Preprint submitted to the 14th International Symposium on Process Systems Engineering – PSE 2021+. © J.Wang et al., January 14, 2022. Preprint downloaded from dowlinglab.nd.edu

Scalable Stochastic Programming with Bayesian Hybrid Models

Jialu Wang[1], Elvis A. Eugene[1], Alexander W. Dowling[1*]

[1]*Department of Chemical and Biomolecular Engineering, University of Notre Dame, Notre Dame, IN 46556, USA*
adowling@nd.edu

Abstract

Bayesian hybrid models (BHMs) fuse physics-based insights with machine learning constructs to correct for systematic bias. In this paper, we demonstrate a scalable computational strategy to embed BHMs in an equation-oriented modelling environment. Thus, this paper generalizes stochastic programming, which traditionally focuses on aleatoric uncertainty (as characterized by a probability distribution for uncertainty model parameters) to also consider epistemic uncertainty, i.e., mode-form uncertainty or systematic bias as modelled by the Gaussian process in the BHM. As an illustrative example, we consider ballistic firing using a BHM that includes a simplified glass-box (i.e., equation-oriented) model that neglects air resistance and a Gaussian process model to account for systematic bias (i.e., epistemic or model-form uncertainty) induced from the model simplification. The gravity parameter and the GP hypermeters are inferred from data in a Bayesian framework, yielding a posterior distribution. A novel single-stage stochastic program formulation using the posterior samples and Gaussian quadrature rules is proposed to compute the optimal decisions (e.g., firing angle and velocity) that minimize the expected value of an objective (e.g., distance from a stationary target). PySMO is used to generate expressions for the GP prediction mean and uncertainty in Pyomo, enabling efficient optimization with gradient-based solvers such as Ipopt. A scaling study characterizes the solver time and number of iterations for up to 2,000 samples from the posterior.

Keywords: Hybrid model; Bayesian uncertainty quantification; Optimization; Gaussian process; Pyomo

1. Introduction

Predictive models play a key role in control and decision-making (Adjiman et al., 2021). While the glass-box models are constructed from scientific principles and have a deeper understanding of the underlying processes, they are often complex to form and solve. Many glass-box models contain unknown parameters that are inferred from experimental data. These data are often subject to random phenomena such as variability between experiments or observation noise (Kalyanaraman et al., 2015), which gives rise to aleatory (i.e., parametric) uncertainties. Stochastic programming and robust optimization are routinely used to directly incorporate parametric uncertainty into decision-making frameworks. However, to maintain computational tractability, glass-box models are often simplified or replaced with surrogate models in multiscale engineering frameworks (Biegler et al., 2014). The systematic bias from model inadequacy arising from such simplifications is often referred to as model-form or epistemic uncertainty (McClarren, 2018).

Bayesian hybrid models (BHM) offer a principled framework to quantify, propagate, and mitigate aleatoric and epistemic uncertainties by combining physical glass-box models with black-box surrogate models. In their seminal work, statisticians Kennedy and O'Hagan (2001) proposed a (Bayesian) hybrid modelling framework using Gaussian process models:

$$y = \eta(\boldsymbol{x}|\boldsymbol{\theta}) + \delta(\boldsymbol{x}|\boldsymbol{\phi}, \boldsymbol{D}) + \varepsilon \tag{1}$$

The prediction y consists of three components: the inadequate (simplified or reduced order) glass-box model $\eta(\boldsymbol{x}|\boldsymbol{\theta})$ which depends on the state variables $\boldsymbol{x}$ and model parameters $\boldsymbol{\theta}$; the Gaussian process discrepancy $\delta(\boldsymbol{x}|\boldsymbol{\phi}, \boldsymbol{D})$ which models epistemic uncertainty as a function of the state variables $\boldsymbol{x}$, hyperparameters $\boldsymbol{\phi}$, and data $\boldsymbol{D}$ = [$\boldsymbol{x_{obs}}$, $\boldsymbol{y_{obs}}$]; and, finally, the observation error ε which is modeled as a random variable with known probability distribution. Unlike other hybrid model architectures, such as a neural differential equation, the probabilistic nature of the GP enables the use of Bayesian calibration (Higdon et al., 2004) to infer the model parameters and hyperparameters and provides readily interpretable uncertainty information. The joint posterior distribution of model parameters resulting from Bayesian model calibration informs the uncertainty in the models; specifically, the distribution of model parameters $\boldsymbol{\theta}$ and observation error ε quantifies aleatory uncertainty while the GP output quantifies epistemic uncertainty. We emphasize that prior applications of the Kennedy-O'Hagan framework in chemical engineering (Mebane et al., 2013, Kalyanaraman et al., 2015, Kalyanaraman et al., 2016, Bhat et al., 2017) predominately considers model calibration and uncertainty propagation and not decision-making under uncertainty.

