Statistics and Computing (2021) 31:43
https://doi.org/10.1007/s11222-021-10014-w

Evaluating Gaussian process metamodels and sequential designs for noisy level set estimation

Xiong Lyu[1] · Mickaël Binois[2] · Michael Ludkovski[3]

Received: 27 February 2020 / Accepted: 12 April 2021 / Published online: 26 May 2021

Abstract
We consider the problem of learning the level set for which a noisy black-box function exceeds a given threshold. To efficiently reconstruct the level set, we investigate Gaussian process (GP) metamodels. Our focus is on strongly stochastic simulators, in particular with heavy-tailed simulation noise and low signal-to-noise ratio. To guard against noise misspecification, we assess the performance of three variants: (i) GPs with Student-t observations; (ii) Student-t processes (TPs); and (iii) classification GPs modeling the sign of the response. In conjunction with these metamodels, we analyze several acquisition functions for guiding the sequential experimental designs, extending existing stepwise uncertainty reduction criteria to the stochastic contour-finding context. This also motivates our development of (approximate) updating formulas to efficiently compute such acquisition functions. Our schemes are benchmarked by using a variety of synthetic experiments in 1–6 dimensions. We also consider an application of level set estimation for determining the optimal exercise policy of Bermudan options in finance.

Keywords Gaussian Process · Stochastic contour-finding · Sequential updating formulas · Student-t process

1 Introduction

1.1 Statement of problem

Metamodeling has become widespread for approximating black-box functions that arise in applications ranging from engineering to environmental science and finance (Santner et al. 2013). Rather than aiming to capture the precise shape of the function over the entire region, in this article we are interested in estimating the *level set* where the function exceeds some particular threshold. There is also research on this problem under the name "regression level sets" (Scott and Davenport 2007; Willett and Nowak 2007; Yang et al. 2014). Level set estimation is common in contexts where we need to quantify the reliability of a system or its performance relative to a benchmark. It also arises intrinsically in control frameworks where one wishes to rank the payoff from several available actions (Hu and Ludkovski 2017).

We consider a setup where the latent $f : D \to \mathbb{R}$ is a continuous function over a d-dimensional input space $D \subseteq \mathbb{R}^d$. The level set estimation problem consists in classifying every input $x \in D = S \cup C$ according to

$$S = \{x \in D : f(x) \geq 0\}, \quad C = \{x \in D : f(x) < 0\}. \tag{1.1}$$

Without loss of generality, the threshold is taken to be zero, so that the level set estimation is equivalent to learning the sign of the response function f. For later use, we also define the corresponding zero contour of f, namely the partition boundary $\partial S = \partial C = \{x \in D : f(x) = 0\}$. (Note that ∂S and ∂C do not mean the topological boundary of the sets.)

For any $x_i \in D$, we have access to a simulator Y_i that generates noisy samples of $f(x_i)$:

$$Y_i = f(x_i) + \epsilon_i, \tag{1.2}$$

✉ Xiong Lyu
lyu@pstat.ucsb.edu

Mickaël Binois
mickael.binois@inria.fr

Michael Ludkovski
ludkovski@pstat.ucsb.edu

[1] Department of Statistics and Applied Probability, University of California UCSB, Santa Barbara, CA 93106-3110, USA

[2] Argonne National Laboratory, Mathematics and Computer Science Division, 9700 South Cass Avenue, Lemont, IL 60439, USA

[3] Department of Statistics and Applied Probability, University of California UCSB, Santa Barbara, CA 93106-3110, USA

where ϵ_i are realizations of independent mean zero random variables with variance $\tau^2(x)$, the distribution of which depends on x_i only. The simulator represents realizations of a stochastic system, capturing settings where the output is a function not just of the inputs, but also of some intrinsic random shocks. Those shocks might represent random awards available to a decision maker, or random disturbances that might tip the system into a failed state. The shocks typically have a simple structure (i.i.d uniform or Gaussian) but are fed through a nonlinear black-box transformation to produce observable system outputs with unknown statistical properties. Such stochastic simulators, where different runs with same x yield distinct Y_i's, are ubiquitous across scientific, engineering and financial domains that rely on stochastic models (Baker et al. 2020).

To assess a level set estimation algorithm, we compare the resulting estimate $\widehat{S}$ with the true S in terms of their symmetric difference. Let μ be a probability measure on the Borel σ-algebra $\mathcal{B}(D)$ (e.g., $\mu = \text{Leb}_D$ the Lebesgue measure on D). Then, our loss function is

$$L(S, \widehat{S}) = \mu(S \Delta \widehat{S}), \tag{1.3}$$

where $S_1 \Delta S_2 := (S_1 \cap S_2^c) \bigcup (S_1^c \cap S_2)$, and S^c is the complement of S: $S^c = \{x \in D : x \notin S\}$. Frequently, the inference is carried out by first producing an estimate $\widehat{f}$ of the response function; in that case, we take $\widehat{S} = \{x \in D : \widehat{f}(x) \geq 0\}$ and rewrite the loss as

$$L(f, \widehat{f}) = \int_D \mathbb{I}(\text{sign}\, \widehat{f}(x) \neq \text{sign}\, f(x)) \mu(\text{d}x), \tag{1.4}$$

where $\mathbb{I}(\cdot)$ is the indicator function.

1.2 Motivation

As a concrete example of level set estimation, consider the problem of evaluating the probability of failure, determined via the positive level set S of a performance function $f(\cdot)$ (Vazquez and Bect 2009; Picheny et al. 2010; Bect et al. 2017). The system is safe when $f(x) \leq 0$ and fails otherwise. In the context where the performance function can be evaluated via deterministic experiments, the estimation of the safe zone (more precisely its volume $\mu(S)$) was carried out in Bect et al. (2012) and Mukhopadhyay et al. (2005) employing a Gaussian process metamodel with a sequential design. A related case study dealing with the probability of failure in a nuclear fissile chain reaction appeared in Chevalier et al. (2014).

Another application, which motivated this present investigation, comes from simulation-based algorithms for valuation of Bermudan options (Gramacy and Ludkovski 2015; Ludkovski 2018). In that context, described in Sect. 5, there is a sequence of problems indexed by t. For each t, S_t is the continuation region which is characterized as the zero level set of the timing value $f(x; t)$. The stochastic interpretation links $f(x; t)$ to a conditional expectation of a path-dependent functional of a Markov process X, and the loss function (1.3) corresponds to the quality of the estimated stopping rule relative to the underlying distribution $\mu(\cdot; t)$ of X_t.

1.3 Design of experiments for contour finding

Reconstructing S via a metamodel can be divided into two steps: the construction of the response model and the development of methods for efficiently selecting the simulation inputs $x_{1:n}$, with n as the design size, known as design of experiments (DoE). Since the level set is intrinsically defined in terms of the unknown f, an *adaptive* DoE approach is needed to select x_n's sequentially.

For the response modeling aspect, GP regression, or kriging, has emerged as the most popular nonparametric approach for both deterministic and stochastic black-box functions (Bect et al. 2012; Gramacy and Lee 2009; Picheny et al. 2013a; Jalali et al. 2017). GPs have also been widely used for the level set estimation problem; see Bryan and Schneider (2008), Gotovos et al. (2013), Hu and Ludkovski (2017), Picheny et al. (2010) and Ranjan et al. (2008). In a nutshell, at step n the GP paradigm constructs a metamodel $\widehat{f}^{(n)}$ that is then used to guide the selection of x_{n+1} and also to construct the estimate $\widehat{S}^{(n)}$. To this end, GPs are well suited for sequential design by offering a rich uncertainty quantification aspect that can be (analytically) exploited to construct information-theoretic DoE heuristics. The standard framework is to develop an acquisition function $\mathcal{I}_n(x)$ that quantifies the value of information from a new sample at input x conditional on an existing dataset $(x_{1:n}, y_{1:n})$ and then to maximize $\mathcal{I}_n$:

$$x_{n+1} = \arg\max_{x \in D} \mathcal{I}_n(x). \tag{1.5}$$

Early level set sampling criteria were proposed by Bryan et al. (2006), Vazquez and Martinez (2006), Bichon et al. (2008), Picheny et al. (2010), and Ranjan et al. (2008) based on modifications to the expected improvement criterion (Jones et al. 1998) for response function optimization. A criterion more targeted to reduce the uncertainty about S itself was developed by Bect et al. (2012) using the concept of stepwise uncertainty reduction (SUR). Specifically, the SUR strategy aims to maximize the global learning rate about S; see also Chevalier et al. (2014) for related computational details. Further criteria using tools from random set theory were developed in Chevalier et al. (2013) and Azzimonti et al. (2016) using the notions of Vorob'ev expectation.

1.4 Summary of contributions

The main goal of this article is to present a comprehensive assessment of GP-based surrogates for stochastic contour-finding. Learning $\widehat{S}$ requires blending the exploitation objective to locally estimate the contour ∂S with the exploration objective of visiting less-sampled regions. To do so, we maintain the sequential design paradigm and GP-based surrogates described above which boil down to accurate inference of the mean response and sampling noise that in turn drive the posterior mean $\widehat{f}$ and the posterior GP variance $s(x)^2$.

Our analysis focuses on the effect of observation noise on contour-finding algorithms and complements Picheny et al. (2013b) and Jalali et al. (2017), who benchmarked GP metamodels for Bayesian optimization (BO) where the objective is to evaluate $\max_x f(x)$. The latter study observed the strong impact of ϵ on performance of BO; in our own analysis, we confirm the need for specialized metamodeling frameworks and sequential design strategies in order to strike the best balance in carrying out uncertainty quantification and constructing a robust surrogate.

Motivated by our application settings, we analyze the joint role of the acquisition function $\mathcal{I}_n(\cdot)$ and the metamodel $\widehat{f}$. On the latter front, we seek approaches that are not too swayed by the simulation noise structure. This issue is fundamental to any realistic stochastic simulator where there is no justification for assuming Gaussian-distributed ϵ (as opposed to the physical experimental setup where ϵ represents observation noise and is expected to be Gaussian thanks to the central limit theorem). This motivates us to study *alternative GP-based metamodels* for learning $\widehat{S}$ that are resistant to non-Gaussian ϵ in (1.2).

First, we investigate two ways to handle heavy-tailed simulation noise: t-observation GPs (Vanhatalo et al. 2009; Jylänki et al. 2011) and Student-t processes (TPs) (Shah et al. 2014; Wang et al. 2017). In particular, we document the strong performance of t-observation GPs; to our knowledge, this is the first use of either tool in sequential analysis and contour-finding. Second, to target the classification-like objective underlying (1.3), we consider the use of *classification GPs* (Rasmussen and Williams 2006; Williams and Barber 1998) that model the sign of the response $Y(x)$ via a probit model driven by a latent GP $Z(\cdot)$: $\mathbb{P}(Y(x) > 0|x) = \text{probit}(Z(x))$. Deployment of the probit regression is expected to "wash out" non-Gaussian features in ϵ beyond its effect on the sign of the observations. This context offers an interesting and novel comparison between regression and classification approaches benchmarked against a shared loss function. Third, we present an original use of monotonic GP metamodels (Riihimäki and Vehtari 2010) for level set estimation. This idea exploits a structure commonly encountered in applications where the level set S is *connected*, suggesting to force $\widehat{f}$ to be monotone in the specified coordinates in order to improve upon agnostic black-box strategies.

We then combine all the above metamodels with specialized acquisition functions targeting level set estimation, cf. Sect. 3. We consider four choices (Contour Upper Confidence Bound (cUCB), targeted mean squared error (tMSE), stepwise uncertainty reduction (SUR) and gradient stepwise uncertainty reduction (gSUR)) that include heuristics that depend only on the posterior standard deviation $s^{(n)}(\cdot)$, as well as those that anticipate information gain from sampling at x_{n+1} via the look-ahead standard deviation $s^{(n+1)}(\cdot)$. For the latter, because in the GPs with non-Gaussian noise $s^{(n+1)}$ depends on $Y(x_{n+1})$, we develop and implement approximate *look-ahead* formulas $\widehat{s}^{(n+1)}$ for all our metamodels. To our knowledge, this is the first presentation of such formulas for GPs with non-Gaussian noise, as well as TPs.

After setting up the methodological toolbox, we then provide a detailed comparison among the proposed acquisition functions, identifying the best-performing combinations of $\mathcal{I}(\cdot)$ and metamodel $\widehat{f}$ and documenting the complex interplay between design geometry and surrogate architecture. To this end, we benchmark across a range of synthetic experiments that aim to stress-test each scheme in terms of model misspecification, in particular in terms of handling non-Gaussian ϵ. This benchmarking reveals new insights relative to most of the cited papers that consider only deterministic contour-finding.

The rest of the article is organized as follows. Section 2 describes the metamodels we employ. Section 3 develops the sequential designs for the level set estimation problem. Section 4 compares the models using synthetic data where ground truth is known. Two case studies from derivative pricing are investigated in Sect. 5. In Sect. 6, we summarize our conclusions. The look-ahead variance formulas for GPs with non-Gaussian noise are discussed in Section C of Supplementary Material.

2 Statistical model

2.1 Gaussian process regression with Gaussian noise

We begin by discussing regression frameworks for contour-finding that target learning the latent $f(\cdot)$. The Gaussian process paradigm treats f as a random function whose posterior distribution is determined from its prior and the collected samples $\mathcal{A}_n \equiv \{(x_i, y_i), 1 \le i \le n\}$. We view $f(\cdot) \sim GP(m(\cdot), K(\cdot,\cdot))$, a priori, as a realization of a Gaussian process specified by its mean function $m(x) := \mathbb{E}[f(x)]$ and covariance function

$$K(x, x') := \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))].$$

In the classical case (Rasmussen and Williams 2006), the noise distribution is homoscedastic Gaussian $\epsilon(x) \sim \mathcal{N}(0, \tau^2)$, and the prior mean is zero, $m(x) = 0$. Given observations $\mathbf{y}_{1:n} = [y_1, \ldots, y_n]^\top$ at inputs $\mathbf{x}_{1:n} = [x_1, \ldots, x_n]^\top$, the conditional distribution $f|\mathcal{A}_n$ is then another Gaussian process, with posterior marginal mean $\widehat{f}^{(n)}_{\text{Gsn}}(x_*)$ and covariance $v^{(n)}_{\text{Gsn}}(x_*, x'_*)$ given by (throughout we use subscripts to indicate the metamodel type, e.g., Gsn for Gaussian noise)

$$\widehat{f}^{(n)}_{\text{Gsn}}(x_*) = k(x_*)[\mathbf{K} + \tau^2\mathbf{I}]^{-1}\mathbf{y}_{1:n}, \tag{2.1}$$

$$v^{(n)}_{\text{Gsn}}(x_*, x'_*) = K(x_*, x'_*) - k(x_*)[\mathbf{K} + \tau^2\mathbf{I}]^{-1}k(x'_*)^\top \tag{2.2}$$

with the $1 \times n$ vector $k(x_*)$ and $n \times n$ matrix $\mathbf{K}$ defined by $k(x_*) := K(x_*, \mathbf{x}_{1:n}) = [K(x_*, x_1), ..., K(x_*, x_n)]$, and $\mathbf{K}_{i,j} := K(x_i, x_j)$.

The posterior mean $\widehat{f}^{(n)}_{\text{Gsn}}(x_*)$ is treated as a point estimate of $f(x_*)$ and the posterior variance $s^{(n)}_{\text{Gsn}}(x_*)^2 = v^{(n)}_{\text{Gsn}}(x_*, x_*)$ as the uncertainty of this estimate. We use $\mathbf{f}$ to denote the random posterior vector $f(\mathbf{x}_{1:n})|\mathcal{A}_n$.

