., $K(\mathbf{x}, \mathbf{x}') = K(\|\mathbf{x} - \mathbf{x}'\|)$. Otherwise, it is said *anisotropic*. For example, a stationary anisotropic squared exponential covariance function is presented below:

$$K(\mathbf{x}, \mathbf{x}') = \sigma^2 \prod_{i=1}^{d} \exp\left(-\frac{|x_i - x_i'|^2}{2\theta_i^2}\right). \tag{5}$$

In Equation (5), the parameters $\theta_i > 0$, $i = 1, \ldots, d$ are called *characteristic length-scale* and determine the correlation strength between $Y(\mathbf{x}^i)$'s. The smaller θ_i, the least two response values at given points are correlated in coordinate i, and vice versa. Figure 2.1 shows sample paths of a GP with two different length-scale.

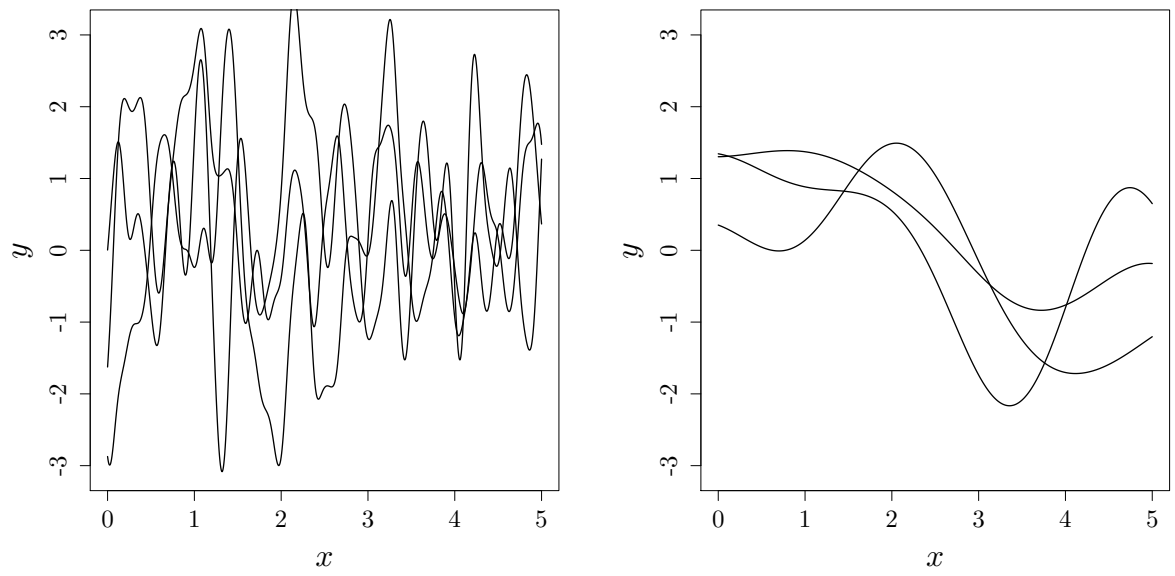
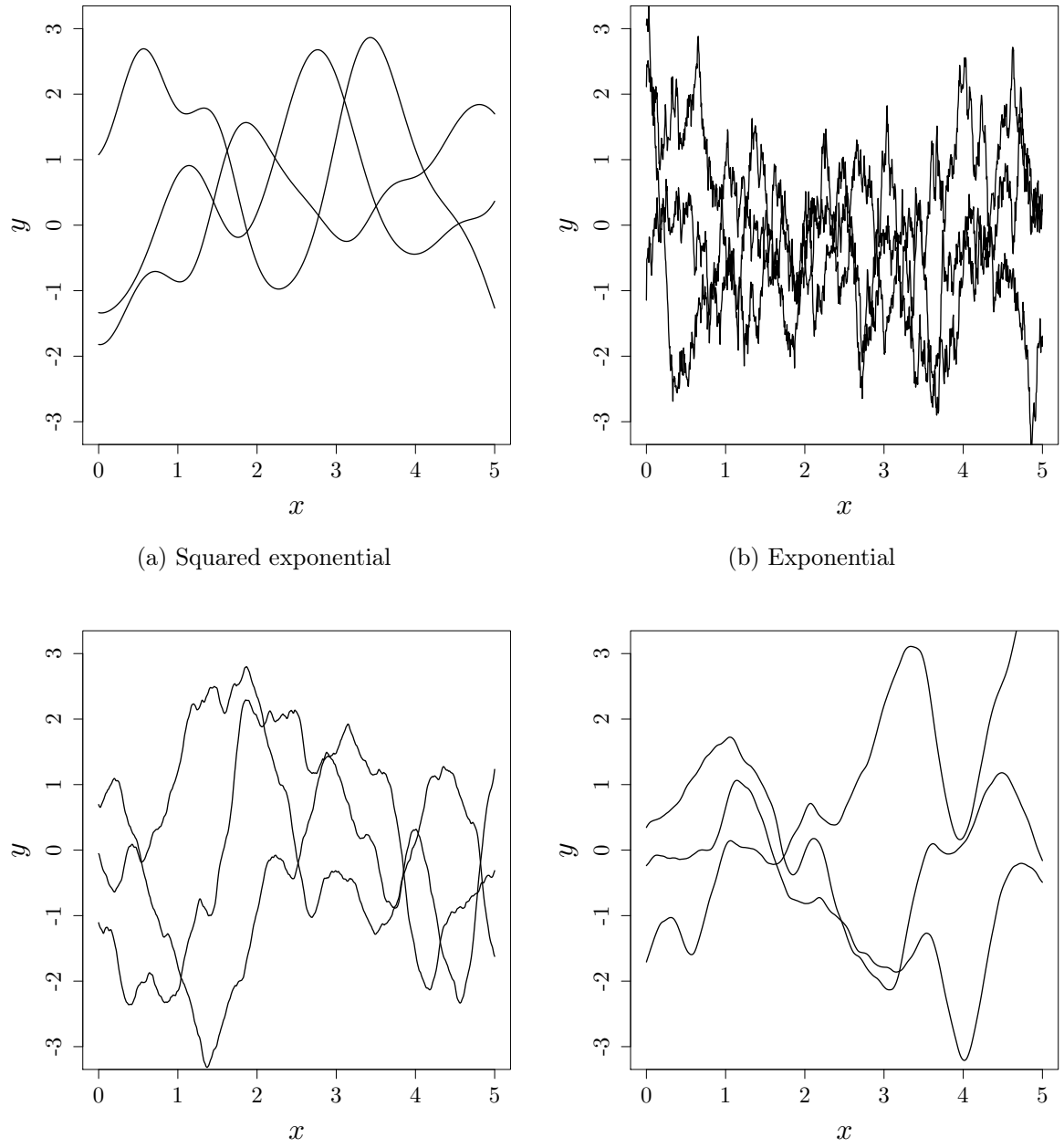


Figure 2.1: Sample paths of a GP with two different length-scales in $1D$, $\theta = 0.1$ (left) and $\theta = 1$ (right). The covariance function is squared exponential.

When the covariance function is Matérn, the GP's sample paths are $\lfloor \nu - 1/2 \rfloor$ times differentiable. Hence, the process with Matérn 5/2 is twice differentiable and with Matérn 3/2 only once [RGD12]. As $\nu \to \infty$, the Matérn kernel becomes squared exponential. Consequently, the process is infinitely differentiable and, therefore, the process is very smooth. Moreover, exponential kernel is a Matérn kernel with $\nu = 1/2$ and the GP is only continuous. Figure 2.2 illustrates the sample paths of a GP generated by different covariance functions.



(a) Squared exponential

(b) Exponential

(c) Matérn $\nu = 3/2$

(d) Matérn $\nu = 5/2$

Figure 2.2: Three sample paths of a GP when the covariance function is: (a) squared exponential, (b) exponential, (c) Matérn $\nu = 3/2$ and (d) Matérn $\nu = 5/2$. Squared exponential and exponential kernels have the most and the least smooth sample paths. The parameters θ and σ^2 are identical in the pictures.

2.2 Kriging

In this thesis, a conditional Gaussian process is referred to as kriging model. Kriging was first developed in the fields of geostatistics to predict quantities based on their spatial correlation [Cre93]. The term "kriging" comes from the name of a South African mining engineer D. Krige, who applied statistical methods to analyze mining data [Kri53]. Kriging has been successfully used in the field of computer experiments after the work of Sacks et al. [SWMW89].

Equation (6) represents a kriging model, $Y_{KG}(\mathbf{x})$. It is composed of two parts: the first one is the regression model $\sum_{i=1}^{p} \beta_i \phi_i(\mathbf{x})$, also known as kriging trend, in which β_i , $i = 1, \ldots, p$, is the coefficient of basis function $\phi_i(\mathbf{x})$. The kriging trend determines the trend in data. Note that in practice the kriging trend could be any functions. The second part is the centered Gaussian process $Y \sim \mathcal{GP}(0, K)$. The coefficients of the kriging trend are estimated from data and the centered GP is learned from the residuals.

$$Y_{KG}(\mathbf{x}) = \sum_{i=1}^{p} \beta_i \phi_i(\mathbf{x}) + Y(\mathbf{x}). \tag{6}$$

Depending on the kriging trend, three types of kriging model are specified in the literature.

- Simple kriging: the trend is known.
- Universal kriging: the trend is an unknown function.
- Ordinary kriging: the trend is constant but unknown.

Let $\mathbf{X} = \{\mathbf{x}^1, \ldots, \mathbf{x}^n\}$ be a set of n design points and $\mathbf{y} = \{f(\mathbf{x}^1), \ldots, f(\mathbf{x}^n)\}$ the associated function values at $\mathbf{X}$. Suppose that the observations are a realization of a stationary Gaussian process $Y(\mathbf{x})$. The kriging model is the Gaussian process conditional on the observations, $(Y(\mathbf{x}) \mid Y(\mathbf{X}) = \mathbf{y})$. For a simple kriging model with the constant trend μ, the prediction (kriging mean) and the prediction variance (kriging variance) at a point $\mathbf{x}$ are

$$m(\mathbf{x}) = \mu + \mathbf{c}(\mathbf{x})^\top \mathbf{C}^{-1}(\mathbf{y} - \mathbf{1}\mu) = \mu + \mathbf{r}(\mathbf{x})^\top \mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\mu), \tag{7}$$

$$s^2(\mathbf{x}) = \hat{\sigma}^2 - \mathbf{c}(\mathbf{x})^\top \mathbf{C}^{-1} \mathbf{c}(\mathbf{x}) = \hat{\sigma}^2 \left(1 - \mathbf{r}(\mathbf{x})^\top \mathbf{R}^{-1} \mathbf{r}(\mathbf{x})\right). \tag{8}$$

Here, $\mathbf{1}$ is a $n \times 1$ vector of ones, $\mathbf{r}(\mathbf{x})$ is the vector of correlations between a point $\mathbf{x}$ and the n sample points, $\mathbf{r}_i = \mathrm{Corr}(Y(\mathbf{x}), Y(\mathbf{x}^i))$, and $\mathbf{R}$ is an $n \times n$ correlation matrix between sample points, $\mathbf{R}_{ij} = \mathrm{Corr}(Y(\mathbf{x}^i), Y(\mathbf{x}^j))$. In Equation (8), the kriging variance does not depend on the observations $\mathbf{y}$. Moreover, the kriging variance is always smaller than the process variance, because of additional information provided by the observations. Figure 2.3, displays the Gaussian process learning from data points.

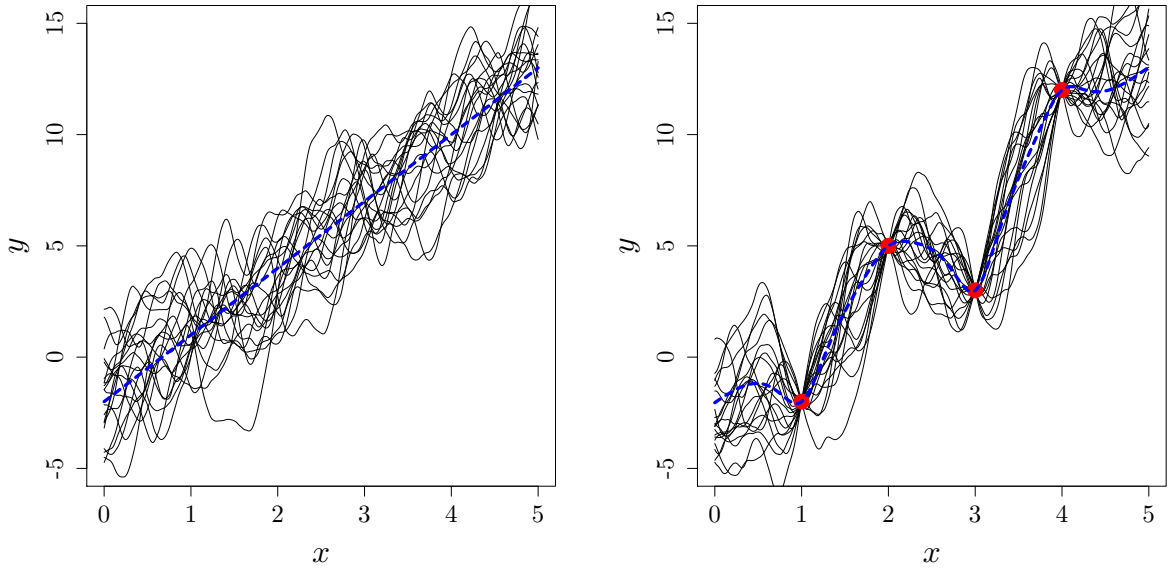


Figure 2.3: Sample paths of a GP (thin solid lines). Left: unconditional GP where the trend (dashed line) is: $3x - 2$. Right: the GP is learned from data points (bullets); the kriging prediction is the posterior mean.

The kriging mean given by Equation (7), interpolates at sample points and the kriging variance, Equation (8), is null there. Indeed at the location of every sample point, say ith sample, $\mathbf{r}(\mathbf{x}^i)$ is equal to $\mathbf{R}^i$, the ith column of the correlation matrix. In this case, the term $\mathbf{r}(\mathbf{x}^i)^\top \mathbf{R}^{-1}$ is equal to the vector $\mathbf{e}_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ in which the ith element is 1. Therefore, $m(\mathbf{x}^i) = \mu + (\mathbf{y}^i - \mu) = f(\mathbf{x}^i)$ and $s^2(\mathbf{x}^i) = \sigma^2 \left(1 - \mathbf{r}(\mathbf{x}^i)\right) = 0$.

2.2.1 Estimation of kriging parameters

In a universal kriging model, to estimate the trend coefficients, one can use the least square method. According to the Gauss-Markov theorem [Kru68], in a linear regression model, if the error terms, ϵ_i, fulfill some properties, the ordinary least squares is the Best Linear

Unbiased Estimator (BLUE). It means that the BLUE estimator has the lowest variance of the estimate among all other linear, unbiased estimators. These properties are:

1. $\mathrm{E}(\epsilon_i) = 0\ , \quad \forall i$ (error terms have zero mean),
2. $\mathrm{Var}(\epsilon_i) = \sigma^2\ , \quad \forall i$ (error terms have the same variance, "homoscedasticity"),
3. $\mathrm{Cov}(\epsilon_i, \epsilon_j) = 0\ , \quad \forall i \neq j$ (error terms are uncorrelated).

The third condition of the Gauss-Markov theorem does not hold for kriging models. However, it is possible to transform the model into the Gauss-Markov framework. After taking n sample points, the kriging model (Equation (6)) can be written in matrix form as follows:

$$\mathbf{y} = \mathbf{\Phi}\mathbf{b} + \mathbf{Y}\ , \tag{9}$$

where $\mathbf{\Phi}$ is an $n \times p$ matrix whose ijth element is: $\mathbf{\Phi}_{ij} = \phi_j(\mathbf{x}^i)$. Also, $\mathbf{b} = [\beta_1, \ldots, \beta_p]^\top$ and $\mathbf{Y} = [Y(\mathbf{x}^1), \ldots, Y(\mathbf{x}^n)]^\top \ \sim \ \mathcal{N}(\mathbf{0}, \mathbf{C})$. For the positive definite matrix $\mathbf{C}^{-1}$ there exists a matrix $\mathbf{B}$ such that $\mathbf{C}^{-1} = \mathbf{B}^\top\mathbf{B}$. If both sides of Equation (9) are multiplied by $\mathbf{B}$, it gives:

$$\underbrace{\mathbf{B}\mathbf{y}}_{\mathbf{y}_*} = \underbrace{\mathbf{B}\mathbf{\Phi}}_{\mathbf{\Phi}_*}\mathbf{b} + \underbrace{\mathbf{B}\mathbf{Y}}_{\mathbf{Y}_*}. \tag{10}$$

Now the linear regression model $\mathbf{y}_* = \mathbf{\Phi}_*\mathbf{b} + \mathbf{Y}_*$ is consistent with the Gauss-Markov theorem because the error terms $\mathbf{Y}_*$ are uncorrelated. The proof makes use of the fact that $\mathrm{Cov}(\mathbf{Y}) = \mathbf{C} = \mathbf{B}^{-1}\left(\mathbf{B}^{-1}\right)^\top$, see Equation (11).

$$\mathrm{Cov}(\mathbf{Y}_*) = \mathrm{Cov}(\mathbf{B}\mathbf{Y}) = \mathbf{B}\mathrm{Cov}(\mathbf{Y})\mathbf{B}^\top = \mathbf{B}\mathbf{B}^{-1}\left(\mathbf{B}^{-1}\right)^\top\mathbf{B}^\top = \mathbf{I}. \tag{11}$$

Finally, the coefficients are estimated using Equation (10)

$$\hat{\mathbf{b}} = \left(\mathbf{\Phi}_*^\top\mathbf{\Phi}_*\right)^{-1}\mathbf{\Phi}_*^\top\mathbf{y}_* = \left(\mathbf{\Phi}^\top\mathbf{C}^{-1}\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^\top\mathbf{C}^{-1}\mathbf{y} = \frac{\mathbf{\Phi}^\top\mathbf{R}^{-1}\mathbf{y}}{\mathbf{\Phi}^\top\mathbf{R}^{-1}\mathbf{\Phi}}. \tag{12}$$

This modified estimation method is called generalized least squares, see [Han07] for more details.

The other unknown parameters, σ^2 and the $\boldsymbol{\theta}$, are often estimated by maximum likelihood (ML). In a simple kriging model the likelihood function is the probability density of the observations

$$L(\boldsymbol{\theta}, \sigma^2|\mathbf{y}) = P(\mathbf{y}|\boldsymbol{\theta}, \sigma^2) = \frac{1}{(2\pi)^{n/2}|\mathbf{C}|^{1/2}}\exp\left(-\frac{(\mathbf{y} - \mathbf{1}\mu)^\top\mathbf{C}^{-1}(\mathbf{y} - \mathbf{1}\mu)}{2}\right), \tag{13}$$

where $|\mathbf{C}|$ is the determinant of the covariance matrix. It is more convenient to work with the natural logarithm of the likelihood function that is:

$$\ln L(\boldsymbol{\theta}, \sigma^2|\mathbf{y}) = -\frac{n}{2}\ln(2\pi) - \frac{1}{2}\ln|\mathbf{C}| - \frac{1}{2}(\mathbf{y} - \mathbf{1}\mu)^\top \mathbf{C}^{-1}(\mathbf{y} - \mathbf{1}\mu). \tag{14}$$

The ML estimator of the process variance σ^2 is

$$\hat{\sigma}^2 = \frac{1}{n}(\mathbf{y} - \mathbf{1}\mu)^\top \mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\mu). \tag{15}$$

Substituting $\hat{\sigma}^2$ on (14) we get the concentrated likelihood, also known as profile likelihood, which depends only on $\boldsymbol{\theta}$

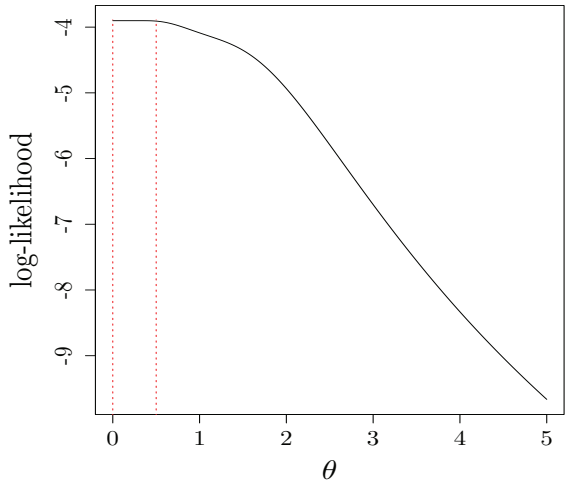
$$2\ln L(\boldsymbol{\theta}|\mathbf{y}) = -n\ln(2\pi) - n\ln\hat{\sigma}^2 - \ln|\mathbf{R}| - n. \tag{16}$$

Finally, $\boldsymbol{\theta}$, as defined in Section 2.1, is estimated by maximizing Equation (16) subject to the constraint that all elements of $\boldsymbol{\theta}$ are positive.

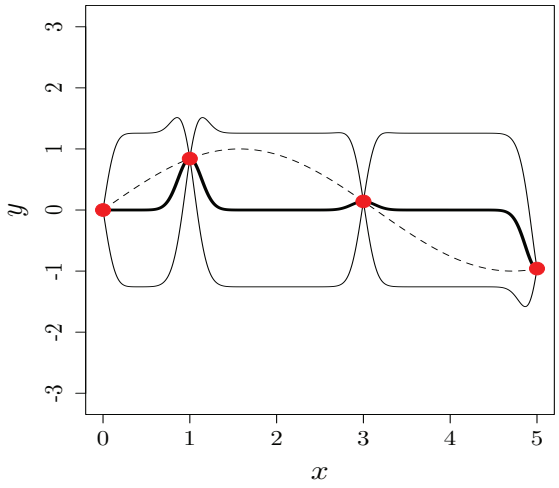
The quality of model prediction depends on the accuracy of the parameters estimated by ML. When the likelihood function is flat around the optimum, the estimated parameters may have high potential error, see Figure 2.4. Sasena [Sas02] in his thesis introduced a metric to detect the plateau in the likelihood function. He came up with the following test

$$2\ln L(\boldsymbol{\theta}^*|\mathbf{y}) \geq -n\ln(2\pi) - n\ln\mathrm{Var}(\mathbf{y}) - n. \tag{17}$$

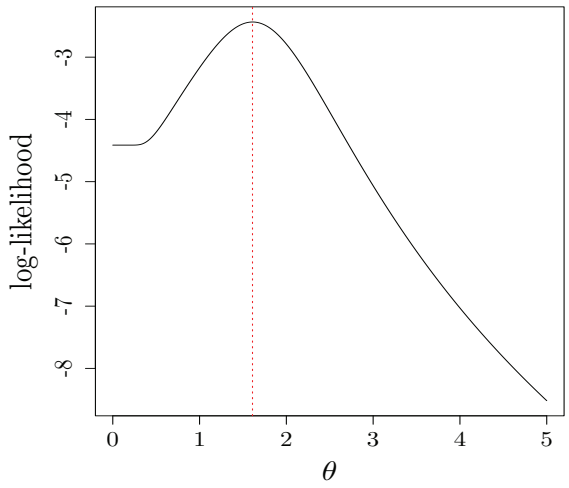
If the optimal value of the log-likelihood function failed the test, then there is a high chance that the likelihood function was flat. The idea is based on the fact that when the length-scales are small, the correlation matrix approaches the identity matrix and $\hat{\sigma}^2$, Equation (15), degenerates to $\mathrm{Var}(\mathbf{y})$. Therefore, the log-likelihood function, Equation (16), becomes $-n\ln(2\pi) - n\ln\mathrm{Var}(\mathbf{y}) - n$. Other methods to avoid this problem are restricted maximum likelihood (REML) [PT71] and penalized likelihood function [LS05].



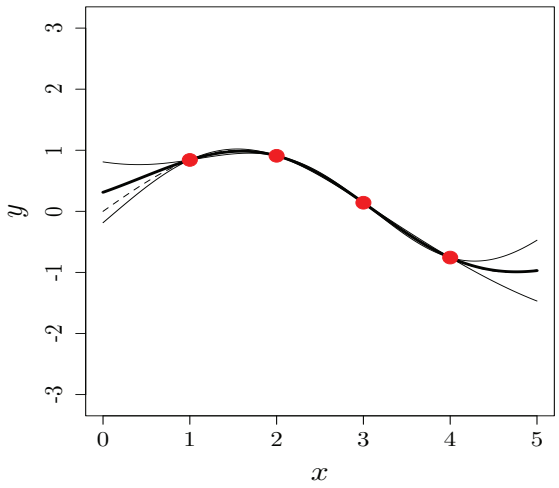
(a) Log-likelihood function, $\hat{\theta} \approx 0.2$.



(b) Kriging model with $\hat{\theta} \approx 0.2$.



(c) Log-likelihood function, $\hat{\theta} \approx 1.6$.



(d) Kriging model with $\hat{\theta} \approx 1.6$.

Figure 2.4: Approximation of the true function (dashed line) by kriging model (kriging mean: thick line, kriging variance: thin lines). (a) There is a plateau in the log-likelihood function and $\hat{\theta}$ is not confident. (c) The log-likelihood function is strongly peaked.

2.2.2 Modeling noise in Gaussian processes and nugget effect

One basic assumption of the kriging models introduced so far is that the function evaluations at design points are deterministic. However, this assumption is not always correct. For example, in stochastic computer simulations, if the simulation runs are repeated with the same input vector, we will observe different responses. In this case, the response values are considered noisy and the kriging model is expressed by

$$Y_{KG}(\mathbf{x}) = \sum_{i=1}^{p} \beta_i \phi_i(\mathbf{x}) + Y(\mathbf{x}) + \epsilon_x, \tag{18}$$

where the error term ϵ_x is an additive Gaussian white noise:

$$(\epsilon_x, \epsilon_{x'}) \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \sigma_\epsilon^2 & 0 \\ 0 & \sigma_{\epsilon'}^2 \end{bmatrix}\right).$$

In this case $Y_{KG}(\mathbf{x})$ has an extra covariance with itself only and the (simple) kriging mean and variance are modified as [RW05]

$$m(\mathbf{x}) = \mu + \mathbf{c}(\mathbf{x})^\top (\mathbf{C} + \mathbf{\Delta})^{-1} (\mathbf{y} - \mathbf{1}\mu), \tag{19}$$

$$s^2(\mathbf{x}) = \sigma^2 - \mathbf{c}(\mathbf{x})^\top (\mathbf{C} + \mathbf{\Delta})^{-1} \mathbf{c}(\mathbf{x}). \tag{20}$$

Here, $\mathbf{\Delta}$ is a diagonal matrix containing the noise variances, $\mathbf{\Delta}_{ii} = \sigma_{\epsilon_i}^2$. Accordingly, the kriging mean no longer interpolates data points and the kriging variance does not vanish there. Also, $s^2(\mathbf{x})$ is globally more inflated than in the noiseless case [RGD12]. If the probability distribution of noises is the same at every point, we say that noises are *homogeneous*, otherwise we say that they are *heterogeneous*. Figure 2.5 illustrates a kriging approximation with heterogeneously noisy observations.

With stochastic simulations, a naive way to estimate the noise variances is to repeat the simulation runs at each point and then calculate the variance of the outputs, see Equation (21). Another method is to estimate noise variances simultaneously with other parameters of kriging model in likelihood function.

$$\hat{\mathbf{\Delta}} = \begin{bmatrix} \mathrm{Var}\left(y(\mathbf{x}^1)\right) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathrm{Var}\left(y(\mathbf{x}^n)\right) \end{bmatrix}. \tag{21}$$

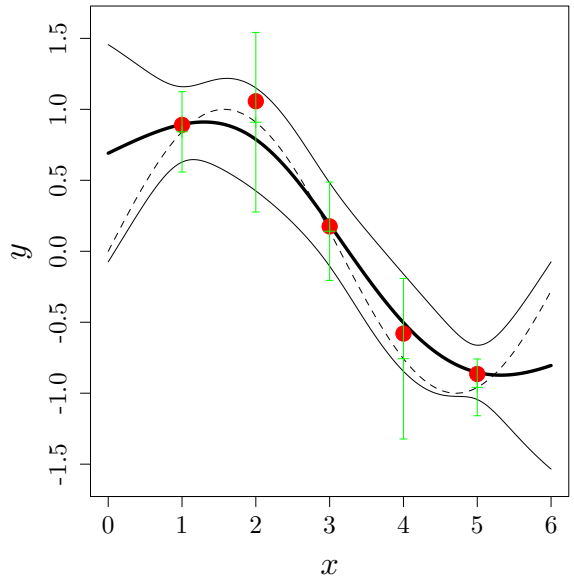


Figure 2.5: Kriging with heterogeneously noisy observations where noise variances are: $\mathbf{\Delta} = diag\left((0.02, 0.1, 0.03, 0.08, 0.01)\right)$. The bars are $\pm$ two times the standard deviation of the noise. The kriging mean does not interpolate the data points and the kriging variance is not zero there.

Sometimes, there is a jump in the simulator outputs, although deterministic, because of numerical instabilities in computations. This phenomenon can happen when a slight change in the input vector yields completely different outputs [RGD12]. To take into account such discontinuity (jump), one can use *nugget* in his model. When a nugget, τ^2, is added to the model, the covariance function is modified as follows:

$$K_\tau(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}, \mathbf{x}') + \tau^2 \delta(\mathbf{x}, \mathbf{x}'), \tag{22}$$

where $\delta(.,.)$ is the Kronecker's delta. Figure 2.6 shows how a nugget effect can handle the discontinuities in responses.

A kriging model with nugget can interpolate data points because the nugget term is added not only to the main diagonal of the covariance matrix $\mathbf{C}$ but also to the covariance vector $\mathbf{c}(\mathbf{x})$, see Equation (19). Moreover, the process variance increases to $\sigma^2 + \tau^2$. However, both kriging models (with nugget and noisy observations) have the same prediction except at the design points, see Figure 2.7.

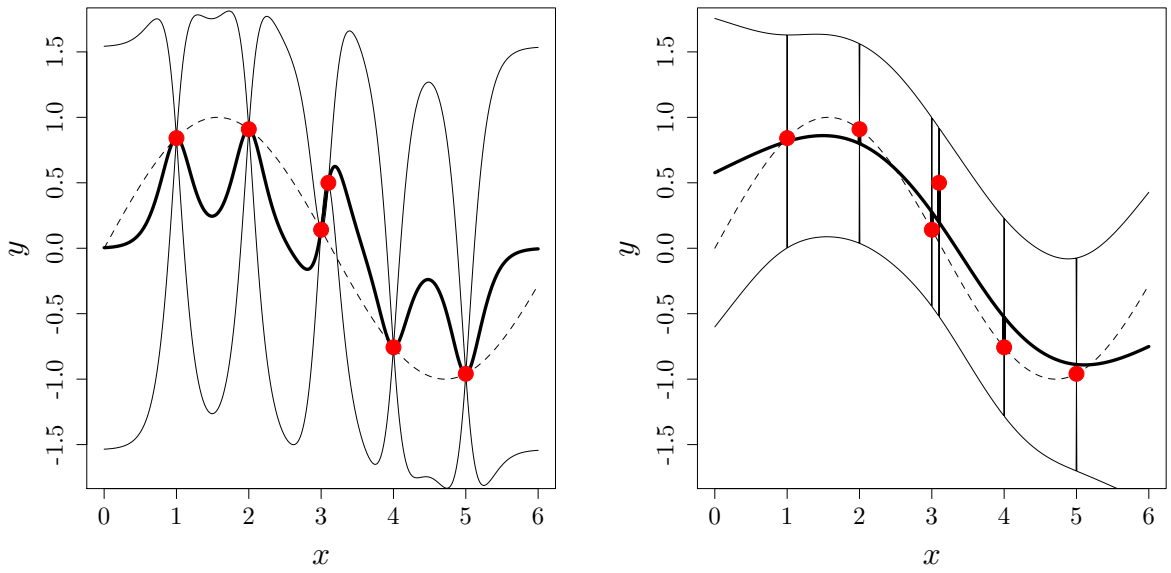


Figure 2.6: Left: kriging model without nugget. Right: kriging with nugget equal to 0.1. The true function is $sin(x)$ (dashed line). The response value at point $x = 3.1$ is 0.5 instead of $sin(3.1) = 0.04$.

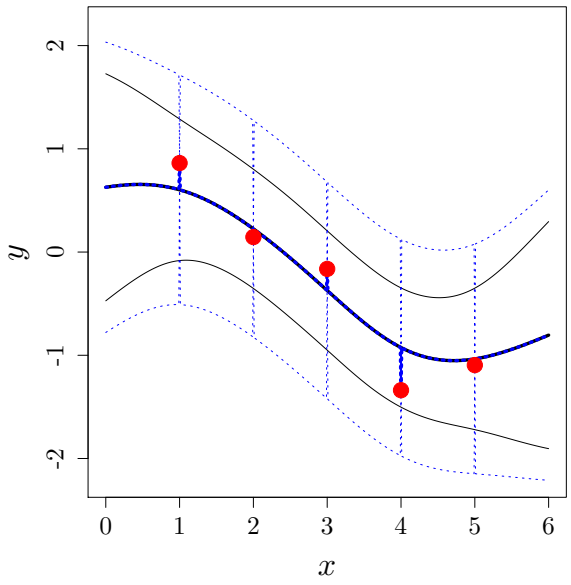


Figure 2.7: Kriging with noisy observations (solid) vs. kriging with nugget (dotted). The nugget value and the noise variance are 0.2. Predicting with nugget or noisy observations is identical everywhere but the design points. The kriging model with nugget has larger variance because the process variance is $\sigma^2 + \tau^2$.

In the presence of nugget the likelihood function is modified as follows

$$2\ln L_\tau(\boldsymbol{\theta}, \sigma^2, \tau^2|\mathbf{y}) = -n\ln(2\pi) - n\ln\hat{\sigma}_\tau^2 - \ln|\mathbf{R}_\tau| - n, \tag{23}$$

where $\hat{\sigma}_\tau^2 = \frac{1}{n}(\mathbf{y} - \mathbf{1}\mu)^\top \mathbf{R}_\tau^{-1}(\mathbf{y} - \mathbf{1}\mu)$ and $\mathbf{R}_\tau = \frac{\sigma^2}{\sigma^2+\tau^2}\mathbf{R} + \frac{\tau^2}{\sigma^2+\tau^2}\mathbf{I}$. Here, the likelihood function depends on $\boldsymbol{\theta}$, σ^2 and τ^2. The inversion of $\mathbf{R}_\tau$ can be done by means of the Woodbury formula [Woo50]. The inverse of the matrix $\mathbf{A} + a\mathbf{I}$ in which $\mathbf{A}$ is positive semidefinite and $a > 0$ using Woodbury formula reads [YNN11]

$$(\mathbf{A} + a\mathbf{I})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}(\mathbf{A}^{-1} + a^{-1}\mathbf{I})^{-1}\mathbf{A}^{-1}. \tag{24}$$

2.3 EGO algorithm

2.3.1 General principle of EGO

The general idea of the EGO algorithm is summarized below. Starting with an initial

Algorithm 2.1 Efficient Global Optimization Algorithm (EGO)

Create an initial design: $\mathbf{X} = [\mathbf{x}^1, \ldots, \mathbf{x}^n]^T$.
Evaluate function at $\mathbf{X}$ and set $\mathbf{y} = f(\mathbf{X})$.
Fit a kriging model on the data points $(\mathbf{X}, \mathbf{y})$.
while not stop **do**
  $\mathbf{x}^{n+1} \leftarrow \arg\max_{\mathbf{x}\in\mathcal{S}} EI(\mathbf{x})$ and add $\mathbf{x}^{n+1}$ to $\mathbf{X}$.
  $y^{n+1} \leftarrow f(\mathbf{x}^{n+1})$ and add y^{n+1} to $\mathbf{y}$.
  Re-estimate the parameters and update the kriging model.
end while

DoE, EGO sequentially adds one point to the existing design points based on the EI infill sampling criterion. Then, the parameters of covariance functions are re-estimated and the kriging model is updated. This process continues until a stopping criterion is met. Jones et al. [JSW98] proposed to stop EGO when the EI is less than 1% of the current best objective function value. A discussion of the stopping criteria used in surrogate-based optimization algorithms can be found in [CH13]. In this work, we use a fixed number of function evaluations for stopping criterion.

2.3.2 Infill sampling criteria

EGO iteratively creates a design of experiments aimed at finding a point with the lower function value thanks to the global optimization oriented infill sampling criterion, also known as acquisition function. There are different types of infill sampling criteria, see [BCdF09, Jon01], but the expected improvement (EI) measure is particularly popular. Some advantages of the EI criterion are:

1. It does not have any arbitrary parameters to be tuned.
2. Under certain assumptions, the EI criterion will lead to a convergence of EGO to the global optimum [Loc97].
3. EI can be used as a stopping criterion in the EGO algorithm: if the maximum value of EI is consistently low, the optimization could be terminated [FJ08, JSW98].
4. It has the capability of being implemented on parallel architectures, [SLK04, GLRC10].

The EI magnitude at a point indicates the amount of improvement one should expect if the function is evaluated there. The improvement over the best objective function value observed so far, $f_{min} = \min(\mathbf{y})$, is defined as [MTZ78]

$$I \;=\; \max\; \{0,\; f_{min} - Y_{KG}(\mathbf{x})\},$$

where $h(\mathbf{x}) = f_{min} - Y_{KG}(\mathbf{x})$ has normal distribution: $h(\mathbf{x}) \sim \mathcal{N}\left(f_{min} - m(\mathbf{x}), s^2(\mathbf{x})\right)$. The expected improvement is calculated through

$$\int_{h=0}^{h=\infty} \frac{1}{\sqrt{2\pi}s(\mathbf{x})} \exp\left(-\frac{(h - f_{min} + m(\mathbf{x}))^2}{2s^2(\mathbf{x})}\right) dh, \tag{25}$$

which yields

$$EI(\mathbf{x}) = \begin{cases} (f_{min} - m(\mathbf{x}))\Phi\left(\frac{f_{min}-m(\mathbf{x})}{s(\mathbf{x})}\right) + s(\mathbf{x})\phi\left(\frac{f_{min}-m(\mathbf{x})}{s(\mathbf{x})}\right) & \text{if } s(\mathbf{x}) > 0 \\ 0 & \text{if } s(\mathbf{x}) = 0\ . \end{cases} \tag{26}$$

Here, Φ and ϕ denote the cumulative distribution function (cdf) and probability density function (pdf) of the standard normal distribution, respectively. Sometimes a predetermined target $T \in \mathbb{R}$ is used in EI instead of f_{min}, see e.g. [QVPH09]. Finally, the next infill sample is taken where the EI is maximum: $\mathbf{x}^{n+1} \;=\; \arg\max_{x \in \mathcal{S}}\; EI(\mathbf{x})$.

EI is a non-negative function and, in the noiseless case, it vanishes at data points. It is strictly increasing with $s(\mathbf{x})$ and decreasing with $m(\mathbf{x})$ [PWG13]. The EI magnitude will augment at a location point $\mathbf{x}$ if:

1. $m(\mathbf{x})$ is small with respect to f_{min} which increases the first term of EI.

2. $s(\mathbf{x})$ is high which increases the second term of EI.

It means that the first term controls the local search and the second term contributes in global search. That is why the expected improvement is a compromise between *exploiting* the surrogate model and *exploring* the search space.

Besides the above-mentioned advantages of EI, the main disadvantage is that EI does not allow the user to have control over exploration / exploitation. To mitigate this pitfall, different methods are proposed. For instance, Schonlau [Sch97] in his PhD thesis introduces the *generalized expected improvement* criterion. Generalized expected improvement has an additional non-negative integer parameter g such that increasing this parameter shifts the emphasis from local exploitation to global exploration. In this case, the improvement is defined as

$$I^g(\mathbf{x}) \;=\; \max\;\{0,\;\;(f_{min} - Y(\mathbf{x}))^g\}. \tag{27}$$

It can be seen when $g = 1$, $I^g(\mathbf{x})$ yields EI.

In another work, a weighted expected improvement criterion is proposed by Sóbester et al. [SLK05]

$$WEI(\mathbf{x}) = \begin{cases} w(f_{min} - m(\mathbf{x}))\Phi\left(\frac{f_{min}-m(\mathbf{X})}{s(\mathbf{X})}\right) + (1-w)s(\mathbf{x})\phi\left(\frac{f_{min}-m(\mathbf{X})}{s(\mathbf{X})}\right) & \text{if } s(\mathbf{x}) > 0 \\ 0 & \text{if } s(\mathbf{x}) = 0\ , \end{cases} \tag{28}$$

where the weighting factor $w \in [0, 1]$ controls the balance between local and global search. $WEI(\mathbf{x})$ does purely local search when $w = 1$ and global search if $w = 0$. Lizotte et al. [LGS12] modifies the EI criterion by introducing an additional parameter $\xi \geq 0$

$$EI_\xi(\mathbf{x}) = \begin{cases} (f_{min} - \xi - m(\mathbf{x}))\Phi\left(\frac{f_{min}-\xi-m(\mathbf{X})}{s(\mathbf{X})}\right) + s(\mathbf{x})\phi\left(\frac{f_{min}-\xi-m(\mathbf{X})}{s(\mathbf{X})}\right) & \text{if } s(\mathbf{x}) > 0 \\ 0 & \text{if } s(\mathbf{x}) = 0\ , \end{cases} \tag{29}$$

in which higher values of the parameter ξ biases the search towards exploration and vice versa.

In practice, there might be some difficulties with the maximization of EI. EI is often highly multimodal function. Moreover, it is possible that the EI and its gradient become numerically zero since Φ and ϕ in Equation (7) diminish exponentially when the term $\mathcal{Z} = \frac{f_{min} - m(\mathbf{x})}{s(\mathbf{x})}$ is small [LGS12]. For example, in R (version i386 3.2.1) the functions Φ and ϕ, with the corresponding commands `pnorm` and `dnorm`, are numerically zero for $\mathcal{Z} \leq -39$. However, EI can be considered as non expensive function since it does not require calculating the function f. This maximization is usually performed by stochastic optimization algorithms. As an example in [JR13] and in the Scilab KRISP toolbox [Jan13] the optimization of EI is done by CMA-ES [Han09b]. Note that using stochastic methods in EGO, makes it a stochastic algorithm.

