Bayesian Optimization for Simulation Optimization of Continuous Parameters

Jianzhong Du

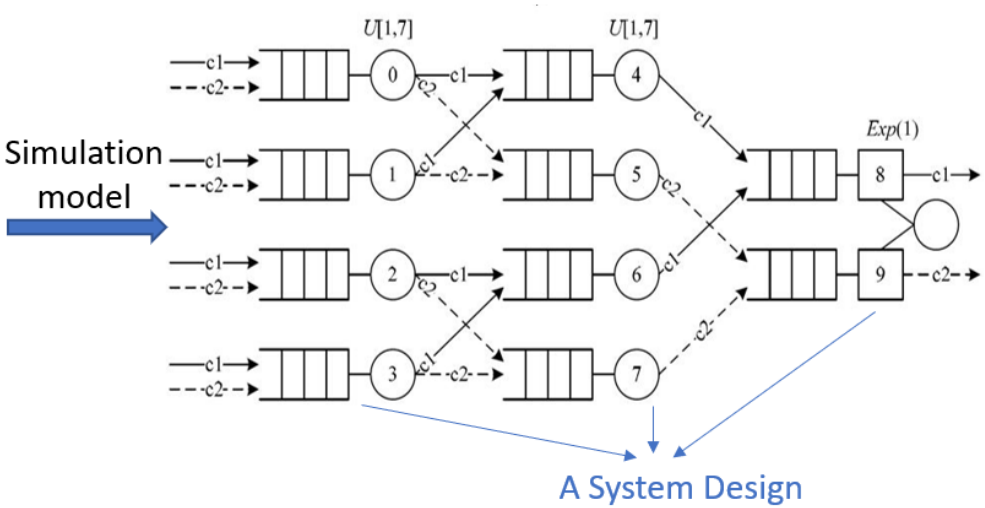
Fudan University

December 27, 2021

Outline

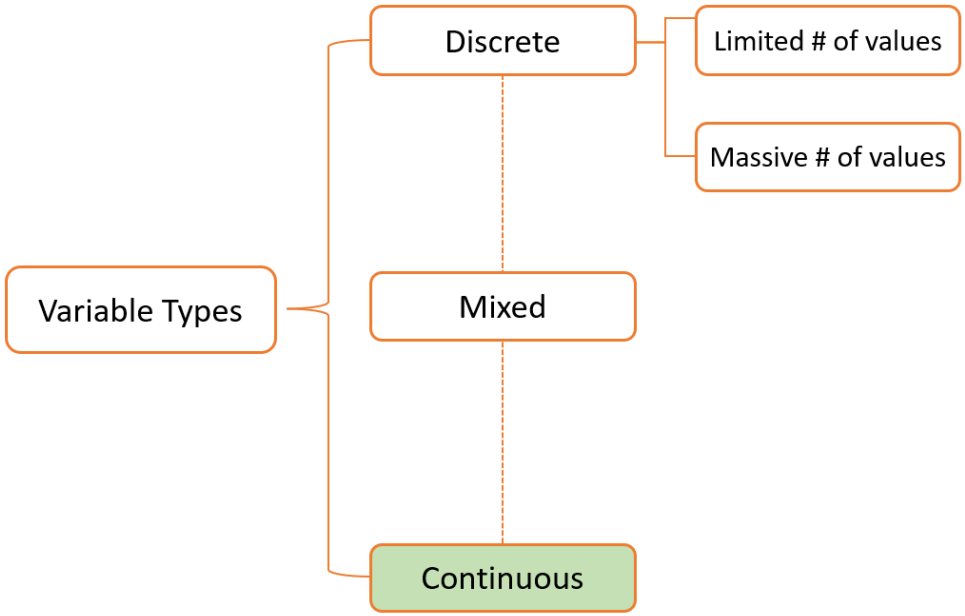
Outline

Simulation Optimization



1. Evaluate the performance of a system design.
2. Select the best design $\mathbf{x}^* = \arg\min_{\mathbf{x}} y(\mathbf{x})$.

Types of Design Variables and Samples



- Sample types:
  1. Noiseless
  2. Noisy
- Variable types.
- Today's topic: Continuous variables + noiseless/noisy samples

Outline

Study Goal: Bayesian optimization for simulation optimization of continuous parameters.

1. Noiseless samples:
   - Jones D R, Schonlau M, Welch W J. Efficient global optimization of expensive black-box functions[J]. *Journal of Global optimization*, 1998, 13(4): 455-492. (citations: 6538)
2. Noisy samples:
   - Scott W, Frazier P, Powell W. The correlated knowledge gradient for simulation optimization of continuous parameters using gaussian process regression[J]. *SIAM Journal on Optimization*, 2011, 21(3): 996-1026.

Outline

Components of Bayesian Optimization

A typical Bayesian optimization consists of two parts:

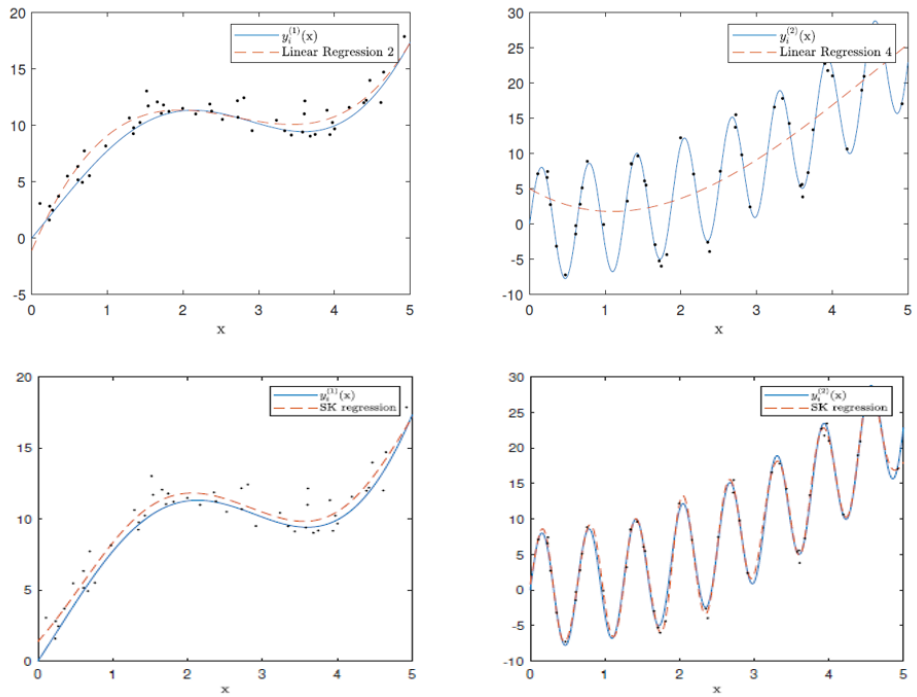
1. Gaussian process (stochastic kriging):
   predicting function values.
2. Sampling methods:
   determining the design point that should be sampled.
   (This slides discuss two acquisition functions (figure of merit): EI and KG.)

Outline

Surrogates

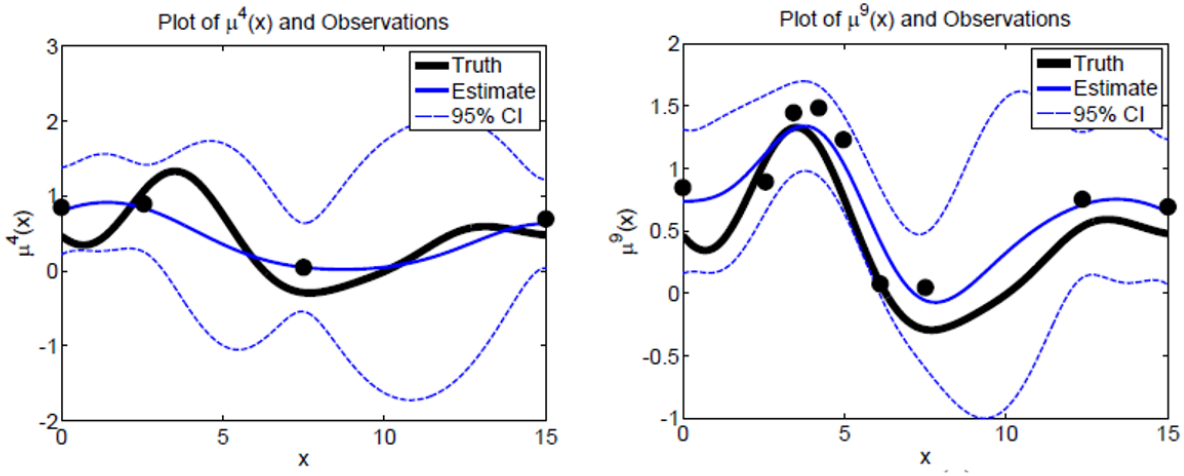
- Goal: to predict surface values $y(\mathbf{x})$, $\mathbf{x} \in \mathcal{X}$, given a limited number of random observations $Y(\mathbf{x}^i)$, $i = 0, 1, \ldots, n$.
- Typical surrogates:
  1. Linear basis function models.
  2. Gaussian process.
- Relationship: they can be unified through the ridge regularization (Hong and Zhang, 2021).

Two examples



- GP regression can capture fluctuated surfaces easily.
- However, linear basis function model can also achieve it.
- Advantage of GP: Bayesian learning.

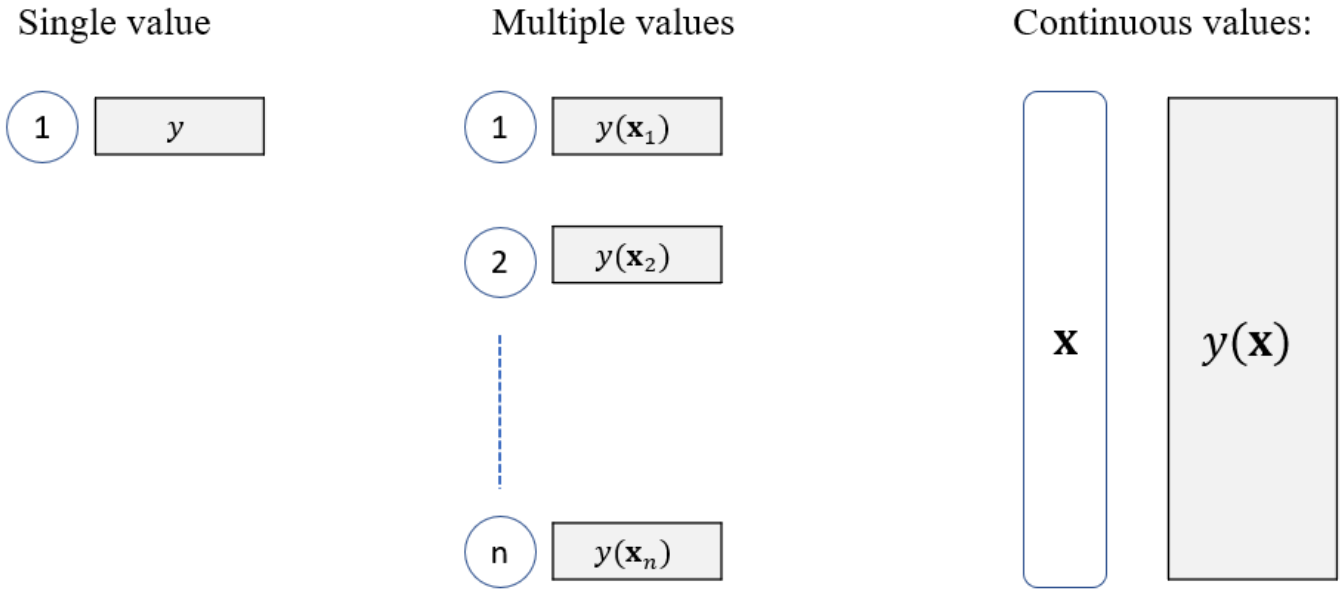
An example (Continued)



- Bayesian learning: construct Bayesian credible region (analogous to confidence intervals in frequentist statistics).

