Practical Bayesian Optimization for Transportation Simulators

Laura Schultz and Vadim Sokolov
George Mason University

First Draft: December 2017
This Draft: October 2018

Abstract

Simulators play a major role in analyzing multi-modal transportation networks. As complexity of simulators increases, development of calibration procedures is becoming an increasingly challenging task. Current calibration procedures often rely on heuristics, rules of thumb and sometimes on brute-force search. In this paper we consider an automated framework for calibration that relies on Bayesian optimization. Bayesian optimization treats the simulator as a sample from a Gaussian process (GP). Tractability and sample efficiency of Gaussian processes enable computationally efficient algorithms for calibration problems. We show how the choice of prior and inference algorithm affect the outcome of our optimization procedure. We develop dimensionality reduction techniques that allow for our optimization techniques to be applicable for real-life problems. We develop a distributed, Gaussian Process Bayesian regression and active learning models. We demonstrate those to calibrate ground transportation simulation models. Finally, we discuss directions for further research.

1 Introduction

Modern urban transportation systems required for urban planning and decision making demands the development and maintenance of complex, stochastic computer simulations to adequately forecast the impacts of regional development and gain insights into traveler behaviors. Complexity of transportation simulators grows with our capability to collect more data and the increase of computing power available to researchers and practitioners. In essence a simulation model is a highly nonlinear function that maps inputs (e.g. demand variables) to outputs (e.g. congestion patterns).

*Schultz is a PhD student at George Mason University, email: lschult2@gmu.edu. Sokolov is an assistant professor at George Mason University, email: vsokolov@gmu.edu
Adjustments, or calibrations, to the model’s inputs are required to align the associated outputs more with the field observations. Models need to be re-calibrated every time a new module or assumption is introduced. The primary goal of a calibration procedure is to find a set of input parameters that lead to simulation outputs that match observed data, such as traffic counts, point-to-point travel times, or transit ridership, as close as possible. A simulator that is capable of reproducing observed flows can then be used confidently to develop forecasts on network performance for different infrastructure changes or policies.

Traditional static assignment techniques [25, 26], such as the four-step model, allowed for a mathematically rigorous framework and set of efficient algorithms for model calibration. However, more behaviorally realistic models that integrate dynamic traffic assignment, transit simulations and disaggregate models for travel behaviors, do not allow for generalized universal solutions [30]. Modern simulators typically integrate multiple modules, such as dynamic traffic assignment, transit simulator and activity-based models (ABM) for travel behavior, which are developed in “isolation”. For example, discrete choice models are estimated using travel survey data before being implemented into the integrated ABM model.

The calibration processes for integrated modes, that include individual components, are ad-hoc. Our focus in this paper is on developing a mathematical framework that relies on methods of Bayesian optimization to develop surrogate models that can be used for calibration problems.

Numerous approaches for the calibration of simulation-based traffic flow models have been produced by treating the problem as an optimization issue. The lack of gradients in these models has led to mainly meta-heuristic methods being used, such as Simulation Optimization (SO) [5, 39], Genetic Algorithm (GA) [4, 24], Simultaneous Perturbation Stochastic Approximation (SPSA) [23, 6, 22], and exhaustive evaluation [16], with relative success [38].

Alternatively, Bayesian inference methods provide a confidence value and analytic capability not necessarily produced by other general-purpose approaches for nonparametric linear and non-linear modeling; although limited, their application in transportation has been successful [17, 11, 12, 40]. One of the first statistical methodologies to address the analysis of computer experiments in general can be found in [34], which introduces a kind of stochastic process known as a Gaussian Process(GP) for use with Bayesian inference [13, 36, 8, 31, 32]. The application of GP regression towards calibration was pioneered by [19], where the concept was deployed as surrogate model, or emulator, which estimated the sources of uncertainty between the simulation and true process to improve the prediction accuracy for unverified variable settings. [18] further expands this framework to address high-dimensional problems with high-dimensional outputs by combining it with Markov chain Monte Carlo(MCMC) methods; others have begun integrating these with Machine Learning techniques [35, 31].

However, the primary focus for several of these applications centered on making the most accurate predictions rather than on aligning the simulators themselves. Successful calibration will require a balance the exploration of unknown portions of the input sample space with the exploitation of all known information. [3] and [33] discuss extensively several Bayesian utility functions and their non-Bayesian Design of Experiments (DOE) equivalents.
In addition, addressing the exponential increase in the dimensions of transportation simulators is becoming more paramount as models become more detailed and cities grow larger. The disaggregation of an origin-destination matrix into individual trip-makers through activity-based decision modeling (ABDM) is one approach to decreasing the complexity of traffic flow dynamics in complex network infrastructures [28]. Pre-processing methods such as Principal Component Analysis (PCA) [9] have been used in transportation to further dimensionality with minimal cost to accuracy. We build on these existing works in simulation-based optimization literature both in transportation and other engineering fields by formalizing a Bayesian optimization framework to calibrate complex transportation simulators. Our contributions are

- Formulation of Gaussian Process Bayesian methods for transportation optimization
- Experiment Design algorithms for active conservation of restricted resource allocations
- Dimensionality reduction techniques for complexity control

The remainder of this paper is organized as follows: Section 2 defines a notation set that will be used throughout the document and details the framework and approaches to calibration; Section 3 addresses the need for dimensionality reduction; Section 4 provides an illustrative example; and Section 5 offers avenues for further research.

