Bayesian Optimisation with Input Uncertainty Reduction

Juan Ungredda
Michael Pearce
Supervisor: Juergen Branke

University of Warwick

25 April 2019

Outline

Introduction

A simulation model of a call-centre, knowing:

- **Calls Arrival Rate:** Poisson process at a fixed rate Λ.
- **Λ**: Not known, but uncertainty is well modelled given observed data.
- **Service time:** exponentially distributed known mean μ^{-1}
- **Costs**: Salaries (S), and penalty costs (PC) per minute that customers wait on hold.

Objective:

- Minimise: $Total_{cost} = Total_S + Total_{PC}$

Decision Variable:

- Staffing Level

Introduction

Question:

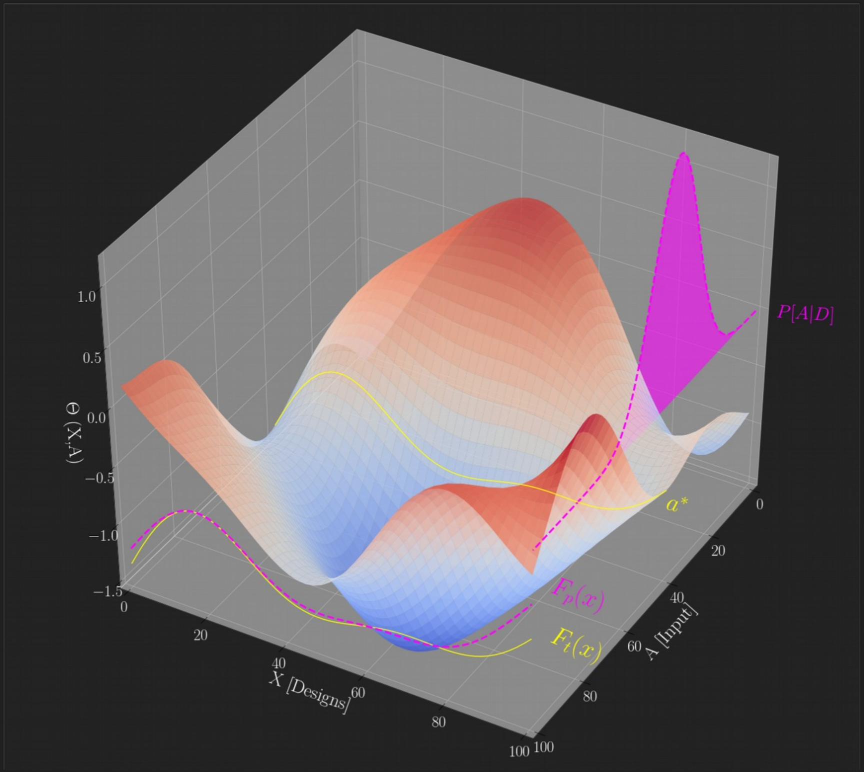
Should we run additional simulations to learn about the "total cost" given staff allocation and current uncertainty for Λ?

OR

Should we collect more data to reduce the input uncertainty?

Introduction

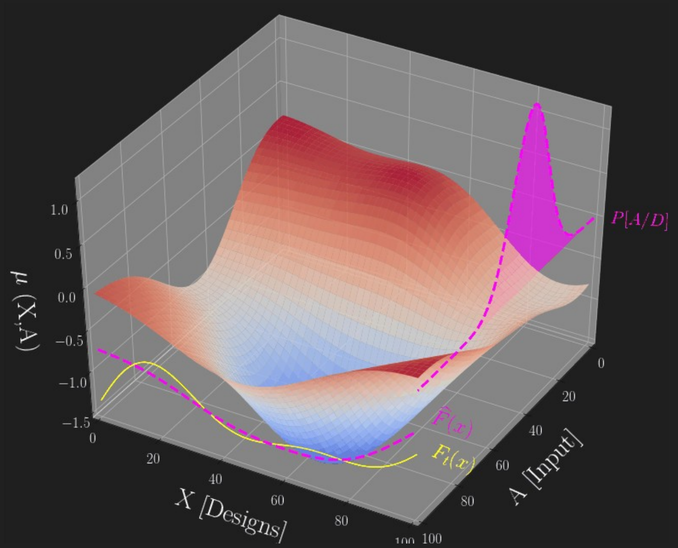
- Output of simulation: $\Theta(x, a)$, given X [Designs] and A [Input].
- True perfomance of x: $F_t(x) = \Theta(x, a^*)$ given true input a^*
- Expected performance of x: $F_p(x) = \int_A \Theta(x, a)\mathbb{P}[a|D]da$ given the data D.