Outline

Learning Targets



General Framework of Bayesian Learning

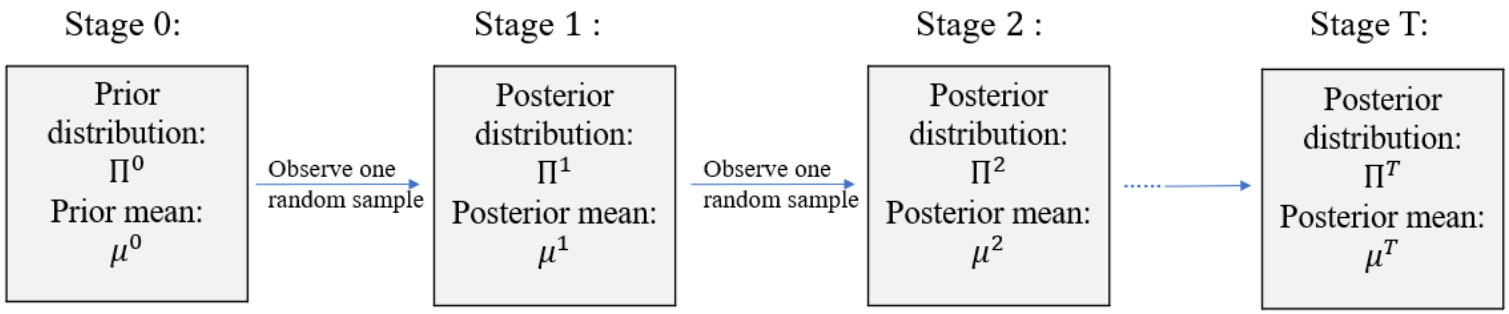
Subjective v.s. Objective. Let's flip a coin:

- Objective: the probability that a coin has a head is 1/2 (or 1/3 if the coin is uneven).
- Subjective: I think the probability that a coin has a head is within [1/3, 2/3] and has an uniform distribution.

Bayesian prior and posterior are subjective probabilities.

General Framework of Bayesian Learning

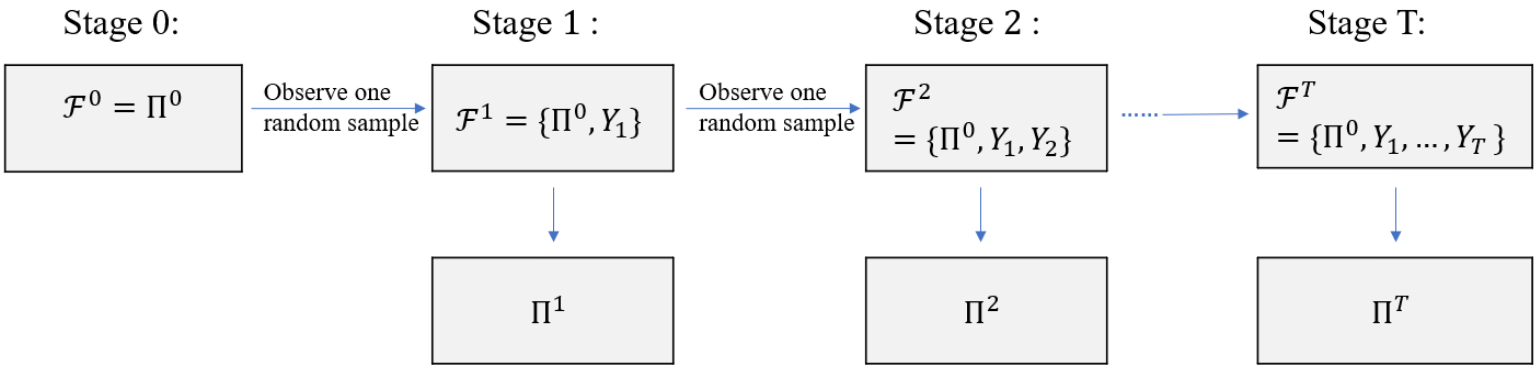
- Suppose we want to learn the value of $y(\mathbf{x}^0)$ (for short, $y(\mathbf{x}^0) := y$). The random sample satisfies $\mathrm{E}[Y_i] = y$.



- The unknown value y is regarded as a random variable.
- Subjective belief: $\mathrm{P}(y \le y_1) = 0.2$, $\mathrm{P}(y \le y_2) = 0.5$, $\mathrm{P}(y \le y_3) = 0.7$, ... Can be a distribution.
- We sequentially update the belief toward its value: $\Pi^0, \Pi^1, \ldots, \Pi^T$.

General Framework of Bayesian Learning

- Filtration:



- Example: Given y, the sample Y_1 follows $\mathcal{N}(y, 1/\tau)$ (τ is known).
  - Before observing Y_1, note that $Y_1 = y + (Y_1 - y)$.
  - In our belief, the prior of y as $\mathcal{N}(\mu^0, 1/\tau^0)$.
  - Sample noise $Y_1 - y \sim \mathcal{N}(0, 1/\tau)$, independent of y's belief.
  - So, in our belief, $Y_1 \sim \mathcal{N}(\mu^0, 1/\tau^0 + 1/\tau)$. (predictive distribution of Y_1)

General Framework of Bayesian Learning

- Example: Given y, the sample follows $\mathcal{N}(y, 1/\tau)$ and τ is known.
  - Given $\mathcal{F}^0$ and before observing Y_1, the joint distribution of $(y, Y_1)^\top$:

$$(y, Y_1)^\top \sim \mathcal{N}\left((\mu^0, \mu^0)^\top, \begin{pmatrix} 1/\tau^0 & 1/\tau^0 \\ 1/\tau^0 & 1/\tau^0 + 1/\tau \end{pmatrix}\right).$$

Hint:
$\mathrm{Cov}(y, Y_1) = \mathrm{Cov}(y, y+(Y_1-y)) = \mathrm{Cov}(y, y)+\mathrm{Cov}(y, Y_1-y) = 1/\tau^0.$

  - The conditional distribution of y given Y_1 is

$$y|Y_1 \sim \mathcal{N}\left(\frac{\tau^0\mu^0 + \tau Y_1}{\tau^0 + \tau}, \frac{1}{\tau + \tau^0}\right) \triangleq \mathcal{N}(\mu^1, 1/\tau^1).$$

(the posterior distribution of y after observing Y_1)

General Framework of Bayesian Learning

- Example: Given y, the sample follows $\mathcal{N}(y, 1/\tau)$ and τ is known.
  - Given $\Pi^0 = \mathcal{N}(\mu^0, 1/\tau^0)$, $\mu^1 = \frac{\tau^0\mu^0 + \tau Y_1}{\tau^0 + \tau}$ is random due to Y_1.
  - Note that the predictive distribution of Y_1 is

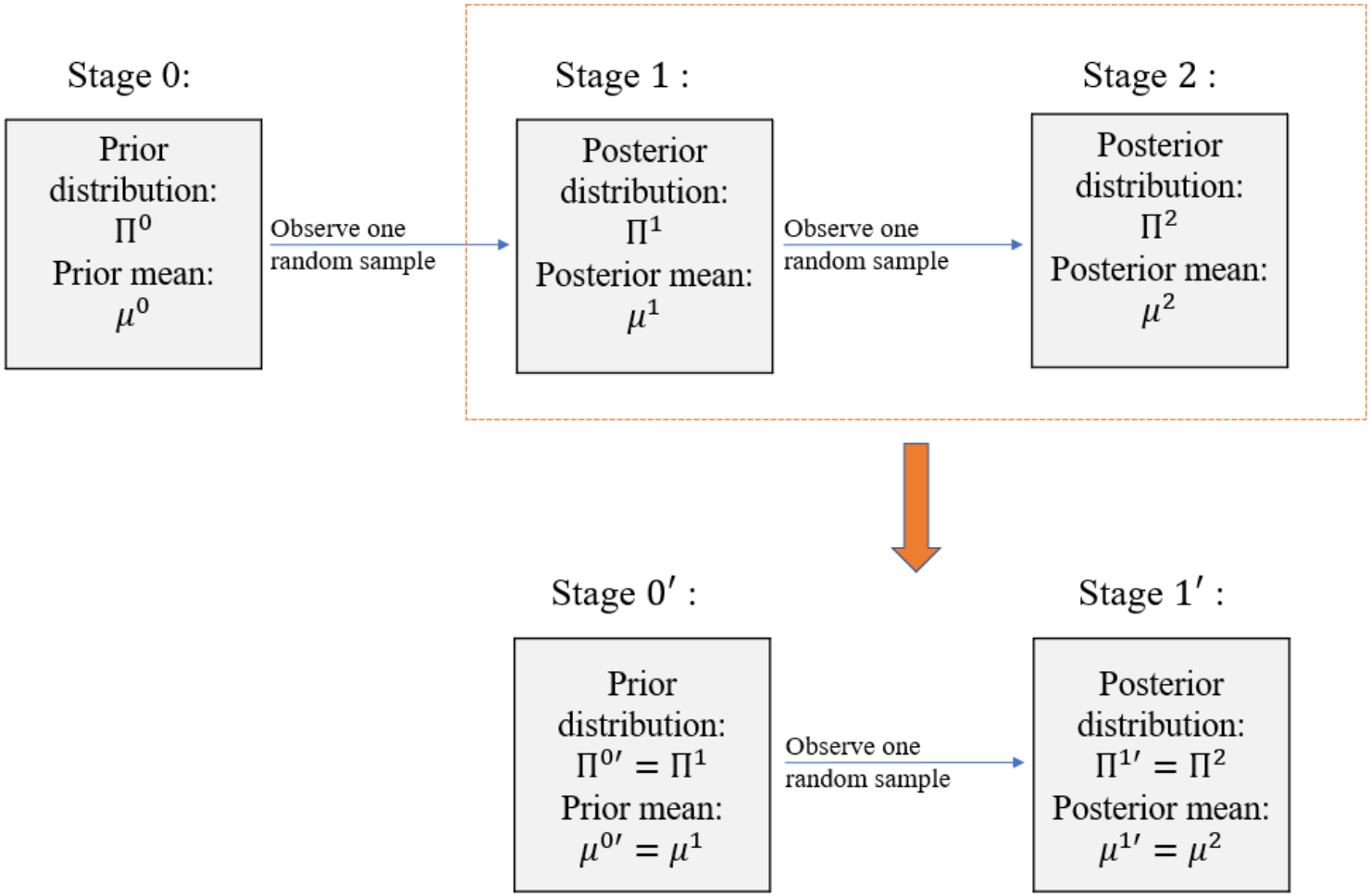
$$\mathcal{N}(\mu^0, 1/\tau^0 + 1/\tau).$$

  - The predictive distribution of $\mu^1 = \frac{\tau^0\mu^0}{\tau^0+\tau} + \frac{\tau}{\tau^0+\tau}Y_1$ given $\mathcal{F}^0$ is

$$\mathcal{N}\left(\mu^0, \frac{\tau}{\tau^0(\tau^0+\tau)}\right).$$

- In summary, we have learned given $\mathcal{F}^0$,
  - the predictive distribution of Y_1 and μ^1 before observing Y_1,
  - the posterior distribution of y ($\Pi^1 = \mathcal{N}(\mu^1, 1/\tau^1)$) after observing Y_1.