2 Calibration Framework

The relationship between a true process’ output and the output of the transportation simulator emulating the behavior is limited by the human ability to quantify an observed process. This results in three primary sources of variation, which must be resolved for in order to provide realistic and useful calibration values [19]:

- Residual Variability – Processes do not always take the same value each time the same identified inputs are used. This can be due to the process being inherently unpredictable or stochastic or due to a set of unrecognized conditions that effect the process
- Observation Error – the inherent limitation in observation recording which can result in measurement errors or unintended correlations
- Model Inadequacy – the difference between the true mean value of the process a model simulates and the model’s output given the limitations resulting from the assumptions its creators have used

In other words, the relationship between a field observation of a process, \( y \), and the true values of the process, \( \xi \), is a function of the inputs, \( x_i \), plus some error:

\[ y = f(x_i) + \varepsilon \]

Kennedy further defines the following sources of variability; however, the goal of this calibration framework needs only to explicitly account at the following level.

---

1. Kennedy further defines the following sources of variability; however, the goal of this calibration framework needs only to explicitly account at the following level.
Here, $e_i$ can be attributed to observation error and the residual variation inherent to the observed process; in practice, the observation error and residual variation cannot be separated out. The true process can be split into two functions: the simulator, $\phi$, and the model inadequacy, $\epsilon$.

$$\zeta(x_i) = \phi(x_i, \psi_i) + \epsilon(x_i)$$

The simulator $\phi$ is a function of the data inputs, $x$, and values of the tunable simulation variables, $\psi$. The goal is to reconstruct optimal values of $\psi$ via the calibration process. Accounting for the measurement noise and model inadequacy results in a final relation between the observed and simulated data:

$$y(x_i) = \phi(x_i, \psi_i) + \epsilon(x_i) + e_i$$

Our framework is designed to execute Bayesian optimization algorithms for calibration of transportation simulators in distributed computing environments. We break our framework into three reiterative stages: Evaluation, Integration, and Exploration. The following sections outline each in detail.

### 2.1 Evaluation

During the evaluation stage, new data is obtained based upon the recommended sampling strategy produced by the previous cycle’s exploration state; this stage is initially skipped when the calibration process begins.

Of primary concern is the increasing complexity of transportation computation modeling, which results in a single input set’s evaluation time ranging from many hours to days. This computational constraint can potentially limit the scale and scope of calibration investigations and result in large areas of sample space unexplored and sub-optimal decisions.

Fortunately, as High-Performance Computing (HPC) resources have become increasingly available in most research environments, new modes of computational processing and experimentation have become possible—parallel tasking capabilities allow multiple simulated runs, referred to as a batch, to be performed simultaneously and HPC programs aid in coordinating worker units to run codes across multiple processors to maximize the available resources and time management. By leveraging these advances and
running a queue of pending input sets concurrently through the simulator, a larger set of unknown inputs can be evaluated in an acceptable time frame.

The HPC master controller program, known as an *emcee*, coordinates assignments to free processors and collects the finished simulation outputs. For this calibration setup, the *emcee* program must also be capable of interaction with a model exploration program that provides the queue of untested input sets to be evaluated at this stage.

Notably, it may appear counter-intuitive to place this stage at the front of the cycle structure, especially given that this stage is initially skipped when the calibration process begins. However, the time or cost constraints related with the simulator in question encourages evaluation to never occur unless improvements are being sought and integration occurs subsequently. Therefore, it’s placement is crucial so as not to misemploy limited resources in pursuit of the best calibration.

### 2.2 Integration

In order to align the output of a simulator with the data collected in the field, tunable variables, \(\psi\), which influence the output of the simulation model, must be adjusted. To calibrate the model, we use observed field data \(\{Y, X\}\) and seek \(\psi^*\) which minimizes the inadequacy error \(\varepsilon\):

\[
\psi^* \in \arg\min_{\psi \in \mathcal{A}} [L(\varepsilon(X)) = L(\phi(\psi, X))]
\]

where the constraint set \(\mathcal{A}\) encodes our prior knowledge about the feasible ranges of the model’s parameters—for example, a traveler’s value of time must be positive. Here, \(L\) is the divergence measure that quantifies the inadequacy between the observed data and the simulator’s output. For example, we can choose \(L\) to be the mean squared error:

\[
L [Y, \phi(\Psi, X)] = \frac{1}{N} \sum_{i=1}^{N} (y(x_i) - \phi(x_i, \psi_i))^2
\]

Recall that each simulation run is expected to require significant computing resources; thus, the goal is to design a sample efficient optimization algorithm that solves the problem with smallest number of simulation evaluations.