Chapter 3

An analytic comparison of regularization methods for Gaussian processes

Gaussian Processes (GPs) are often used to predict the output of a parameterized deterministic experiment. They have many applications in the field of Computer Experiments, in particular to perform sensitivity analysis, adaptive design of experiments and global optimization. Nearly all of the applications of GPs to Computer Experiments require the inversion of a covariance matrix. Because this matrix is often ill-conditioned, regularization techniques are required. Today, there is still a need to better regularize GPs.

The two most classical regularization methods to avoid degeneracy of the covariance matrix are *i)* pseudoinverse (PI) and *ii)* adding a small positive constant to the main diagonal, i.e., the case of noisy observations. In this chapter, we will refer to the second regularization technique with a slight abuse of language as nugget. This chapter provides algebraic calculations which allow comparing PI and nugget regularizations. It is proven that pseudoinverse regularization averages the output values and makes the variance null at redundant points. On the opposite, nugget regularization lacks interpolation properties but preserves a non-zero variance at every point. However, these two regularization techniques become similar as the nugget value decreases. A distribution-wise GP is introduced which interpolates Gaussian distributions instead of data points and mitigates the drawbacks of pseudoinverse and nugget regularized GPs. Finally, data-model discrepancy is discussed and serves as a guide for choosing a regularization technique.

3.1 Introduction

Although GPs can model stochastic or deterministic spatial phenomena, the focus of this work is on experiments with deterministic outputs. Computer simulations provide examples of such noiseless experiments. Furthermore, we assume that the location of the observed points and the covariance function are given a priori. This occurs frequently within algorithms performing adaptive design of experiments [BGL+12], global sensitivity analysis [OO02] and global optimization [JSW98].

Kriging models require the inversion of a covariance matrix which is made of the covariance function evaluated at every pair of observed locations. In practice, anyone who has used a kriging model has experienced one of the circumstances where the covariance matrix is not numerically invertible. This happens when observed points are repeated, or even are close to each other, or when the covariance function makes the information provided by observations redundant.

In the literature, various strategies have been employed to avoid degeneracy of the covariance matrix. A first set of approaches proceed by controlling the locations of design points (the Design of Experiments or DoE). The influence of the DoE on the condition number of the covariance matrix has been investigated in [SB97]. [Ren09] proposes to build kriging models from a uniform subset of design points to improve the condition number. In [OGR09], new points are taken suitably far from all existing data points to guarantee a good conditioning.

Other strategies select the covariance function so that the covariance matrix remains well-conditioned. In [DM97] for example, the influence of all kriging parameters on the condition number, including the covariance function, is discussed. Ill-conditioning also happens in the related field of linear regression with the Gauss-Markov matrix $\mathbf{\Phi}^\top\mathbf{\Phi}$ that needs to be inverted, where $\mathbf{\Phi}$ is the matrix of basis functions evaluated at the DoE. In regression, work has been done on diagnosing ill-conditioning and the solution typically involves working on the definition of the basis functions to recover invertibility [Bel91]. The link between the choice of the basis functions and the choice of the covariance functions is given by Mercer's theorem, [RW05].

Instead of directly inverting the covariance matrix, an iterative method has been pro-

posed in [Gib97] to solve the kriging equations and avoid numerical instabilities.

Two generic solutions to overcome the degeneracy of covariance matrix are the pseudoinverse (PI) and the “nugget" regularizations. They have a wide range of applications because, contrarily to the methods mentioned above, they can be used a posteriori in computer experiments algorithms without major redesign of the methods. This is the reason why most kriging implementations contain PI or nugget regularization.

The singular value decomposition and the idea of pseudoinverse have already been suggested in [JSW98] to overcome degeneracy. The Model-Assisted Pattern Search (MAPS) software [STT00] relies on an implementation of the pseudoinverse to invert the (covariance) matrices.

The most often used approach to deal with ill-conditioning in the covariance is to introduce a “nugget” [BDJ+98, SWN03, Nea97, AC12], that is to say add a small positive scalar on the covariance diagonal. The popularity of the nugget regularization may be either due to its simplicity or to its interpretation as the variance of a noise on the observations. The value of the nugget term can be estimated by maximum likelihood (ML). It is reported in [Pep10] that the presence of a nugget term significantly changes the modes of the likelihood function of a GP. Similarly in [GL09], the authors have advocated a nonzero nugget term in the design and analysis of their computer experiments. They have also stated that estimating a nonzero nugget value may improve some statistical properties of the kriging models such as their predictive accuracy [GL12]. However, some references like [RHK11] recommend that the magnitude of nugget remains as small as possible to preserve the interpolation property.

Because of the diversity of arguments regarding GP regularization, we feel that there is a need to provide analytical explanations on the effects of the main approaches. This chapter provides new results regarding the analysis and comparison of pseudoinverse and nugget kriging regularizations in the context of deterministic outputs. Our analysis is made possible by approximating ill-conditioned covariance matrices with the neighboring truly non-invertible covariance matrices that stem from redundant points. Some properties of kriging regularized by PI and nugget are stated and proved. The chapter finishes with the description of a new type of regularization associated to a distribution-wise GP.

3.2 Kriging models and degeneracy of the covariance matrix

3.2.1 Context: conditional Gaussian processes

Let $Y(\mathbf{x})_{\mathbf{x}\in D}$ be a GP with kernel $K(.,.)$ and zero mean ($\mu(.) = 0$). $\mathbf{X} = (\mathbf{x}^1, ..., \mathbf{x}^n)$ denotes the n data points where the samples are taken and the corresponding response values are $\mathbf{y} = (y_1, ..., y_n)^\top = (f(\mathbf{x}^1), ..., f(\mathbf{x}^n))^\top$. The posterior distribution of the GP ($Y(\mathbf{x})$) knowing it interpolates the data points is still Gaussian with mean and covariance [RW05]

$$
\begin{aligned}
m_K(\mathbf{x}) &= \mathbb{E}(Y(\mathbf{x})|Y(\mathbf{X}) = \mathbf{y}) \;=\; \mathbf{c}(\mathbf{x})^\top \mathbf{C}^{-1}\mathbf{y} \;, && (1)\\
c_K(\mathbf{x}, \mathbf{x}') &= \mathrm{Cov}(Y(\mathbf{x}), Y(\mathbf{x}')|Y(\mathbf{X}) = \mathbf{y}) \\
&= K(\mathbf{x}, \mathbf{x}') - \mathbf{c}(\mathbf{x})^\top \mathbf{C}^{-1}\mathbf{c}(\mathbf{x}') \;, && (2)
\end{aligned}
$$

where $\mathbf{c}(\mathbf{x}) = (K(\mathbf{x}, \mathbf{x}^1), ..., K(\mathbf{x}, \mathbf{x}^n))^\top$ is the vector of covariances between a new point $\mathbf{x}$ and the n already observed sample points. The $n \times n$ matrix $\mathbf{C}$ is a covariance matrix between the data points and its elements are defined as $\mathbf{C}_{i,j} = K(\mathbf{x}^i, \mathbf{x}^j) = \sigma^2 \mathbf{R}_{i,j}$, where $\mathbf{R}$ is the correlation matrix. Hereinafter, we call $m_K(\mathbf{x})$ and $v_K(\mathbf{x}) = c_K(\mathbf{x}, \mathbf{x})$ the kriging mean and variance, respectively.

3.2.2 Degeneracy of the covariance matrix

Computing the kriging mean (Equation (1)) or (co)variance (Equation (2)) or even samples of GP trajectories, requires inverting the covariance matrix $\mathbf{C}$. In practice, the covariance matrix should not only be invertible, but also well-conditioned. A matrix is said to be near singular or ill-conditioned or degenerated if its condition number is too large. For covariance matrices, which are symmetric and positive semidefinite, the condition number $\kappa(\mathbf{C})$ is the ratio of the largest to the smallest eigenvalue. In this chapter we assume that $\kappa(\mathbf{C}) \to \infty$ is possible.

There are many situations where the covariance matrix is near singular. The most frequent and easy to understand case is when some data points are too close to each other,

where closeness is measured with respect to the metric induced by the covariance function. This is a recurring issue in sequential DoEs like the EGO algorithm [JSW98] where the search points tend to pile up around the points of interest such as the global optimum [RHK11]. When this happens, the resulting covariance matrix is no longer numerically invertible because some columns are almost identical.

Here, to analyze PI and nugget regularizations, we are going to consider matrix degeneracy pushed to its limit, that is true non-invertibility (or rank deficiency) of $\mathbf{C}$. Non invertibility happens if a linear dependency exists between $\mathbf{C}$'s columns (or rows). Section 3.5.3 provides a collection of examples where the covariance matrix is not invertible with calculation details that will become clear later. Again, the easiest to understand and the most frequent occurrence of $\mathbf{C}$'s rank deficiency is when some of the data points tend towards each other until they are at the same $\mathbf{x}^i$ position. They form *repeated* points, the simplest example of what we more generally call redundant points which will be formally defined shortly. Figure 3.6 in Section 3.5.3 is an example of repeated points. Repeated points lead to strict non-invertibility of $\mathbf{C}$ since the corresponding columns are identical. The special case of repeated points will be instrumental in understanding some aspects of kriging regularization in Sections 3.3.2 and 3.4.2 because the eigenvectors of the covariance matrix associated to null eigenvalues are known.

The covariance matrix of GPs may loose invertibility even though the data points are not close to each other in Euclidean distance. This occurs for example with additive GPs for which the kernel is the sum of kernels defined in each dimension, $K(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{d} K_i(x_i, x_i')$. The additivity of a kernel may lead to linear dependency in some columns of the covariance matrix. For example, in the DoE shown in Figure 3.5, only three of the first four points which form a rectangle provide independent information in the sense that the GP response at any of the four points in fully defined by the response at the three other points. This is explained by a linear dependency between the first four columns, $\mathbf{C}^4 = \mathbf{C}^3 + \mathbf{C}^2 - \mathbf{C}^1$, which comes from the additivity of the kernel and the rectangular design [DGR12]:

$$\mathbf{C}_i^4 \;=\; \mathrm{Cov}(x_1^i, x_1^4) + \mathrm{Cov}(x_2^i, x_2^4) \;=\; \mathrm{Cov}(x_1^i, x_1^2) + \mathrm{Cov}(x_2^i, x_2^3) \;,$$

and completing the covariances while accounting for $x_2^2 = x_2^1$, $x_1^3 = x_1^1$, yields

$$\mathbf{C}_i^4 \;=\; \mathrm{Cov}(\mathbf{x}^i, \mathbf{x}^3) + \mathrm{Cov}(\mathbf{x}^i, \mathbf{x}^2) - \mathrm{Cov}(\mathbf{x}^i, \mathbf{x}^1) \;=\; \mathbf{C}_i^3 + \mathbf{C}_i^2 - \mathbf{C}_i^1 \;.$$

Note that if the measured outputs $y^1, \ldots, y^4$ are not additive ($y^4 \neq y^2 + y^3 - y^1$), none of the four measurements can be easily deleted without loss of information, hence the need for the general regularization methods that will be discussed later.

Periodic kernels may also yield non-invertible covariance matrices although data points are far from each other. This is illustrated in Figure 3.9 where points 1 and 2, and points 3 and 4, provide the same information as they are one period away from each other. Thus, $\mathbf{C}^1 = \mathbf{C}^2$ and $\mathbf{C}^3 = \mathbf{C}^4$.

Our last example comes from the dot product (or linear) kernel (cf. Section 3.5.3). Because the GP trajectories and mean are linear, no uncertainty is left in the model when the number of data points n reaches $d + 1$ and when $n > d + 1$ the covariance matrix is no longer invertible.

3.2.3 Eigen analysis and definition of redundant points

We start by introducing our notations for the eigendecomposition of the covariance matrix. Let the $n \times n$ covariance matrix $\mathbf{C}$ have rank r, $r \leq n$. A covariance matrix is positive semidefinite, thus its eigenvalues are greater than or equal to zero. The eigenvectors associated to strictly positive eigenvalues are denoted $\mathbf{V}^i$, $i = 1, \ldots, r$, and those associated to null eigenvalues are $\mathbf{W}^i$, $i = 1, \ldots, (n - r)$, that is $\mathbf{C}\mathbf{V}^i = \lambda_i \mathbf{V}^i$ where $\lambda_i > 0$ and $\mathbf{C}\mathbf{W}^i = \mathbf{0}$. The eigenvectors are grouped columnwise into the matrices $\mathbf{V} = [\mathbf{V}^1, \ldots, \mathbf{V}^r]$ and $\mathbf{W} = [\mathbf{W}^1, \ldots, \mathbf{W}^{n-r}]$. In short, the eigenvalue decomposition of the covariance matrix $\mathbf{C}$ obeys

$$\mathbf{C} \;=\; [\mathbf{V}\ \mathbf{W}]\, \mathbf{\Sigma}\, [\mathbf{V}\ \mathbf{W}]^\top, \tag{3}$$

where $\mathbf{\Sigma}$ is a diagonal matrix containing the eigenvalues of $\mathbf{C}$, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_r > 0$ and $\lambda_{r+1} = \ldots = \lambda_n = 0$. $\mathbf{V}$ spans the image space and $\mathbf{W}$ spans the null space of $\mathbf{C}$, $Im(\mathbf{C})$ and $Null(\mathbf{C})$, respectively. $[\mathbf{V}\ \mathbf{W}]$ is an orthogonal matrix,

$$[\mathbf{V}\ \mathbf{W}]^\top [\mathbf{V}\ \mathbf{W}] = [\mathbf{V}\ \mathbf{W}][\mathbf{V}\ \mathbf{W}]^\top = \mathbf{V}\mathbf{V}^\top + \mathbf{W}\mathbf{W}^\top = \mathbf{I}\;. \tag{4}$$

$\mathbf{V}\mathbf{V}^\top$ is the orthogonal projection matrix onto $Im(\mathbf{C})$. Similarly, $\mathbf{W}\mathbf{W}^\top$ is the orthogonal projection matrix onto $Null(\mathbf{C})$. For a given matrix $\mathbf{C}$, the eigenvectors $\mathbf{W}^i$ are not

uniquely defined because any linear combination of them is also an eigenvector associated to a null eigenvalue. However, the orthogonal projection matrices onto the image and null spaces of $\mathbf{C}$ are unique and will be cornerstones in the definition of redundant points.

Before formally defining redundant points, we present the examples of singular covariance matrices of Section 3.5.3. These examples are two dimensional to allow for a graphical representation. The kernels, designs of points, eigenvalues and eigenvectors and the $\mathbf{V}\mathbf{V}^\top$ projection matrix are given.
The first example detailed in Section 3.5.3 has two groups of repeated data points (points 1, 2 and 6, on the one hand, points 3 and 4, on the other hand), in which there are 3 redundant, points. The covariance matrix has 3 null eigenvalues. It should be noted that the off-diagonal coefficients of the $\mathbf{V}\mathbf{V}^\top$ projection matrix associated to the indices of repeated points are not 0.
Figure 3.7 shows how additive kernels may generate singular covariance matrices: points 1, 2, 3 and 4 are arranged in a rectangular pattern which makes columns 1 to 4 linearly dependent (as already explained in Section 3.2.2). The additive property makes any one of the 4 points of a rectangular pattern redundant in that the value of the GP there is uniquely set by the knowledge of the GP at the 3 other points. The same stands for points 5 to 8. Two points are redundant (1 in each rectangle) and there are two null eigenvalues. Again, remark how the off-diagonal coefficients of $\mathbf{V}\mathbf{V}^\top$ associated to the points of the rectangles are not zero. Another example of additivity and singularity is depicted in Figure 3.8: although the design points are not set in a rectangular pattern, there is a shared missing vertex between two orthogonal triangles so that, because of additivity, the value at this missing vertex is defined twice. In this case, there is one redundant point, one null eigenvalue, and all the points of the design are coupled: all off-diagonal terms in $\mathbf{V}\mathbf{V}^\top$ are not zero.
Finally, Figure 3.9 is a case with a periodic kernel and a periodic pattern of points so that points 1 and 2 provide the same information, and similarly with points 3 and 4. There are 2 null eigenvalues, and the (1,2) and (3,4) off-diagonal terms in $\mathbf{V}\mathbf{V}^\top$ are not zero.

In general, we call *redundant* the set of data points that make the covariance matrix non-invertible by providing linearly dependent information.

Definition 1 (Redundant points set)**.**
Let $\boldsymbol{C}$ *be the* $n \times n$ *non-invertible positive semidefinite covariance matrix of a Gaussian process. It has rank* r*,* $r < n$*.* $\boldsymbol{V}$ *is the* $n \times r$ *matrix of the eigenvectors associated to strictly positive eigenvalues.* R *is a set of at least two redundant points indices if for any* i *and* j *in* R*,* $(\boldsymbol{V}\boldsymbol{V}^\top)_{ij} \neq 0$*.*

Redundant points could be equivalently defined with the $\mathbf{W}$ matrix since, from Equation (4), $\mathbf{V}\mathbf{V}^\top$ and $\mathbf{W}\mathbf{W}^\top$ have the same non-zero off-diagonal terms with opposite signs. Subsets of redundant points are also redundant. The *degree of redundancy* of a set of points R is the number of zero eigenvalues of the covariance matrix restricted to the points in R, i.e., $[\mathbf{C}_{ij}]$ for all $(i, j) \in R^2$. The degree of redundancy is the number of points that should be removed from R to recover invertibility of the covariance restricted to the points in R. When $r = n$, $\mathbf{C}$ is invertible and there is no redundant point. An interpretation of redundant points will be made in the next Section on pseudoinverse regularization.

In the repeated points example of Section 3.5.3, the two largest redundant points sets are $\{1, 2, 6\}$ and $\{3, 4\}$ with degrees of redundancy 2 and 1, respectively. The first additive example has two sets of redundant points, $\{1, 2, 3, 4\}$ and $\{5, 6, 7, 8\}$ each with a degree of redundancy equal to 1. In the second additive example, all the points are redundant with a degree equal to 1. In the same section, the periodic case has two sets of redundant points of degree 1, $\{1, 2\}$ and $\{3, 4\}$. With the linear kernel all data points are redundant and in the given example where $n = d + 2$ the degree of redundancy is 1.

3.3 Pseudoinverse regularization

3.3.1 Definition

In this Section, we state well-known properties of pseudoinverse matrices without proofs (which can be found, e.g., in [BIC66]) and apply them to the kriging equations (1) and (2). Pseudoinverse matrices are generalizations of the inverse matrix. The most popular pseudoinverse is the *Moore–Penrose pseudoinverse* which is hereinafter referred to as pseudoinverse.

When $\mathbf{C}^{-1}$ exists (i.e., $\mathbf{C}$ has full rank, $r = n$), we denote as $\boldsymbol{\beta}$ the term $\mathbf{C}^{-1}\mathbf{y}$ of the kriging mean formula, Equation (1). More generally, when $\mathbf{C}$ is not a full rank matrix, we

are interested in the vector $\boldsymbol{\beta}$ that simultaneously minimizes[1] $\|\mathbf{C}\boldsymbol{\beta} - \mathbf{y}\|_2$ and $\|\boldsymbol{\beta}\|_2$. This solution is unique and obtained by $\boldsymbol{\beta}^{PI} = \mathbf{C}^{\dagger}\mathbf{y}$ where $\mathbf{C}^{\dagger}$ is the pseudoinverse of $\mathbf{C}$. Each vector $\boldsymbol{\beta}$ can be uniquely decomposed into

$$\boldsymbol{\beta} = \boldsymbol{\beta}^{PI} + \boldsymbol{\beta}_{Null(\mathbf{C})}, \tag{5}$$

where $\boldsymbol{\beta}^{PI}$ and $\boldsymbol{\beta}_{Null(\mathbf{C})}$ belong to the image space and the null space of the covariance matrix, respectively. The decomposition is unique since, $\mathbf{C}$ being symmetric, $Im(\mathbf{C})$ and $Null(\mathbf{C})$ have no intersection.

The pseudoinverse of $\mathbf{C}$ is expressed as

$$\mathbf{C}^{\dagger} = [\mathbf{V}\ \mathbf{W}] \begin{bmatrix} diag\left(\frac{1}{\boldsymbol{\lambda}}\right)_{r\times r} & \mathbf{0}_{r\times(n-r)} \\ \mathbf{0}_{(n-r)\times r} & \mathbf{0}_{(n-r)\times(n-r)} \end{bmatrix} [\mathbf{V}\ \mathbf{W}]^{\top}, \tag{6}$$

where $diag(\frac{1}{\boldsymbol{\lambda}})$ is a diagonal matrix with $\frac{1}{\lambda_i}$, $i = 1, \ldots, r$, as diagonal elements. So $\boldsymbol{\beta}^{PI}$ reads

$$\boldsymbol{\beta}^{PI} = \sum_{i=1}^{r} \frac{\left(\mathbf{V}^i\right)^{\top}\mathbf{y}}{\lambda_i}\mathbf{V}^i. \tag{7}$$

Equation (7) indicates that $\boldsymbol{\beta}^{PI}$ is in the image space of $\mathbf{C}$, because it is a linear combination of eigenvectors associated to positive eigenvalues. A geometrical interpretation of $\boldsymbol{\beta}^{PI}$ and pseudo-inverse is given in Figure 3.1. The kriging mean (Equation (1)) with PI regularization can be written as

$$m^{PI}(\mathbf{x}) = \mathbf{c}(\mathbf{x})^{\top} \sum_{i=1}^{r} \frac{\left(\mathbf{V}^i\right)^{\top}\mathbf{y}}{\lambda_i}\mathbf{V}^i. \tag{8}$$

Similarly, the kriging covariance (2) regularized by PI is,

$$\begin{aligned} c^{PI}(\mathbf{x}, \mathbf{x}') &= K(\mathbf{x}, \mathbf{x}') - \mathbf{c}(\mathbf{x})^{\top} \sum_{i=1}^{r} \left(\frac{\left(\mathbf{V}^i\right)^{\top}\mathbf{c}(\mathbf{x}')}{\lambda_i} \right) \mathbf{V}^i \\ &= K(\mathbf{x}, \mathbf{x}') - \sum_{i=1}^{r} \frac{\left(\left(\mathbf{V}^i\right)^{\top}\mathbf{c}(\mathbf{x})\right)\left(\left(\mathbf{V}^i\right)^{\top}\mathbf{c}(\mathbf{x}')\right)}{\lambda_i}. \end{aligned} \tag{9}$$

[1]Indeed, in this case the minimizer of $\|\mathbf{C}\boldsymbol{\beta} - \mathbf{y}\|_2$ is not unique but defined up to any sum with a vector in the $Null(\mathbf{C})$

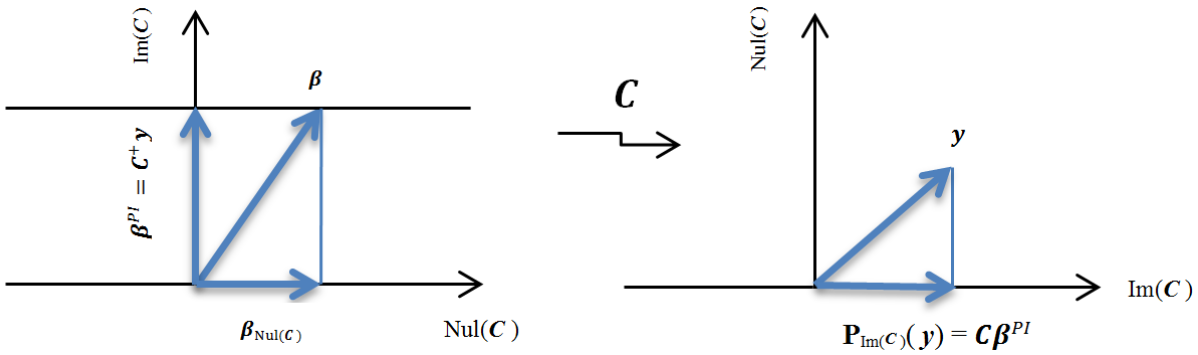


Figure 3.1: Geometrical interpretation of the Moore-Penrose pseudoinverse. In the left picture, infinitely many vectors $\boldsymbol{\beta}$ are solutions to the system $\mathbf{C}\boldsymbol{\beta}= \mathbf{y}$. But the minimum norm solution is $\mathbf{C}^\dagger\mathbf{y}$. The right picture shows the orthogonal projection of $\mathbf{y}$ onto the image space of $\mathbf{C}$, $\mathbf{P}_{Im(\mathbf{C})}(\mathbf{y})$, which is equal to $\mathbf{C}\mathbf{C}^\dagger\mathbf{y}$ (Property 1).

3.3.2 Properties of PI kriging

The PI kriging mean averages the outputs. Before proving this property, let us illustrate it with the simple example of Figure 3.2: there are redundant points at $\mathbf{x} = 1.5$, $\mathbf{x} = 2$ and $\mathbf{x} = 2.5$. We observe that the kriging mean with PI regularization is equal to the mean of the outputs, $m^{PI}(1.5) = -0.5 = (-1 + 0)/2$, $m^{PI}(2) = 5 = (1.5 + 4 + 7 + 7.5)/4$ and $m^{PI}(2.5) = 5.5 = (5 + 6)/2$. The PI averaging property is due to the more abstract fact that PI projects the observed $\mathbf{y}$ onto the image space of $\mathbf{C}$.

Property 1 (PI as projection of outputs onto $Im(\mathbf{C})$)**.**
The PI kriging prediction at $\boldsymbol{X}$ is the projection of the observed outputs onto the image space of the covariance matrix, $Im(\boldsymbol{C})$.

Proof: The PI kriging means at all design points is given by

$$m^{PI}(\mathbf{X}) = \mathbf{C}\mathbf{C}^\dagger\mathbf{y} \ . \tag{10}$$

Performing the eigendecompositions of the matrices, one gets,

$$\begin{aligned} m^{PI}(\mathbf{X}) &= [\mathbf{V}\ \mathbf{W}] \begin{bmatrix} diag(\boldsymbol{\lambda}) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}^\top \\ \mathbf{W}^\top \end{bmatrix} [\mathbf{V}\ \mathbf{W}] \begin{bmatrix} diag(\frac{1}{\boldsymbol{\lambda}}) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}^\top \\ \mathbf{W}^\top \end{bmatrix} \mathbf{y} \\ &= \mathbf{V}\mathbf{V}^\top\mathbf{y} \end{aligned} \tag{11}$$

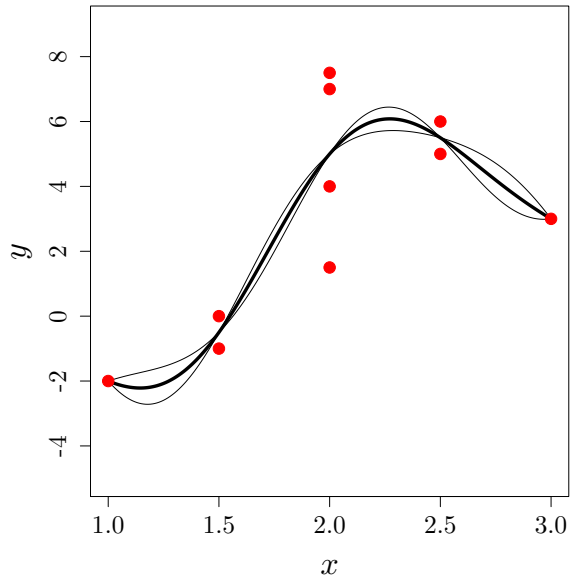


Figure 3.2: Kriging mean $m^{PI}(x)$ (thick line) and prediction intervals $m^{PI}(x) \pm 2\sqrt{v^{PI}(x)}$ (thin lines). Kriging mean using pseudoinverse goes exactly through the average of the outputs. The observed values are $\mathbf{y} = (-2, -1, 0, 1.5, 4, 7, 7.5, 6, 5, 3)^\top$. $m^{PI}(1.5) = -0.5$, $m^{PI}(2) = 5$, and $m^{PI}(2.5) = 5.5$. Note that v^{PI} is zero at redundant points.

The matrix

$$\mathbf{P}_{Im(\mathbf{C})} \;=\; \mathbf{V}\mathbf{V}^\top \;=\; (\mathbf{I} - \mathbf{W}\mathbf{W}^\top) \tag{12}$$

is the orthogonal projection onto the image space of $\mathbf{C}$ because it holds that

$$\begin{aligned}
&\mathbf{P}_{Im(\mathbf{C})} = \mathbf{P}_{Im(\mathbf{C})}^\top; \\
&\mathbf{P}_{Im(\mathbf{C})}^2 = \mathbf{P}_{Im(\mathbf{C})}; \\
&\forall \mathbf{v} \in Im(\mathbf{C}) \;,\; \mathbf{P}_{Im(\mathbf{C})}\mathbf{v} = \mathbf{v}; \\
&\text{and } \forall \mathbf{u} \in Null(\mathbf{C}) \;,\; \mathbf{P}_{Im(\mathbf{C})}\mathbf{u} = \mathbf{0} \quad \square
\end{aligned}$$

Redundant points can be further understood thanks to Property 1 and Equation (11): points redundant with $\mathbf{x}^i$ are points $\mathbf{x}^j$ where the observations influences $m^{PI}(\mathbf{x}^i)$. The kriging predictions at the redundant data points $m^{PI}(\mathbf{x}^i)$ and $m^{PI}(\mathbf{x}^j)$ are not $\mathbf{y}_i$ and $\mathbf{y}_j$, as it happens at non-redundant points where the model is interpolating, but a linear combination of them. The averaging performed by PI becomes more clearly visible in the important case of repeated points.

Property 2 (PI Averaging Property for Repeated Points)**.**

The PI kriging prediction at repeated points is the average of the outputs at those points.

Proof: Suppose that there are N repeated points at k different locations with N_i points at each repeated location, $\sum_{i=1}^{k} N_i = N$, see Figure 3.3. The corresponding columns in the covariance matrix are identical,

$$\mathbf{C} = \left(\underbrace{\mathbf{C}^1, ..., \mathbf{C}^1}_{N_1 \text{ times}}, \ldots, \underbrace{\mathbf{C}^k, ..., \mathbf{C}^k}_{N_k \text{ times}}, \mathbf{C}^{N+1}, ..., \mathbf{C}^n \right).$$

In this case, the dimension of the image space, or rank of the covariance matrix, is $n-N+k$ and the dimension of the null space is equal to $\sum_{i=1}^{k}(N_i - 1) = N - k$.

To prove this property we need to show that the matrix $\mathbf{P}$ defined as

$$\mathbf{P} = \begin{pmatrix} \frac{\mathbf{J}_{N_1}}{N_1} & & & & \\ & \ddots & & 0 & \\ & & \frac{\mathbf{J}_{N_k}}{N_k} & & \\ & 0 & & & \\ & & & & \mathbf{I}_{n-N} \end{pmatrix}, \tag{13}$$

is the projection matrix onto the image space of $\mathbf{C}$, or $\mathbf{P} = \mathbf{P}_{Im(\mathbf{C})}$. In matrix $\mathbf{P}$, $\mathbf{J}_{N_i}$ is the $N_i \times N_i$ matrix of ones and $\mathbf{I}_{n-N}$ is the identity matrix of size $n - N$. If $\mathbf{P} = \mathbf{P}_{Im(\mathbf{C})}$, based on Property 1, $m^{PI}(\mathbf{X})$ is expressed as

$$m^{PI}(\mathbf{X}) = \mathbf{P}_{Im(\mathbf{C})}\mathbf{y} = \begin{bmatrix} \overline{y_1} \\ \vdots \\ \overline{y_1} \\ \vdots \\ \overline{y_k} \\ \vdots \\ \overline{y_k} \\ y_{N+1} \\ \vdots \\ y_n \end{bmatrix}, \tag{14}$$

in which $\overline{y_i} = \frac{\sum_{j=N_1+\ldots+N_{i-1}+1}^{N_i} y_j}{N_i}$. It means that the PI kriging prediction at repeated points is the average of the outputs at those points.

It is easy to see that $\mathbf{P}^\top = \mathbf{P}$ and $\mathbf{P}^2 = \mathbf{P}$. We now check the two remaining characteristic properties of projection matrices

1. $\forall \mathbf{u} \in Null(\mathbf{C})\ ,\ \mathbf{Pu} = \mathbf{0}$

2. $\forall \mathbf{v} \in Im(\mathbf{C})\ ,\ \mathbf{Pv} = \mathbf{v}$.

We first construct a set of non-orthogonal basis vectors of $Null(\mathbf{C})$. The basic idea is that when two columns of the covariance matrix $\mathbf{C}$ are identical, e.g., the two first columns, $\mathbf{C} = \big(\mathbf{C}^1, \mathbf{C}^1, \ldots\big)$, then vector $\mathbf{u}^1 = (1, -1, 0, \ldots, 0)^\top/\sqrt{2}$ belongs to $Null(\mathbf{C})$ because

$$\mathbf{C}^1 - \mathbf{C}^1 = \mathbf{C}\mathbf{e}_1 - \mathbf{C}\mathbf{e}_2 = \mathbf{C}(\underbrace{\mathbf{e}_1 - \mathbf{e}_2}_{\mathbf{u}^1}) = \mathbf{0}. \tag{15}$$

Generally, all such vectors can be written as

$$\mathbf{u}^j \ = \ \frac{\mathbf{e}_{j+1} - \mathbf{e}_j}{\sqrt{2}}\ ,\ j = \sum_{l \le i-1} N^l + 1, \ldots, \sum_{l \le i} N^l - 1\ ,\ i = 1, \ldots, k\ .$$

There are $N - k = dim(Null(\mathbf{C}))$ such $\mathbf{u}^j$'s which are not orthogonal but linearly independent. They make a basis of $Null(\mathbf{C})$. It can be seen that $\mathbf{Pu}^j = \mathbf{0}\ ,\ j = 1, \ldots, N-k$. Since every vector in $Null(\mathbf{C})$ is a linear combination of the $\mathbf{u}^j$'s, the equation $\mathbf{Pu} = \mathbf{0}$ holds for any vector in the null space of $\mathbf{C}$ which proves the first characteristic property of the projection matrix.

The second property is also proved by constructing a set of vectors that span $Im(\mathbf{C})$. There are $n - N + k$ such vectors. The k first vectors have the form

$$\mathbf{v}^i = (\underbrace{0, \ldots, 0}_{N_1 + \ldots + N_{i-1} \text{ times}}\ ,\ \underbrace{1, \ldots, 1}_{N_i \text{ times}}, \ 0, \ldots, 0)^\top/\sqrt{N_i}\ ,\ i = 1, \ldots, k. \tag{16}$$

The $n - N$ other vectors are: $\mathbf{v}^j = \mathbf{e}_{j-k+N}\ , j = k+1, \ldots, n-N+k$. Because these $n - N + k$ $\mathbf{v}^j$'s are linearly independent and perpendicular to the null space (to the above $\mathbf{u}^j\ ,\ j = 1, \ldots, N-k$), they span $Im(\mathbf{C})$. Furthermore, $\mathbf{Pv}^i = \mathbf{v}^j\ ,\ j = 1, \ldots, n-N+k$. The equation $\mathbf{Pv} = \mathbf{0}$ is true for every $\mathbf{v} \in Im(\mathbf{C})$, therefore, $\mathbf{P}$ is the projection matrix onto the image space of $\mathbf{C}$ and the proof is complete. □

Property 3 (Null variance of PI regularized models at data points)**.**
The variance of Gaussian processes regularized by pseudoinverse is zero at data points.

Therefore $v^{PI}(\cdot)$ is null at redundant points.

Proof: From Equation (2), the PI kriging variances at all design points are

$$v^{PI}(\mathbf{X}) \;=\; c^{PI}(\mathbf{X},\mathbf{X}) \;=\; K(\mathbf{X},\mathbf{X}) - \mathbf{c}(\mathbf{X})^\top \mathbf{C}^\dagger \mathbf{c}(\mathbf{X}) \;=\; \mathbf{C} - \mathbf{C}^\top \mathbf{C}^\dagger \mathbf{C} = \mathbf{C} - \mathbf{C} \;=\; 0\;,$$

thanks to the pseudoinverse property [Str09], $\mathbf{C}\mathbf{C}^\dagger\mathbf{C} \;=\; \mathbf{C}$. □

3.4 Nugget regularization

3.4.1 Definition and covariance orthogonality property

When regularizing a covariance matrix by nugget, a positive value, τ^2, is added to the main diagonal. This corresponds to a probabilistic model with an additive white noise of variance τ^2, $Y(\mathbf{x}) \;\mid\; Y(\mathbf{x}^i) + \varepsilon_i = \mathbf{y}_i,\;\; i = 1,\ldots,n,$ where the ε_i's are i.i.d. $\mathcal{N}(0,\tau^2)$. Nugget regularization improves the condition number of the covariance matrix by increasing all the eigenvalues by τ^2: if λ_i is an eigenvalue of $\mathbf{C}$, then $\lambda_i + \tau^2$ is an eigenvalue of $\mathbf{C} + \tau^2\mathbf{I}$ and the eigenvectors remain the same (the proof is straightforward). The associated condition number is $\kappa(\mathbf{C} + \tau^2\mathbf{I}) = \frac{\lambda_{max} \,+\, \tau^2}{\lambda_{min} \,+\, \tau^2}$. The nugget parameter causes kriging to smoothen the data and become non-interpolating.

Property 4 (Loss of interpolation in models regularized by nugget)**.**
A conditional Gaussian process regularized by nugget has its mean no longer, in general, equal to the output at data points, $m^{Nug}(\boldsymbol{x}^i) \;\neq\; y^i$, $i = 1, n$.

This property can be understood as follows. A conditional GP with invertible covariance matrix is interpolating because $c(\mathbf{x}^i)^\top \mathbf{C}^{-1}\mathbf{y} = \mathbf{C}^{i\top}\mathbf{C}^{-1}\mathbf{y} = \mathbf{e}_i^\top \mathbf{y} = y_i$. This does not stand when $\mathbf{C}^{-1}$ is replaced by $(\mathbf{C} + \tau^2\mathbf{I})^{-1}$.

Recall that the term $\mathbf{C}^{-1}\mathbf{y}$ in the kriging mean of Equation (1) is denoted by $\boldsymbol{\beta}$. When nugget regularization is used, $\boldsymbol{\beta}$ is shown as $\boldsymbol{\beta}^{Nug}$ and, thanks to the eigenvalue decomposition of $(\mathbf{C} + \tau^2\mathbf{I})^{-1}$, it is written

$$\boldsymbol{\beta}^{Nug} \;=\; \sum_{i=1}^{r} \frac{\left(\mathbf{V}^i\right)^\top \mathbf{y}}{\lambda_i + \tau^2}\mathbf{V}^i + \sum_{i=r+1}^{n} \frac{\left(\mathbf{W}^i\right)^\top \mathbf{y}}{\tau^2}\mathbf{W}^i. \tag{17}$$

The main difference between $\boldsymbol{\beta}^{PI}$ (Equation (7)) and $\boldsymbol{\beta}^{Nug}$ lies in the second part of $\boldsymbol{\beta}^{Nug}$: the part that spans the null space of the covariance matrix. In the following, we show that this term cancels out when multiplied by $\mathbf{c}(\mathbf{x})^\top$, a product that intervenes in kriging.

Property 5 (Orthogonality Property of **c** and Null(**C**))**.**
For all $\boldsymbol{x} \in D$, the covariance vector $\boldsymbol{c}(\boldsymbol{x})$ is perpendicular to the null space of the covariance matrix $\boldsymbol{C}$.

Proof: The kernel $K(.,.)$ is a covariance function [Aro50], hence the matrix

$$\mathbf{C}_x = \begin{bmatrix} K(\mathbf{x},\mathbf{x}) & \mathbf{c}(\mathbf{x})^\top \\ \mathbf{c}(\mathbf{x}) & \mathbf{C} \end{bmatrix} \tag{18}$$

is positive semidefinite.

Let $\mathbf{w}$ be a vector in the null space of $\mathbf{C}$. According to the definition of positive semidefinite matrices, we have

$$\begin{pmatrix} 1 \\ \mathbf{w} \end{pmatrix}^\top \mathbf{C}_x \begin{pmatrix} 1 \\ \mathbf{w} \end{pmatrix} = K(\mathbf{x},\mathbf{x}) + 2\sum_{i=1}^{n} K(\mathbf{x},x_i)w_i + 0 \geq 0. \tag{19}$$

The above equation is valid for any vector $\gamma\mathbf{w}$ as well, in which γ is a real number. This happens only if $\sum_{i=1}^{n} K(\mathbf{x},x_i)w_i$ is zero, that is to say, $\mathbf{c}(\mathbf{x})^\top$ is perpendicular to the null space of $\mathbf{C}$. □

As a result of the Orthogonality Property of c and Null($\mathbf{C}$), the second term in Equation (17) disappears in the kriging mean with nugget regularization which becomes

$$m^{Nug}(\mathbf{x}) = \mathbf{c}(\mathbf{x})^\top \sum_{i=1}^{r} \frac{\left(\mathbf{V}^i\right)^\top \mathbf{y}}{\lambda_i + \tau^2}\mathbf{V}^i. \tag{20}$$

The Orthogonality Property applies similarly to the kriging covariance (Equation (2)), which yields

$$\begin{aligned} c^{Nug}(\mathbf{x},\mathbf{x}') &= K(\mathbf{x},\mathbf{x}') - \mathbf{c}(\mathbf{x})^\top \sum_{i=1}^{r} \frac{\left(\mathbf{V}^i\right)^\top \mathbf{c}(\mathbf{x}')}{\lambda_i + \tau^2}\mathbf{V}^i \\ &= K(\mathbf{x},\mathbf{x}') - \sum_{i=1}^{r} \frac{\left(\left(\mathbf{V}^i\right)^\top \mathbf{c}(\mathbf{x})\right)\left(\left(\mathbf{V}^i\right)^\top \mathbf{c}(\mathbf{x}')\right)}{\lambda_i + \tau^2}. \end{aligned} \tag{21}$$

Comparing equations (8) and (20) indicates that the behavior of m^{PI} and m^{Nug} will be similar to each other if τ^2 is small. The same holds for kriging covariances (hence variances) c^{PI} and c^{Nug} in equations (9) and (21).

Property 6 (Equivalence of PI and nugget regularizations)**.**
The mean and covariance of conditional GPs regularized by nugget tend toward the ones of GPs regularized by pseudoinverse as the nugget value τ^2 tends to 0.

