

# Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design

Niranjan Srinivas, Andreas Krause<sup>†</sup>, Sham Kakade<sup>††</sup>, Matthias Seeger<sup>†††</sup>

<sup>†</sup>: California Institute of Technology

<sup>††</sup>: University of Pennsylvania

<sup>†††</sup>: Saarland University

January 12, 2023

# 目次

- 1 はじめに
- 2 ベイズ最適化
- 3 GP-UCB
- 4 実験
- 5 まとめ

### 1 はじめに

背景

ブラックボックス関数

ベイズ最適化

本論文の貢献

### 2 ベイズ最適化

### 3 GP-UCB

### 4 実験

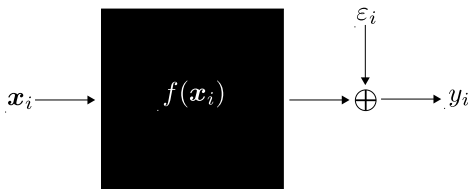
### 5 まとめ

- 多くの実問題は、目的関数に対する最適変数探索問題として定式化できる
  - 耐久性の高いロボットの開発

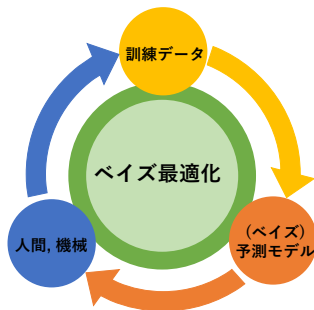
- ブラックボックス関数  $f$

$$y_i = f(x_i) + \varepsilon_i$$

入力： $x_i$ , 出力： $y_i$ , 誤差： $\varepsilon_i$



- 実応用で扱う対象はブラックボックス関数であることが多い
- ブラックボックス関数は、具体的な形状が不明



## モデリング

訓練データから  
予測モデルを構築



## 決定

予測モデルを最も改良する点を  
モデルに基づき決定する



## 観測

選ばれた点を  
実際に観測

- 多くの実問題は、ブラックボックス関数最適化と等価
- 各入力における関数値を得るには、大きなコストがかかる
- できるだけ少ない関数評価回数で最適化を見つけたい
- 有効な手法として**ベイズ最適化**がある

- 本研究の目的
- 本研究の意義
  - 
  -

## 1 はじめに

## 2 ベイズ最適化

問題設定

ガウス過程

ベイズ最適化 1

獲得関数

従来の獲得関数

## 3 GP-UCB

## 4 実験

## 5 まとめ



- 候補入力  $\mathcal{X} = \{x_1, \dots, x_n\}$  が与えられている
- 関数  $f$  を評価して出力  $y_i = f(x_i)$  を得るにはコストがかかる
- できるだけ少ないコストで関数  $f$  を最大化するパラメータ  $x$  を求めたい

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

- ガウス過程は、確率分布を用いた複雑で非線形な関数を扱うためのモデルである。
- このような複雑で非線形な関数を扱う場合には、一般的な最適化手法では、解決が困難になることがある。

## ガウス過程

任意の入力  $\{x_1, \dots, x_n\}$  に対して  $\{f(x_1), \dots, f(x_n)\}$  が  $n$  次元正規分布に従うなら,  $f$  はガウス過程に従う.

- 関数  $f$  がガウス過程に従うことを  $f(x) \sim \mathcal{GP}(\mu(x), k(x, x'))$
- 平均関数:  $\mu(x) = \mathbb{E}[f(x)]$
- 共分散関数 (カーネル関数):  
 $k(x, x') = \mathbb{E}[(f(x) - \mu(x))(f(x') - \mu(x'))]$

- $f$  にガウス過程事前分布を仮定する
- 訓練データに基づき,  $f$  の事後分布を求める
- 事後分布に基づき最も最大値となりそうな点を次に観測する
- 観測した  $(x_{next}, y_{next})$  を訓練データに追加し, 再び事後分布を求める



variance only

$$\boldsymbol{x} = \arg \max_{\boldsymbol{x}} \sigma_T^2(\boldsymbol{x})$$

mean only

$$\boldsymbol{x} = \arg \max_{\boldsymbol{x}} \mu_T(\boldsymbol{x})$$



1 はじめに

2 ベイズ最適化

3 GP-UCB

提案手法

GP-UCB のアルゴリズム

4 実験

5 まとめ

- 探索・活用に特化した

## GP-UCB: Gaussian Process Upper Confidence Bound

$$\mathbf{x}_t = \arg \max_{\mathbf{x}} \mu_{t-1}(\mathbf{x}) + \beta_t^{1/2} \sigma_{t-1}(\mathbf{x}).$$

- $\beta_t$  は，探索の度合いを表す定数
  - $\beta_t$  が小さい  $\rightarrow$  探索
  - $\beta_t$  が大きい

---

**Algorithm 1** The GP-UCB algorithm.

---

**Input:** Input space  $\mathcal{D}$ ; GP Prior  $\mu_0 = 0, \sigma_0, k$

**for**  $t = 1, 2, \dots$  **do**

    Choose  $\mathbf{x}_t = \arg \max_{\mathbf{x} \in \mathcal{D}} \mu_{t-1}(\mathbf{x}) + \sqrt{\beta_t} \sigma_{t-1}(\mathbf{x})$

    Sample  $y_t \leftarrow f(\mathbf{x}_t) + \varepsilon_t$

    Perform Bayesian update to obtain  $\mu_t$  and  $\sigma_t$

**end for**

---



1 はじめに

2 ベイズ最適化

3 GP-UCB

4 実験

実験手法

実験対象データ

実験結果

5 まとめ

- 実験手法
  - variance only
  - maximum mean
  - Expected Improvement(EI)
  - Most Probable Improvement(MPI)
  - GP-UCB
- Mean Average Regret を用いた評価
  - $\text{Regret} = (\text{真の関数の最大値}) - (\text{観測済みデータの最大値})$
  - Average Regret : 1 回の試行中の Regret の平均
  - Mean Average Regret : 全試行の Average Regret の平均

## 1. 合成データ

- $\mathcal{D} \in [0, 1]$  の範囲を一様に 1000 点で区切った点を候補点とする
- $T = 1000, \sigma^2 = 0.025, \delta = 0.1$
- 30 回試行

## 2. 温度データ

- 46 個のセンサーから 1 分間隔で 5 日以上計測された気温
- $T = 46, \sigma^2 = 0.5, \delta = 0.1$
- 2000 回試行

## 3. 交通データ

- 357 個のセンサーから 1 ヶ月, 午前 6 時から午前までに通過する車の速度
- $T = 357, \sigma^2 = 4.78, \delta = 0.1$



- 1 はじめに
- 2 ベイズ最適化
- 3 GP-UCB
- 4 実験
- 5 **まとめ**  
**結論**