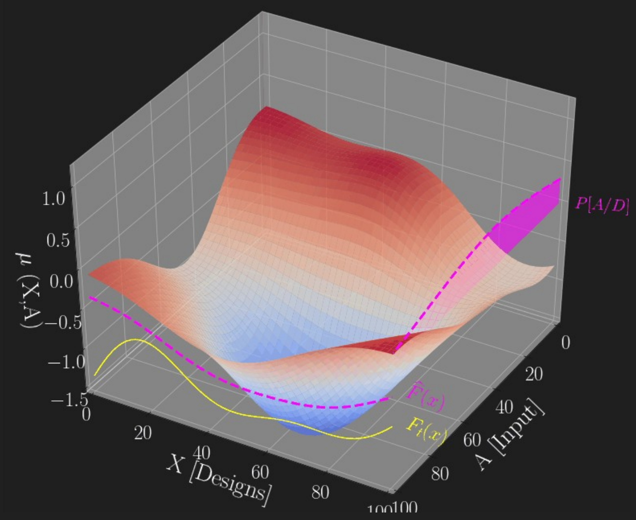
Introduction

Approximating the simulation runs, $\Theta(x, a)$, with $\mu(x, a)$.

Sample $\{(x, a)\}$
Update $\mu(x, a)$



Collect data
Update $\mathbb{P}[A|D]$



$$\hat{F}(x) = \int_A \mu(x, a)\mathbb{P}[a|D]da$$

Motivation

- Goal: Minimise the difference between the maximum of the expected and true performance

Constraint:

- Fixed budget N.

Standard Approach:
Decide how to split N, then first collect more input distribution data, spend remaining budget on simulations.

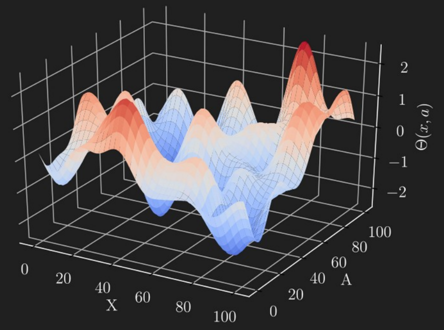
Proposed Approach:
Sequentially allocate budget to either input data collection and update $\mathbb{P}[a|D]$, or run more simulations and update $\mu(x, a)$, depending on what seems to have largest benefit

Gaussian Process Approximation

Consider the possible designs $x \in X$, an unknown input value $a \in A$, and a function $\theta \colon X \times A \to \mathbb{R}$.

$$f(x, a) = \theta(x, a) + \epsilon$$

where $\epsilon \sim N(0, \sigma_\epsilon^2)$



Modelled by the mean $\mu^n(x, a)$ and covariance $k^n((\mathbf{x}, \mathbf{a}); (\mathbf{x}', \mathbf{a}'))$ of a Gaussian process.

Problem Formulation: Expected Performance

Identify the design **x** that maximises the expected performance:

$$\hat{F}(x) = \mathbb{E}_{\mathbb{P}[a|D^m]}[\mu(\mathbf{x}, \mathbf{a})] = \int_A \mu^n(x, a)\mathbb{P}[a|D^m]da$$

Data collection from simulation runs:

$R^n = \{(x, a, y)^i | i = 1, ..., n\}$

Data collection from input sources:

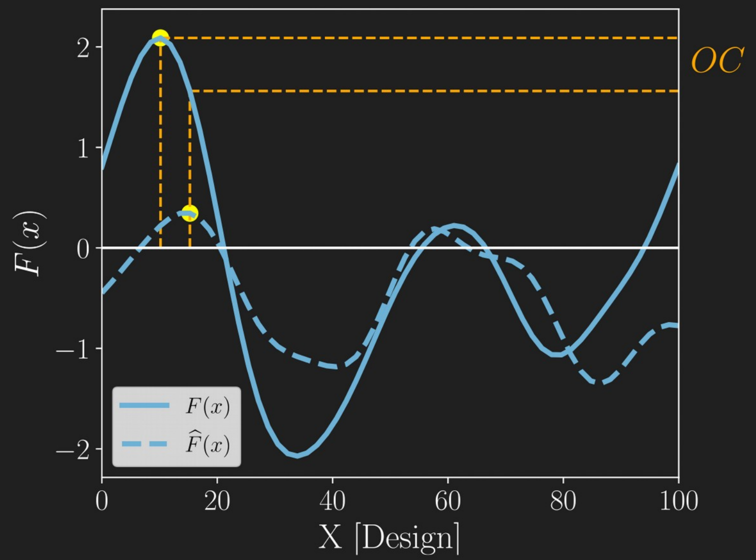
$D^m = \{(j, d)^i | i = 1, ..., m\}$; d is an observation from the input $j \in \{1, ..., l\}$

