Bayesian Optimization

31/12/2022

Quang-Huy Nguyen

Outline

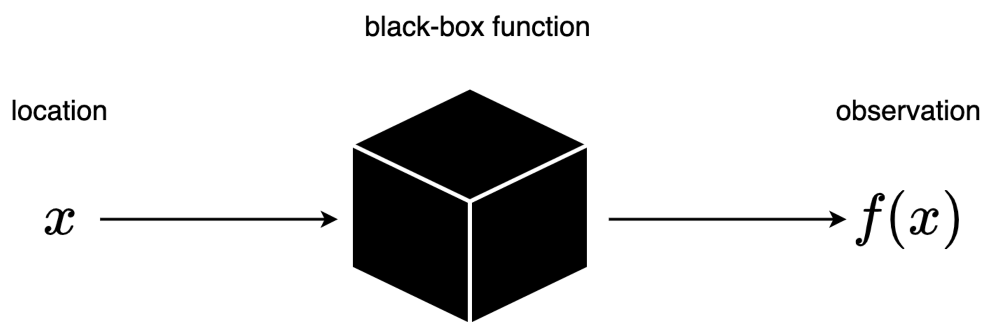
- Black-box Optimization
- Bayesian Optimization framework
- Surrogate Model
- Acquistion Function
- Summarization and Application

Black-box Optimization: Overview

- Given a black-box objective function $f(x)$, the *goal* of the black-box optimization is to search for the *optimal value point* x^* attain with the global maximal $f(x^*)$:

$$x^* = \arg\max_{x \in \mathcal{X}} f(x)$$

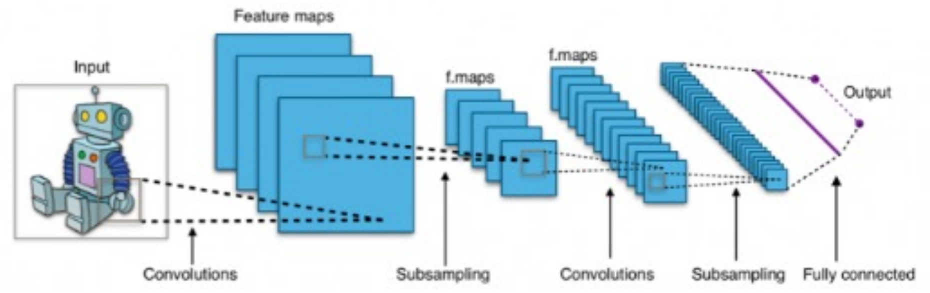
- However, $f(x)$ might be very complex and experiments on the function are ***very expensive***



Black-box optimization: Hyper-parameter tuning

- One popular application of Bayesian Optimization is *Hyperparameter tuning* optimization for training the deep learning model. The challenges are:
  - *Black box*: The deep learning model can be considered as a very complex **black-box** function
  - *Evaluation cost*: The cost of training model (with a set of hyperparameter) can take hours to complete
  - *Noise affection*: Deep model may return different outcome given a same set of hyperparameter (with SGD)
  - *No derivative*: as many hyperparameter can be continuous (LR, weight decay,…) or discrete (number of layers,…), or conditional (number of units per layers)

⇒ The objective is *maximizing the performance* while *mimizing the training time*

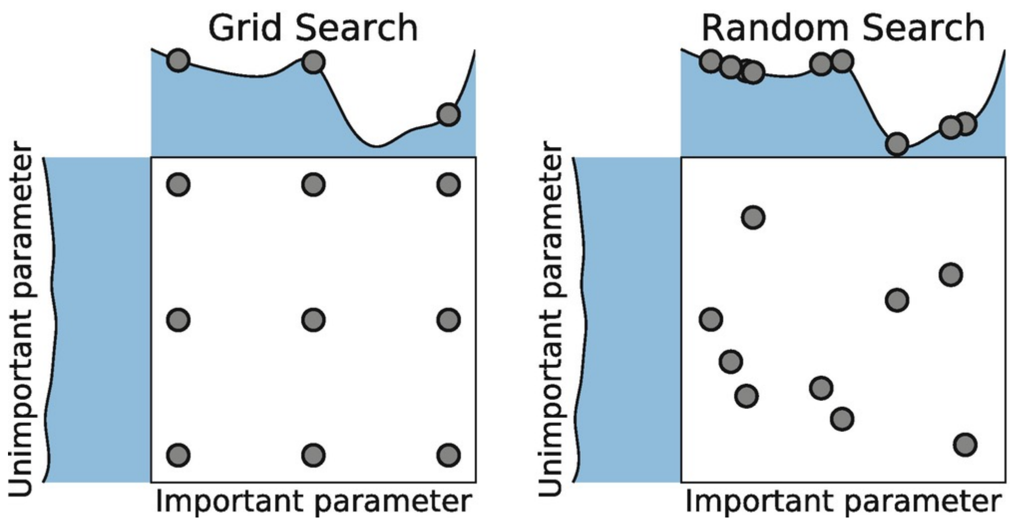


Black-box Optimization: Challenges

- *Search space*: various type of inputs (discrete, continuous, hybrid, conditional,…)
- *Objective function*: Need to perform *expensive experiment* for any observation on x
- *Optimization problem*: how to find the x^* with optimal objective function value?
  - *Trial and error methods*: cost time significantly
  - *Gradient-based methods*: unable to optimize **without gradient**.
  - *Derivative-free methods*: expensive to evaluate => sample inefficient

$\Rightarrow$ **Bayesian optimization:**

- Back-box optimization
- Global optimization
- Derivative-free, Sample efficient
- Accept various types of input



Black-box Optimization: Sequential optimization

- Naturally, optimization is a sequence of making decisions:
  - *Where* to make the next observation?
  - *When* to stop making decision?