General Framework of Bayesian Learning



General Framework of Bayesian Learning

- Example: Given y, the sample follows $\mathcal{N}(y, 1/\tau)$ and τ is known.
  - Given $\mathcal{F}^1$ and before observing Y_2,
    - the predictive distribution of Y_2 is $\mathcal{N}(\mu^1, 1/\tau^1 + 1/\tau)$.
    - the predictive distribution of μ^2 is $\mathcal{N}\left(\mu^1, \frac{\tau}{\tau^1(\tau^1+\tau)}\right)$
  - Given $\mathcal{F}^2$ (given $\mathcal{F}^1$ and after observing Y_2), the posterior distribution of y is $\mathcal{N}(\mu^2, 1/\tau^2)$, where

$$\tau^2 = \tau + \tau^1 = 2\tau + \tau^0,$$

$$\mu^2 = \frac{\tau^1\mu^1 + \tau Y_2}{\tau^1 + \tau} = \frac{\tau^1 \frac{\tau^0\mu^0 + \tau Y_1}{\tau^0+\tau} + \tau Y_2}{2\tau + \tau^0} = \frac{\tau^0\mu^0 + \tau Y_1 + \tau Y_2}{2\tau + \tau^0}.$$

General Framework of Bayesian Learning

- Example: Given y, the sample follows $\mathcal{N}(y, 1/\tau)$ and τ is known.
  - Given $\mathcal{F}^{n-1}$ and before observing Y_n,
    - the predictive distribution of Y_n is $\mathcal{N}(\mu^{n-1}, 1/\tau^{n-1} + 1/\tau)$.
    - the predictive distribution of μ^n is $\mathcal{N}\left(\mu^{n-1}, \frac{\tau}{\tau^{n-1}(\tau^{n-1}+\tau)}\right)$
  - Given $\mathcal{F}^n$ (given $\mathcal{F}^{n-1}$ and after observing Y_n), the posterior distribution of y is $\mathcal{N}(\mu^n, 1/\tau^n)$, where

$$\begin{aligned}\tau^n =& \tau + \tau^{n-1} = n\tau + \tau^0,\\ \mu^n =& \frac{\tau^{n-1}\mu^{n-1} + \tau Y_1}{\tau^{n-1} + \tau} = \frac{\tau^0\mu^0 + \tau(Y_1 + Y_2 + \ldots Y_n)}{n\tau + \tau^0}.\end{aligned}$$

  - Question 1: what if the observation has no noise, i.e., $\tau = \infty$? $(y|y \sim \mathcal{N}(y, 0))$

Conjugate Family II

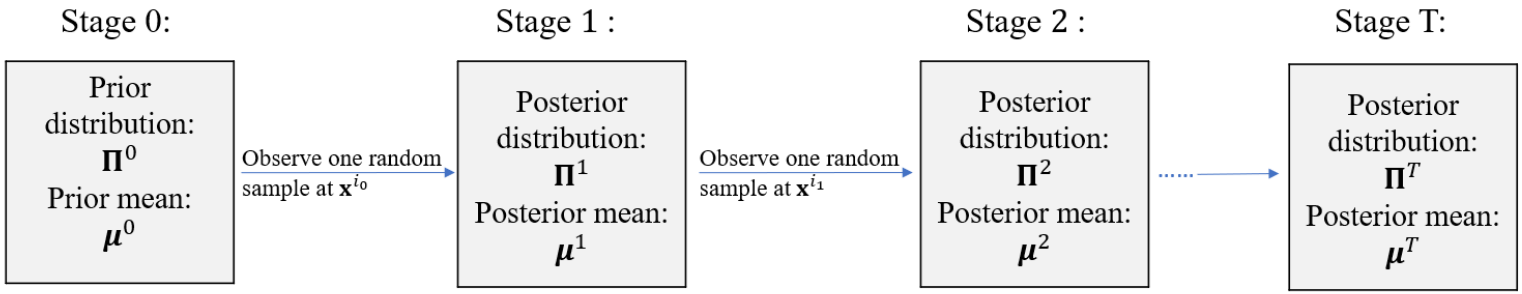
- Conjugacy: the posterior distribution is in the same family as the prior distribution.
- Other conjugate families:

Sampling Distribution	Conjugate Family
Exponential	Gamma
Poisson	Gamma
Uniform	Pareto
Bernoulli	Beta
Normal with unknown variance	Normal-Gamma

- Non-conjugate priors: posterior may not be analytically tractable.
  - numerical calculation: Markov chain Monte Carlo, importance sampling.

Bayesian Learning of Multivariate Normal

- A vector $\mathbf{y} = (y(\mathbf{x}^1), y(\mathbf{x}^2), \ldots, y(\mathbf{x}^d))^\top$ to estimate.



- The sample at $\mathbf{x}^i$ is normal: $\mathcal{N}(y(\mathbf{x}^i), \lambda(\mathbf{x}^i))$.
- Set the prior $\mathbf{\Pi}^0$ as $\mathcal{N}(\boldsymbol{\mu}^0, \Sigma^0)$. The posterior distribution $\mathbf{\Pi}^1$ after observing $Y(\mathbf{x}^{i_0})$ is $\mathcal{N}(\boldsymbol{\mu}^1, \Sigma^1)$ where

$$\begin{aligned}\boldsymbol{\mu}^1 =& \boldsymbol{\mu}^0 + \Sigma^0 e_{\mathbf{x}^{i_0}} \left(\Sigma^0(\mathbf{x}^{i_0}, \mathbf{x}^{i_0}) + \lambda(\mathbf{x}^{i_0})\right)^{-1} \left(Y(\mathbf{x}^{i_0}) - \mu^0(\mathbf{x}^{i_0})\right)\\ \Sigma^1 =& \Sigma^0 - \Sigma^0 e_{\mathbf{x}^{i_0}} \left(\Sigma^0(\mathbf{x}^{i_0}, \mathbf{x}^{i_0}) + \lambda(\mathbf{x}^{i_0})\right)^{-1} \left(\Sigma^0 e_{\mathbf{x}^{i_0}}\right)^\top\end{aligned}$$

- Hint: $\mathbf{\Pi}^1 = (y(\mathbf{x}^1), y(\mathbf{x}^2), \ldots, y(\mathbf{x}^d))^\top | Y(\mathbf{x}^{i_0})$.

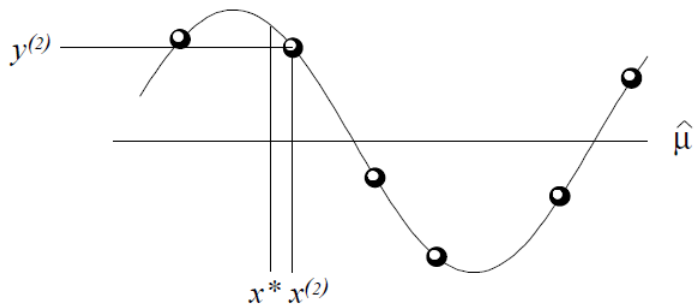
Answer to Questions for Noiseless Sampling

- Question 2: what if the observation has no noise, i.e., $\lambda(\mathbf{x}^n) = 0$?
  Prior:

$$(y(\mathbf{x}^0), y(\mathbf{x}))^\top \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}^0(\mathbf{x}^0) \\ \boldsymbol{\mu}^0(\mathbf{x}) \end{pmatrix}, \begin{pmatrix} \Sigma^0(\mathbf{x}^0, \mathbf{x}^0) & \Sigma^0(\mathbf{x}, \mathbf{x}^0) \\ \Sigma^0(\mathbf{x}^0, \mathbf{x}) & \Sigma^0(\mathbf{x}, \mathbf{x}) \end{pmatrix}\right)$$

  Posterior:

$$(y(\mathbf{x}^0), y(\mathbf{x}))^\top | y(\mathbf{x}^0) \sim$$
$$\mathcal{N}\left(\begin{pmatrix} y(\mathbf{x}^0) \\ \boldsymbol{\mu}^0(\mathbf{x}) + \frac{\Sigma^0(\mathbf{x}, \mathbf{x}^0)}{\Sigma^0(\mathbf{x}^0, \mathbf{x}^0)} \left(y(\mathbf{x}^0) - \mu^0(\mathbf{x}^0)\right) \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0 & \Sigma^0(\mathbf{x}, \mathbf{x}) - \frac{(\Sigma^0(\mathbf{x}, \mathbf{x}^0))^2}{\Sigma^0(\mathbf{x}^0, \mathbf{x}^0)} \end{pmatrix}\right)$$

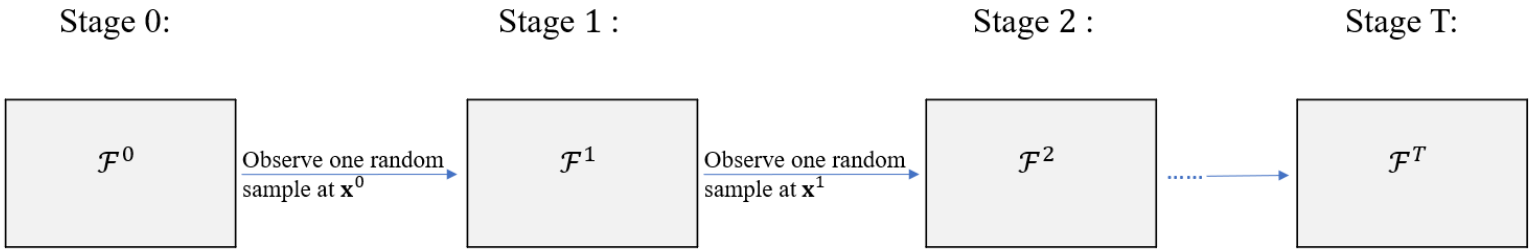


Gaussian Process

- A function $y(\mathbf{x})$, $\mathbf{x} \in \mathcal{X}$, to estimate.
- A prior on $y(\mathbf{x})$, $\mathbf{x} \in \mathcal{X}$, is a Gaussian process if the prior of any $(y(\mathbf{x}^0), y(\mathbf{x}^1), \ldots, y(\mathbf{x}^n))^\top$ has a multivariate Gaussian distribution.
  - Mean function: $\mu^0(\mathbf{x})$
  - Covariance function: $\Sigma^0(\mathbf{x}, \mathbf{x}') = Cov(\mu^0(\mathbf{x}), \mu^0(\mathbf{x}'))$.
  - The prior on $(y(\mathbf{x}^0), y(\mathbf{x}^1), \ldots, y(\mathbf{x}^n))^\top$ is multivariate Gaussian
    - mean: $\mu^0([\mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^n])$: $(\mu^0(\mathbf{x}^0), \mu^0(\mathbf{x}^1), \ldots, \mu^0(\mathbf{x}^n))^\top$
    - covariance matrix $\Sigma^0([\mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^n])$:

$$\begin{pmatrix} \Sigma^0(\mathbf{x}^0, \mathbf{x}^0) & \Sigma^0(\mathbf{x}^0, \mathbf{x}^1) & \ldots & \Sigma^0(\mathbf{x}^0, \mathbf{x}^n) \\ \Sigma^0(\mathbf{x}^1, \mathbf{x}^0) & \Sigma^0(\mathbf{x}^1, \mathbf{x}^1) & \ldots & \Sigma^0(\mathbf{x}^1, \mathbf{x}^n) \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma^0(\mathbf{x}^n, \mathbf{x}^0) & \Sigma^0(\mathbf{x}^n, \mathbf{x}^1) & \ldots & \Sigma^0(\mathbf{x}^n, \mathbf{x}^n) \end{pmatrix}$$

Gaussian Process Regression



Gaussian Process Regression

- For any $\mathbf{x} \in \mathcal{X}$, the prior distribution on $(y(\mathbf{x}^0), y(\mathbf{x}^1))^\top$ is $\mathcal{N}(\boldsymbol{\mu}^0([\mathbf{x}^0, \mathbf{x}^1]), \Sigma^0([\mathbf{x}^0, \mathbf{x}^1]))$.
- Given a sample $\hat{y}^1 = Y(\mathbf{x}^0)$ at $\mathbf{x}^0$. The posterior distribution of $(y(\mathbf{x}^0), y(\mathbf{x}^1)))^\top$ is $\mathcal{N}(\boldsymbol{\mu}^1([\mathbf{x}^0, \mathbf{x}^1]), \Sigma^1([\mathbf{x}^0, \mathbf{x}^1]))$ where

$$
\begin{aligned}
&\boldsymbol{\mu}^1([\mathbf{x}^0, \mathbf{x}^1]) \\
=&\boldsymbol{\mu}^0([\mathbf{x}^0, \mathbf{x}^1]) + \Sigma^0([\mathbf{x}^0, \mathbf{x}^1]) e_{\mathbf{x}^0} \left(\Sigma^0(\mathbf{x}^0, \mathbf{x}^0) + \lambda(\mathbf{x}^0)\right)^{-1} \left(\hat{y}^1 - \mu^0(\mathbf{x}^0)\right), \\
&\Sigma^1([\mathbf{x}^0, \mathbf{x}^1]) \\
=&\Sigma^0([\mathbf{x}^0, \mathbf{x}^1]) - \Sigma^0([\mathbf{x}^0, \mathbf{x}^1]) e_{\mathbf{x}^0} \left(\Sigma^0(\mathbf{x}^0, \mathbf{x}^0) + \lambda(\mathbf{x}^0)\right)^{-1} e_{\mathbf{x}^0}^\top \Sigma^0([\mathbf{x}^0, \mathbf{x}^1]).
\end{aligned}
$$

