One-Shot Optimal Design for Gaussian Process Analysis of Randomized Experiments

Anonymous Author(s)
Affiliation
Address
email

Abstract

Bayesian Optimization provides a sample-efficient approach to optimize Internet systems that are evaluated with randomized experiments. Such evaluations are often resource- and time- consuming in order to measure noisy and long-term outcomes. Thus, the initial randomized design, i.e. determining number of test groups and sample sizes, plays a critical role in building an accurate Gaussian Process model to optimize efficiently and decreasing experimentation cost. We develop a simulation-based method with meta-learned priors to decide the optimal design for the initial batch of GP-modeled randomized experiments. The meta-learning is performed on a large corpus of randomized experiments conducted at Meta and obtains sensible GP priors for simulating across different designs. The one-shot optimal design policy is derived by training a machine learning model with simulation data to map experiment characteristics to an optimal design. Our evaluations show that our proposed optimal design significantly improves resource-efficiency while achieving a target GP model accuracy.

1 Introduction

Randomized experiments are the gold-standard method for Internet firms to evaluate the performance of changes to their service [Kohavi and Longbotham, 2017]. Running these experiments is often time- and resource- consuming since it requires using large sample sizes to detect the effects of high-variance outcomes and running over long periods of time to observe long-term effects. Especially when experimenting with high-dimensional action spaces, such as continuous parameters in a ranking policy [Letham and Bakshy, 2019], the prototypical practice that runs a parallel set of treatments chosen adhocly and selects the best one based on observed outcomes is not sustainable.

Bayesian optimization (BayesOpt) provides a sample-efficient approach for optimizing black-box functions with expensive evaluations and is widely used to tune online systems that are evaluated with A/B tests [Letham et al., 2019, Feng et al., 2020, Liu et al., 2022]. Considering the long evaluation time via randomized experiments, batch BayesOpt is used in practice to take full advantage of the experimental resources and test multiple parameter values (referred to as *arms*) simultaneously. It starts with an initial batch with arms generated by space-filling designs, e.g., Latin hypercube, builds a Gaussian process (GP) surrogate based on previous outcomes, and uses an acquisition function defined from this surrogate to select the arms to evaluate next [Jones et al., 1998, Snoek et al., 2012].

Although BayesOpt is sequential in nature, the experiment design of the initial random batch is critical for building an accurate GP model to generate good candidates sequentially and decreasing experimentation cost. Having enough unique arms is known to be important to model the underlying

Submitted to the 6th Workshop on Meta-Learning at NeurIPS 2022, New Orleans. Do not distribute.

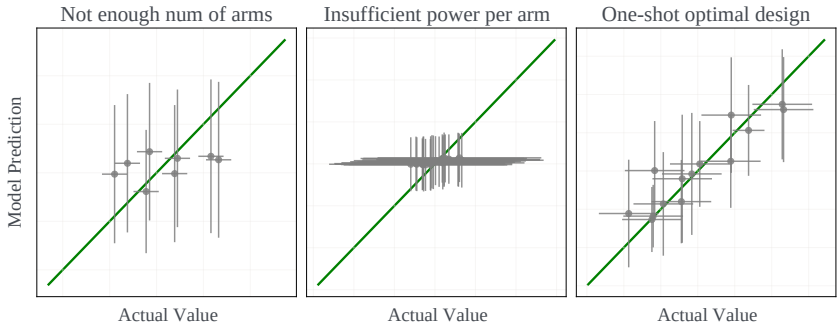

Figure 1: Leave-one-out cross validation predictions (mean, and 95% posterior predictive interval) for the GP fits to the results of a 3-d real-world experiment under three initial designs. Not having enough number of arms (left) or not having enough sample size per arm (middle) leads to a poor fit, while our proposed one-shot optimal design (right) makes accurate out-of-sample predictions.