$\Rightarrow$ Sequential Optimization:

- *Optimization policy*: examine observed data and select the next location x for observation
- *Observation model*: conduct observation on x to guide the search for the optimal point
- *Termination*: decide whether to continue or to stop the process

---

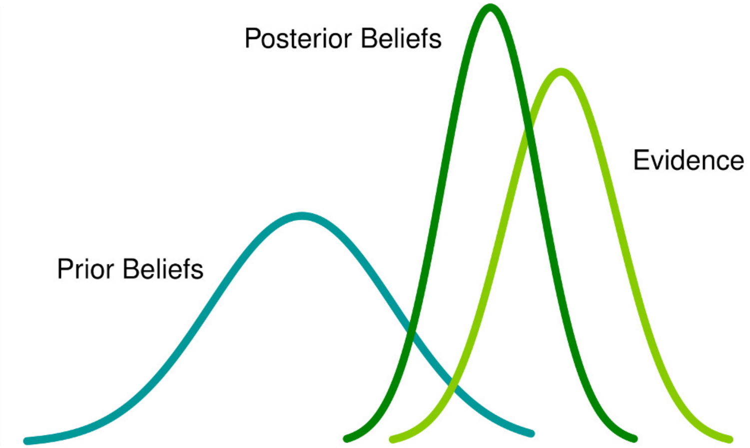
input: initial dataset $\mathcal{D}$ ▸ can be empty
repeat
  $x \leftarrow \textsc{policy}(\mathcal{D})$ ▸ select the next observation location
  $y \leftarrow \textsc{observe}(x)$ ▸ observe at the chosen location
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{(x, y)\}$ ▸ update dataset
until termination condition reached ▸ e.g., budget exhausted
return $\mathcal{D}$

Bayesian Optimization: Bayesian Inference

- “Bayesian inference is a method of statistical inference in which Bayes' theorem is used to update the probability for a hypothesis as more evidence or information become available.”

$$\overbrace{p(\theta \mid \text{data})}^{\text{Posterior}} = \frac{\overbrace{p(\text{data} \mid \theta)}^{\text{Likelihood}} \cdot \overbrace{p(\theta)}^{\text{Prior}}}{\underbrace{p(\text{data})}_{\text{Normalization}}}$$

- Basically, Bayesian inference helps us refine or update our *prior distribution* given *new evidence* to obtain the *posterior distribution* of the objective statistical model.

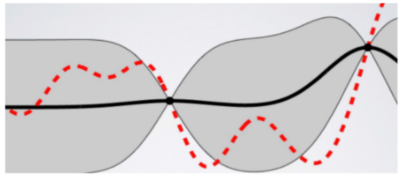


Bayesian Optimization: Key Idea

1. Build a ***surrogate statistical model*** over the function for searching the optimal value

2. Replace the exepensive f by a ***cheaper function*** to find next location

Statistical surrogate model $\mathcal{N}$



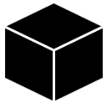
Gaussian Process

Optimize acquistition function α

$$x_{next} = \arg\max_{x \in \mathcal{X}} \alpha(\mathcal{N}, x)$$

Expected Improvement

Expensive function observation

$y_{next} \leftarrow$  $\leftarrow x_{next}$

Update the statistical model

$$p(\mathcal{N} \mid \mathcal{D}) = \frac{p(\mathcal{D}|\mathcal{N}) \cdot p(\mathcal{N})}{p(\mathcal{D})}$$

3. Obtain the new observation on f and *augment the data*

4. Refine the surrogate model with Bayesian Inference

[Bayesian Optimization](#)

Bayesian Optimization: Algorithm

- The central idea of Bayesian Optimization is to build a *statistical distribution* over the objective function and refine it through additional observation with the help of *cheaper-to-optimize function* for the search of global optimal

Algorithm 1 Bayesian Optimization

Require: Set of data points $\mathcal{D}_0 = \{(x, y)\}$; number of step N

Require: Objective function f; Acquisition function α; Gaussian Process $\mathcal{N}(\mu, \sigma)$

1: Place the *surrogate model* (*prior* distribution) on f with the current data $\mathcal{D}_0$:

$$f|\mathcal{D}_0 \sim \mathcal{N}(\mu, \sigma)$$

2: **for** $n = 0, \cdots, N$ **do**

3: Finding the next point by optimizing the *acquisition function* over the distribution:

$$x_{n+1} = \arg\max_{x \in \mathcal{X}} \alpha(x;\ \mathcal{D}_n)$$

4: Measure the objective function observation: $y_{n+1} = f(x_{n+1})$

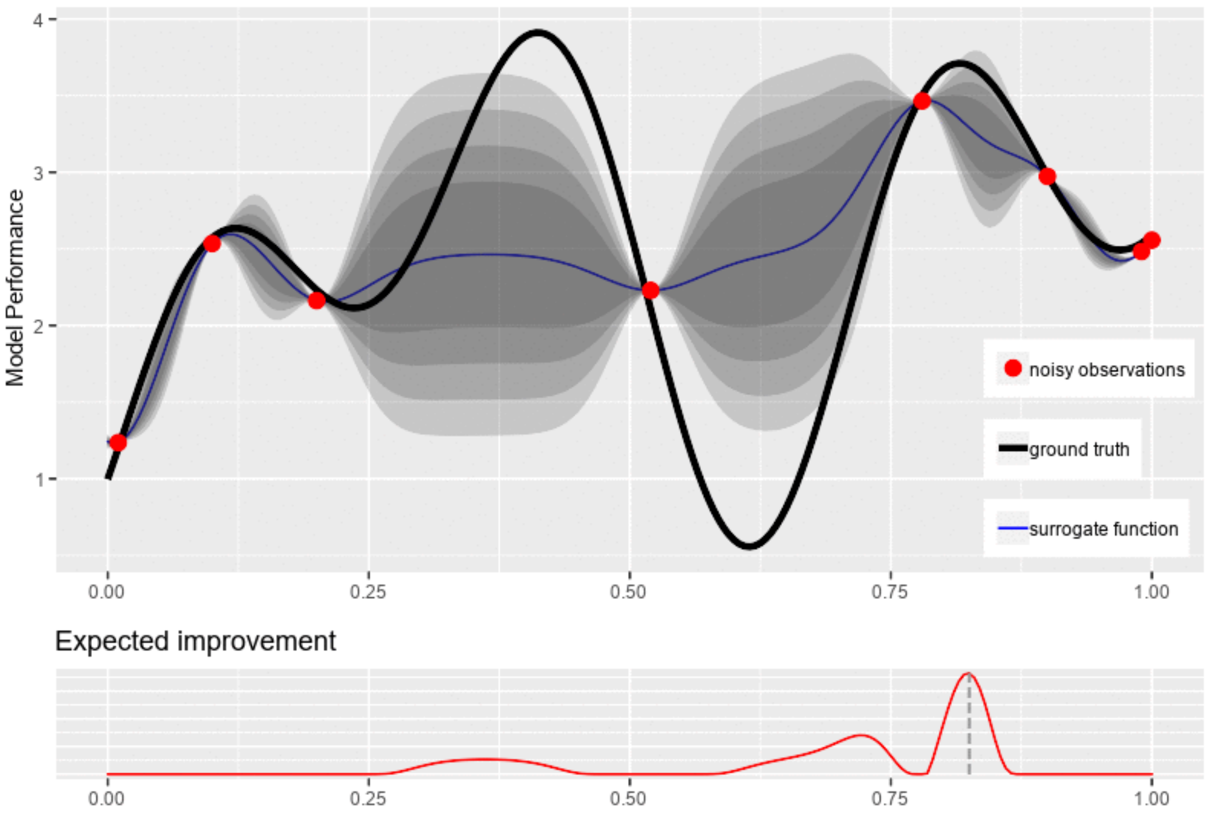
5: Augment data with new observation: $\mathcal{D}_{n+1} = \{\mathcal{D}_n, (x_{n+1}, y_{n+1})\}$

6: Update the surrogate model (*posterior* distribution) based on *Bayesian Inference*:

$$p(\mathcal{N}|\mathcal{D}_{n+1}) = p(\mathcal{D}_{n+1}|\mathcal{N}) \cdot p(f|\mathcal{D}_n)$$

7: **end for**

Return a solution: either the global optimal point x^* with the highest $f(x)$ or the point with the largest posterior mean.

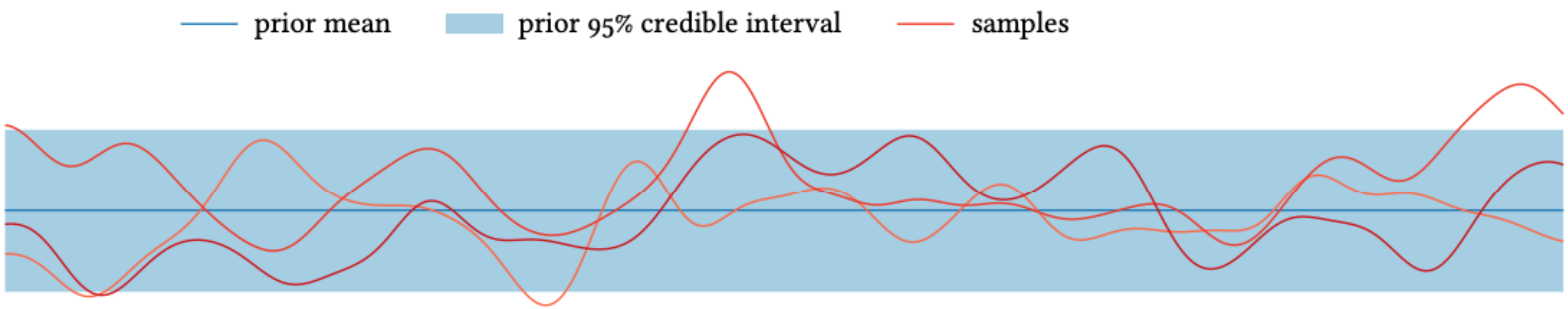


A very beautiful GIF to illustrate how Bayesian Optimization works

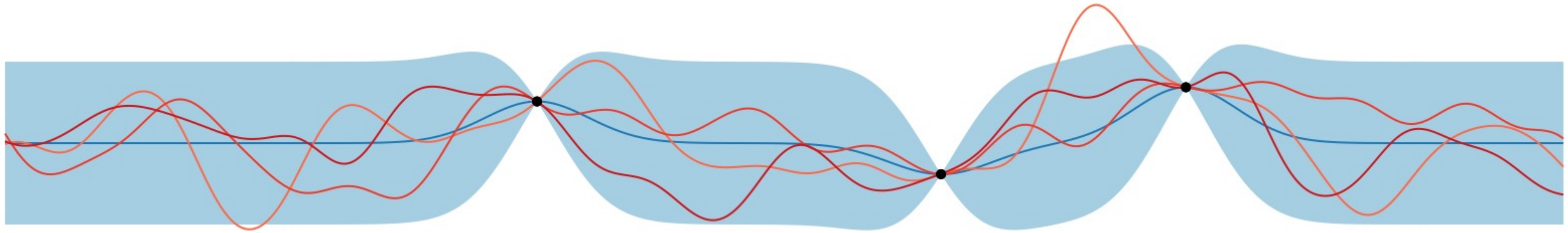
[Bayesian Optimization](#)

Surrogate Model: Overview

- As the objective function f is unknown, is is possible to draw various samples of f. We assume that these samples *follow a certain statistical distribution.*



- The surrogate model is a statistical distribution over f, represent the *uncertainty of the objective function* on any arbitrary location.
- One widely used surrogate model is **Gaussian Process**, assume that any observation on f follows the Normal (Gaussian) distribution