2. Methods

2.1 Stochastic Programming Formulation

In this work, we develop and implement a single-stage stochastic program formulation in Pyomo (Hart et al., 2017) to optimize decisions using BHMs by minimizing the expected values of an arbitrary objective function $u(y)$ in the form of Eqs. (2a):

$$\min_{\boldsymbol{x}} \underset{\boldsymbol{\theta},\boldsymbol{\phi}}{\mathrm{E}}[u(y)] \approx \frac{1}{\sqrt{\pi}} \sum_{s\in S} \sum_{j\in J} w_s w_j u_{s,j} \tag{2a}$$

$$\delta_j = \mu(\boldsymbol{x}|\boldsymbol{\phi}, \boldsymbol{D}) + \sqrt{2}\, z_j\, \sigma(\boldsymbol{x}\,|\boldsymbol{\phi}, \boldsymbol{D}), \qquad \forall j \in J \tag{2b}$$

$$\eta_s = \eta(\boldsymbol{x}|\boldsymbol{\theta}_s), \ \ \forall s \in S \tag{2c}$$

$$y_{s,j} = f(\eta_s, \delta_j), \forall s, j \in S \times J \tag{2d}$$

$$u_{s,j} = u(y_{s,j})\,, \forall s, j \in S \times J \tag{2e}$$

In Eq. (2a), the expectation E of $u(y)$ is approximated using scenario weights $w_s = 1/|S|$. Set S contains samples from the posterior distribution (trace) of $\boldsymbol{\theta}$. Set J contains Gauss-Hermite quadrature nodes z_j and weights w_j, which are used in Eq. (2b) to approximate the GP output distribution characterized by GP prediction mean μ and standard deviation σ. In Eq. (2c), the glass-box model is evaluated at samples θ_s. In Eqs.

(2d) and (2e), the BHM output y and objective function $u(y)$ are evaluated over the set $S \times J$. This formulation is computationally attractive because the highly nonlinear GP prediction mean and standard deviation are evaluated only once, while the glass-box model, and the objective function are evaluated $|S|$, and $|S| \times |J|$ times, respectively.

2.2 Ballistics Firing Example

We apply the stochastic program to the ballistics example from Eugene et al. (2020):

$$\eta_s = \frac{2v_0^2}{g_s} \cdot \sin\psi \cdot \cos\psi, \forall s \in S \tag{3a}$$

$$y_{s,j} = \eta_s + \delta_j, \forall s, j \in S \times J \tag{3b}$$

$$y_{s,j} - \bar{y} = u_{s,j}^{+} - u_{s,j}^{-},\ u_{s,j} = u_{s,j}^{+} + u_{s,j}^{-},\ u_{s,j}^{+}, u_{s,j}^{-} \geq 0, \forall s, j \in S \times J \tag{3c}$$

Eqs. (3a, b) describe the BHM. The state variables are $\boldsymbol{x} = [v_0, \psi]$, where v_0 (m/s) is the firing velocity and ψ (°) is the firing angle. The glass-box model has one uncertain parameter, the acceleration due to gravity g. y (m) is the distance to the impact location of the projectile measured horizontally from y=0 which is the firing location of the projectile. We seek to predict the optimum conditions $\boldsymbol{x}$ to hit a target a fixed distance $\bar{y} = 100$ m away, despite neglecting air resistance (epistemic uncertainty) in the glass-box model. The objective function $u(y) = |y - \bar{y}|$ is reformulated using slack variables in Eq. (3c) to provide differentiable constraints for gradient-based optimization. By combining Eqs. (2) and (3), the expectation of u is minimized to find the optimum $\boldsymbol{x} = [v_0, \psi]$ to hit the target.