response surfaces, e.g., empirical rule $n = 10d$ such that the number of arms is 10 times the input dimension, which has been proposed and studied by Loeppky et al. [2009]. In addition, being able to isolate signal from noise is known to be important to the GP model accuracy and BayesOpt performance [Binois et al., 2019]. Thus, each arm requires a certain amount of samples (adequate statistical power) to detect effects for noisy outcomes. Figure 1 demonstrates the importance of initial design to achieve a good GP model fit and optimization performances. This brings an important design question in the face of noisy evaluations: how to decide the number of unique arms and sample size per arm in order to balance exploring response surfaces with more arms while keeping an adequate signal-to-noise ratio for a fixed experiment budget.

There are two main challenges in tackling this design question: 1) In practice, the prior knowledge of underlying response surfaces, e.g., the variance of outcome and the distribution of GP hyper-parameters, is typically unknown. Without such a priori information, one can hardly make the design, as it largely depends on the effective dimensionality of the parameter space and the smoothness of the response surfaces. Sequential design strategies that iterate between design and learning has been proposed and studied as a remedy [Chen and Zhou, 2017]. However, this does not solve the one-shot design that is desired in the batch BayesOpt, especially when evaluations take a long time. 2) Solving a full batched version of selecting locations of unique arms and numbers of samples allocated on each arm, is not computationally tractable as discussed in Binois et al. [2019].

In this paper, we propose a simulation-based method with meta-learned priors to decide on the optimal design for the initial batch of GP-modeled randomized experiments. The contributions of this work are: 1) We frame this one-shot design problem as a constrained optimization problem of obtaining the best GP model prediction accuracy given a fixed sample budget, in which we use a random uniform design and decide the number of arms and sample size per arm. 2) We conduct a meta-analysis of a large corpus of historical experiments at Meta to generate empirical GP priors. The meta-learned GP priors capture and transfer the information to facilitate the one-shot design for experiments with different characteristics e.g. parameter dimensions and outcomes. 3) We devised a simulation-based approach to take the complexity of input-output relationship, i.e., the true underlying response surface and the noise surface into account through sampling true GP from meta-learned priors. 4) We proposed an easy-to-deploy one-shot design policy by building a machine learning surrogate with the datasets from simulations.

2 Methodology

2.1 One-Shot Optimal Design for Gaussian Processes

Let $\mathbf{Y} = [y_1, \cdots, y_n]^T$ denote the outcomes of n arms from randomized experiments with total sample size N. We assume the model $y_i = f(\mathbf{x}_i) + \epsilon_i$, $i = 1, \ldots, n$, where the function $f : \mathbb{R}^d \to \mathbb{R}$

is the true response surface and $\epsilon_i \overset{iid}{\sim} \mathcal{N}(0, \sigma_e^2)$ is the observation noise in which σ_e^2 depends on the sample size per arm $\sigma_e^2 \propto \frac{n}{N}$. For simplicity and comparability, the outcomes are standardized to zero mean and unit variance; the observation noise is scaled accordingly. The input variables $\mathbf{X} = [\mathbf{x}_1^T, \cdots, \mathbf{x}_n^T]$ are scaled to $[0, 1]^d$, and each $\mathbf{x}_i$ is a d-dimensional vector.

We assume $f(\mathbf{x})$ to be zero-mean, constant variance, i.e., outputscales σ_f^2, with a covariance function K_θ from Matérn family with parameter $\gamma = 2.5$, as suggested in Rasmussen [2003]. The kernel matrix K of GP with fixed observation noise is parameterized by lengthscales $\theta = (\theta_1, \cdots, \theta_d)$, outputscale σ_f^2 and noise σ_e^2, represented as $K(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 K_\theta(\mathbf{x}_i, \mathbf{x}_j) + \delta_{ij}\sigma_e^2$ where δ_{ij} is a Kronecker delta that equals to one when $i = j$ and zero otherwise. The kernel parameters θ and σ_f^2 are estimated through optimizing the marginal log likelihood and σ_e^2 is a given input.