In addition, equations (9) and (21) show that c^{Nug} is always greater than c^{PI}. These results will be illustrated later in the Discussion Section.

3.4.2 Nugget and maximum likelihood

It is common to estimate the nugget parameter by maximum likekihood (Equation (28)). As will be detailed below, the amplitude of the nugget estimated by ML is increasing with the spread of observations at redundant points. It matches the interpretation of nugget as the amount of noise put on data: an increasing discrepancy between responses at a given point is associated to more observations noise.

In Figure 3.3 two vectors of response values are shown, $\mathbf{y}$ (bullets) and $\mathbf{y}^+$ (crosses), located at k different $\mathbf{x}$ sites. The spread of response values $\mathbf{y}^+$ is larger than that of $\mathbf{y}$ at some redundant points. Let s_i^2 and ${s_i^+}^2$, $1 \leq i \leq k$, denote the variances of $\mathbf{y}$ and $\mathbf{y}^+$ at the redundant points,

$$s_i^2 = \frac{\sum_{j=N_1+\ldots+N_{i-1}+1}^{N_1+\ldots+N_i} (y_j - \overline{y_i})^2}{N_i - 1}, \tag{22}$$

and the same stands with $\mathbf{y}^+$ and its variance ${s_i^+}^2$. The nugget that maximizes the likelihood, the other GP parameters being fixed (the length-scales θ_i and the process variance σ^2), is increasing when the variance of the outputs increases.

Theorem 1.
Suppose that there are observations located at k different sites. If we are given two vectors of response values $\boldsymbol{y}$ and $\boldsymbol{y}^+$ such that

1. ${s_i^+}^2 \geq s_i^2$ *for all* $i = 1, \ldots, k$ *and*
2. $\overline{y_i} = \overline{{y^+}_i}$ *for all* $i = 1, ..., k$,

then the nugget amplitudes $\hat{\tau}^2$ and $\widehat{{\tau^+}^2}$ that maximize the likelihood with other GP parameters being fixed are such that $\widehat{{\tau^+}^2} \geq \hat{\tau}^2$.

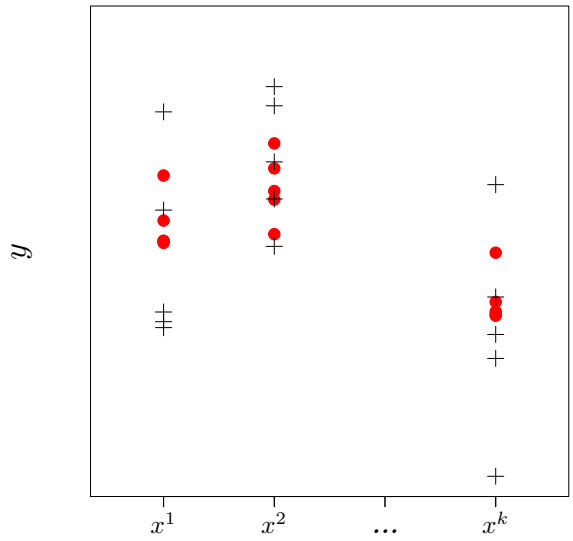


Figure 3.3: The response values $\mathbf{y}$ and $\mathbf{y}^+$ are denoted by bullets and crosses, respectively. At each location, the mean of $\mathbf{y}$ and $\mathbf{y}^+$ are identical, $\overline{y_i} = \overline{y^+}_i$, but the spread of observations in $\mathbf{y}^+$ is never less than that of $\mathbf{y}$ at redundant points.

Proof: Before starting the proof, we need equations resulting from the positive definiteness of the covariance matrix $\mathbf{C}$:

$$\mathbf{y} = \mathbf{P}_{Null(\mathbf{C})}\mathbf{y} + \mathbf{P}_{Im(\mathbf{C})}\mathbf{y} \tag{23}$$

$$\mathbf{P}_{Im(\mathbf{C})}\mathbf{y} = \sum_{i=1}^{n-N+k} \langle \mathbf{y}, \mathbf{V}^i \rangle \mathbf{V}^i \tag{24}$$

$$\mathbf{P}_{Null(\mathbf{C})}\mathbf{y} = \sum_{i=1}^{N-k} \langle \mathbf{y}, \mathbf{W}^i \rangle \mathbf{W}^i \tag{25}$$

$$\left\| \mathbf{P}_{Null(\mathbf{C})}\mathbf{y} \right\|^2 = \left\| \mathbf{y} - \mathbf{P}_{Im(\mathbf{C})}\mathbf{y} \right\|^2, \tag{26}$$

where $\langle ., . \rangle$ denotes the inner product.

The natural logarithm of the likelihood function is

$$\ln L(\mathbf{y}|\boldsymbol{\theta}, \sigma^2) = -\frac{n}{2}\ln(2\pi) - \frac{1}{2}\ln|\mathbf{C}| - \frac{1}{2}\mathbf{y}^\top \mathbf{C}^{-1}\mathbf{y}, \tag{27}$$

where after removing fixed terms and incorporating nugget effect, becomes:

$$-2\ln L(\mathbf{y}|\tau^2) \approx \ln\left(\left|\mathbf{C} + \tau^2\mathbf{I}\right|\right) + \mathbf{y}^\top \left(\mathbf{C} + \tau^2\mathbf{I}\right)^{-1}\mathbf{y}. \tag{28}$$

The eigenvalue decomposition of matrix $\mathbf{C} + \tau^2\mathbf{I}$ in (28) consists of

$$\left(\mathbf{V}^1, ..., \mathbf{V}^{n-N+k}, \mathbf{W}^1, ..., \mathbf{W}^{N-k}\right) \tag{29}$$

$$\mathbf{\Sigma} = diag(\tau^2 + \lambda_1, ..., \tau^2 + \lambda_{n-N+k}, \underbrace{\tau^2, ..., \tau^2}_{N-k}). \tag{30}$$

If Equation (28) is written based on the eigenvalue decomposition, we have

$$-2\ln L(\mathbf{y}|\tau^2) \approx \sum_{i=1}^{n} \ln(\tau^2 + \lambda_i) + \frac{1}{\tau^2}\sum_{i=1}^{N-k} \langle \mathbf{y}, \mathbf{W}^i\rangle^2 + \sum_{i=1}^{n-N+k} \frac{\langle \mathbf{y}, \mathbf{V}^i\rangle^2}{\tau^2 + \lambda_i}, \tag{31}$$

or equivalently

$$-2\ln L(\mathbf{y}|\tau^2) \approx \sum_{i=1}^{n} \ln(\tau^2 + \lambda_i) + \frac{1}{\tau^2}\left\|\mathbf{y} - \mathbf{P}_{Im(\mathbf{C})}\mathbf{y}\right\|^2 + \sum_{i=1}^{n-N+k} \frac{\langle \mathbf{P}_{Im(\mathbf{C})}\mathbf{y}, \mathbf{V}^i\rangle^2}{\tau^2 + \lambda_i}, \tag{32}$$

with the convention $\lambda_{n-N+k+1} = \lambda_{n-N+k+2} = ... = \lambda_n = 0$. In the above equations, $\approx$ means "equal up to a constant". Based on (14), the term $\mathbf{y} - \mathbf{P}_{Im(\mathbf{C})}\mathbf{y}$ in Equation (32) is

$$\mathbf{y} - \mathbf{P}_{Im(\mathbf{C})}\mathbf{y} = \begin{bmatrix} y_1 - \overline{y}_1 \\ \vdots \\ y_{N_1} - \overline{y}_1 \\ \vdots \\ y_{N_1+...+N_{k-1}+1} - \overline{y}_k \\ \vdots \\ y_{N_1+...+N_k} - \overline{y}_k \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{33}$$

where $\overline{y}^i,\ i = 1, ..., k$, designates the mean of response values at location i.
According to equations (33) and (22), $\left\|\mathbf{y} - \mathbf{P}_{Im(\mathbf{C})}\mathbf{y}\right\|^2 = \sum_{i=1}^{k} N_i s_i^2$. Hence, Equation (32) using s_i^2 is updated as

$$-2\ln L(\mathbf{y}|\tau^2) \approx \sum_{i=1}^{n} \ln(\tau^2 + \lambda_i) + \frac{1}{\tau^2}\sum_{i=1}^{k} N_i s_i^2 + \sum_{i=1}^{n-N+k} \frac{\langle \mathbf{P}_{Im(\mathbf{C})}\mathbf{y}, \mathbf{V}^i\rangle^2}{\tau^2 + \lambda_i}. \tag{34}$$

Let function $\Delta(\tau^2)$ express the difference between$-2\ln L(\mathbf{y}|\tau^2)$ and $-2\ln L(\mathbf{y}^+|\tau^2)$. Remark that $\mathbf{P}_{Im(\mathbf{C})}\mathbf{y} = \mathbf{P}_{Im(\mathbf{C})}\mathbf{y}^+$ because of our hypothesis $\overline{y}^i = \overline{y}^{+i}\ , i = 1, ..., k$. The

function $\Delta(\tau^2)$ is defined as

$$\Delta(\tau^2) \equiv -2\ln L(\mathbf{y}^+|\tau^2) + 2\ln L(\mathbf{y}|\tau^2) = \frac{1}{\tau^2}\sum_{i=1}^{k} N_i\left(s_i^{+2} - s_i^2\right), \tag{35}$$

and is monotonically decreasing.

Now we show that $\widehat{\tau^+}^2$, the ML estimation of nugget from $\mathbf{y}^+$, is never smaller than $\hat{\tau}^2$, the ML estimation of nugget from $\mathbf{y}$. Firstly, $\widehat{\tau^+}^2$ cannot be smaller than $\hat{\tau}^2$. Indeed, if $\tau^2 \leq \hat{\tau}^2$, then

$$\begin{aligned} -2\ln L(\mathbf{y}^+|\tau^2) &= -2\ln L(\mathbf{y}|\tau^2) + \Delta(\tau^2) \\ &\geq -2\ln L(\mathbf{y}|\hat{\tau}^2) + \Delta(\tau^2) \\ &\geq -2\ln L(\mathbf{y}|\hat{\tau^2}) + \Delta(\hat{\tau}^2) \\ &= -2\ln L(\mathbf{y}^+|\hat{\tau}^2), \end{aligned} \tag{36}$$

which shows that $\widehat{\tau^+}^2 \geq \hat{\tau}^2$. Secondly, if s_i^{+2} is strictly larger than s_i^2, then $\widehat{\tau^+}^2 > \hat{\tau}^2$ because the slope of $-2\ln L(\mathbf{y}^+|\tau^2)$ is strictly negative at $\tau^2 = \hat{\tau}^2$: The derivative of $-2\ln L(\mathbf{y}^+|\tau^2)$ with respect to τ^2 can be written as

$$\frac{d}{d\tau^2}\left(-2\ln L(\mathbf{y}^+|\tau^2)\right) = \frac{d}{d\tau^2}\left(-2\ln L(\mathbf{y}|\tau^2)\right) + \frac{d\Delta(\tau^2)}{d\tau^2}. \tag{37}$$

Since $\hat{\tau}^2 = \arg\min -2\ln L(\mathbf{y}|\tau^2)$, the second term in the right hand side of the above equation is equal to zero. Therefore, the derivative of $-2\ln L(\mathbf{y}^+|\tau^2)$ with respect to τ^2 reduces to

$$\frac{d}{d\tau^2}\left(-2\ln L(\mathbf{y}^+|\hat{\tau}^2)\right) = \frac{d}{d\tau^2}\left(\frac{1}{\tau^2}\sum_{i=1}^{k} N_i\left(s_i^{+2} - s_i^2\right)\right) = \frac{-1}{\tau^4}\sum_{i=1}^{k} N_i\left(s_i^{+2} - s_i^2\right). \tag{38}$$

The above derivative is strictly negative because $s_i^{+2} - s_i^2$ is positive and the proof is complete. □

3.5 Discussion: choice and tuning of the classical regularization methods

This section carries out a practical comparison of PI and nugget regularization methods, which are readily available in most GP softwares [STT00, RGD12]. We start with a discussion of how data and model match, which further allows to decide whether nugget or PI should be used. Finally, we provide guidelines to tune the regularization parameters.

Note that nugget regularization should be used when the observed data is known to be noisy since it has a physical meaning [RGD12]. The loss of the interpolating property at data points associated to nugget regularization is here a beneficial filtering effect. This discussion on non-deterministic outputs is out of the scope of this work.

3.5.1 Model-data discrepancy

Model-data discrepancy can be measured as the distance between the observations $\mathbf{y}$ and the GP model regularized by pseudoinverse.

Definition 2 (Model-data discrepancy)**.** *Let* $\mathbf{X}$ *be a set of design points with associated observations* $\mathbf{y}$*. Let* $\mathbf{V}$ *and* $\mathbf{W}$ *be the normalized eigenvectors spanning the image space and the null space of the covariance matrix* $\mathbf{C}$*, respectively. The model-data discrepancy is defined as*

$$discr \;=\; \frac{\|\mathbf{y} - m^{PI}(\mathbf{X})\|^2}{\|\mathbf{y}\|^2} \;=\; \frac{\|\mathbf{W}\mathbf{W}^\top \mathbf{y}\|^2}{\|\mathbf{y}\|^2} \tag{39}$$

where $m^{PI}(.)$ *is the pseudoinverse regularized GP model of Equation (10).*

The last equality in the definition of *discr* comes from Equations (11) and (12). The discrepancy is a normalized scalar, $0 \leq discr \leq 1$, where $discr = 0$ indicates that the model and the data are perfectly compatible, and vice versa when $discr = 1$. The definition of redundant points does not depend on the observations $\mathbf{y}$ and the model-data discrepancy is a scalar globalizing the contributions of all observations. An intermediate object between redundant points and discrepancy is the gradient of the squared model-data error with respect to the observations,

$$\nabla_{\mathbf{y}} \|\mathbf{y} - m^{PI}(\mathbf{X})\|^2 \;=\; \mathbf{W}\mathbf{W}^\top \mathbf{y} \;. \tag{40}$$

It appears that the gradient of the error, $\|\mathbf{y} - m^{PI}(\mathbf{X})\|^2$, is equal to the model-data distance, $\mathbf{W}\mathbf{W}^\top \mathbf{y}$. This property comes from the quadratic form of the error. The magnitude of the components of the vector $\mathbf{W}\mathbf{W}^\top \mathbf{y}$ measures the sensitivity of the error to a particular observation. At repeated points, a gradient-based approach where the y's are optimized would advocate to make the observations closer to their mean proportionally to their distance to the mean.

In other words, $-\mathbf{W}\mathbf{W}^\top \mathbf{y}$ is a direction of reduction of the model-data distance in the space of observations. Because the distance considered is quadratic, this direction is

colinear to the error, $(\mathbf{y} - m^{PI}(\mathbf{X}))$. The indices of the non-zero components of $\mathbf{W}\mathbf{W}^\top\mathbf{y}$ also designate the redundant points.

3.5.2 Two detailed examples

A common practice when the nugget value, τ^2, is not known beforehand is to estimate it by ML or cross-validation. We showed that the ML estimated nugget value, $\hat{\tau}^2$, is increasing with the spread of responses at redundant points. This is one situation (among others, e.g., the additive example hereafter) where the data and the model mismatch, and $\hat{\tau}^2$ is large. Figure 3.4 is an example where $\hat{\tau}^2$ is equal to 7.06. Some authors such as in [Wag10, Bac13] recommend using cross-validation instead of ML for learning the kriging parameters. In the example of Figure 3.4, the estimated nugget value by leave-one-out cross-validation, denoted by $\hat{\tau}^2_{CV}$, is 1.75. The dash-dotted lines represent the kriging model regularized by nugget that is estimated by cross-validation. The model-data discrepancy is $discr = 0.36$ and $\mathbf{W}\mathbf{W}^\top\mathbf{y} = (0, 0, -3, 3, 0, 0)^\top$ which shows that points 3 and 4 are redundant and their outputs should be made closer to reduce the model-data error. Whether or not in practice the outputs can be controlled is out of the scope of our discussion. But our analysis considers data points that are not compatible with the model.

We now give a two-dimensional example of a kriging model with additive kernel defined over $\mathbf{X} = [(1,1),(2,1),(1,2),(2,2),(1.5,1.5),(1.25,1.75),(1.75,1.25)]$, cf. Figure 3.5. As explained in Section 3.2.2, the first four points of the DoE make the additive covariance matrix non-invertible even though the points are not near each other in Euclidean distance. Suppose that the design points have the response values $\mathbf{y} = (1, 4, -2, 1, 1, -0.5, 2.5)^\top$ which correspond to the additive true function $f(\mathbf{x}) = x_1^2 - x_2^2 + 1$. The covariance matrix is the sum of two parts

$$\mathbf{C}_{add} = \sigma_1^2 K_1 + \sigma_2^2 K_2 \ ,$$

where σ_i^2 are the process variances and $\sigma_i^2 K_i$ the kernel in dimension $i = 1, 2$.

To estimate the parameters of $\mathbf{C}_{add}$, the negative of the likelihood is minimized (see Equation (28)) which yields a nugget value $\hat{\tau}^2 \approx 10^{-12}$ (the lower bound on nugget used). A small nugget value is obtained because the associated output value follows an additive function compatible with the kernel: there is no discrepancy between the model and the data. Because of the small nugget value, the models regularized by PI and nugget are very

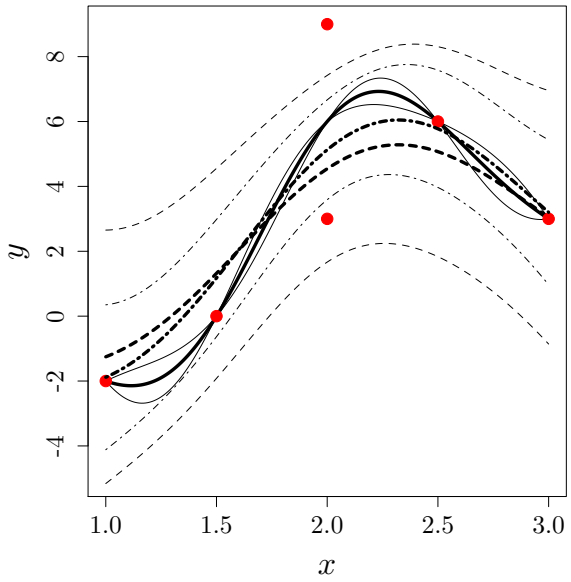


Figure 3.4: Comparison of kriging regularized by PI (solid lines), nugget estimated by ML (dashed lines) and nugget estimated by cross-validation (dash-dotted lines). $\mathbf{X} = [1; 1.5; 2; 2.00001; 2.5; 3]$ and $\mathbf{y} = (-2, 0, 3, 9, 6, 3)^\top$. The estimated nugget values are $\hat{\tau}^2 = 7.06$ and $\hat{\tau}^2_{CV} = 1.75$.

close to each other (the left picture in Figure 3.5).

Let us now introduce model-data discrepancy in this example: the observations of the first four data points no longer follow an additive function after changing the third response from -2 to 2; additive kriging models cannot interpolate these outputs. The nugget value estimated by ML is equal to 1.91, so $m^{Nug}(\mathbf{x})$ does not interpolate any of the data points ($\mathbf{x}^1$ to $\mathbf{x}^7$). Regarding $m^{PI}(\mathbf{x})$, the projection onto $Im(\mathbf{C})$ make the GP predictions different from the observations at $\mathbf{x}^1$, $\mathbf{x}^2$, $\mathbf{x}^3$ and $\mathbf{x}^4$. For example, $m^{PI}(\mathbf{x}^4) = 2$. The projection applied to points $\mathbf{x}^5$ to $\mathbf{x}^7$ where no linear dependency exists show that $m^{PI}(\mathbf{x})$ is interpolating there, which is observed on the right picture of Figure 3.5.

Our observations reflect that large estimated values of nugget (whether by ML or cross-validation) indicate model-data discrepancy. This agrees with the calculated discrepancies: in the last additive kernel example when all the outputs were additive, $discr = 0$ and $\mathbf{W}\mathbf{W}^\top\mathbf{y} = (0, 0, 0, 0, 0, 0, 0)^\top$ (no redundant point); when the value of the third output was increased to 2, $discr = 0.37$ and $\mathbf{W}\mathbf{W}^\top\mathbf{y} = (-1, 1, 1, -1, 0, 0, 0)^\top$ showing that points 1 to 4 are redundant and that, to reduce model error, points 1 and 4 should increase their

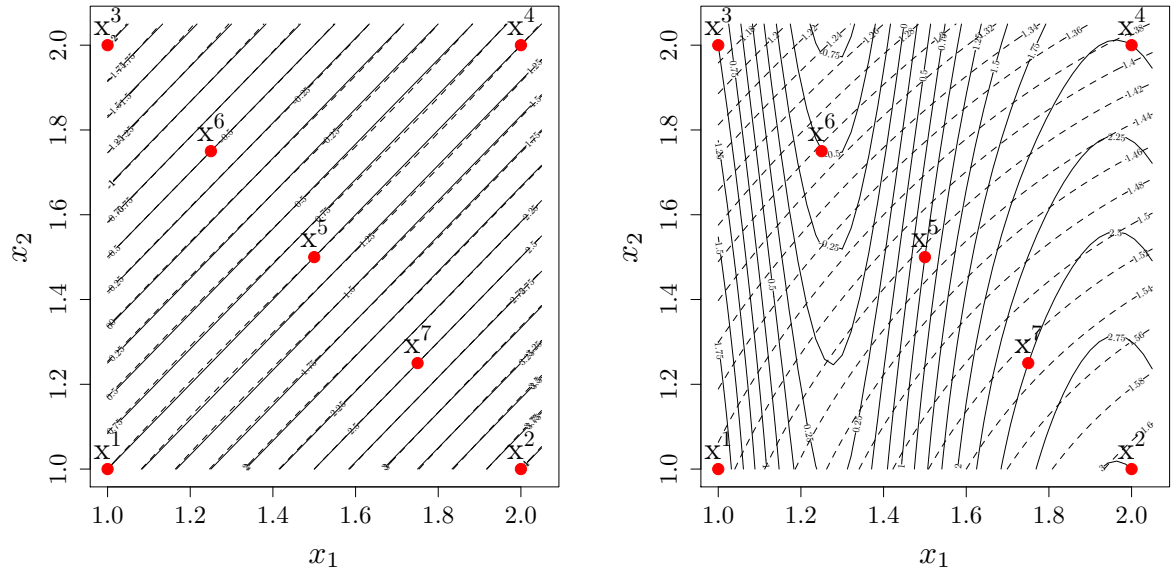


Figure 3.5: Contour plots of kriging mean regularized by pseudoinverse (solid line) vs. nugget (dashed line) for an additive GP. The bullets are data points. Left: the response values are additive, $\mathbf{y} = (1, 4, -2, 1, 1, -0.5, 2.5)^\top$ and $\hat{\tau}^2 = 10^{-12}$. Right: the third observation is replaced by 2, creating non-additive observations and $\hat{\tau}^2 \approx 1.91$; $m^{Nug}(\mathbf{x})$ is no longer interpolating, $m^{PI}(\mathbf{x})$ still interpolates $\mathbf{x}^5$ to $\mathbf{x}^7$.

outputs while points 2 and 3 should decrease theirs.

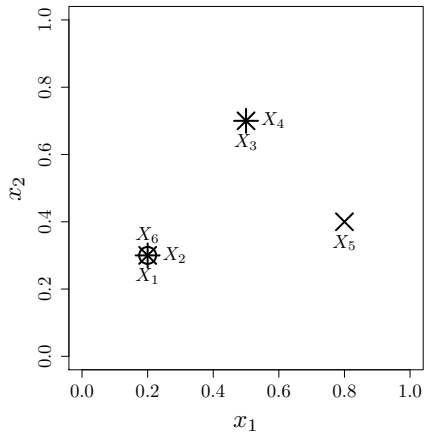
Of course, for the sole purpose of quantifying model-data discrepancy it is more efficient to use Formula (39) which involves one pseudo-inverse calculation and two matrix products against a nonlinear likelihood maximization with repeated embedded $\mathbf{C}$ eigenvalues analyses for the nugget estimation.

3.5.3 Examples of redundant points

This section gives easily interpretable examples of DoEs with associated kernels that make the covariance matrix non-invertible. The eigenvalues, eigenvectors and orthogonal projection matrix onto the image space (cf. also Section 3.2.3) are described.

Repeated points

Repeated design points are the simplest example of redundancy in a DoE since columns of the covariance matrix $\mathbf{C}$ are duplicated. An example is given in Figure 3.6 with a two-dimensional design, and a classical squared exponential kernel. The eigenvalues and



$$k(\mathbf{x}, \mathbf{x'}) = \exp\left(-\frac{(x_1 - x_1')^2}{2 \times .25^2}\right) \times \exp\left(-\frac{(x_2 - x_2')^2}{2 \times .25^2}\right)$$

$$\mathbf{X} = \begin{bmatrix} 0.20 & 0.30 \\ 0.20 & 0.30 \\ 0.50 & 0.70 \\ 0.50 & 0.70 \\ 0.80 & 0.40 \\ 0.20 & 0.30 \end{bmatrix}$$

Figure 3.6: Kernel and DoE of the repeated points example

eigenvectors of the covariance matrix associated to Figure 3.6 are

$$\boldsymbol{\lambda} = \begin{bmatrix} 3.12 \\ 1.99 \\ 0.90 \\ 0.00 \\ 0.00 \\ 0.00 \end{bmatrix}, \ \mathbf{V} = \begin{bmatrix} -0.55 & 0.19 & 0.00 \\ -0.55 & 0.19 & 0.00 \\ -0.22 & -0.64 & -0.21 \\ -0.22 & -0.64 & -0.21 \\ -0.09 & -0.28 & 0.96 \\ -0.55 & 0.19 & 0.00 \end{bmatrix} \text{ and } \mathbf{W} = \begin{bmatrix} 0.00 & -0.30 & 0.76 \\ -0.71 & 0.12 & -0.39 \\ -0.04 & 0.66 & 0.26 \\ 0.04 & -0.66 & -0.26 \\ 0.00 & 0.00 & 0.00 \\ 0.71 & 0.18 & -0.37 \end{bmatrix},$$

with the orthogonal projection matrix onto $Im(\mathbf{C})$

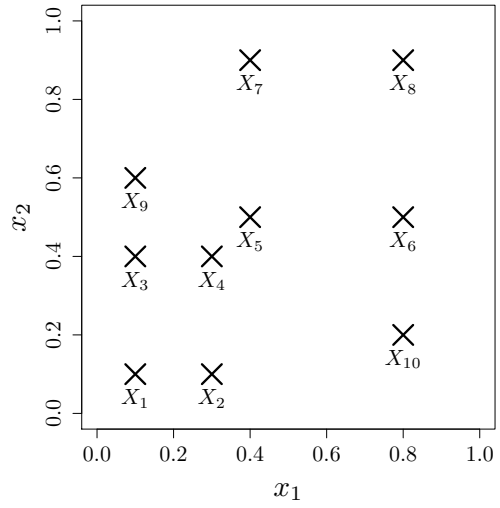
$$\mathbf{V}\mathbf{V}^\top = \begin{bmatrix} 0.33 & 0.33 & 0.00 & 0.00 & 0.00 & 0.33 \\ 0.33 & 0.33 & 0.00 & 0.00 & 0.00 & 0.33 \\ 0.00 & 0.00 & 0.50 & 0.50 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.50 & 0.50 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 1.00 & 0.00 \\ 0.33 & 0.33 & 0.00 & 0.00 & 0.00 & 0.33 \end{bmatrix}$$

Points $\{1, 2, 6\}$ and $\{3, 4\}$ are repeated and redundant.

First additive example

The first example of GP with additive kernel is described in Figure 3.7. As explained in Section 3.2.2, the rectangular patterns of points $\{1, 2, 3, 4\}$ and $\{5, 6, 7, 8\}$ create linear dependencies between the columns of $\mathbf{C}$. The eigenvalues and eigenvectors of the covariance matrix are,

$$\boldsymbol{\lambda} = \begin{bmatrix} 9.52 \\ 3.58 \\ 2.60 \\ 2.31 \\ 1.46 \\ 0.39 \\ 0.09 \\ 0.06 \\ 0.00 \\ 0.00 \end{bmatrix}, \ \mathbf{V} = \begin{bmatrix} -0.30 & -0.32 & 0.45 & -0.15 & 0.34 & -0.10 & 0.22 & 0.40 \\ -0.33 & -0.24 & 0.29 & -0.43 & -0.22 & -0.30 & -0.43 & 0.04 \\ -0.38 & -0.22 & -0.01 & 0.31 & 0.22 & 0.59 & 0.17 & 0.17 \\ -0.41 & -0.14 & -0.17 & 0.04 & -0.34 & 0.40 & -0.47 & -0.19 \\ -0.38 & 0.01 & -0.37 & 0.03 & -0.40 & -0.29 & 0.43 & 0.18 \\ -0.28 & 0.45 & 0.03 & 0.44 & -0.13 & -0.27 & -0.15 & 0.40 \\ -0.25 & 0.19 & -0.38 & -0.62 & 0.11 & 0.13 & 0.30 & -0.07 \\ -0.15 & 0.64 & 0.02 & -0.22 & 0.38 & 0.15 & -0.29 & 0.15 \\ -0.34 & -0.13 & -0.24 & 0.26 & 0.54 & -0.43 & -0.10 & -0.51 \\ -0.25 & 0.34 & 0.59 & 0.05 & -0.22 & 0.08 & 0.35 & -0.54 \end{bmatrix}$$



$$k(\mathbf{x}, \mathbf{x'}) = \exp\left(-\frac{(x_1 - x_1')^2}{2 \times .25^2}\right) + \exp\left(-\frac{(x_2 - x_2')^2}{2 \times .25^2}\right)$$

$$\mathbf{X} = \begin{bmatrix} 0.10 & 0.10 \\ 0.30 & 0.10 \\ 0.10 & 0.40 \\ 0.30 & 0.40 \\ 0.40 & 0.50 \\ 0.80 & 0.50 \\ 0.40 & 0.90 \\ 0.80 & 0.90 \\ 0.10 & 0.60 \\ 0.80 & 0.20 \end{bmatrix}$$

Figure 3.7: Kernel and DoE of the first additive GP example

$$\text{and } \mathbf{W} = \begin{bmatrix} 0.00 & 0.50 \\ 0.00 & -0.50 \\ 0.00 & -0.50 \\ 0.00 & 0.50 \\ 0.50 & 0.00 \\ -0.50 & 0.00 \\ -0.50 & 0.00 \\ 0.50 & 0.00 \\ 0.00 & 0.00 \\ 0.00 & 0.00 \end{bmatrix}.$$

The projection matrix onto the image space is

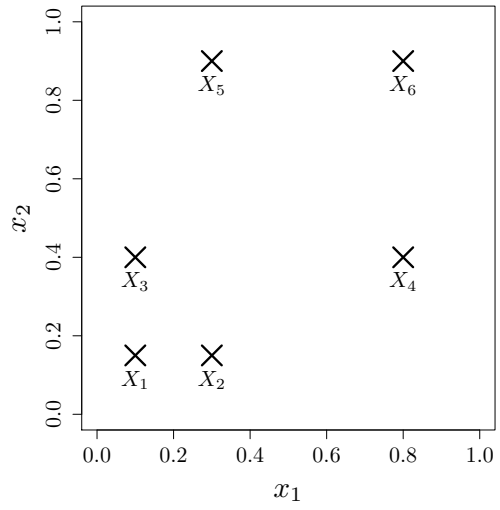
$$\mathbf{V}\mathbf{V}^\top = \begin{bmatrix} 0.75 & 0.25 & 0.25 & -0.25 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.25 & 0.75 & -0.25 & 0.25 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.25 & -0.25 & 0.75 & 0.25 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ -0.25 & 0.25 & 0.25 & 0.75 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.75 & 0.25 & 0.25 & -0.25 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.25 & 0.75 & -0.25 & 0.25 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.25 & -0.25 & 0.75 & 0.25 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & -0.25 & 0.25 & 0.25 & 0.75 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 \end{bmatrix}.$$

The redundancy between points 1 to 4 on the one hand, and 5 to 8 on the other hand, is readily seen on the matrix.

Second additive example

This example shows how an incomplete rectangular pattern with additive kernels can also make covariance matrices singular. In Figure 3.8, the point at coordinates $(0.3, 0.4)$, which is not in the design, has a GP response defined twice, once by the points $\{1, 2, 3\}$ and once by the points $\{4, 5, 6\}$. This redundancy in the DoE explains why $\mathbf{C}$ has one null eigenvalue:

$$\boldsymbol{\lambda} = \begin{bmatrix} 5.75 \\ 2.90 \\ 2.07 \\ 0.80 \\ 0.49 \\ 0.00 \end{bmatrix}, \ \mathbf{V} = \begin{bmatrix} -0.50 & 0.34 & -0.01 & 0.18 & 0.66 \\ -0.49 & 0.25 & 0.20 & 0.57 & -0.40 \\ -0.48 & 0.17 & -0.29 & -0.69 & -0.01 \\ -0.32 & -0.39 & -0.65 & 0.17 & -0.35 \\ -0.36 & -0.28 & 0.66 & -0.33 & -0.28 \\ -0.20 & -0.75 & 0.09 & 0.15 & 0.45 \end{bmatrix}, \ \mathbf{W} = \begin{bmatrix} -0.41 \\ 0.41 \\ 0.41 \\ -0.41 \\ -0.41 \\ 0.41 \end{bmatrix}.$$



$$k(\mathbf{x}, \mathbf{x'}) = \exp\left(-\frac{(x_1 - x_1')^2}{2 \times .25^2}\right) + \exp\left(-\frac{(x_2 - x_2')^2}{2 \times .25^2}\right)$$

$$\mathbf{X} = \begin{bmatrix} 0.10 & 0.15 \\ 0.30 & 0.15 \\ 0.10 & 0.40 \\ 0.80 & 0.40 \\ 0.30 & 0.90 \\ 0.80 & 0.90 \end{bmatrix}$$

Figure 3.8: Kernel and DoE of the second additive GP example

The orthogonal projection matrix onto the image space of $\mathbf{C}$ tells us that all the points in the design are redundant,

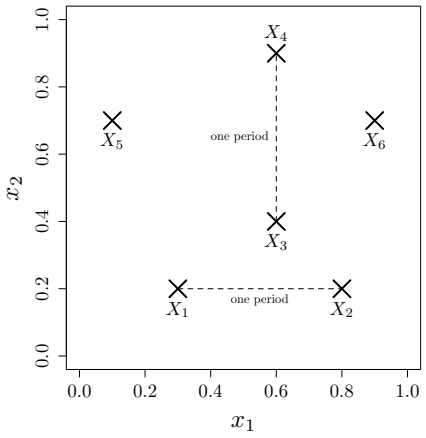
$$\mathbf{V}\mathbf{V}^\top = \begin{bmatrix} 0.83 & 0.17 & 0.17 & -0.17 & -0.17 & 0.17 \\ 0.17 & 0.83 & -0.17 & 0.17 & 0.17 & -0.17 \\ 0.17 & -0.17 & 0.83 & 0.17 & 0.17 & -0.17 \\ -0.17 & 0.17 & 0.17 & 0.83 & -0.17 & 0.17 \\ -0.17 & 0.17 & 0.17 & -0.17 & 0.83 & 0.17 \\ 0.17 & -0.17 & -0.17 & 0.17 & 0.17 & 0.83 \end{bmatrix}.$$

Periodic example

The kernel and DoE of the periodic example are given in Figure 3.9.

The eigenvalues and eigenvectors of the associated covariance matrix $\mathbf{C}$ are,

$$\boldsymbol{\lambda} = \begin{pmatrix} 2.00 \\ 2.00 \\ 1.01 \\ 0.99 \\ 0.00 \\ 0.00 \end{pmatrix}, \ \mathbf{V} = \begin{bmatrix} -0.50 & 0.50 & 0.01 & -0.01 \\ -0.50 & 0.50 & 0.01 & -0.01 \\ -0.50 & -0.50 & 0.01 & -0.01 \\ -0.50 & -0.50 & 0.01 & -0.01 \\ -0.03 & 0.00 & -0.70 & 0.72 \\ 0.00 & 0.00 & -0.72 & -0.70 \end{bmatrix} \text{ and } \mathbf{W} = \begin{bmatrix} 0.00 & 0.71 \\ 0.00 & -0.71 \\ 0.71 & 0.00 \\ -0.71 & 0.00 \\ 0.00 & 0.00 \\ 0.00 & 0.00 \end{bmatrix}.$$



$$k(\mathbf{x}, \mathbf{x'}) = \exp\left(-\frac{\sin(4\pi(x_1 - x_1'))^2}{2 \times .25^2}\right) \times \exp\left(-\frac{\sin(4\pi(x_2 - x_2'))^2}{2 \times .25^2}\right)$$

$$\mathbf{X} = \begin{bmatrix} 0.30 & 0.20 \\ 0.80 & 0.20 \\ 0.60 & 0.40 \\ 0.60 & 0.90 \\ 0.10 & 0.70 \\ 0.90 & 0.70 \end{bmatrix}$$

Figure 3.9: Kernel and DoE of the periodic example

There are two null eigenvalues. The projector onto the image space is

$$\mathbf{V}\mathbf{V}^\top = \begin{bmatrix} 0.50 & 0.50 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.50 & 0.50 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.50 & 0.50 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.50 & 0.50 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 1.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 \end{bmatrix}$$

which shows that points 1 and 2, on the one hand, and points 3 and 4, on the other hand, are redundant.

Dot product kernel example

The non-stationary dot product or linear kernel is $k(\mathbf{x}, \mathbf{x'}) = 1 + \mathbf{x}^\top \mathbf{x'}$.
We consider a set of three one dimensional, non-overlapping, observation points: $\mathbf{X} =$

$\begin{bmatrix} 0.20 \\ 0.60 \\ 0.80 \end{bmatrix}$. The associated eigenvalues and eigenvectors are,

$$\boldsymbol{\lambda} = \begin{bmatrix} 3.90 \\ 0.14 \\ 0.00 \end{bmatrix}, \ \mathbf{V} = \begin{bmatrix} -0.49 & 0.83 \\ -0.59 & -0.09 \\ -0.64 & -0.55 \end{bmatrix} \text{ and } \mathbf{W} = \begin{bmatrix} 0.27 \\ -0.80 \\ 0.53 \end{bmatrix}$$

The projection matrix onto the image space of $\mathbf{C}$ is

$$\mathbf{V}\mathbf{V}^\top = \begin{bmatrix} 0.93 & 0.21 & -0.14 \\ 0.21 & 0.36 & 0.43 \\ -0.14 & 0.43 & 0.71 \end{bmatrix}$$

Because there are 3 data points which is larger than $d + 1 = 2$, all points are redundant. With less than 3 data points, the null space of $\mathbf{C}$ is empty.

3.5.4 PI or nugget?

On the one hand, models regularized by PI have predictions, $m^{PI}(.)$, that interpolate uniquely defined points and go through the average output at redundant points (Property 2). The associated kriging variances, $v^{PI}(.)$, are null at redundant points (Property 3). On the other hand, models regularized by nugget have predictions which are neither interpolating nor averaging (Property 4) while their variances are non-zero at data points. Note that kriging variance tends to σ^2 as the nugget value increases (see Equation (21)). These facts can be observed in Figure 3.10. Additionally, this Figure illustrates that nugget regularization tends to PI regularization as the nugget value decreases (Property 6). If there is a good agreement between the data and the GP model, the PI regularization or equivalently, a small nugget, should be used. This can also be understood through the Definition of model-data discrepancy and Property 1: when $discr = 0$, the observations are perpendicular to $Null(\mathbf{C})$ and, equivalently, $m^{PI}(\mathbf{X}) = \mathbf{y}$ since $m^{PI}(.)$ performs a projection onto $Im(\mathbf{C})$. Vice versa, if the model-data discrepancy measure is large, choosing PI or nugget regularization is a matter of choice: either the prediction averaging property is regarded as most important and PI should be used, or a non-zero variance at redundant points is favored and nugget should be selected; If the discrepancy is concentrated on few redundant points, nugget regularized models will distribute the uncertainty (additional model

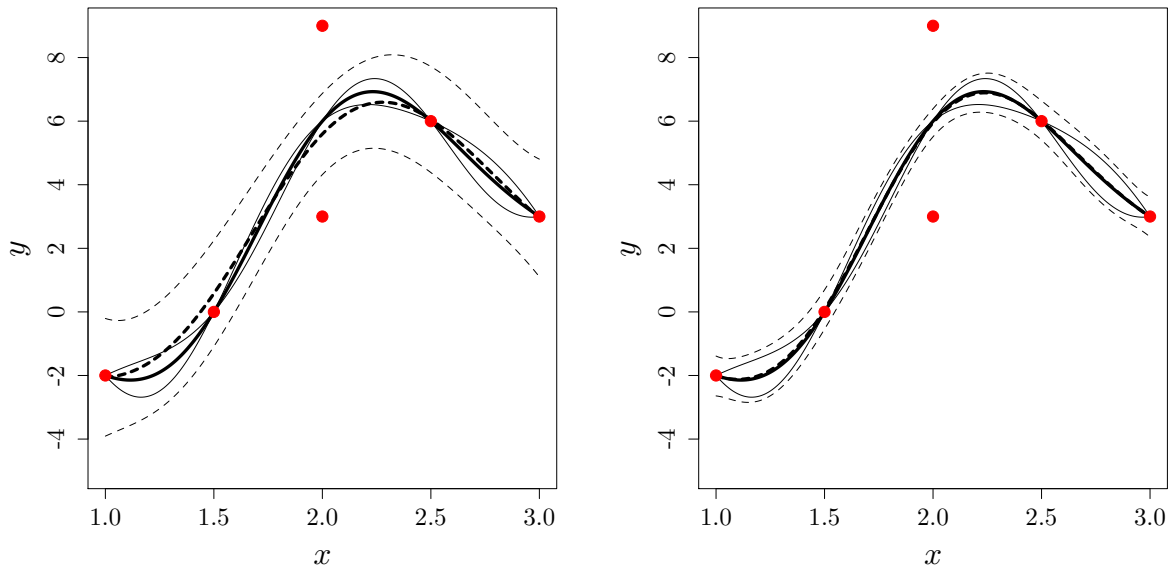


Figure 3.10: One dimensional kriging regularized by PI (solid lines) and nugget (dashed lines). The nugget amplitude is 1 on the left and 0.1 on the right. The cut-off eigenvalue for the pseudoinverse is $\eta = 10^{-3}$. $m^{Nug}(x)$ is not interpolating which is best seen at the second point on the left. On the right, the PI and nugget models are closer to each other. Same $\mathbf{X}$ and $\mathbf{y}$ as Figure 3.4.

variance) throughout the x domain while PI regularized models will ignore it. Based on the above argument, the decision for using PI or nugget regularizations should be problem dependent.

