Adaptive Probability of Failure Estimation with Gaussian Processes

Aleksei G. Sorokin[1,2]
asorokin@hawk.iit.edu

Vishwas Rao[2]

[1]Illinois Institute of Technology, Department of Applied Mathematics.
[2]Argonne National Laboratory, Mathematics and Computer Science Division.

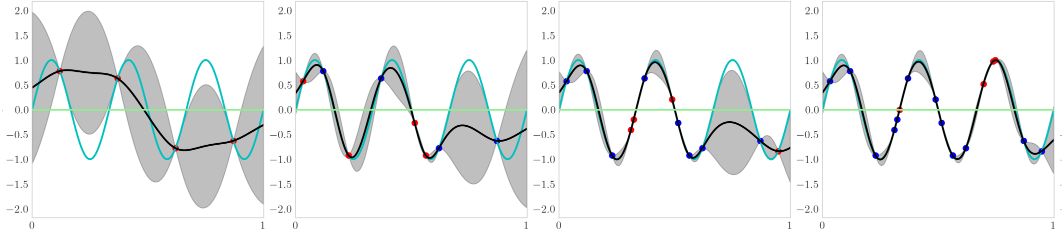
Background

Goal Estimate a small probability of failure to desired error tolerance

Failure A expensive simulation exceeds a threshold

Uses Reliability certification, structural design, power grids

Method Iteratively update a Gaussian Process [6] with intelligent sampling [2], similar to Bayesian Optimization [4] or Level Set Estimation [7]



Formulation

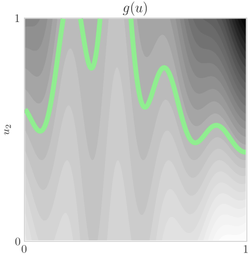
Distribution $U \overset{\mathbb{U}}{\sim} \mathcal{U}[0,1)^d$

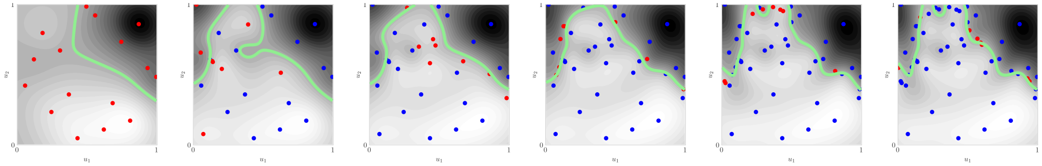
Model GP G with distribution $\mathbb{P}$

Simulation $g : [0,1)^d \to \mathbb{R}$, path of G

Failure Region $F = \{u \in [0,1)^d : g(u) \geq 0\}$

Failure Probability $P = \mathbb{U}(F) = \mathbb{E}_{\mathbb{U}}[1_F(U)]$, random variable in G





GP Regression i.e. Kriging

Simulated Data $\boldsymbol{X} \in [0,1)^{n \times d}, \quad \boldsymbol{Y} = g(\boldsymbol{X})$

Prior Covariance Kernel $k : (0,1)^{n_1 \times d} \times (0,1)^{n_2 \times d} \to \mathbb{R}^{n_1 \times n_2}$

Posterior Mean $m_n(u) = k(u, \boldsymbol{X}) K_{\boldsymbol{X},\boldsymbol{X}}^{-1} \boldsymbol{Y}, \quad K_{\boldsymbol{X},\boldsymbol{X}} = k(\boldsymbol{X}, \boldsymbol{X})$

Posterior Covariance $k_n(u_1, u_2) = k(u_1, u_2) - k(u_1, \boldsymbol{X}) K_{\boldsymbol{X},\boldsymbol{X}}^{-1} k(\boldsymbol{X}, u_2)$

Posterior Variance $\sigma_n^2(u) = k_n(u, u)$

Posterior Distribution at $u \in [0,1)^d$ **with Conditional Probability** $\mathbb{P}_n$

$$G(u) \overset{\mathbb{P}_n}{\sim} \mathcal{N}\left(m_n(u), \sigma_n^2(u)\right)$$

Posterior Failure Probability at $u \in [0,1)^d$

$$p_n(u) = \mathbb{P}_n(G(u) \geq 0) = \Phi\left(\frac{m_n(x)}{\sigma_n(u)}\right)$$

Estimator & Error Bound

Posterior Failure Probability $p_n(u) = \Phi\left(\frac{m_n(u)}{\sigma_n(u)}\right)$, fixed $u \in [0,1)^d$