- Noiseless: $y(\mathbf{x}^1)|y(\mathbf{x}^0) \sim$
$\mathcal{N}\left(\boldsymbol{\mu}^0(\mathbf{x}^1) + \frac{\Sigma^0(\mathbf{x}^1, \mathbf{x}^0)}{\Sigma^0(\mathbf{x}^0, \mathbf{x}^0)} \left(y(\mathbf{x}^0) - \mu^0(\mathbf{x}^0)\right), \Sigma^0(\mathbf{x}^1, \mathbf{x}^1) - \frac{(\Sigma^0(\mathbf{x}^1, \mathbf{x}^0))^2}{\Sigma^0(\mathbf{x}^0, \mathbf{x}^0)}\right)$

Gaussian Process Regression

- More generally, after collecting $\hat{y}^{n+1} = Y(\mathbf{x}^n)$ (given $\mathcal{F}^{n+1}$), the posterior mean of $(y(\mathbf{x}^0), y(\mathbf{x}^1), \ldots, y(\mathbf{x}^n))^\top$ in the recursive form is

$$\begin{pmatrix} \boldsymbol{\mu}^{n+1}(\mathbf{x}^0) \\ \boldsymbol{\mu}^{n+1}(\mathbf{x}^1) \\ \vdots \\ \boldsymbol{\mu}^{n+1}(\mathbf{x}^n) \end{pmatrix} = \begin{pmatrix} \boldsymbol{\mu}^{n}(\mathbf{x}^0) \\ \boldsymbol{\mu}^{n}(\mathbf{x}^1) \\ \vdots \\ \boldsymbol{\mu}^{n}(\mathbf{x}^n) \end{pmatrix} + \Sigma^n([\mathbf{x}^0, \ldots, \mathbf{x}^n]) e_{\mathbf{x}^n} \left(\Sigma^n(\mathbf{x}^n, \mathbf{x}^n) + \lambda(\mathbf{x}^n)\right)^{-1} \left(\hat{y}^{n+1} - \mu^n(\mathbf{x}^n)\right)$$

$$= \begin{pmatrix} \boldsymbol{\mu}^{n}(\mathbf{x}^0) \\ \boldsymbol{\mu}^{n}(\mathbf{x}^1) \\ \vdots \\ \boldsymbol{\mu}^{n}(\mathbf{x}^n) \end{pmatrix} + \frac{\Sigma^n([\mathbf{x}^0, \ldots, \mathbf{x}^n]) e_{\mathbf{x}^n}}{\sqrt{\Sigma^n(\mathbf{x}^n, \mathbf{x}^n) + \lambda(\mathbf{x}^n)}} \frac{\hat{y}^{n+1} - \mu^n(\mathbf{x}^n)}{\sqrt{\Sigma^n(\mathbf{x}^n, \mathbf{x}^n) + \lambda(\mathbf{x}^n)}}$$

$$\triangleq \begin{pmatrix} \boldsymbol{\mu}^{n}(\mathbf{x}^0) \\ \boldsymbol{\mu}^{n}(\mathbf{x}^1) \\ \vdots \\ \boldsymbol{\mu}^{n}(\mathbf{x}^n) \end{pmatrix} + \tilde{\sigma}(\Sigma^n([\mathbf{x}^0, \ldots, \mathbf{x}^n]), \mathbf{x}^n) \frac{\hat{y}^{n+1} - \mu^n(\mathbf{x}^n)}{\sqrt{\Sigma^n(\mathbf{x}^n, \mathbf{x}^n) + \lambda(\mathbf{x}^n)}}$$

Gaussian Process Regression

- Before collecting $\hat{y}^{n+1} = Y(\mathbf{x}^n)$ and given $\mathcal{F}^n$,

$$\hat{y}^{n+1} = y(\mathbf{x}^n) + (Y(\mathbf{x}^n) - y(\mathbf{x}^n)) \sim \mathcal{N}(\boldsymbol{\mu}^n(\mathbf{x}^n), \Sigma^n(\mathbf{x}^n, \mathbf{x}^n) + \lambda(\mathbf{x}^n)).$$

- the predictive distribution of $(\boldsymbol{\mu}^{n+1}(\mathbf{x}^0), \boldsymbol{\mu}^{n+1}(\mathbf{x}^1), \ldots, \boldsymbol{\mu}^{n+1}(\mathbf{x}^n))^\top$

$$\begin{pmatrix} \boldsymbol{\mu}^{n+1}(\mathbf{x}^0) \\ \boldsymbol{\mu}^{n+1}(\mathbf{x}^1) \\ \vdots \\ \boldsymbol{\mu}^{n+1}(\mathbf{x}^n) \end{pmatrix} = \begin{pmatrix} \boldsymbol{\mu}^{n}(\mathbf{x}^0) \\ \boldsymbol{\mu}^{n}(\mathbf{x}^1) \\ \vdots \\ \boldsymbol{\mu}^{n}(\mathbf{x}^n) \end{pmatrix} + \frac{\Sigma^n([\mathbf{x}^0, \ldots, \mathbf{x}^n])e_{\mathbf{x}^n}}{\sqrt{\Sigma^n(\mathbf{x}^n, \mathbf{x}^n) + \lambda(\mathbf{x}^n)}} \frac{\hat{y}^{n+1} - \mu^n(\mathbf{x}^n)}{\sqrt{\Sigma^n(\mathbf{x}^n, \mathbf{x}^n) + \lambda(\mathbf{x}^n)}}$$

$$= \begin{pmatrix} \boldsymbol{\mu}^{n}(\mathbf{x}^0) \\ \boldsymbol{\mu}^{n}(\mathbf{x}^1) \\ \vdots \\ \boldsymbol{\mu}^{n}(\mathbf{x}^n) \end{pmatrix} + \tilde{\sigma}(\Sigma^n([\mathbf{x}^0, \ldots, \mathbf{x}^n]), \mathbf{x}^n)Z^{(n+1)} \text{ (Note: } Z^{(n+1)} \sim \mathcal{N}(0, 1).)$$

Gaussian Process Regression

- After collecting $\hat{y}^{n+1} = Y(\mathbf{x}^n)$ (given $\mathcal{F}^{n+1}$), the posterior distribution of $y(\mathbf{x})$ in the direct form is $\mathcal{N}(\mu^{n+1}(\mathbf{x}), \Sigma^{n+1}(\mathbf{x}))$ where (hint: $y(\mathbf{x})|\hat{y}^1, \hat{y}^2, \ldots, \hat{y}^{n+1}$)

$$\mu^{n+1}(\mathbf{x}) = \boldsymbol{\mu}^0(\mathbf{x}) + \left(\Sigma^0(\mathbf{x}^0, \mathbf{x}), \ldots, \Sigma^0(\mathbf{x}^n, \mathbf{x})\right) (S^n)^{-1} \begin{pmatrix} \hat{y}^1 - \mu^0(\mathbf{x}^0) \\ \vdots \\ \hat{y}^{n+1} - \mu^0(\mathbf{x}^n) \end{pmatrix},$$

$$\Sigma^{n+1}(\mathbf{x}) = \Sigma^0(\mathbf{x}, \mathbf{x}) - \left(\Sigma^0(\mathbf{x}^0, \mathbf{x}), \ldots, \Sigma^0(\mathbf{x}^n, \mathbf{x})\right) (S^n)^{-1} \begin{pmatrix} \Sigma^0(\mathbf{x}^0, \mathbf{x}) \\ \vdots \\ \Sigma^0(\mathbf{x}^n, \mathbf{x}) \end{pmatrix}$$

  and $S^n = \Sigma^0([\mathbf{x}^0, \ldots, \mathbf{x}^n]) + diag([\lambda(\mathbf{x}^0), \ldots, \lambda(\mathbf{x}^n)])$.
- Question 3: what if the observation has no noise?

Answer to Questions for Noiseless Sampling

- Question 3: Note that given $\lambda(\mathbf{x}^i) = 0$, $i = 0, 1, \ldots, n$, $S^n = \Sigma^0([\mathbf{x}^0, \ldots, \mathbf{x}^n])$. Then,

$$(S^n)^{-1} \begin{pmatrix} \Sigma^0(\mathbf{x}^0, \mathbf{x}_i) \\ \vdots \\ \Sigma^0(\mathbf{x}^n, \mathbf{x}_i) \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \triangleq e_i$$

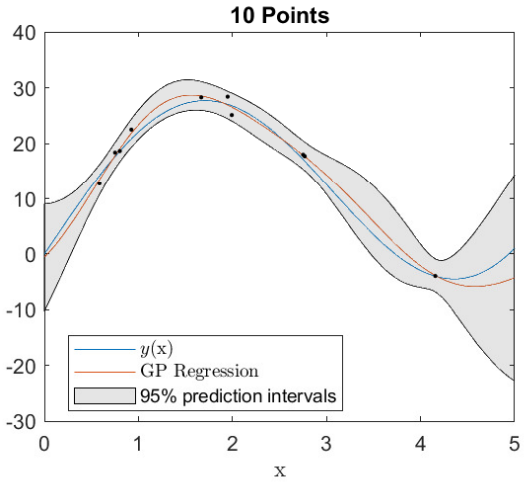
- So, $y(\mathbf{x}^i) \sim \mathcal{N}(\mu^{n+1}(\mathbf{x}^i), \Sigma^{n+1}(\mathbf{x}^i))$ where $\mu^{n+1}(\mathbf{x}^i) = y(\mathbf{x}^i)$ and $\Sigma^{n+1}(\mathbf{x}^i) = 0$, $i = 0, 1, \ldots, n$.

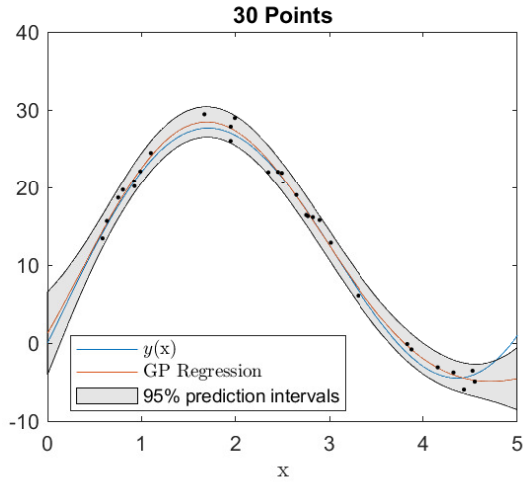
Notations Check Table

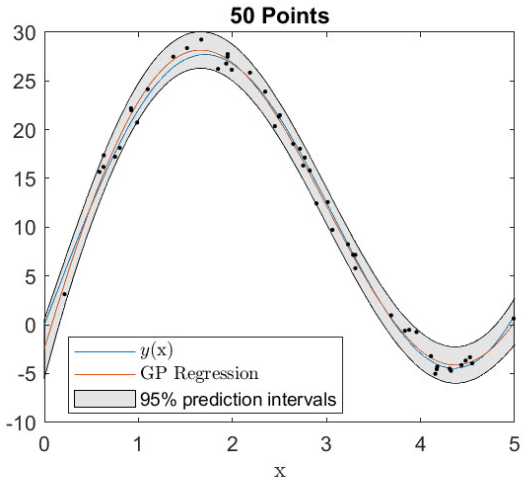
This document	KG	EGO
$y(\mathbf{x})$	$\mu(\mathbf{x})$	$y(\mathbf{x})$
$\min_{\mathbf{x}\in\mathcal{X}} y(\mathbf{x})$	$\max_{\mathbf{x}\in\mathcal{X}} \mu(\mathbf{x})$	$\min_{\mathbf{x}\in\mathcal{X}} y(\mathbf{x})$
$\Sigma^0(\mathbf{x},\mathbf{x}')$	$\Sigma^0(\mathbf{x},\mathbf{x}')$	$\sigma^2\mathrm{Corr}[\epsilon(\mathbf{x}),\epsilon(\mathbf{x}')]$
β	β	σ^2
$\mathcal{S}^{n-1}$	$\mathcal{S}^{n-1}$	$\sigma^2\mathbf{R}$
$\mu^0(\mathbf{x})$	$\mu^0(\mathbf{x})$	$\sum_h \beta_h f_h(\mathbf{x})$ and μ
$\mu^n(\mathbf{x}^*)$	$\mu^n(\mathbf{x}^*)$	$\hat{y}(\mathbf{x}^*)$
$\hat{y}^i$	$\hat{y}^i$	$y(\mathbf{x}^{(i+1)})$
$\mathbf{x}^0,\mathbf{x}^1,\ldots,\mathbf{x}^{n-1}$	$\mathbf{x}^0,\mathbf{x}^1,\ldots,\mathbf{x}^{n-1}$	$\mathbf{x}^{(1)},\mathbf{x}^{(2)}\ldots,\mathbf{x}^{(n)}$