3.5.5 Tuning regularization parameters

How small can a nugget value be? Adding nugget to the main diagonal of a covariance matrix increments all the eigenvalues by the nugget amplitude. The condition number of the covariance matrix with nugget is $\kappa(\mathbf{C}+\tau^2\mathbf{I}) = \frac{\lambda_{max}+\tau^2}{\lambda_{min}+\tau^2}$. Accordingly, a "small" nugget is the smallest value of τ^2 such that $\kappa(\mathbf{C}+\tau^2\mathbf{I})$ is less than a reasonable condition number after regularization, κ_{max} (say, $\kappa_{max} = 10^8$). With such targeted condition number, the smallest nugget would be $\tau^2 = \frac{\lambda_{max}-\kappa_{max}\lambda_{min}}{\kappa_{max}-1}$ if $\lambda_{max}-\kappa_{max}\lambda_{min} \geq 0$, $\tau^2 = 0$ otherwise.

Computing a pseudoinverse also involves a parameter, the positive threshold η below which an eigenvalue is considered as null. The eigenvectors associated to eigenvalues smaller than η are numerically regarded as null space basis vectors (even though they may not, strictly speaking, be part of the null space). A suitable threshold should filter out eigenvectors associated to points that are almost redundant. The heuristic we propose is to tune η so that λ_1/η, which is an upper bound of the PI condition number[2], is equal to κ_{max}, i.e., $\eta = \lambda_1/\kappa_{max}$.

In the example shown in Figure 3.11, the covariance matrix is not numerically invertible because the points 3 and 4 are near $x = 2$. The covariance matrix has six eigenvalues, $\lambda_1 = 34.89 \geq ... \geq \lambda_5 = 0.86 \geq \lambda_6 = 8.42 \times 10^{-11} \approx 0$ and the eigenvector related to the smallest eigenvalue is $\mathbf{W}^1 = (\mathbf{e}^4 - \mathbf{e}^3)/\sqrt{2}$. In Figure 3.11, we have selected $\eta = 10^{-3}$, hence $\kappa_{PI}(\mathbf{C}) = 40.56$. Any value of η in the interval $\lambda_6 < \eta < \lambda_5$ would have yielded the same result. But if the selected tolerance were e.g., $\eta = 1$, which is larger than λ_5, the obtained PI kriging model no longer interpolates data points.

[2]By PI condition number we mean $\kappa_{PI}(\mathbf{C}) = \|\mathbf{C}\|\|\mathbf{C}^\dagger\| = \lambda_1/\lambda_r \leq \lambda_1/\eta$

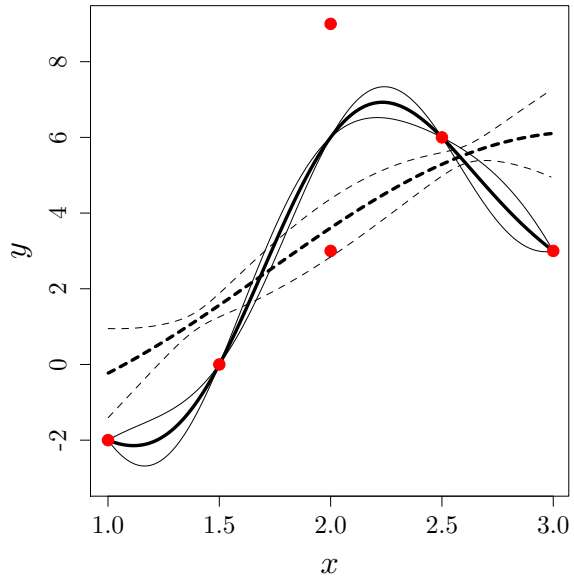


Figure 3.11: Effect of the tolerance η on the kriging model regularized by PI. Dashed line, $\eta = 1$; continuous line, $\eta = 10^{-3}$. Except for η, the setting is the same as that of Figure 3.10. When the tolerance is large ($\eta = 1$), the 5th eigenvector is deleted from the effective image space of $\mathbf{C}$ in addition to the 6th eigenvector, and the PI regularized model is no longer interpolating. Same $\mathbf{X}$ and $\mathbf{y}$ as Figure 3.4.

3.6 Interpolating Gaussian distributions

3.6.1 Interpolation and repeated points

In our context of deterministic experiments, we are interested in interpolating data. The notion of interpolation should be clarified in the case of repeated points with different outputs (e.g., Figure 3.3) as a function cannot interpolate them. Here, we seek GPs that have the following interpolation properties.

Definition 3 (Interpolation properties at repeated points)**.** *A GP exhibits interpolation properties when*

- *its trajectories pass through uniquely defined data points (therefore the GP has a null variance there),*
- *and at repeated points the GP's mean and variance are the empirical average and variance of the outputs, respectively.*

The following GP model has the above interpolation properties for deterministic outputs, even in the presence of repeated points. In this sense, it can be seen as a new regularization technique, although its potential use goes beyond regularization.

3.6.2 A GP model with interpolation properties

Here, we introduce a new GP model with the desirable interpolation properties in the presence of repeated points. This model which is called *distribution-wise* model is not degenerated and, therefore, can be regarded as a regularization method. Moreover, it is computationally more efficient than the *point-wise* GP models.

Following the same notations as in Section 3.3.2, the model is built from observations at k different $\mathbf{x}$ sites. The basic assumption is that, at each location, we consider repeated points as realizations of random variables of known joint Gaussian probability distribution. In distribution-wise GP, it is assumed that the distribution at each location is observed (hence known), as opposed to usual conditional GPs where only values of the process are observed, hence the name "distribution-wise GP". Let $Z(\mathbf{x}^i) \sim \mathcal{N}(\mu_{Z_i}, \sigma^2_{Z_i})$ denotes the probability distribution at location $\mathbf{x}^i$, $i = 1, \ldots, k$. Together, the k sets of observations

make the random vector $\mathbf{Z} = (Z(\mathbf{x}^1), \ldots, Z(\mathbf{x}^k)) \sim \mathcal{N}(\boldsymbol{\mu}_Z, \boldsymbol{\Gamma}_Z)$ in which the diagonal elements of the matrix $\boldsymbol{\Gamma}_Z$ is made of the $\sigma^2_{Z_i}$'s.

The distribution-wise GP is derived in two steps through conditioning: first it is assumed that the vector $\mathbf{Z}$ is given, and the usual conditional GP (kriging) formula can be applied; then the randomness of $\mathbf{Z}$ is accounted for and the conditional mean and variance of the distribution-wise GP, m^{Dist} and v^{Dist} respectively, come from the laws of total expectation and variance applied to $\mathbf{Z}$ and the GP outcomes $\omega \in \Omega$:

$$\begin{aligned} m^{Dist}(\mathbf{x}) &= \mathbb{E}_Z\left(\mathbb{E}_\Omega(Y(\mathbf{x})|Y(\mathbf{x}^i) = Z(\mathbf{x}^i)\ ,\ 1 \le i \le k\right) \ = \\ &\mathbb{E}_Z\left(\mathbf{c}_Z(\mathbf{x})^\top \mathbf{C}_Z^{-1}\mathbf{Z}\right) \ = \ \mathbf{c}_Z(\mathbf{x})^\top \mathbf{C}_Z^{-1}\boldsymbol{\mu}_Z \end{aligned} \tag{41}$$

where the index Z is used to distinguish between the point-wise and the distribution-wise covariances. For example, $\mathbf{C}$ is $n \times n$ and not necessarily invertible while $\mathbf{C}_Z$ is $k \times k$ and invertible. The variance is calculated in a similar way

$$\begin{aligned} v^{Dist}(\mathbf{x}) = \ & \mathbb{E}_Z\left(\mathrm{Var}_\Omega(Y(\mathbf{x})|Y(\mathbf{x}^i) = Z(\mathbf{x}^i)\ ,\ 1 \le i \le k\right) + \\ & \mathrm{Var}_Z\left(\mathbb{E}_\Omega(Y(\mathbf{x})|Y(\mathbf{x}^i) = Z(\mathbf{x}^i)\ ,\ 1 \le i \le k\right) = \\ & \mathbf{c}_Z(\mathbf{x},\mathbf{x}) - \mathbf{c}_Z(\mathbf{x})^\top \mathbf{C}_Z^{-1}\mathbf{c}_Z(\mathbf{x}) \ + \ \mathbf{c}_Z(\mathbf{x})^\top \mathbf{C}_Z^{-1} \underbrace{(\mathrm{Var}_Z\mathbf{Z})}_{\boldsymbol{\Gamma}_Z} \mathbf{C}_Z^{-1}\mathbf{c}_Z(\mathbf{x})\ . \end{aligned} \tag{42}$$

The distribution-wise GP model interpolates the mean and the variance of the distributions at the k locations. At an arbitrary location i, the term $\mathbf{c}_Z(\mathbf{x})^\top \mathbf{C}_Z^{-1}$ that appears in both m^{Dist} and v^{Dist} becomes $\mathbf{e}_i^\top$ because $\mathbf{c}_Z(\mathbf{x}^i)$ is the ith column of $\mathbf{C}_Z$ in this case. As a result

$$m^{Dist}(\mathbf{x}^i) = \mathbf{c}_Z(\mathbf{x}^i)^\top \mathbf{C}_Z^{-1}\boldsymbol{\mu}_Z = \mu_{Z_i} \tag{43}$$

$$\begin{aligned} v^{Dist}(\mathbf{x}^i) = \ & \mathbf{c}_Z(\mathbf{x}^i,\mathbf{x}^i) - \mathbf{c}_Z(\mathbf{x}^i)^\top \mathbf{C}_Z^{-1}\mathbf{c}_Z(\mathbf{x}^i) \ + \\ & \mathbf{c}_Z(\mathbf{x}^i)^\top \mathbf{C}_Z^{-1}\boldsymbol{\Gamma}_Z\mathbf{C}_Z^{-1}\mathbf{c}_Z(\mathbf{x}^i) = \sigma^2_{Z_i}. \end{aligned} \tag{44}$$

In practice, $\boldsymbol{\mu}_Z$ and $\boldsymbol{\Gamma}_Z$ can be approximated by the empirical mean and variance. Suppose repeated points are grouped by sites, e.g., $y_1, \ldots, y_{N_1}$ are the observations at $\mathbf{x}^1$. Recall that the output empirical mean and variance at $\mathbf{x}^i$ are $\overline{y_i}$ and $\overline{s_i^2}$ that we gather in the vector $\overline{\mathbf{y}}$ and the $k \times k$ matrix $\hat{\Gamma}$ whose diagonal elements are $\overline{s_i^2}$'s. Then, the mean

and the variance of the distribution-wise GP are expressed as

$$m^{Dist}(\mathbf{x}) \equiv \mathbf{c}_Z(\mathbf{x})^\top \mathbf{C}_Z^{-1} \overline{\mathbf{y}}, \tag{45}$$

$$v^{Dist}(\mathbf{x}) \equiv \mathbf{c}_Z(\mathbf{x},\mathbf{x}) - \mathbf{c}_Z(\mathbf{x})^\top \mathbf{C}_Z^{-1} \mathbf{c}_Z(\mathbf{x}) + \mathbf{c}_Z(\mathbf{x})^\top \mathbf{C}_Z^{-1} \hat{\Gamma} \mathbf{C}_Z^{-1} \mathbf{c}_Z(\mathbf{x}). \tag{46}$$

As an example, a distribution-wise GP is illustrated in Figure 3.12 where the output empirical mean and variance are used in the model.

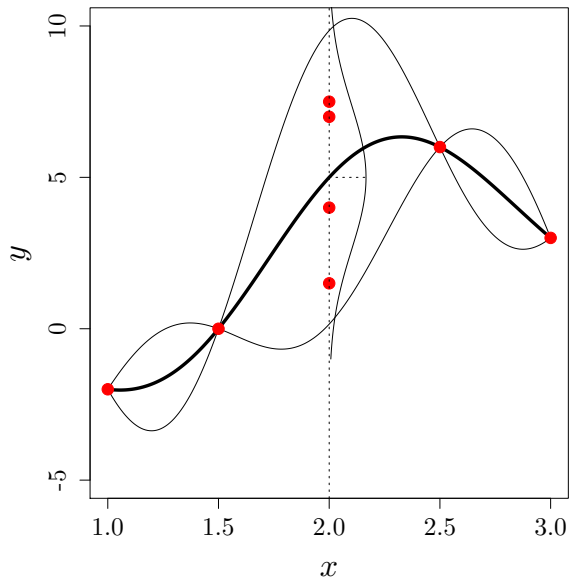


Figure 3.12: Distribution-wise GP, $m^{Dist}(x)$ (thick line) $\pm 2\sqrt{v^{Dist}(x)}$ (thin lines). At the redundant point $x = 2$, the outputs are 1.5, 4, 7 and 7.5. The mean of the distribution-wise GP passes through the average of outputs. Contrarily to PI (cf. Figure 3.2), distribution-wise GP preserves the empirical variance: the kriging variance at $x = 2$ is equal to $s^2_{x=2} = 5.87$.

So far, we have observed that both v^{Dist} and v^{Nug} are non-zero at repeated points. However, there is a fundamental difference between the behaviors of a distribution-wise GP and a GP regularized by nugget; as the number of observations N_i at a redundant point $\mathbf{x}^i$ increases, $v^{Nug}(\mathbf{x}^i)$ tends to 0 while $v^{Dist}(\mathbf{x}^i)$ remains equal to $\sigma^2_{Z_i}$.

This can be analytically seen by assuming that there is only one location site, $\mathbf{x}^1$, with several observations, say n. In this situation, the correlation between every two observations is one and so, the kriging variance regularized by nugget at $\mathbf{x}^1$ is

$$v^{Nug}(\mathbf{x}^1) = \sigma^2 \left(1 - [1,\ldots,1] \left(\mathbf{R} + \tau^2/\sigma^2 \mathbf{I}\right)^{-1} [1,\ldots,1]^\top\right). \tag{47}$$

Here, the correlation matrix $\mathbf{R}$ is a matrix of 1's with only one strictly positive eigenvalue equal to $\lambda_1 = n$, all other eigenvalues being equal to 0. The eigenvector associated to λ_1 is $(1, \ldots, 1)^\top/\sqrt{n}$. Adding nugget will increase all the eigenvalues of $\mathbf{R}$ by τ^2/σ^2.

In Equation (47) one can replace $(\mathbf{R} + \tau^2/\sigma^2\mathbf{I})^{-1}$ by its eigendecomposition that is,

$$\begin{bmatrix} 1/\sqrt{n} & \\ \vdots & \mathbf{W} \\ 1/\sqrt{n} & \end{bmatrix} \begin{bmatrix} \sigma^2/n\sigma^2 + \tau^2 & & & \mathbf{0} \\ & \sigma^2/\tau^2 & & \\ & & \ddots & \\ \mathbf{0} & & & \sigma^2/\tau^2 \end{bmatrix} \begin{bmatrix} 1/\sqrt{n} & \ldots & 1/\sqrt{n} \\ & \mathbf{W}^\top & \end{bmatrix}. \tag{48}$$

This replacement yields

$$v^{Nug}(\mathbf{x}^1) = \frac{\tau^2}{n\sigma^2 + \tau^2}\sigma^2, \tag{49}$$

since $[1, \ldots, 1]$ is perpendicular to any of the other eigenvectors making the columns of $\mathbf{W}$. Consequently, $v^{Nug}(\mathbf{x}^1) \to 0$ when $n \to \infty$. Figure 3.13 further illustrates the difference between distribution-wise and nugget regularization models in GPs. The red bullets are data points generated by sampling from the given distribution of $\mathbf{Z}$'s,

$$\mathbf{Z} \sim \mathcal{N}\left(\begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}, \begin{bmatrix} 0.25 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0.25 \end{bmatrix}\right)$$

and the right plot has more data points at $x = 1$ than the left plot. We observe that the distribution-wise GP model is independent from the number of data points and, in that sense, it "interpolates the distributions": the conditional variance of the distribution-wise GP model does not change with the increase in data points at $x = 1$ while the variance of the GP model regularized by nugget decreases; the mean of the distribution-wise GP is the same on the left and right plots but that of the GP regularized by nugget changes and tends to the mean of the distribution as the number of data points grows.

3.7 Conclusions

This chapter provides a new algebraic comparison of pseudoinverse and nugget regularizations, two classical solutions to overcome the degeneracy of the covariance matrix in Gaussian processes (GPs). We propose a practical strategy when confronted with bad

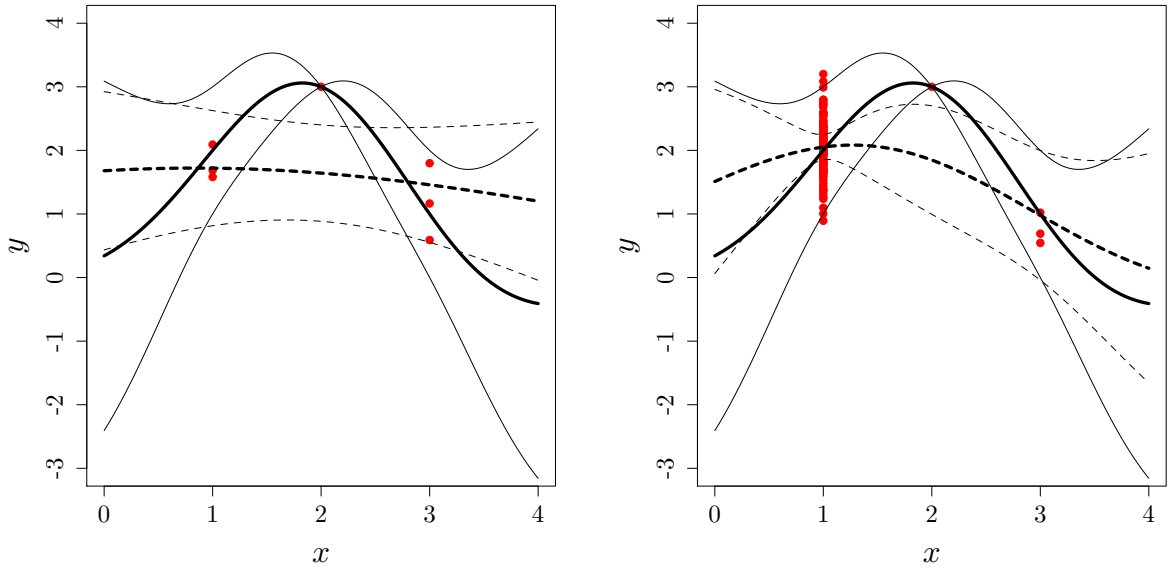


Figure 3.13: Distribution-wise GP (solid lines) versus a GP model regularized by nugget (dashed lines). At $x = 1$, the number of repeated points is 3 (left) and is 100 (right). $v^{Nug}(x = 1)$ (thin dashed lines) shrinks as the number of repeated points increases while $v^{Dist}(x = 1)$ remains constant.

conditioning in GP regression. The analysis focuses on the interpolation properties of GPs when outputs are deterministic. Clear differences between pseudoinverse and nugget regularizations arise by looking at redundant points as a limit case of covariance matrix degeneracy. We have proved that, contrarily to GPs with nugget, GPs with pseudoinverse average the values of outputs and have null variance at redundant points. In GPs regularized by nugget, the discrepancy between model and data translates into a departure of the GP from observation points throughout the domain. In GPs regularized by pseudoinverse, this departure only occurs at redundant points, but the variance is null there.

We have proposed a distribution-wise GP model that interpolates normal distributions instead of data points. This model does not have the drawbacks from both nugget and pseudoinverse regularizations: it not only averages the outputs at redundant points but also preserves the redundant points variances.

Distribution-wise GPs shed a new light on regularization, which starts with the creation of redundant points by clustering. A potential benefit is the reduction in covariance matrix size. Further studying distribution-wise GPs is the main continuation of this work.

Chapter 4

Making EGO and CMA-ES Complementary for Global Optimization

Abstract

The global optimization of expensive-to-calculate continuous functions is of great practical importance in engineering. Among the proposed algorithms for solving such problems, *Efficient Global Optimization (EGO)* and *Covariance Matrix Adaptation Evolution Strategy (CMA-ES)* are regarded as two state-of-the-art unconstrained continuous optimization algorithms. Their underlying principles and performances are different, yet complementary: EGO fills the design space in an order controlled by a Gaussian process (GP) conditioned by the objective function while CMA-ES learns and samples multi-normal laws in the space of design variables and uses it to find a descent direction towards a local minimum. This work proposes a new algorithm, called EGO-CMA, which combines EGO and CMA-ES. In EGO-CMA, the EGO search is interrupted early and followed by a CMA-ES search whose starting point, initial step size and covariance matrix are calculated from the already sampled points and the associated conditional GP. EGO-CMA improves the performance of both EGO and CMA-ES in our experiments.

4.1 Introduction

One approach to deal with expensive and multimodal optimization problems is to use GP as (meta)models for the objective function. For example, EGO algorithm has become

a standard for continuous global optimization in less than twenty dimensions when the number of function evaluations is inferior to 1000.

Another popular algorithm in continuous global optimization is the stochastic Covariance Matrix Adaptation Evolution Strategy (CMA-ES, [HO01]). CMA-ES is interpreted as a robust local search method in [HO96]. Its robustness is attributed to invariance properties with respect to objective function scaling and coordinate system rotations. This algorithm was consistently found to be highly performing in the BBOB contests for low, moderate, and highly multimodal functions for problems dimensions between 5 and 40 [HAR+10] if it is coupled with a restart mechanism. In [AH05, Han09a], restart strategies are proposed to prevent premature convergence of CMA-ES to local optima.

A comparison of how EGO and CMA-ES search a design space shows fundamental differences: while EGO is a deterministic space-filling strategy, CMA-ES can be seen as a converging[1] stochastic algorithm. Such a difference in principles, i.e., being space-filling for EGO and converging for CMA-ES, makes them complementary. In this chapter, we propose to start a global optimization with EGO and rapidly switch to CMA-ES for a robust local convergence. The cooperation between the two algorithms goes beyond a plain succession as the Gaussian process learned by EGO allows improving the initial value of CMA-ES parameters such as the starting point and the covariance matrix.

Past works on global optimization of costly functions have already involved augmenting Evolution Strategies (ESs) with metamodels [Jin11, KHK06, LSS13]. The general idea is to replace some evaluations of the true objective function with metamodel estimates and trigger true evaluations through an error rate measure. In [KHK06], CMA-ES has been coupled with a local regression metamodel, making the lmm-CMA algorithm, where the metamodel allows savings in the ranking of the candidate solutions. References [LSS12, LSS13] present the s*ACM-ES (surrogate Assisted Covariance Matrix adaptation Evolution Strategy), an algorithm with a ranking support vector machine as metamodel and where the number of iterations (generations) done with the metamodel depend on its error rate.

Kriging has sometimes been the metamodel added to the ESs. The motivation for using kriging is the availability of a prediction uncertainty. In [USZ03], a pre-selection of the most

[1] by "converging", we mean that the CMA-ES algorithm, in finite time, will devote most of its evaluations for fine tuning the location of the current best point, as exemplified in Section 4.3.2.

promising points is done based on a kriging model, which enables sampling more solutions and makes the search more efficient. Two criteria are investigated as performance measures, the (mean) objective function prediction and the probability of improvement over the best observed point. In [BSK04], kriging serves as a local metamodel and various performances are measured by different compromises between search intensification around the current best solution and exploration. In [KEDB10], a local kriging enables dealing with noisy objective functions by easing the estimation of the objective function expectation.

The optimization algorithm introduced in this chapter is based on a new idea: using first EGO for exploration and then CMA-ES from the best point obtained by EGO for final convergence. The motivation is that EGO is efficient in the early design of experiments (DoE) stage of the optimization (volume search), while CMA-ES is a converging search process that efficiently switches from volume to local search.

4.2 The CMA-ES Algorithm

First introduced by Hansen, Ostermeier, and Gawelczyk [HOG95], CMA-ES adapts a complete covariance matrix of multivariate normal distribution. It is considered as the state-of-the-art algorithm for unconstrained continuous numerical black-box optimization if sufficient budget is afforded. It efficiently optimizes unimodal functions and has superior performance on ill-conditioned and non-separable functions [HK04].

CMA-ES is an iterative stochastic optimization algorithm such that in each iteration a population of individuals (search points) are generated, according to a multivariate normal distribution. Then, some individuals are selected to become the parents in the next iteration based on their objective function value. This process allows individuals with better function values are generated over the course of optimization. Let $\mathbf{m}^{(g)}$ be the mean vector of the multivariate normal distribution in generation g. The ith individual denoted by $\mathbf{x}_i^{(g+1)}$ is generated according to:

$$\mathbf{x}_i^{(g+1)} \sim \mathcal{N}\left(\mathbf{m}^{(g)}, \left(\sigma^{(g)}\right)^2 \mathbf{C}^{(g)}\right) = \mathbf{m}^{(g)} + \sigma^{(g)}\mathcal{N}\left(\mathbf{0}, \mathbf{C}^{(g)}\right), \ for\ i = 1, ..., \lambda, \tag{1}$$

where $\sigma^{(g)} \in \mathbb{R}^+$ is called mutation step size and $\mathbf{C}^{(g)} \in \mathbb{R}^{d\times d}$ is a covariance matrix and d is the number of variables. The former controls the step length and the later governs the shape of the distribution ellipsoid.

We denote the i-th best search point by $\mathbf{x}_{i:\lambda}^{(g+1)}$. The mean of the next generation is obtained from $\mathbf{x}_{1:\lambda}^{(g+1)}, ..., \mathbf{x}_{\lambda:\lambda}^{(g)}$ as follows:

$$\mathbf{m}^{(g+1)} = \sum_{i=1}^{\mu} \omega_i \mathbf{x}_{i:\lambda}^{(g+1)} = \mathbf{m}^{(g)} + \sigma^{(g)} \sum_{i=1}^{\mu} \omega_i \mathbf{y}_{i:\lambda} \tag{2}$$

$$\sum_{i=1}^{\mu} \omega_i = 1, \qquad \omega_1 \geqslant \omega_2 ... \geqslant \omega_\mu > 0, \tag{3}$$

where $\mathbf{y}_{i:\lambda} = \frac{\left(\mathbf{x}_{i:\lambda}^{(g+1)} - \mathbf{m}^{(g)}\right)}{\sigma^{(g)}}$ and the weights ω_i are strictly positive and normalized. This update moves the mean vector towards the best solutions.

As we observe, the update of the mean vector is done by μ best individuals that are selected based on their function value. That the selection is only based on the fitness ranking makes the algorithm invariant to any monotonous transformation of the objective function. Furthermore, CMA-ES is invariant to angle preserving transformation of search space i.e., rotation, reflection and transformation. Invariance is a favorable property because it implies identical performance of the search algorithm on equivalence classes of objective functions [HK04].

Usually, the weights are assigned in such a way that $\mu_{eff} \approx \lambda/4$ in which the measure μ_{eff} denotes the *variance effective selection mass*. μ_{eff} is defined as $\mu_{eff} = \left(\sum_{i=1}^{\mu} \omega_i^2\right)^{-1}$ and $\mu_{eff} = \mu$ if $\omega_i = 1/\mu$. This measure is frequently used to calibrate and tune parameters in the algorithm.

The adaptation of the covariance matrix $\mathbf{C}^{(g)}$ and the step size $\sigma^{(g)}$ uses the notion of "evolution path", denoted by $\mathbf{p}_c^{(g)}$ and $\mathbf{p}_\sigma^{(g)}$ respectively. The evolution path expresses the correlation between consecutive steps and stores information of the previous updates, see [HO01] for more information. The update of $\mathbf{p}_c^{(g)}$, $\mathbf{p}_\sigma^{(g)}$, the covariance matrix and the step size is given by

$$\mathbf{p}_c^{(g+1)} = (1 - c_c)\mathbf{p}_c^{(g)} + \sqrt{c_c(2 - c_c)\mu_{eff}} \frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}}, \tag{4}$$

$$\mathbf{p}_\sigma^{(g+1)} = (1 - c_\sigma)\mathbf{p}_\sigma^{(g)} + \sqrt{c_\sigma(2 - c_\sigma)\mu_{eff}} \mathbf{C}^{(g)^{-\frac{1}{2}}} \frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}}, \tag{5}$$

$$\begin{aligned} \mathbf{C}^{(g+1)} = (1 - c_{cov})\mathbf{C}^{(g)} &+ \frac{c_{cov}}{\mu_{cov}} \mathbf{p}_c^{(g+1)} \mathbf{p}_c^{(g+1)^T} \\ &+ c_{cov} \left(1 - \frac{1}{\mu_{cov}}\right) \sum_{i=1}^{\mu} \omega_i \mathbf{y}_{i:\lambda} \mathbf{y}_{i:\lambda}^T, \end{aligned} \tag{6}$$

$$\sigma^{(g+1)} = \sigma^{(g)} \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|\mathbf{p}_\sigma^{(g+1)}\|}{\mathrm{E}\|\mathcal{N}(\mathbf{0},\ \mathbf{I})\|}\right)\right), \tag{7}$$

where c_c, c_σ, c_{cov}, c_σ and d_σ are the parameters of the algorithm. The default values of the parameters can be found in [HK04].

The initialized covariance matrix is the identity matrix, $\mathbf{C}^{(0)} = \mathbf{I}$. The initial values of the evolution paths are: $\mathbf{p}_\sigma^{(g)} = \mathbf{p}_c^{(g)} = \mathbf{0}$. Notice that $\mathbf{x}^{(0)}$ and $\sigma^{(0)}$ are problem dependent. For example, too small initial step size should be avoided in the optimization of multimodal functions.

Default parameter values of λ and μ and the weights are

$$\lambda = 4 + \lfloor 3\ln(d) \rfloor, \qquad \mu = \lfloor \frac{\lambda}{2} \rfloor, \tag{8}$$

$$\omega_i = \frac{\ln(\mu+1) - ln(i)}{\mu\ln(\mu+1) - ln(\mu!\,)}. \tag{9}$$

We end up this section by giving a summary of CMA-ES algorithm.

Algorithm 4.1 Covariance Matrix Adaptation Evolution Strategy (CMA-ES)

Initialize the distribution parameters: $\mathbf{m}^{(0)}, \mathbf{C}^{(0)}, \sigma^{(0)}$.

Set parameters λ and μ to their default values.

while not stop **do**

Generate new population sampled from multivariate normal distribution:
$\mathbf{x}_i^{(g+1)} = \mathcal{N}\left(\mathbf{m}^{(g)}, \left(\sigma^{(g)}\right)^2 \mathbf{C}^{(g)}\right) = \mathbf{m}^{(g)} + \sigma^{(g)}\mathcal{N}\left(\mathbf{0}, \mathbf{C}^{(g)}\right),\ for\ i = 1, ..., \lambda.$

Update the mean value $\mathbf{m}^{(g)}$, the step size $\sigma^{(g)}$ and the covariance matrix $\mathbf{C}^{(g)}$.

end while

4.3 The EGO-CMA Algorithm

4.3.1 Experimental Setup and initial observations

The optimization algorithms compared in this work are EGO, CMA-ES, and (later) EGO-CMA. They are tested on three well-known functions called Sphere, Ackley, and Rastrigin. These functions are defined in Table 4.1. The Sphere function is unimodal, separable and differentiable function. This function is used to observe the pure convergence speed of the algorithms. The Ackley function has many local minima with a large hole at the center

which is the location of the global minimum. The Rastrigin function is highly multimodal, but locations of the minima are regularly distributed. In the optimization procedure, the

Table 4.1: Test functions

Function	Expression	Defined region
Sphere	$f(\mathbf{x}) = \sum_{i=1}^{d} (x_i)^2$	[-5.12, 5.12]
Ackley	$f(\mathbf{x}) = -a \exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d} \cos(cx_i)\right)$	[-32.768, 32.768]
	$+ a - \exp(1), a = 20, b = 0.2, c = 2\pi$	
Rastrigin	$f(\mathbf{x}) = 10d + \sum_{i=1}^{d} [x_i^2 - 10\cos(2\pi x_i)]$	[-5.12, 5.12]

search spaces of the functions have been rescaled to $[-5, 5]^d$, $d = 2, 5, 10$. All the test functions have one global minimum located at $(2.5, \ldots, 2.5)_{1\times d}$. The total number of calls to the objective function or *budget* is $70 \times d$. The initial design points of EGO are obtained by Latin Hypercube Samples (LHS) of size $3 \times d$.

We repeat EGO three times on each function. CMA-ES being a stochastic optimizer, it arguably exhibits larger performance variation so its runs are repeated ten times from three different starting points. For running EGO and CMA-ES, we use the R packages *DiceOptim*[2] and *cmaes*[3] with their default parameter values.

Figure 4.1 illustrates one typical run of EGO and CMA-ES on the Sphere function in dimension 5. The solid line represents each function value obtained by the optimization algorithm and the dashed-dotted line shows the best observed function value so far. In the left picture, EGO makes early progress and then tries to explore the rest of the search space. Here, the exploration is unfruitful because Sphere function is unimodal and the global minimum has been already detected. While CMA-ES, right picture, steadily converges to the minimum as the number of calls to the objective function increases. Such an observation was confirmed on the other test functions and started the idea of combining EGO for the early exploration phase and CMA-ES for the final converging phase.

[2]https://cran.r-project.org/web/packages/DiceOptim/index.html
[3]https://cran.r-project.org/web/packages/cmaes/index.html

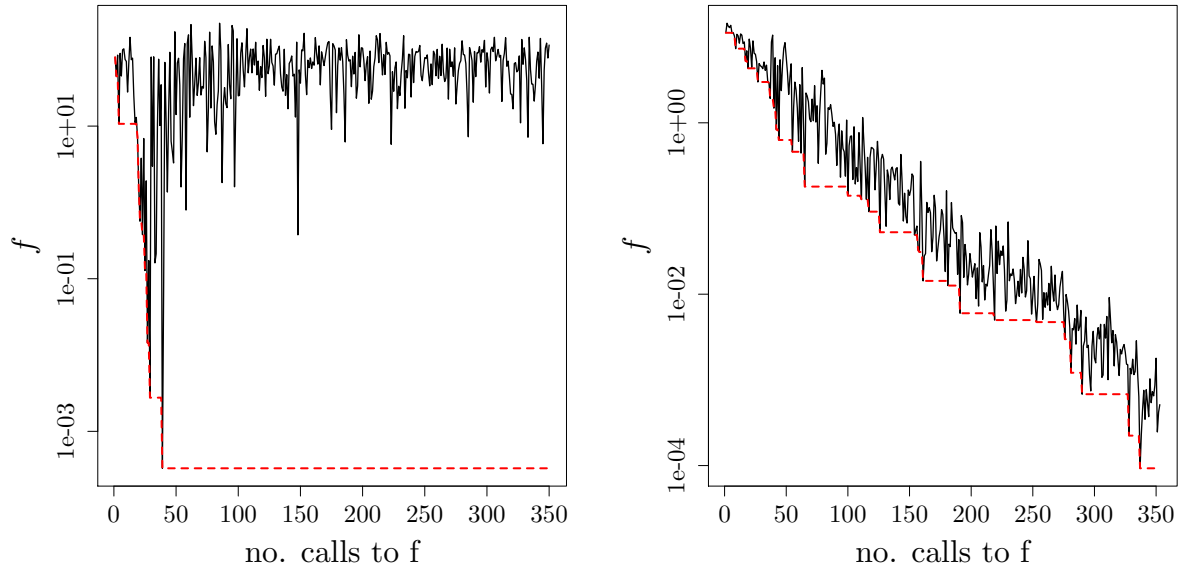


Figure 4.1: One typical run of EGO (left) and CMA-ES (right) on the Sphere function, $d = 5$. Solid line: f history during optimization. Dashed line: best f.

4.3.2 Comparing EGO and CMA-ES

To compare EGO and CMA-ES the median of the best function value obtained by each algorithm is calculated. In addition, we consider three different starting points for CMA-ES. The results of this comparison in dimension 5 and 10 are illustrated in Figure 4.2.

The analysis of convergence curves of EGO and CMA-ES reveals that EGO is quick at the beginning. But after some iterations, EGO loses its efficiency. Moreover, it does not converge to the global optimum. On the other side, CMA-ES shows a monotone improvement, as we see this phenomenon in higher dimension with larger budget. Here we use a $2D$ example to better understand the search principle of EGO and CMA-ES. Figure 4.3 demonstrates the search points obtained by each algorithm in the optimization of Ackley function in dimension 2. EGO is a space-filling algorithm; i.e., it tries to find the global optimum by filling the holes in the search space. This space-filling characteristic resulted from the expected improvement criterion. While the search points in CMA-ES tend to converge the optimum and not filling the space.

To investigate the characteristics of the two algorithms in higher dimensions, we use a criterion called discrepancy. This criterion measures how far a given distribution of points

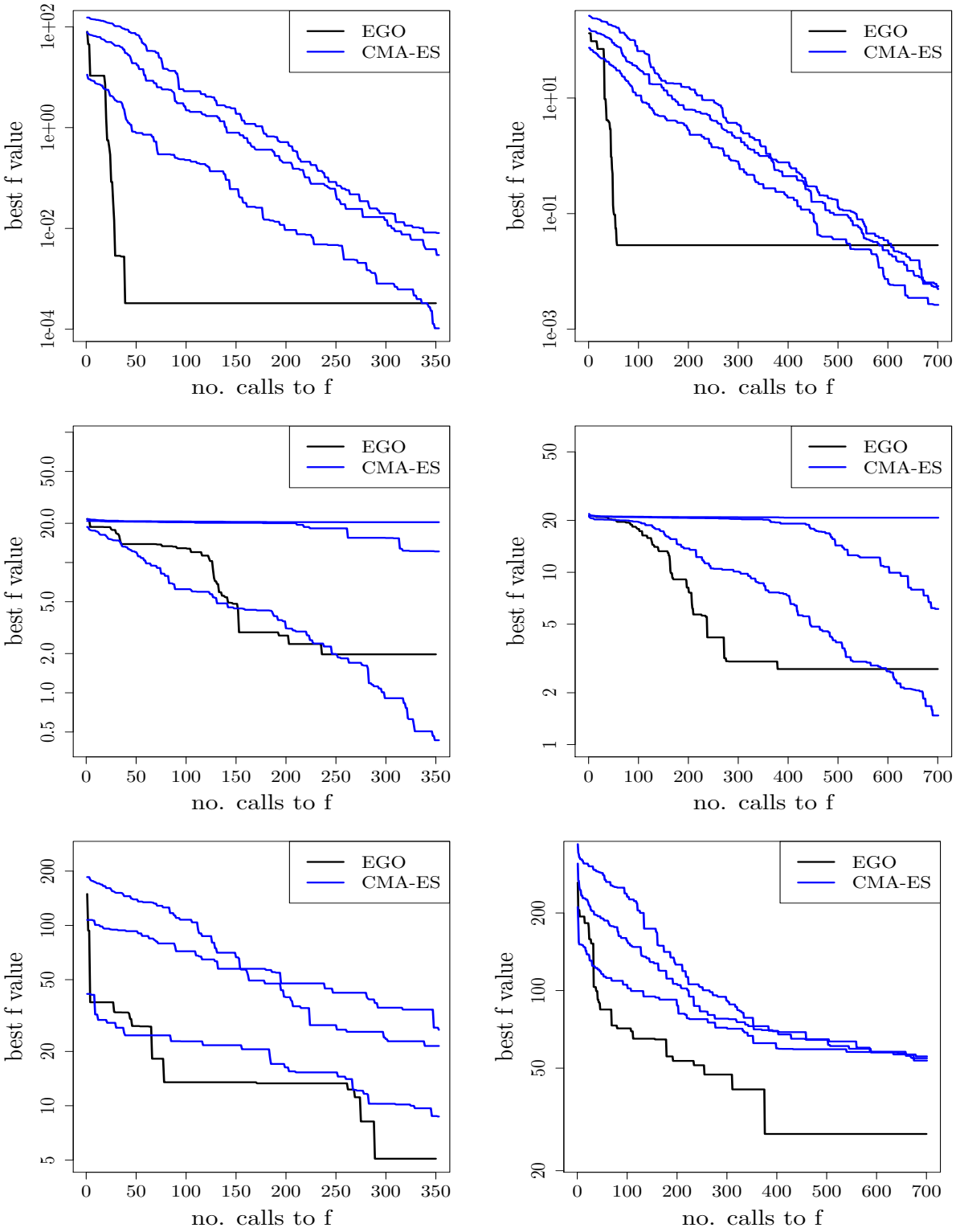


Figure 4.2: Median of the best objective function vs. number of calls of EGO and CMA-ES (with three different starting points) in dimensions 5 (left) and 10 (right) on functions: Sphere (first row), Ackley (second row), and Rastrigin (third row). Generally, EGO makes early progress and then loses efficiency while CMA-ES steadily converges to the optimum.