Surrogate Model: Normal Distribution

- Given a random variable x with mean μ and variance σ that follows the *standard normal (Gaussian) distribution*, we have:

$$x \sim \mathcal{N}(\mu, \sigma^2)$$

  - The p.d.f. and c.d.f. of the standard Gaussian distribution are:

$$\phi(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad ; \quad \Phi(x) = \int_{-\infty}^{x} \phi(x') dx'$$

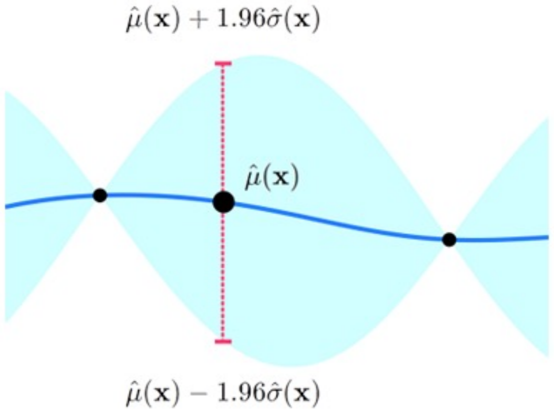
- The *multivariance normal distribution* is the generalization of standard normal distribution for a random vector $\mathbf{x} \in \mathbb{R}^{k}$ with mean vector $\mu \in \mathbb{R}^{k}$ and covariance matrix K $\in \mathbb{R}^{k \times k}$:

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

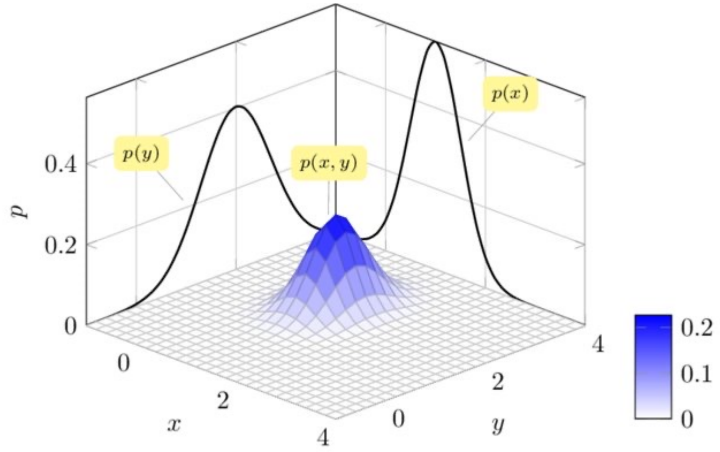
where

$$\mu = \mathbb{E}[x]$$

$$K_{i,j} := k(x_i, x_j) = \mathbb{E}\left[(x_i - \mu_i) \cdot (x_j - \mu_j)\right]$$



Gaussian distribution with 95% credible interval



Two-variance Gaussian distribution

Surrogate Model: Gaussian Process

- **Defining GP**: Given n observed location $\mathbf{x} = \{x\}_{i=1}^{n}$ on an unknown objective function $\boldsymbol{y} = f(\mathbf{x})$, the Gaussian Process assumes that the uncertainty of y follows the multivariance normal distribution, resulting in a *prior distribution*:

$$\mathbf{y} \mid \mathbf{x} \sim \mathcal{N}(\mu, K)$$

  - For the convenience, the *mean value* μ is 0 and the *covariance function* (also called *kernel*) is calculated with parameter θ by:

$$k(x_i, x_j) = \exp\left(-\frac{1}{2\theta^2}\|x_i - x_j\|^2\right)$$

- **Updating GP**: Suppose we have a new, arbitrary data point x_n, and we wish to infer the *posterior distribution* $y_n = f(x_n)$. We can consider y_n and $\boldsymbol{y}$ are a jointly Gaussian:

$$\begin{bmatrix} y_n \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(x_n) \\ \mu(\mathbf{x}) \end{bmatrix}, \begin{bmatrix} K(x_n, x_n) & K(x_n, \mathbf{x}) \\ K(\mathbf{x}, x_n) & K(\mathbf{x}, \mathbf{x}) \end{bmatrix}\right)$$

  - Based on Bayes' theorem, y_n can be distributed by:

$$y_n \mid x_n, \mathbf{x} \sim \mathcal{N}(\mu_n(x_n), \sigma_n^2(x_n))$$

where

$$\mu_n(x_n) = \mu(x_n) + K(x_n, \mathbf{x})^\top K(\mathbf{x}, \mathbf{x})^{-1}(\mathbf{y} - \mu(x_n))$$
$$\sigma_n^2(x_n) = k(x_n, x_n) - K(x_n, \mathbf{x})^\top K(\mathbf{x}, \mathbf{x})^{-1} K(\mathbf{x}, x_n)$$