Problem Formulation: Quality of Sampling

The Opportunity Cost (OC): Difference in true performance between the design with the highest predicted value and the true best design

$$OC = \max_{x} F(x) - F(x_r)$$

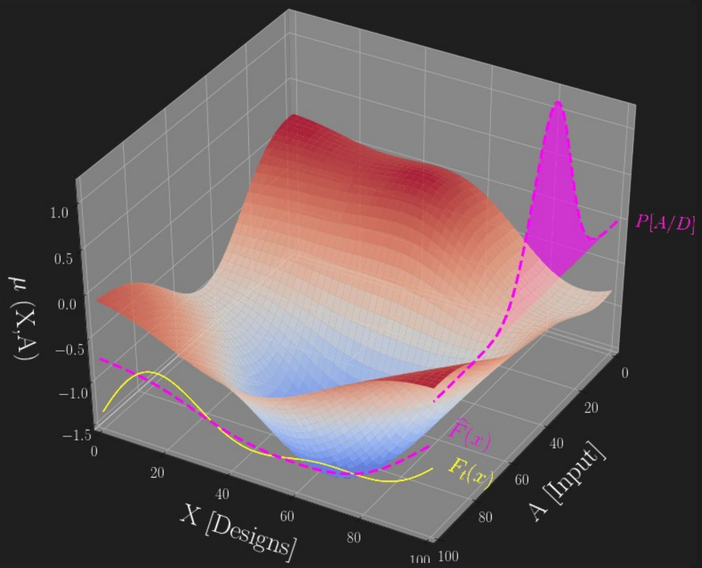
where $F(x) = \theta(x, a^*)$ and $\mathbf{x}_r = \arg\max_{\mathbf{x}} \hat{F}(\mathbf{x})$



Algorithm: Knowledge Gradient for Input Uncertainty

[Pearce and Branke, (2017)]

- From current $\max_{\mathbf{x}\in X}\{\widehat{F}^n(\mathbf{x})\}$
- Given a sample $(\mathbf{x},\mathbf{a})^{n+1}$
- Update posterior $\mu^n(x,a)$
- Update to $\max_{\mathbf{x}\in X}\{\widehat{F}^{n+1}(\mathbf{x})\}$



$$\widehat{F}(x) = \int_A \mu^n(x,a)\mathbb{P}[a|D]da$$

Algorithm: Knowledge Gradient for Input Uncertainty

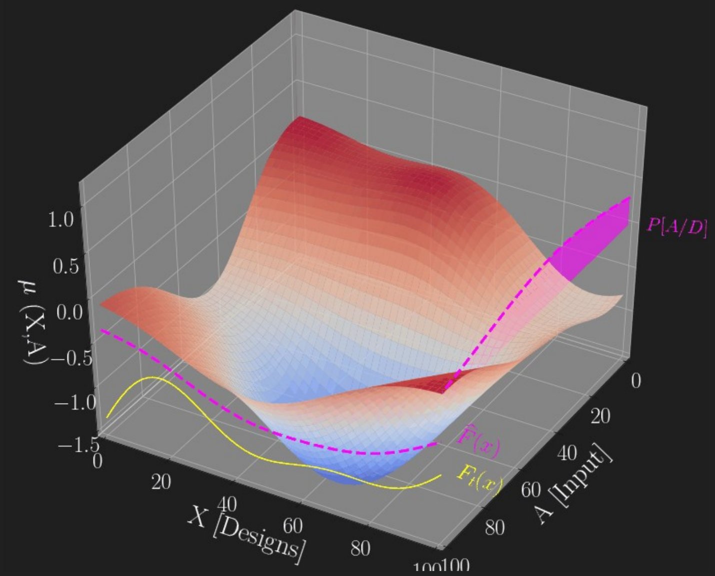
[Pearce and Branke, (2017)]
Given a discretised set X, evaluate sample $(\mathbf{x}, \mathbf{a})^{n+1}$ such maximises,

$$KG_R(\mathbf{x}, \mathbf{a}) = \mathbb{E}[\max_{\mathbf{x}'' \in X}\{\hat{F}^{n+1}(\mathbf{x}'')\}|(\mathbf{x}, \mathbf{a})^{n+1}] - \max_{\mathbf{x}' \in X}\{\hat{F}^{n}(\mathbf{x}')\}$$