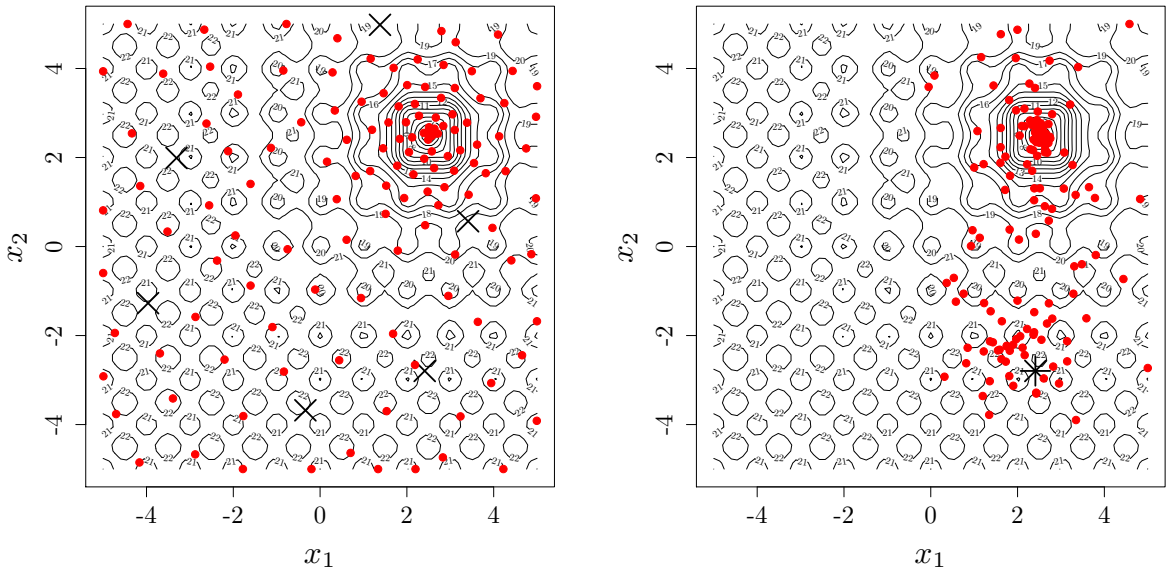


Figure 4.3: Illustration of the search points obtained by EGO (left) and CMA-ES (right) in the optimization of Ackley function. The bullets are the points generated by the optimization algorithms. The crosses in the leftmost picture are the initial DoE for EGO. The asterisk in the rightmost picture is the starting point of CMA-ES. EGO is space-filling while the search points in CMA-ES tend to converge the optimum.

deviates from a perfectly uniform one [DHF15]. Let $|S|$ denotes the number of points in a set, S, then the discrepancy of the design matrix $\mathbf{X}$ is given by [DK10]:

$$D(\mathbf{X}) = \left\| \frac{|\mathbf{X} \cap c^d|}{n} - Vol\left(c^d\right) \right\| \tag{10}$$

where $\|.\|$ represents an appropriate norm over all d dimensional rectangular subsets, c^d, of the unit hypercube $[0,1]^d$.

A small value of $D(\mathbf{X})$ means that the design $\mathbf{X}$ is close to a uniform design. If EGO and CMA-ES are compared based on the discrepancy criterion, the discrepancy of the points obtained by EGO is less than CMA-ES. The reason is that while EGO tends to fill the space, CMA-ES tries to converge the (optimum) point. For example, the discrepancy of the two algorithm has been calculated on Ackley function in dimensions 5. In this example, the discrepancy of EGO and CMA-ES are about 0.002 and 0.12, respectively.

4.3.3 Comparing EGO and CMA-ES using COCO

Here, we further investigate the performance of EGO and CMA-ES by using Comparing Continuous Optimisers (COCO) [HAFR09] methodology. The numerical experiments are carried out on 24 noise-free real-parameter test functions [HFRA09]. These test functions have properties such as multimodality, non-convexity, ill-conditioning and non separability which are related to real-world problems. All functions are defined in $[-5,5]^d$ and have their global optimum in $[-4,4]^d$. For each function and each dimension d, 15 trials are performed on 15 different function instances (a function with different optimal value).

An *optimization problem* is defined from a function instance and a target function value. Let f_{opt} be the optimal function value and Δf be the precision to reach. Then, the target function value is defined as: $f_{target} = f_{opt} + \Delta f$. Solving a problem (i.e., a successful trial) means finding a solution whose function evaluation is below the target value. Note that the algorithm can also be restarted. The number of evaluations needed to solve a problem is called *runtime*. In the COCO framework, the Expected Running Time (ERT), which is the expected number of function evaluations to reach a target value for the first time, is used to measure the performance of an algorithm.

The COCO results are presented using the bootstrapped empirical cumulative distribution of ERT divided by the problem dimension, also known as the Empirical Cumulative Distribution Function (ECDF). In the bootstrapping process, for each target, 100 instances

of ERT are generated. Each ERT instance is calculated by repeatedly drawing single trials with replacement, from 15 algorithm runs, until obtaining a successful trial [PH12]. We refer to [HAFR09] for more information.

Figures 4.4 and 4.5 show the ECDF plots of EGO, CMA-ES and random search (denoted by RandSearch) in dimensions 3 and 5, respectively. The budget, indicated by a cross on each curve, is $70 \times d$ for EGO and $500 \times d$ for CMA-ES and random search. The results are illustrated based on the function groups which are:

1. separable functions $f_1 - f_5$,
2. unimodal functions with moderate conditioning $f_6 - f_9$,
3. unimodal ill-conditioned functions $f_{10} - f_{14}$,
4. multimodal functions $f_{15} - f_{19}$,
5. multimodal functions with weak structure $f_{20} - f_{24}$.

It can be seen that EGO is able to solve more problems than CMA-ES at the beginning of the search. However, the performance of CMA-ES constantly improves and the slope of its empirical cumulative distribution curve is often steeper. Both algorithms have similar performance on separable functions ($f_1 - f_5$). CMA-ES outperforms EGO on moderate conditioning and ill-conditioned functions ($f_6 - f_9$ and $f_{10} - f_{14}$). But the performance of EGO is better than CMA-ES on multimodal functions with weak structure ($f_{20} - f_{24}$).

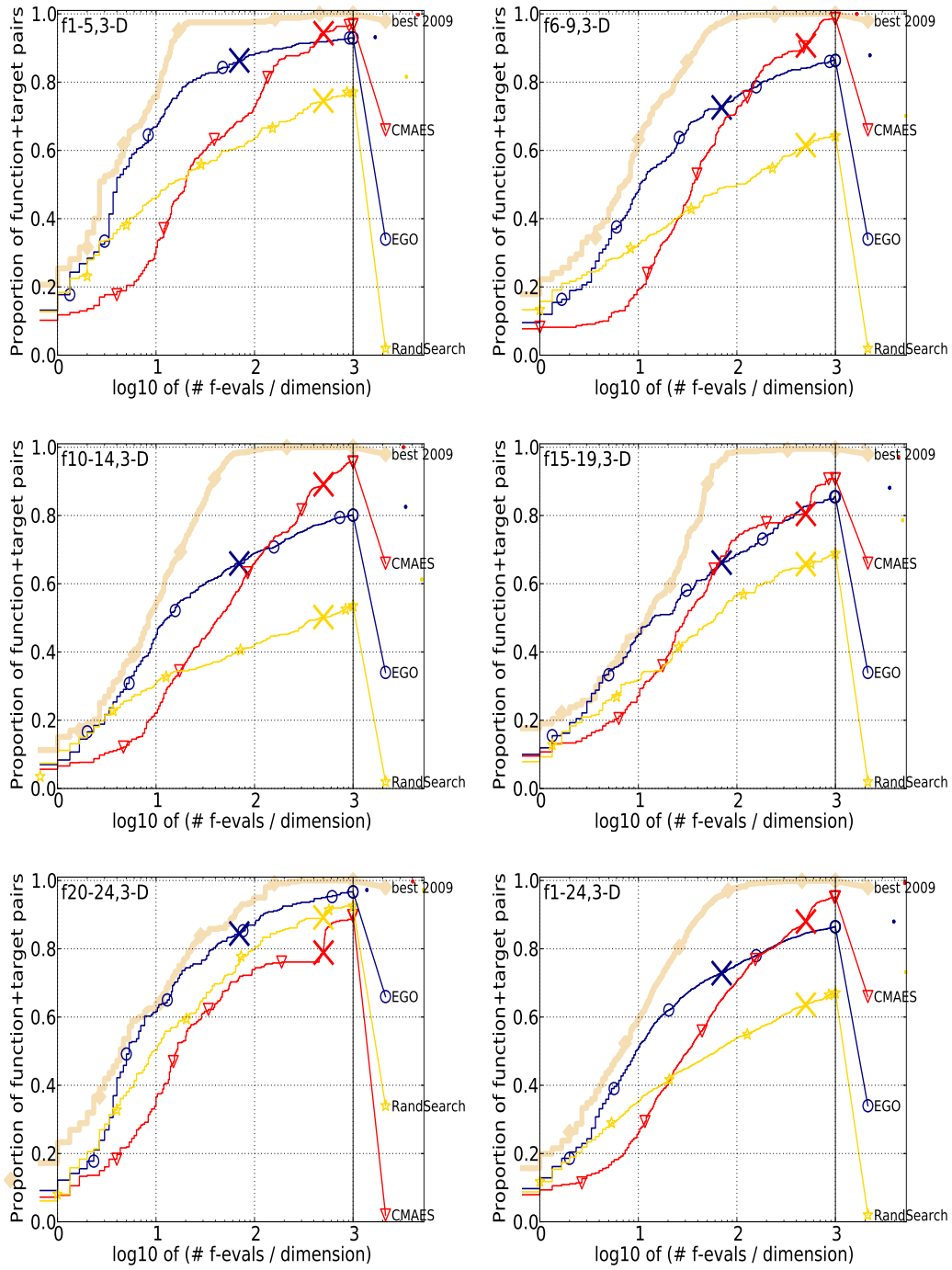


Figure 4.4: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension for all functions and subgroups in $3D$. The targets are chosen from $10^{[-8..2]}$ such that the bestGECCO2009 artificial algorithm just not reached them within a given budget of $k \times d$, with $k \in 0.5, 1.2, 3, 10, 50$. The "best 2009" line corresponds to the best ERT observed during BBOB 2009 for each selected target.

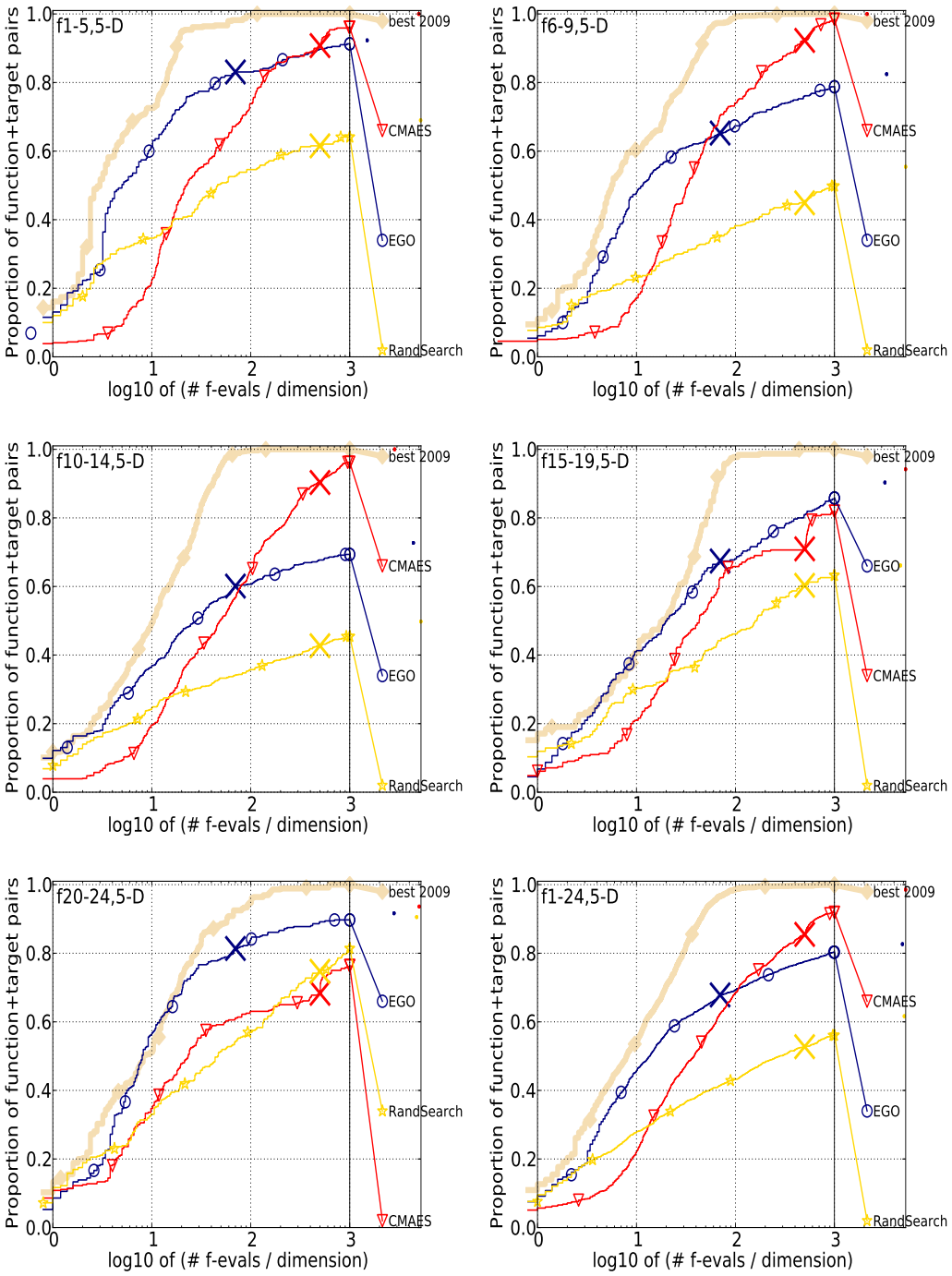


Figure 4.5: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension for for all functions and subgroups in $5D$. See caption of Figure 4.4 for more details.

4.3.4 Combining EGO and CMA-ES

We now introduce the EGO-CMA algorithm, which first explores the search space with EGO and then switches to CMA-ES in order to converge to the optimum.
The switch occurs after the best observed f has not improved for at least $0.1 \times budget$ analyses and if one of the following conditions is met (based on several observations):

i) 50 percent of the *budget* is exhausted or

ii) $\overline{EI} < 0.01 \times \left(f_{DoE}^{best} - f^{best}\right)$.

$\overline{EI}$ is the average of the maximum expected improvement over the 5 last iterations. f_{DoE}^{best} and f^{best} are the best f values in the initial design of experiments and the current best point, respectively. When the switch takes place, the best point obtained by EGO, $\mathbf{x}^{best}$, becomes CMA-ES's starting point. Furthermore, EGO-CMA uses of the fitted kriging mean as an approximation to the true function to warm start CMA-ES.

Let us provide some background on CMA-ES initialization. Consider first the optimization of a convex-quadratic function $f_{\mathbf{H}}(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}_{\mathbf{H}}^*)^\top \mathbf{H}(\mathbf{x} - \mathbf{x}_{\mathbf{H}}^*)$, where $\mathbf{H}$ is positive definite and $\mathbf{x}_{\mathbf{H}}^*$ is the optimum. $\mathbf{H}$ can be decomposed into $\mathbf{H} = \mathbf{B}\mathbf{D}^2\mathbf{B}^\top$, where $\mathbf{B}$ is made of the eigenvectors of $\mathbf{H}$ as columns ($\mathbf{B}^\top\mathbf{B} = \mathbf{B}\mathbf{B}^\top = \mathbf{I}$) and $\mathbf{D}$ is a diagonal matrix with the square roots of $\mathbf{H}$'s eigenvalues as diagonal elements. The optimal ES covariance matrix has lines of equiprobable mutation aligned with the level sets of the objective function [Rud92]. This happens when the covariance matrix of the search distribution, $\mathbf{C}$ (from (1) without superscript), is proportional to the inverse of $\mathbf{H}$ and so we set

$$\mathbf{C} = \mathbf{B}\mathbf{D}^{-2}\mathbf{B}^\top \ . \tag{11}$$

The step size σ can now be tuned by performing a change of variable to turn to a spherical landscape : define the new variable $\mathbf{t} = \mathbf{D}\mathbf{B}^\top(\mathbf{x} - \mathbf{x}_{\mathbf{H}}^*)$, the objective function becomes $f_{\mathbf{H}}(\mathbf{t}) = \frac{1}{2}\mathbf{t}^\top\mathbf{t}$. In the t-space, the CMA-ES search points distribution (1) becomes $\mathbf{t} \sim \mathbf{D}\mathbf{B}^\top(\mathbf{m} - \mathbf{x}_{\mathbf{H}}^*) + \sigma\mathcal{N}(\mathbf{0}, \mathbf{I})$. In terms of t, one optimizes a spherical function with a spherical distribution, a situation in which one would like that the average step length (the expectation of the square root of a χ_d^2 random variable times σ) equals the distance to the optimum

$$\sigma\sqrt{d - 0.5} \ = \ \left\|\mathbf{D}\mathbf{B}^\top(\mathbf{m} - \mathbf{x}_{\mathbf{H}}^*)\right\| \quad \Rightarrow \quad \sigma \ = \ \frac{\left\|\mathbf{D}\mathbf{B}^\top(\mathbf{m} - \mathbf{x}_{\mathbf{H}}^*)\right\|}{\sqrt{d - 0.5}} \ . \tag{12}$$

We can now return to the EGO-CMA description. EGO is stopped and CMA-ES is started at $\mathbf{m}^{(0)} = \mathbf{x}^{best}$. To obtain $\sigma^{(0)}$ and $\mathbf{C}^{(0)}$ from the above quadratic considerations, we take the second order Taylor expansion of the kriging mean (an approximation to the objective function) at point $\mathbf{x}^{best}$:

$$f(\mathbf{x}) \approx f_{\mathbf{H}}(\mathbf{x}) = m(\mathbf{x}^{best}) + \nabla m(\mathbf{x}^{best})^\top(\mathbf{x} - \mathbf{x}^{best}) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^{best})\mathbf{H}(\mathbf{x} - \mathbf{x}^{best}) \ .$$

The initial covariance of CMA-ES is set equal to the inverse of the Hessian of the kriging mean at $\mathbf{x}^{best}$,

$$\mathbf{C}^{(0)} \ = \ \mathbf{H}^{-1} \ . \tag{13}$$

Cases when $\mathbf{H}$ is not strictly positive definite, among which the non invertibility case, are discussed later. Minimization of $f_{\mathbf{H}}$ gives $\mathbf{x}^*_{\mathbf{H}}$, an approximation to the optimum, by which we can complete Equation (12) and calculate $\sigma^{(0)}$:

$$\begin{aligned} \mathbf{x}^*_{\mathbf{H}} - \mathbf{x}^{best} &= -\mathbf{H}^{-1}(\mathbf{x}^{best})\nabla m(\mathbf{x}^{best}) \\ \Rightarrow \quad \sigma^{(0)} &= \frac{\left\|\mathbf{D}\mathbf{B}^\top\mathbf{H}^{-1}(\mathbf{x}^{best})\nabla m(\mathbf{x}^{best})\right\|}{\sqrt{d - 0.5}} \ . \end{aligned} \tag{14}$$

We now discuss the cases when the Hessian matrix is not strictly positive definite, i.e., $f_{\mathbf{H}}$ is concave in some directions. $f_{\mathbf{H}}$ is convexified, i.e., the Hessian is forced to be positive definite, by substituting 10^{-6} for the negative eigenvalues in $\mathbf{D}^2$. However, this might increase the condition number of the Hessian matrix that is the ratio of the largest to the smallest eigenvalue, $cond(\mathbf{H}) = \frac{\lambda_{max}}{\lambda_{min}}$. To improve the condition number, we add a positive value, τ^2, to the main diagonal of the Hessian matrix, $\mathbf{H}_{\text{conv}} = \mathbf{B}\mathbf{D}^2_{\text{conv}}\mathbf{B}^\top = \mathbf{B}(\mathbf{D}^2 + \tau^2\mathbf{I})\mathbf{B}^\top$. τ^2 can be calculated by defining an upper bound on the condition number, $\mathcal{CU}$,

$$\frac{\lambda_{max} + \tau^2}{\lambda_{min} + \tau^2} \ \leq \ \mathcal{CU} \quad \Rightarrow \quad \tau^2 \ \geq \ \frac{\mathcal{CU}\lambda_{min} - \lambda_{max}}{1 - \mathcal{CU}} \ . \tag{15}$$

In our experiments, we set the condition number limit $\mathcal{CU}$ equal to 10^4 and the initial CMA-ES covariance and step size (equations (13) and (14)) are calculated with $\mathbf{H}_{\text{conv}}$ and $\mathbf{D}_{\text{conv}}$. Finally, the step size is bounded through

$$\frac{0.3\ 10^{-8}}{\sqrt{d}} \times \|\mathbf{D}_{\text{conv}}\mathbf{B}^\top(\mathbf{u} - \mathbf{l})\| \leq \sigma^{(0)} \leq \frac{0.3}{\sqrt{d}} \times \|\mathbf{D}_{\text{conv}}\mathbf{B}^\top(\mathbf{u} - \mathbf{l})\| \ . \tag{16}$$

4.4 Simulation Results

The performance of EGO-CMA is tested by repeating each run of EGO-CMA 5 times on each test function, see Figure 4.6. In the figure, the time (number of calls) that the algorithm switches from EGO to CMA-ES is indicated by a cross. For the Sphere function, EGO quickly detects the basin of attraction of the global minimum which allows EGO-CMA to further increase the accuracy. However, with the more multimodal Ackley function, the switch occurs at more diverse times of the search.

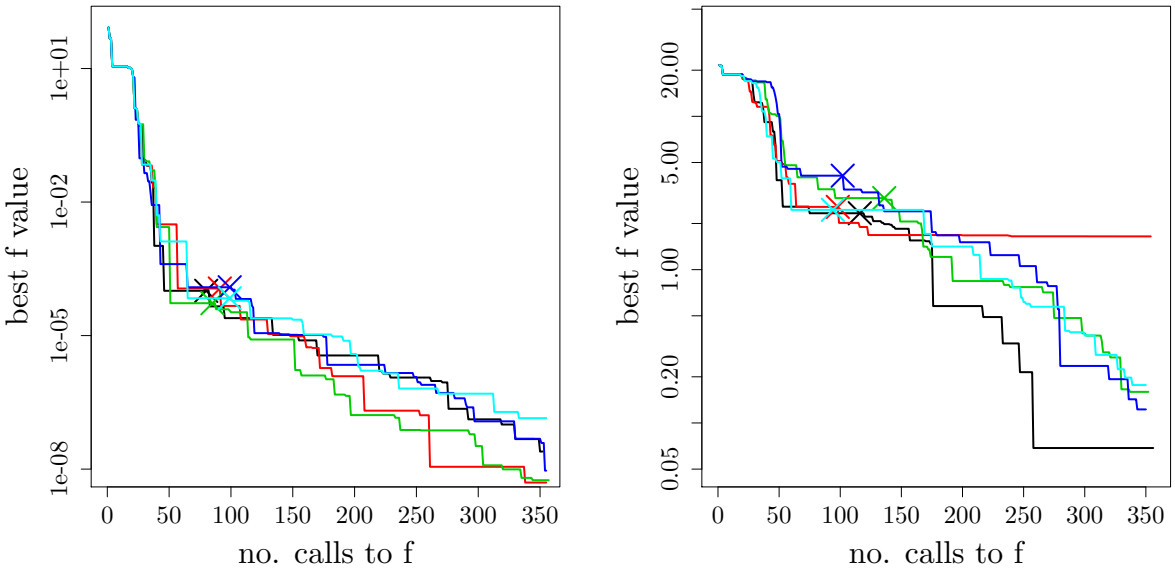


Figure 4.6: 5 runs of EGO-CMA on Sphere (left) and Ackley function (right) in dimension 5. The crosses show the time (number of calls) that the algorithm switchs from EGO to CMA-ES.

Finally the performance of EGO-CMA is compared to EGO and CMA-ES. The comparison in dimensions 5 and 10 is shown in Figure 4.7. On average, we observe a better performance of EGO-CMA over EGO and CMA-ES. For example, the accuracy of EGO-CMA is about 10^{-8} for the Sphere function with a gain of two orders of magnitude over CMA-ES. The switch from EGO to CMA-ES in EGO-CMA can clearly be seen on the Sphere function before 100 function evaluations as the EGO-CMA curve first follows EGO and then is parallel to CMA-ES.

The question is that whether in the EGO-CMA algorithm starting CMA-ES with the

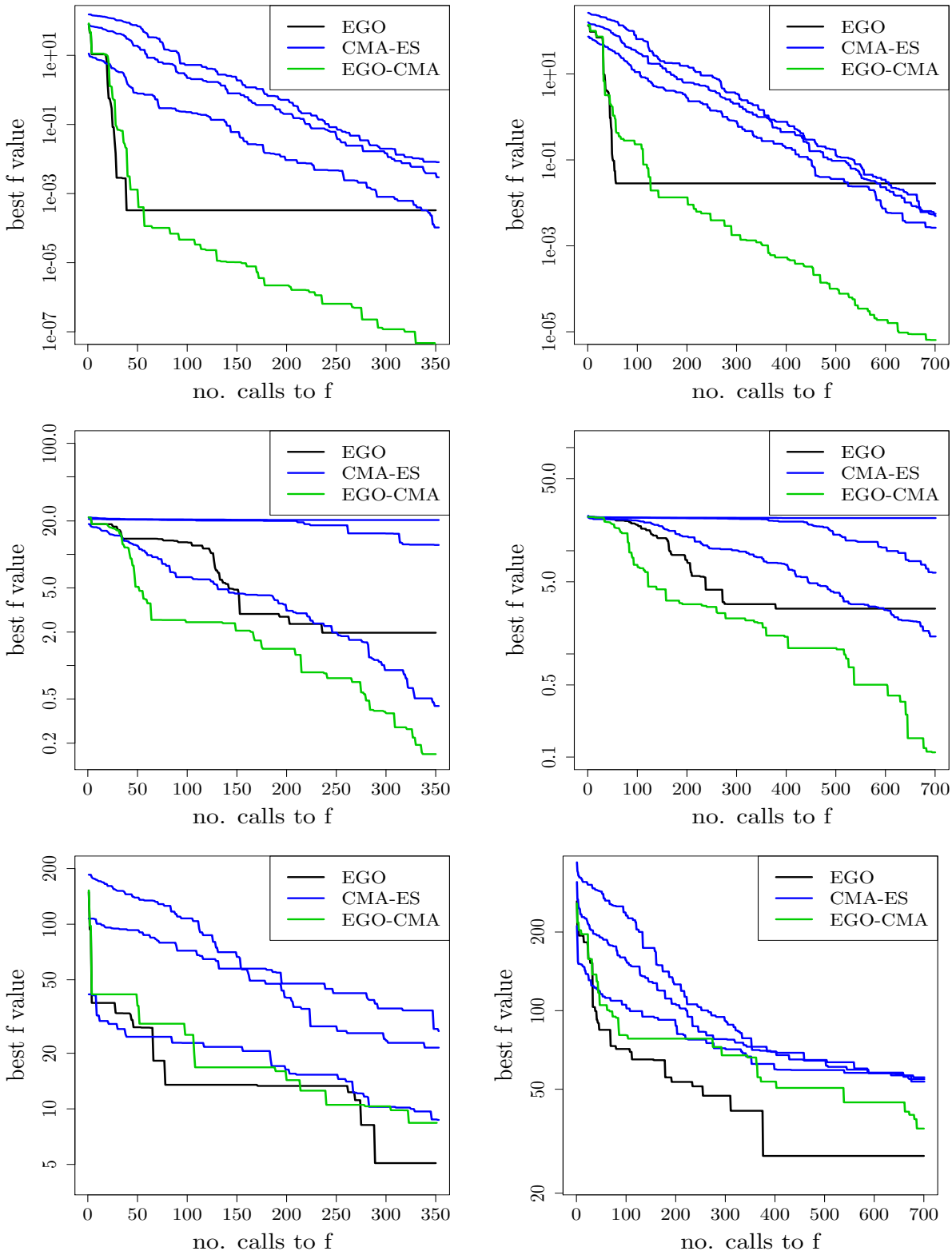


Figure 4.7: Median of the best objective function vs. number of calls of EGO, CMA-ES (with three different starting points) and EGO-CMA in dimensions 5 (left) and 10 (right) on functions: Sphere (first row), Ackley (second row), and Rastrigin (third row).

inverse of Hessian of kriging mean at $\mathbf{x}^{best}$ as the initial covariance matrix for CMA-ES is helpful. To answer the question, we perform two experiments in which a Quadratic function with the condition number of 10^3 is optimized by EGO-CMA. In the first experiment the initial covariance matrix of CMA-ES is $\mathbf{H}^{-1}$ and in the second one is the identity matrix, $\mathbf{I}$. The runs are repeated 5 times and the median of them are compared, see Figure 4.8. It is seen that using $\mathbf{H}^{-1}$ significantly improves the algorithm's performance.

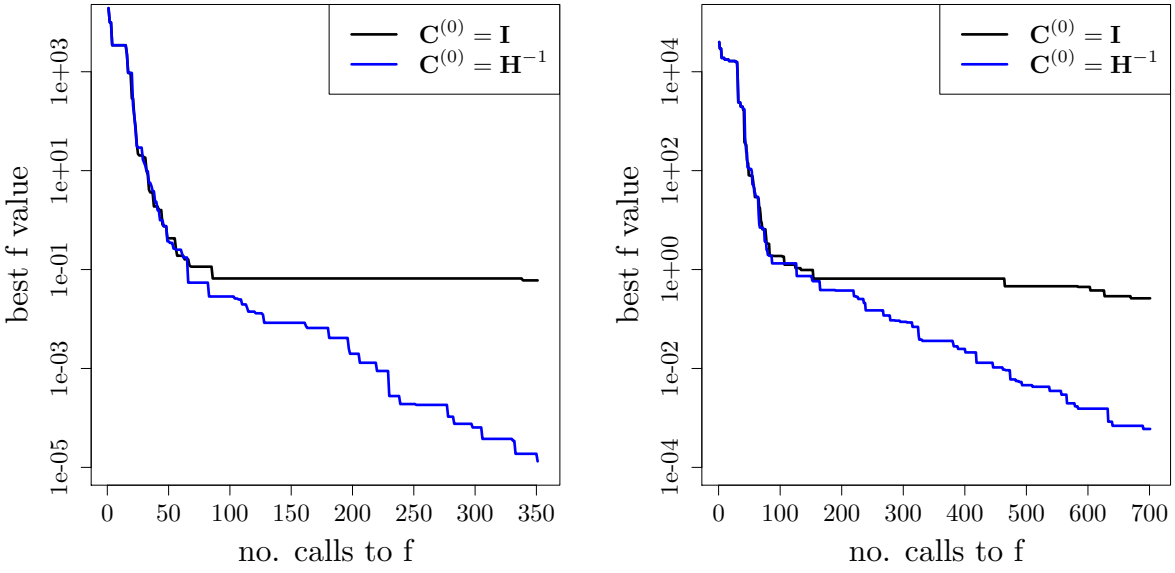


Figure 4.8: Median of the best objective function vs. number of calls of EGO-CMA in dimensions 5 (left) and 10 (right) on Quadratic function with the condition number of 10^3. Using $\mathbf{H}^{-1}$ instead of $\mathbf{I}$ as the initial covariance matrix of CMA-ES in the EGO-CMA algorithm can significantly improve the algorithm's performance.

4.5 Conclusions

This chapter presents a new algorithm, EGO-CMA, for unconstrained continuous black-box optimization. The EGO-CMA combines the strengths of EGO and CMA-ES in such a way that search domain is first explored by EGO and a point with the lowest function value is selected, $\mathbf{x}^{best}$. Then CMA-ES, as a robust local search, is started from $\mathbf{x}^{best}$ in order to converge the minimum with high accuracy. Besides, the initial values of CMA-ES step-size and covariance matrix are improved. The results of our experiments show that

the EGO-CMA algorithm outperforms EGO and CMA-ES in most of the cases.

Chapter 5

A detailed analysis of kernel parameters in Gaussian process-based optimization

Abstract

The efficiency of EGO algorithm is mainly determined by the Gaussian process covariance function which must be chosen together with the objective function. Traditionally, a parameterized family of covariance functions is considered whose parameters are learned by maximum likelihood or cross-validation. In this chapter, we theoretically and empirically analyze the effect of length-scale covariance parameters and nugget on the design of experiments generated by EGO and the associated optimization performance.

5.1 Introduction

The way the kriging model is learned from data points is essential to the EGO performance. A kriging model is mainly described by the associated kernel and this kernel determines the set of possible functions processed by the algorithm to make optimization decisions. Several methods alternative to cross-validation or Maximum Likelihood (ML) have been proposed to tune the kernel parameters. For example, a fully Bayesian approach is used in [BBV11]. In [Jon01], the process of estimating parameters and searching for the optimum are combined together through a likelihood which encompasses a targeted objective. In

[WZH+13], the bounds on the parameter values are changing with the iterations following an a priori schedule. Nevertheless, we feel that the existing methods for learning kernel parameters are complex so that the basic phenomena taking place in the optimization when tuning the kernel cannot be clearly observed. This study allows to more deeply understand the influence of kriging parameters on the efficiency of EGO by studying the convergence of EGO with fixed parameters on a unimodal and a multimodal function. The effect of nugget is also investigated.

5.2 Kriging model summary

To make this chapter self-contained, we provide a short introduction to the kriging model. But for more details see Chapter 2. Let $\mathbf{X} = \{\mathbf{x}^1, \ldots, \mathbf{x}^n\}$ be a set of n design points and $\mathbf{y} = \{f(\mathbf{x}^1), \ldots, f(\mathbf{x}^n)\}$ the associated function values at $\mathbf{X}$. Suppose the observations are a realization of a stationary GP, $Y(\mathbf{x})$. The kriging model is the GP conditional on the observations, $Y(\mathbf{x}) \mid Y(\mathbf{x}^1) = \mathbf{y}_1, \ldots, Y(\mathbf{x}^n) = \mathbf{y}_n$, also written in a more compact notation, $Y(\mathbf{x}) \mid Y(\mathbf{X}) = \mathbf{y}$. The GP's prediction (simple kriging mean) and variance of prediction (simple kriging variance) at a point $\mathbf{x}$ are

$$m(\mathbf{x}) = \mu + \mathbf{r}(\mathbf{x})^\top \mathbf{R}^{-1}(\mathbf{y} - \mu\mathbf{1}), \quad (1)$$

$$s^2(\mathbf{x}) = \sigma^2 \left(1 - \mathbf{r}(\mathbf{x})^\top \mathbf{R}^{-1}\mathbf{r}(\mathbf{x})\right). \quad (2)$$

Here, μ and σ^2 are the constant process mean and variance, $\mathbf{1}$ is a $n \times 1$ vector of ones, $\mathbf{r}(\mathbf{x})$ is the vector of correlations between point $\mathbf{x}$ and the n sample points, $\mathbf{r}(\mathbf{x}) = [\mathrm{Corr}(Y(\mathbf{x}), Y(\mathbf{x}^1)), \ldots, \mathrm{Corr}(Y(\mathbf{x}), Y(\mathbf{x}^n))]$, and $\mathbf{R}$ is an $n \times n$ correlation matrix between sample points of general term $\mathbf{R}_{ij} = \mathrm{Corr}(Y(\mathbf{x}^i), Y(\mathbf{x}^j))$. The covariance function (i.e., the kernel) used here is the isotropic Matérn 5/2 function defined as [RW05]

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \mathrm{Corr}(Y(\mathbf{x}), Y(\mathbf{x}')) = \sigma^2 \left(1 + \frac{\sqrt{5}\|\mathbf{x}-\mathbf{x}'\|}{\theta} + \frac{5\|\mathbf{x}-\mathbf{x}'\|^2}{3\theta^2}\right) \exp\left(-\frac{\sqrt{5}\|\mathbf{x}-\mathbf{x}'\|}{\theta}\right), \quad (3)$$

in which the parameter $\theta > 0$ is *characteristic length-scale* that controls the correlation strength between pairs of response values. More generally, all stationary isotropic covariance functions have such a characteristic length-scale. Anisotropic covariance functions have d such length-scales, one per dimension, as can be seen below with the usual tensor

product kernel,

$$k(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}) = \sigma^2 \prod_{i=1}^{d} k_i \left(\frac{|x_i - x_i'|}{\theta_i} \right) \tag{4}$$

In order to simplify the analysis, we will focus in the following on the unique length-scale case, $\theta_1 = \cdots = \theta_d = \theta$. The smaller the characteristic length-scale θ, the least two response values at given points are correlated, and vice versa, see Figure 5.1.

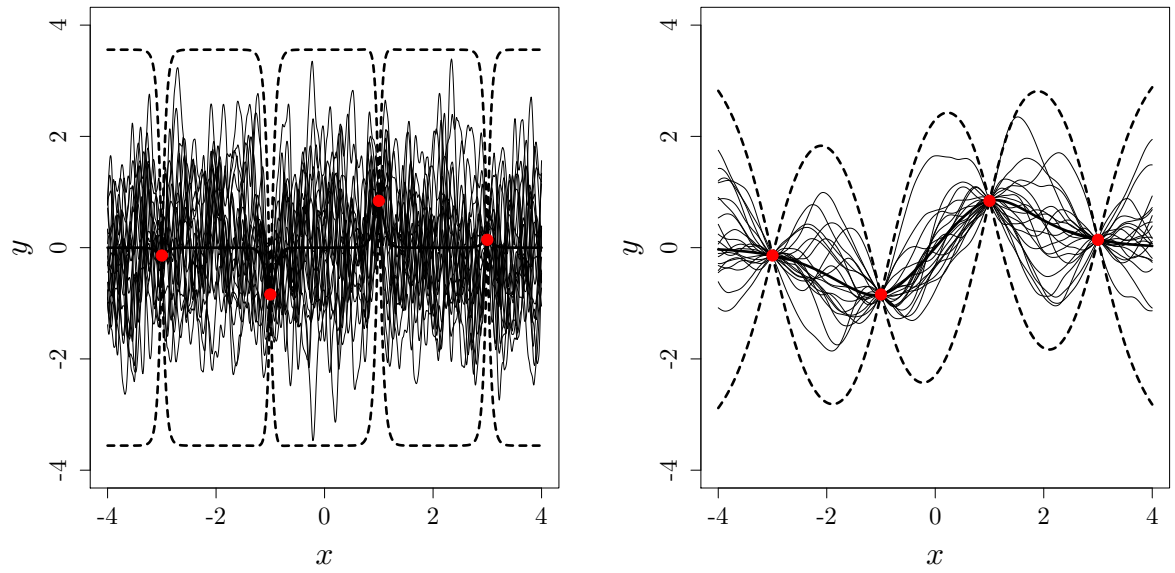


Figure 5.1: Kriging mean (thick solid line) along with the 95% confidence intervals (thick dashed lines), i.e., $m(\mathbf{x}) \pm 1.96 s(\mathbf{x})$, for $\theta = 0.1$ (left) and $\theta = 1$ (right). The thin lines are the sample paths of the GP. As θ changes, the class of possible functions considered for the optimization decision changes. Therefore, θ is a central decision for the optimization that deserves an in-depth study.

When a nugget, τ^2, is added to the model, the covariance function becomes

$$k_{\tau^2}(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') + \tau^2 \delta(\mathbf{x}, \mathbf{x}'), \tag{5}$$

where $\delta(.,.)$ is the Kronecker's delta. Adding nugget to the model means that the observations are perturbed by an additive Gaussian noise $\mathcal{N}(0, \tau^2)$. The resulting kriging

predictions, $m(\mathbf{x})$, are smoother as they no longer interpolate the observations[1]. Nugget also increases kriging variance throughout the search domain since, beside the changes in the covariance matrix $\mathbf{R}$, the term σ^2 becomes $\sigma^2 + \tau^2$ in Equation (2).

Classically here, the process mean and variance, without nugget, are estimated by the following ML closed-form expressions [RW05],

$$\hat{\mu} = \frac{\mathbf{1}^\top \mathbf{R}^{-1}\mathbf{y}}{\mathbf{1}^\top \mathbf{R}^{-1}\mathbf{1}} \quad , \quad \hat{\sigma}^2 = \frac{(\mathbf{y} - \mathbf{1}\hat{\mu})^\top \mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu})}{n} \ , \tag{6}$$

so that the only kernel parameters left are θ and τ^2.

At any point $\mathbf{x}$ in $\mathcal{S}$, the improvement is defined as the random variable $I(\mathbf{x}) = \max(0, f_{min} - Y(\mathbf{x}))$ where f_{min} is the best objective function value observed so far. The improvement is the random excursion of the process at any point below the best observed function value. The expected improvement can be calculated analytically as

$$EI(\mathbf{x}) = \begin{cases} (f_{min} - m(\mathbf{x}))\Phi\left(\frac{f_{min} - m(\mathbf{X})}{s(\mathbf{X})}\right) + s(\mathbf{x})\phi\left(\frac{f_{min} - m(\mathbf{X})}{s(\mathbf{X})}\right) & \text{if } s(\mathbf{x}) > 0 \\ 0 & \text{if } s(\mathbf{x}) = 0 \ , \end{cases} \tag{7}$$

where Φ and ϕ denote the cumulative distribution function and probability density function of the standard normal distribution, respectively. The first term in Equation (7) is dominated by the contribution of kriging mean to the improvement while the second term is dominated by the contribution of kriging variance. The EGO algorithm consists in the sequential maximization of EI, $\mathbf{x}^{n+1} \in \arg\max_{x \in \mathcal{S}} EI(\mathbf{x})$ followed by the updating of the kriging model with $\mathbf{X} \cup \{\mathbf{x}^{n+1}\}$ and the associated responses $\mathbf{y}$.

5.3 EGO with fixed length-scale

We start by discussing the behavior of EGO with two different fixed length-scales (small and large). The magnitude of length-scale is measured with respect to the longest possible distance in the search space, $Dist_{max}$ which, in our d-dimensional box search space is equal to $(UB - LB)\sqrt{d}$. θ is large if it is close to or larger than $Dist_{max}$ and vice versa. Here,

[1]Strictly speaking, if the covariance function of Equation (5) is directly input into the kriging model, the trajectories are discontinuous and interpolating the observations. Therefore, often, nugget is only put on the covariance matrix and not on the covariance vector, which means that the observations are noisy but the prediction is not. This last strategy to handle noise is called "`noise.var=`" in the DiceKriging package [RGD12] and is further discussed in Chapter 2, Section 2.2.2.

$LB = -5$ and $UB = 5$. Figure 5.5 illustrates the kriging models on the Ackley test function (defined below) in 1 dimension and the associated EIs for small and large length-scales.