Many algorithms exist to solve such an optimization problem; however, the complex interactions represented within the transportation simulator are assumed to beyond the scope of our knowledge and, as a result, our objective function, \(L\), cannot be further decomposed and solved analytically. By treating the objective function instead as an unknown, stochastic function, black-box optimization methodologies, which assume to only know the inputs and outputs of a process, can be leveraged.

To solve our black-box optimization problem, we use a surrogate to adequately represent the relationship between our transportation model and the field observations, denoted as \(f(\cdot)\). The Bayesian approach attempts to further quantify the uncertainty in this assumed surrogate model’s accuracy by constructing a probability distribution over all

\[\text{Section 2.2.2}\]
potential linear and nonlinear functions. Data is used to calculate the posterior, or conditional, probability of all potential input sets not yet tried given the evidential results collected in the Evaluation stages, $P(\text{potential} \mid \text{evaluated})$.

The Integration stage performs this derivation through the use of Bayes’ theorem, which states that the posterior probability of a model, given a set of evidential data, is proportional to the likelihood of the evidence given the model multiplied by the prior probability of the model:

$$P(\text{Model} \mid \text{Evidence}) \propto P(\text{Evidence} \mid \text{Model})P(\text{Model})$$

Once this distribution is sufficiently mapped, a valued estimation for our parameters can be made with minimal uncertainty. For a more in-depth review of Bayesian optimization and previous work in black-box methodologies, refer to [2].

### 2.2.1 GP Prior Distribution

The prior probability of a model in Bayes’ theorem is the unconditional probability that is assigned before any relevant evidence is taken into account. In other words, this probability distribution captures the initial beliefs regarding our surrogate function prior to any evidence being collected. The most common surrogate model used to approximate the functionality of a simulation is a Gaussian Process (GP) [15].

Recall that in statistics, a probability distribution is essentially an equation, or function, which links the possible outcomes of a random phenomenon with its probability of occurrence. An important and often-encountered member of these distributions is referred to as a Gaussian Distribution, which is most known for its characteristic, symmetric “bell shape”. These distributions are very useful for any analytical manipulations required in statistics because many of the integrals involving Gaussians tend to have simple, closed forms. Set over a finite, k-dimensional space, Gaussians are described by a $k \times 1$ scalar mean vector, $\mu$, and $k \times k$ covariance matrix, $\Sigma$:

$$x \sim \mathcal{N}(\mu, \Sigma)$$

The infinite-dimensional extension of this distribution is referred to as a Gaussian Process and represents a distribution over functions. Formally, $p(g)$ is a GP if, for any finite-dimension $d$ subset of its random variables, the joint distribution $g_{x_1, \ldots, x_d}$ produces a Gaussian Distribution.

It may help to consider a single function $g(x)$ drawn from one of these GPs as an infinitely-long vector drawn from an extremely high-dimensional Gaussian [31]. In application, having a GP described in the same manner as a Gaussian Distribution, by
an infinite-dimensional scalar mean vector ($\infty 1$) and infinite-squared covariance matrix ($\infty \infty$), would be impractical. A GP over an infinite collection of random variables, is, instead, described with a mean function $m$ and a covariance function $k$

$$g(x) \sim GP(m(x), k(x, x'))$$

where

$$m(x) = E[x]$$

$$k(x, x') = E[(x - m(x))(x' - m(x'))]$$

The covariance function $k(x, x')$ of a distribution specifies the spatial relationship between two input sets $x$ and $x'$; that is, it acts as an information source detailing the degree at which a change in the distribution value at $x$ will correlate with a change in the distribution value at $x'$. The mean function can be represented as any function; however, the covariance function must result in a positive semi-definite matrix for any of its subset Gaussian Distributions.

This positive semi-definite requirement for covariance matrices is identical to Mercer’s condition for kernels and, therefore, any kernel-based function is a valid covariance. The most commonly applied kernel is known as a Squared Exponential or Radial Basis kernel, which is both stationary and isotropic and results in a homogenous and smooth function estimate.\(^3\)

$$k_{SE}(x, x') = \sigma^2 \exp \left[-\frac{1}{2} \left(\frac{x - x'}{\lambda}\right)^2\right]$$

where $\sigma$ is the output variance and $\lambda$ is the lengthscale.

A second common choice is known as the Matérn covariance function, which provides less smoothness through the reduction of the covariance differentiability.

$$k_{Matern}(x, x') = \sigma^2 \frac{2^{1-v}}{\Gamma(v)} \left[\frac{\sqrt{2v}|x - x'|}{\lambda}\right]^v K_v \left[\frac{\sqrt{2v}|x - x'|}{\lambda}\right]$$

where $\sigma$ is the output variance, $\Gamma$ is the gamma function, $K_v$ is a modified Bessel function, and $\lambda$ and $v$ are non-negative hyperparameters for lengthscale and smoothness respectively.