The observed data $\boldsymbol{D}$ = $[\boldsymbol{x_{obs}}, \boldsymbol{y_{obs}}]$ is generated from the true physical model which includes the effects of air-resistance on the projectile resulting in its non-parabolic trajectory as described by Eugene et al. (2020). Six data points corresponding to observations from experiments 1 to 5 and 6c in Table 1 of Eugene et al., 2020 were used for the *sequential Bayesian calibration* of the hybrid model. First, the glass-box model is calibrated using the data $\boldsymbol{D}$, a likelihood function, and priors (see Eugene et al. 2020 for details) via Hamiltonian Monte Carlo in PyMC3 (Salvatier et al., 2016) which returns a trace of 2,000 samples from the posterior distribution of the glass-box model parameter g. Next, using the mean value of g from the trace, $\bar{g}$, we compute the residuals $\boldsymbol{y_{obs}} - \eta(\boldsymbol{x_{obs}}, \bar{g})$ which represents the systematic bias in the model due to epistemic uncertainty. These residuals are used to train a discrepancy function for which we assume a Gaussian process with kriging kernel:

$$\boldsymbol{y_{obs}} - \eta(\boldsymbol{x_{obs}}, \bar{g}) = \delta(\boldsymbol{x}|\boldsymbol{\phi}, \boldsymbol{D}) \sim \mathcal{GP}(\mu, k(\cdot,\cdot)) \tag{4a}$$

$$k(x_j, x_k) = \sigma_f^2 \exp\left(-\sum_{i=1}^{n} \beta_i \left|x_{ij} - x_{ik}\right|^2\right),\ j, k \in 1, \dots, m,\ \beta_i \geq 0 \tag{4b}$$

where μ and σ_f^2 are the prediction mean and the variance of the GP model, respectively and β_i is the kriging weight. n=2 denotes the number of input dimensions for $\boldsymbol{x} = [v_0, \psi]$, and m = 6 is the number of observations in the training data set for the GP. The GP model hyperparameters $\boldsymbol{\phi} = [\mu, \sigma_f^2, \beta_1, \beta_2]$ = [-0.68, 2.28, 1.53, 0.13] are trained using maximum likelihood estimation (MLE) (Forrester et al., 2008) implemented in the

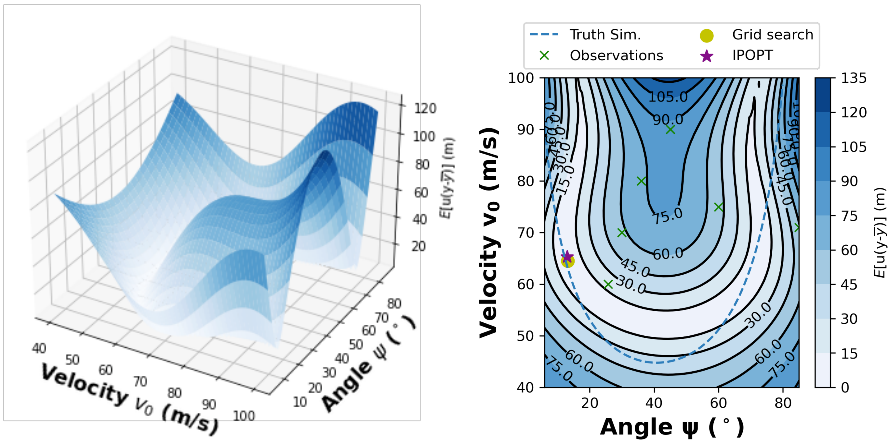

Fig.1. The expectation of the utility function changes with the velocity and the angle visualized in a 3D surface (left) and contour plot (right).

PySMO toolbox of IDAES (Lee et al., 2021). We do not explicitly model aleatory uncertainty due to observation noise in this study.