Model Fitting: In this article, we model the covariance between the values of f at two inputs x and x' with the squared exponential (SE) function:

$$K_{\text{se}}(x, x') := \sigma^2_{\text{se}} \exp\left(-\sum_{i=1}^{d} \frac{(x^i - x'^i)^2}{2\theta_i^2} \right), \tag{2.3}$$

defined in terms of the hyperparameters $\boldsymbol{\vartheta} = \{\sigma^2_{\text{se}}, \theta_1, ..., \theta_d, \tau\}$ known as the process variance and lengthscales, respectively. Simulation variance τ is also treated as unknown and part of $\boldsymbol{\vartheta}$. To estimate the hyperparameters $\boldsymbol{\vartheta}$, we use the marginal likelihood

$$p(\mathbf{y}_{1:n}|\mathbf{x}_{1:n}, \boldsymbol{\vartheta}) = \int p(\mathbf{y}_{1:n}|\mathbf{x}_{1:n}, \mathbf{f})\, p(\mathbf{f}|\boldsymbol{\vartheta}) \mathrm{d}\mathbf{f}. \tag{2.4}$$

One may similarly express the posterior over the hyperparameters $\boldsymbol{\vartheta}$, where $p(\mathbf{y}_{1:n}|\mathbf{x}_{1:n}, \boldsymbol{\vartheta})$ plays the role of the likelihood. To avoid expensive MCMC integration, we use the maximum likelihood (ML) estimate $\widehat{\boldsymbol{\vartheta}}$ which maximizes the likelihood (2.4). Given the estimated hyperparameters $\widehat{\boldsymbol{\vartheta}}$, we take the posterior of f as $p(\mathbf{f}|\mathbf{y}_{1:n}, \mathbf{x}_{1:n}, \widehat{\boldsymbol{\vartheta}})$. The same method is used for all metamodels throughout this article.

Remark: In the present article, we target the non-Gaussian aspects, in particular the likely heavy-tailed property of simulation noise. A complementary strand of the literature focuses on heteroscedastic simulation variance; see the stochastic kriging approach of Ankenman et al. (2008) and the earlier works by two of the authors (Binois et al. 2018, 2019).

2.2 Gaussian process regression with Student *t* noise

Taking the noise term ϵ_i as Gaussian is widely used since the marginal likelihood is then analytically tractable. In a stochastic simulation setting, however, the exact distribution of the outputs relative to their mean is unknown and often is clearly non-Gaussian. A more robust choice is to assume that ϵ_i has a Student-t distribution (Jylänki et al. 2011). In particular, this may work better when the noise is heavy-tailed by making inference more resistant to outliers (O'Hagan 1979). In the resulting t-GP formulation, ϵ_i is assumed to be t-distributed with variance τ^2 and $\nu > 2$ degrees of freedom. (The latter is treated as another hyperparameter.) Using the Gamma function Γ, the marginal likelihood of observing $\mathbf{y}_{1:n}$ can be written as

$$p_{t\text{GP}}(\mathbf{y}_{1:n}|\mathbf{x}_{1:n}, \mathbf{f}) = \prod_{i=1}^{n} \frac{\Gamma((\nu+1)/2)}{\Gamma(\nu/2)\sqrt{\nu\pi}\sigma_n} \times \left(1 + \frac{(y_i - f_i)^2}{\nu\sigma_n^2}\right)^{-(\nu+1)/2}. \tag{2.5}$$

The likelihood $p_{t\text{GP}}(\mathbf{y}_{1:n}|\mathbf{x}_{1:n}, \mathbf{f})$ is no longer Gaussian, and integrating (2.5) against the Gaussian prior $p(f|\boldsymbol{\vartheta})$ is intractable; we therefore use the Laplace approximation (LP) method (Vanhatalo et al. 2009) to calculate the posterior. A second-order Taylor expansion of $\log p_{t\text{GP}}(\mathbf{f}|\mathbf{x}_{1:n}, \mathbf{y}_{1:n})$ around its mode, $\tilde{\mathbf{f}}^{(n)}_{t\text{GP}} := \arg\max_{\mathbf{f}} p_{t\text{GP}}(\mathbf{f}|\mathbf{x}_{1:n}, \mathbf{y}_{1:n})$, gives a Gaussian approximation

$$p_{t\text{GP}}(\mathbf{f}|\mathbf{x}_{1:n}, \mathbf{y}_{1:n}) \approx q_{t\text{GP}}(\mathbf{f}|\mathbf{x}_{1:n}, \mathbf{y}_{1:n}) = \mathcal{N}\left(\tilde{\mathbf{f}}^{(n)}_{t\text{GP}}, \text{״}^{-1}_{t\text{GP}}\right), \tag{2.6}$$

where $\text{״}_{t\text{GP}}$ is the Hessian of the negative conditional log posterior density at $\tilde{\mathbf{f}}^{(n)}_{t\text{GP}}$:

$$\text{״}_{t\text{GP}} = -\nabla^2 \log p_{t\text{GP}}(\mathbf{f}|\mathbf{x}_{1:n}, \mathbf{y}_{1:n})|_{\mathbf{f}=\tilde{\mathbf{f}}^{(n)}_{t\text{GP}}} = \mathbf{K}^{-1} + \mathbf{W}_{t\text{GP}}, \tag{2.7}$$

and $\mathbf{W}_{t\text{GP}} = -\nabla^2 \log p_{t\text{GP}}(\mathbf{y}_{1:n}|\mathbf{f}, \mathbf{x}_{1:n})|_{\mathbf{f}=\tilde{\mathbf{f}}^{(n)}_{t\text{GP}}}$ is diagonal, since the likelihood factorizes over observations.

Using (2.6), the approximate posterior distribution is also Gaussian $f(x_*)|\mathcal{A}_n \sim \mathcal{N}(\widehat{f}^{(n)}_{t\text{GP}}(x_*), s^2_{t\text{GP}}(x_*))$, defined by its mean $\widehat{f}^{(n)}_{\text{t}}(x_*)$ and covariance $v^{(n)}_{t\text{GP}}(x_*, x'_*)$:

$$\widehat{f}^{(n)}_{t\text{GP}}(x_*) = k(x_*)\mathbf{K}^{-1}\tilde{\mathbf{f}}^{(n)}_{t\text{GP}}, \tag{2.8}$$

$$v^{(n)}_{\text{t}}(x_*, x'_*) = K(x_*, x'_*) - k(x_*)[\mathbf{K} + \mathbf{W}^{-1}_{t\text{GP}}]^{-1}k(x'_*)^\top. \tag{2.9}$$

Note the similarity to (2.1)–(2.2): With Student-t likelihood, the mode $\tilde{\mathbf{f}}^{(n)}_{t\text{GP}}$ plays the role of $\mathbf{y}_{1:n}$ and $\boldsymbol{W}^{-1}_{t\text{GP}}$ replaces

the noise matrix $\tau^2\mathbf{I}$. Critically, the latter implies that the posterior variance is a function of both designs $\mathbf{x}_{1:n}$ and observations $\mathbf{y}_{1:n}$.

2.3 Gaussian process classification

Our target in (1.1) is to learn where the mean response is positive, which is equivalent to classifying each $x \in D$ as belonging either to S or to N. Assuming that $\epsilon(x)$ is symmetric, $\{x \in S\} = \{f(x) \geq 0\} = \{\mathbb{P}(Y(x) > 0) > 0.5\}$. This motivates us to consider the alternative of directly modeling the response sign (rather than overall magnitude) via a classification GP model (Cl-GP) (Williams and Barber 1998; Rasmussen and Williams 2006). The idea is to model the probability of a positive observation $Y(x)$ by using a probit regression: $\mathbb{P}(Y(x) > 0|x) = \Phi(Z(x))$, with $\Phi(\cdot)$ the standard normal cdf. We follow the notations used by Rasmussen and Williams (2006) to formulate the Cl-GP. Other link functions can also be used to model the probability: For example, Williams and Barber (1998) used logistic regression with $\mathbb{P}(Y(x) > 0|x) = \frac{1}{1+e^{-Z(x)}}$. The latent classifier function is taken as the GP $Z \sim GP(0, K(\cdot,\cdot))$. After learning Z, we then set $\widehat{S} = \{x \in D : \widehat{Z}(x) > 0\}$.

Remark: Cl-GP fundamentally recovers the "median-zero" level set where $\{x : \mathbb{P}(Y(x) \geq 0) = 0.5\}$. In cases where the noise is skewed, the recovered Cl-GP would therefore be biased relative to the level set $\{x : f(x) \geq 0\}$. In that situation, use of Cl-GP could be viewed as a bias-variance trade-off.

To compute the posterior distribution of Z conditional on $\mathcal{A}_n$, we use the fact that for an observation (x_i, y_i) and conditional on $z_i = Z(x_i)$, the likelihood of $y_i > 0$ is $\Phi(z_i)1_{\{y_i \geq 0\}} + (1 - \Phi(z_i))1_{\{y_i < 0\}}$. To simplify notation, we use $\check{Y}(x) = \operatorname{sign} Y(x) \in \{-1, 1\}$ to represent the signed responses driving Cl-GP, leading to $p_{\mathrm{Cl}}(\check{\mathbf{y}}_{1:n}|\mathbf{z}, \mathbf{x}_{1:n}) = \prod_{i=1}^n \Phi(\check{y}_i z_i)$. The posterior of the latent $\mathbf{z} = Z(x_{1:n})$ is therefore

$$p_{\mathrm{Cl}}(\mathbf{z}|\mathbf{x}_{1:n}, \check{\mathbf{y}}_{1:n}) = \frac{p(\mathbf{z}|\mathbf{x}_{1:n}) \prod_{i=1}^n \Phi(\check{y}_i z_i)}{p(\check{\mathbf{y}}_{1:n}|\mathbf{x}_{1:n})}. \tag{2.10}$$

Similar to t-GP, we follow the implementation described in Williams and Barber (1998) and Rasmussen and Williams (2006) and use a Laplace approximation for the non-Gaussian $p_{\mathrm{Cl}}(\mathbf{z}|\mathbf{x}_{1:n}, \check{\mathbf{y}}_{1:n})$ in Eq. (2.10) (details to be found in Supplementary Material Section B). The posterior mean for $Z(\cdot)$ at x_* is then expressed by using the GP predictive mean equation (2.1) and LP approximation (B.1 of Supplementary Material):

$$\widehat{z}^{(n)}(x_*) = k(x_*)\mathbf{K}^{-1}\tilde{\mathbf{z}}^{(n)}, \tag{2.11}$$

$$v_{\mathrm{Cl}}^{(n)}(x_*, x'_*) = K(x_*, x'_*) - k(x_*)[\mathbf{K} + \mathbf{V}^{-1}]^{-1}k(x'_*)^\top. \tag{2.12}$$

We see the same algebraic structure, with $\tilde{\mathbf{z}}^{(n)}$ a stand-in for $\mathbf{y}_{1:n}$ in (2.1) and $\mathbf{V}^{-1}$ a stand-in for $\tau^2\mathbf{I}$ in (2.2).

2.4 Student-*t* process regression with Student *t* noise

Instead of just adding Student-t likelihood to the observations, Shah et al. (2014) proposed t-processes (TPs) as an alternative to GPs, deriving closed-form expressions for the marginal likelihood and posterior distribution of the t-process by imposing an inverse Wishart process prior over the covariance matrix of a GP model. They found the t-process to be more robust to model misspecification and to be particularly promising for BO as TPs retain most of the appealing properties of GPs, including analytical expressions, with increased flexibility.

Dealing with noisy observations is less straightforward with TPs, since the sum of two independent Student-t distributions has no closed form. Shah et al. (2014) showed that this drawback can be circumvented by incorporating the noise directly in the kernel. The corresponding data-generating mechanism is taken to be multivariate-t $\mathbf{y}_{1:n} \sim \mathcal{T}\left(\nu, m(\mathbf{x}_{1:n}), \mathbf{K} + \tau^2\mathbf{I}\right)$, where the degrees of freedom are $\nu \in (2, \infty)$. The posterior predictive distribution is then $f(x_*)|\mathcal{A}_n \sim \mathcal{T}\left(\nu + n, \widehat{f}_{\mathrm{TP}}^{(n)}(x_*), v_{\mathrm{TP}}^{(n)}(x_*, x_*)\right)$, where (Shah et al. 2014)

$$\widehat{f}_{\mathrm{TP}}^{(n)}(x_*) = k(x_*)[\mathbf{K} + \tau^2\mathbf{I}]^{-1}\mathbf{y}_{1:n}, \tag{2.13}$$

$$v_{\mathrm{TP}}^{(n)}(x_*, x'_*) = \frac{\nu + \beta^{(n)} - 2}{\nu + n - 2}\left\{K(x_*, x'_*) - k(x_*)[\mathbf{K} + \tau^2\mathbf{I}]^{-1}k(x'_*)^\top\right\}, \tag{2.14}$$

with

$$\beta^{(n)} := \mathbf{y}_{1:n}^\top[\mathbf{K} + \tau^2\mathbf{I}]^{-1}\mathbf{y}_{1:n}.$$

Comparing with the regular GPs, we have the same posterior mean $\widehat{f}_{\mathrm{TP}}^{(n)}(x_*) = \widehat{f}_{\mathrm{Gsn}}^{(n)}(x_*)$, but the posterior covariance now depends on observations $\mathbf{y}_{1:n}$ and is inflated: $v_{\mathrm{TP}}^{(n)}(x_*, x'_*) = \frac{\nu+\beta^{(n)}-2}{\nu+n-2}v_{\mathrm{Gsn}}^{(n)}(x_*, x'_*)$. Moreover, the latent function f and the noise are uncorrelated but not independent. Assuming the same hyperparameters, as n goes to infinity, the above predictive distribution becomes Gaussian.

Inference of TPs can be performed similarly as for a GP, for instance, based on the marginal likelihood:

$$p_{\mathrm{TP}}(\mathbf{y}_{1:n}|\mathbf{x}_{1:n},\boldsymbol{\vartheta}) = \frac{\Gamma(\frac{\nu+n}{2})}{((\nu-2)\pi)^{\frac{n}{2}}\Gamma(\frac{\nu}{2})}|\mathbf{K}|^{-1/2} \times \left(1+\frac{\mathbf{y}_{1:n}^{\top}\mathbf{K}^{-1}\mathbf{y}_{1:n}}{\nu-2}\right)^{-\frac{\nu+n}{2}}. \tag{2.15}$$

One issue is estimation of ν, which plays a central role in the TP predictions. It is found in Shah et al. (2014) that restricting ν to be small when maximizing (2.15) is important in order to avoid degenerating to the plain GP setup.

2.5 Metamodel performance for level set inference

To evaluate the performance of different metamodels, we consider several metrics. The first statistic is the error rate $\mathcal{ER}$ based on the loss function L defined in Eq. (1.3), measuring the distance between the level set S and its estimate $\widehat{S}$:

$$\begin{aligned}\mathcal{ER} &:= \mu(S\Delta\widehat{S}) \\ &= \int_D \mathbb{I}\left[\operatorname{sign} f(x) \neq \operatorname{sign}\widehat{f}(x)\right]\mu(\mathrm{d}x).\end{aligned} \tag{2.16}$$

For Cl-GP, we replace $f(x)$ with $Z(x)$, namely use $\mathcal{ER} = \int_D \mathbb{I}\left[\operatorname{sign} Z(x) \neq \operatorname{sign}\widehat{z}(x)\right]\mu(\mathrm{d}x)$.