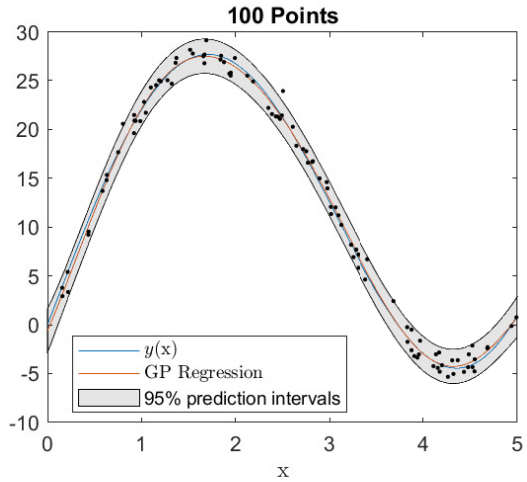
Outline

Example 1

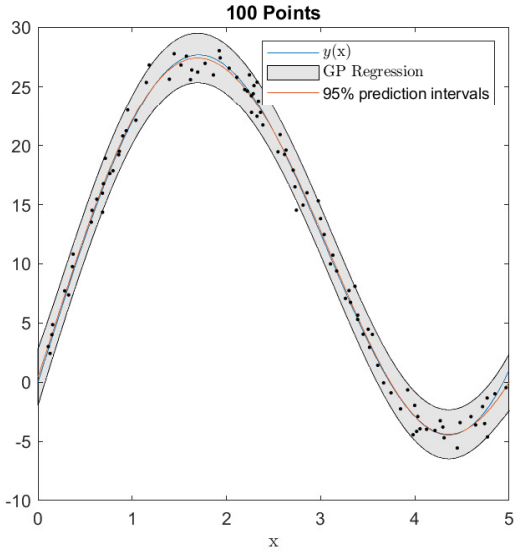

10 Points
y(x)
GP Regression
95% prediction intervals
x

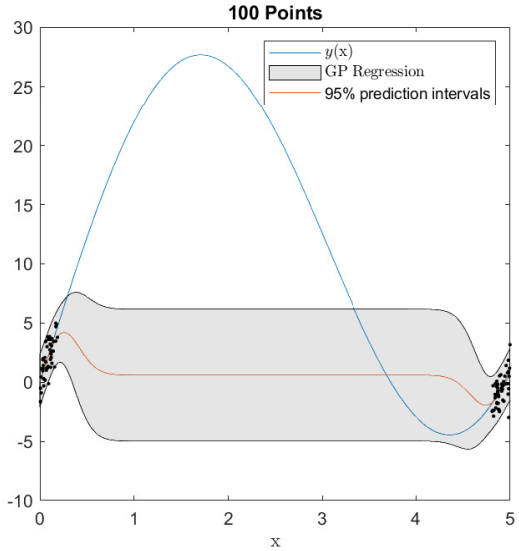

30 Points
y(x)
GP Regression
95% prediction intervals
x


50 Points
y(x)
GP Regression
95% prediction intervals
x


100 Points
y(x)
GP Regression
95% prediction intervals
x

Example 2


100 Points
y(x)
GP Regression
95% prediction intervals
x


100 Points
y(x)
GP Regression
95% prediction intervals
x

Outline

Section Contents

Contents:

1. For simulation optimization with noiseless samples, Efficient Global Optimization algorithm (Expected Improvement algorithm).
2. For simulation optimization with noisy samples, Knowledge Gradient algorithm.

Outline

Basic Settings

- Prior on $y(\mathbf{x})$ is a Gaussian process.
- Mean function of the prior: $\boldsymbol{\mu}^0(\mathbf{x}) = \mu$ for all $\mathbf{x} \in \mathcal{X}$.
- Covariance function of the prior: depends on the distance:
  - $d(\mathbf{x}, \mathbf{x}') = \sum_{h=1}^{k} \theta_h |x_h - x'_h|^{p_h}$ $(\theta_h \geq 0,\ p_h \in [1, 2])$
  - $\Sigma^0(\mathbf{x}, \mathbf{x}') = \beta \exp[-d(\mathbf{x}, \mathbf{x}')]$.
- Noiseless samples: $(\hat{y}^1, \hat{y}^2, \ldots, \hat{y}^n) = (y(\mathbf{x}^0), y(\mathbf{x}^1), \ldots, y(\mathbf{x}^{n-1}))^\top$.

Gaussian Process Regression

Predict at $\mathbf{x}^* \in \mathcal{X}$:

- $(\mathbf{y}^\top, y(\mathbf{x}^*))^\top$ has the distribution

$$\mathcal{N}\left(\mathbf{1}_{n+1}\mu, \begin{pmatrix} \Sigma^0([\mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^{n-1}]) & \Sigma^0([\mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^{n-1}], \mathbf{x}^*) \\ \Sigma^0([\mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^{n-1}], \mathbf{x}^*)^\top & \Sigma^0(\mathbf{x}^*, \mathbf{x}^*) \end{pmatrix}\right),$$

  where $\Sigma^0([\mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^{n-1}], \mathbf{x}^*) = \left(\Sigma^0(\mathbf{x}^0, \mathbf{x}^*), \ldots, \Sigma^0(\mathbf{x}^{n-1}, \mathbf{x}^*)\right)^\top$.

- $\mu^n(\mathbf{x}^*) =$

$$\mu + \left(\Sigma^0(\mathbf{x}^0, \mathbf{x}^*), \ldots, \Sigma^0(\mathbf{x}^{n-1}, \mathbf{x}^*)\right) \left(S^{n-1}\right)^{-1} \begin{pmatrix} y(\mathbf{x}^0) - \mu \\ \vdots \\ y(\mathbf{x}^{n-1}) - \mu \end{pmatrix}$$

  (also the maximum point of the augmented likelihood function, see Appendix 1 of the paper)

Parameter Estimation

- Parameter estimation (from the frequentist perspective):

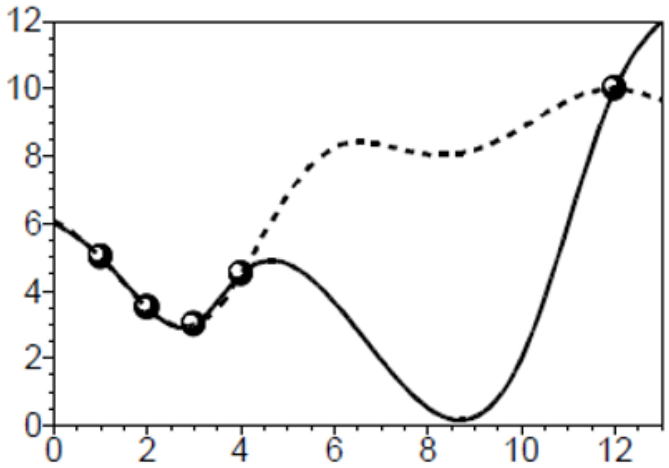
$$\frac{\exp\left[-\frac{1}{2}\left(\begin{pmatrix} y(\mathbf{x}^0) \\ \vdots \\ y(\mathbf{x}^{n-1}) \end{pmatrix} - \mathbf{1}_n\mu\right)' (S^{n-1})^{-1} \left(\begin{pmatrix} y(\mathbf{x}^0) \\ \vdots \\ y(\mathbf{x}^{n-1}) \end{pmatrix} - \mathbf{1}_n\mu\right)\right]}{(2\pi)^{n/2}|S^{n-1}|^{\frac{1}{2}}}$$

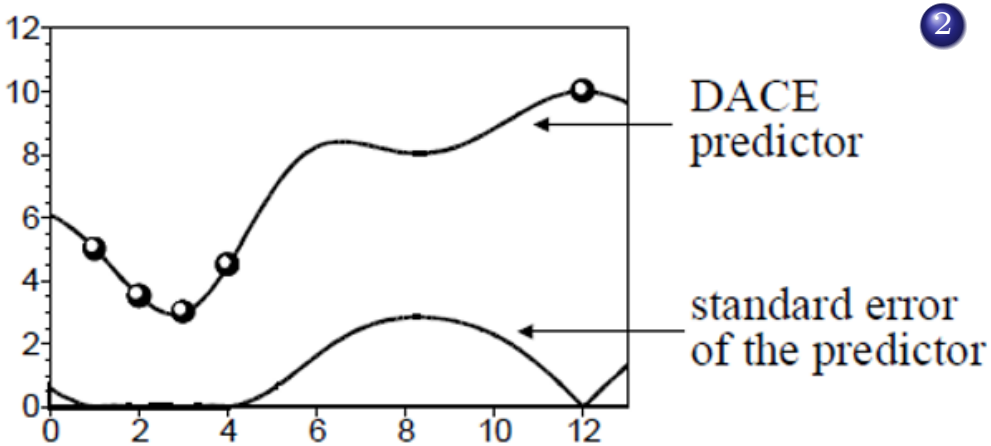
$$\hat{\mu} = \frac{\mathbf{1}_n'(S^{n-1})^{-1}\mathbf{y}}{\mathbf{1}_n'(S^{n-1})^{-1}\mathbf{1}_n}.$$

- Mean squared error of $\mu^n(\mathbf{x}^*)$ (underestimate if σ^2, θ_h, and p_h are estimated by MLE):

$$s^{2,n}(\mathbf{x}^*) = \Sigma^n(\mathbf{x}^*, \mathbf{x}^*) + \frac{\left(1 - \mathbf{1}'(S^{n-1})^{-1}\Sigma^0([\mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^{n-1}], \mathbf{x}^*)\right)^2}{\mathbf{1}'(S^{n-1})^{-1}\mathbf{1}}.$$

Motivating Example

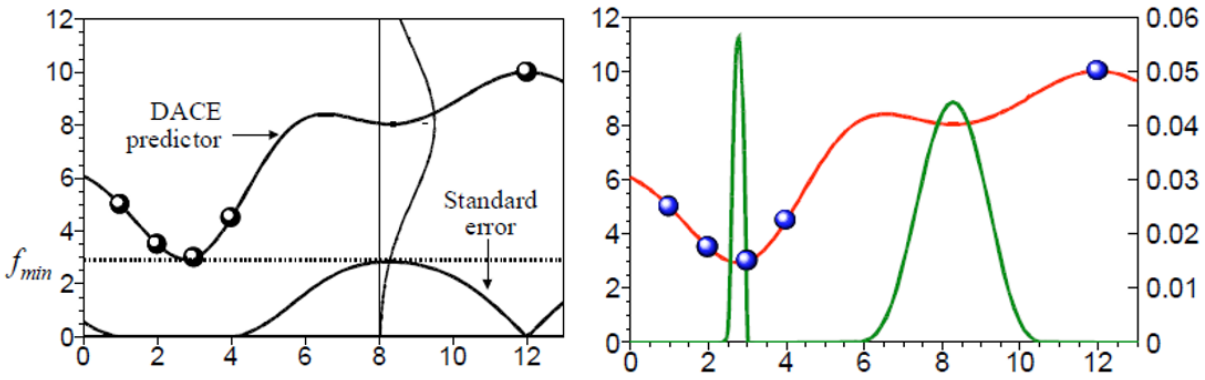




Objective: $\min_{\mathbf{x}} y(\mathbf{x})$

1. Exploitation: sampling the points around the local minimum
   - more accurate estimate of the local minimum.
2. Exploration: sampling points with high uncertainty.
   - discovering potential local minima.

Expected Improvement



- Current best: $y_{\min} = \min(y(\mathbf{x}^0), \ldots, y(\mathbf{x}^{n-1}))$.
- Best in next iteration if $\mathbf{x}^{\mathrm{EI},n+1} = \mathbf{x}$: $\min(y_{\min}, y(\mathbf{x}))$.
- $y(\mathbf{x})$ has posterior distribution $\mathcal{N}(\mu^n(\mathbf{x}), s^{2,n}(\mathbf{x}))$.
- $\xi(\mathbf{x}) \sim \mathcal{N}(\mu^n(\mathbf{x}), s^{2,n}(\mathbf{x}))$. Expected improvement:

$$\begin{aligned}\mathrm{E}[I(\mathbf{x})|\mathcal{F}^n] =& y_{\min} - \mathrm{E}[\min(\xi(\mathbf{x}), y_{\min})] = \mathrm{E}[\max(y_{\min} - \xi(\mathbf{x}), 0)] \\ =& (y_{\min} - \mu^n(\mathbf{x}))\, \Phi\left(\frac{y_{\min} - \mu^n(\mathbf{x})}{s^n(\mathbf{x})}\right) + s^n(\mathbf{x})\phi\left(\frac{y_{\min} - \mu^n(\mathbf{x})}{s^n(\mathbf{x})}\right).\end{aligned}$$

Efficient Global Optimization (EGO) Algorithm

Initialization:

Specify a set of space-filling initial points.