Other common but less used forms include periodic, which can capture repeated structure, and linear, a special case of which is known to model white noise. General references for families of correlation functions and possible combinations are provided by [1] and [10].

\(^3\)smooth covariance structures encode that the influence of a point on its neighbors is strong but nearly zero for points further away
2.2.2 Integrating with Noise

For simplification of notation throughout this section, \( \theta \) represents \( \langle \Psi, X \rangle \). Recall from Section 2, the three sources of variation: residual variability, observation error, and model inadequacy. While model inadequacy is explicitly captured through the objective function in the Integration stage, \( L \), the residual and observational variabilities have yet to be addressed.

Fortunately, the flexibility of using a GP prior allows for these remaining discrepancies to be adequately captured without significant complication via an addition to the covariance function [14]:

\[
L(Y, \phi(\theta)) \sim \mathcal{GP}(m(\theta), k(\theta, \theta') + \sigma_e^2 \delta_{\theta,\theta'})
\]

where \( \delta_{\theta,\theta'} \) is the Kronecker delta which is one if \( \theta = \theta' \) and zero otherwise and \( \sigma_e^2 \) is the variation of the error term.

This formulation continues to result in the variance of an input set \( \theta_j \) increasing away from the nearest alternative input \( \theta_i \) as before; however, it no longer results in zero if \( \theta_j = \theta_i \) but rather \( \delta \sigma_e^2 \).

To sample from this prior distribution, function values \( f_t \) would be drawn for \( t \) input sets \( \theta_{1:t} \) according to a Gaussian Distribution \( \mathcal{N}(\mu = m(\theta), K = k(\theta, \theta') + \sigma_e^2 \delta_{\theta,\theta'}) \) with the kernel matrix given by:

\[
K = \begin{bmatrix}
    k(\theta_1, \theta_1) + \delta \sigma^2 & \ldots & k(\theta_1, \theta_t) \\
    \vdots & \ddots & \vdots \\
    k(\theta_t, \theta_1) & \ldots & k(\theta_t, \theta_t) + \delta \sigma^2
\end{bmatrix}
\]

where \( k(\cdot, \cdot) \) is the chosen kernel function. Figure 3(a) shows an example of potential representative functions given only a prior knowledge of the simulator.

The posterior is conditioned from a joint distribution between the input sets which have been evaluated, designated as \( \theta_{ev} \), and the sets which have not, designed as \( \theta_* \):

\[
p \left( \begin{bmatrix}
    L(Y, \phi(\theta_{ev})) \\
    L(Y, \phi(\theta_*))
\end{bmatrix} \right) \sim \mathcal{N}\left( \mu = \begin{bmatrix}
    \mu_{ev} \\
    \mu_{ev*}
\end{bmatrix}, K = \begin{bmatrix}
    K_{ev,ev} & K_{ev,*} \\
    K_{*,ev} & K_{*,*}
\end{bmatrix}\right)
\]
To sample from this posterior distribution, function values, $L(Y, \phi(\theta_t))$, would be computed for $t$ input sets, $\theta_{1:t}$, according to a conditional Gaussian Distribution $p[L(Y, \phi(\theta_*)) \mid \theta_*, \theta_{ev}, L(Y, \phi(\theta_{ev}))] \sim \mathcal{N}(\mu_{post}, K_{post})$ with the following summary statistics:

$$
\mu_{post} = \mu_* + K_{ev, ev}^{-1} [L(Y, \phi(\theta_{ev})) - \mu_*], \quad K_{post} = K_{*, *} - K_{*, ev} K_{ev, ev}^{-1} K_{ev, *}
$$

(8)

Figure 3(b) shows an example of the potential representative functions given a few data points evaluated by the simulator.

2.3 Exploration

The Exploration stage provides a recommended input set for the next iteration. Typical Design of Experiment (DOE) methods such as randomization, factorial, and space-filling designs begin with the entire state space and provide a pre-determined list of candidates that should be run regardless of the previous evaluation outcomes. However, the large time-consumptions related to transportation simulators require attention be given to minimizing the number of samples without compromising the final recommendation and, consequently, candidates need to be ordered in such a manner that redundant sampling in areas of the state space already adequately mapped does not occur.

A machine learning technique known as active learning, also known as optimal experimental design in DOE, provides such a scheme. A utility function (a.k.a. acquisition function) is built to balance the exploration of unknown portions of the input sample space with the exploitation of all information gathered by the previous stages and cycles, resulting in a prioritized ordering reflecting the motivation and objectives behind the calibration effort. Formally, this is written as [33]:

$$
d^* = \arg\max_{d \in D} \mathbb{E}[U(d, \theta, f(\theta))]
$$

(9)

where $d^*$ is the optimal design choice, or input set, decision from the potential candidate set $D$; $U(\cdot)$ is the chosen utility function reflecting the experiment’s purpose (inference or prediction of unknown parameters $\theta$); and $f(\cdot)$ is the surrogate function.