The error rate $\mathcal{ER}$ evaluates the accuracy of the estimated $\widehat{S}$ when the ground truth is known. In a realistic case study when the latter is unavailable, we replace $\mathcal{R}$ by its empirical counterpart, based on quantifying the uncertainty in $\widehat{S}$ through the associated uncertainty of $\widehat{f}$. Following Azzimonti et al. (2016), we define the integrated posterior error $\mathcal{E}$ as the expected distance in measure between the random set $S|\mathcal{A}$ and $\widehat{S}$:

$$\mathcal{E} := \mathbb{E}\left[\mu(S\Delta\widehat{S})|\,\mathcal{A}\right] = \int_D \bar{E}(x)\mu(\mathrm{d}x), \tag{2.17}$$

with local posterior error $\bar{E}(x)$ calculated by using (2.1) and (2.2):

$$\begin{aligned}\bar{E}(x) &:= \mathbb{E}\left[\mathbb{I}[\operatorname{sign} f(x) \neq \operatorname{sign}\widehat{f}(x)]|\mathcal{A}\right] \\ &= \int_{\mathbb{R}} \mathbb{I}[\operatorname{sign} f(x) \neq \operatorname{sign}\widehat{f}(x)]p(f(x)|\mathcal{A})\mathrm{d}f(x) \\ &= \Phi\left(\frac{-|\widehat{f}(x)|}{s(x)}\right).\end{aligned} \tag{2.18}$$

The local posterior error $\bar{E}(x)$ is the posterior probability of wrongly classifying x conditional on the training dataset $\mathcal{A}$. It is intrinsically tied to the point estimate $\widehat{f}(x)$ and the associated posterior variance $s(x)^2$ through the Gaussian uncertainty quantification. For TP, the predictive distribution is Student-t, so that the Gaussian cdf Φ is replaced with survival function. For Cl-GP, we replace $\widehat{f}(x)$ with $\widehat{z}(x)$ in (2.18).

Uncertainty Quantification: To quantify the overall uncertainty about S (rather than local uncertainty about $f(x)$), a natural criterion is the *volume* of the credible band $CI_{\partial S}^{\alpha}$ that captures inputs x whose sign remains ambiguous given $\mathcal{A}$. Given a credibility level α (e.g., $\alpha = 0.05$), we define

$$CI_{\partial S}^{\alpha\,(n)} := \left\{x \in D : |\widehat{f}^{(n)}(x)| \leq z_{1-\frac{\alpha}{2}}s^{(n)}(x)\right\}, \tag{2.19}$$

where $z_{1-\frac{\alpha}{2}}$ is the appropriate Gaussian/Student-t α-quantile. Thus (2.19) is the region where the sign of f is non-constant over the posterior α-CI of f. Observe that $x \in CI_{\partial S}^{\alpha\,(n)} \Leftrightarrow \bar{E}(x) > \frac{\alpha}{2}$. Noting that the posterior error is $\bar{E}(x) = 0.5$ for $x \in \partial S$, we have the approximation $\int_D \bar{E}(x)\mathrm{d}x \simeq \int_D[0.5 \cdot 1_{\bar{E}(x)>0.5\alpha} + 0 \cdot 1_{\bar{E}(x)<0.5\alpha}]\mathrm{d}x = 0.5\mu(CI_{\partial S}^{\alpha})$. We have confirmed empirically that the area of $CI_{\partial S}^{\alpha}$ is roughly proportional to the integrated posterior error $\mathcal{E}$. We also mention that $\mathcal{E}$ is equivalent to the Vorob'ev deviation Chevalier et al. (2013) with a median Vorob'ev threshold.

3 Sequential design

We estimate the level set S in a sequential design setting that assumes that f is expensive to evaluate, for example, because of the complexity of the underlying stochastic simulator. Therefore, efficient selection of the inputs $\mathbf{x}_{1:n}$ is important. In sequential design, at each step the next sampling location x_{n+1} is selected given all previous measurements. We follow the standard approach to sequential design that is based on greedily optimizing a posterior-based acquisition function $\mathcal{I}$ as in (1.5). These strategies got popularized thanks to the success of the expected improvement (EI) criterion and the associated efficient global optimization (EGO) algorithm (Jones et al. 1998). The basic loop for sequential design is:

- Initialize $\mathcal{A}_{n_0} = \{(x_i, y_i), 1 \leq i \leq n_0\}$.
- Loop for $n = n_0{+}1, \ldots, N$.
  - Choose the next input $x_{n+1} = \arg\max_{x\in\mathcal{M}} \mathcal{I}_n(x)$, and sample $y_{n+1} = Y(x_{n+1})$.
  - Augment $\mathcal{A}_{n+1} = \mathcal{A}_n \bigcup\{(x_{n+1}, y_{n+1})\}$.
  - Update $\widehat{S}^{(n+1)}$ with $\mathcal{A}_{n+1}$.

We now propose several acquisition functions $\mathcal{I}_n(x)$ in Eq. (1.5). The key plan is to target regions close to the boundary $\partial\widehat{S}$. A second strategy is to use the look-ahead posterior standard deviation $s^{(n+1)}$ conditional on sampling at x, in

order to assess the corresponding *information gain*. This links the constructed design to the metamodel for f, since different surrogate architectures quantify uncertainty differently.

3.1 Contour upper confidence bound

The first metric, dubbed Contour Upper Confidence Bound (cUCB), stems from the Upper Confidence Bound (UCB) strategies proposed by Srinivas et al. (2012) for Bayesian optimization. The idea of UCB is to express the exploitation–exploration trade-off through the posterior mean $\widehat{f}(x)$ and standard deviation $s(x)$. Following the spirit of UCB, cUCB blends the minimization of $|\widehat{f}^{(n)}(x)|$ (exploitation) with maximization of the posterior uncertainty $s^{(n)}(x)$ (exploration):

$$\mathcal{I}_n^{\text{cUCB}}(x) := \left\{-|\widehat{f}^{(n)}(x)| + \gamma^{(n)} s^{(n)}(x)\right\} \mu(x), \tag{3.1}$$

where $\gamma^{(n)}$ is a step-dependent sequence of weights. Thus, cUCB targets inputs with high uncertainty (large $s^{(n)}(x)$) and close to the boundary $\partial\widehat{S}$ (small $|\widehat{f}^{(n)}|$), additionally weighted by $\mu(\cdot)$. Small $\gamma^{(n)}$ leads to aggressive sampling concentrated along the estimated $\partial\widehat{S}$; large $\gamma^{(n)}$ leads to space-filling sampling that effectively minimizes the ultimate integrated mean-squared error. Thus, the choice of γ's is critical for the performance; in particular, $\gamma^{(n)}$ should be increasing to avoid being trapped in local minima of $|\widehat{f}^{(n)}(x)|$. In the original application to BO, Srinivas et al. (2012) proposed $\gamma^{(n)} = C \log n$ and showed that for a certain choice of C, one can then control with high probability the respective cumulative regret. A constant choice of $\gamma^{(n)} = 1.96$ corresponds to the Straddle scheme in Bryan et al. (2006) and leads to $\mathcal{I}_n(x) \geq 0 \Leftrightarrow x \in CI^{0.95}(\partial S)$. Gotovos et al. (2013) employed $\gamma^{(n)} = 3$ for a confidence region instead of confidence interval as in Straddle and (3.1), and Bogunovic et al. (2016) suggested $\gamma^{(n)} = \sqrt{\log(|D|n^2)}$; both papers mention that the recommendation in Srinivas et al. (2012) is too conservative and tends to over-exploration. Based on our experiments (see Supplementary Material Section A), we recommend to adapt $\gamma^{(n)}$ to the relative ratio between $\widehat{f}^{(n)}(x)$ (for steeper response surfaces, γ should be larger) and $s^{(n)}(x)$. (γ needs to rise as posterior uncertainty decreases.). Our recipe is

$$\gamma^{(n)} = \frac{\text{IQR}(\widehat{f}^{(n)})}{3\text{Ave}(s^{(n)})}, \tag{3.2}$$

where $\text{Ave}(s^{(n)})$ denotes the average of posterior standard deviation and IQR is the inter-quantile range of $\widehat{f}^{(n)}$ over D. This keeps both terms in (3.1) comparable as n changes. In Supplementary Material Section A, we investigate the performance of cUCB with different $\gamma^{(n)}$'s, see Table 1 in Supplementary Material. We observe that (1) there is no single choice that consistently performs best; (2) the adaptive $\gamma^{(n)}$ is most frequently the best, achieving the smallest error rate in roughly half the cases. Note that since $s^{(n)}$ decreases in n (signal-to-noise ratio increases over time), (3.2) makes $\gamma^{(n)}$ increase in n, which is consistent with the theoretical results of Srinivas et al. (2012).

Remark 1 The local posterior error $\bar{E}(x)$ as defined in Eq. (2.18) could be directly used as an acquisition function, i.e.,

$$\mathcal{I}_n^{\text{LPPM}}(x) \equiv \bar{E}(x) = \Phi\left(-\frac{|\widehat{f}^{(n)}(x)|}{s^{(n)}(x)}\right). \tag{3.3}$$

This Local Posterior Probability of Misclassification (LPPM) acquisition function is similar to the sequential criteria in Echard et al. (2010), Ranjan et al. (2008), Bichon et al. (2008), and Bryan et al. (2006), all based on the idea of sampling at x where the event $\{f(x) \geq 0\}|\mathcal{A}_n$ is most uncertain. However, (3.3) is not suitable for our purposes since it is maximized across the entire $\partial\widehat{S}$ (namely $\mathcal{I}_n^{\text{LPPM}}(x) = 0.5$ for any x where $\widehat{f}^{(n)}(x) = 0$), so does not possess a unique maximizer as soon as $\partial\widehat{S}$ is non-trivial. One potential solution could be to maximize (3.3) over a finite candidate set, which, however, requires significant fine-tuning.

3.2 Integrated stepwise uncertainty reduction

In order to take into account the spatial structure of D, we next consider a criterion that targets the *global* reduction in the uncertainty of $\widehat{S}$. The integrated Stepwise Uncertainty Reduction metric (SUR), first proposed by Bect et al. (2012), is linked to the posterior error $\mathcal{E}$ from Sect. 2.5:

$$\begin{aligned}\mathcal{I}_n^{\text{SUR}}(x) &:= \mathcal{E}^{(n)} - \mathbb{E}_{Y(x)}[\mathcal{E}^{(n+1)}|x_{n+1} = x] \\ &= \mathcal{E}^{(n)} - \mathbb{E}_{Y(x)}\left[\int_{u\in D} \Phi\left(\frac{-|\widehat{f}^{(n+1)}(u)|}{s^{(n+1)}(u)|_{x_{n+1}=x}}\right)\mu(\mathrm{d}u)\right].\end{aligned} \tag{3.4}$$

Crucially, $\mathcal{I}^{SUR}$ ties the selection of x_{n+1} to the look-ahead mean $\widehat{f}^{(n+1)}(x_{n+1})$ and look-ahead standard deviation $s^{(n+1)}(x_{n+1})$ that appear on the right-hand side of (3.4). To compute the integral over $Y(x)$, we replace $\widehat{f}^{(n+1)}(x)$ with its average $\widehat{f}^{(n)}(x) = \mathbb{E}_n[f(x)] = \mathbb{E}_n[\mathbb{E}_{n+1}[f(x)]] = \mathbb{E}_n[\widehat{f}^{(n+1)}(x)]$. Similarly, we plug in the approximate one-step-ahead standard deviation $\widehat{s}^{(n+1)}$, cf. Supplementary Material Section C (especially Equations (C.1), (C.13) and (C.15)) for $s^{(n+1)}(x)$. Lastly, we replace the integral over D with a sum over a finite subset $\mathcal{D}$ of size M leading to

$$\widehat{\mathcal{I}}_n^{\text{SUR}}(x) = -\sum_{x_m\in\mathcal{D}} \Phi\left(\frac{-|\widehat{f}^{(n)}(x_m)|}{\widehat{s}^{(n+1)}(x_m)|_{x_{n+1}=x}}\right)\mu(x_m). \tag{3.5}$$

Then, $\mathcal{I}^{SUR}(x)$ can be viewed as measuring the overall information gain about S from sampling at x. The motivation behind SUR is to minimize the expected one-step-ahead posterior error $\mathcal{E}$, which would correspond to 1-step Bayes-optimal design.

Remark 2 In the case where $\epsilon_i \sim \mathcal{N}(0, \tau^2)$, Chevalier (2013) calculated an analytical expression for $\mathbb{E}_{Y(x)}\left[\mathcal{E}^{(n+1)}\right]$, see Supplementary Material Section E. That computation does not work with non-Gaussian noise, or for Cl-GP and TP. In Supplementary Material Section E, we compare the two resulting expressions and find that the approximate (3.4) with $\widehat{s}^{(n+1)}(x)$ performs generally better, being less sensitive to the noise structure.

3.3 Gradient SUR

Our third strategy focuses on quickly *reducing* the posterior error by comparing the current $\bar{E}(x_{n+1})$ given $\mathcal{A}_n$ and the expected $\bar{E}(x_{n+1})$ conditional on the one-step-ahead sample, $\mathcal{A}_n \cup \{x_{n+1}, y_{n+1}\}$. This is conceptually similar to SUR above but considers just the local information gain, dropping the computationally intensive integral over D and only integrating out the effect of $Y(x_{n+1})$ on $\bar{E}(x_{n+1})$:

$$\begin{aligned}\mathcal{I}_n^{\mathrm{gSUR}}(x) &:= \{\mathcal{I}_n^{\mathrm{LPPM}}(x) - \mathbb{E}_{Y(x)}\left[\mathcal{I}_{n+1}^{\mathrm{LPPM}}(x)\right]\}\mu(x)\\ &= \left\{\Phi\left(-\frac{|\widehat{f}^{(n)}(x)|}{s^{(n)}(x)}\right)\right.\\ &\quad \left. - \mathbb{E}_{Y(x)}\left[\Phi\left(-\frac{|\widehat{f}^{(n+1)}(x)|}{s^{(n+1)}(x)}\right)\right]\right\}\mu(x). \qquad (3.6)\end{aligned}$$

The name gSUR is because (3.6) is related to the knowledge gradient strategy of Frazier et al. (2008), modified to target contour-finding. We apply the same approximation as for SUR to simplify the expectation over $Y(x)$

$$\begin{aligned}\widehat{\mathcal{I}}_n^{\mathrm{gSUR}}(x) = &\left\{\Phi\left(-\frac{|\widehat{f}^{(n)}(x)|}{s^{(n)}(x)}\right)\right.\\ &\left. - \Phi\left(-\frac{|\widehat{f}^{(n)}(x)|}{\widehat{s}^{(n+1)}(x)|_{x_{n+1}=x}}\right)\right\}\mu(x). \qquad (3.7)\end{aligned}$$

Note that if x is such that $\widehat{f}^{(n)}(x) = 0$, then both terms above are $1/2$ and $\mathcal{I}_n^{\mathrm{gSUR}}(x) = 0$. Thus, the gSUR criterion will not place samples *directly* on $\partial\widehat{S}$, but will aim to bracket the zero contour. To our knowledge, gSUR is a new criterion for sequential level set estimation that interpolates between the local cUCB and the integrated SUR while still making use of the predictive local gain in information about the contour. As for SUR, one may utilize Equation (E.2) in Supplementary Material for $\mathcal{I}^{\mathrm{gSUR}}$ under the implicit assumption that the simulation noise is Gaussian.

3.4 Targeted mean squared error

As a last alternative, we utilize the targeted mean squared error (tMSE) criterion. tMSE is a localized form of the targeted IMSE criterion in Picheny et al. (2010)

$$\mathcal{I}_n^{\mathrm{tIMSE}}(x) := \int s^{(n)}(x)^2 \cdot W_n^{\mathrm{tMSE}}(x)\mu(\mathrm{d}x), \qquad (3.8)$$

and is defined as

$$\mathcal{I}_n^{\mathrm{tMSE}}(x) := s^{(n)}(x)^2 \cdot W_n^{\mathrm{tMSE}}(x)\mu(x), \qquad (3.9)$$

$$\text{where} \quad W_n^{\mathrm{tMSE}}(x) := \frac{1}{\sqrt{2\pi}s^{(n)}(x)} \exp\left(-\frac{\widehat{f}_n(x)^2}{2s^{(n)}(x)^2}\right). \qquad (3.10)$$

The tMSE criterion upweighs regions close to the zero contour through the weight function $W_n^{\mathrm{tMSE}}(x)$ which measures the distance of x to $\partial\widehat{S}^{(n)}$ using the Gaussian posterior density $\mathcal{N}(\widehat{f}^{(n)}, s^{(n)}(x)^2)$. Like cUCB, tMSE is based only on the posterior at step n and does not integrate over future $Y(x)$'s. The tMSE is implemented in the `KrigInv` R package. Similar to cUCB and gSUR, tMSE picks the input in the region close to the zero contour at step n and favors exploitation, while SUR and tIMSE pick the input that will minimize the integrated one-step-ahead uncertainty over the entire space and is explorative. Computationally, the localized criteria are more efficient since they skip the expensive integration. In this article, we use SUR as a representative of the exploratory criteria and compare its performance with the other localized criteria.