Evaluate the function on the initial design points.

Fit the Gaussian process model (DACE).

Model validation.

repeat

Maximize $\mathbf{x}^{\mathrm{EI},n+1} = \arg\max_{\mathbf{x}} \mathrm{E}[I(\mathbf{x})|\mathcal{F}^n]$ by the branch-and-bound algorithm.

Evaluate $y(\mathbf{x}^{\mathrm{EI},n+1})$.

Update the Gaussian process model.

$n \leftarrow n+1$.

until $\max_{\mathbf{x}} \mathrm{E}[I(\mathbf{x})|\mathcal{F}^n] \leq \alpha$.

Efficient Global Optimization (EGO) Algorithm

Table 1. Test function results for the EGO algorithm.

Test problem	Evaluations to meet stopping criterion	Actual error when stopped	Evaluations required for 1% accuracy
Branin	28	0.2%	28
Goldstein–Price	32	0.1%	32
Hartman 3	34	1.7%	35
Hartman 6	84	1.9%	121

Note: The stopping rule does not guarantee that the actual error is less than α.

Convergence of EGO

- Bull (2011): When priors are typically estimated sequentially from the data, EGO may never find the minimum.
- Modified EGO (using ϵ-greedy) can achieve the near-optimal convergence rate.

Outline

Knowledge Gradient Algorithm

Initialization:

Specify the prior information and its parameters.

repeat

Maximize $\mathbf{x}^{\mathrm{KG},n} = \arg\max_{\mathbf{x}} \bar{\nu}^{\mathrm{KG},n}(\mathbf{x})$.

Evaluate noisy $Y(\mathbf{x}^{\mathrm{KG},n})$.

Update the Gaussian process model.

$n \leftarrow n + 1$.

until $n = N$.

Knowledge Gradient Definition

- Ideal Definition:
  - Current best: $\min_{u\in\mathcal{X}} \mu^n(u)$.
  - Best in the next iteration if $\mathbf{x}^n = \mathbf{x}$: $\min_{u\in\mathcal{X}} \mu^{n+1}(u)\big|_{\mathbf{x}^n=\mathbf{x}}$.
  - $$\nu^{\mathrm{KG},n}(\mathbf{x}) \triangleq \min_{u\in\mathcal{X}} \mu^n(u) - \mathrm{E}\left[\min_{u\in\mathcal{X}} \mu^{n+1}(u) \mid \mathcal{F}^n, \mathbf{x}^n = \mathbf{x}\right].$$
- Approximation:
  - Current best if $\mathbf{x}^n = \mathbf{x}$: $\min_{i=0,..,n} \mu^n\left(\mathbf{x}^i\right)\big|_{\mathbf{x}^n=\mathbf{x}}$.
  - Best in the next iteration if $\mathbf{x}^n = \mathbf{x}$: $\min_{i=0,..,n} \mu^{n+1}\left(\mathbf{x}^i\right)\big|_{\mathbf{x}^n=\mathbf{x}}$.
  - $$\bar{\nu}^{\mathrm{KG},n}(\mathbf{x}) \triangleq \min_{i=0,..,n} \mu^n\left(\mathbf{x}^i\right)\Big|_{\mathbf{x}^n=\mathbf{x}} - \mathrm{E}\left[\min_{i=0,\ldots,n} \mu^{n+1}\left(\mathbf{x}^i\right) \mid \mathcal{F}^n, \mathbf{x}^n = \mathbf{x}\right]$$

Knowledge Gradient Properties

- Non-negative:

$$\begin{aligned}
&\mathbb{E}\left[\min_{i=0,\ldots,n} \mu^{n+1}\left(\mathbf{x}^i\right) \mid \mathcal{F}^n, \mathbf{x}^n = \mathbf{x}\right] \\
=&\mathbb{E}\left[\min_{i=0,\ldots,n} \mu^{n}\left(\mathbf{x}^i\right) + \frac{e_{\mathbf{x}^i}^\top \Sigma^{(n)}\left(\mathbf{x}^1,\ldots,\mathbf{x}^n\right) e_{\mathbf{x}^n}}{\sqrt{\Sigma^{(n)}\left(\mathbf{x}^n,\mathbf{x}^n\right)+\sigma^2\left(\mathbf{x}^n\right)}} Z^{n+1} \mid \mathcal{F}^n, \mathbf{x}^n = \mathbf{x}\right] \\
\leq& \min_{i=0,\ldots,n} \mu^{n}\left(\mathbf{x}^i\right).
\end{aligned}$$

- $\bar{\nu}^{\mathrm{KG},0}(\mathbf{x}) = 0$. Indifferent about the first sampling.

Knowledge Gradient Properties

Proposition

In the case of no observation noise, $\bar{\nu}^{KG,n}(\mathbf{x}) \leq \mathrm{E}[I(\mathbf{x})|\mathcal{F}^n]$. *Further,* $\mathrm{E}[I(\mathbf{x})|\mathcal{F}^n] = \min_{i=0,..,n-1} \mu^n\left(\mathbf{x}^i\right) - \mathrm{E}\left[\min_{i=0,..,n} \mu^{n+1}\left(\mathbf{x}^i\right) \mid \mathcal{F}^n, \mathbf{x}^n = \mathbf{x}\right]$.

Proof.

$$
\begin{aligned}
&\bar{\nu}^{\mathrm{KG},n}(\mathbf{x}) \triangleq \min_{i=0,..,n} \mu^n\left(\mathbf{x}^i\right) - \mathrm{E}\left[\min_{i=0,\ldots,n} \mu^{n+1}\left(\mathbf{x}^i\right) \mid \mathcal{F}^n, \mathbf{x}^n = \mathbf{x}\right] \\
&= \min(\min_{i=0,\ldots,n-1} y(\mathbf{x}^i), \mu^n\left(\mathbf{x}^n\right)) - \mathrm{E}\left[\min\left(\mu^{n+1}\left(\mathbf{x}^n\right), \min_{i=0,\ldots,n-1} y(\mathbf{x}^i)\right) \mid \mathcal{F}^n, \mathbf{x}^n = \mathbf{x}\right] \\
&\leq \min_{i=0,\ldots,n-1} y(\mathbf{x}^i) - \mathrm{E}\left[\min\left(\mu^{n+1}\left(\mathbf{x}^n\right), \min_{i=0,\ldots,n-1} y(\mathbf{x}^i)\right) \mid \mathcal{F}^n, \mathbf{x}^n = \mathbf{x}\right] \\
&= \mathrm{E}\left[I^n(\mathbf{x}) \mid \mathcal{F}^n\right].
\end{aligned}
$$

(Note: With noiseless samples, the predictive distribution of μ^{n+1} = the prior of y.) □

Consistency of Knowledge Gradient

Theorem

Under the KGCP policy, if Assumptions 5.0.1, 5.0.2, 5.0.3, and 5.0.4 are satisfied, then $\lim_{n\to\infty} \Sigma^n(\mathbf{x},\mathbf{x}) = 0$ *for all* $\mathbf{x}$.

Assumptions:

5.0.1 $\lambda(\mathbf{x})$ and $\mu^0(\mathbf{x})$ are constant and fixed, and the parameters in covariance function (α, β)are fixed.

5.0.2 $\limsup_{n\to\infty} |\mu^n(\mathbf{x}) - \mu^n(\mathbf{x}')|$ is bounded for every $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ a.s..

5.0.3 $\limsup_{n\to\infty} |\Sigma^n(\mathbf{x},\mathbf{x}')/\lambda| \le c < 1$ a.s..

5.0.4 We can exactly maximize the KGCP; $\mathbf{x}^n = \arg\max_{\mathbf{x}\in\mathcal{X}} \bar{\nu}^{\mathrm{KG},n}(\mathbf{x})$

Framework of the Proof

Theorems, propositions and corollaries

P5.1 Upper bound of $\bar{\nu}^{\mathrm{KG},n}(\mathbf{x})$: $\bar{\nu}^{\mathrm{KG},n}(\mathbf{x}) \leq \sqrt{\frac{2\beta\Sigma^n[\mathbf{x},\mathbf{x}]}{\pi\lambda}}$.

P5.2 Upper bound of $\Sigma^n[\mathbf{x},\mathbf{x}]$, where $\mathbf{x} \in B(\mathbf{x}^{acc},\epsilon)$.

P5.3 Upper bound of $\lim_{n\to\infty}\Sigma^n[\mathbf{x},\mathbf{x}]$, where $\mathbf{x} \in B(\mathbf{x}^{acc},\epsilon)$.

C5.4 $\lim_{n\to\infty}\Sigma^n[\mathbf{x}^{acc},\mathbf{x}^{acc}] = 0$.

T5.5 $\liminf_{n\to\infty}\sup_{\mathbf{x}\in\mathcal{X}}\bar{\nu}^{\mathrm{KG},n}(\mathbf{x}) = 0$.

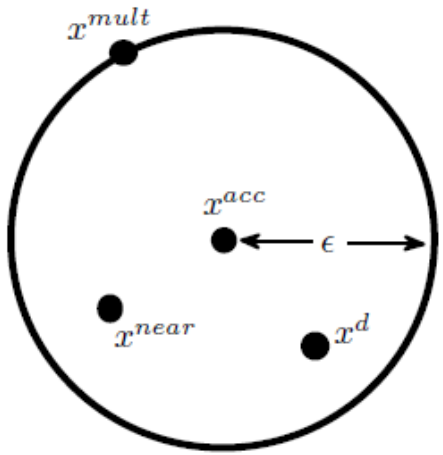
T5.6 If $\liminf_{n\to\infty}\sup_{\mathbf{x}\in\mathcal{X}}\bar{\nu}^{\mathrm{KG},n}(\mathbf{x}) = 0$, then $\lim_{n\to\infty}\Sigma^n(\mathbf{x},\mathbf{x}) = 0$ for all $\mathbf{x}\in\mathcal{X}$.

C5.7 $\lim_{n\to\infty}\Sigma^n(\mathbf{x},\mathbf{x}) = 0$ for all $\mathbf{x}\in\mathcal{X}$.

Framework of the Proof

Proposition 5.2: Upper bound of $\Sigma^n[\mathbf{x}, \mathbf{x}]$, where $\mathbf{x} \in B(\mathbf{x}^{acc}, \epsilon)$.

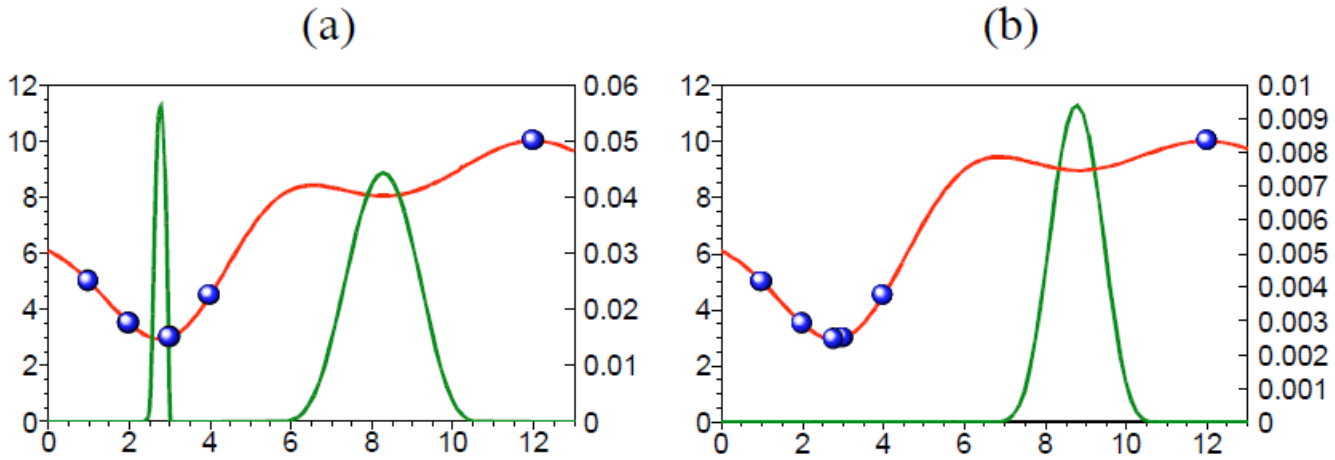
- Accumulation point $\mathbf{x}^{acc}$ of $\mathbf{x}^n$: for every ϵ, there are infinitely many natural numbers n such that $\mathbf{x}^n \in B(\mathbf{x}^{acc}, \epsilon)$.