5.3.1 EGO with small characteristic length-scale

When θ is small, there is a low correlation between response values so that data points have an influence on the process only in their immediate neighborhood. As $\theta \to 0$ and away from the data points, the kriging mean and variance of Equations (1) and (2) turn into the constants μ and σ^2, respectively, thus the EI becomes a constant flat function: when $\mathbf{x}$ is away from $\mathbf{x}^i$, $EI(\mathbf{x}) \to EI^{\text{asymp}} := (f_{min} - \hat{\mu})\Phi\left(\frac{f_{min}-\hat{\mu}}{\hat{\sigma}}\right) + \hat{\sigma}\phi\left(\frac{f_{min}-\hat{\mu}}{\hat{\sigma}}\right)$, where $\hat{\mu} \to \frac{\sum_{i=1}^{n} y^i}{n}$ and $\hat{\sigma}^2 \to \frac{\sum_{i=1}^{n}(y^i-\hat{\mu})^2}{n}$ since $\mathbf{R}$ tends to the identity matrix in Equation (6).

Proposition 1 (EGO iterates for small length-scale)**.** *Without loss of generality, we assume that the best observed point is unique. As the characteristic length-scale of the GP kernels tend to 0, the EGO iterates are located in a shrinking neighborhood of the best observed point.*

This proposition is explained and proved below.

Irrespectively of the function being optimized and the current DoE (provided the best observed point is uniquely defined), the set of design points created by EGO with small θ has characteristically repeated samples near the best observed points. An example is provided in Figure 5.2 where $\theta = 0.001$. Elements of proof of this phenomenon is given below.

When the length-scale is small, the observations have a low range of influence. In the limit case, one can assume that in a vicinity of ith design point the correlation between $Y(\mathbf{x}^i)$ and the other observations is zero, i.e., $\text{Corr}(Y(\mathbf{x}^i), Y(\mathbf{x}^j)) \to 0$, $1 \leq j \leq n$, $j \neq i$, so that $R \to I$. Let $\mathbf{x}$ be in the neighborhood of $\mathbf{x}^i$, $B_\epsilon(\mathbf{x}^i) = \{\mathbf{x} \in \mathcal{S} : \ \|\mathbf{x} - \mathbf{x}^i\| \leq \epsilon\}$, for a sufficiently small ϵ and away from the other points of the Design of Experiments (DoE) $j \neq i$ so that the correlation vector tends to $\mathbf{r}(\mathbf{x}) \to [0, \ldots, 0, r, 0, \ldots 0]$ where $r = \text{Corr}(Y(\mathbf{x}), Y(\mathbf{x}^i))$. In this situation, the kriging mean and variance can be fully expressed in terms of the correlation r (a scalar in $[0, 1]$):

$$m(r) = \hat{\mu} + r(y_i - \hat{\mu}) = \hat{\mu}(1 - r) + r y_i, \tag{8}$$

$$s^2(r) = \hat{\sigma}^2(1 - r^2), \tag{9}$$

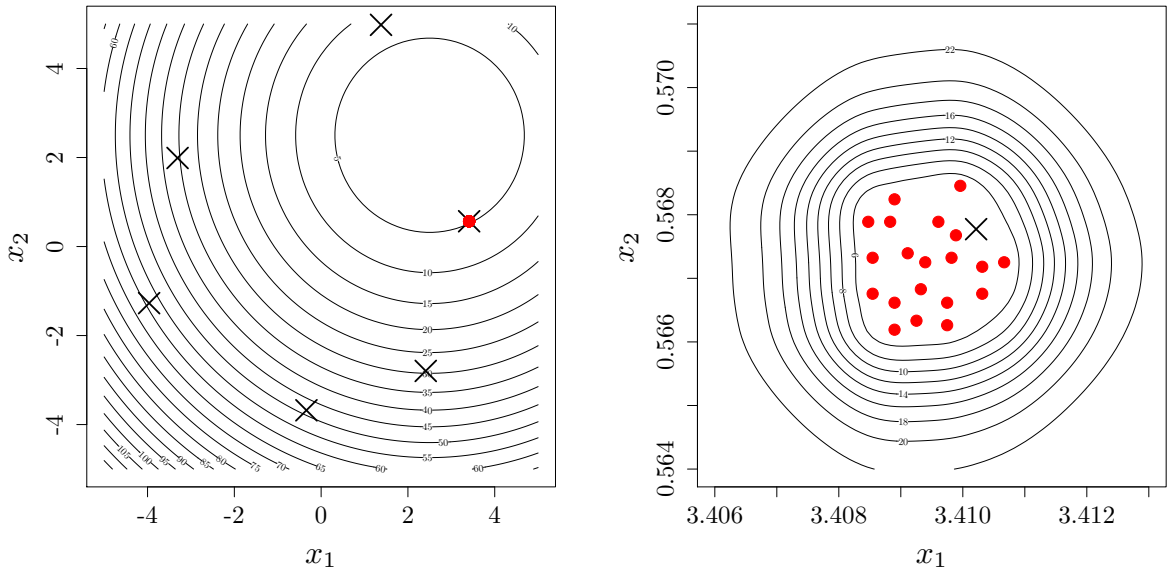


Figure 5.2: Left: search points obtained during 20 iterations of EGO with a small length-scale ($\theta = 0.001$) on the Sphere function whose contour lines are plotted. Crosses are the initial design points. The points accumulate in the vicinity of the design point with the lowest function value. Right picture: zoom around the best observed point; the contour lines show the kriging mean.

It is visible from the above equations that, among the points of the DoE, the expected improvement will be the largest near the best observed point as, for any given r, the variance will be the same and the mean will be the lowest. If many points of the DoE share the same best performance f_{min}, we will consider $\mathbf{x}^{min}$, the most isolated[2] one. By setting $y_i = f_{min}$ in Eqs. (8) and (9), the expected improvement (Equation (7)) in the vicinity of the best observed point becomes,

$$\begin{aligned} EI(r) \ =&(1-r)(f_{min}-\hat{\mu})\Phi\left(\frac{f_{min}-\hat{\mu}}{\hat{\sigma}}\sqrt{\frac{1-r}{1+r}}\right)+ \\ &\hat{\sigma}\sqrt{1-r^2}\phi\left(\frac{f_{min}-\hat{\mu}}{\hat{\sigma}}\sqrt{\frac{1-r}{1+r}}\right). \end{aligned} \tag{10}$$

Dividing both sides of Equation (10) by $\hat{\sigma}$ and introducing the new variable $A := \frac{f_{min}-\hat{\mu}}{\hat{\sigma}}$, the normalized expected improvement $EI(r)/\hat{\sigma}$, reads

$$EI(r)/\hat{\sigma} = (1-r)A\Phi\left(A\sqrt{\frac{1-r}{1+r}}\right)+\sqrt{1-r^2}\phi\left(A\sqrt{\frac{1-r}{1+r}}\right). \tag{11}$$

The normalized improvement is handy in that, for small length scale, it sums up what happens for all objective functions, design of experiments and kernels in terms of only two scalars, the correlation r and A. Note that because $f_{min} \leq y_i \ , \ \forall i$, $A \leq 0$. Instances of normalized EI are plotted for a set of A's in $[-2, -0.001]$ in the left of Figure 5.3. The value of EI when $r \to 0^+$ is the asymptotic value of expected improvement as $\mathbf{x}$ moves away from data points. The maximum of EI (equivalently $EI/\hat{\sigma}$) is reached at $r^\star$ which is strictly larger than 0. All the values of $r^\star$ are represented as a function of A in the right plot of Figure 5.3. As A decreases (i.e., f_{min} further drops below $\hat{\mu}$, or the best observation improves with respect to the other observations), $r^\star$ tends to 1, that is EGO will create the next iterate closer to $\mathbf{x}^{min}$, which makes sense since the point gets better. Vice versa, as the advantage of the best observation reduces (A diminishes), $r^\star$ approaches 0, which means that EGO will put the next iterate further from $\mathbf{x}^{min}$. Note that the analytical formulas for the first and second derivative of EI with respect to r are given in Section 5.4.

[2]the most isolated in terms of the metric used by the covariance functions of the GP.

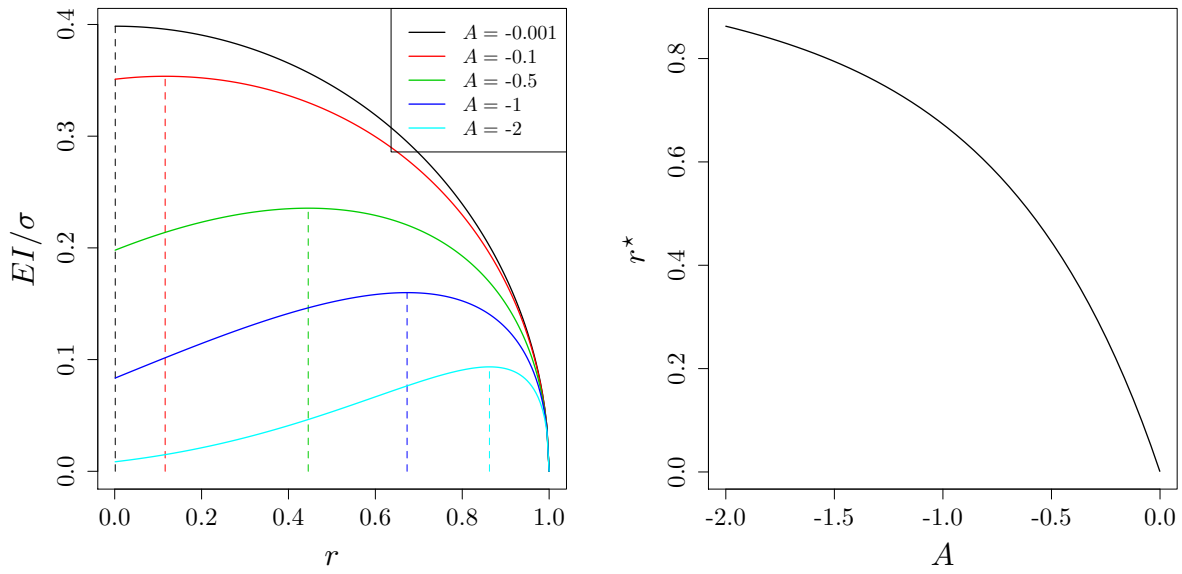


Figure 5.3: Left: Normalized EI as a function of $r \in]0, 1]$ in the vicinity of the sample point with the lowest function value for a small length-scale. Right: location of the next EGO iterate ($r^\star$ where EI is maximized) as a function of A.

5.3.2 EGO with large characteristic length-scale

Proposition 2 (EGO iterates for large length-scale)**.** *As the characteristic length-scale of the GP kernels increases, $\theta \to \infty$, the EGO algorithm degenerates into the sequential minimization of the kriging mean $m(\boldsymbol{x})$.*

This behavior of EGO can be understood by seeing that as the length-scale increases, the points have more influence on each other and the uncertainty, as described by kriging variance $s^2(\mathbf{x})$ in Equation (2), vanishes. Then, we will see that maximizing the expected improvement is equivalent to minimizing the kriging mean when kriging variance is null.

Let us demonstrate the above statements. We first establish that the term $\mathbf{r}(\mathbf{x})^\top \mathbf{R}^{-1}\mathbf{r}(\mathbf{x})$ in the kriging variance of Equation (2) tends to 1. As $\theta \to \infty$, all the responses $Y(x)$ are strongly correlated, therefore $\mathbf{r}(\mathbf{x})$ and $\mathbf{R}$ become a vector and a matrix of 1's. This matrix $\mathbf{R}$ has only one non-zero eigenvalue that equals n, the matrix size [AC12]. The corresponding eigenvector is $\mathbf{v} = \frac{\sqrt{n}}{n}(1, \ldots, 1)^\top$.

To invert such a non-invertible matrix, we use *Moore-Penrose pseudoinverse* [Str88], which is equivalent to regularizing it with a very small nugget (see [MLRD+16]). The

pseudoinverse of $\mathbf{R}$, denoted by $\mathbf{R}^\dagger$, is

$$\mathbf{R}^\dagger = [\mathbf{v}\ \mathbf{W}] \begin{bmatrix} \frac{1}{n} & \mathbf{0}_{1\times(n-1)} \\ \mathbf{0}_{(n-1)\times 1} & \mathbf{0}_{(n-1)\times(n-1)} \end{bmatrix} [\mathbf{v}\ \mathbf{W}]^\top , \tag{12}$$

in which $\mathbf{W}$ contains the $n-1$ eigenvectors associated with the zero eigenvalues. Regularizing $\mathbf{R}^{-1}$ as $\mathbf{R}^\dagger$ in $\mathbf{r}(\mathbf{x})^\top \mathbf{R}^{-1} \mathbf{r}(\mathbf{x})$ and since $\mathbf{r}(\mathbf{x})^\top \to (1, \ldots, 1)$ as $\theta \to \infty$, it is easy to show that $\mathbf{r}(\mathbf{x})^\top \mathbf{R}^\dagger \mathbf{r}(\mathbf{x}) = 1$. As a result, $s^2(\mathbf{x}) \to 0$ and $EI(\mathbf{x}) \to f_{min} - m(\mathbf{x})$. In this case, the EGO search degenerates to an iterative minimization and updating of the kriging mean $m(\mathbf{x})$.

Minimizing kriging mean does not define a valid global optimization scheme for two reasons. Firstly, because premature convergence occurs as soon as the minimum of $m(\mathbf{x})$ coincides with an observation of the true function [Jon01]: when $m(\mathbf{x}^{n+1}) = f(\mathbf{x}^{n+1})$ where $\mathbf{x}^{n+1} = \arg\min_{\mathbf{x}\in\mathcal{S}} m(\mathbf{x})$, the EGO iterations with large θ stop producing new points, however $\mathbf{x}^{n+1} \cup \mathbf{X}$ may not even contain a local optimum of f. Secondly, it should be remembered that the kriging mean discussed here is that stemming from large length-scale, which may not allow an accurate prediction of the objective function considered: it would suit a function like the sphere with a Matérn kernel, but it would not suit a multimodal function like Ackley.

The DoE created by EGO with large θ can vary greatly depending on the function and the initial DoE. On the one hand, if the function is regular and well predicted by $m(.)$ around $\mathbf{x}^{n+1}$, like the Sphere function, the kriging mean rapidly converges to the true function and points are accumulated in this region which may or not be the global optimum. Figure 5.4 illustrates both situations (true and false convergence) with the DoEs created by an EGO algorithm with large length-scale on a unimodal and a multimodal function (Sphere and Rastrigin functions, respectively). The Rastrigin function is defined as

$$f_{\text{Rastrigin}(\mathbf{x})} = 10d + \sum_{i=1}^{d} \left(x_i^2 - 10\cos(2\pi x_i) \right) . \tag{13}$$

On the other hand, if $m(\mathbf{x}^{n+1})$ is different from $f(\mathbf{x}^{n+1})$, the kriging mean changes a lot between iterations because new observations have a long range influence. The kriging mean overshoots observations in both upper and lower directions (cf. the dotted blue curve in the upper left plot of Figure 5.5). The resulting DoE is more space-filling than the DoE of small length scale. An example of such DoE is provided at the bottom right of Figure 5.5.

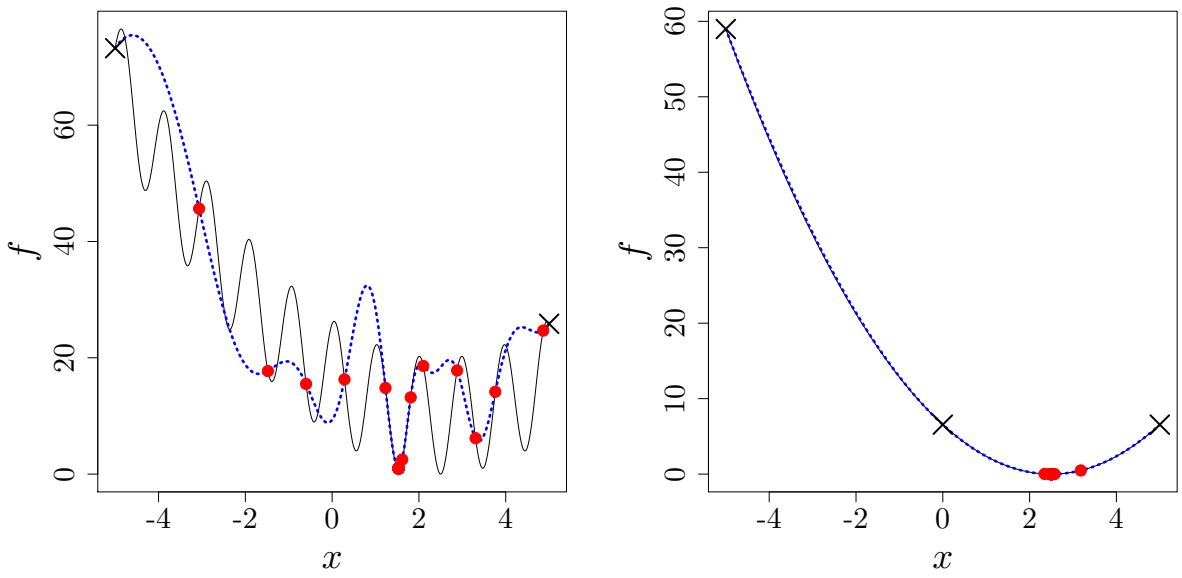


Figure 5.4: DoE created by EGO with $\theta = 100$. For such a large θ, the global search turns into the sequential minimization of the kriging mean. Left: premature convergence of the algorithm in a local minimum of the Rastrigin function because $m(\mathbf{x}^{n+1}) = f(\mathbf{x}^{n+1})$. The true optimum is at $x^\star = 2.5$ in the neighboring basin of attraction. Right: the algorithm converges to the global minimum of the unimodal Sphere function. In both functions the global minimum is located at 2.5.

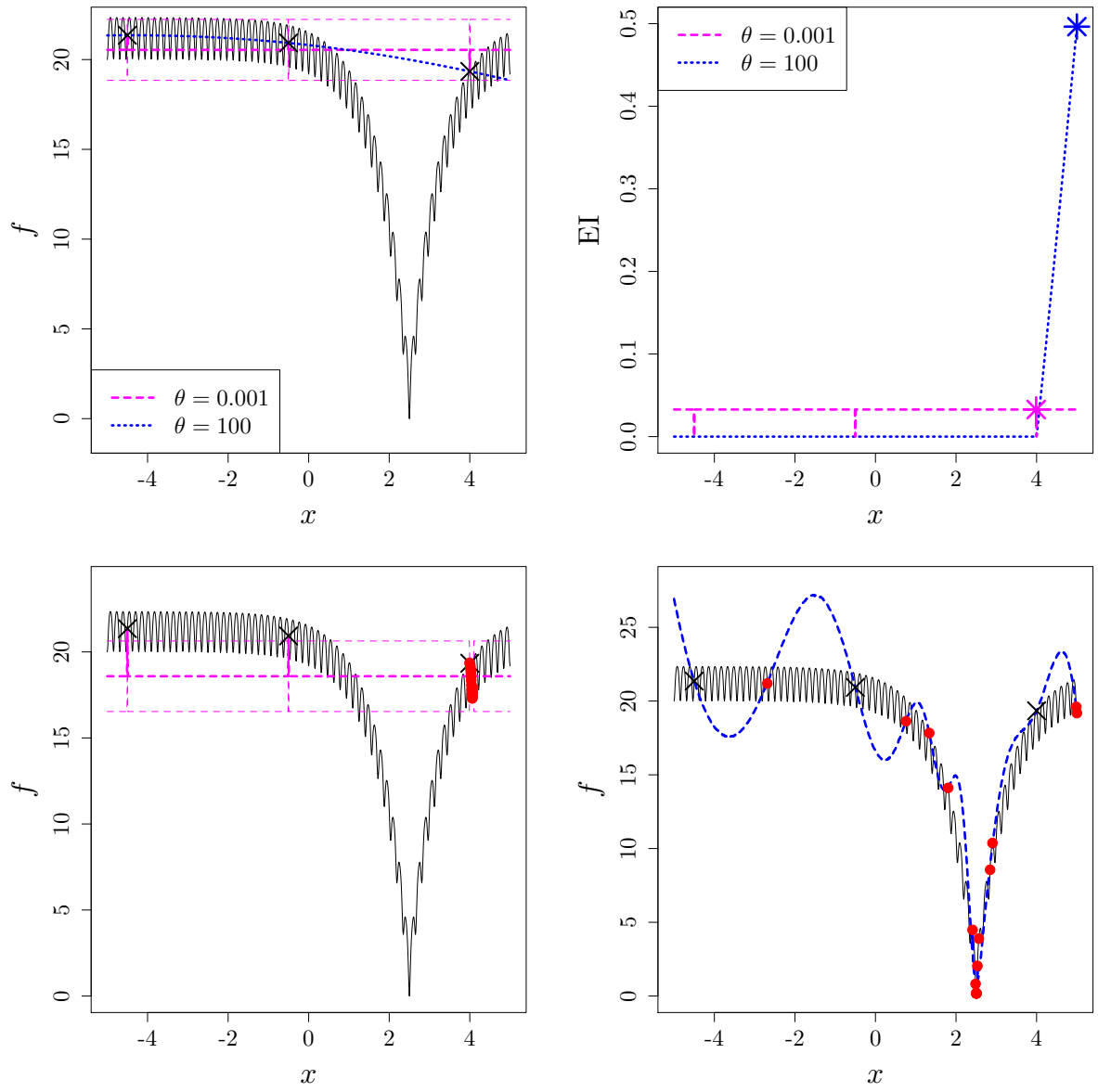


Figure 5.5: Ackley function (black solid line and defined in (21)) approximated by a kriging model (mean ± std. deviation, thick/thin lines) with $\theta = 0.001$ (dashed pink) and $\theta = 100$ (dotted blue). The crosses are the initial DoE. Top, right: EIs at iteration 1 with the stars indicating the EI maximums. Bottom, red bullets: DoEs created by EGO after 20 iterations with $\theta = 0.001$ (left) and $\theta = 100$ (right).

5.4 Expected Improvement and its derivatives for small length-scale

When the length-scale is small, the normalized expected improvement tends to the following analytical expression

$$\frac{EI(r)}{\sigma} = (1-r)A\Phi\left(A\sqrt{\frac{1-r}{1+r}}\right) + \sqrt{1-r^2}\phi\left(A\sqrt{\frac{1-r}{1+r}}\right), \tag{14}$$

where r is the correlation with the best observed point and $A = \frac{f_{min}-\hat{\mu}}{\hat{\sigma}}$. Such expression applies to any objective functions, designs of experiment and kernels as long as the length-scale tends to 0. We want to calculate the first and the second derivatives of the normalized expected improvement with respect to r: To do so, we need to calculate the derivative of each term. Here, we present the derivatives of the terms $\Phi\left(A\sqrt{\frac{1-r}{1+r}}\right)$, $\phi\left(A\sqrt{\frac{1-r}{1+r}}\right)$ and $\sqrt{\frac{1-r}{1+r}}$ which are

$$\frac{\partial}{\partial r}\Phi\left(A\sqrt{\frac{1-r}{1+r}}\right) = A\left(\frac{\partial}{\partial r}\sqrt{\frac{1-r}{1+r}}\right)\phi\left(A\sqrt{\frac{1-r}{1+r}}\right), \tag{15}$$

$$\frac{\partial}{\partial r}\phi\left(A\sqrt{\frac{1-r}{1+r}}\right) = -\left(A\sqrt{\frac{1-r}{1+r}}\right)\frac{\partial}{\partial r}\left(A\sqrt{\frac{1-r}{1+r}}\right)\phi\left(A\sqrt{\frac{1-r}{1+r}}\right), \tag{16}$$

$$\frac{\partial}{\partial r}\sqrt{\frac{1-r}{1+r}} = \frac{-\sqrt{1-r}}{2(1+r)^{3/2}} - \frac{1}{2\sqrt{1-r^2}}. \tag{17}$$

After calculating all the derivatives and simplification, the first derivative of $\frac{EI(r)}{\sigma}$ with respect to r can be written as

$$\frac{\partial EI(r)}{\sigma\partial r} = -A\Phi\left(A\sqrt{\frac{1-r}{1+r}}\right) - \frac{r}{\sqrt{1-r^2}}\phi\left(A\sqrt{\frac{1-r}{1+r}}\right). \tag{18}$$

In Figure 5.6, the first derivative of $\frac{EI(r)}{\sigma}$ for different values of A is numerically calculated. The location of a stationary point, $r^\star$, is where $\frac{\partial EI(r^\star)}{\sigma\partial r} = 0$, and it is also numerically estimated.

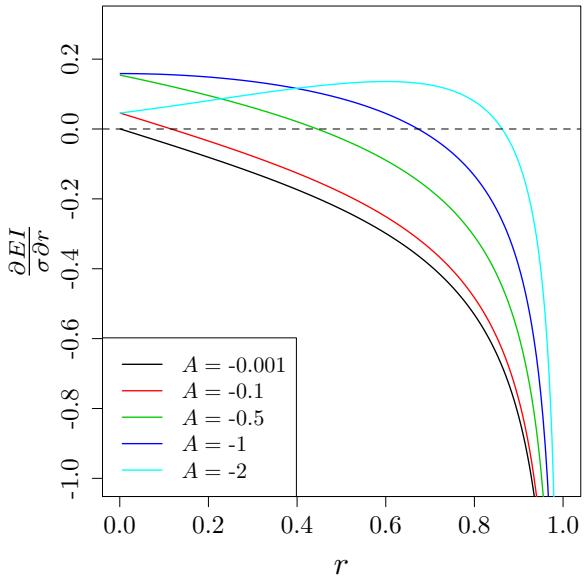


Figure 5.6: First derivative of $\frac{EI(r)}{\sigma}$ with respect to r for different values of A. The location of the stationary point becomes closer to $r = 0$ as $A \to 0^-$. In other words, for (negative) values of A different from 0, r is finite and the maximum of the EI is achieved near the best known point.

To determine the nature of the stationary points, the second derivative of $\frac{EI(r)}{\sigma}$, i.e., $\frac{\partial^2 EI}{\sigma \partial r^2}$, is required which is:

$$\frac{\partial^2 EI}{\sigma \partial r^2} = \left[\frac{A^2(1-r) - (1+r)}{(1+r)^{5/2}(1-r)^{3/2}}\right] \phi\left(A\sqrt{\frac{1-r}{1+r}}\right). \tag{19}$$

In the left picture of Figure 5.7 the second derivative of $\frac{EI(r)}{\sigma}$, $\frac{\partial^2 EI}{\sigma \partial r^2}$, with the same A values as used in Figure 5.6 is shown. In the right picture, the value of $\frac{\partial^2 EI}{\sigma \partial r^2}$ is plotted at the stationary points $r^\star$. It can be seen that the second derivatives are always negative. In other words, the curvature of the function $\frac{EI(r)}{\sigma}$ at any stationary points is negative and the function has a maximum there.

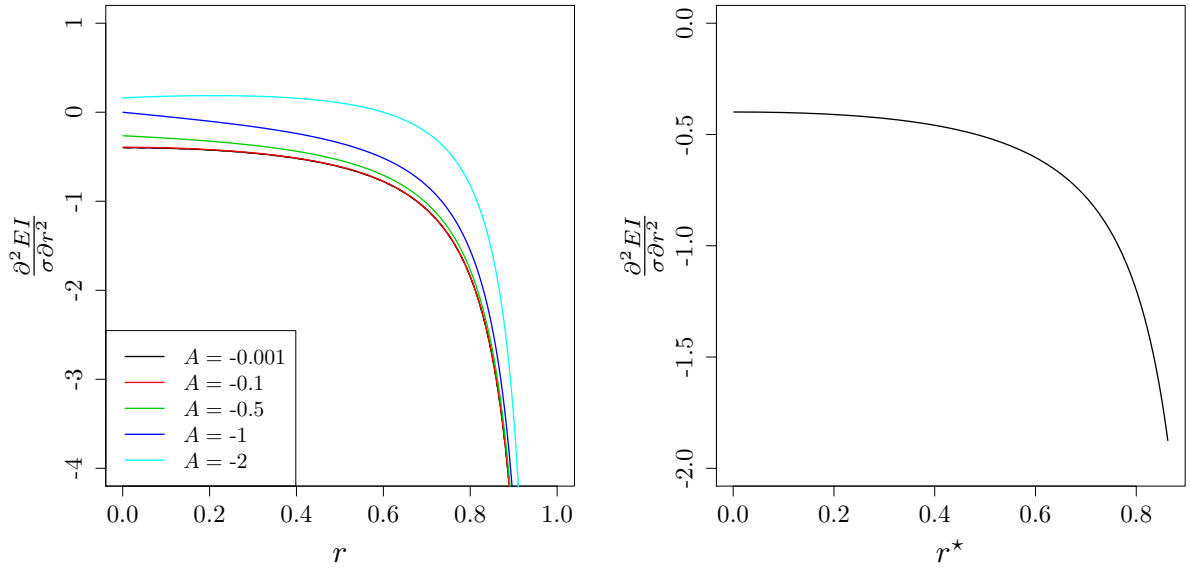


Figure 5.7: Left: second derivative of $\frac{EI(r)}{\sigma}$ when A equals to $-2, -1, -0.5, -0.1, -0.01$. The second derivative is negative most of the time excepted when A is small and r is close to 0 (compare to Figure 5.3). Right: the value of $\frac{\partial^2 EI}{\sigma \partial r^2}$ is plotted for different values of $r^\star$. This curvature is always negative.

5.4.1 Comparison of EGO with fixed and adapted length-scale

In the sequel, the efficiency of EGO with different fixed length-scale is compared with the standard EGO whose length-scale is learned by ML. Tests are carried out on two isotropic functions, the unimodal sphere and the highly multimodal Ackley functions:

$$f_{\text{Sphere}(\mathbf{x})} = \sum_{i=1}^{d} (x_i)^2, \tag{20}$$

$$f_{\text{Ackley}(\mathbf{x})} = -20 \exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d} \cos\left(2\pi x_i\right)\right) + 20 - \exp(1). \tag{21}$$

Each optimization is repeated 5 times on 5 dimensional instances of the problems, $d = 5$. The initial DoE is fixed and has size $3 \times d$. The search length is $70 \times d$. To allow comparisons of the results, the functions are scaled (multiplied) by $\frac{2}{f_{DoE}^{max} - f_{DoE}^{min}}$, where f_{DoE}^{min} and f_{DoE}^{max} are the smallest and the largest value of function f in the initial DoE.

Figure 5.8 shows the results of the comparison in terms of median objective functions. Moreover, the first and the third quartiles are plotted in Figure 5.9. The θ values belong

to the set $\{0.01, 0.1, 1, 5, 10, 20\}$. On both test functions, the algorithm does not converge quickly towards the minimum when $\theta = 0.01$ or $\theta = 0.1$ because, as explained in Section 5.3, it focuses on the neighborhoods of the best points found early in the search. On the Sphere function, EGOs with large length-scale, $\theta = 20$ or $\theta = 10$, have performances equivalent to that of the standard EGO. Indeed, the Sphere function is very smooth and, as can be seen on the rightmost plot of Figure 5.8, ML estimates of θ are equal to 20 (the upper bound of the ML) rapidly after a few iterations. With the multimodal Ackley function, the best fixed θ is equal to 1. It temporarily outperforms the standard EGO at the beginning of the search (until about 70 evaluations) but then ML allows decreasing the θ's until about 0.5 (see rightmost plot) and fine tuning the search in the already located high performance region. Note however that this early advantage of $\theta = 1$ over the adapted θ seem to be dependent on the initial DoE (cf. experiment with an alternative DoE in Figure 5.10).

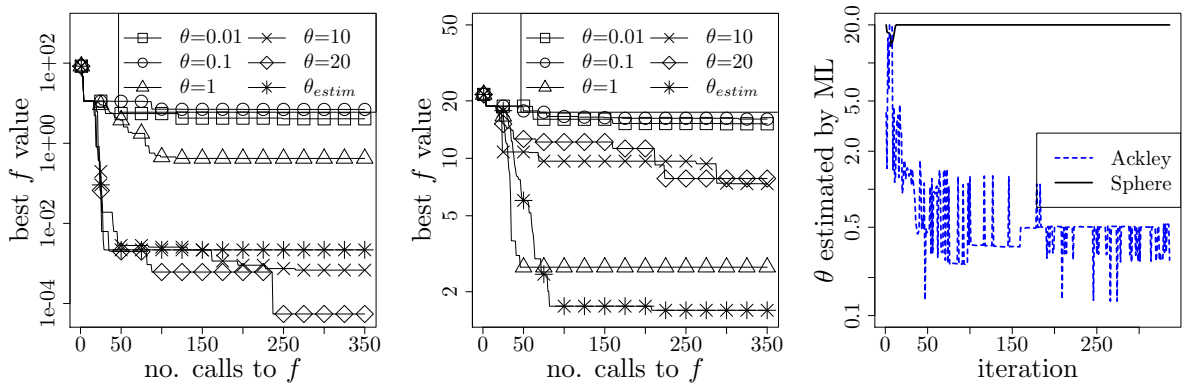


Figure 5.8: Median of the best objective function vs. number of calls of standard EGO and EGO with different fixed length-scale on the Sphere (left) and the Ackley (middle) functions, $d = 5$. Right: evolution of θ learned by ML in standard EGO.

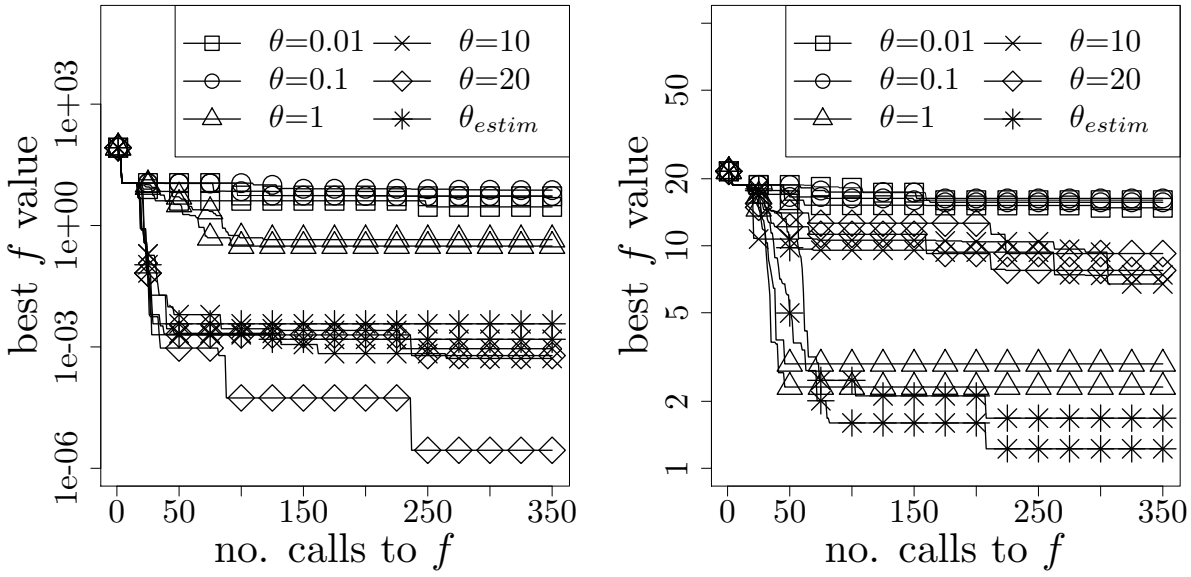


Figure 5.9: Dispersion of the results of Figure 5.8 : first and the third quartiles of the results for the Sphere (left) and Ackley (right) functions.

In order to investigate the effect of initial DoE on the above results, we repeat the same experiments with another fixed DoE. The results with the new DoE are given in Figure 5.10. These results are similar to those already reported in Figure 5.8, therefore suggesting a low sensitivity of EGO to the initial DoE. The main difference is visible in the initial iterations (before 100 calls) for the multimodal Ackley function and questions the early advantage at using $\theta = 1$ over θ adapted by ML.

A complementary view on convergence, focusing on distances to the optimum in the x-space and the whole set of search points created, as opposed to the objective function of the best point in the convergence plots (e.g., Figure 5.8), is given in Figure 5.11. Each curve represents the probability distribution of search points closer to the global minimum than a given distance. The procedure for calculating this density is to divide the number of points closer to the global minimum by the total number of the points of the search (here 350 when $d = 5$). The distances are normalized by dividing them by the square root of the problem dimension. This measure is invariant with respect to the monotonic scaling of the objective function. However, such curves that show the distribution of the points created by the algorithm are not used for ranking the algorithm.

For small distances to the optimum ($< 0.3 \times \sqrt{d}$), the algorithms hierarchy recovered

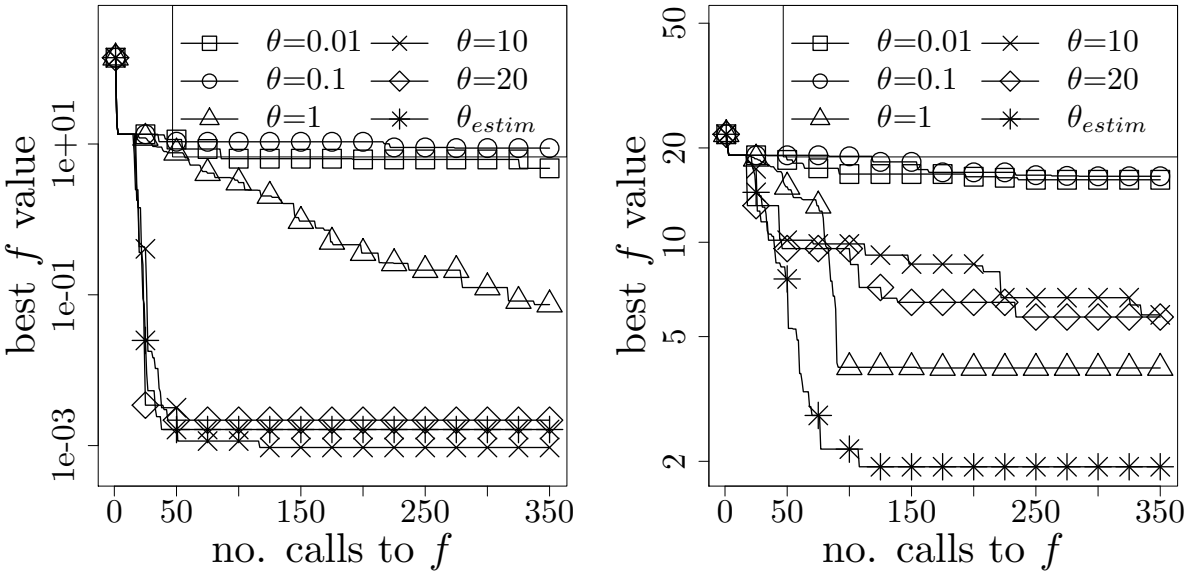


Figure 5.10: Median of the best objective function vs. number of calls of standard EGO and EGO with different fixed length-scale on the Sphere (left) and the Ackley (middle) functions, $d = 5$. Although the initial DoE is different from the one used in Figure 5.8, the EGO performance does not change a lot.

from these graphs is based on the best points and is similar to that of Figure 5.8. For larger distances, we find out that EGO with fixed $\theta = 1$ performs very well at creating many points within a distance of $1 \times \sqrt{d}$ to the optimum.

5.5 Effect of nugget on EGO convergence

To investigate the effect of nugget on EGO, we carry out the same test protocol as above but the length-scales are set by ML and two scenarios are considered: 1) the nugget τ^2 is estimated by ML, 2) a fixed nugget is taken from the set $\tau^2 \in \{10^{-2}, 10^{-4}, 10^{-6}, 10^{-8}, 0\}$ ($\tau^2 = 0$ means no nugget). Figure 5.12 shows the results. For both test functions, when the nugget value is large (10^{-2} or 10^{-4} or ML estimated on Ackley), EGO exhibits the worst performances: it does not converge faster and stops further from the optimum. The reason is that a large nugget deteriorates the interpolation quality of a kriging model when observations are not noisy like here. On the Sphere function, EGO rapidly locates the area of the optimum but the EI without nugget, which is null at data points, pushes the

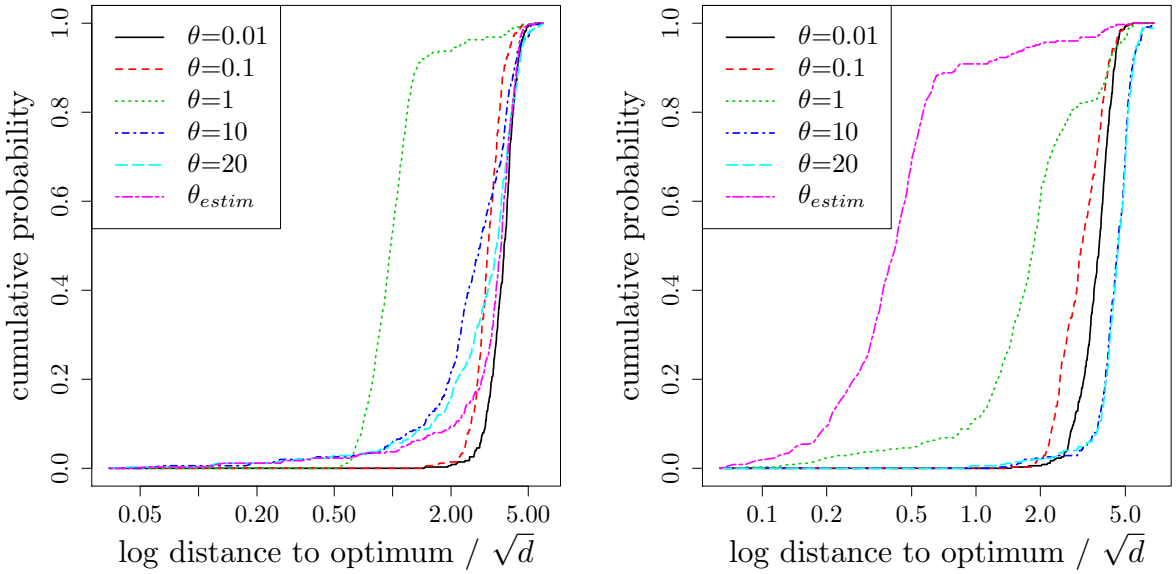


Figure 5.11: Density of points closer to the optimum than a given distance on Sphere (left) and Ackley (right) functions. Each curve is the median of 5 runs.

search away from it. However, a nugget value equal to 10^{-6} or 10^{-8} hardly slows down convergence and significantly improves the accuracy with which the optimum is found. Indeed, by increasing the uncertainty $s^2(\mathbf{x})$ everywhere including in the immediate vicinity of data points, where it would be null without nugget, nugget increases the EI there and allows a higher concentration of EGO iterates near the best observed point. The nugget learned by ML on the Sphere tends to 0 which, as just explained, is not the best setting for optimization.