Remark 3 In Picheny et al. (2010), an additional parameter σ_ϵ was added to the definition of $W_n^{\mathrm{tMSE}}(x)$ by replacing $s^{(n)}(x)$ everywhere with $\sqrt{s^{(n)}(x)^2 + \sigma_\epsilon^2}$. Larger σ_ϵ yields more space-filling designs as $W_n^{\mathrm{tMSE}}(x)$ becomes flatter. Since Picheny et al. (2010) dealt with deterministic experiments, σ_ϵ was necessary to ensure that $W_n^{\mathrm{tMSE}}(x)$ is well defined at existing $x_{1:n}$ and the recommendation was for σ_ϵ to be 5% of the range of f. In our case, $s^{(n)}(x)$ is intrinsically bounded away from zero and (3.10) works well as is. Additional experiments (available upon request) indicate that the performance of (3.9) is not sensitive to σ_ϵ, so to minimize the number of tuning parameters we stick to $\sigma_\epsilon = 0$ in (3.10).

In the TP case, for cUCB, gSUR and SUR, we replace the standard normal cdf $\Phi(\cdot)$ appearing in the formulas by its Student-t counterpart (with the estimated degrees of freedom ν_n). For tMSE, to maintain tractability, we keep the same expression (3.10) for the weights W^{tMSE}.

3.5 Illustration

For instructive purposes, we consider a one-dimensional case where we use the Gaussian observation GP to learn the sign of the quadratic $f(x) = x^2 - 0.75^2$ on $D = [0, 1]$, where $S = [0, 0.75]$ and with the unique zero contour at $\partial S = 0.75$. The initial design $\mathbf{x}_{1:10}$ consists of $n = 10$ inputs drawn according to Latin hypercube sampling (LHS). The observations are $Y(x) = f(x) + \epsilon$, where $\epsilon \sim t_3(0, 0.1^2)$. In the top plot in Fig. 1, we plot the true $f(\cdot)$, the posterior mean $\widehat{f}^{(100)}(\cdot)$, and the associated 95%-CI. We also show the credible band for $\partial\widehat{S}$; in the respective bottom panel, we plot the acquisition functions $\mathcal{I}_n^{\text{cUCB}}(\cdot)$, $\mathcal{I}_n^{\text{gSUR}}(\cdot)$, $\mathcal{I}_n^{\text{SUR}}(\cdot)$ and $\mathcal{I}_n^{\text{tMSE}}(\cdot)$ as defined in Eqs. (3.1), (3.7), (3.5), and (3.9).

Comparing the acquisition functions of the four criteria, we find that, besides SUR, all of the others have maxima within the shaded credible interval of the boundary $CI_{\partial S}^{0.95}$. In practice, we care only about the maximizer of the acquisition function, rather than its full shape, since the former drives the selection of the next sample x_{n+1}. The x_{n+1}'s selected by cUCB and tMSE criteria are close. For the gSUR criterion, because $\mathcal{I}_n^{\text{gSUR}}(x) = 0$ at $\partial\widehat{S}$, there are two local maxima with a "valley" between them. The interval between the two local maxima is roughly the credible interval $CI_{\partial S}^{0.95}$ for the boundary (2.19). Both cUCB and tMSE select a location very close to the boundary $\widehat{f}^{(n)}(x_{n+1}) \simeq 0$. We note that SUR has the flattest acquisition function among all the criteria, consistent with the idea that it will tend to be more explorative.

After using the various acquisition functions to select x_{n+1} at $n = 11, \ldots, 100$, we show in Fig. 2 the resulting designs $\mathbf{x}_{1:n}$ and the final estimate $\widehat{f}^{(100)}$ with a Gaussian observation GP metamodel. As desired, all methods target the true zero contour at $\partial S = 0.75$. As a result, the posterior variance $s^{(n)}(x)^2$ is much lower in this neighborhood; in contrast, especially for tMSE and cUCB, few samples are taken far from $x = 0.75$, and the posterior uncertainty there remains high. The true zero contour is within the estimated posterior CI for all the criteria.

The bottom row in Fig. 2 shows the sampled location x_n as a function of step n. We observe that cUCB and tMSE heavily concentrate their search around the zero contour, leading to few samples (and consequently relatively large posterior errors $\mathcal{E}^{(n)}$) in other areas, and much wider posterior CIs for these two criteria although the overall error rate $\mathcal{ER}$ is comparable. All criteria exhibit an "edge" effect; that is, besides the desired zero contour $x = 0.75$, multiple samples are taken close to the edges of the input space at $x = 0$ and $x = 1$. This occurs due to the relatively large posterior variance $s^2(\cdot)$ in those regions (which arises intrinsically with any spatially-based metamodel) that in turn strongly influences $\mathcal{I}$ in (3.1), (3.7), (3.5) and (3.9). Inputs sampled by the gSUR criterion bracket the contour ∂S from both directions, matching the two-hill-and-a-valley shape of $\mathcal{I}^{\text{gSUR}}$ in Fig. 1. We note that the two sampling "curves" get closer as n grows, indicating a gradual convergence of the estimated zero contour $\partial\widehat{S}^{(n)}$, akin to a shrinking credible interval of $\widehat{S}^{(n)}$. The SUR cri-

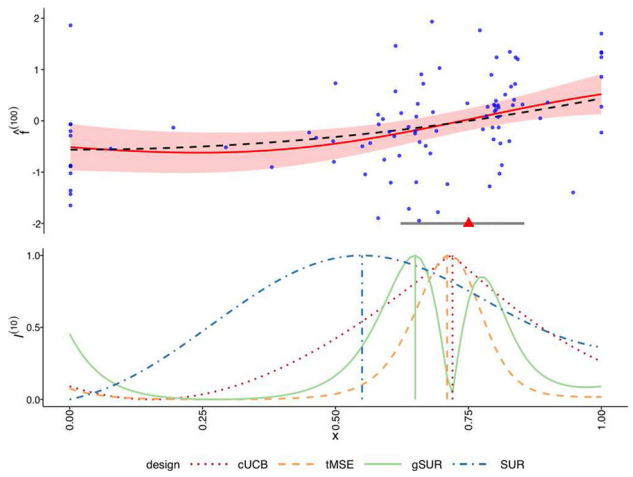


Fig. 1 Comparison of acquisition functions. *Upper* panel: true function $f = (x + 0.75)(x - 0.75)$ (black dashed line), the posterior mean $\widehat{f}(\cdot)$ (solid line) and 95% CI_f^α (shaded area) based on observed samples $(\mathbf{x}_{1:100}, \mathbf{y}_{1:100})$ (blue dots). Along the x-axis, we also show the credible interval of the partition boundary $CI_{\partial S}^\alpha$ (gray solid line) relative the true zero contour $S = [0, 0.75]$ (red triangle). *Lower* panel: acquisition functions $\mathcal{I}_n(\cdot)$ for cUCB, gSUR, SUR and tMSE criteria, with vertical lines marking the respective maxima $\arg\max_x \mathcal{I}_n(x)$

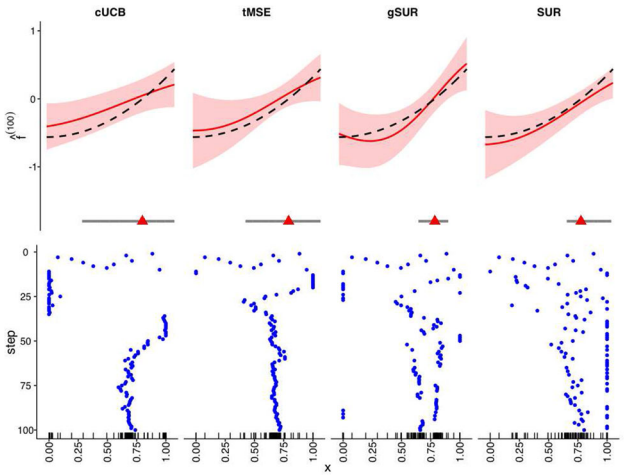


Fig. 2 *Top row*: Fitted metamodel $\widehat{f}^{(100)}$ (solid red line) and its 95%-CI (shaded region) versus the true $f = (x + 0.75)(x - 0.75)$ (dashed black), for each of the four design strategies. The estimated 95% CI for the zero contour ∂S is marked on the x-axis with a gray interval; red triangle indicates the true zero contour $\partial S = 0.75$. *Bottom row*: sampled inputs x_n (on the x-axis to match the top row) as a function of step $n = 1, \ldots, 100$ (on the y-axis, moving from top to bottom) for cUCB, tMSE, gSUR and SUR criteria. The rug plots at the bottom visualize the overall distribution of $\mathbf{x}_{1:n}$ at $n = 100$. The first ten inputs are selected using a (fixed-across schemes) LHS design on $D = [0, 1]$

terion generates a much more diffuse design: It engages in more exploration and is less dependent on the current levels of the posterior error $\mathcal{E}$. This eventually creates a flatter profile for $\bar{E}(x)$.

The preceding discussion considered a single metamodel choice for f. Although Figs. 1 and 2 only present results of one run for each design with plain GP, the features observed are generic. Other metamodels will generate different design features; in particular, sensitivity to ϵ will lead to a different mix of exploration (x_n's far from the zero contour) and exploitation even for the same choice of a $\mathcal{I}_n$ criterion. Figures 6 and 8, as well as Table 3, emphasize our message that one must jointly investigate the *combinations* of $\mathcal{I}(\cdot)$ and $\widehat{f}$ when benchmarking the performance of the algorithm.

4 Synthetic experiments

4.1 Benchmark construction

As synthetic experiments, we consider four benchmark problems in dimension $d = 1, 2$ and 6. For the latter two, we borrow the widely used *Branin-Hoo* 2-D, *Michalewicz* 2-D and *Hartman* 6-D functions; see, for example, Picheny et al. (2013b). The original functions have been rescaled to map their sample space D onto $[0, 1]^d$; see Table 1.

The latent functions are chosen to cover a variety of problem properties. The quadratic f in 1-D is strictly monotonically increasing, yielding a single boundary ∂S. The original Branin-Hoo function (Picheny et al. 2013b) is modified so that f is increasing in x^1 and the zero-level set has a non-trivial shape in x^2. The *Hartman* is a multimodal function with a complex zero contour. The parameters in the original *Hartman* function described, for instance, in Picheny et al. (2013b) are adjusted to reduce the "bumps" in the zero contour and make the problem more appropriate for the sign classification task. These three functions are used to provide a comprehensive comparison between all metamodels.

The Michalewicz function features two intersecting "valleys" surrounded by plateaus and is known to be challenging for GP Bayesian optimization because the spatial correlation is high along the plateaus, but very low (response almost discontinuous) around the local minima. This implies a spatially non-stationary response covariance which makes a Gaussian-noise GP highly misspecified. By construction, CI-GP focuses only on the sign of $f(\cdot)$ and so should outperform level set estimation for such a case study. We used this function to compare performance of classification and regression GPs.

A large number of factors can influence the performance of metamodels and designs. In line with the stochastic simulation perspective, we concentrate on the impact of the simulation noise and consider four observation setups. These cover a variety of noise distributions and signal-to-noise ratio (SNR), measured through the proportion of standard deviation σ_τ to the range R_f of the response. The first two settings use *Student-t* distributed noise, with (i) low σ_τ and (ii) high σ_τ. The third setting uses (iii) Gaussian mixture noise to further test misspecification of ϵ. Lastly, we consider the challenging case of (iv) a heteroscedastic Student-t noise with state-dependent degrees of freedom. In total, we have $3 \times 4 \times 4 \times 6$ experiments (indexed by dimensionality, noise setting, design heuristic and metamodel type).

Besides the noise distribution, we fix all other metamodeling aspects. All schemes are initialized with $n_0 = 10d$ inputs drawn from an LHS design on $[0, 1]^d$ and use the SE kernel (2.3) for the covariance matrix $\boldsymbol{K}$. To analyze for the variability due to the initial design and the noise realizations, we

Table 1 Response surfaces $x \mapsto f(x)$ for synthetic experiments

Quadratic (1-D)	$f(x) = (x + 0.75)(x - 0.75)$ with $x \in [0, 1]$
Modified Branin-Hoo (2-D)	$f(x) = \frac{1}{178}\left[\left(\bar{x}^1 - \frac{5.1(\bar{x}^2)^2}{4\pi^2} + \frac{5\bar{x}^2}{\pi} - 20\right)^2 + (10 - \frac{10}{8\pi})\cos(\bar{x}^1) - 181.47\right]$ with: $\bar{x}^1 = 15x^1$, $\bar{x}^2 = 15x^2 - 5$, $x^1, x^2 \in [0, 1]$
Modified Michalewicz (2-D)	$f(x) = 8 \times \left(-\sum_{i=1}^{2} \sin(\pi x^i)\sin^{20}\left(i\pi(x^i)^2\right) + 0.5\right)$ with $\mathbf{x} \in [0, 1]^2$
Modified Hartman6 (6-D)	$f(x) = \frac{-1}{0.1}\left[\sum_{i=1}^{4} C_i \exp\left(-\sum_{j=1}^{6} a_{ji}(x^j - p_{ji})^2\right) - 0.1\right]$ with: $\mathbf{C} = [0.2, 0.22, 0.28, 0.3]$ $\mathbf{a} = \begin{bmatrix} 8.00 & 0.50 & 3.00 & 10.00 \\ 3.00 & 8.00 & 3.50 & 6.00 \\ 10.00 & 10.00 & 1.70 & 0.50 \\ 3.50 & 1.00 & 8.00 & 8.00 \\ 1.70 & 6.00 & 10.00 & 1.00 \\ 6.00 & 9.00 & 6.00 & 9.00 \end{bmatrix}$, $\mathbf{p} = \frac{1}{10^4}\begin{bmatrix} 1312 & 2329 & 2348 & 4047 \\ 1696 & 4135 & 1451 & 8828 \\ 5569 & 8307 & 3522 & 8732 \\ 124 & 3736 & 2883 & 5743 \\ 8283 & 1004 & 3047 & 1091 \\ 5886 & 9991 & 6650 & 381 \end{bmatrix}$

Table 2 Stochastic simulation setup for synthetic experiments

Initial design	Latin hypercube sampling of size $n_0 = 10d$
Total budget n	$d = 1, n = 100$; $d = 2, n = 150$; $d = 6, n = 1000$
Test set size $M = \lvert\mathcal{D}\rvert$	$d = 1, M = 1000$; $d = 2, M = 500$; $d = 6, M = 1000$
Noise setting for ϵ_i	(i) t/small : $t_3(0, (0.1R_f)^2)$
	(ii) t/large : $t_3(0, (0.5R_f)^2)$
	(iii) Gsn/mix: 50/50 mix of $\mathcal{N}(0, (0.5R_f)^2)$ and $\mathcal{N}(0, R_f^2)$
	(iv) t/hetero : $t_{6-4x^1}(0, (0.4(4x^1 + 1))^2)$

$R_f \equiv \max_x f(x) - \min_x f(x) = 1$

perform 20 macro-runs of each design/acquisition function combination. For each run, the same initial inputs are used across all GP metamodels and designs; however, the initial $\mathbf{x}_{1:n_0}$ vary across runs.