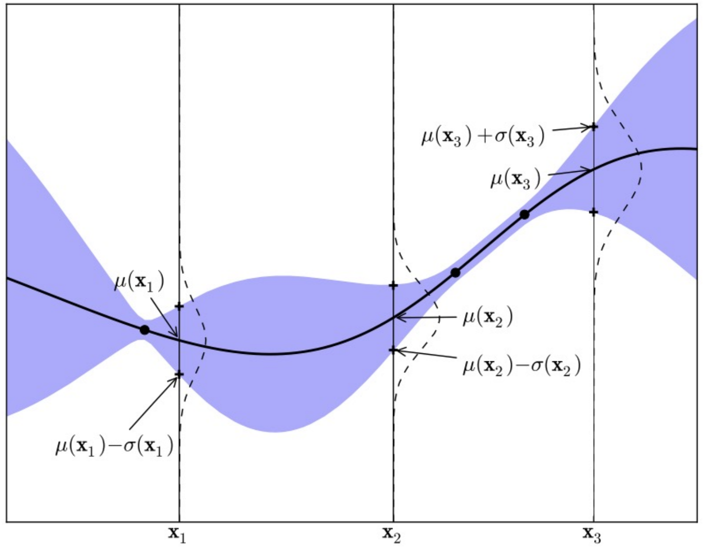
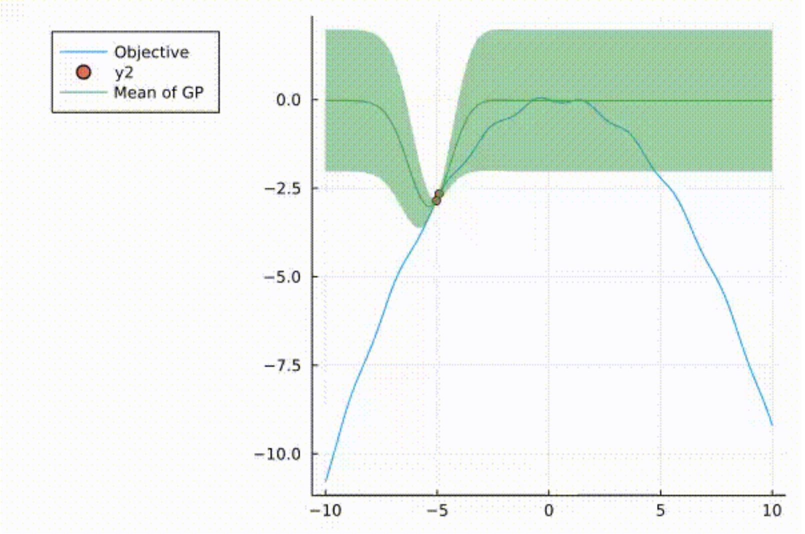


Figure 2: *Simple 1D Gaussian process with three observations. The solid black line is the GP surrogate mean prediction of the objective function given the data, and the shaded area shows the mean plus and minus the variance. The superimposed Gaussians correspond to the GP mean and standard deviation ($\mu(\cdot)$ and $\sigma(\cdot)$) of prediction at the points, $\mathbf{x}_{1:3}$.*



A GIF to illustrate how to the Gaussian Process updates

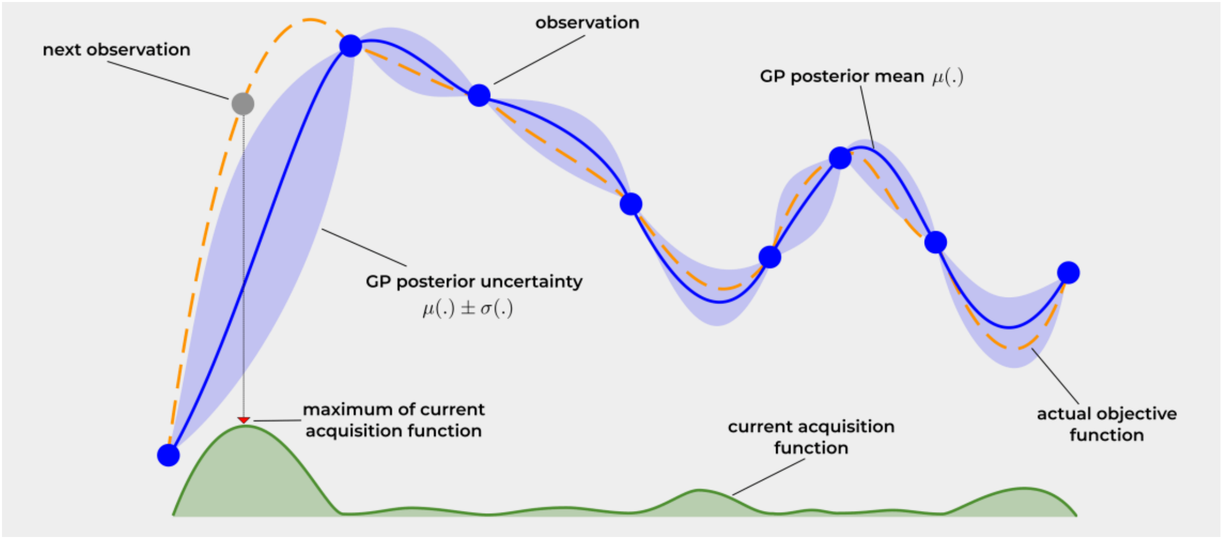
Acquisition Function: Overview

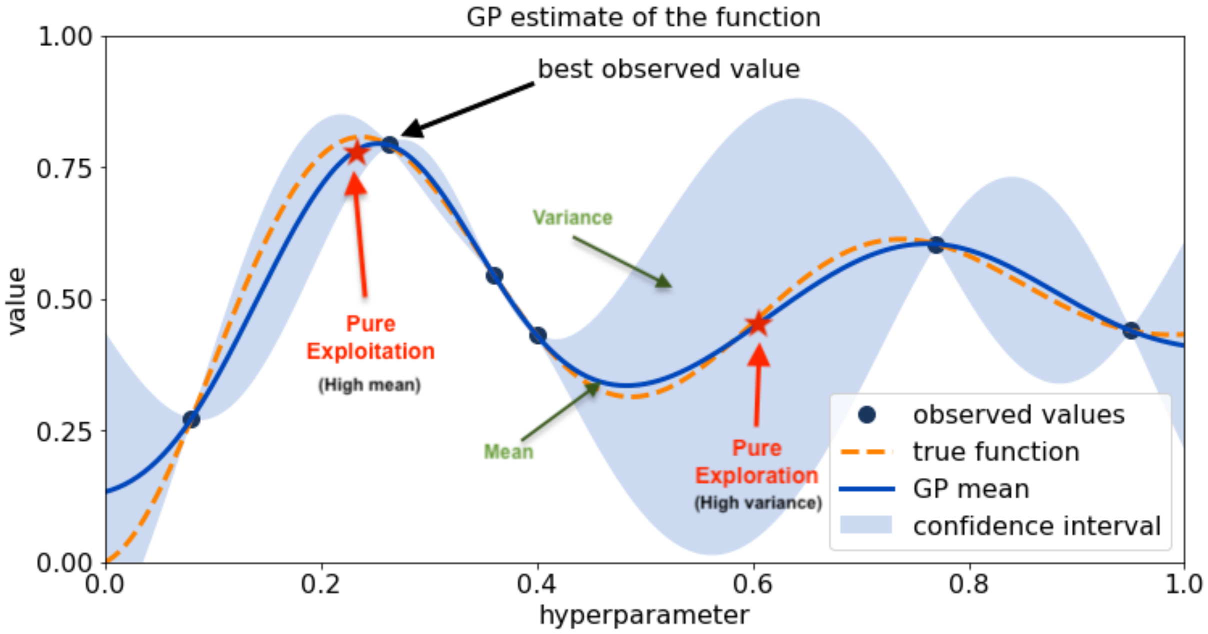
- To update the surrogate model, the observed data should be augmented with new data point x^{n+1}. For the efficiency, *an acquistition function* α is required to guide the search for the optimum value:

$$x^{n+1} = \underset{x \in \mathcal{X}}{\arg\max}\ \alpha(x \mid D_n)$$

- Choosing a suitable acquisition function requires two main challenges:

  1. The optimization cost *must be very cheap* comparing to the black-box f
  2. Balance the trade-off between *exploitation* (high mean) and *exploration* (high variance) to locate x^{n+1}





Acquisition Function: Improvement-based functions

1. **Probability of Improvement (PI)**:
   - Given x^+ is the *current optimum point*, PI measure the probability that any point x *lead to an improvement* on $f(x^+)$, which ε is the trade-off parameter:

   $$\alpha_{PI}(x) = P\big(f(x) \geq f(x^+) + \varepsilon\big) = \Phi\left(\frac{\mu(x) - f(x^+ - \varepsilon)}{\sigma(x)}\right)$$

   - The main drawback of PI is prone to an *exploitation approach*. Moreover, PI also not measure the *amount* of improvement.

2. **Expected Improvement (EI)**:
   - Based on PI, we can calculate the *amount of improvement* by:

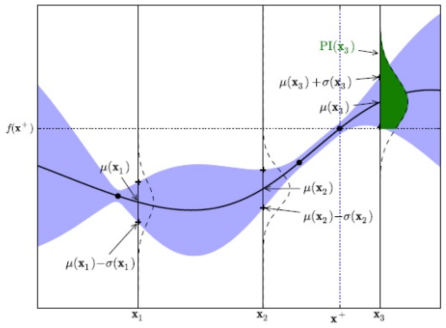
   $$I(x) = \max\left(0, f(x) - f(x^+)\right)$$

   - Therefore, with trade-off parameter ε, the acquistition function can be determined by the *expected of these improvements*, resulting in:

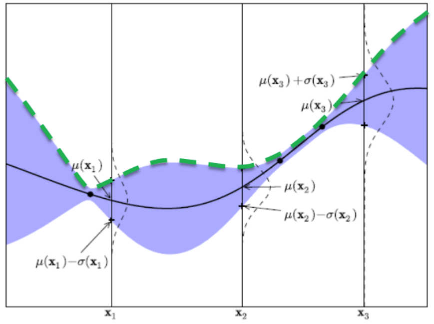
   $$\begin{aligned}\alpha_{EI}(x) &= \mathbb{E}\big[I(x - \varepsilon)\big] \\ &= \mathbb{E}\big[\max\left(0, f(x) - f(x^+) - \varepsilon\right)\big]\end{aligned}$$

3. **Upper Confidence Bound (GP-UCB)**:
   - GP-UCB selects the new sample based on the upper confidence bound of the statistical distribution uncertainty, thus come with a pure *exploration approach*. With ν is the condidence bound parameter, we have:

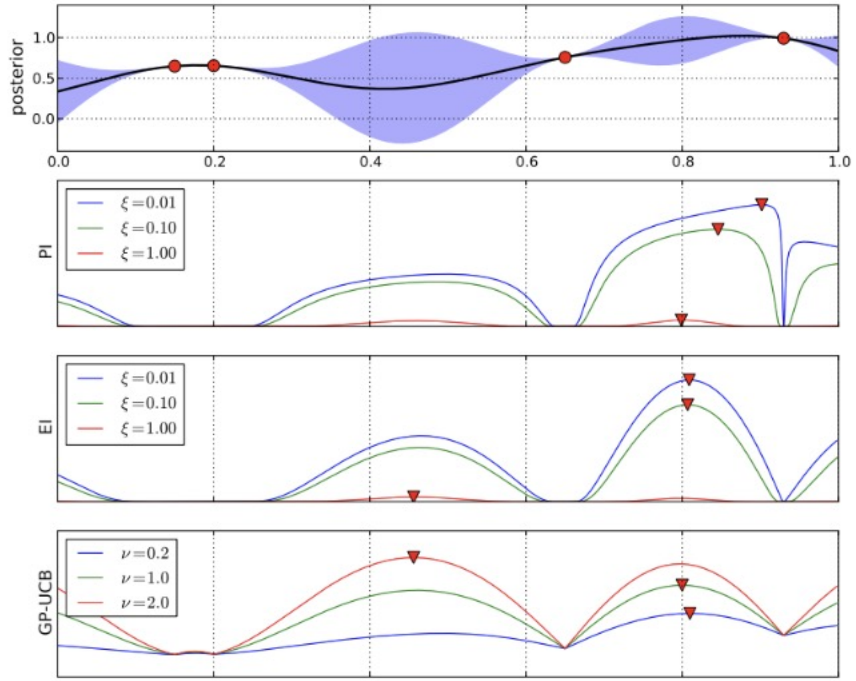
   $$\alpha_{GP-UCB}(x) = \mu(x) + \nu.\sigma(x)$$