In one-shot optimal design, the goal is to achieve good GP prediction accuracy which is evaluated on out-of-sample mean squared error (MSE). The locations of arms are decided by a random uniform design. The decision parameters are thus the number of arms n and the sample size N. Denote a fixed sample size budget as B. Thus the problem can be formulated as constrained optimization problems based on use case: minimizing out-of-sample MSE given a fixed budget, i.e., $\min_{n,N} \text{MSE}_f(n, N)$ s.t. $N \leq B$; or minimizing sample size N to achieve a target out-of-sample MSE (τ), i.e., $\min_{n,N} N$ s.t. $\text{MSE}_f(n, N) \leq \tau$.

2.2 Empirical Priors from Meta-learning

One main challenge of one-shot optimal design is that prior distributions of variance σ_f^2, noise σ_e^2, and distribution of kernel hyper-parameters θ are unknown. To optimize MSE that depends on these unknown priors, we perform a meta-analysis based on a large corpus of historical randomized experiments at Meta to obtain empirical priors. For each experiment, a GP with fixed noise model is fitted on the outcomes from the initial batch. A leave-one-out cross validation is performed to filter out estimates with poor fitted GP. This meta-analysis generates empirical prior distributions of kernel hyper-parameters and outcome noise for experiments with different parameter dimensions, conducted in different product areas etc. Figure 2 visualizes the distribution of estimated lengthscales segmented based on parameter dimension d. It can be seen that the effective dimension does not linearly increase with d: higher dimensional spaces tend to be sparse, i.e., with less important parameters. These functions priors from meta-learning thus play an important role in guiding efficient experiment designs.

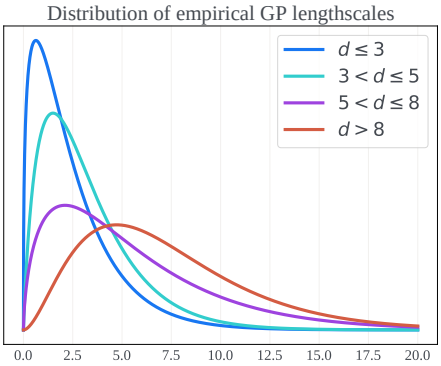

Figure 2: Empirical lengthscales distribution from meta-learning. Each density curve corresponds to the lengthscale distribution segmented based on dimension d.

2.3 Simulation-Based Optimal Design Policy

Directly solving the one-shot optimal design is not computationally tractable. Therefore, we consider a simulation-based algorithm. For each design of n arms and N sample size, simulation is used to obtain an out-of-sample MSE for a given GP prior obtained from meta-analysis. Intuitively, this allows us to measure the GP model fit of realistic response surfaces learned from historical experiments using different experiment designs. Based on these simulation data, we can derive an optimal design policy by training a neural network as surrogate to suggest (n, N) based on experiments' characteristics, e.g., parameter dimensions, product area, and outcomes of interest.

Simulation. We simulate realistic "true" response surfaces by sampling GPs from the distributions obtained from the meta-learning. In each simulation run, given a true GP f, we can evaluate a grid of designs (n, N). For each design, n training inputs $\mathbf{X}$ are generated by random uniform design and training outputs $f(\mathbf{X})$ are perturbed with noise randomly sampled from Gaussian distrbution with