Optimization of the Improvement Metric: We employed the cUCB, SUR, tMSE and gSUR criteria to maximize the improvement metric $\mathcal{I}$ and select the next input x_{n+1}. This maximization task is non-trivial in higher dimensions because $\mathcal{I}$ is frequently multimodal and can be flat around its local maxima. We use a genetic optimization approach as implemented in the `ga` (genetic algorithm) function in `MATLAB` using the default parameter settings. (We considered increasing the number of generations with no material impact on results.) This is a global, gradient-free optimizer that uses an evolutionary algorithm to explore the input space D.

Evaluation of Performance Metrics: Recall that evaluating the quality of $\partial\widehat{S}$ is based on $\mathcal{ER}$ and $\mathcal{E}$ from (2.16) and (2.17) that require integration over D. In practice, these are computed based on a weighted sum over a finite $\mathcal{D}$, $\widehat{\mathcal{E}} := \sum_{m=1}^{M} \Phi\big(\frac{-|\widehat{f}(x_m)|}{s(x_m)}\big)\mu(x_m)$ for a space-filling sequence $\mathcal{D} \equiv x_{1:M} \in D$ of test points. In 1-D experiments, $\mathcal{D}$ was an equispaced grid of size $M = 1000$. In higher dimensions, to avoid the use of a lot of test points that are required to ensure an accurate approximation, we adaptively pick $\mathcal{D}$ that targets the critical region close to the zero contour. To do so, we replace the integral with a weighted sum:

$$\mathcal{R} \simeq \frac{p_c}{M_1} \sum_{x_{1:M_1} \in D_1} \mathbb{I}(\operatorname{sign} f(x_m) \neq \operatorname{sign} \widehat{f}(x_m)) + \frac{(1-p_c)}{M_2} \sum_{x_{1:M_2} \in D_2} \mathbb{I}(\operatorname{sign} f(x_m) \neq \operatorname{sign} \widehat{f}(x_m)), \tag{4.1}$$

where $M = M_1 + M_2$ and the test locations $x_{1:M_1}$ and $x_{1:M_2}$ are subsampled from a large space-filling (scrambled Sobol) sequence on D. The weight p_c determines the relative volume of D_1 and $D_2 = D \backslash D_1$, where on $D_1 = \{x : f(x) \simeq 0\}$ we are close to the zero contour. In the experiments below, we use $M_1 = 0.8M$, $M_2 = 0.2M$, and $p_c = 0.4$, so that the density of test points close to ∂S is six times relative to those far from the zero contour. We employ the same strategy for speeding the evaluation of the posterior error $\mathcal{E}$.

Surrogate Inference: Values of hyperparameters $\boldsymbol{\vartheta}$ are crucial for a good performance of GP metamodels. We estimate $\boldsymbol{\vartheta}$ using maximum likelihood. Except for TP, all models are fitted with the open source package `GPstuff` (Vanhatalo et al. 2013) in `MATLAB`. TPs are fitted with the `hetGP` (Binois et al. 2018) package in `R`, with procedures matching the `MATLAB` ones as much as possible. Auxiliary tests did not reveal any significant effects from using other available tools for plain GPs and t-GP, such as `GPML` (Rasmussen and Nickisch 2010).

In principle, the hyperparameters $\boldsymbol{\vartheta}$ change at every step of the sequential design, in other words, whenever $\mathcal{A}_n$ is augmented with (x_{n+1}, y_{n+1}). To save time, however, we do not update $\boldsymbol{\vartheta}$ at each step. Instead, we first estimate the hyperparameters $\boldsymbol{\vartheta}$ based on the initial design $\mathcal{A}_{n_0}$ and then freeze them, updating their values only every few steps. Specifically, $\boldsymbol{\vartheta}$ is re-estimated at steps n_0+1, n_0+2, n_0+4, n_0+8, n_0+16, ... This is done solely for computational efficiency and is driven by the fact that in later stages ($n \gg 100$), inferred hyperparameters tend to change minimally step to step. We observed minimal impact on accuracy from doing so.

The lengthscales θ_i are the most significant for surrogate goodness of fit. A too-small lengthscale will make the estimated $\widehat{f}$ look "wiggly" and might lead to overfitting, while θ_i too large will fail to capture an informative shape of the true f and hence S. Since our input domain is always $[0, 1]^d$, we restrict $\theta_i \in [0.3, 2]\ \forall i$ to be on the order of the length of the sample space D.

Computational Overhead: All the considered metamodels are computationally more demanding than the baseline Gaussian GP. For t-GP and Cl-GP, additional cost arises due to the Laplace approximation. TP necessitates estimation of the parameter ν and also the computation of β in (2.14). In the experiments considered, the respective computation times were roughly double to triple relative to the Gaussian-noise GP. In terms of sequential design, cUCB, tMSE and gSUR have approximately equal overhead; SUR is significantly more expensive because it requires evaluating the sum in (3.5). Note that all heuristics include two expensive

steps: optimization for x_{n+1} and computation of $\widehat{f}^{(n)}$ and $s^{(n)}$ (and/or $\widehat{s}^{(n+1)}$).

Overall timing of the schemes is complicated because of the combined effects of n (design budget), M (size of test set), and the use of different software (some schemes run in `R` and others in `MATLAB`). Most important, the ultimate computation time is driven by the simulation cost of generating $Y(x)$-samples, which is trivial in the synthetic experiments but assumed to be large in the motivating context.

All of the computer programs that are used to produce results in this section and Sect. 5 are uploaded on Zenodo https://zenodo.org/record/4584456#.

4.2 Comparison of regression GP metamodels

Figure 3 shows the boxplots of the error rate $\mathcal{ER}$ of $\widehat{S}^{(N)}$ at the final design ($N = 100$ in 1-D; $N = 150$ in 2-D; $N = 1000$ in 6-D). The plots are sorted by noise settings and design strategies, facilitating comparison between the discussed metamodels. In Table 3, we list the best metamodel and design combination in each case. Several high-level observations can be made. First, we observe the limitations of the baseline Gaussian GP metamodel, which cannot tolerate too much model misspecification. As the noise structure gets more complex, the classical GP surrogate begins to show increasing strain; in the last $t/hetero$ setup, it is both unstable (widely varying performance across runs) and inaccurate, with error rates upward of 30% on "bad" runs. In addition, according to results shown in Table 3, across all of the twelve cases, besides the 1-D experiment with $t/small$ noise, the Gaussian-noise GP never performs as the best metamodel. This result is not surprising but confirms that the noise distribution is key for the contour-finding task and illustrates the non-robustness of the Gaussian observation model, due to which outliers strongly influence the inference.

Second, we document that the simple adjustment of using Student-t observations significantly mitigates the above issue. t-GP performs consistently and significantly better than Gaussian-noise GP in essentially all settings. This result is true even when both models are misspecified (the Gsn/mix and $t/hetero$ cases). The performance of t-GP was still better (though not statistically significantly so) when we tested it in the setting of homoscedastic Gaussian noise (see Table 3 in Supplementary Material Section F). The latter fact is not surprising—t-GP adaptively learns the degrees-of-freedom parameter ν and hence can "detect" Gaussian noise by setting ν to be large. Conversely, in heavy-tailed noise cases, the use of Student-t likelihood will effectively ignore outliers (O'Hagan 1979) and thus produce more accurate predictions than working with a Gaussian observation assumption. We find that t-GP can handle complex noise structures and offers a good choice for all-around performance, making it a good default selection for applications. It brings smaller error rate $\mathcal{ER}$, more stable hyperparameter estimation, less contour bias and tighter contour CI. Moreover, t-GP is significantly better than all the other GPs in eight of the twelve setups, indicating that t-GP is essentially the best out of all GP metamodels in most cases. Figure 4 shows the mean error rate $\mathcal{ER}$ (2.16) as a function of step n in the 6-D $t/large$ and $t/hetero$ experiments. This illustrates the learning rates of different models and schemes as data are collected. In both cases, t-GP starts from more accu-

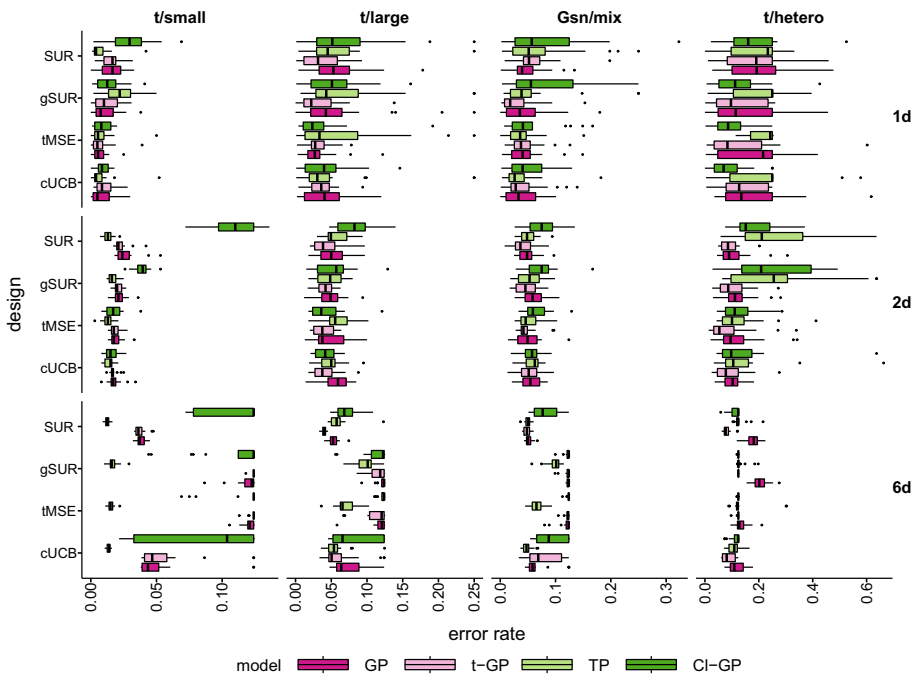


Fig. 3 Boxplots of final error rate $\mathcal{ER}^{(n)}$ from (2.16) across designs (rows) and noise setups (columns). Colors correspond to different GP metamodels. Note that x-axis limits are different across columns. Top row is for the 1-D experiment and design size $n = 100$; middle row: 2-D modified Branin-Hoo function with $n = 150$; bottom row: 6-D modified Hartman6 function with $n = 1000$

Table 3 Mean (w/standard deviation) error rate $\mathcal{ER}$ and corresponding best-performing sequential design heuristic for the 1-D exponential function, 2-D Modified Branin-Hoo function and 6-D Modified Hartman6 function synthetic case studies

Model	$t/small$		$t/large$		Gsn/mix		$t/hetero$	
1-D Quadratic								
GP	**tMSE**	0.73% (0.60%)	tMSE	3.24% (2.79%)	cUCB	3.87% (3.17%)	gSUR	15.68% (12.15%)
t-GP	tMSE	0.80% (0.93%)	**tMSE**	3.15% (1.83%)	**gSUR**	3.28% (3.74%)	gSUR	12.50% (9.05%)
TP	cUCB	0.97% (0.84%)	cUCB	5.93% (5.60%)	tMSE	5.09% (4.40%)	SUR	16.44% (10.14%)
Cl-GP	tMSE	0.87% (0.64%)	tMSE	3.39% (4.16%)	cUCB	4.99% (3.77%)	**cUCB**	8.83% (7.35%)
2-D Branin-Hoo								
GP	cUCB	1.78% (0.57%)	gSUR	4.75% (1.95%)	SUR	4.92% (1.86%)	cUCB	10.36 % (3.94%)
t-GP	cUCB	1.70% (0.29%)	**tMSE**	3.95% (1.47%)	**SUR**	4.10% (2.07%)	**tMSE**	9.00% (8.66%)
TP	**tMSE**	1.27% (0.41%)	cUCB	4.79% (1.84%)	SUR	5.19% (1.68%)	cUCB	12.75 % (9.02%)
Cl-GP	cUCB	1.56% (0.51%)	cUCB	4.27% (1.59%)	cUCB	5.71% (1.85%)	tMSE	13.23% (7.74%)
6-D Hartman6								
GP	SUR	3.81% (0.34%)	SUR	5.33% (0.54%)	SUR	5.19% (0.70%)	cUCB	11.67% (2.89%)
t-GP	SUR	3.75% (0.40%)	**SUR**	3.98% (0.47%)	**SUR**	4.86% (0.67%)	**SUR**	8.25% (1.60%)
TP	**SUR**	1.25% (0.20%)	cUCB	5.66% (1.98%)	cUCB	4.88% (0.88%)	cUCB	10.69% (2.34%)
Cl-GP	cUCB	7.99% (4.69%)	SUR	7.20% (0.66%)	SUR	8.31% (2.44%)	SUR	11.11% (2.20%)

Results are based on 20 macro-replications of each scheme. Best combinations for each column are indicated in bold

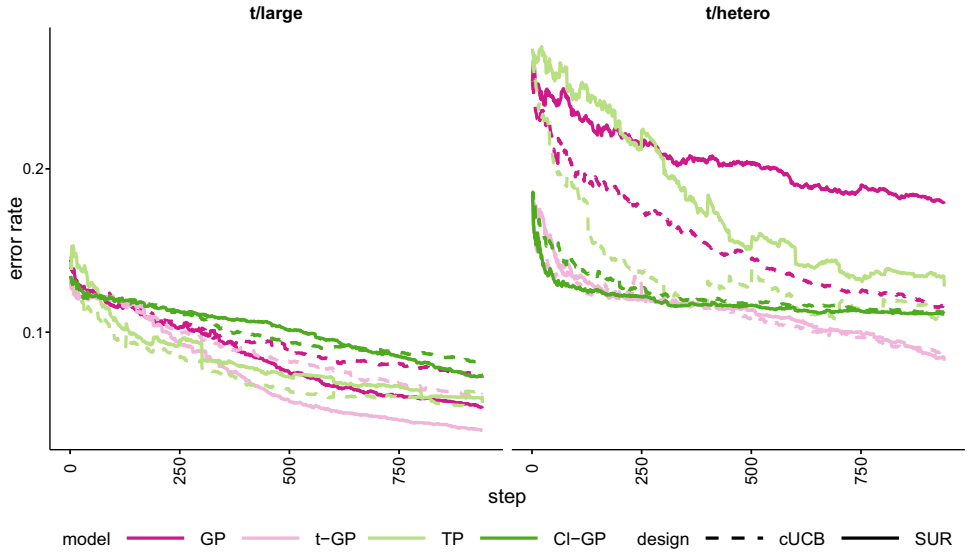


Fig. 4 Error rate $\mathcal{ER}^{(n)}$ (2.16) as a function of step n in the 6-D $t/large$ and $t/hetero$ settings. We compare four metamodels (colors) and two DoE's (line types). We plot mean results across 20 macro-replications of each scheme

rate estimation, especially in $t/hetero$ case where the initial error rate $\mathcal{ER}$ for t-GP is approximately two-thirds of that for GP. It learns the contour of interest over iterations, has a fast decreasing error rate $\mathcal{ER}$ and ends up with the smallest error rate in both cases.

Third, we also inspect the performance of the TP metamodel. As shown in Table 3, TP is the best in two cases out of the twelve, both of which are with the $t/small$ noise. We note that TP works worst in $t/hetero$ cases, having both large error rate $\mathcal{ER}$ and posterior error $\mathcal{E}$. Therefore, TP does not work well in cases with low SNR or greatly misspecified noise. This may be related to the parameterization of TPs, with noise handled in the kernel, which seems less robust to misspecification. Also, since TPs revert to GPs as n increases, the advantage of flexibility offered by the modeling decreases as iterations goes and thus may not last enough for low SNRs, which require more samples. It is apparent, for instance, in Fig. 4, where the learning rate at early stage (step < 250) is larger than for its counterparts.