Probability/Expected Improvement



Upper confidence bound



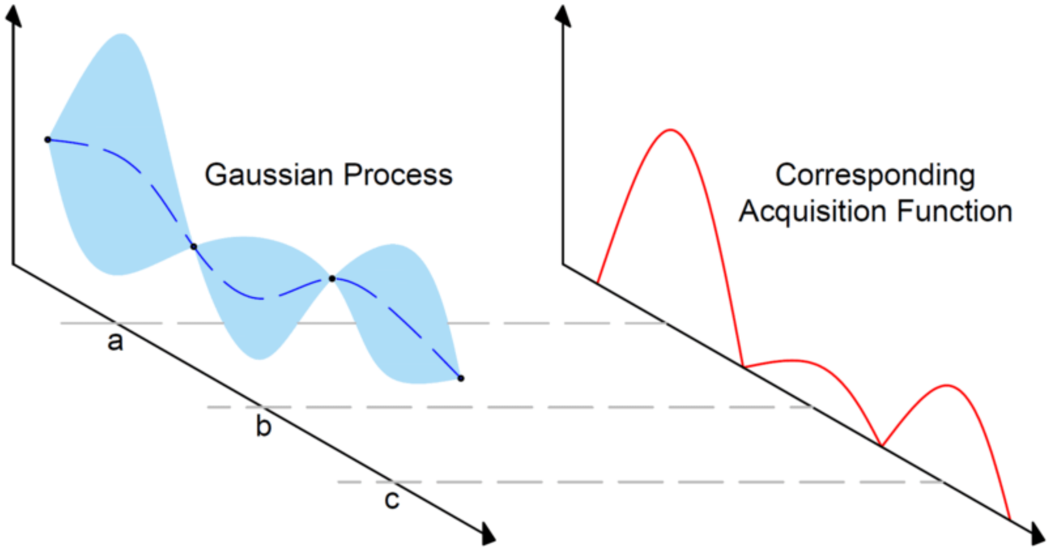
Comparison between 3 functions with different value of parameters

Acquisition Function: Optimization

- As mentioned before, the acquisition function α is often *simple and very cheap* in terms of optimization, comparing to the black-box function f.

- However, as α always is a *non-convex function* and Bayesian Optimization can accept various types of input space, optimizing α share the same challenges with *non-convex* or *multimodal* optimization problems

- For continuous input space, some approaches for optimizing α can be *gradient-based* methods (local search) or *derivative-free* methods (DIRECT).



Summarization and Application: Overview

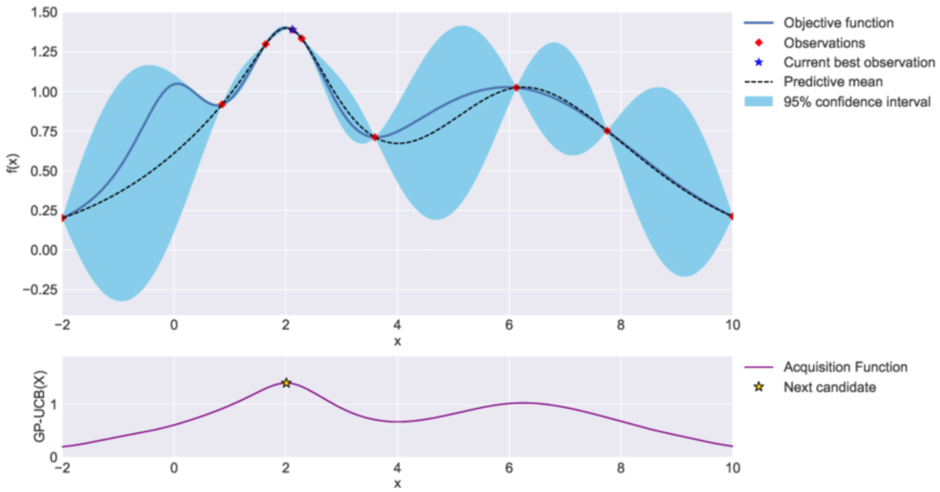
Algorithm 1 Bayesian optimization

1: **for** $n = 1, 2, \ldots$ **do**
2: select new $\mathbf{x}_{n+1}$ by optimizing acquisition function α

$$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}} \alpha(\mathbf{x}; \mathcal{D}_n)$$

3: query objective function to obtain y_{n+1}
4: augment data $\mathcal{D}_{n+1} = \{\mathcal{D}_n, (\mathbf{x}_{n+1}, y_{n+1})\}$
5: update statistical model
6: **end for**

- **Summarization**: Bayesian Optimization seek to find the global optimal of a black-box function through a loop by:
  - Building a statistical ***surrogate model***: distribution over the unknown function
    - *Gaussian Process* is a common choice for surrogage model
  - Sampling new data sample by optimizing an ***acquistition function***
    - The function must be cheap and the balance between the exploration-exploitation trade-off should be made
    - *Expected Improvement* is a common choice with balance trade-off
  - Refine the statistical model with updated data based on Bayesian theorem
    - $\Rightarrow$ Reduce the uncertainty of the statistical model to be approximate with the objective function



- **Challenges and Improvements**:
  - High-dimentional processing: $O(n^3)$ for Gaussian Process $\Rightarrow$ **Linear embeddings: $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{A}\boldsymbol{x})$ *where* $\boldsymbol{g}: \mathbb{R}^{\boldsymbol{d}} \rightarrow \mathbb{R}$**
    Addictive Structure: $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{g_1}(\boldsymbol{x_1}) + \boldsymbol{g_2}(\boldsymbol{x_2})$
  - Discrete/Hybrid input spaces (sequences, trees, graphs) $\Rightarrow$ **Encode them as binary variables (a common practice)**
    Modeling and reasoning over categorical variables
  - Information-based acquisition function:
    - Select samples based on maximize information (*Thompson Sampling, Entropy search*) $\Rightarrow$ **Intuitive but requires expensive approximations**