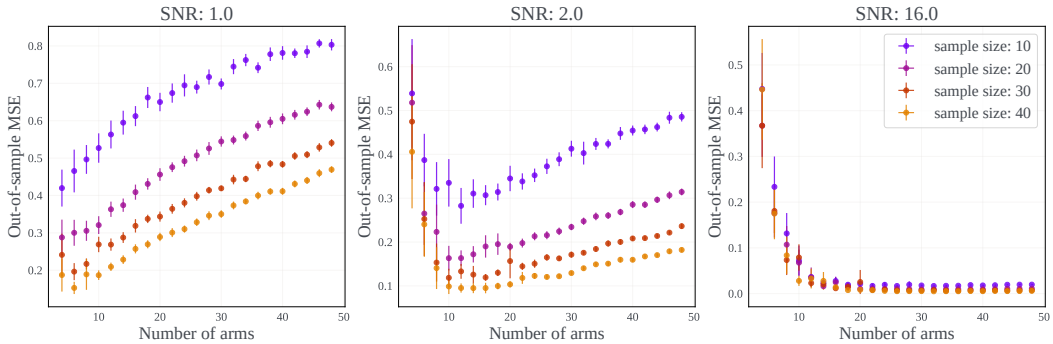

Figure 3: Out-of-sample MSE obtained from simulation for different designs i.e number of arms (n) and sample sizes (N). The simulations are based on three 2-dimensional real-world randomized experiments. The signal-to-noise ratios of these three experiment outcome metrics vary from low (left) to high (right). Different sample sizes are indicated by different colors.

variance being $\frac{n}{N}s^2$, in which s^2 is the observed metric variance. We fit a GP on the training data and compute the out-of-sample MSE, measuring the prediction accuracy of fitted GP. We run this simulation multiple times for each design and compute the average of out-of-sample MSE. From the simulation, we measure the performance (out-of-sample MSE) of different designs for a variety of experiments with different characteristics.

Optimal design policy. Given the simulation data, we build a machine learning model to predict the model accuracy (MSE) using design (n, N) and experiment characteristics, e.g., parameter dimension d, signal-to-noise ratio (SNR) of the outcome metric, and product area as features. For future experiments, the model can predict the performance of different designs and decide the optimal one based on their characteristics. Please see Algorithm 1 in the Appendix for more details.

3 Results and Discussions

Illustrative real-world experiments. Here we illustrate the simulation-based one-shot design with three real-world experiments with 2-dimensional parameter spaces. Figure 3 shows the out-of-sample MSE of different design choices for these experiments. The left plot shows the results of outcomes with lowest SNR, from which we see that increasing the number of arms is always harmful for such extremely noisy outcomes. Increasing statistical power per arm is preferable in order to isolate signal from noise. On the other hand, for outcomes with high SNR (right plot), having more arms is more useful to achieve better GP model fit since each arm is adequately powered and the marginal gain from increasing the sample size decreases.

For outcomes with average SNR, the middle plot shows that the optimal design choice balances exploring new arms versus keeping enough power per arm. When there are a few arms, increasing the number of arms n improves out-of-sample MSE; and there is a sharp transition that once n passes a point, having more arms hurts the GP model fit.

Evaluation with a large corpus of randomized experiments. We conduct an evaluation of proposed one-shot optimal design on a large corpus of real-world randomized experiments. The evaluation is done in a train-test split fashion so that the optimal design policy is trained on training data and evaluated on a separate set of experiments. We compare our method against two different design strategies: 1) historical design used by experimenters; 2) the empirical rule $n = 10d$ recommended by Chapman et al. [1994] and N being historical (observed) sample size. The design is decided by finding a lowest N that achieves a target GP model accuracy and the performance is evaluated based on recommended sample size. The evaluation shows that our method is close to 3X more resource-efficient. See Table 2 in the Appendix for details.

References

Mickaël Binois, Jiangeng Huang, Robert B Gramacy, and Mike Ludkovski. Replication or exploration? sequential design for stochastic simulation experiments. *Technometrics*, 61(1):7–23, 2019.

William L Chapman, William J Welch, Kenneth P Bowman, Jerome Sacks, and John E Walsh. Arctic sea ice variability: Model sensitivities and a multidecadal simulation. *Journal of Geophysical Research: Oceans*, 99(C1):919–935, 1994.