Table 3 shows that there is no one overall "best" design for all metamodels across all cases. However, it does suggest some design/metamodel "combos" that work better than others. The classification GPs seem to prefer more aggressive designs, such as cUCB and tMSE, while t-GP prefers more exploratory designs, such as SUR, especially in higher

dimension. Additional discussion regarding the relative performance of design heuristics is in Sect. 4.4.

4.3 Classification GPs for level set estimation

We generally find that classification GPs are not the best metamodel. Namely, Cl-GP with cUCB design has the smallest error rate only in one ($t/hetero$ in 1-D) out of 12 cases shown in Table 3. However, Cl-GP is often competitive and is better than Gaussian-noise GP in some cases with tMSE and cUCB designs (except for the 6-D cases, where the error rate $\mathcal{ER}$ of cUCB is not significantly different from that of SUR, although mean of SUR is slightly smaller). There is significant improvement for models with low SNR; the only exception is for the low-noise setup where Cl-GP underperforms baseline GP. Figure 4 shows that in the 6-D $t/hetero$ case, Cl-GP achieves error rate $\mathcal{ER}$ which is two-thirds of $\mathcal{ER}$ for GP and TP at step 0. Also, it enjoys faster reduction in $\mathcal{ER}^{(n)}$, while the design size n is small. This matches the intuition that employing classification "flattens" the signal by removing outliers. By considering only the sign of the response, the classification model disregards its magnitude, simplifying the noise structure at the cost of some information loss. The net effect is helpful when the noise is misspecified or too strong so as to interfere with learning the mean response. It is detrimental if the above gain is outweighed by the information loss, as apparently happens in the 6-D experiments when the design size n is large.

We also observe that the stability of Cl-GP is highly dependent on the design: Some designs create large across-run variations in performance. We hypothesize that this is due to a more complex procedure for learning the hyperparameters of Cl-GP; therefore, designs that are not aggressive enough to explore the zero contour region (such as gSUR) face difficulties in estimating $\boldsymbol{\vartheta}$. As a result, relative to t-GP, Cl-GP has larger sampling variances.

To provide better intuition regarding the settings where Cl-GP works best, we test the performance of classification GP and regression GP on the 2-D modified Michalewicz function with Gsn/mix noise. Table 4 and Fig. 5 present the results comparing Cl-GP with cUCB against t-GP with SUR. The latter scheme was chosen as being the typical best-performing combination. We evaluate $\mathcal{ER}$ with budgets of $n = 150$ and $n = 500$. In both cases, Cl-GP performs better than t-GP, achieving significantly smaller error rate $\mathcal{ER}$. This is visualized in Fig. 5 that shows the estimated contours $\partial\widehat{S}$ at $n = 500$. While both models detect the vertical zero contour, only Cl-GP identifies the horizontal boundary and samples inputs in that neighborhood. Figure 5 also shows the error rate $\mathcal{ER}$ as a function of step for Cl-GP and t-GP. Cl-GP starts with a smaller $\mathcal{ER}$ and higher learning rate for both cUCB and SUR in the early stage (step < 200) compared with t-GP. Cl-GP with cUCB ends up with the most accurate estimate, echoing the results shown in Table 4.

Table 4 Mean (w/standard deviation) error rate $\mathcal{ER}$ for Cl-GP and t-GP with cUCB and SUR in 2-D modified Michalewicz synthetic experiments

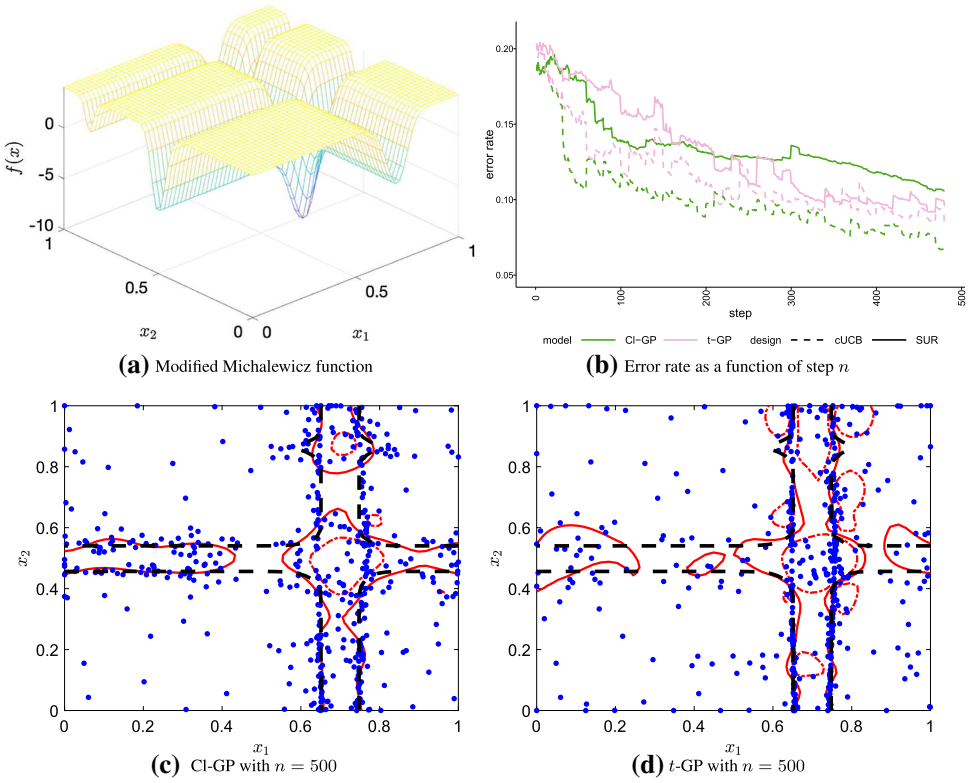
Design	Cl-GP	t-GP
Budget	$n = 150$	
cUCB	**10.77% (2.75%)**	13.14% (5.00%)
SUR	13.97% (1.30%)	15.50% (3.55%)
Budget	$n = 500$	
cUCB	**6.72% (2.33%)**	8.20% (0.92%)
SUR	10.60% (1.81%)	9.57% (1.41%)

Results are based on 20 macro-replications of each scheme

In conclusion, the modified Michalewicz function illustrates the potential gain of using Cl-GP over regression GPs where the covariance is not spatially stationary. The non-stationary space also benefits the localized criteria over the global ones, since the global change might be neutralized and may not provide an exact measure of uncertainty. This also explains why Cl-GP performs significantly better with cUCB than SUR.

4.4 Designs for contour-finding

A key goal of our study is qualitative insights about experimental designs most appropriate for noisy level set estimation. Through identifying the best-performing heuristics, we get an inkling regarding the structure of near-optimal designs for (1.1). In this section, we illustrate the latter within a 2-D setup that can be conveniently visualized. Taking the $t/large$ experiment as an example, in Fig. 6 we plot the fitted zero contour $\partial\widehat{S}$ at $N = 150$ together with the chosen inputs $\mathbf{x}_{1:150}$ across the six metamodels and the four $\mathcal{I}$ heuristics. As expected, most of the designs are around the contour ∂S, which is the intuitive approach to minimize the error $\mathcal{ER}$. Nevertheless, we observe significant differences in designs produced by different $\mathcal{I}$'s. The cUCB criterion places most of the samples close to the estimated zero contour $\partial\widehat{S}$, reflecting its aggressive exploitation nature. It leads to little information collected about other regions than the contour ∂S, especially around the boundaries of sample space D and hence relatively large $\bar{E}(x)$ there, inflating $\mathcal{E}$ for cUCB. As visualized in Fig. 7, the posterior error $\mathcal{E}$ tends to be the largest for cUCB. For tMSE, the samples tend to cluster at several subregions of $\partial\widehat{S}$ and on the edges of D. For gSUR, $\mathbf{x}_{1:n}$ cover a band along $\partial\widehat{S}$, resembling the shape of the cUCB design but more dispersed. Both approaches are better at reducing $\mathcal{E}$ compared with cUCB but are not directly aimed at this. For SUR, the design is much more exploratory, covering a large swath of D. All these findings echo the 1-D setting in Fig. 2.



(a) Modified Michalewicz function **(b)** Error rate as a function of step n

(c) Cl-GP with $n = 500$ **(d)** t-GP with $n = 500$

Fig. 5 **a** Modified Michalewicz function; **b** error rate $\mathcal{ER}^{(n)}$ (2.16) as a function of step n in the 2-D modified Michalewicz experiments; **c** and **d**: The estimated zero contour $\partial\widehat{S}$ (red solid line) and its 95% credible band (dot dashed lines) for Cl-GP and t-GP at $n = 500$. Blue dots are the samples selected by cUCB. Black dashed line is the true zero contour ∂S

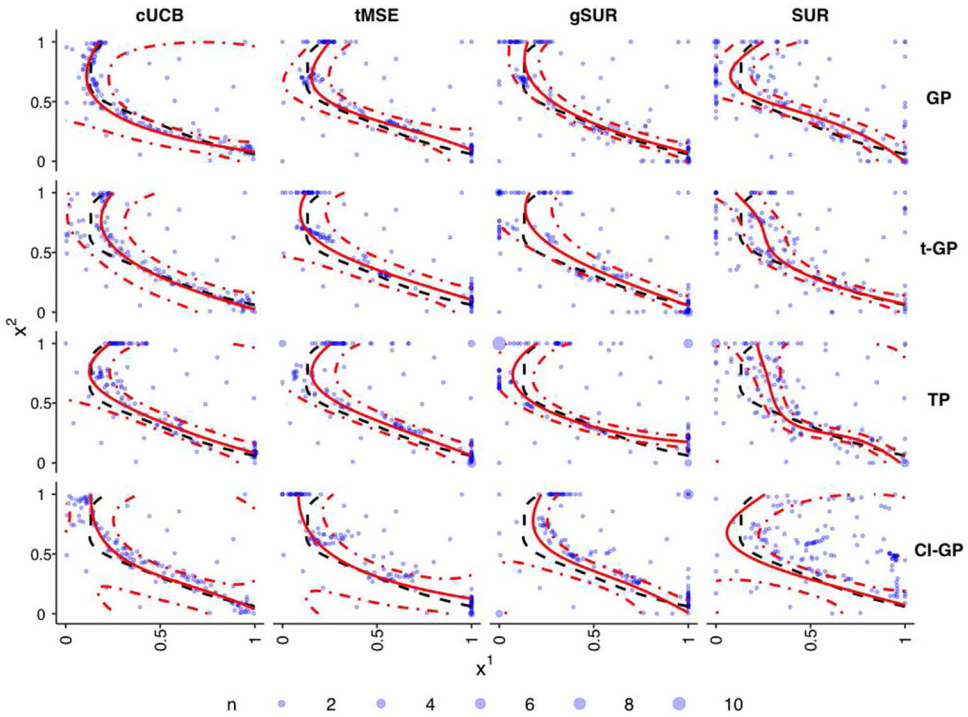


Fig. 6 Estimates of the zero contour $\partial\widehat{S}$ for the 2-D modified Branin-Hoo example with $t/large$ noise setting. We show $\partial\widehat{S}^{(n)}$ (red solid line) at step $n = 150$, with its 95% credible band (red dot dashed lines), the true zero contour ∂S (black dashed line) and the sampled inputs $\mathbf{x}_{1:n}$ (replicates indicated with larger symbols). We compare across the six metamodels (rows) and four DoE heuristics (columns)

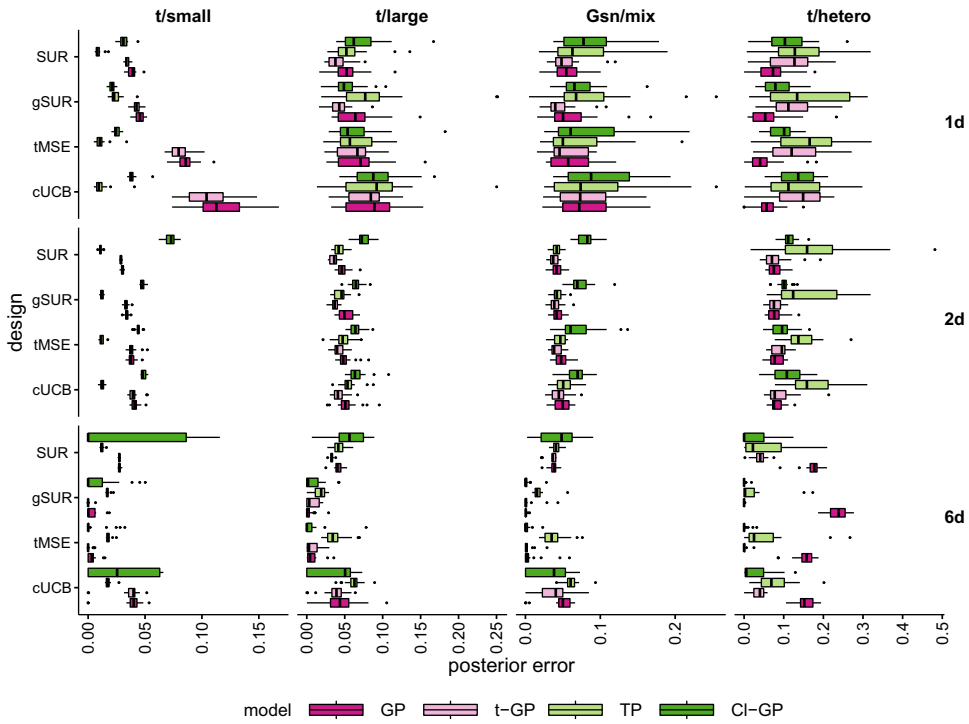


Fig. 7 Integrated posterior error $\mathcal{E}^{(n)}$ in Eq. (2.17) for GP, t-GP, TP and Cl- GP metamodels (colors), using cUCB, tMSE, gSUR and SUR-based designs (sub rows) with $n = 100$ in 1-D, $n = 150$ in 2-D and $n = 1000$ in the 6-D experiments (rows)

Another noteworthy feature is *replication* of some inputs, that is, repeated selection of the same $\mathbf{x}$ sites. This does not occur for cUCB, but happens for SUR, tMSE and gSUR that frequently (across algorithm runs) sample repeatedly at the vertices of D (indicated by the size of the corresponding marker in Fig. 6). The replication is typically mild. (We observe 145+ unique designs among a total of 150 x_n's.) This finding echoes (Binois et al. 2019) on the importance of replication to distinguish between signal and noise, which is a key distinction with the noise-free setting $\epsilon \equiv 0$.

Given the above discussion and the relative overhead of the different heuristics, we conclude that in lower-dimensional problems, there is little benefit to using the more sophisticated SUR criterion, while for higher-dimensional problems, SUR criterion is significantly better than the others. Beyond that, tMSE appears to be an adequate and cheaper choice. However, as the input space becomes more complicated (with a higher dimension or lower SNR observations), we need more exploration over the input space and the explorative criteria like SUR start to shine.