- $\mathbf{x}^{mult}$ is farther away from $\mathbf{x}$ than any $\mathbf{x}^{near} \in B(\mathbf{x}^{acc}, \epsilon)$.
- $\Sigma^n[\mathbf{x}, \mathbf{x}]$ will be the largest if we always sample $\mathbf{x}^{mult}$ in the first iterations.

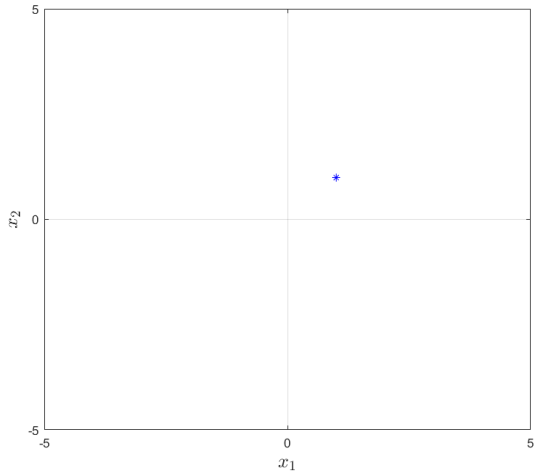
Outline

Expected Improvement Calculation



- The expected improvement function is highly multi-modal.
- $\mathrm{E}[I(x)|\mathcal{F}^n] = 0$ if $y(\mathbf{x})$ has been evaluated.
- $\frac{\partial \mathrm{E}[I(x)|\mathcal{F}^n]}{\partial \mu^n(\mathbf{x})} = -\Phi\left(\frac{y_{\min} - \mu^n(\mathbf{x})}{s^n(\mathbf{x})}\right) < 0.$
- $\frac{\partial \mathrm{E}[I(x)|\mathcal{F}^n]}{\partial s^n(\mathbf{x})} = \phi\left(\frac{y_{\min} - \mu^n(\mathbf{x})}{s^n(\mathbf{x})}\right) > 0.$

Maximizing Expected Improvement



Proposed method: branch-and-bound algorithm.

- Upper bound on $\mathrm{E}[I(\mathbf{x})|\mathcal{F}^n]$ over a sub-region $l_h \leq x_h \leq u_h$, $h = 1, \ldots, d$: a lower bound on $\mu^n(\mathbf{x})$ and an upper bound on $s(\mathbf{x})$.
  - Add an "α term" to make the objective convex.
  - Replacing the nonlinear term with linear under-estimators.

Knowledge Gradient Calculation

By definition,

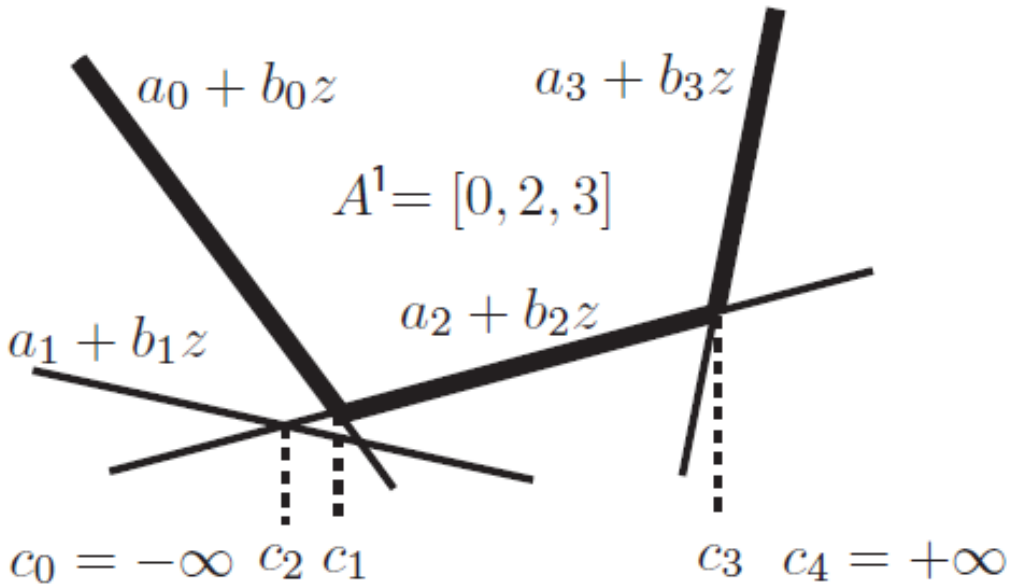
$$\bar{\nu}^{\mathrm{KG},n}(\mathbf{x}) \triangleq \min_{i=0,..,n} \mu^n\left(\mathbf{x}^i\right)\Big|_{\mathbf{x}^n=\mathbf{x}} - \mathrm{E}\left[\min_{i=0,\ldots,n} \mu^{n+1}\left(\mathbf{x}^i\right) \mid \mathcal{F}^n, \mathbf{x}^n = \mathbf{x}\right],$$

where $\min_{i=0,..,n} \mu^n\left(\mathbf{x}^i\right)$ are known.

$$\begin{aligned}
&- \mathrm{E}\left[\min_{i=0,\ldots,n} \mu^{n+1}\left(\mathbf{x}^i\right) \mid \mathcal{F}^n, \mathbf{x}^n = \mathbf{x}\right] \\
=& \mathrm{E}\left[\max_{i=0,\ldots,n} -\mu^{n+1}\left(\mathbf{x}^i\right) \mid \mathcal{F}^n, \mathbf{x}^n = \mathbf{x}\right] \\
=& \mathrm{E}\left[\max_{i=0,\ldots,n} -\mu^{n}\left(\mathbf{x}^i\right) - \tilde{\sigma}_i(\Sigma^n, \mathbf{x}^n) Z^{(n+1)}\right],
\end{aligned}$$

where $Z^{(n+1)} \sim \mathcal{N}(0,1)$.

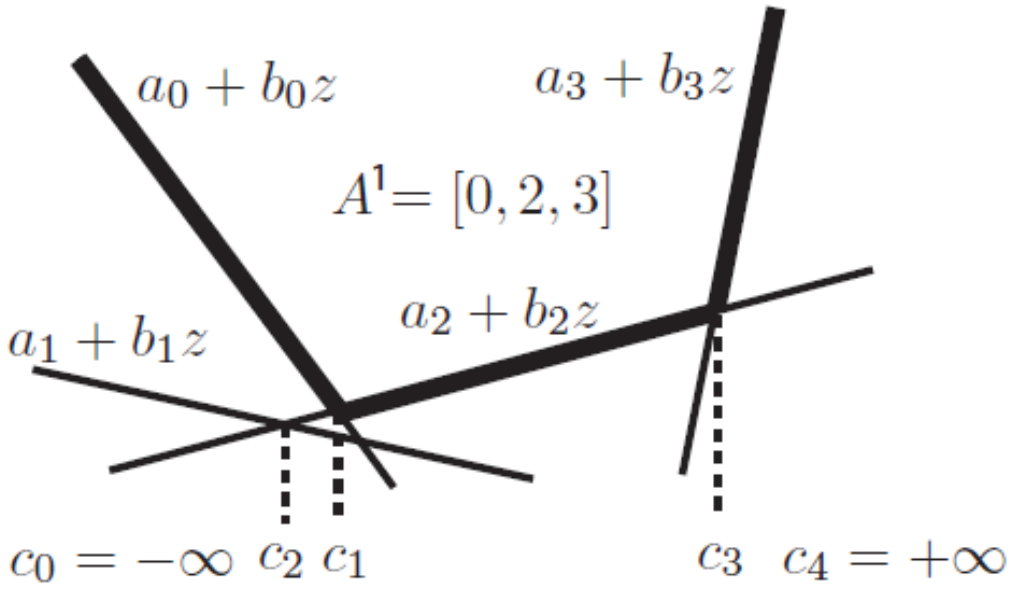
Knowledge Gradient Calculation



- Sort $\left(-\mu^n\left(\mathbf{x}^i\right), -\tilde{\sigma}_i(\Sigma^n, \mathbf{x}^n)\right)$, $i = 0, 1, \ldots, n$, as (a_i, b_i), $i = 0, 1, \ldots, n$ such that $b_i \leq b_{i+1}$.

$$-\mathrm{E}\left[\min_{i=0,\ldots,n} \mu^{n+1}\left(\mathbf{x}^i\right) \mid \mathcal{F}^n, \mathbf{x}^n = \mathbf{x}\right] = \mathrm{E}\left[\max_{i=0,\ldots,n} a_i + b_i Z^{(n+1)}\right],$$

Knowledge Gradient Calculation



- Obtain a point set A^1 such that A^1_i corresponds to the i-th part of the epigraph.
- An intersection point set $\tilde{c}_{i+1}$: $a_{A^1_i} + b_{A^1_i} z$ intersects with $a_{A^1_{i+1}} + b_{A^1_{i+1}} z$ at $\tilde{c}_{i+1}$.

Knowledge Gradient Calculation

$$
\begin{aligned}
&\mathrm{E}\left[\max_{i=0,\ldots,n} -\mu^{n+1}\left(\mathbf{x}^i\right) \mid \mathcal{F}^n, \mathbf{x}^n = \mathbf{x}\right] \\
=&\mathrm{E}\left[\max_{i=0,\ldots,n} -\mu^{n}\left(\mathbf{x}^i\right) - \tilde{\sigma}_i(\Sigma^n, \mathbf{x}^n) Z^{(n+1)} \mid \mathcal{F}^n, \mathbf{x}^n = \mathbf{x}\right] \\
=&\mathrm{E}\left[\sum_{i=1}^{\tilde{n}} \left(a_{A_i^1} + b_{A_i^1} Z\right) \mathbf{1}_{[c_i, c_{i+1})}(Z)\right] \\
=&\mathrm{E}\left[\sum_{i=1}^{\tilde{n}} a_{A_i^1} \left(\Phi(c_i) - \Phi(c_{i+1})\right) + b_{A_i^1} \left(\phi(c_i) - \phi(c_{i+1})\right)\right]
\end{aligned}
$$

where $Z^{(n+1)} \sim \mathcal{N}(0,1)$.

Gradient of Knowledge Gradient

- $\arg\max_{\mathbf{x}\in\mathcal{X}} \bar{\nu}^{\mathrm{KG},n}(\mathbf{x})$ can use gradient ascent algorithm with multi-start.
  - Product rule: $\frac{\partial f(x)g(x)}{\partial x} = g(x)\frac{\partial f(x)}{\partial x} + f(x)\frac{\partial g(x)}{\partial x}$.
  - Quotient rule: $\frac{\partial f(x)/g(x)}{\partial x} = \frac{g(x)\partial f(x) - f(x)\partial g(x)}{(g(x))^2}$.
  - $\mu^{(n)}(\mathbf{x}^i) = Y(\mathbf{x}^i)$, $i = 0, 1, \ldots, n-1$. Thus, $\frac{\partial \mu^{(n)}(\mathbf{x}^i)}{\partial \mathbf{x}^n} = 0$.
- "It may be acceptable if on one iteration the algorithm chooses a point which does not exactly maximize the knowledge gradient for continuous parameters."