The expectation of the utility function is taken over the posterior distribution calculated in the previous Integration stage.

$$
d^* = \arg\max_{d \in D} \mathbb{E}[U(d, \theta, f(\theta))]
$$

(10)

Thus, the optimal design choice will be the input set which maximizes the posterior expected utility. It should be noted at this framework does not aim to specify a single utility function to be used in all employed circumstances but to provide context behind which active learning utilities should be used for specific calibration situations.

In addition, in order to meet the needs of the Evaluation stage’s HPC capabilities, a batch-mode version will be utilized. Instead of providing a single recommendation, a predetermined number of recommendations should be produced using the appropriate utility function.\[4]

\[4\] expectations of the chosen recommendations yet to be evaluated by an Evaluation stage will be used when determining subsequent sampling decisions if the predetermined set is greater than 1
2.3.1 Acquisition Functions

Within the active learning framework, utility functions are often referred to as acquisition functions. With several different parameterized acquisition functions in the literature, it is often unclear which one to use. Some rely solely on exploration-based objectives, choosing samples in areas where variance is still large, or exploitation-based objectives, choosing samples where the mean is low, while others lie somewhere in between.

Determining the best acquisition function depends upon the overall purpose of the experiments. For example, predicting future values rely on minimizing variance across the state space and result in the chosen acquisition function skewing towards exploration while parameter estimation concentrates on finding the lowest or highest mean of the function through a bias towards exploitation. This framework concentrates on the later.

[3] discusses extensively several Bayesian utility functions and their non-Bayesian DOE equivalents. Below are two most widely used acquisition functions for Bayesian designs:

**Probability of Improvement** The Probability of Improvement function was first proposed by Harold Kushner [20]. Given the current minimum value found, \( f(\theta^+) \), where \( \theta^+ = \arg \max_{\theta_i \in \Theta_{\text{in}}} f(\theta_i) \), the function seeks to find the candidate from the unevaluated set of possible samples with the largest potential to improve upon the current minimal evaluation, \( f(\theta^+) \).

\[
PI(\theta) = P(f(\theta) \geq f(\theta^+))
\] (11)

For application to our Gaussian posterior distribution, this can be written as

\[
PI(\theta) = \Phi \left( \frac{\mu(\theta) - f(\theta^+)}{\sigma(\theta)} \right)
\] (12)

where \( \Phi(\cdot) \) is the Gaussian cumulative distribution function. In the original form above, this function is purely exploitative and relies heavily on the initial placement of the original samples. A simple modification exists, however, to encourage exploitation by adding a trade-off parameter, \( \lambda \geq 0 \), to the left-hand side.

\[
PI(\theta) = P(f(\theta) \geq f(\theta^+ + \lambda))
\] (13)
Although it can be set at the user’s discretion, Kusher recommended a decaying function to encourage exploration at the beginning and exploitation thereafter.

**Expected Improvement** Defined by Mockus [27], this utility function maximizes the expected improvement with respect to \( f(\theta^+) \):

\[
EI(\theta) = E[\max[0, f_{n+1}(\theta) - f(\theta^+)] | \theta_1..\theta_n]
\]

For application to our Gaussian posterior distribution, this can be written as

\[
EI(\theta) = \sigma(\theta)[u\Phi(u) + \phi(u)]
\]

where \( u = \frac{f(\theta^+) - \mu}{\sigma(\theta)} \), \( \Phi(\cdot) \) is the normal cumulative distribution, and \( \phi(\cdot) \) is the normal density function. The magnitude of the expected difference \( (f_{n+1}(\theta) - f(\theta^+)) \) is now explicitly taken into account.

With this formulation, a trade-off of exploitation and exploration is automatic; the expected improvement can be influenced by a reduction in the mean function (exploitation) or by increasing the variance (exploration).

Of note, in order to meet the needs of the Evaluation stage’s HPC capabilities, predetermined number of recommendations should be produced using the appropriate utility function; expectations of the chosen recommendations yet to be evaluated by an Evaluation stage will be used when determining subsequent sampling decisions if the predetermined set is greater than 1.

### 3 Dimensionality Reduction

The cost of constructing a surrogate function for a simulator increases exponentially as the dimension of the input space, or input parameters, increases; this is often referred to as the ‘curse of dimensionality’. Active subspace is one approach to reduce the parameter dimensions of the surrogate model by identifying and segregating the input dimensions into important, or active, and less-important, or inactive, directional categories. By identifying a reduced dimensional space, analysis methods such as Gaussian surrogate techniques, become more powerful and favorable.

Active subspace identifies linear combinations of inputs which significantly influence, on average, the output of the simulation when minor adjustments are made. These combinations are derived from the gradients between the inputs and quantities of interest decomposed into eigenvector principal components. The resulting eigenvalues of the decomposition are plotted on a log-scale and a gap, or space, is looked for.