5 Application to optimal stopping problems in finance

In our next case study, we consider contour-finding for determining the optimal exercise policy of a Bermudan financial derivative. The underlying simulator is based on a d-dimensional geometric Brownian motion $(\boldsymbol{X}_t)$ that represents prices of d assets with interest rate r, dividend yield δ and volatility σ and follows the log-normal dynamics

$$\boldsymbol{X}_{t+\Delta t} = \boldsymbol{X}_t \exp\left((r - \delta - \frac{1}{2}\sigma^2)\Delta t + \Delta \boldsymbol{W}_t\right), \tag{5.1}$$

where $\Delta \boldsymbol{W}_t \sim \mathcal{N}(\mathbf{0}, \Delta t \boldsymbol{\Sigma})$ i.i.d. across t, and $\boldsymbol{\Sigma}$ is a given $d \times d$ covariance matrix. Let $h(t, x)$ be the option payoff from exercising when $\boldsymbol{X}_t = x \in \mathbb{R}^d$. The Bermudan option pricing problem consists of maximizing the expected reward $h(\tau, X_\tau)$ over all stopping times $\tau \in \{0, \Delta t, 2\Delta t, \ldots, T\}$ (exercising is allowed every Δt time units) bounded by the specified horizon T:

$$V(t, x) := \sup_{\tau \geq t, \tau \in \mathcal{S}} \mathbb{E}[h(\tau, X_\tau) | X_t = x]. \tag{5.2}$$

The approach in the so-called regression Monte Carlo (RMC) methods (Longstaff and Schwartz 2001; Tsitsiklis and Van Roy 2001) is to convert the decision of whether to exercise the option $\tau(t, x) = t$ or continue $\tau(t, x) > t$, when $X_t = x$ at intermediate step t, into comparing the immediate reward $h(t, x)$ vis à vis the reward-to-go $C(t, x)$. The optimal strategy is to dynamically pick the action with the higher expected payoff and is equivalent to determining the zero level set (known as the stopping region) $S_t = \{x \in D : f(x; t) \leq 0\}$ of the timing value $f(x; t) := C(t, x) - h(t, x)$. During backward dynamic programming, we iterate over $t = T, T - \Delta t, \ldots, 0$, recursively estimating the respective $\widehat{S}_t$. To do so, we simulate trajectories $X^x_{t:T}$ emanating from input x; the simulator $Y(x; t)$ returns the difference between

the pathwise payoff based on the exercise strategy summarized by the forward-looking $\{\widehat{S}_s, s > t\}$ along the trajectory $(X^x_{t:T})$, and $h(t, x)$.

We refer to Ludkovski (2018) for the full details of employing a GP metamodel for learning the timing value $f(\cdot; t)$; as noted there this setting implies a skewed, non-Gaussian, heteroscedastic distribution of the simulation noise ϵ and is a challenging stochastic contour-finding problem. Note that in order to reflect the underlying distribution of $\boldsymbol{X}_t$ at time t (conditional on the given initial value $\boldsymbol{X}_0 = x_0$), the weighting measure $\mu(x) = p_{X_t}(x|x_0)$ is used. Thus, $\mu(\cdot)$ is log-normal based on (5.1). In line with the problem context, we no longer directly measure the accuracy of learning $\{S_t\}$ but instead focus on the ultimate output of RMC, which is the estimated option value in (5.2). The latter must itself be numerically evaluated via an out-of-sample Monte Carlo simulation that averages realized payoffs along a large database of M paths $x^{1:M}_{0:T}$:

$$\widehat{V}(0, x_0) = \frac{1}{M}\sum_{m=1}^{M} h(\tau^m, x^{(m)}_{\tau^m}),$$
$$\tau^m = \inf\{t : x^{(m)}_t \in \widehat{S}_t\}. \tag{5.3}$$

Since our goal is to find the *best* exercise strategy, higher $\widehat{V}$'s indicate a better approximation of the ground truth $\{S_t\}$.

To allow a direct comparison, we set parameters matching the test cases in Ludkovski (2018), considering a 2-D and 3-D example. In both cases, the volatility matrix $\boldsymbol{\Sigma} = \sigma \boldsymbol{I}$ in (5.1) is diagonal with constant terms; that is, the coordinates $\mathbf{X}^1_{1:n}, \ldots, \mathbf{X}^d_{1:n}$ are independently and identically distributed. As a first example, we consider a 2-D basket Put option with parameters $r = 0.06, \sigma = 0.2, \delta = 0, \Delta t = 0.04, \mathcal{K} = 40, T = 1$. The payoff is $h(t, x) = e^{-rt}(\mathcal{K} - \frac{x^1+x^2}{2})_+$ with $\mathcal{K} = 40$. Here, it is known that stopping becomes optimal once both asset prices x^1 and x^2 become sufficiently low, so the level set S_t is toward the bottom-left of D; see Fig. 8. In contrast, stopping is definitely suboptimal when $h(t, x) = 0 \Leftrightarrow (x^1+x^2)/2 > \mathcal{K}$. Consequently, the input sample space is taken to be $D = [25, 55] \times [25, 55] \cap \{x^1 + x^2 \le 80\}$.

In this first case study, the timing value $f(x; t)$ is known to be *monotonically* increasing in the asset price x. To incorporate this constraint, we augment the four main metamodels (GP, t-GP, Cl-GP and TP) with two monotonic versions, M-GP and MCl-GP. By constraining the fitted $\widehat{f}$ to be monotone, we incorporate structural knowledge about the ground truth, which in turn reduces posterior uncertainty and thus might produce more accurate estimates of S. Monotonicity of the metamodel for f is also one sufficient way to guarantee that the outputted level set $\widehat{S}$ is a *connected* subset of D.

Our monotone GPs are based on Riihimäki and Vehtari (2010). In general, any infinite-dimensional Gaussian process is intrinsically non-monotone, since the multivariate Gaussian distribution is always supported on the entire $\mathbb{R}^d$, rather than an orthant. Nevertheless, local monotonicity in $\widehat{f}$ can be enforced by considering the gradient ∇f of f which is also a Gaussian process. Specifically, Riihimäki and Vehtari (2010) proposed to adaptively add virtual observations for ∇f; we employ the resulting implementation in the public `GPstuff` library (Vanhatalo et al. 2013) to build our own version dubbed M-GP. We employ the same strategy to restrict the coordinates z^j of the latent probit GP Z to be increasing (decreasing) across D. Implementation details are included in Supplementary Material Section D.

As a second example, we consider a 3-D max-Call $x \in \mathbb{R}^3$ with payoff $h(t, x) = e^{-rt}(\max(x^1, x^2, x^3) - \mathcal{K})_+$. The parameters are $r = 0.05, \delta = 0.1, \sigma = 0.2, X_0 = (90, 90, 90), \mathcal{K} = 100, T = 3$ and $\Delta t = 1/3$. Since stopping is ruled out when $h(t, x) = 0 \Leftrightarrow \max(x^1, x^2, x^3) < \mathcal{K}$, the sample space is taken to be $D = [50, 150]^3 \cup \{\max(x^1, x^2, x^3) > \mathcal{K}\}$. In this case, stopping is optimal if *one* of the coordinates x^i is significantly higher than the other two, so S_t consists of three disconnected components. In this problem, there is no monotonicity, so we employ only the GP, t-GP, Cl-GP and TP metamodels.

Because of the iterative construction of the simulator, the SNR gets low for small t's. The variance $\tau^2(x)$ is also highly state dependent, tending to be smaller for sites further from the zero contour. While in this case study, the simulations are very fast, low SNR requires many hundreds of observations to reliably detect the level set. Since the expense of sequential design of GP metamodels comes mainly from choosing the new input at each step, it is impractical to have such a large n. To reduce metamodel overhead, we employ *batched* designs (Ludkovski 2018; Ankenman et al. 2008), reusing $x \in D$ for r replications to collect observations $y^{(1)}(x), \ldots, y^{(r)}(x)$ from the corresponding simulator $Y(x)$. Then, we treat the mean of the r observations,

$$\bar{y}(x) = \frac{1}{r}\sum_{i=1}^{r} y^{(i)}(x), \tag{5.4}$$

as the response for input x and use $(x, \bar{y}(x))$ as a single design entry. Such reduction by a factor of r in the number of unique inputs $n = N/r$ significantly speeds fitting and updating. Moreover, the statistical properties of $\bar{y}$ are improved thanks to the central limit theorem (CLT): Noise variance $\bar{\tau}^2(x) = \tau^2(x)/r$ is much smaller, and its distribution is more Gaussian. In the case studies below, we observe raw skewness in the range of $[-2, 1]$ and raw kurtosis in the range of $[2, 20]$; after batching skewness decreases by a factor of $\sqrt{r}$ and excess kurtosis mostly disappears for $r \ge 10$. Nevertheless, unless r is in the dozens, the distribution of $\bar{y}$

Table 5 Performance of different designs and models on the 2-D Bermudan Put option in Sect. 5

	LHS	cUCB	tMSE	gSUR	SUR
$r = 3, n = 80$					
GP	1.211 (0.120)	1.425 (0.008)	1.427 (0.007)	1.431 (0.009)	**1.431 (0.007)**
t-GP	1.125 (0.113)	1.409 (0.013)	**1.417 (0.008)**	1.409 (0.010)	1.406 (0.013)
TP	1.179 (0.133)	1.408 (0.022)	**1.414 (0.008)**	1.378 (0.044)	1.316 (0.037)
M-GP	1.403 (0.014)	1.438 (0.007)	1.440 (0.006)	**1.442 (0.009)**	1.433 (0.005)
Cl-GP	1.111 (0.121)	1.395 (0.015)	**1.402 (0.013)**	1.393 (0.013)	1.391 (0.013)
MCl-GP	1.407 (0.008)	1.429 (0.010)	1.429 (0.013)	**1.431 (0.007)**	1.396 (0.019)
$r = 15, n = 80$					
GP	1.425 (0.017)	1.448 (0.003)	**1.450 (0.002)**	1.450 (0.003)	1.449 (0.003)
t-GP	1.406 (0.033)	1.445 (0.003)	**1.447 (0.002)**	1.444 (0.005)	1.446 (0.004)
TP	1.414 (0.023)	**1.443 (0.003)**	1.443 (0.004)	1.441 (0.004)	1.430 (0.006)
M-GP	1.407 (0.008)	1.449 (0.003)	1.451 (0.002)	**1.454 (0.002)**	1.451 (0.003)
Cl-GP	1.353 (0.050)	**1.441 (0.004)**	1.440 (0.003)	1.435 (0.004)	1.436 (0.005)
MCl-GP	1.416 (0.010)	1.448 (0.004)	**1.449 (0.003)**	1.443 (0.003)	1.418 (0.008)
$r = 48, n = 25$					
GP	1.341 (0.068)	**1.450 (0.003)**	1.449 (0.003)	1.443 (0.004)	1.448 (0.003)
t-GP	1.336 (0.126)	1.449 (0.003)	**1.452 (0.003)**	1.442 (0.004)	1.449 (0.003)
TP	1.367 (0.063)	**1.433 (0.006)**	1.430 (0.011)	1.421 (0.039)	1.423 (0.023)
M-GP	1.415 (0.007)	**1.446 (0.002)**	1.444 (0.002)	1.445 (0.004)	1.442 (0.004)
Cl-GP	1.110 (0.144)	1.430 (0.010)	**1.434 (0.005)**	1.409 (0.008)	1.388 (0.016)
MCl-GP	1.423 (0.015)	1.446 (0.004)	**1.448 (0.003)**	1.413 (0.024)	1.414 (0.024)

Results are the mean (standard deviation) payoff of 25 runs of experiments evaluating on the same out-of-sample testing set of $M = 160{,}000$ $\boldsymbol{X}_{0:T}$-paths at each run. Best combinations indicated in bold. Best designs highlighted in gray

remains heteroscedastic and highly state dependent in both x and t, making a Gaussian-noise GP strongly misspecified.

For the 2-D Put case study, we then test a total of three budget settings: (i) $r = 3, n = 80$ (low budget of $N = 240$ simulations); (ii) $r = 15, n = 80$ (high budget $N = 800$ with moderate replication); and (iii) $r = 48, n = 25$ (high $N = 800$ with high replication). Comparing (ii) and (iii) shows the competing effects of having non-Gaussian noise (for lower r) and small design size (low n). The initial design size is $n_0 = 10$. In this example, taking $n \gg 80$ gives only marginally better performance but significantly raises the computation time and hence is ruled out as impractical. Three setups are investigated for the 3-D example: $r = 3, n = 100$ (low-budget of $N = 300$), $r = 20, n = 100$ (moderate-budget of $N = 2000$) and $r = 20, n = 200$ (high budget $N = 4000$), both with $n_0 = 30$. In all examples, the results are based on 25 runs of each scheme and are evaluated through the resulting expected reward $\widehat{V}(0, x_0)$ (5.3) on a fixed out-of-sample testing set of $M = 160{,}000$ paths of $\boldsymbol{X}_{0:T}$.

The results for Gaussian-noise GP, t-GP and TP can be reproduced via the publicly available mlOSP `R` package at http://github.com/mludkov/mlOSP, which in particular implements all the discussed acquisition functions.

5.1 Results

Tables 5 and 6 compare the different designs and metamodels. To assess the sequential design gains, we also report the results from using a baseline non-adaptive LHS design on D. At low budget, we observe the dramatic gains of using adaptive designs for level set estimation, which allow us to obtain the same performance with an order-of-magnitude smaller simulation budget. The tMSE and gSUR criteria work best for the 2-D Put, while SUR is the best for the 3-D max-Call, indicating that the exploratory designs start to win out in more complex settings with higher d.

Regarding the metamodels, in the low-budget setups, the monotonic GP metamodel works best for the 2-D Put and t-GP for the 3-D max-Call. For the higher budget, which also coincides with higher $r \in \{10, 50\}$, the metamodel performance is similar, with t-GP slightly better than the other GP variants. In particular, once the SNR is high, classical Gaussian GP is effectively as good as any alternative. In both examples, TP and classification metamodels do not work well, possibly because of being more sensitive to the heteroscedastic aspect. We note that TP as well as the classification metamodels suffers from instability, so that lower $\widehat{V}(0, x_0)$ is matched with a high sampling standard devia-

Table 6 Performance of different designs and models on the 3-D Bermudan max-Call in Sect. 5

	LHS	cUCB	tMSE	gSUR	SUR
$r = 3, n = 100$					
GP	10.036 (0.331)	10.725 (0.095)	**10.773 (0.071)**	10.711 (0.086)	10.753 (0.072)
t-GP	9.894 (0.447)	10.736 (0.088)	10.747 (0.087)	10.720 (0.104)	**10.782 (0.076)**
TP	9.169 (0.354)	10.101 (0.218)	9.872 (0.102)	8.867 (0.357)	**10.482 (0.156)**
Cl-GP	9.552 (0.567)	10.566 (0.084)	**10.657 (0.097)**	10.586 (0.099)	10.604 (0.119)
$r = 20, n = 100$					
GP	10.924 (0.076)	11.078 (0.029)	11.072 (0.028)	11.055 (0.032)	**11.101 (0.023)**
t-GP	10.923 (0.071)	11.061 (0.039)	11.055 (0.027)	11.044 (0.029)	**11.100 (0.027)**
TP	10.385 (0.178)	**10.815 (0.039)**	10.745 (0.045)	10.620 (0.087)	10.507 (0.087)
Cl-GP	10.761 (0.112)	**11.026 (0.032)**	10.991 (0.037)	10.901 (0.049)	10.937 (0.041)
$r = 20, n = 200$					
GP	11.105 (0.036)	11.147 (0.021)	11.119 (0.022)	11.131 (0.018)	**11.178 (0.020)**
t-GP	11.090 (0.034)	11.141 (0.019)	11.126 (0.020)	11.115 (0.027)	**11.175 (0.021)**
TP	10.585 (0.118)	**10.896 (0.030)**	10.811 (0.035)	10.764 (0.041)	10.638 (0.038)
Cl-GP	10.995 (0.059)	**11.109 (0.025)**	11.056 (0.040)	10.985 (0.027)	11.010 (0.029)

Results are the mean (w/standard deviation) payoff of 25 macro-replications evaluating on the same out-of-sample testing set of $M = 160{,}000$ $\boldsymbol{X}_{0:T}$-paths at each run. Best combinations indicated in bold. Best designs highlighted in gray.

