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

Niranjan Srinivas, Andreas Krause[†], Sham Kakade[††], Matthias Seeger[†††]

†: California Institute of Technology
††: University of Pennsylvania
†††: Saarland University

January 15, 2023

目次

Next Section

背景

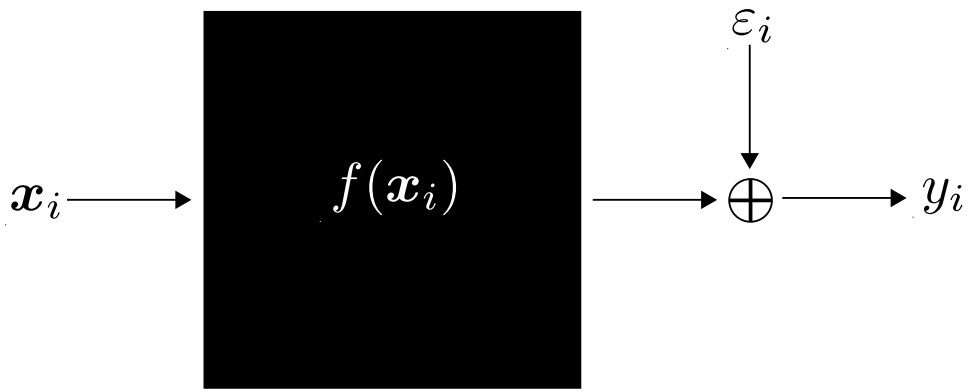
- 多くの実問題は目的関数に対する最適変数探索問題として定式化できる
  - 耐久性の高いロボットの開発

ブラックボックス関数

- ブラックボックス関数 f

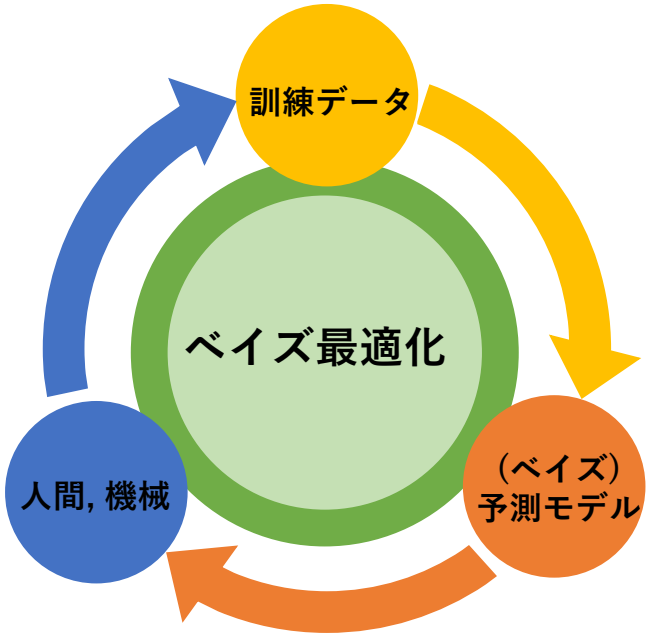
$$y_i = f(\boldsymbol{x}_i) + \varepsilon_i$$

入力：$\boldsymbol{x}_i$，　出力：y_i，　誤差：ε_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

ベイズ最適化



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

本論文の貢献

貢献

- ガウス過程を用いた最適化問題を解決するためのアルゴリズムを提案
- 実データ実験において優れた性能を発揮

Next Section

問題設定

- 候補入力 $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ が与えられている
- 関数 f を評価して出力 $y_i = f(\boldsymbol{x}_i)$ 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

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

  - 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

ガウス過程

任意の入力 $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ に対して $\{f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)\}$ 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 $f(\boldsymbol{x}) \sim \mathcal{GP}(\mu(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}'))$
- 平均関数: $\mu(\boldsymbol{x}) = \mathbb{E}[f(\boldsymbol{x})]$
- 共分散関数 (カーネル関数):
  $k(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}[(f(\boldsymbol{x}) - \mu(\boldsymbol{x}))(f(\boldsymbol{x}') - \mu(\boldsymbol{x}'))]$

ベイズ最適化 1

- f にガウス過程事前分布 $\mathcal{GP}(0, k(\boldsymbol{x}, \boldsymbol{x}'))$ を仮定する
- 訓練データ $(X, \boldsymbol{y}) = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{t}$ に基づき f の事後分布を求める
- 事後分布に基づき最適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

$$f(\boldsymbol{x}_*) \mid \mathcal{D} \sim N\left(\underbrace{\boldsymbol{k}_{\mathcal{D},\boldsymbol{x}_*}^{\top} K_{\mathcal{D}\mathcal{D}}^{-1} \boldsymbol{y}}_{\text{posterior mean}}, \underbrace{k_{\boldsymbol{x}_*\boldsymbol{x}_*} - \boldsymbol{k}_{\mathcal{D},\boldsymbol{x}_*}^{\top} K_{\mathcal{D}\mathcal{D}}^{-1} \boldsymbol{k}_{\mathcal{D},\boldsymbol{x}_*}}_{\text{posterior variance}}\right)$$

$$K_{\mathcal{D}\mathcal{D}} = \begin{pmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}_1, \boldsymbol{x}_t) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_t, \boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}_t, \boldsymbol{x}_t) \end{pmatrix}, \boldsymbol{k}_{\mathcal{D},\boldsymbol{x}_*} = \begin{pmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_*) \\ \vdots \\ k(\boldsymbol{x}_t, \boldsymbol{x}_*) \end{pmatrix}, k_{\boldsymbol{x}_*,\boldsymbol{x}_*} = k(\boldsymbol{x}_*, \boldsymbol{x}_*)$$