For the calibration framework outlined in this paper, the chosen active subspace is applied to the surrogate model, \( f(\theta) \), outlined in Section 2.2 as follows:

\[
f(\theta) = f(\hat{W}_1 y + \hat{W}_2 z)
\]

where \( \hat{W}_1 \) contains the first \( n \) eigenvectors found to be active, \( \hat{W}_2 \) contains the remaining \( m - n \) eigenvectors, \( y \) contains the active subspace variables, and \( z \) contains the inactive subspace variables.
Figure 5: An example of the eigenvalues plotted for active subspace detection. The largest eigenvalue gap is designated by a red circle.

Consider the following algorithmic procedure to determine the active subspace for a generic function $f(x)$:

1. Normalize the input parameters around zero to remove units and prevent larger inputs from biasing the subspace

2. Draw $i = 1..N$ independent samples according to a pre-determined sampling method $\rho(x)$, such as a uniform sampling over a hypercube

3. For each sample $x_i$, compute the quantity of interest $f_i = f(x_i)$

4. Use a gradient approximation method to determine the gradients $\nabla_x f_i$. For example, use least-squares to fit the coefficients $\hat{\beta_0}$ and $\hat{\beta}$ of a local linear regression model

$$f_i \approx \hat{\beta}_0 + \hat{\beta}^T x_i$$

$$\nabla_x f_i = \hat{\beta}$$

5. Compute the matrix $\hat{\mathbf{C}}$ and its eigenvalue decomposition

$$\hat{\mathbf{C}} = \frac{1}{M} \sum_{i=1}^{M} (\nabla_x f_i)(\nabla_x f_i)^T = \hat{\mathbf{W}} \hat{\lambda} \hat{\mathbf{W}}^T$$

Where $\hat{\mathbf{W}}$ is the matrix of eigenvectors, and $\hat{\lambda} = \text{diag}(\hat{\lambda}_1, ..., \hat{\lambda}_m)$ is the diagonal matrix of eigenvalues ordered in decreasing order.

For more information behind the derivation of this procedure and its variations, see [7].

Once the procedure above is completed, the eigenvalues should be plotted and large gaps identified, as shown in [5]. Because the error is inversely proportional to the corresponding gap in the eigenvalues, the dimension $n$ should be chosen such that the largest eigenvalue gap is the separator. These $n$ directions, where $n$ is less than the original dimension set $m$, are considered the active subspace; the remaining $m - n$ directions are the inactive subspace.
If no gap can be found, compiling larger sets of eigenvalues or sampling more within the current eigenvalue framework to increase the eigenvalue accuracy is suggested. For example, if the gap between the 2nd and 3rd eigenvalues is larger than the gap between the first and second eigenvalues, the estimates of a 2-dimensional active subspace is more accurate than a 1-dimensional subspace, as shown in Figure 5.

Using this formulation, the inactive variables \( z \) can be fixed at nominal values while the important variables are varied. This follows the concept that small perturbations in \( z \) change \( f \) relatively little, on average. For situations in which the inactive eigenvalues are exactly zero, \( f(x) \) can be projected onto the active subspace without regard to the inactive subspace by setting \( z = 0 \):

\[
\theta \approx f(\hat{W}_1 y) \approx f(\hat{W}_1 \hat{W}_1^T \theta) \tag{17}
\]

However, for situations in which the inactive eigenvalues are small but not zero, the active subspace structure should be constructed by optimizing over \( y \) such that:

\[
\min_{y \in Y} \left[ \min_{z \in Z} f(\hat{W}_1 y + \hat{W}_2 z) \right] \text{ subject to } x_{lb} - \hat{W}_1 y \leq \hat{W}_2 z \leq x_{ub} - \hat{W}_1 y \tag{18}
\]

where \( x_{lb} \) and \( x_{ub} \) are the lower and upper bounds of the original \( x \) state-space, respectively.

### 4 Empirical Results

For the purposes of analysis and demonstration, we evaluate the calibration framework using a commonly cited transportation network known as Sioux-Falls [21], which consists of 24 zones and 24 intersections with 76 directional roads, or arcs. The network structure and input data provided by [37] have been adjusted from the original dataset to approximate hourly demand flows in the form of Origin-Destination (O-D) pairs, the simulation’s input set.

The input data is provided to a simulator package which implements the iterative Frank-Wolfe method to determine the traffic equilibrium flows and outputs average travel times across each arc.

Due to limited computing availability, we treat only the first twenty O-D pairs as unknown input variables which need to be calibrated while the other O-D pairs are assumed to be known. Random noise is added to the simulator to emulate the observational and variational errors expected in real-world applications.

The calibration framework’s objective function is to minimize the mean discrepancy between the simulated travel times resulting from the calibrated O-D pairs and the ‘true’ times resulting from the full set of true O-D pair values. To quantify this discrepancy, the mean squared error function \( L \) is used on the first 10 links:

\[
L [y, \phi(\theta)] = \frac{1}{10} \sum_{i=1}^{10} (y - \phi(\theta_i))^2. \tag{19}
\]
Figure 6: (a) Comparison of Calibrated and True Travel Time Outputs with Above Average Differences. (b) Results of demand matrix calibration using Bayesian optimization. Dimensionality reduction was performed using active substance approach.