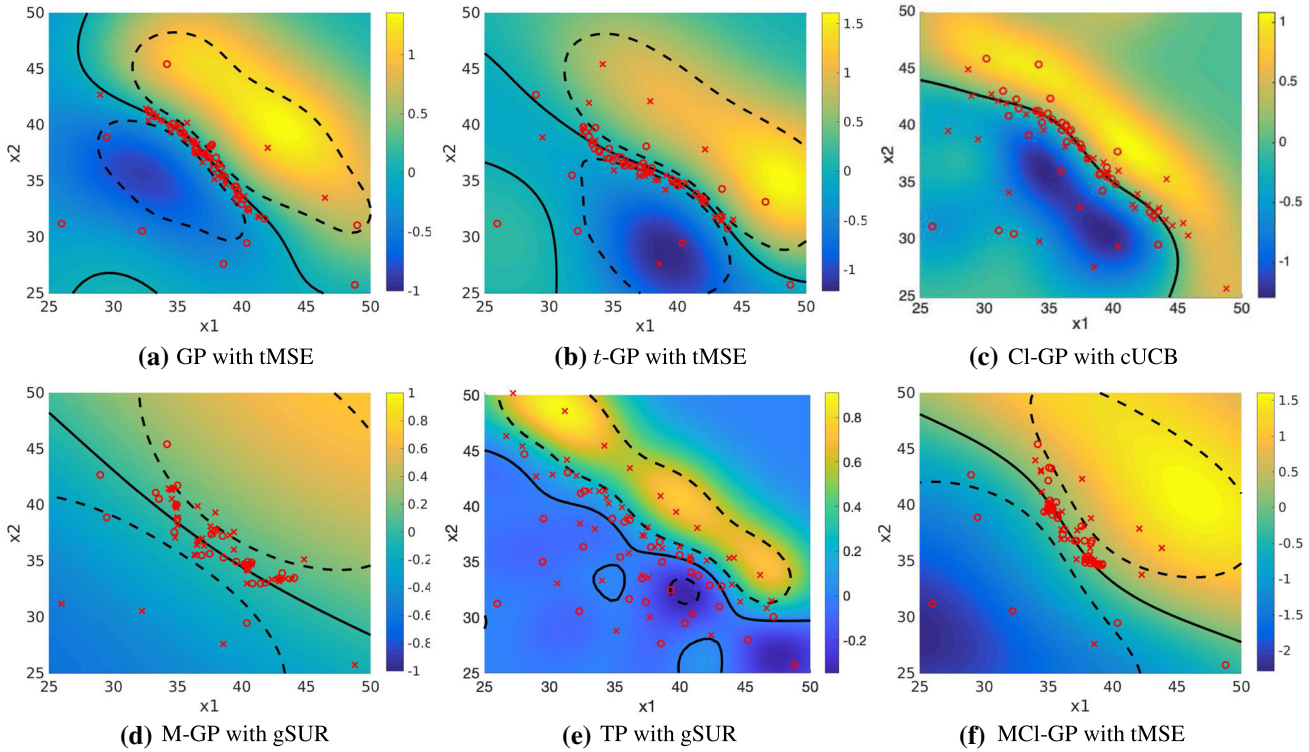


Fig. 8 The estimated exercise boundary $\partial\widehat{S}$ (solid line with 95% CI as dashed lines) at $t = 0.6$ for 2-D Bermudan Put from Sect. 5. Shading, which varies panel to panel, indicates the point estimate for the latent $\widehat{f}(x)$ or $\widehat{z}(x)$. We also show the design $(\mathbf{x}_{1:n}, \mathbf{y}_{1:n})$ with positive y_n's marked by $\times$ and negative y_n's by $\circ$. All schemes used $r = 15$, $n = 80$. **a** GP with tMSE. **b** t-GP with tMSE. **c** Cl-GP with cUCB. **d** M-GP with gSUR. **e** TP with gSUR. **f** MCl-GP with tMSE

tion. Another observation is that Cl-GP and MCl-GP perform poorly with an exploratory heuristic like SUR, especially with high budget, which echoes conclusions in Sects. 4.3 and 4.4. Due to the strong skewness of the underlying simulator, classification GP is biased which explains its poor performance for low $r = 3$ in (5.4).

Figure 8 shows the estimated exercise boundary $\partial\widehat{S}_t$ with its 95% CI at $t = 0.4$ for the 2-D Put, for each of the five

metamodels, each with the design yielding the highest payoff. We observe that all the best-performing designs look similar, placing about a dozen x_n's (some of which are from the initial design $x_{1:n_0}$) throughout D and the rest tightly along the zero contour. The results suggest that the criteria are largely interchangeable and that simpler $\mathcal{I}_n$ heuristics are able to reproduce the features of the more sophisticated or expensive SUR. The heuristics *do* differ in their uncertainty quantification; t-GP and GP generate tightest CI bands, while those of classification GPs and TP are too wide, indicating lack of confidence in the estimate. Of note, the regression GP metamodels (GP, t-GP and M-GP) also generate the lowest sampling variance for $\widehat{V}(0, x_0)$.

Based on these results, our take-aways are threefold. First, similar to Ludkovski (2018) we document significant gains from sequential design. Second, we find that while using SUR is helpful for more complicated settings with higher dimension d and larger budget, tMSE is the recommended DoE heuristic for lower-dimensional cases, achieving excellent results with minimal overhead (in particular without requiring look-ahead variance). Third, we find that for applications with thousands of simulations, the Gaussian observation model is sufficient, since the underlying design needs to be replicated $r \gg 1$ in order to avoid excessively large $\mathbf{K}$-matrices. Therefore, there is little need for more sophisticated metamodels, although useful gains can be realized from enforcing the monotonic structure, if available.

6 Conclusion

We have carried a comprehensive comparison of five metamodels and four design heuristics on 19 case studies ($4\times3+1$ synthetic, plus six real worlds). In sum, the considered alternatives to standard Gaussian-observation GP do perform somewhat better. In particular, t-GP directly nests plain GP and hence essentially always matches or exceeds the performance of the latter. We also observe gains from using Cl-GP when SNR is low or the response covariance is not spatially stationary and from monotonic surrogates when the underlying response is monotone. That being said, final recommendation regarding the associated benefit depends on computational considerations, as the respective overhead becomes larger (and exact updating of the metamodel no longer possible).

In terms of design, we advocate the benefits of tMSE in low-dimensional simulations, which generates high-performing experimental designs without requiring expensive acquisition function (or even look-ahead variance). The tMSE criterion does sometimes suffer from the tendency to put many designs at the edge of the input space but otherwise tends to match the performance of more complex and computationally intensive $\mathcal{I}_n$'s. For complex simulations, SUR is probably still the best choice (although in that case, random-set-based heuristics should also be considered). Particularly in higher dimensions with misspecified noise, SUR is the best choice among all designs for t-GP. We also stress that the user ought to thoughtfully pick the *combination* of sequential design and metamodel, since cross-dependencies are involved (e.g., classification metamodels generally do not work well with the SUR criterion in lower dimension).

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11222-021-10014-w.

Acknowledgements XL and ML are partially supported by NSF DMS-1521743 and DMS-1821240. The work of MB is partially supported by NSF DMS-1521702 and the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under Contract No. DE-AC02-06CH11357. The authors also thank two anonymous reviewers for their helpful suggestions.

References

Ankenman, B., Nelson, B.L., Staum, J.: Stochastic kriging for simulation metamodeling. In: 2008 Winter Simulation Conference, pp. 362–370. IEEE (2008)

Azzimonti, D., Bect, J., Chevalier, C., Ginsbourger, D.: Quantifying uncertainties on excursion sets under a Gaussian random field prior. SIAM/ASA J. Uncertain. Quantif. **4**(1), 850–874 (2016)

Baker, E., Barbillon, P., Fadikar, A., Gramacy, R.B., Herbei, R., Higdon, D., Huang, J., Johnson, L.R., Mondal, A., Pires, B., et al.: Analyzing stochastic computer models: an overview with opportunities. Technical report (2020). arXiv preprint arXiv:2002.01321

Bect, J., Ginsbourger, D., Li, L., Picheny, V., Vazquez, E.: Sequential design of computer experiments for the estimation of a probability of failure. Stat. Comput. **22**(3), 773–793 (2012)

Bect, J., Li, L., Vazquez, E.: Bayesian subset simulation. SIAM/ASA J. Uncertain. Quantif. **5**(1), 762–786 (2017)

Bichon, B.J., Eldred, M.S., Swiler, L.P., Mahadevan, S., McFarland, J.M.: Efficient global reliability analysis for nonlinear implicit performance functions. AIAA J. **46**(10), 2459–2468 (2008)

Binois, M., Gramacy, R.B., Ludkovski, M.: Practical heteroskedastic Gaussian process modeling for large simulation experiments. J. Comput. Graph. Stat. **27**(4), 808–821 (2018)

Binois, M., Huang, J., Gramacy, R.B., Ludkovski, M.: Replication or exploration? Sequential design for stochastic simulation experiments. Technometrics **61**(1), 7–23 (2019)

Bogunovic, I., Scarlett, J., Krause, A., Cevher, V.: Truncated variance reduction: a unified approach to Bayesian optimization and level-set estimation. In: Advances in Neural Information Processing Systems, pp. 1507–1515 (2016)

Bryan, B., Schneider, J.: Actively learning level-sets of composite functions. In: Proceedings of the 25th International Conference on Machine Learning, pp. 80–87. ACM (2008)

Bryan, B., Nichol, R.C., Genovese, C.R., Schneider, J., Miller, C.J., Wasserman L.: Active learning for identifying function threshold boundaries. In: Advances in Neural Information Processing Systems, pp. 163–170 (2006)

Chevalier, C.: Fast uncertainty reduction strategies relying on Gaussian process models. Ph.D. thesis, Universitat Bern (2013)

Chevalier, C., Ginsbourger, D., Bect, J., Molchanov, I.: Estimating and quantifying uncertainties on level sets using the Vorob'ev expectation and deviation with Gaussian process models. In: mODa 10—Advances in Model-Oriented Design and Analysis, pp. 35–43. Springer (2013)

Chevalier, C., Bect, J., Ginsbourger, D., Vazquez, E., Picheny, V., Richet, Y.: Fast parallel kriging-based stepwise uncertainty reduction with application to the identification of an excursion set. Technometrics **56**(4), 455–465 (2014)

Echard, B., Gayton, N., Lemaire, M.: Kriging based Monte Carlo simulation to compute the probability of failure efficiently: AK-MCS method. 6emes Journées Nationales de Fiabilité, 24–26 mars, Toulouse, France (2010)

Frazier, P.I., Powell, W.B., Dayanik, S.: A knowledge-gradient policy for sequential information collection. SIAM J. Control. Optim. **47**(5), 2410–2439 (2008)

Gotovos, A., Casati, N., Hitz, G., Krause, A.: Active learning for level set estimation. In: Twenty-Third International Joint Conference on Artificial Intelligence (2013)

Gramacy, R.B., Lee, H.K.: Adaptive design and analysis of supercomputer experiments. Technometrics **51**(2), 130–145 (2009)

Gramacy, R.B., Ludkovski, M.: Sequential design for optimal stopping problems. SIAM J. Financ. Math. **6**(1), 748–775 (2015)

Hu, R., Ludkovski, M.: Sequential design for ranking response surfaces. SIAM/ASA J. Uncertain. Quantif. **5**(1), 212–239 (2017)

Jalali, H., Van Nieuwenhuyse, I., Picheny, V.: Comparison of kriging-based algorithms for simulation optimization with heterogeneous noise. Eur. J. Oper. Res. **261**(1), 279–301 (2017)

Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. J. Global Optim. **13**(4), 455–492 (1998)

Jylänki, P., Vanhatalo, J., Vehtari, A.: Robust Gaussian process regression with a Student-t likelihood. J. Mach. Learn. Res. **12**(Nov), 3227–3257 (2011)

Longstaff, F.A., Schwartz, E.S.: Valuing American options by simulation: a simple least-squares approach. Rev. Financ. Stud. **14**(1), 113–147 (2001)

Ludkovski, M.: Kriging metamodels for Bermudan option pricing. J. Comput. Finance **22**(1), 1–42 (2018)

Mukhopadhyay, S., Mahmoodi, H., Roy, K.: Modeling of failure probability and statistical design of SRAM array for yield enhancement in nanoscaled CMOS. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **24**(12), 1859–1880 (2005)

O'Hagan, A.: On outlier rejection phenomena in Bayes inference. J. R. Stat. Soc. Ser. B (Methodol.) **41**, 358–367 (1979)

Picheny, V., Ginsbourger, D., Roustant, O., Haftka, R.T., Kim, N.H.: Adaptive designs of experiments for accurate approximation of a target region. J. Mech. Des. **132**(7), 071008 (2010)

Picheny, V., Ginsbourger, D., Richet, Y., Caplin, G.: Quantile-based optimization of noisy computer experiments with tunable precision. Technometrics **55**(1), 2–13 (2013a)

Picheny, V., Wagner, T., Ginsbourger, D.: A benchmark of kriging-based infill criteria for noisy optimization. Struct. Multidiscip. Optim. **48**(3), 607–626 (2013b)

Ranjan, P., Bingham, D., Michailidis, G.: Sequential experiment design for contour estimation from complex computer codes. Technometrics **50**(4), 527–541 (2008)

Rasmussen, C.E., Nickisch, H.: Gaussian processes for machine learning (GPML) toolbox. J. Mach. Learn. Res. **11**(Nov), 3011–3015 (2010)

Rasmussen, C.E., Williams, C.K.: Gaussian Processes for Machine Learning. MIT Press, Cambridge (2006)

Riihimäki, J., Vehtari, A.: Gaussian processes with monotonicity information. AISTATS **9**, 645–652 (2010)

Santner, T.J., Williams, B.J., Notz, W.I.: The Design and Analysis of Computer Experiments. Springer, Berlin (2013)

Scott, C., Davenport, M.: Regression level set estimation via cost-sensitive classification. IEEE Trans. Signal Process. **55**(6), 2752–2757 (2007)

Shah, A., Wilson, A., Ghahramani, Z.: Student-t processes as alternatives to Gaussian processes. In: Artificial Intelligence and Statistics, pp. 877–885 (2014)

Srinivas, N., Krause, A., Kakade, S.M., Seeger, M.W.: Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. IEEE Trans. Inf. Theory **58**(5), 3250–3265 (2012)

Tsitsiklis, J.N., Van Roy, B.: Regression methods for pricing complex American-style options. IEEE Trans. Neural Netw. **12**(4), 694–703 (2001)

Vanhatalo, J., Jylänki, P., Vehtari, A.: Gaussian process regression with Student-t likelihood. In: Advances in Neural Information Processing Systems, pp. 1910–1918 (2009)

Vanhatalo, J., Riihimäki, J., Hartikainen, J., Jylänki, P., Tolvanen, V., Vehtari, A.: Bayesian modeling with Gaussian processes using the GPstuff toolbox. J. Mach. Learn. Res. **14**(1), 1175–1179 (2013)

Vazquez, E., Bect, J.: A sequential Bayesian algorithm to estimate a probability of failure. IFAC Proc. Vol. **42**(10), 546–550 (2009)

Vazquez, E., Martinez, M.P.: Estimation of the volume of an excursion set of a Gaussian process using intrinsic kriging (2006). arXiv preprint, arXiv:math/0611273

Wang, Z., Shi, J.Q., Lee, Y.: Extended t-process regression models. J. Stat. Plan. Inference **189**, 38–60 (2017)

Willett, R.M., Nowak, R.D.: Minimax optimal level-set estimation. IEEE Trans. Image Process. **16**(12), 2965–2979 (2007)

Williams, C.K., Barber, D.: Bayesian classification with Gaussian processes. IEEE Trans. Pattern Anal. Mach. Intell. **20**(12), 1342–1351 (1998)

Yang, J., Wang, Z., Wu, J.: Level set estimation with dynamic sparse sensing. In: 2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP), pp. 487–491. IEEE (2014)