Algorithm: Input Uncertainty Reduction

Collect data

Update $\mathbb{P}[A|D]$



$$\hat{F}(x) = \int_A \mu^n(x,a)\mathbb{P}[a|D]da$$

Algorithm: Input Uncertainty Reduction

Given a sample $(j, d)^{m+1}$ from an input source,

$$Loss^j(D^{m+1}) = \mathbb{E}_{\mathbb{P}[d_{m+1}|D^m]}[\mathbb{E}_{\mathbb{P}[a|D^{m+1}]}[\max_x \mu(\mathbf{x}, a) - \mu(\mathbf{x}_r(D^{m+1}), a)]]$$

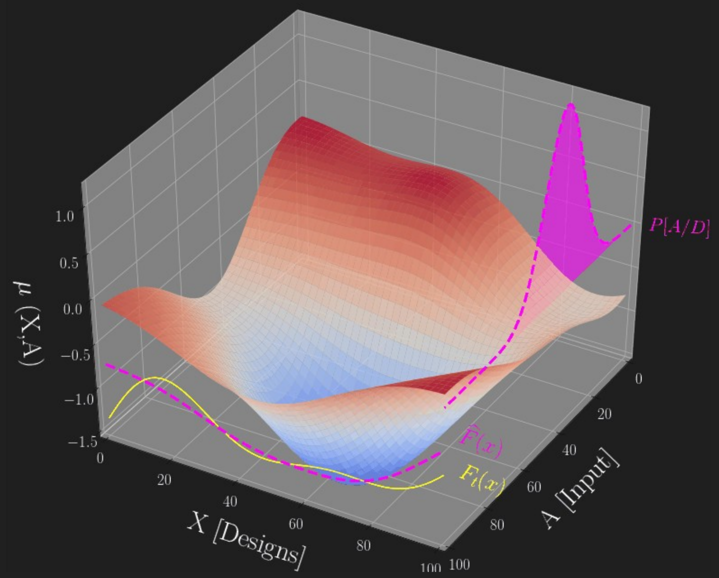
Finally, the expected difference reduction is as follows:

$$\begin{aligned} KG_l^j &= Loss^m(D^m) - Loss^j(D^{m+1}) \\ &= \mathbb{E}_{\mathbb{P}[d_{m+1}|D^m]}[\mathbb{E}_{\mathbb{P}[a|D^{m+1}]}[\mu(\mathbf{x}_r(D^{m+1}), a) - \mu(\mathbf{x}_r(D^m), a)]] \end{aligned}$$

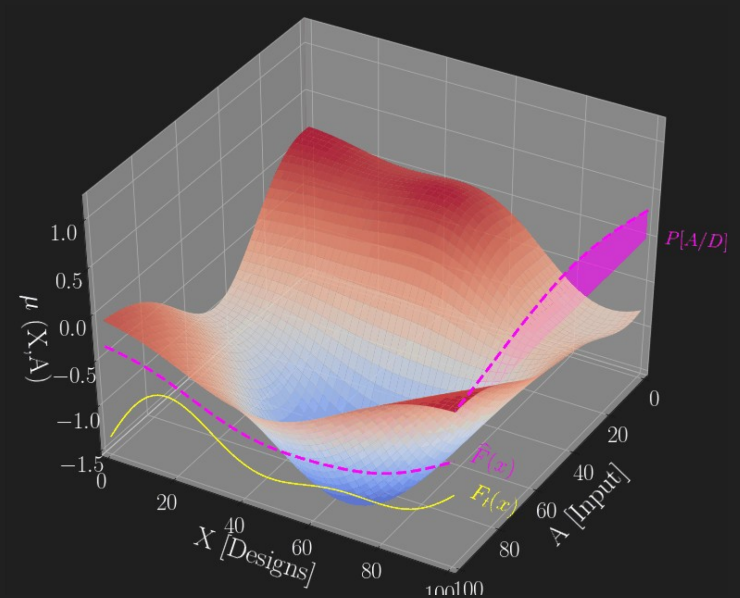
Algorithm: Decision Rule (DR)

The measure that gives greater improvement, either KG_R or KG_I^j for any of the inputs $j \in \{1, \ldots, n\}$, will state whether if we sample $(x, a)^{n+1}$, or $(j, d)^{m+1}$.