Xi Chen and Qiang Zhou. Sequential design strategies for mean response surface metamodeling via stochastic kriging with adaptive exploration and exploitation. *European Journal of Operational Research*, 262(2):575–585, 2017.

Qing Feng, Benjamin Letham, Hongzi Mao, and Eytan Bakshy. High-dimensional contextual policy search with unknown context rewards using Bayesian optimization. In *Advances in Neural Information Processing Systems 33*, NeurIPS, 2020.

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

Ron Kohavi and Roger Longbotham. Online controlled experiments and a/b testing. *Encyclopedia of machine learning and data mining*, 7(8):922–929, 2017.

Benjamin Letham and Eytan Bakshy. Bayesian optimization for policy search via Online-Offline experimentation. April 2019.

Benjamin Letham, Brian Karrer, Guilherme Ottoni, Eytan Bakshy, et al. Constrained Bayesian optimization with noisy experiments. *Bayesian Analysis*, 14(2):495–519, 2019.

Sulin Liu, Qing Feng, David Eriksson, Benjamin Letham, and Eytan Bakshy. Sparse bayesian optimization. *arXiv preprint arXiv:2203.01900*, 2022.

Jason L Loeppky, Jerome Sacks, and William J Welch. Choosing the sample size of a computer experiment: A practical guide. *Technometrics*, 51(4):366–376, 2009.

Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, 25, 2012.

A Appendix

A.1 One-shot Optimal Design Algorithm Details

Starting with a list of real historical experiments, Algorithm 1 describes the steps for getting the surrogate experiment data. Function `Sample-GP` draws new training and test datasets from a given GP with the noise level computed based on the given snr adjusted to correspond to the given sample size per arm. By drawing multiple datasets, function `Single-Design-MSE` computes the average out-of-sample MSE. Finally, `Surrogate-Data` fits a GP to each historical real experiment and uses it as the ground truth for simulating more experiment datasets from a grid of designs.

Algorithm 1 Simulation Algorithm

function SAMPLE-GP(f, s_f, n, N)
  Sample input $\mathbf{X}_{n\times d}$ based on Sobol design and out-of-sample points $\mathbf{X^\star}_{m\times d} \overset{i.i.d.}{\sim} \mathbf{U}(0,1)$
  Sample ground truth GP $f([\mathbf{X}, \mathbf{X}^\star])$
  Set the noise level: $\sigma_\epsilon = s_f * N/n$
  Perturb input points with noise $\epsilon, \epsilon^\star \overset{i.i.d.}{\sim} N(0, \sigma_\epsilon^2)$: $\mathbf{Y} = f(\mathbf{X}) + \epsilon$ and $\mathbf{Y}^\star = f(\mathbf{X}^\star) + \epsilon^\star$
  return $\mathbf{X}, \mathbf{Y}, \sigma_\epsilon, \mathbf{X}^\star, \mathbf{Y}^\star$
function SINGLE-DESIGN-MSE(f, s_f, n, N)
  Initialize average out-of-sample MSE: avg(MSE) $\leftarrow 0$
  for i **in** $[T]$ **do**
    $\mathbf{X}, \mathbf{Y}, \sigma_\epsilon, \mathbf{X}^\star, \mathbf{Y}^\star \leftarrow$ **Sample-GP**(f, s_f, n, N)
    Fit a fixed noise GP $\hat{f}$ using training inputs $\mathbf{X}, \mathbf{Y}, \sigma_\epsilon$
    $\hat{f}(\mathbf{X}^\star) \leftarrow$ Predict on testing inputs $\mathbf{X}^\star$
    Compute out-of-sample MSE: MSE$_i = \|\hat{f}(\mathbf{X}^\star) - \mathbf{Y}^\star\|_2^2/m$
    Update avg(MSE) $\leftarrow avg$(MSE) + MSE$_i/T$
  return avg(MSE)
procedure SURROGATE-DATA(List of historical experiment data $\{\mathbf{X}_i, \mathbf{Y}_i, \sigma_i\}, i = 1, \ldots, E$)
  for i **in** $[E]$ **do**
    Initially empty list of surrogate data results: $SD = \emptyset$
    Fit a fixed noise GP $\hat{f}$ to $(\mathbf{X}_i, \mathbf{Y}_i, \sigma_i)$
    Estimate signal-to-noise ratio $\hat{s}_f$
    for n **in** [min # of arms, max # of arms] **do**
      for N **in** [min sample size, max sample size] **do**
        avg(MSE) $\leftarrow$ **Single-Design-MSE**($\hat{f}, \hat{s}_f, n, N$)
        Append $(\hat{s}_f, d, n, N, avg$(MSE)$)$ to SD
  return SD