Outline

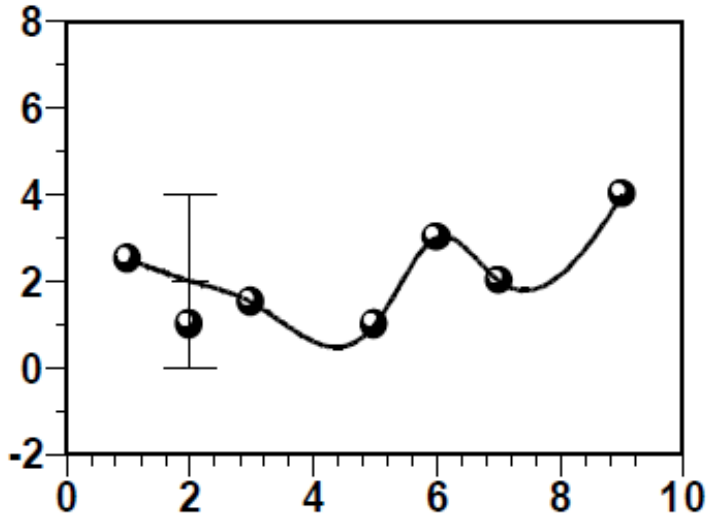
Outline

Cross Validation

- Basic idea: leave one sample $y(\mathbf{x}^i)$ out, predict $y(\mathbf{x}^i)$ based on remaining points n points.
- If the Gaussian process model is appropriate, $y(\mathbf{x}^i)$ and $\hat{y}_{-i}(\mathbf{x}^i)$ should be close:

$$\frac{\hat{y}_{-i}(\mathbf{x}^i) - y(\mathbf{x}^i)}{s^2_{-i}(\mathbf{x}^i)}$$

should be roughly in $[-3, 3]$. (standardized cross-validated residual)



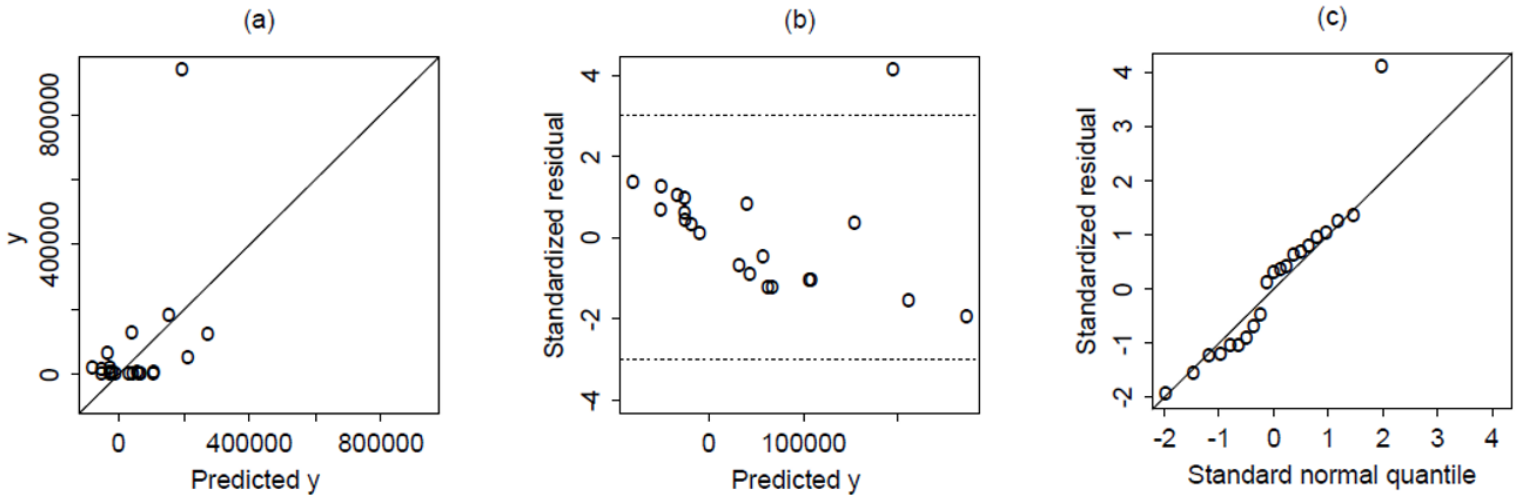
Cross Validation

Diagnostic plots

(a) $y(\mathbf{x}^i)$ v.s. $\hat{y}_{-i}(\mathbf{x}^i)$

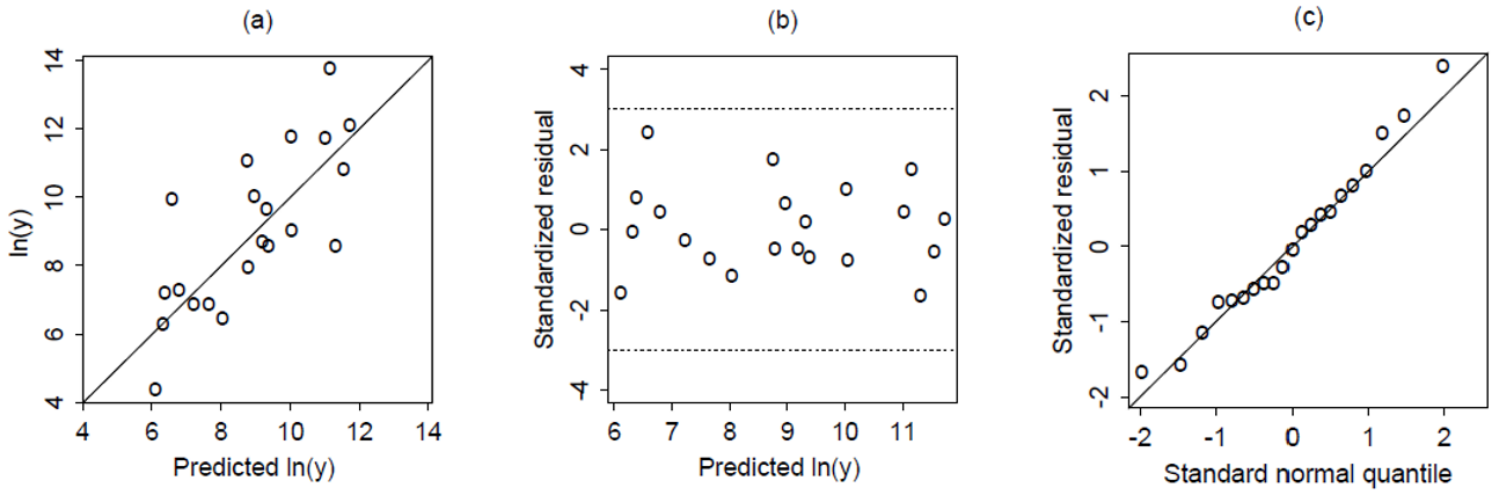
(b) standardized cross-validated residual v.s. $\hat{y}_{-i}(\mathbf{x}^i)$

(c) Q-Q plot: standardized cross-validated residual v.s. random normal variables



Improving Model Fitness

Transform the function: $\log(y(\mathbf{x}))$ or $-1/y(\mathbf{x})$.



Outline

Identifying Important Factors

Example: $y(\mathbf{x}) = y(x_1, x_2)$, $x_1, x_2 \in [0, 1]$. Gaussian process regression $\hat{y}(x_1, x_2)$.

- Overall average: $a_0 = \int_0^1 \int_0^1 \hat{y}(x_1, x_2) dx_1 dx_2$.
- Average effect of x_1: $a_1(x_1) = \int_0^1 \hat{y}(x_1, x_2) dx_2$.
- Average effect of x_2: $a_1(x_2) = \int_0^1 \hat{y}(x_1, x_2) dx_1$.
- Decomposing $\hat{y}(x_1, x_2)$:

$$\begin{aligned}\hat{y}(x_1, x_2) - a_0 =& (a_1(x_1) - a_0) + (a_1(x_2) - a_0) \\ &+ [\hat{y}(x_1, x_2) - a_0 - (a_1(x_1) - a_0) - (a_1(x_2) - a_0)].\end{aligned}$$

Identifying Important Factors

- Decomposing total variance:

$$\int_0^1 \int_0^1 (\hat{y}(x_1, x_2) - a_0)^2 dx_1 dx_2$$

$$= \int_0^1 (a_1(x_1) - a_0)^2 dx_1 + \int_0^1 (a_1(x_2) - a_0)^2 dx_2$$

$$+ \int_0^1 \int_0^1 [\hat{y}(x_1, x_2) - a_0 - (a_1(x_1) - a_0) - (a_1(x_2) - a_0)]^2 dx_1 dx_2.$$

Total Variance

$$=\text{Variance explained by } x_1 + \text{Variance explained by } x_2$$

$$+ \text{Variance explained by the interaction of } x_1 \text{ and } x_2.$$

Identifying Important Factors

- More generally, for $y(\mathbf{x}) = y(x_1, \ldots, x_d)$, $x_1, \ldots, x_d \in [0, 1]$,

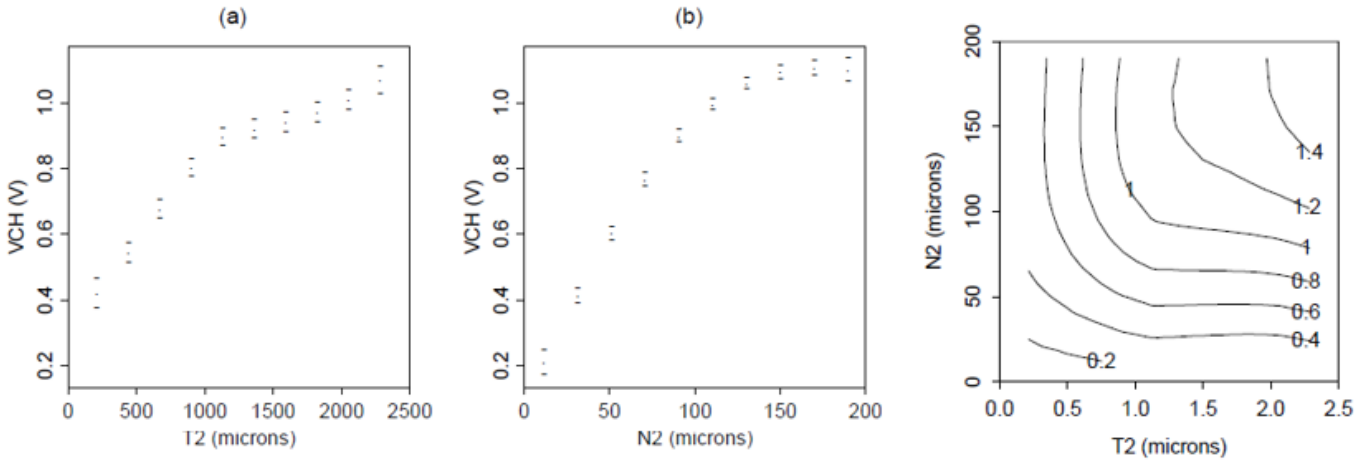
$$
\begin{aligned}
&\text{Total Variance} \\
&= \sum_{h=1}^{d} \text{Variance explained by } x_h \\
&\quad + \sum_{h_1, h_2} \text{Variance explained by the interaction of } x_{h_1} \text{ and } x_{h_2} \\
&\quad + \sum_{h_1, h_2, h_3} \text{Variance explained by the interaction of } x_{h_1}, x_{h_2}, \text{ and } x_{h_3} + \ldots \\
&\quad + \sum_{h_1, h_2, \ldots, h_d} \text{Variance explained by the interaction of } x_{h_1}, x_{h_2}, \ldots, \text{ and } x_{h_d}.
\end{aligned}
$$

  (a result from the sparse grid integration)

- The importance of a variable set depends on the percentage of variance it can explain.

Identifying Important Factors: an example

- The performance of an integrated circuit depends on 36 variables.
- Using Gaussian process regression, 2 variables and its interaction contributed 66.4% of the total variation.



Other discussions

1. Predicting multiple performance measure and make trade-offs.
2. Calculation issue: ill-conditioning correlation matrix, better bounding the expected improvement over a region.
3. Adding gradient information.
4. Multi-fidelity simulation models: use the low-fidelity but fast simulation model to add optimization.
5. Other acquisition functions:
   - Probability of Improvement
   - Upper Confidence Bound
   - Entropy Search and Predictive Entropy Search

Outline

Summary

This work focuses on the Bayesian optimization for simulation optimization of continuous parameters.

1. Components of Bayesian optimization:
   1. Gaussian process.
   2. Sampling methods.
2. Acquisition function:
   1. Noiseless samples: Efficient Global Optimization algorithm.
   2. Noisy samples: Knowledge Gradient algorithm.
3. Besides optimization:
   1. Model Validation.
   2. Identifying important factors and visualization.

References I

Bull, A. D. (2011). Convergence rates of efficient global optimization algorithms. *J. Mach. Learn. Res.*, 12:2879–2904.

Hong, L. J. and Zhang, W. (2021). Surrogate-based simulation optimization. *INFORMS TutORials in Operations Research*, null(null):287–311.

THANK YOU!