Preicted Failure Region $\hat{F}_n = \{u \in [0,1)^d : p_n(u) \geq 1/2\}$

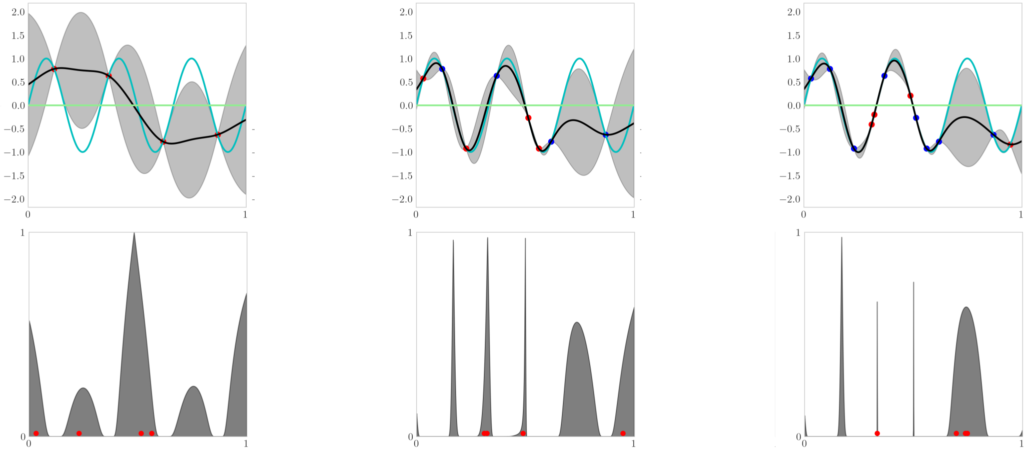
$1-\alpha$ **Confidence Interval (CI)** $P \in \left[\hat{P}_n - \hat{\gamma}_n, \hat{P}_n + \hat{\gamma}_n\right]$ under $\mathbb{P}_n$

GP Based Estimate $\hat{P}_n = \mathbb{E}_{\mathbb{U}}[1_{\hat{F}_n}(U)] = \mathbb{U}\left(\hat{F}_n\right)$

GP Based CI Half-Width $\hat{\gamma}_n = \mathbb{E}_{\mathbb{U}}[\tilde{\varrho}_n(U)]/\alpha$

$$\begin{aligned} \tilde{\varrho}_n(u) &= \min\{p_n(u), 1 - p_n(u)\} \\ &= \underbrace{[1 - p_n(u)]1_{\hat{F}_n}}_{\mathbb{P}_n(\text{False Positive}(u))} + \underbrace{p_n(u)1_{\hat{F}_n^c}}_{\mathbb{P}_n(\text{False Negative}(u))} \end{aligned}$$

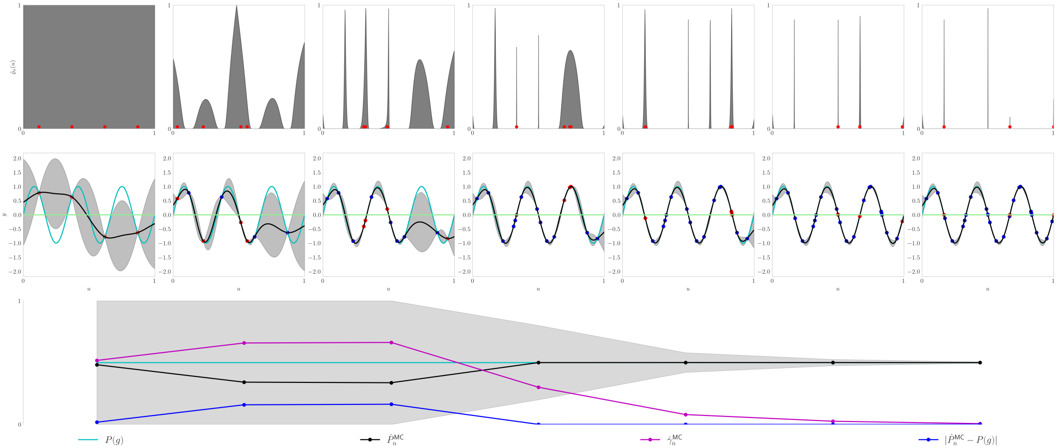
Expected Misclassification Rate $\tilde{\varrho}_n = \mathbb{P}_n(\mathsf{FP}) + \mathbb{P}_n(\mathsf{FN})$