Argonne National Laboratory’s Extreme-Scale Model Exploration with Swift (EMEWS) was used as the HPC program, which coordinates worker units to run the simulation code across multiple processors. The master controller, developed in Swift/T, successfully interacted with the Bayesian optimization model exploration program.

For the integration and exploration phases, we utilized and integrated into our HPC framework portions of a python package known as Spearmint, which was developed by the authors of and for deployment of the hyper-parameter tuning algorithms documented in [35]. Points were densely sampled on the unit hypercube and evaluated according to the Expected Improvement utility criterion.

Finally, the network setup was run using three batch sizes, the number of points chosen during each exploration phase, (1,6,10) and, to demonstrate dimensionality reductions, the experiment is run with and without the Active Subspace preprocessing developed by Constantine et al [7].

Overall, the performance of the calibration framework provided a calibrated solution set which resulted in outputs, on average, within 4% of the experiment’s true output. With a standard deviation of 5.66%, Figure 6(a) provides a visualization for those links which possessed greater than average variation from the true demand’s output.

Most notably, the inclusion of pre-processing via Active Subspaces brought the dimensionality of the problem to 2 from 20 and ultimately found the optimal minimum for calibration, see Figure 6(b). Its quick convergence and low variability show promise for larger experiments with dimensionality in the 100s.

Finally, Figure 7(a) shows a definitive trade-off between the utilization of parallel evaluations and the performance in minimizing the objective function. Using a batch size of 6, this experiment completed 8 iterations before being stopped. Figure 7(b) shows how the reduced dimension method quickly found a minimum function value while the original dimension method took several additional iterations before finding significant reductions. While the time constraints for this particular experiment are lax, more complex examples will result in a need to scrutinize the advantages to find the correct balance.
Figure 7: (a) Comparison of Batch Size Influence in terms of Objective Evaluations applied to original parameter space (blue line) and reduced dimensionality parameter space (black line). (b) Comparison of Minimum Function Value in terms of Iterations using a Batch Size of 6. Dimensionality reduction was performed using active substance approach.

5 Conclusion

We presented a framework for performing Bayesian calibration for transportation problems and introduced a Gaussian Process treatment in conjunction with active learning methods to maximize potential computational and time constraints. Although active subspace has potential for dimensionality reductions, additional techniques need to be further explored. Finally, the optimization model has been developed and demonstrated on the problem of updating entries of the OD matrix. However, our framework is input agnostic and can be used to calibrate other types of inputs, such as parameters of ABM models or land use variables. Further work will include the application of this framework to an implemented ABM transportation simulation.

References

[1] Petter Abrahamsen. A Review of Gaussian Random Fields and Correlation Functions, April 1997.

[2] Eric Brochu, Vlad M. Cora, and Nando De Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv preprint arXiv:1012.2599, 2010.

[3] Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. Statistical Science, pages 273–304, 1995.

[4] R.L. Cheu, X. Jin, K.C. Ng, Y.L. Ng, and D. Srinivasan. Calibration of FRESIM for Singapore expressway using genetic algorithm. Journal of Transportation Engineering, 124(6):526–535, November 1998.

[5] Linsen Chong. Computationally efficient simulation-based optimization algorithms for large-scale urban transportation problems. Thesis, Massachusetts Institute of Technology, 2017.
[6] Ernesto Cipriani, Michael Florian, Michael Mahut, and Marialisa Nigro. A gradient approximation approach for adjusting temporal origin destination matrices. *Transportation Research Part C: Emerging Technologies*, 19(2):270–282, April 2011.

[7] Paul G. Constantine. *Active Subspaces: Emerging Ideas for Dimension Reduction in Parameter Studies*. Society for Industrial and Applied Mathematics, 2015.

[8] C. Danielski, T. Kacprzak, G. Tinetti, and P. Jagoda. Gaussian Process for star and planet characterisation. *arXiv:1304.6673 [astro-ph]*, April 2013. arXiv: 1304.6673.

[9] Tamara Djukic, Gunnar Fløtterød, Hans Van Lint, and Serge Hoogendoorn. Efficient real time OD matrix estimation based on Principal Component Analysis. In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pages 115–121. IEEE, 2012.

[10] David Duvenaud. *Automatic model construction with Gaussian processes*. PhD thesis, University of Cambridge, 2014.

[11] Gunnar Fløtterød. A general methodology and a free software for the calibration of DTA models. In *The Third International Symposium on Dynamic Traffic Assignment*, 2010.

[12] Gunnar Fløtterød, Michel Bierlaire, and Kai Nagel. Bayesian demand calibration for dynamic traffic simulations. *Transportation Science*, 45(4):541–561, 2011.

[13] Robert B. Gramacy and Herbert K. H. Lee. Adaptive Design and Analysis of Supercomputer Experiments. *Technometrics*, 51(2):130–145, May 2009.

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

[15] Robert B. Gramacy and Nicholas G. Polson. Particle Learning of Gaussian Process Models for Sequential Design and Optimization. *Journal of Computational and Graphical Statistics*, 20(1):102–118, January 2011.