3. Results

The optimization problem in Eqs. (2, 3) is constructed in Pyomo and solved using IPOPT and the linear solver MA57 (HSL, 2007) using $|S| = 1000$ and $|J| = 10$. To visualize the objective function surface and contours in Fig. 1, square instances of Eqs. (2,3), i.e., $\boldsymbol{x}$ is fixed such that there are no degrees of freedom, are evaluated using a grid of 2750 uniformly space samples over $v_0 \in [40, 100]$ (m/s) and $\psi \in [5.7, 85.5]$ (°). Fig. 1 (left) shows that firing at large angles (near vertical orientation) with large velocities results in undesirably high objective values with expected miss distances over 120 m. In contrast, firing at shallower angles (near horizontal orientation) and modest velocities results in expected miss distances less than 15 m. The dashed blue line in Fig. 1 (right) shows the combinations of angles and velocities that result in a direct hit based on simulations of the true physical model. The optimum calculated by the gradient-based solver Ipopt (purple star), and the grid search optimum (yellow dot), are nearly identical and both close to the direct hit line, indicating Bayesian hybrid model and stochastic programming formulation accurately account for epistemic uncertainty. The optimization problem is reliably solved by Ipopt in approximately 4 s on a MacBook with a 2.6 GHz Intel Core i7 CPU, finding an optimal solution as $v_0 = 65.34$ m/s, $\psi = 12.99$ ° and an objective function value of 5.07 m. The optimum found by the grid search is $v_0 = 64.49$ m/s, $\psi = 13.1$ °, and the optimized expectation value is 5.14 m. As expected, the gradient-based solver outperforms the grid search.

Next, we demonstrate the scalability of the proposed stochastic programming formulation. Fig. 2 shows the variation in CPU time and the number of Ipopt iterations as the number of scenarios $|S|$ increases. For each value of $|S|$, Eqs. (2, 3) are resolved 50 times using samples randomly drawn from the posterior distribution trace. For $|S| = 2000$, Eqs. (2, 3) are solved once using the entire trace. The dots represent the medians for both metrics, while the error bar represents the 75th and 25th percentiles. Fig. 2 (left) shows that as $|S|$ increases, the median CPU time increases from 0.03 s with 10 scenarios, to 4.8 s with

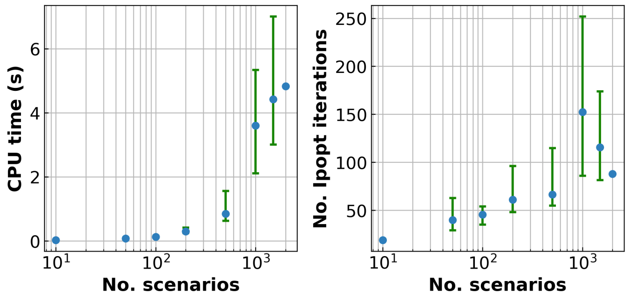

Fig.2. The CPU time and number of IPOPT iterations increase with the number of samples $|S|$. The dots represent the median and the error bars denote the 25%-ile and 75%-ile for 50 random samples exct for $|S|$ =2,000.

2000 scenarios. Likewise, Fig. 2 (right) shows median of the number of Ipopt iterations increases from 19 with 10 scenarios, achieving a peak at 152.5 with 1,000 scenarios, and decreases to 88 with 2,000 scenarios. We hypothesize the large variability in metrics for a given $|S|$ is due to the different scenario data considered for each of the 50 replicates. The problem considering 2,000 scenarios can be solved in less than 4.8 s and 88 iterations with 82,014 variables and 62,012 equality constraints, showing that this stochastic program can be reliably solved by gradient-based solvers with small computational burden. It is also noted that the optimum decisions are practically the same as $|S|$ increases. This highlights the potential to accommodate larger problems with multiple sources of model-form uncertainty, i.e., multiple GP discrepancy functions, by using only a modest number of posterior scenarios, e.g., $|S|$ is 10 or 50.

4. Conclusions

In this paper, we demonstrate a scalable stochastic programming formulation for optimization under both aleatoric (i.e., parametric) and epistemic (i.e., model-form) uncertainties using Bayesian hybrid models (BHMs). By leveraging both Gaussian quadrature rules and PySMO, we demonstrate efficient optimization using the equation-oriented Pyomo modeling environment and gradient-based Ipopt nonlinear programming solver. Through an illustrative ballistics example, we show the Kennedy-O'Hagan inspired BHMs effectively capture epistemic uncertainty; their predictions are consistent with the full-physics true model. Moreover, accounting for epistemic uncertainty in the stochastic programming formulation may be accomplished using as little as 10 samples. As future work, we plan to explore the application of optimization-supported decision-making with hybrid models for diverse applications in process system engineering and adjacent domains. We are especially interested in using BHMs to account for information-loss from model simplified in molecular-to-systems engineering frameworks.