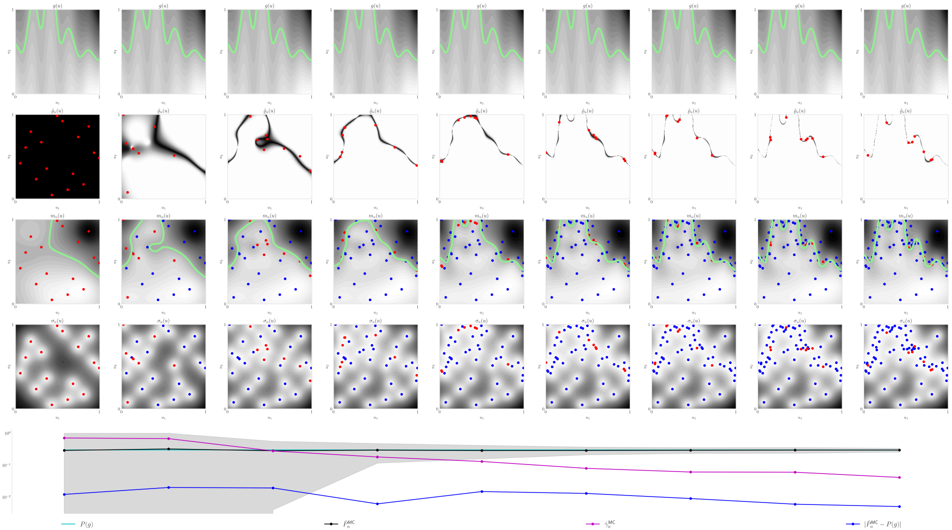
Algorithm

1. Fix $U_0,\ldots,U_{N-1} \sim \mathcal{U}[0,1)^d$, large N e.g. 1 million, possible quasi-random design
2. Initialize $\boldsymbol{X} \leftarrow \emptyset, \boldsymbol{Y} \leftarrow \emptyset$ and set GP priors with fixed hyperparameters.
3. Draw $X_1,\ldots,X_b \overset{\text{IID}}{\sim} \tilde{\varrho}_n$ using rejection sampling from the unnormalized density
$$\tilde{\varrho}_n(u) = \min\{p_n(u), 1-p_n(u)\} = \mathbb{P}_n(\mathsf{FP}(u)) + \mathbb{P}_n(\mathsf{FN}(u)).$$
4. Simulate $\tilde{\boldsymbol{Y}} = g(\tilde{\boldsymbol{X}})$ at $\tilde{\boldsymbol{X}} = \{X_i\}_{i=1}^b$ and update $\boldsymbol{X} \leftarrow \boldsymbol{X} \cup \tilde{\boldsymbol{X}}, \boldsymbol{Y} \leftarrow \boldsymbol{Y} \cup \tilde{\boldsymbol{Y}}$
5. Compute $\{p_n(U_i)\}_{i=0}^{N-1}$ from the GP posterior using efficient block-matrix updates.
6. Approximate estimate $\hat{P}_n$ and CI half-width $\hat{\gamma}_n$ with Monte Carlo
$$\hat{P}_n^{\mathsf{MC}} = \frac{1}{N}\sum_{i=0}^{N-1} 1_{\hat{F}_n}(U_i), \qquad \hat{\gamma}_n^{\mathsf{MC}} = \frac{1}{\alpha N}\sum_{i=0}^{N-1} \tilde{\varrho}_n(U_i)$$
7. If $\hat{\gamma}_n^{\mathsf{MC}}$ too large and sample budget *not* expired, go to step 3.
8. Return $1-\alpha$ approximate confidence interval $[\hat{P}_n^{\mathsf{MC}} - \hat{\gamma}_n^{\mathsf{MC}}, \hat{P}_n^{\mathsf{MC}} + \hat{\gamma}_n^{\mathsf{MC}}]$

Algorithm Example: $g(u) = \sin(6\pi u)$

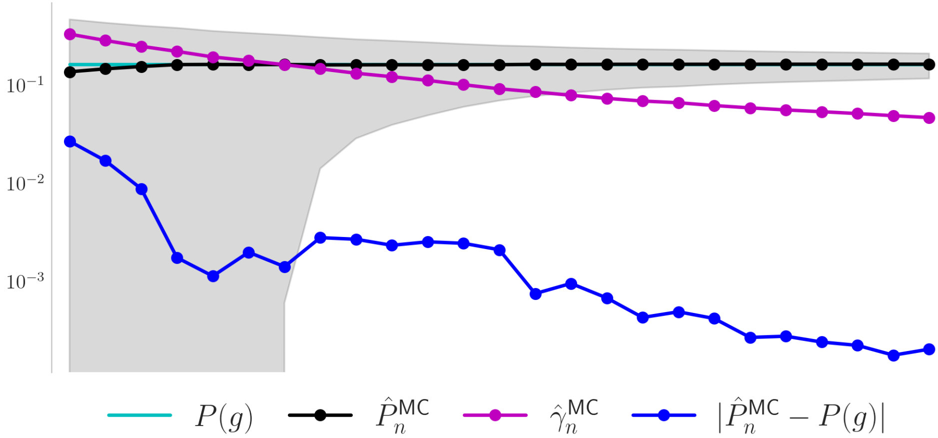


Algorithm Example: Multimodal Function [3]