[16] David K. Hale, Constantinos Antoniou, Mark Brackstone, Dimitra Michalaka, Ana T. Moreno, and Kavita Parikh. Optimization-based assisted calibration of traffic simulation models. *Transportation Research Part C: Emerging Technologies*, 55:100–115, June 2015.

[17] Martin L. Hazelton. Statistical inference for time varying origin destination matrices. *Transportation Research Part B: Methodological*, 42(6):542–552, July 2008.

[18] Dave Higdon, James Gattiker, Brian Williams, and Maria Rightley. Computer Model Calibration Using High-Dimensional Output. *Journal of the American Statistical Association*, 103(482):570–583, June 2008.

[19] Marc C. Kennedy and Anthony O’Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, January 2001.
[20] H J Kushner. A new method of locating the maximum point of an arbitrary multi-peak curve in the presence of noise. *Journal of Basic Engineering*, 86:97–106, 1964.

[21] Larry J. LeBlanc, Edward K. Morlok, and William P. Pierskalla. An efficient approach to solving the road network equilibrium traffic assignment problem. *Transportation Research*, 9(5):309–318, 1975.

[22] Jung Beom Lee and Kaan Ozbay. New calibration methodology for microscopic traffic simulation using enhanced simultaneous perturbation stochastic approximation approach. *Transportation Research Record*, (2124):233–240, 2009.

[23] Lu Lu, Yan Xu, Constantinos Antoniou, and Moshe Ben-Akiva. An enhanced SPSA algorithm for the calibration of Dynamic Traffic Assignment models. *Transportation Research Part C: Emerging Technologies*, 51:149–166, February 2015.

[24] T. Ma and B. Abdulhai. Genetic algorithm-based optimization approach and generic tool for calibrating traffic microscopic simulation parameters. *Intelligent Transportation Systems and Vehicle-highway Automation 2002: Highway Operations, Capacity, and Traffic Control*, (1800):6–15, 2002.

[25] Deepak K. Merchant and George L. Nemhauser. A Model and an Algorithm for the Dynamic Traffic Assignment Problems. *Transportation Science*, 12(3):183–199, August 1978.

[26] Deepak K. Merchant and George L. Nemhauser. Optimality Conditions for a Dynamic Traffic Assignment Model. *Transportation Science*, 12(3):200–207, August 1978.

[27] J Mockus, V Tiesis, and A Zilinskas. The application of Bayesian methods for seeking the extremum. In *Towards Global Optimization*, volume 2, pages 117–129. Elsevier Science Ltd., 1978.

[28] Kai Nagel and Gunnar Fløttø. Agent-based traffic assignment: Going from trips to behavioural travelers. In *Travel Behaviour Research in an Evolving World: Selected papers from the 12th international conference on travel behaviour research*, pages 261–294. International Association for Travel Behaviour Research, 2012.

[29] J. Ozik, N. Collier, J.M. Wozniak, and C. Spagnuolo. From Desktop to Large-Scale Model Exploration with Swift/T. *Proceedings of the 2016 Winter Simulation Conference*, 2016.

[30] Srinivas Peeta and Athanasios K. Ziliaskopoulos. Foundations of dynamic traffic assignment: The past, the present and the future. *Networks and Spatial Economics*, 1(3):233–265, 2001.

[31] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass, 2006. OCLC: ocm61285753.
[32] Philip A. Romero, Andreas Krause, and Frances H. Arnold. Navigating the protein fitness landscape with Gaussian processes. *Proceedings of the National Academy of Sciences*, 110(3):E193–E201, 2013.

[33] Elizabeth G. Ryan. *Contributions to Bayesian experimental design*. PhD thesis, Queensland University of Technology, 2014.

[34] Jerome Sacks, William J. Welch, Toby J. Mitchell, and Henry P Wynn. Design and Analysis of Computer Experiments. *Statistical Science*, 4(4):409–423, November 1989.

[35] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[36] Jasper Snoek, Kevin Swersky, Richard S. Zemel, and Ryan P. Adams. Input Warping for Bayesian Optimization of Non-Stationary Functions. In *ICML*, pages 1674–1682, 2014.

[37] Ben Stabler. *TransportationNetworks: Transportation Networks for Research*, September 2017. original-date: 2016-03-12T22:38:10Z.

[38] Miao Yu and Wei (David) Fan. Calibration of microscopic traffic simulation models using metaheuristic algorithms. *International Journal of Transportation Science and Technology*, 6(1):63–77, June 2017.

[39] Chao Zhang, Carolina Osorio, and Gunnar Fltterd. Efficient calibration techniques for large-scale traffic simulators. *Transportation Research Part B: Methodological*, 97:214–239, March 2017.

[40] Zheng Zhu, Chenfeng Xiong, Xiqun (Michael) Chen, and Lei Zhang. Calibrating supply parameters of large-scale DTA models with surrogate-based optimisation. *IET Intelligent Transport Systems*, April 2018.