On Ackley, besides large nugget values ($\tau^2 \geq 10^{-4}$) which significantly degrade the EGO search, values ranging from $\tau^2 = 0$ to 10^{-6} do not notably affect performance. In this case, the global optimum is not accurately located after $70 \times d$ evaluations of f, there is no need to allow through nugget an accumulation of points near the best observation.

Note that on both functions, when considering the best point found so far, ML estimation of nugget is not a good strategy.

Finally, the dispersion of all the search points the across the x-space is characterized in Figure 5.13 through the number (the density) of points closer to the optimum than a given distance (cf. previous section for a more detailed definition). For the Sphere function, $\tau^2 = 10^{-6}, 10^{-4}$ and 10^{-2} allow locating more points in a larger neighborhood of

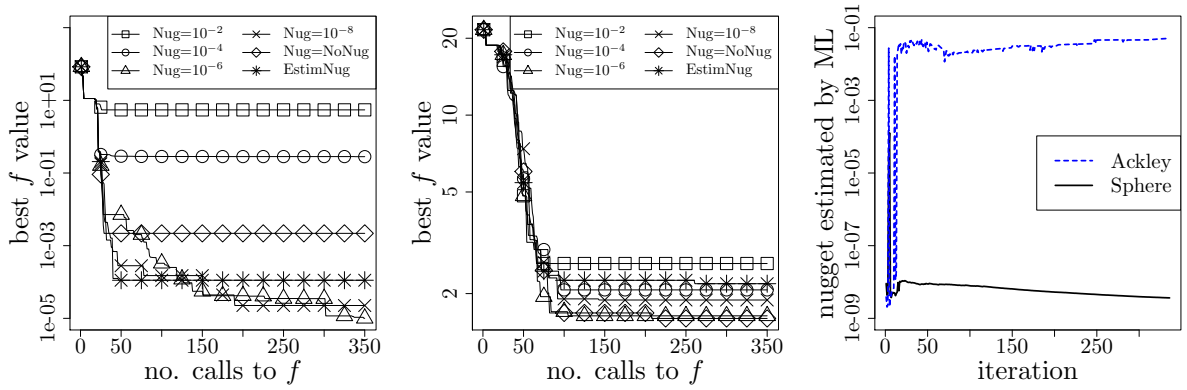


Figure 5.12: Median of the best objective function vs. number of calls to f for EGO with different nugget values on the Sphere (left) and Ackley (middle) functions in dimension 5. Right: ML estimated nugget, τ^2, vs. number of calls to f.

the optimum, respectively. For the Ackley function, no to moderate ($\tau^2 = 10^{-4}$) nuggets produce similar densities of points around the optimum; $\tau^2 = 10^{-2}$ seems to be often missing high performance areas; the ML estimate of τ^2, which after initial oscillations between 0 and 5.10^{-2}, stabilizes over 5.10^{-2}, puts 7% of the search points within a distance of $0.07 \times \sqrt{d}$ of the optimum (which makes it the best strategy at this distance to the optimum) but then puts the remaining points far from the optimum.

5.6 Conclusions

To sum up, this chapter carefully explains the DoEs generated by EGO with fixed length-scale and nugget. In terms of performance, ML estimation of the length-scale is a good choice but ML estimation of nugget is not recommended (a fixed small nugget value should be preferred). Based on our tests, as a perspective, EGO strategies starting with a large fixed length-scale and then decreasing it while keeping a small amount of nugget should be efficient while avoiding ML estimations which require $O(n^3)$ computations [CJ08]. Space-filling strategies can be created either by random jumps or by extrapolation. The reason for promoting large length-scale early in the search is motivated by extrapolation rather pure exploration.

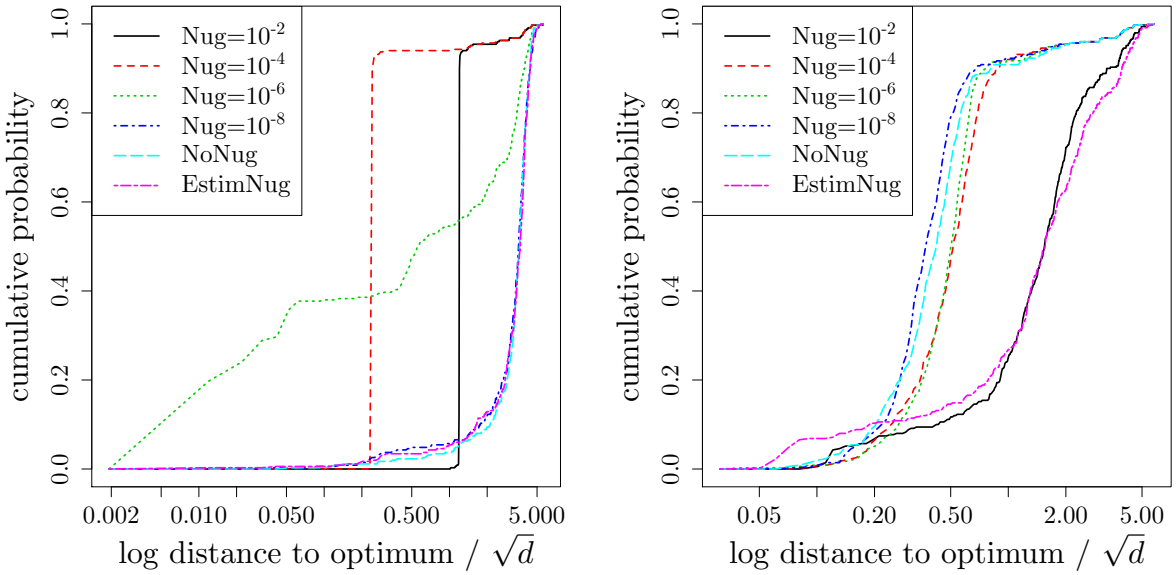


Figure 5.13: Cumulative probability of search points under different scenarios of nugget values on Sphere (left) and Ackley (right) function.

Chapter 6

Small ensembles of kriging models for optimization

The Efficient Global Optimization (EGO) algorithm uses a conditional Gaussian Process (GP) to approximate an objective function known at a finite number of observation points and sequentially adds new points which maximize the Expected Improvement criterion according to the GP. The important factor that controls the efficiency of EGO is the GP covariance function (or kernel) which should be chosen according to the objective function. Traditionally, a parameterized family of covariance functions is considered whose parameters are learned through statistical procedures such as maximum likelihood or cross-validation. However, it may be questioned whether statistical procedures for learning covariance functions are the most efficient for optimization as they target a global agreement between the GP and the observations which is not the ultimate goal of optimization. Furthermore, statistical learning procedures are computationally expensive. The main alternative to the statistical learning of the GP is self-adaptation, where the algorithm tunes the kernel parameters based on their contribution to objective function improvement. After questioning the possibility of self-adaptation for kriging based optimizers, we propose a novel approach for tuning the length-scale of the GP in EGO: At each iteration, a small ensemble of kriging models structured by their length-scales is created. All of the models contribute to an iterate in an EGO-like fashion. Then, the set of models is densified around the model whose length-scale yielded the best iterate and further points are produced. Numerical experiments are provided which motivate the use of many length-scales. The tested implementation does not perform better than the classical EGO algorithm in

a sequential context but show the potential of the approach for parallel implementations.

6.1 Introduction

The EGO optimization algorithm uses a kriging model, which is a conditional Gaussian process (GP) [RW05], for predicting objective function values and quantifying the prediction uncertainty. The shapes of sample paths of a GP such as its smoothness, periodicity, etc. are controlled by the covariance function of the process, also known as its kernel. Traditionally, a parameterized family of covariance functions is considered whose parameters are estimated.

The kernel parameters are often estimated by statistical approaches like maximum likelihood (ML)[Yin91] or cross validation (CV) [ZW10]. ML and CV are compared in [Bac13] when the covariance structure of a GP is misspecified. It is recommended in [LS05] to use a penalized likelihood for the kriging models when the sample size is small. However, the efficiency of such statistical approaches, which aims at learning the objective function globally, remains questionable in the context of optimization. For example, in the EGO algorithm if the design points do not carry enough information about the true function, the parameters are not estimated correctly. Theses parameters are then plugged into the expected improvement (EI) criterion that may lead to disappointing results [Jon01, BBV11].

Not surprisingly, several methods alternative to ML and CV have been proposed to tune the kernel parameters. For instance, in [FB08] the kernel parameters are estimated with a log normal prior density assumption over them. A fully Bayesian approach is used in [BBV11, TCR15]. In [JSW98, FJ08], the process of estimating parameters and searching for the optimum are combined together through a likelihood which encompasses a targeted objective. In [WZH$^+$13], the bounds on the length-scales values are changing with the iterations following an a priori schedule.

Another drawback of statistical learning procedures such as ML and CV in the context of moderately expensive functions[1] is their computational complexity as they involve the repeated inversion of an $n \times n$ covariance matrix (where n is the number of available

[1]We call "moderately expensive" functions that take between 10 seconds and an hour to be evaluated at one point.

observations) where each inversion needs of the order of n^3 operations.

This chapter considers isotropic kernels and investigates an alternative approach to tuning the length-scale parameter. In this approach, a small set of length scales (hence GP models) is first tested as alternative ways to consider the objective function, independently of their statistical relevance. The set is completed based on the direct contribution of the best model to the optimization. The method is based on ensembles of surrogates. It can also be seen as weakly self-adaptive in the sense of self-adaptive algorithms [B̈96, HO01] where no statistical measure intervenes in the building of the representation which the optimization algorithm has of the objective function.

Ensembles of surrogates have attracted a lot of attention from the machine learning community for prediction [HWB13], but fewer contributions seem to address surrogate ensembles for optimizing. Several approaches have been proposed that aggregate the metamodels of the ensemble into a hopefully better metamodel either by model selection or by mixing the models. This better metamodel is then used by the optimization algorithm [ARR09, CLRM13, GHSQ07].

On the opposite, other previous optimization methods take advantage of all the metamodels in the set as a diversity preserving mechanism (in addition to, of course, a way to reduce the number of calls to the objective function), in the context of evolutionary computation [JS04, LLJ13] or more generally [VHW13]. The algorithm studied in this text belongs to this category.

Another classification can be made with respect to the homogeneity (all metamodels are of the same type) or heterogeneity of the ensemble. There has been recent contributions to optimization algorithms that rely on a homogeneous set of kriging models: in [Kle14] the ensembles are built by bootstrap on the data and serve as a way to estimate model uncertainty for later use in optimization; in [VVW08], the metamodels are the trajectories of a GP and their contributions are aggregated through an uncertainty reduction criterion (on the entropy of the global optima of the trajectories). The optimization algorithm investigated here also relies on an homogeneous ensemble of GP models.

6.2 EGO algorithm overview

EGO is a sequential model-based optimization algorithm. It starts with an initial design of experiments (DoE). At each iteration, one point which maximizes the Expected Improvement (EI) according to the current kriging model is added to the DoE. Then, the kernel parameters are re-estimated and the kriging model is updated. The location of $\mathbf{x}^{n+1}$, where $\mathbf{x}^{n+1} = \arg\max_{x \in \mathcal{S}} EI(\mathbf{x})$, depends on the current DoE, $\mathbf{X}$, $\mathbf{y}$, the kriging trend, μ, and the kernel parameters: the length-scale, θ, and the process variance, σ^2. We use $\mathbf{x}^{n+1} = g(\mathbf{X}, \mu, \theta, \sigma^2)$ to denote that $\mathbf{x}^{n+1}$ is a function of the above-mentioned parameters. Figure 6.1 illustrates how the DoE and the magnitude of length-scale affect the EI.

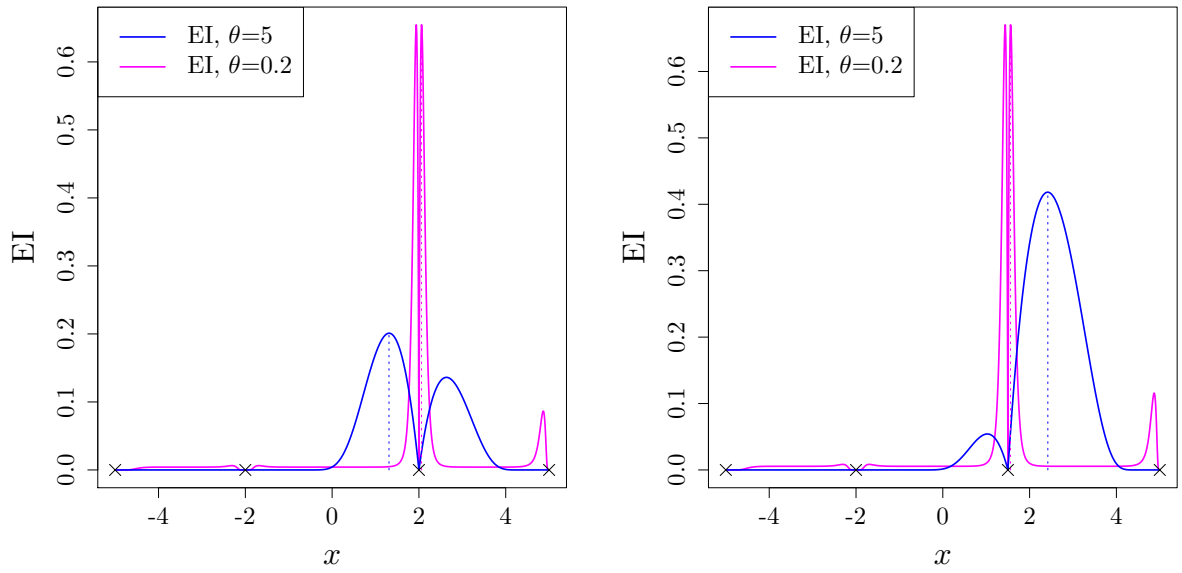


Figure 6.1: Effect of DoE and length-scale on EI function. The function to be optimized is the Sphere whose global minimum is located at 2.5. The blue and magenta curves represent the EI of kriging models with length-scales equal to 5 and 0.2, respectively. The crosses indicate the location of design points. The other parameters are fixed. The location of the third sample point changes from 2 to 1.5 in the right picture.

Among the parameters of the EI criterion, $\mathbf{X}$ and θ play a prominent role because once $\mathbf{X}$ and θ are fixed, the ML estimations of μ and σ^2 have a closed-form expression [RW05]:

$$\hat{\mu} = \frac{\mathbf{1}^\top \mathbf{R}^{-1}(\theta)\mathbf{y}}{\mathbf{1}^\top \mathbf{R}^{-1}(\theta)\mathbf{1}}, \tag{1}$$

$$\hat{\sigma}^2 = \frac{(\mathbf{y}-\hat{\mu}\mathbf{1})^\top \mathbf{R}^{-1}(\theta)(\mathbf{y}-\hat{\mu}\mathbf{1})}{n}. \tag{2}$$

Accordingly, $\mathbf{x}^{n+1}$ can be expressed as a function of $\mathbf{X}$ and θ. For example, Figure 6.2 shows all plausible next infill sample points by changing the length-scale for a given DoE.

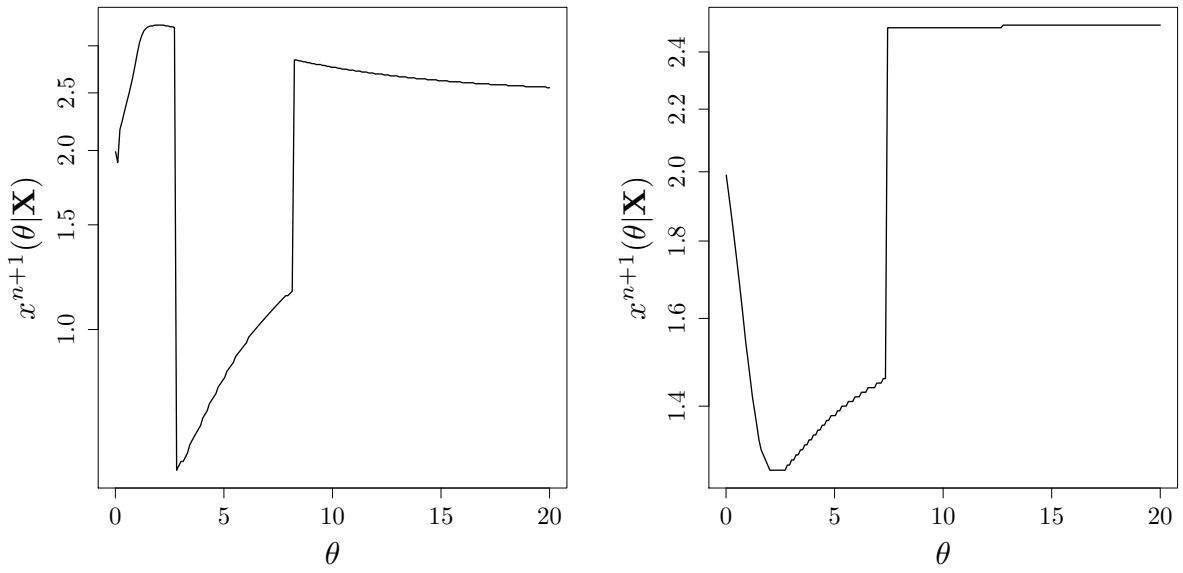


Figure 6.2: Illustration of all possible next infill sample points with $\mathbf{X} = \{-5, -2, 2, 5\}$ as the DoE. The true functions are Sphere (left, as in Figure 6.1) and Ackley (right) in dimension 1. For θ values larger than, say $\theta \geq 8$, the location of x^{n+1} is quite stable and close to 2.5, the location of the global minimum. While large θ's lead to the global optimum of the Sphere for any $\mathbf{X}$, it is a coincidence for Ackley's function.

6.3 Tuning the length-scale from an optimization point of view: a study on self-adaptation

When the kernel parameters are estimated by ML, the selected kriging model has statistical "best agreement" with the observed data. However, the goal of using EGO, like other optimization algorithms, is to solve an optimization problem with the least number of function evaluations. In other words, the main goal is the fast convergence of EGO even if the kriging model does not represents well the true function. This idea is similar to the notion of "self-adaptation" in evolutionary optimization [B̈96, HO01].

To investigate the potential of tuning the length-scale θ in an optimization oriented, greedy, self-adaptive way, we first tested a theoretical algorithm that tries a large number

θ's in the range $[0.01, 20]$. The true objective function values of the points that maximize the expected improvement for each of these length-scale θ value is calculated, $\mathbf{x}^{n+1}(\theta|\mathbf{X}) = \arg\max_{x\in\mathcal{S}} EI(\mathbf{x};\theta)$. This makes this algorithm not practical in the context of expensive problems. Then, the iterate associated to the best objective function, $\mathbf{x}^{\text{sel}} = \arg\min_{\mathbf{x}^{n+1}} f(\mathbf{x}^{n+1}(\theta|\mathbf{X}))$, is added to the Design of Experiment $\mathbf{X}$, the kriging model is updated, and the algorithm loops. This algorithm is sketched in the flow chart 6.1.

From a one step ahead optimization point of view, the "best" length-scale, denoted by θ^*, is the one that yields the next infill sample with the lowest objective function value, $\theta^* = \arg\min_\theta f(\mathbf{x}^{n+1}(\theta|\mathbf{X}))$. In the examples provided in Figure 6.3, the best length-scales are shown for the two test functions (Ackley and Sphere). In this example, the best length-scales are different from the length-scales estimated by ML, see the caption of Figure 6.3.

Algorithm 6.1 Toy EGO with greedy θ tuning

Create an initial design: $\mathbf{X} = [\mathbf{x}^1, \ldots, \mathbf{x}^n]^T$
Evaluate the functions at $\mathbf{X}$, $\mathbf{y} = f(\mathbf{X})$
while not stop (typically a limit on budget) **do**
  Set $\mathbf{x}^{\text{sel}} \leftarrow \arg\max_{\mathbf{x}^j\in\mathbf{X}}(f(\mathbf{x}^j))$
  for $\theta_i \in [\theta_{min}, \ldots, \theta_{max}]$ **do**
    $\mathbf{x}^{n+1}(\theta_i|\mathbf{X}) = \arg\max_{\mathbf{x}\in\mathcal{S}} EI(\mathbf{x};\theta_i)$
    if $f\left(\mathbf{x}^{n+1}(\theta_i|\mathbf{X})\right) < f(\mathbf{x}^{\text{sel}})$ **then**
      $\mathbf{x}^{\text{sel}} \leftarrow \mathbf{x}^{n+1}(\theta_i|\mathbf{X})$
    end if
  end for
  $\mathbf{X} \leftarrow \mathbf{X} \cup \mathbf{x}^{\text{sel}}$
end while

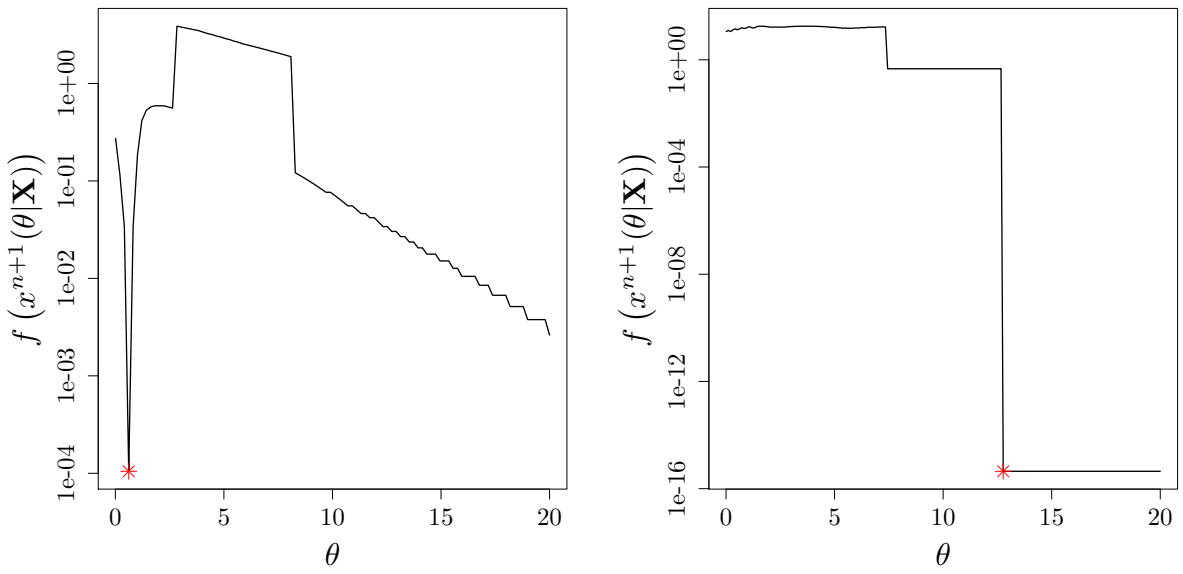


Figure 6.3: Function values of x^{n+1} already shown in Figure 6.2. The asterisk indicate the correlation length-scale, θ^*, which causes the maximum improvement in the objective function. In this example, θ^* is different from $\hat{\theta}_{ML}$, estimated by ML,: $\theta^* = 0.61271$ and $\hat{\theta}_{ML} = 5.34$ (Sphere; left), $\theta^* = 12.7674$ and $\hat{\theta}_{ML} = 0.01$ (Ackley; right), the lower bound on θ. Both functions have their global minimum at 2.5 and the DoE is $\mathbf{X} = \{-5, -2, 2, 5\}$.

We now analyze this approach in more details by providing some examples in $2D$. Figure 6.4 illustrates the first and the second iterations of this algorithm again on the Sphere and Ackley functions. In this Figure, the location of the points that maximize the expected improvement for different length-scale values is plotted on the top of the true function contour lines. In total, 64 length-scales, started from 0.01, are used. The length-scales are divided into eight groups. Each group consists of eight length-scales in ascending order. The ith group is denoted by $\theta^{(i:8)}, i = 1, \ldots, 8$ and is defined as $[0.01 + 8(i-1) \times \alpha_{\text{increment}}, 0.01 + 8i \times \alpha_{\text{increment}})$ where $\alpha_{\text{increment}} \approx 0.1$. The infill sample points obtained by the length-scales of a particular group have identical color, see the legend of Figure 6.4.

The first remark that can be done, and which motivates this study, is that the points visited as θ changes make a one dimensional manifold (obviously since it is parameterized by the scalar θ), continuous by parts and, most interestingly, often curved towards the global optimum of the function. The discontinuities of the trajectory are associated to

changes of basin of attraction during the maximization of the expected improvement. This simple observation, even though only based on a few cases, is a hint that the volume search of global optimization algorithms might be iteratively transformed into a one dimensional search in θ, with potentials for containing the "curse of dimensionality": most global optimization algorithm like EGO and evolution strategies undergo a geometric increase in search space volume as the number of dimensions increases; the current modified EGO always searches along a 1-dimensional curve. The difficulties of the associated problem and a possible implementation will be discussed in the next section.

In Figure 6.4, it can be seen that the magnitude of the "best" length-scale in the first iteration is between 2 and 3, i.e., $\theta^* \in \theta^{(3:8)}$ *or* $\theta^{(4:8)}$. While EGO with a small length-scale samples near the best observed point (cf. the black points), EGO with large length-scale is more explorative (see yellow and grey points) [MLRT15]. The search points and the length-scales obtained by the algorithm after 15 iterations are given in Figure 6.5. It can be observed that, after the first iterations where the "best" length-scale magnitude, θ^*, is of order 1, θ^* oscillates at usually small values. Because θ^* oscillates, self-adaptive strategies and Bayesian strategies based on assuming a prior density over the length-scale may not be a good strategy for optimization (at least if θ^* makes an efficient strategy).

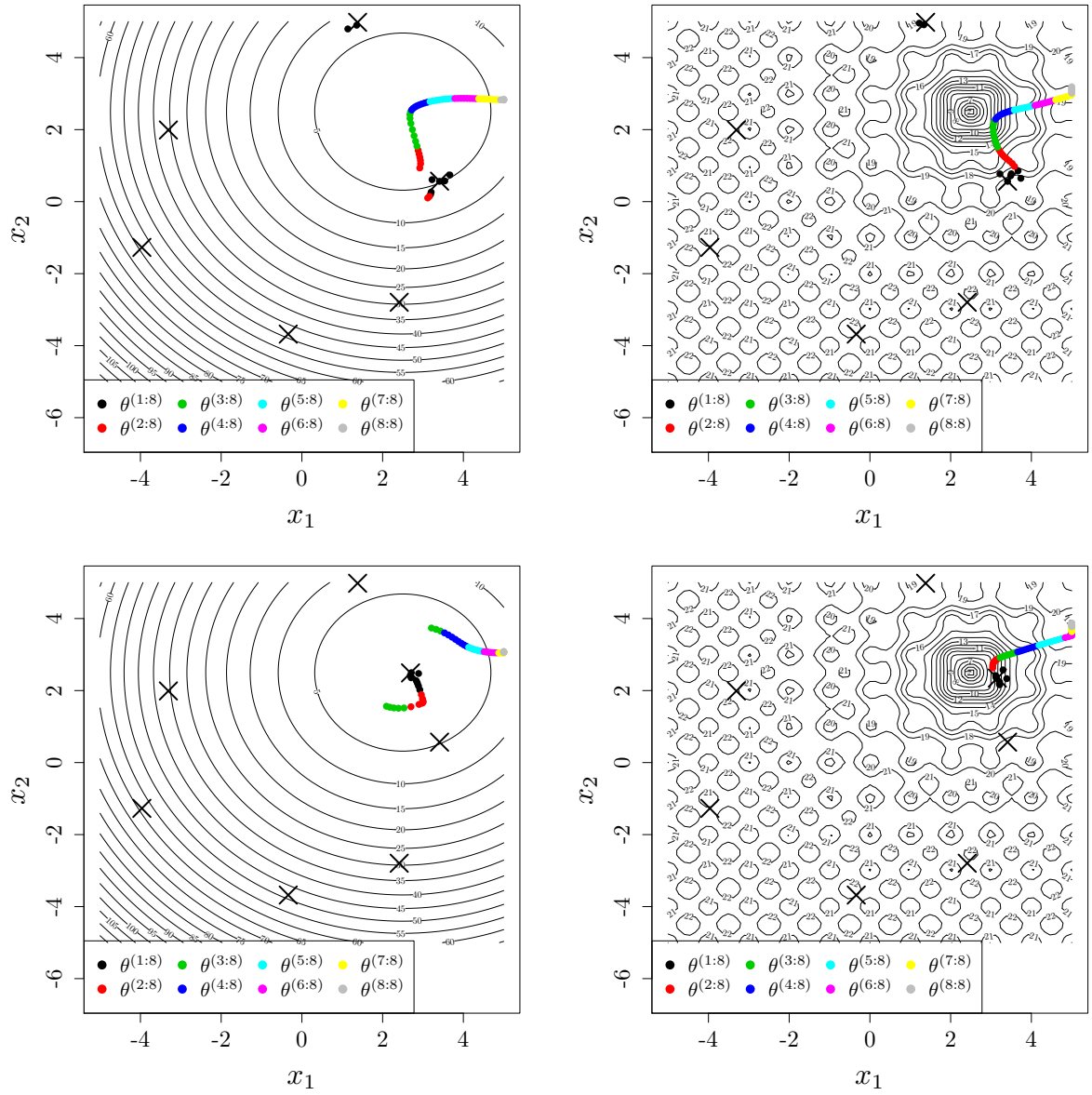


Figure 6.4: First (top row) and second (bottom row) iterations of EGO in which $\mathbf{x}^{n+1}(\theta^*|\mathbf{X}) = \arg\max_{x \in \mathcal{S}} EI(\mathbf{x}|\theta^*)$ is added to the existing DoE, the crosses, on the Sphere (left) and the Ackley (right) functions. 64 equally distant length-scales are grouped into eight equal sized intervals, $\theta^{(i:8)}, i = 1, \ldots, 8$. The infill sample points obtained by the length-scales of a particular group have identical color.

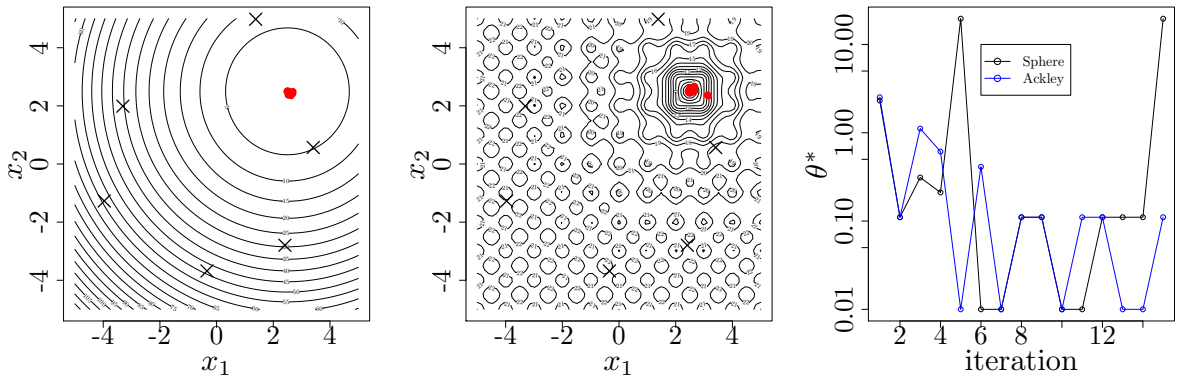


Figure 6.5: DoEs created by the toy greedy algorithm 6.1 after 15 iterations on the Sphere (left) and the Ackley (middle) functions. Right: plot of "best" length-scale, θ^*. θ^* oscillates during optimization iterations and usually has a small magnitude after the first iterations. The y-axis is in logarithmic scale.

In order to investigate the effect of initial DoE on the algorithm performance, the above experiments are repeated with another initial DoE. Figure 6.6 shows the results which are similar to the previous experiments. For example, the length-scales tend to be small especially in the case of highly multimodal Ackley function. The algorithm's behavior, typical of small θ's (as explained in details in [MLRT15]) is greedy, that of a local search algorithm: local convergences can be seen in Figure 6.8 where the function to be optimized is Rastrigin with several local minima.

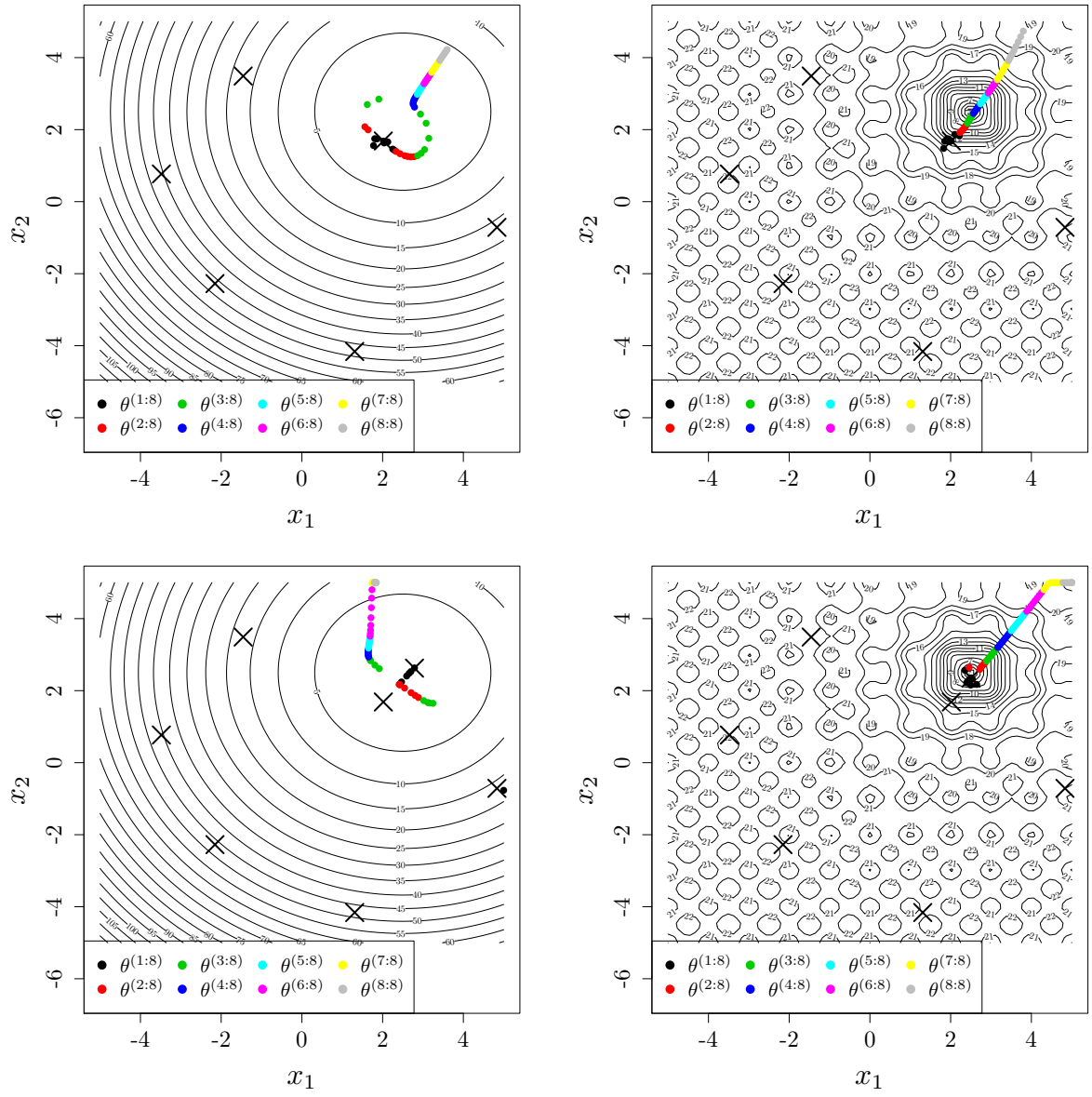


Figure 6.6: First (top row) and second (bottom row) iteration of the toy greedy algorithm 6.1 on the Sphere (left) and the Ackley functions(right). The initial DoE is different from the one shown in Figure 6.4. For more information see the caption of Figure 6.4.

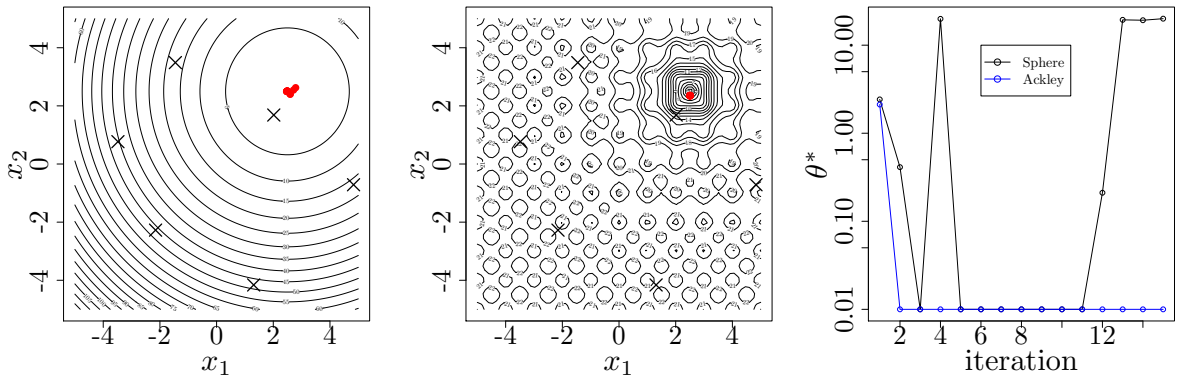


Figure 6.7: DoEs created by the toy greedy algorithm 6.1 after 15 iterations on the Sphere (left) and the Ackley (middle) functions. Right: plot of "best" length-scale, θ^*. The initial DoE is different from the one shown in Figure 6.5.

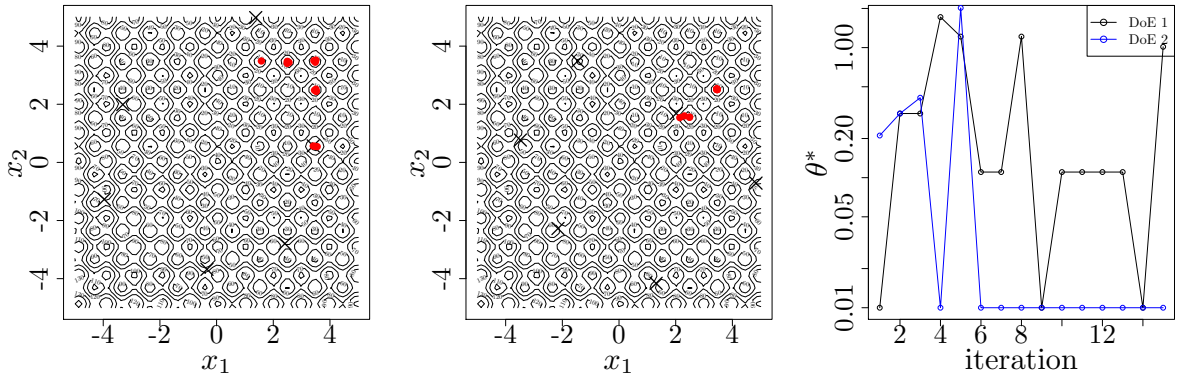


Figure 6.8: DoEs created by the toy greedy algorithm 6.1 after 15 iterations on the Rastrigin function with two DoEs (left and middle). Right: plot of "best" length-scale, θ^*. The global minimum is located at $(2.5, 2.5)$.

6.4 An EGO algorithm with a small ensemble of kriging models

6.4.1 Description of the algorithm

EGO is used for the optimization of computationally intensive functions. So, it is practically impossible to calculate $f\left(\mathbf{x}^{n+1}(\theta|\mathbf{X})\right)$ for many length-scales in order to obtain θ^*. Herein, we propose an approach that works with a limited number of kriging models. The

ensemble of kriging models is structured by the length-scales. The pseudo-code is given below (Algorithm 6.2) followed by a detailed explanation of the approach.

Algorithm 6.2 EGO based on a small ensemble of kriging models

Create an initial design: $\mathbf{X} = [\mathbf{x}^1, \ldots, \mathbf{x}^n]^\top$.
Evaluate function at $\mathbf{X}$ and set $\mathbf{y} = f(\mathbf{X})$.
Set the maximum number of evaluations, $t_{\max}$.
for t $\leftarrow$ n+1 **to** $t_{\max}$ **do**
  Define a neighborhood of radius $R^{(t)}$ around the current sample points.
  Set $\mathbf{X}^{(n+1)} = \emptyset$ and $\mathbf{X}^{\text{sel}} = \emptyset$.
  Generate q length-scales, $\theta_1, \ldots, \theta_q$.
  for i $\leftarrow$ 1 **to** q **do**
    $\mathbf{x}^{n+1} \leftarrow \arg\max_{\mathbf{x}\in\mathcal{S}} \; EI(\mathbf{x}; \theta_i)$.
    $\mathbf{X}^{(n+1)} \leftarrow \mathbf{X}^{(n+1)} \cup \mathbf{x}^{n+1}$.
    if $\mathbf{x}^{n+1}$ is not inside the defined neighborhoods **then**
      $\mathbf{X}^{\text{sel}} \leftarrow \mathbf{X}^{\text{sel}} \cup \mathbf{x}^{n+1}$.
    end if
  end for
  if $\mathbf{X}^{\text{sel}} = \emptyset$ **then**
    $\mathbf{X}^{\text{sel}} \leftarrow \arg\max_{\mathbf{x}\in\mathbf{X}^{(n+1)}} (\min dist(\mathbf{x}, \mathbf{X}))$
  end if
  Evaluate function at $\mathbf{X}^{\text{sel}}$ and set $\mathbf{y}^{\text{sel}} = f(\mathbf{X}^{\text{sel}})$.
  Select θ^*, for which $f(\arg\max_{\mathbf{x}\in\mathcal{S}} \; EI(\mathbf{x}; \theta^*)) = \min(\mathbf{y}^{\text{sel}})$.
  Generate two length-scales close to θ^*. This yields two new infill samples by EI maximization, $\mathbf{X}^{\text{new}} = [\mathbf{x}^{\text{new1}}, \mathbf{x}^{\text{new2}}]^\top$.
  Evaluate function at $\mathbf{X}^{\text{new}}$ and set $\mathbf{y}^{\text{new}} = f(\mathbf{X}^{\text{new}})$.
  Update the DoE: $\mathbf{X} \leftarrow \mathbf{X} \cup \mathbf{X}^{\text{sel}} \cup \mathbf{X}^{\text{new}}$, $\mathbf{y} \leftarrow \mathbf{y} \cup \mathbf{y}^{\text{sel}} \cup \mathbf{y}^{\text{new}}$.
end for

Let $(\mathbf{X}, \mathbf{y})$ be the initial design of experiments. The covariance function we use here is the isotropic Matérn 5/2 kernel [RW05]. Thus, there exists only one length-scale to be tuned. The first reason for using an isotropic kernel is simplicity and clarity in the analysis. By taking isotropic functions and kernels, a difficult aspect of the algorithm (anisotropy,

which is related to variables sensitivity) is neutralized to focus on other (also quite complex) phenomena. By taking isotropic kernels, the results of the numerical experiments are more stable. The second reason is that isotropic kernels have been found to perform well for EGO in high-dimension in the context of expensive-to-evaluate functions [HHLB13].