Sample $\{(x, a)\}$



Sample $(j, d)^{m+1}$



$$\widehat{F}(x) = \int_A \mu(x, a)^n \mathbb{P}[a|D] da$$

Numerical Experiments: Test Problem

Test Function (1 Design, 1 Input):

- Gaussian process with a squared exponential kernel.
- Hyperparameters: $l_{XA} = 10$, $\sigma_0^2 = 1$ $\sigma_\epsilon^2 = 0.1$
- Design $x \in X = [0, 100]$, and an input $a \in A = [0, 100]$.

Input parameter:

- Data $d^j \sim N(a_j^*, \sigma_j^2)$ for $j = 1$
- We use a Normal Likelihood and Uniform prior for inference $\mathbb{P}[A|D^m]$

Numerical Experiments: Test Problem

Test Function (1 Design, 2 Inputs):

- Gaussian process with a squared exponential kernel.
- Hyperparameters: $l_{XA} = 10$, $\sigma_0^2 = 1$ $\sigma_\epsilon^2 = 0.1$
- Design $x \in X = [0, 100]$, and an input $a^1, a^2 \in A = [0, 100]$.

Input parameter:

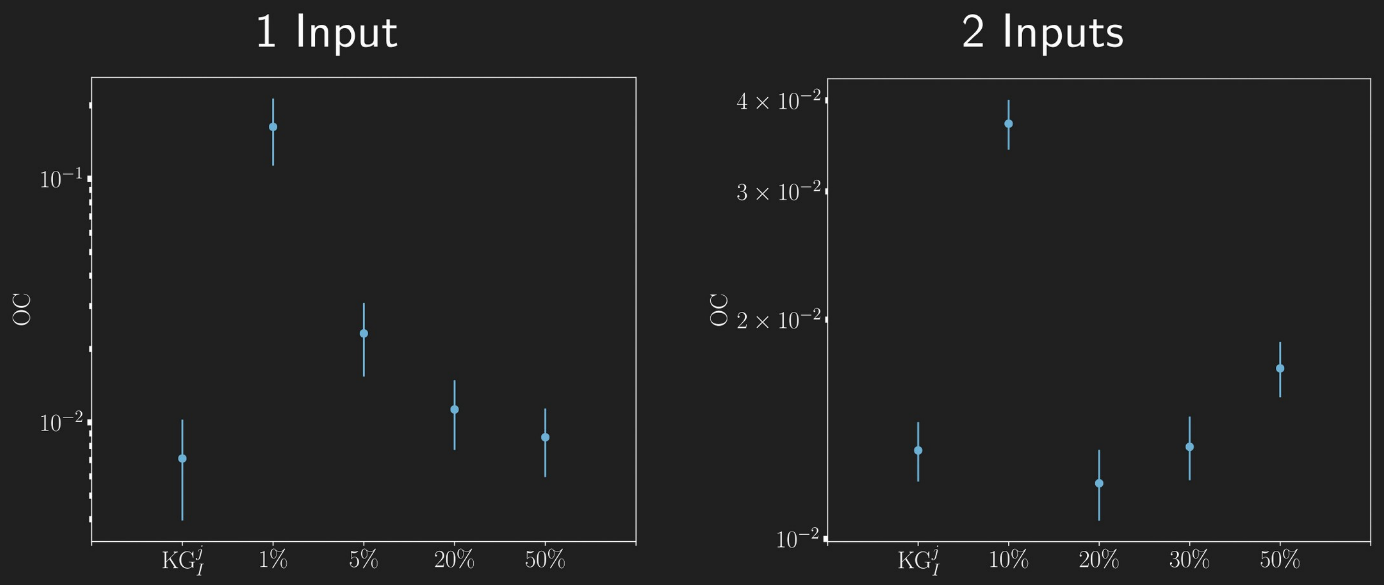
- Data $d^j \sim N(a_j^*, \sigma_j^2)$ for $j = 1, 2$
- We use a Normal Likelihood and Uniform prior for inference $\mathbb{P}[A|D^m]$

Numerical Experiments: Benchmark Method

Given a total budget of N and ratio p from total budget.

- Stage 1: Sample Np and update the input distribution $P[a_j|D^m]$. Samples are uniformly distributed for multiple inputs.
- Stage 2: Update $\mu^n(x, a)$ with $N(1-p)$ samples allocated using $KG_R(x, a)$.

Numerical Experiments: Results



Conclusions

- The algorithm is capable of balancing between running additional simulations and reducing the input uncertainty.
- Including KG_l^j to allocate samples presents a similar performance respect of choosing an "adequate" fixed proportion in a 2-stage sampling.
- The developed metric does not depend of parameters set by the user.