Acknowledgement

This project was funded by NSF CBET Award CBET-1941596. E. A. E. gratefully acknowledges partial support from the Patrick and Jana Eiler's Graduate Fellowship. J.W. gratefully acknowledges partial support from the Carbon Capture Simulation for Industry Impact ($CCSI^2$), funded through the U.S. DOE office of Fossil Energy by the Lawrence Berkeley National Laboratory through contract #DE-AC02-05CH11231.

References

C.S. Adjiman, N.V. Sahinidis, D.G. Vlachos, B. Bakshi, C.T. Maravelias and C. Georgakis, 2021, Process systems engineering perspective on the design of materials and molecules, Ind. Eng. Chem. Res, 60, 14, 5194-5206, 10.1021/acs.iecr.0c05399

K.S. Bhat, D.S. Mebane, P. Mahapatra and C.B. Storlie, 2017, Upscaling uncertainty with dynamic discrepancy for a multi-scale carbon capture system, Journal of the American Statistical Association, 112, 520, 1453-1467,

L.T. Biegler, Y.-d. Lang and W. Lin, 2014, Multi-scale optimization for process systems engineering, Computers & chemical engineering, 60, 17-30, 10.1016/j.compchemeng.2013.07.009

E.A. Eugene, X. Gao and A.W. Dowling, 2020, Learning and optimization with bayesian hybrid models, 2020 American Control Conference (ACC), 3997-4002, 10.23919/ACC45564.2020.9148007

A. Forrester, A. Sobester and A. Keane, 2008, Engineering design via surrogate modelling: A practical guide, John Wiley & Sons,

W.E. Hart, C.D. Laird, J.-P. Watson, D.L. Woodruff, G.A. Hackebeil, B.L. Nicholson and J.D. Siirola, 2017, Pyomo-optimization modeling in python, Springer, 67,

D. Higdon, M. Kennedy, J.C. Cavendish, J.A. Cafeo and R.D. Ryne, 2004, Combining field data and computer simulations for calibration and prediction, SIAM journal on scientific computing, 26, 2, 448-466, 10.1137/s1064827503426693

HSL, 2007, Collection of fortran codes for large-scale scientific computation, http://www. hsl. rl. ac. uk,

J. Kalyanaraman, Y. Fan, Y. Labreche, R.P. Lively, Y. Kawajiri and M.J. Realff, 2015, Bayesian estimation of parametric uncertainties, quantification and reduction using optimal design of experiments for CO_2 adsorption on amine sorbents, Computers & chemical engineering, 81, 376, 10.1016/j.compchemeng.2015.04.028

J. Kalyanaraman, Y. Kawajiri, R.P. Lively and M.J. Realff, 2016, Uncertainty quantification via bayesian inference using sequential monte carlo methods for CO_2 adsorption process, AIChE Journal, 62, 9, 3352-3368,

M.C. Kennedy and A. O'Hagan, 2001, Bayesian calibration of computer models, Journal of the Royal Statistical Society. Series B, Statistical methodology, 63, 3, 425-464, 10.1111/1467-9868.00294

A. Lee, J.H. Ghouse, J.C. Eslick, C.D. Laird, J.D. Siirola, M.A. Zamarripa, D. Gunter, J.H. Shinn, A.W. Dowling, D. Bhattacharyya, L.T. Biegler, A.P. Burgard and D.C. Miller, 2021, The idaes process modeling framework and model library—flexibility for process simulation and optimization, Journal of advanced manufacturing and processing, 3, 3, 10.1002/amp2.10095

R.G. McClarren, 2018, Uncertainty quantification and predictive computational science : A foundation for physical scientists and engineers, Cham : Springer International Publishing : Imprint: Springer,

D.S. Mebane, K.S. Bhat, J.D. Kress, D.J. Fauth, M.L. Gray, A. Lee and D.C. Miller, 2013, Bayesian calibration of thermodynamic models for the uptake of CO_2 in supported amine sorbents using ab initio priors, Physical Chemistry Chemical Physics, 15, 12, 4355-4366,

J. Salvatier, T.V. Wiecki and C. Fonnesbeck, 2016, Probabilistic programming in python using PyMC3, PeerJ. Computer science, 2, e55, 10.7717/peerj-cs.55

A. Wächter and L.T. Biegler, 2006, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, Mathematical programming, 106, 1, 25-57