At each iteration, five length-scales are generated. They are sampled on a basis 10 logarithmic scale from $[-2, 1]$ based on a Latin Hypercube Sampling (LHS) plan (that is θ ranges from 10^{-2} to 10^{1}). Then, they are sorted and scaled back, $\theta_i = 10^{\log \theta_i}$, $1 \leq i \leq 5$; $\theta_1 < \theta_2 < \cdots < \theta_5$. Corresponding to each length-scale θ_i, a kriging model is created which gives a new infill sample: $\mathbf{x}^{n+1}(\theta_i|\mathbf{X}) = \arg\max_{\mathbf{x}\in\mathcal{S}} \ EI(\mathbf{x}; \theta_i)$.

In the next step, the $\mathbf{x}^{n+1}(\theta_i|\mathbf{X})$, $1 \leq i \leq 5$, that are not close to the design points are selected and the function is evaluated there. The notion of closeness is expressed by defining a neighborhood of radius $R^{(t)}$ around design points, see Figure 6.9. It is important to prevent the points from converging around early good performers, otherwise such greedy algorithm where decisions are taken solely on the account of objective function values would not be sufficiently explorative for global optimization. Further explanations about the neighborhood definition are provided in the next paragraph. The eligible $\mathbf{x}^{n+1}(\theta_i|\mathbf{X})$, $1 \leq i \leq 5$, are selected and stored in the matrix $\mathbf{X}^{\text{sel}}$. $\mathbf{y}^{\text{sel}}$ contains the function values at $\mathbf{X}^{\text{sel}}$.

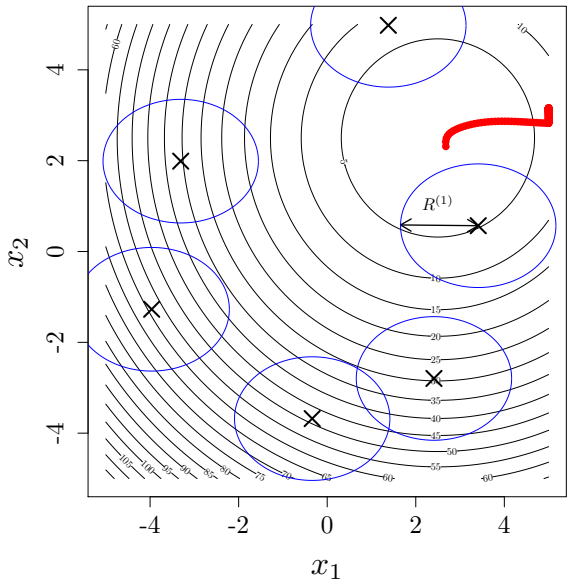


Figure 6.9: DoE and neighborhoods as balls around the design points (blue circles). The infill samples occurring inside any neighborhood are not considered by the optimizer.

The neighborhood defined around every design point is a ball with radius $R^{(t)}$ where the index t is the iteration. As the optimization progresses, the radius shrinks according to the following linear scheme:

$$R^{(t)} = \begin{cases} R^{(1)} - \frac{R^{(1)}}{t_{\text{threshold}}} \times (t-1) & \text{if } t \leq t_{\text{threshold}} \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

in which $t_{\text{threshold}}$ is 70% of total number of iterations, $t_{\max}$. The initial radius $R^{(1)}$, is half of the distance between the best initial DoE (based on its f value) and the closest design point to it. Again, defining such neighborhoods prevents the algorithm from focusing around good points too early.

Now, among the five generated length-scales, the best one is selected and is denoted by θ^*. Recall that the best length-scale is the one that yields $f\left(\mathbf{x}^{n+1}(\theta_i|\mathbf{X})\right) = \min(\mathbf{y}^{\text{sel}})$. Then, two length-scales, θ^*_- and θ^*_+, close to θ^* are generated. They are defined as:

- If $\theta^* = \theta_i, 2 \leq i \leq 4$, $\theta^*_- = \theta^* - \frac{1}{3}(\theta^* - \theta_{i-1})$ and $\theta^*_+ = \theta^* + \frac{1}{3}(\theta_{i+1} - \theta^*)$.
- If $\theta^* = \theta_1$, $\theta^*_- = 0.01$ and $\theta^*_+ = \theta^* + \frac{1}{3}(\theta_2 - \theta^*)$.
- If $\theta^* = \theta_5$, $\theta^*_- = \theta^* - \frac{1}{3}(\theta^* - \theta_4)$ and $\theta^*_+ = 10$.

The two new infill samples obtained with the kriging models with length-scales θ^*_- and θ^*_+ are stored in the $\mathbf{X}^{new}$ matrix,

$$\mathbf{X}^{\text{new}} = \left[\mathbf{x}^{n+1}(\theta^*_-|\mathbf{X}) \ , \ \mathbf{x}^{n+1}(\theta^*_+|\mathbf{X})\right]^\top . \tag{4}$$

Finally, the current DoE ($\mathbf{X}$, $\mathbf{y}$) is updated by adding $\mathbf{X}^{\text{new}}$ and $\mathbf{X}^{\text{sel}}$ to $\mathbf{X}$ and $\mathbf{y}^{\text{new}}$ and $\mathbf{y}^{\text{sel}}$ to $\mathbf{y}$. This procedure continuous until the budget is exhausted.

6.4.2 Tests of the algorithm

The performance of this EGO method that is based on a small ensemble of kriging models (5+2 models) is tested on three isotropic functions, Sphere, Ackley and Rastrigin. The functions are defined in $\mathcal{S} = [-5, 5]^d$ where $d = 5$. The total number of iterations is $15 \times d$. Each optimization run is repeated eight times (thin black lines). Figure 6.10 shows the results and the performance of the standard EGO method (thin blue lines) which is repeated five times with a budget equal to $70 \times d$. The plots show the best objective

functions observed so far. The initial DoE is fixed for both algorithms and has a size equal to $3 \times d$. The thick lines are the median of the runs.

The small ensemble version of EGO is slightly better on the sphere function because it benefits from its greedy choice of points that are never misleading. On Rastrigin and Ackley, the small ensemble EGO is slower early in the search, which might be due to the schedule of $R^{(t)}$: because $R^{(t)}$ is large at the beginning of the search, the algorithm cannot be greedy early on. Later on, still on Rastrigin and Ackley, EGO with a small ensemble shows both the worst and best performances, therefore illustrating a tendency to get trapped in local optima. In terms of median performance, after 250 evaluations of the objective function (at the time when the neighborhood control ceases), the small ensemble EGO is equivalent to EGO on Rastrigin and worse on Ackley.

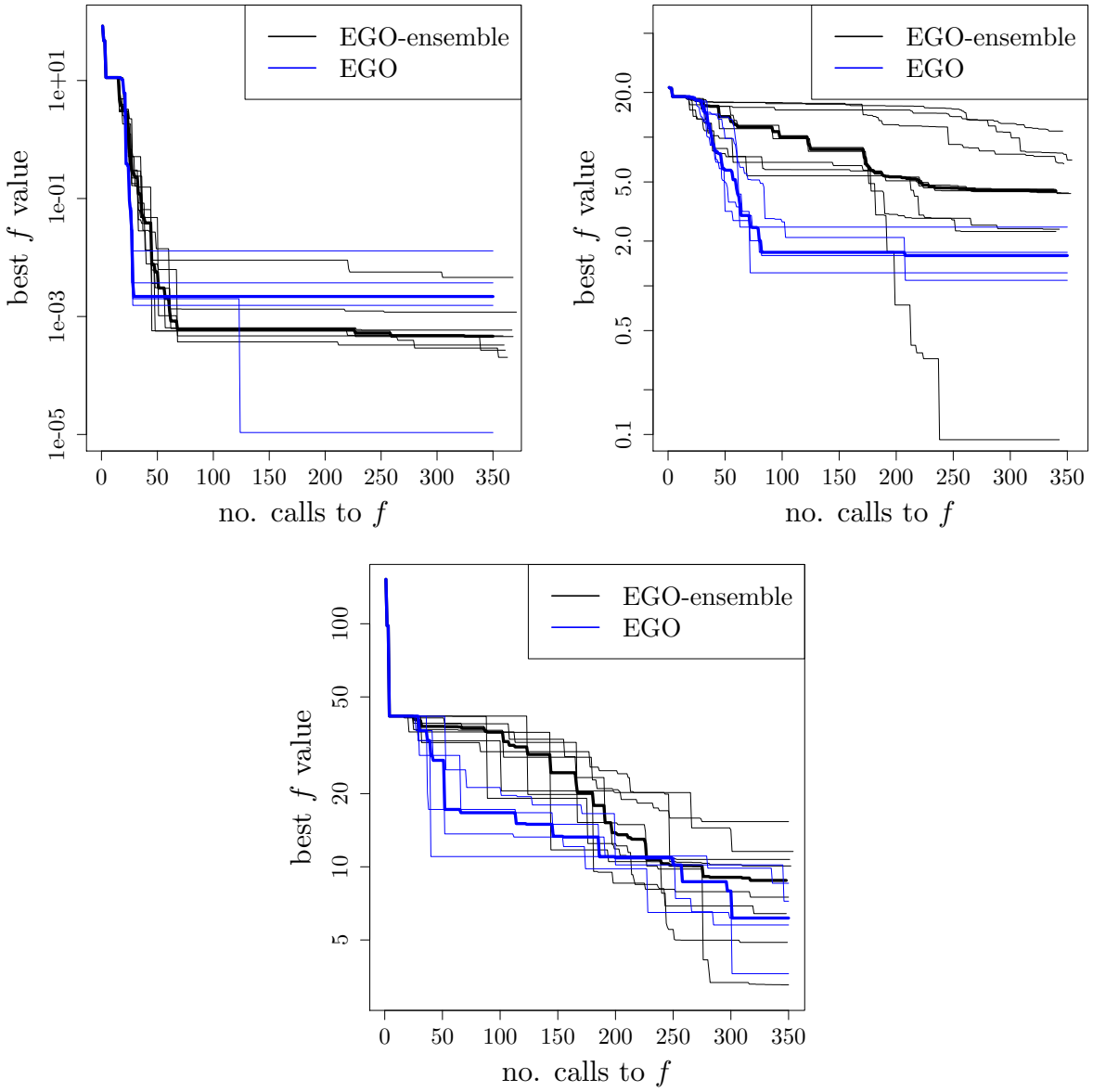


Figure 6.10: Best objective function vs. number of calls of EGO with the ensemble of kriging models (thin black lines) and standard EGO (thin blue lines) on Sphere(top left), Ackley (top right) and Rastrigin (bottom) functions. The thick lines show the median of the runs.

6.5 Conclusions

We have investigated a variant of the EGO optimization algorithm where, instead of using at each iteration a kriging model learned through a statistical estimation procedure such

as maximum likelihood, a small set of models with different (adapted) length-scale is employed. The motivations are threefold. Firstly, it has been noticed in two-dimensions that the manifolds of the points that maximize expected improvement for various length-scales approach rapidly the global optimum. Secondly, ensemble methods have a lower computational complexity since the number of kriging covariance matrices inversions is limited to the number of elements in the ensemble, seven in the current work. On the contrary, maximum likelihood or cross-validation approaches require the inversion of the covariance matrix at each of their internal iteration. Thirdly, ensemble methods may more easily lead to parallel versions of EGO as the maximization of expected improvement can be distributed on several computing nodes, one for each kriging model.

Our first investigations have led to the following conclusions: tuning the length-scale to achieve an immediate improvement in the objective function may not be as efficient a strategy as two-dimensional plots of the manifold seem to indicate; the greediness of the method is a source of premature convergence to good performing points; optimal values of the length scale (in the sense of short term improvement) change a lot from one iteration to the next as the design of experiments evolves, rendering self-adaptive and Bayesian strategies not efficient for this purpose.

Nevertheless, we believe that the idea of searching in the space of length-scales as a proxy for searching in the space of optimization variables deserves further investigations because of its potential for tackling the curse of dimensionality. In particular, the schedule of the neighborhood radius, an iteration-smoothing learning procedure for the length-scales, and alternative strategies for making the ensemble of kriging should be studied.

Chapter 7

Conclusions and perspectives

This thesis contributes to the field of Gaussian process-based optimization. More precisely, we have addressed the following issues:

(I) The non-invertibility of covariance matrix in Gaussian process (GP) modeling;

(II) The comparison and complementarity between the stochastic algorithm CMA-ES and EGO;

(III) The mode of convergence of the EGO algorithm in relation to the kernel parameters;

(IV) Methods alternative to statistical learning for tuning the kernel parameters during an EGO search.

The problem (I) was addressed in Chapter 3. In this chapter, we have provided a new algebraic comparison of pseudoinverse and nugget regularizations, two classical solutions to overcome the degeneracy of the covariance matrix in GPs. We have proved that, when the covariance matrix is regularized by pseudoinverse, the Gaussian process mean averages the outputs and its variance is zero at redundant points. In the case of nugget regularization, the discrepancy between model and data translates into a departure of the GP from observation points throughout the domain. We have also proposed a new regularization approach called distribution-wise GP model in which normal distributions are interpolated instead of data points. This approach unlike nugget and pseudoinverse regularizations averages the outputs at redundant points and preserves the redundant points variances.

The problem (II) was addressed in Chapter 4 by introducing a new algorithm called EGO-CMA. EGO-CMA combines the strengths of EGO and CMA-ES. EGO is a space

filling algorithm and is used first to explore the search space. Then CMA-ES, as a converging search, is started from the best point obtained by EGO to converge towards the global optimum with high accuracy. Moreover, we have proposed a warm-start for both the step-size and the covariance matrix of CMA-ES. The performance of EGO-CMA was compared with that of EGO and CMA-ES. EGO-CMA had better performance in our experiments.

The question (III) was answered in Chapter 5 by performing experiments to study the effect of kernel parameters on the EGO performance. We have carefully explained the design of experiments generated by EGO when the kernel parameters are fixed. To do so, we have isolated two simple landscapes where EGO behaves differently. On purpose, one function is unimodal (Sphere), the other multimodal (Ackley). The limit cases of small and large length-scales have been mathematically analyzed. This study provided a solid understanding of the EGO behavior that allows further improvement of this algorithm.

The problem (IV) was addressed in Chapter 6 by introducing a variant of the EGO optimization algorithm. At each iteration of this algorithm, instead of learning the length-scales by statistical techniques, a small ensemble of kriging models structured by their length-scales is created. Then, the model whose length-scale yielded the best iterate is selected and further points are produced through intensifying around the selected model. Encouraging observations have been made in two dimensions. In addition, ensemble methods have a lower computational complexity than statistical learning approaches. Yet, the proposed algorithm did not beat the traditional EGO on multi-modal functions.

The work described in this manuscript opens the way to many further investigations.

The distribution-wise GP model introduced in Chapter 3 can be used in EGO with high number of data points. In the sequential design created by EGO it is common that some sample points tend to pile up near local optima. A possible algorithm would, therefore, be to cluster these points and consider them as repeated points. By this approach, the size of the covariance matrix shrinks.

The EGO-CMA algorithm introduced in Chapter 4 can be implemented in such a way to make a multi-start CMA-ES possible. For example, among the DoEs created by EGO, one can select the best, say 10, design points which are “far away”. Then, each of these points serve as an initial point of CMA-ES.

As a continuation of Chapter 5, one should study dynamic EGO strategies where the length-scales vary in time, starting with a large length-scale and then decreasing it. This, again, would be an alternative to statistical learning procedures, such as maximum likelihood estimation which requires $O(n^3)$ computations where n is the number of data points. This computation cost is not negligible when the number of data points is high as it may result in minutes to hours of computation on a standard machine.

List of Figures

List of Tables

List of Algorithms

Bibliography

[AC12] Ioannis Andrianakis and Peter G. Challenor. The effect of the nugget on Gaussian process emulators of computer models. *Computational Statistics & Data Analysis*, 56(12):4215–4228, 2012.

[AH05] Anne Auger and Nikolaus Hansen. A restart CMA evolution strategy with increasing population size. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 2, pages 1769–1776, Piscataway, NJ, USA, 2005. IEEE Press.

[AJ02] Charles Audet and J. E. Dennis Jr. Analysis of generalized pattern searches. *SIAM Journal on Optimization*, 13(3):889–903, 2002.

[Aro50] Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68, 1950.

[ARR09] Erdem Acar and Masoud Rais-Rohani. Ensemble of metamodels with optimized weight factors. *Structural and Multidisciplinary Optimization*, 37(3):279–294, 2009.

[B$\ddot{9}$6] Thomas Bäck. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, Oxford, UK, 1996.

[Bac13] François Bachoc. Cross validation and maximum likelihood estimations of hyper-parameters of Gaussian processes with model misspecification. *Computational Statistics & Data Analysis*, 66:55–69, 2013.

[BBV11] Romain Benassi, Julien Bect, and Emmanuel Vazquez. Robust gaussian process-based global optimization using a fully bayesian expected improvement criterion. In CarlosA.Coello Coello, editor, *Learning and Intelligent Optimization*, volume 6683 of *Lecture Notes in Computer Science*, pages 176–190. Springer Berlin Heidelberg, 2011.

[BCdF09] Eric Brochu, Vlad M. Cora, and Nando de Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. Technical Report TR-2009-23, Department of Computer Science, University of British Columbia, November 2009.

[BDJ+98] Andrew J. Booker, J. E. Dennis, Jr., Paul D. Frank, David B. Serafini, Virginia Torczon, and Michael Trosset. A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization*, 17(17):1–13, 1998.

[Bel91] David A. Belsley. *Conditioning Diagnostics: Collinearity and Weak Data in Regression*. Wiley series in probability and mathematical statistics. Wiley, New York, 1991.

[BGL+12] Julien Bect, David Ginsbourger, Ling Li, Victor Picheny, and Emmanuel Vázquez. Sequential design of computer experiments for the estimation of a probability of failure. *Statistics and Computing*, 22(3):773–793, 2012.

[BIC66] Adi Ben-Israel and Dan Cohen. On iterative computation of generalized inverses and associated projections. *SIAM Journal on Numerical Analysis*, 3(3):410–419, 1966.

[Bis95] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., New York, NY, USA, 1995.

[BSK04] Dirk Bueche, Nicol N. Schraudolph, and Petros Koumoutsakos. Accelerating evolutionary algorithms with Gaussian process fitness function models. *IEEE Transactions on Systems, Man and Cybernetics*, 35:183–194, 2004.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[CH13] Anirban Chaudhuri and Raphael T Haftka. A stopping criterion for surrogate based optimization using ego. In *10th World Congress on Structural and Multidisciplinary Optimization*, 2013.

[CH14] Anirban Chaudhuri and Raphael T Haftka. Efficient global optimization with adaptive target setting. *AIAA*, 52(7):1573–1578, 2014.

[CJ08] Noel Cressie and Gardar Johannesson. Fixed rank kriging for very large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):209–226, 2008.

[CLRM13] Anirban Chaudhuri, Rodolphe Le Riche, and Mickael Meunier. Estimating Feasibility Using Multiple Surrogates and ROC Curves. In *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Boston, France, April 2013.

[Cre93] Noel A. C. Cressie. *Statistics for spatial data*. Wiley series in probability and mathematical statistics. J. Wiley & Sons, New York, Chichester, Toronto, 1993.

[dBKMR05] Pieter-Tjerk de Boer, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein. A tutorial on the cross-entropy method. *Annals of Operations Research*, 134(1):19–67, 2005.

[DGR12] Nicolas Durrande, David Ginsbourger, and Olivier Roustant. Additive covariance kernels for high-dimensional Gaussian process modeling. *Annales de la Faculté de Sciences de Toulouse*, Tome 21(numéro 3):481–499, 2012.

[DHF15] Delphine Dupuy, Celine Helbert, and Jessica Franco. DiceDesign and DiceEval: two R packages for design and analysis of computer experiments. *Journal of Statistical Software*, 65(11):1–38, 2015.

[DK10] Keith R. Dalbey and George N. Karystinos. Fast generation of space-filling latin hypercube sample designs. In *13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, 2010.

[DM97] George Davis and Max Morris. Six factors which affect the condition number of matrices associated with kriging. *Mathematical Geology*, 29(5):669–683, 1997.

[DNR11] David K Duvenaud, Hannes Nickisch, and Carl E. Rasmussen. Additive Gaussian processes. In *Advances in Neural Information Processing Systems 24*, pages 226–234. Curran Associates, Inc., 2011.

[FB08] Marcus R. Frean and Phillip Boyle. Using Gaussian processes to optimize expensive functions. In *AI 2008: Advances in Artificial Intelligence, 21st Australasian Joint Conference on Artificial Intelligence, Auckland, New Zealand, December 1-5, 2008. Proceedings*, pages 258–267, 2008.

[FJ08] Alexander I.J. Forrester and Donald R. Jones. Global optimization of deceptive functions with sparse sampling. In *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. American Institute of Aeronautics and Astronautics, 2008.

[FK09] Alexander I.J. Forrester and Andy J. Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1-3):50–79, 2009.

[FSK08] Alexander Forrester, Andras Sobester, and Andy Keane. *Engineering design via surrogate modelling: a practical guide*. Wiley, 2008.

[Gen02] Marc G. Genton. Classes of kernels for machine learning: A statistics perspective. *Journal of Machine Learning Research*, 2:299–312, 2002.

[GHSQ07] Tushar Goel, RaphaelT. Haftka, Wei Shyy, and NestorV. Queipo. Ensemble of surrogates. *Structural and Multidisciplinary Optimization*, 33(3):199–216, 2007.

[Gib97] Mark Gibbs. *Bayesian Gaussian Processes for Regression and Classification*. PhD thesis, University of Cambridge, 1997.

[GL09] Robert B. Gramacy and Herbert K. Lee. Adaptive design and analysis of supercomputer experiments. *Technometrics*, 51(2):130–144, 2009.

[GL12] Robert B. Gramacy and Herbert K. Lee. Cases for the nugget in modeling computer experiments. *Statistics and Computing*, 22(3):713–722, 2012.

[GLRC10] David Ginsbourger, Rodolphe Le Riche, and Laurent Carraro. *Computational Intelligence in Expensive Optimization Problems*, chapter Kriging Is Well-Suited to Parallelize Optimization, pages 131–162. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[HAFR09] N. Hansen, A. Auger, S. Finck, and R. Ros. Real-parameter black-box optimization benchmarking 2009: Experimental setup. Technical Report RR-6828, INRIA, 2009.

[Hal60] J. H. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2(1):84–90, 1960.

[Han07] Christian Hansen. Generalized least squares inference in panel and multilevel models with serial correlation and fixed effects. *Journal of Econometrics*, 140(2):670–694, 2007.

[Han09a] Nikolaus Hansen. Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed. In *Workshop Proceedings of the GECCO Genetic and Evolutionary Computation Conference*, pages 2389–2395. ACM, 2009.

[Han09b] Nikolaus Hansen. *The CMA Evolution Strategy: A Tutorial*, 2009.

[HAR+10] Nikolaus Hansen, Anne Auger, Raymond Ros, Steffen Finck, and Petr Pošík. Comparing results of 31 algorithms from the black-box optimization benchmarking bbob-2009. In *Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation*, GECCO '10, pages 1689–1696, New York, NY, USA, 2010.

[HFRA09] Nikolaus Hansen, Steffen Finck, Raymond Ros, and Anne Auger. Real-Parameter Black-Box Optimization Benchmarking 2009: Noiseless Functions Definitions. Research Report RR-6829, INRIA, 2009.

[HHLB13] Frank Hutter, Holger Hoos, and Kevin Leyton-Brown. An evaluation of sequential model-based optimization for expensive blackbox functions. In *Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation*, GECCO '13 Companion, pages 1209–1216, New York, NY, USA, 2013. ACM.

[HK04] Nikolaus Hansen and Stefan Kern. Evaluating the CMA evolution strategy on multimodal test functions. In *Parallel Problem Solving from Nature - PPSN VIII*, volume 3242 of *Lecture Notes in Computer Science*, pages 282–291. Springer Berlin Heidelberg, 2004.

[HO96] Nicolaus Hansen and Andreas Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. In *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 312 –317, 1996.

[HO01] Nicolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[HOG95] Nikolaus Hansen, Andreas Ostermeier, and Andreas Gawelczyk. On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation. In *Sixth International Conference on Genetic Algorithms*, pages 312–317. Morgan Kaufmann, 1995.

[Hol75] John Henry Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, USA, 1975.

[HS04] Holger Hoos and Thomas Sttzle. *Stochastic Local Search: Foundations & Applications*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[HWB13] Stefan Hess, Tobias Wagner, and Bernd Bischl. *Learning and Intelligent Optimization: 7th International Conference, LION 7, Catania, Italy, January 7-11, 2013, Revised Selected Papers*, chapter PROGRESS: Progressive Reinforcement-Learning-Based Surrogate Selection, pages 110–124. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[Jan13] Janis Janusevskis. KRISP: KRIging based regression and optimization Scilab Package. `https://atoms.scilab.org/toolboxes/krisp/`, 2013.

[Jin11] Yaochu Jin. Surrogate-Assisted Evolutionary Computation: Recent Advances and Future Challenges. *Swarm and Evolutionary Computation*, 1(2):61–70, 2011.

[Jon01] Donald R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21:345–383, 2001.

[JPS93] D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.

[JR13] Janis Janusevskis and Rodolphe Le Riche. Simultaneous kriging-based estimation and optimization of mean response. *Journal of Global Optimization*, 55(2):313–336, 2013.

[JS04] Yaochu Jin and Bernhard Sendhoff. *Genetic and Evolutionary Computation – GECCO 2004: Genetic and Evolutionary Computation Conference, Seattle, WA, USA, June 26-30, 2004. Proceedings, Part I*, chapter Reducing Fitness Evaluations Using Clustering Techniques and Neural Network Ensembles, pages 688–699. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[JSW98] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

[KEDB10] Johannes W. Kruisselbrink, Michael T. M. Emmerich, Andre H. Deutz, and Thomas Back. A robust optimization approach using kriging metamodels for

robustness approximation in the CMA-ES. In *IEEE Congress on Evolutionary Computation*, pages 1–8. IEEE, 2010.

[KGV83] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[KHK06] Stefan Kern, Nikolaus Hansen, and Petros Koumoutsakos. Local meta-models for optimization using evolution strategies. In *Parallel Problem Solving from Nature - PPSN IX*, pages 939–948. Springer, 2006.

[Kle14] Jack P. C. Kleijnen. Simulation-optimization via kriging and bootstrapping: a survey. *J. Simulation*, 8(4):241–250, 2014.

[Kri53] D. G. Krige. A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand. *OR*, 4(1), 1953.

[Kru68] William Kruskal. When are Gauss-Markov and least squares estimators identical? a coordinate-free approach. *The Annals of Mathematical Statistics*, 39(1):70–75, 1968.

[LGS12] Daniel J. Lizotte, Russell Greiner, and Dale Schuurmans. An experimental methodology for response surface optimization methods. *Journal of Global Optimization*, 53(4):699–736, 2012.

[LLJ13] Jianfeng Lu, Bin Li, and Yaochu Jin. An evolution strategy assisted by an ensemble of local gaussian process models. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, GECCO '13, pages 447–454, New York, NY, USA, 2013. ACM.

[LLR04] Marco A. Luersen and Rodolphe Le Riche. Globalized Nelder-Mead method for engineering optimization. *Computers & Structures*, 82(23-26):2251–2260, 2004.

[Loc97] M. Locatelli. Bayesian algorithms for one-dimensional global optimization. *Journal of Global Optimization*, 10(1):57–76, 1997.

[Los13] Ilya Loshchilov. *Surrogate-Assisted Evolutionary Algorithms*. Theses, Université Paris Sud - Paris XI ; Institut national de recherche en informatique et en automatique - INRIA, January 2013.

[LS05] Runze Li and Agus Sudjianto. Analysis of computer experiments using penalized likelihood in Gaussian kriging models. *Technometrics*, 47(2), 2005.

[LSS12] Ilya Loshchilov, Marc Schoenauer, and Michele Sebag. Self-adaptive surrogate-assisted covariance matrix adaptation evolution strategy. In *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, GECCO '12, pages 321–328, New York, NY, USA, 2012. ACM.

[LSS13] Ilya Loshchilov, Marc Schoenauer, and Michele Sebag. Intensive Surrogate Model Exploitation in Self-adaptive Surrogate-assisted CMA-ES (saACM-ES). In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 439–446. ACM Press, 2013.

[MBC00] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61, 2000.

[MLRD+16] Hossein Mohammadi, Rodolphe Le Riche, Nicolas Durrande, Eric Touboul, and Xavier Bay. An analytic comparison of regularization methods for Gaussian Processes. Research report, Ecole Nationale Supérieure des Mines de Saint-Etienne ; LIMOS, January 2016.

[MLRT15] Hossein Mohammadi, Rodolphe Le Riche, and Eric Touboul. A detailed analysis of kernel parameters in Gaussian process-based optimization. Technical report, Ecole Nationale Supérieure des Mines ; LIMOS, 2015.

[Mon01] Douglas C. Montgomery. *Design and analysis of experiments*. John Wiley & Sons, New York / Chichester, 5th edition, 2001.

[MTZ78] Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. The application of bayesian methods for seeking the extremum. *Towards Global Optimization*, 2(117-129):2, 1978.

[Nea97] Radford M. Neal. *Monte Carlo Implementation of Gaussian Process Models for Bayesian Regression and Classification.* Technical report (University of Toronto .Dept. of Statistics). University of Toronto, 1997.

[NM65] John A. Nelder and Roger. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[OGR09] Michael A. Osborne, Roman Garnett, and Stephen J. Roberts. Gaussian processes for global optimization. In *Proceedings of the 3rd Learning and Intelligent OptimizatioN Conference (LION 3)*, 2009.

[OO02] Jeremy E. Oakley and Anthony O'Hagan. Probabilistic sensitivity analysis of complex models: A bayesian approach. *Journal of the Royal Statistical Society, Series B*, 66:751–769, 2002.

[OZL06] Yew-Soon Ong, Zongzhao Zhou, and Dudy Lim. Curse and blessing of uncertainty in evolutionary algorithm using approximation. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 2928–2935, 2006.

[Pep10] Andrey Pepelyshev. The role of the nugget term in the Gaussian process method. In *mODa 9 – Advances in Model-Oriented Design and Analysis*, pages 149–156. Physica-Verlag HD, 2010.

[PH12] Petr Pošík and Waltraud Huyer. Restarted local search algorithms for continuous black box optimization. *Evol. Comput.*, 20(4):575–607, December 2012.

[Pow02] M.J.D. Powell. Uobyqa: unconstrained optimization by quadratic approximation. *Mathematical Programming*, 92(3):555–582, 2002.

[Pow09] M. J. D. Powell. The BOBYQA algorithm for bound constrained optimization without derivatives. Technical report, University of Cambridge, 2009.

[PT71] H. D. Patterson and R. Thompson. Recovery of Inter-Block information when block sizes are unequal. *Biometrika*, 58(3):545, December 1971.

[PWG13] Victor Picheny, Tobias Wagner, and David Ginsbourger. A benchmark of kriging-based infill criteria for noisy optimization. *Structural and Multidisciplinary Optimization*, 48(3):607–626, 2013.

[QHS+05] Nestor V. Queipo, Raphael T. Haftka, Wei Shyy, Tushar Goel, Rajkumar Vaidyanathan, and P. Kevin Tucker. Surrogate-based analysis and optimization. *Progress in Aerospace Sciences*, 41(1):1 – 28, 2005.

[QVPH09] Nestor V. Queipo, Alexander Verde, Salvador Pintos, and Raphael T. Haftka. Assessing the value of another cycle in gaussian process surrogate-based optimization. *Structural and Multidisciplinary Optimization*, 39(5):459–475, 2009.

[Ren09] Gijs Rennen. Subset selection from large datasets for kriging modeling. *Structural and Multidisciplinary Optimization*, 38(6):545–569, 2009.

[RGD12] Olivier Roustant, David Ginsbourger, and Yves Deville. DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization. *Journal of Statistical Software*, 51(1):1–55, 2012.

[RHK11] Pritam Ranjan, Ronald Haynes, and Richard Karsten. A computationally stable approach to Gaussian process interpolation of deterministic computer simulation data. *Technometrics*, 53(4):366–378, 2011.

[RKL14] David M. Rosen, Michael Kaess, and John J. Leonard. RISE: An incremental trust-region method for robust online sparse least-squares estimation. *IEEE Transactions on Robotics*, 30(5):1091–1108, 2014.

[RPD98] John O. Rawlings, Sastry G. Pantula, and David A. Dickey. *Applied Regression Analysis: A Research Tool*. Springer Texts in Statistics. Springer, 1998.

[RS13] Luis Rios and Nikolaos Sahinidis. Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56(3):1247–1293, 2013.

[Rud76] Walter Rudin. *Principles of mathematical analysis*. McGraw-Hill Book Co., New York, third edition, 1976. International Series in Pure and Applied Mathematics.

[Rud92] Gunter Rudolph. On correlated mutations in evolution strategies. In *Proceedings of the 2nd Conference on Parallel Problem Solving from Nature*, pages 107–116. North-Holland, Amsterdam, 1992.

[RW05] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. The MIT Press, 2005.

[Sas02] Michael J. Sasena. *Flexibility and Efficiency Enhancements For Constrainted Global Design Optimization with Kriging Approximations*. PhD thesis, University of Michigan, 2002.

[SB97] Raviprakash Salagame and Russell Barton. Factorial hypercube designs for spatial correlation regression. *Journal of Applied Statistics*, 24(4):453–474, 1997.

[Sch97] Matthias Schonlau. *Computer experiments and global optimization*. PhD thesis, Waterloo, Ont., Canada, Canada, 1997.

[SD98] Michèle Sebag and Antoine Ducoulombier. Extending population-based incremental learning to continuous search spaces. In *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature*, PPSN V, pages 418–427, London, UK, UK, 1998. Springer-Verlag.

[Shu72] Bruno O. Shubert. A sequential method seeking the global maximum of a function. *SIAM Journal on Numerical Analysis*, 9(3):379–388, 1972.

[SLK04] András Sóbester, Stephen J. Leary, and Andy J. Keane. A parallel updating scheme for approximating and optimizing high fidelity computer simulations. *Structural and Multidisciplinary Optimization*, 27(5):371–383, July 2004.

[SLK05] András Sóbester, Stephen J. Leary, and Andy J. Keane. On the design of optimization strategies based on global response surface approximation models. *Journal of Global Optimization*, 33(1):31–59, 2005.

[Sob67] I.M Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86 – 112, 1967.

[SS04] Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

[Str88] Gilbert Strang. *Linear Algebra and Its Applications*. Brooks Cole, 1988.

[Str09] Gilbert Strang. *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 2009.

[STT00] Christopher Siefert, Virginia Torczon, and Michael W. Trosset. MAPS: Model-assisted pattern search, 1997–2000. www.cs.wm.edu/∼va/software/maps/.

[SWMW89] Jerome Sacks, William J. Welch, Toby J. Mitchell, and Henry P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):433–435, 1989.

[SWN03] Thomas J. Santner, Brian J. Williams, and William Notz. *The Design and Analysis of Computer Experiments*. Springer series in statistics. Springer-Verlag, New York, 2003.

[TCR15] Sam Davanloo Tajbakhsh, Enrique Castillo, and James L Rosenberger. A Bayesian approach to sequential optimization based on computer experiments. *Quality and Reliability Engineering International*, 31(6):1001–1012, 2015.

[USZ03] Holger Ulmer, Felix Streichert, and Andreas Zell. Evolution strategies assisted by Gaussian processes with improved pre-selection criterion. In *Proceedings of the 2003 Congress on Evolutionary Computation CEC2003, Canberra, Australia*, pages 692–699. IEEE Press, 2003.

[VHW13] Felipe A.C. Viana, Raphael T. Haftka, and Layne T. Watson. Efficient global optimization algorithm assisted by multiple surrogate techniques. *Journal of Global Optimization*, 56(2):669–689, 2013.

[VVW08] Julien Villemonteix, Emmanuel Vazquez, and Eric Walter. An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization*, 44(4):509–534, 2008.

[VVW09] Julien Villemonteix, Emmanuel Vazquez, and Eric Walter. An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization*, 44(4):509–534, 2009.

[Wag10] Tobias Wagner. A subjective review of the state of the art in model-based parameter tuning. In *the Workshop on Experimental Methods for the Assessment of Computational Systems (WEMACS)*, 2010.

[Woo50] Max A. Woodbury. *Inverting Modified Matrices*. Number 42 in Statistical Research Group Memorandum Reports. Princeton University, Princeton, NJ, 1950.

[WZH+13] Ziyu Wang, Masrour Zoghi, Frank Hutter, David Matheson, and Nando de Freitas. Bayesian optimization in high dimensions via random embeddings. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, 2013.

[Yin91] Zhiliang Ying. Asymptotic properties of a maximum likelihood estimator with data from a Gaussian process. *Journal of Multivariate Analysis*, 36(2):280 – 296, 1991.

[YNN11] J. Yin, S.H. Ng, and K.M. Ng. Kriging metamodel with modified nugget-effect: The heteroscedastic variance case. *Computers & Industrial Engineering*, 61(3):760 – 777, 2011.

[ZW10] Hao Zhang and Yong Wang. Kriging and cross-validation for massive spatial data. *Environmetrics*, 21(3-4):290–304, 2010.

École Nationale Supérieure des Mines de Saint-Étienne

NNT : 2016LYSEM005

Hossein Mohammadi

Kriging-based black-box global optimization: analysis and new algorithms

Speciality: Applied Mathematics

Keywords: Kriging, Gaussian processes, EGO, CMA-ES, Global Optimization

Abstract:

The Efficient Global Optimization (EGO) is regarded as the state-of-the-art algorithm for global optimization of costly black-box functions. Nevertheless, the method has some difficulties such as the ill-conditioning of the GP covariance matrix and the slow convergence to the global optimum. The choice of the parameters of the GP is critical as it controls the functional family of surrogates used by EGO. The effect of different parameters on the performance of EGO needs further investigation. Finally, it is not clear that the way the GP is learned from data points in EGO is the most appropriate in the context of optimization.

This work deals with the analysis and the treatment of these different issues. Firstly, this dissertation contributes to a better theoretical and practical understanding of the impact of regularization strategies on GPs and presents a new regularization approach based on distribution-wise GP. Moreover, practical guidelines for choosing a regularization strategy in GP regression are given. Secondly, a new optimization algorithm is introduced that combines EGO and CMA-ES which is a global but converging search. The new algorithm, called EGO-CMA, uses EGO for early exploration and then CMA-ES for final convergence. EGO-CMA improves the performance of both EGO and CMA-ES. Thirdly, the effect of GP parameters on the EGO performance is carefully analyzed. This analysis allows a deeper understanding of the influence of these parameters on the EGO iterates. Finally, a new self-adaptive EGO is presented. With the self-adaptive EGO, we introduce a novel approach for learning parameters directly from their contribution to the optimization.

École Nationale Supérieure des Mines de Saint-Étienne

NNT : 2016LYSEM005

Hossein MOHAMMADI

Optimisation Globale et processus Gaussiens: analyse et nouveaux algorithmes

Spécialité: Mathématiques Appliquées

Mots clefs : Krigeage, Processus Gaussiens, EGO, CMA-ES, Optimisation Globale

Résumé:

L'«Efficient Global Optimization» (EGO) est une méthode de référence pour l'optimisation globale de fonctions «boites noires» coûteuses. Elle peut cependant rencontrer quelques difficultés, comme le mauvais conditionnement des matrices de covariance des processus Gaussiens (GP) qu'elle utilise, ou encore la lenteur de sa convergence vers l'optimum global. De plus, le choix des paramètres du GP, crucial car il contrôle la famille des fonctions d'approximation utilisées, mériterait une étude plus poussée que celle qui en a été faite jusqu'à présent. Enfin, on peut se demander si l'évaluation classique des paramètres du GP est la plus appropriée à des fins d'optimisation.
Ce travail est consacré à l'analyse et au traitement des différentes questions soulevées ci-dessus.

La première partie de cette thèse contribue à une meilleure compréhension théorique et pratique de l'impact des stratégies de régularisation des processus Gaussiens, développe une nouvelle technique de régularisation, et propose des règles pratiques. Une seconde partie présente un nouvel algorithme combinant EGO et CMA-ES (ce dernier étant un algorithme d'optimisation globale et convergeant). Le nouvel algorithme, nommé EGO-CMA, utilise EGO pour une exploration initiale, puis CMA-ES pour une convergence finale. EGO-CMA améliore les performances des deux algorithmes pris séparément. Dans une troisième partie, l'effet des paramètres du processus Gaussien sur les performances de EGO est soigneusement analysé. Finalement, un nouvel algorithme EGO auto-adaptatif est présenté, dans une nouvelle approche où ces paramètres sont estimés à partir de leur influence sur l'efficacité de l'optimisation elle-même.