A.2 Results Evaluation Details

Figure 4 shows the output of the logistic regression used to model probability of a good model fit for $(s_f = 5, d = 8)$. As a final step of the optimal sizing algorithm, we provide a *sizing recommendation* corresponding to the smallest sample size for which the predicted probability is above a given threshold (taken to be 0.8 in our algorithm). Table 1 in the Appendix shows the sizing recommendations for other experiment dimensions.

Table 1 shows the recommended number of arms and the total sample size as a percentage of population given by the proposed optimal design algorithm for a given d and with $s_f = 5$. Since this is the median signal-to-noise ratio in our dataset, for the metrics for which there is no prior SNR estimate one can use the median SNR as input into the optimal sizing algorithm proposed in this paper.

Table 2 shows the average number of arms and sample size across several designs from different algorithms. We see that the proposed algorithm needs 2.75 times less sample size than the other

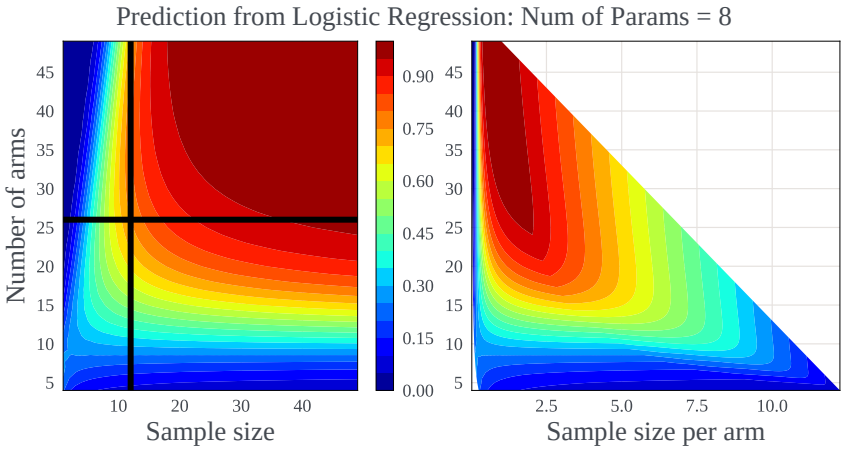

Figure 4: Predicted probabilities of good model fit based on logistic regression used for surrogate model for varying number of arms and sample size and fixed number of parameters (eight) and fixed signal-to-noise metric ratio (two). The thick straight lines indicate the winning design.

d	n	N
1	9	5
2	12	6
3	15	7
4	17	8
5	19	9
6	22	10
7	24	11
8	26	12
9	29	13
10	31	14
11	33	15
12	35	16

Table 1: Sizing recommendations for the number of arms (n) and the sample size (N) as a percentage of population given the number of parameters (d) and metric signal-to-noise ratio ($s_f = 5$).

algorithms while the average MSE for each algorithm is kept below the set threshold $\tau = 0.2$, thus the new algorithm is the most resource-efficient.

Design	n	N
Recommended by the optimal sizing alg.	13	4
Observed design	22	11
Observed sample size, num. of arms $= 10 * d$	30	11

Table 2: The median of the number of arms and the sample size of different experiment designs.