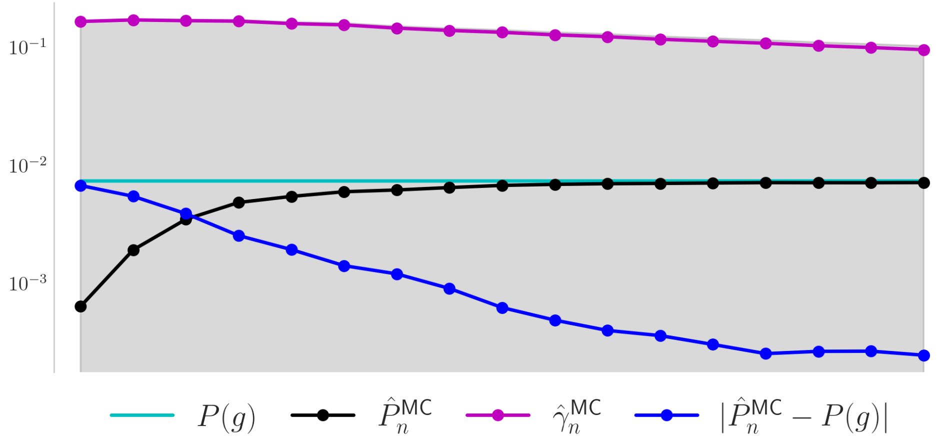
Algorithm Convergence: Ishigami function [5]

$d = 3, \quad P(g) = 0.162, \qquad$ **samples:** initial $= 128$, batch $= 16$, budget $= 512$



Algorithm Convergence: Hartmann function from BoTorch [1]

$d = 6, \quad P(g) = 7.37 \times 10^{-3}, \qquad$ **samples:** initial $= 512$, batch $= 32$, budget $= 1024$



Future Work

Implementation

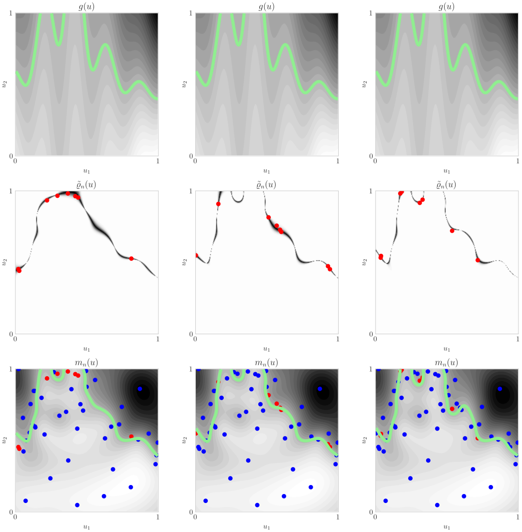
- open source
- accommodate alternative sampling schemes
- larger suite of examples

Analysis

- include uncertainty in $\hat{P}_n^{\text{MC}}, \hat{\gamma}_n^{\text{MC}}$ in CI
- convergence rate of CI width

Extensions

- multi-fidelity problems
- multi-level problems



References I

[1] Maximilian Balandat et al. “BoTorch: a framework for efficient Monte-Carlo Bayesian optimization”. In: *Advances in neural information processing systems* 33 (2020), pp. 21524–21538.

[2] Julien Bect et al. “Sequential design of computer experiments for the estimation of a probability of failure”. In: *Statistics and Computing* 22.3 (2012), pp. 773–793.

[3] Barron J Bichon et al. “Efficient global reliability analysis for nonlinear implicit performance functions”. In: *AIAA journal* 46.10 (2008), pp. 2459–2468.

[4] Peter I Frazier. “Bayesian optimization”. In: *Recent advances in optimization and modeling of contemporary problems*. Informs, 2018, pp. 255–278.

[5] Tsutomu Ishigami and Toshimitsu Homma. “An importance quantification technique in uncertainty analysis for computer models”. In: *[1990] Proceedings. First international symposium on uncertainty modeling and analysis*. IEEE. 1990, pp. 398–403.

References II

[6] Carl Edward Rasmussen. "Gaussian processes in machine learning". In: *Summer school on machine learning*. Springer. 2003, pp. 63–71.

[7] Andrea Zanette, Junzi Zhang, and Mykel J Kochenderfer. "Robust super-level set estimation using gaussian processes". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2018, pp. 276–291.