Summarization and Application: Practical application

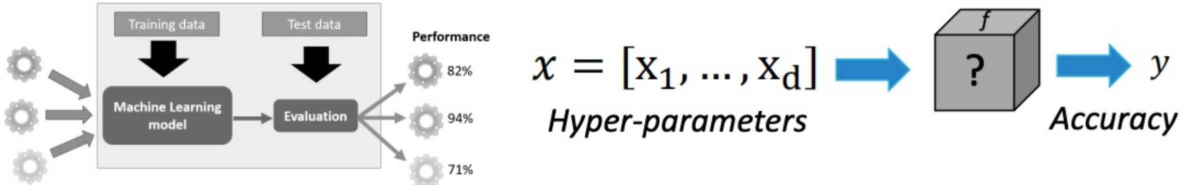
Automotive & Car design

Heat treatment

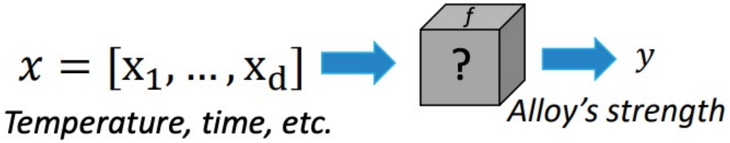
Maximize driving safety
Minimize weight

Maximize alloys' strength
Minimize cost

Real-world optimizations are **costly** and **black-box**.



- ➢ Require training and testing the model.

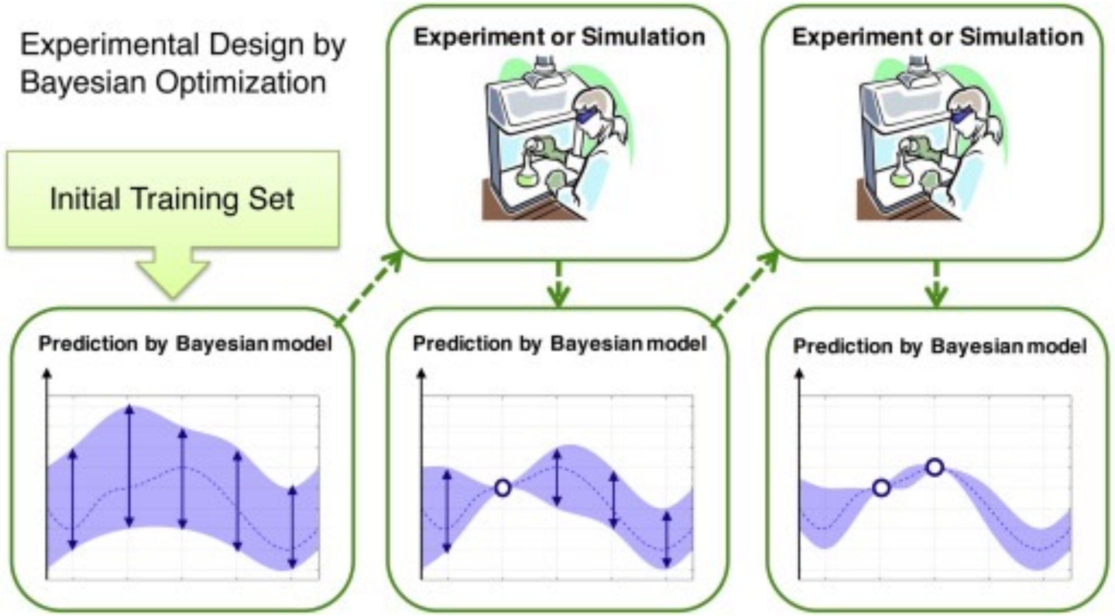


- ➢ Need real materials and time for experiments.

$\Rightarrow$ **Can be tackled with Bayesian Optimization**

Summarization and Application: Future works

- **Agolrithmic research**:
  - *Goal*: Efficiently and optimum
  - *Challenge*: Multimodal , high dimentional, combinatorial structures, resource constraints, noise affection
  - *Approaches*: (i) Reduce the search space; (ii) Modify the surrogate model; (iii) Improve acquisition function
- **Practical application**:
  - Designing new material, alloy, chemical compounds
  - Developing new drug, vaccine, hardware
  - Optimizing manufacturing cost for industrial
- **Cutting edge researches**:
  - *Advances in Bayesian Optimization*, Tutorial, NeurIPS 2022
  - *Recent Advances in Bayesian Optimization*, Tutorial, AAAI 2023
  - *Gaussian Processes, and Spatiotemporal Modeling Systems*, Workshop, NeurIPS 2022
  - *Bayesian Deep Learning*, Workshop, NeurIPS 2021
  - *Bayesian Optimization*, Workshop, NeurIPS 2017



How to design an experiment scheme with BayesOpt effectively

References

- Williams & Rasmussen (2006). *Gausian process for machine learning*. The MIT Press.
- Brochu, E. et al. (2010). *A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning*. arXiv preprint arXiv:1012.2599
- Shahriari, B. et al. (2015). *Taking the human out of the loop: A review of Bayesian optimization*. Proceedings of the IEEE
- Frazier, P. I. (2018). *A tutorial on Bayesian optimization*. arXiv preprint arXiv:1807.02811
- Adams, R. P. (2018). *A Tutorial on Bayesian Optimization for Machine Learning*. University of Toronto, Lecture slide.
- Doppa, J. et al. (2022). *Tutorial on Bayesian Optimization*. Advances in Bayesian Optimization, Tutorial, NeurIPS 2022.

Thank you