獲得関数

探索する点を選択するために使用される関数

- 獲得関数をを $\alpha(\boldsymbol{x}_{t-1})$ とすると次に観測する点 $\boldsymbol{x}_t$ は以下のように決定

$$\boldsymbol{x}_t = \arg\max_{\boldsymbol{x}} \ \alpha(\boldsymbol{x}_{t-1})$$

獲得関数の設計指針

従来の獲得関数

variance only

$$\boldsymbol{x}_t = \arg\max_{\boldsymbol{x}} \ \sigma^2_{t-1}(\boldsymbol{x})$$

- 予測分散が最大となる x を選択
  $\Rightarrow$ 探索中心

mean only

$$\boldsymbol{x}_t = \arg\max_{\boldsymbol{x}} \ \mu_{t-1}(\boldsymbol{x})$$

- 予測平均が最大となる x を選択
  $\Rightarrow$ 活用中心

Next Section

提案手法

- 探索・活用に特化した手法では最適化がうまくいかない場合が多い
  - $\Rightarrow$ 探索と活用のバランスが取れるような手法を提案

GP-UCB: Gaussian Process Upper Confidence Bound

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

- β_t は，探索の度合いを表す定数
  - β_t が小さい $\Rightarrow$ 予測平均の比重が大きくなる $\Rightarrow$ 活用重視
  - β_t が大きい $\Rightarrow$ 予測分散の比重が大きくなる $\Rightarrow$ 探索重視

GP-UCB のアルゴリズム

Algorithm 1 The GP-UCB algorithm.

Input: Input space $\mathcal{D}$; GP Prior $\mu_0 = 0$, σ_0, k

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

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

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

Perform Bayesian update to obtaion μ_t and σ_t

end for

Next Section

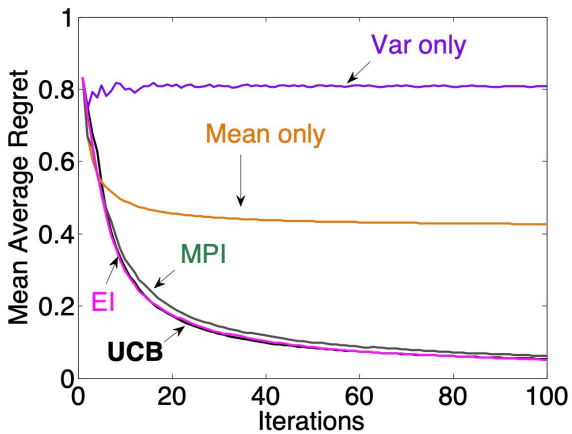
実験手法

- 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
  - Regret = (真の関数の最大値) - (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

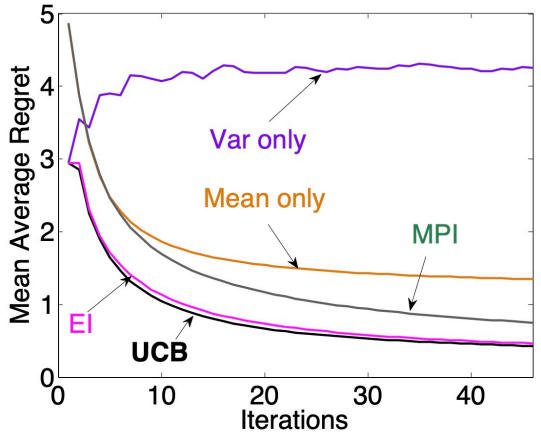
実験対象データ

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 ヶ月間午前 6 時から午前までに通過する車の速度
   - $T = 357$, $\sigma^2 = 4.78$, $\delta = 0.1$

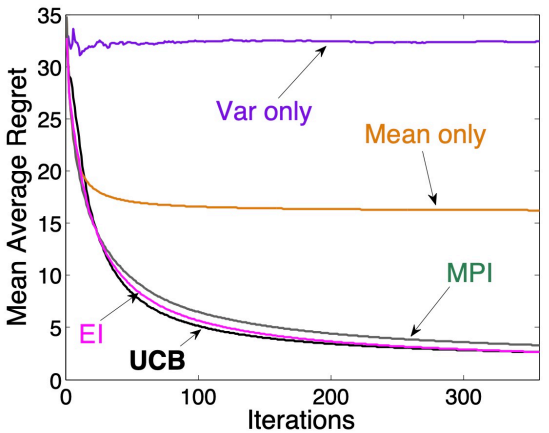
実験結果



(a) *Squared exponential*



(b) *Temperature data*



(c) *Traffic data*

- すべての実験結果において GP-UCB が損失少なく最適化を行なっている
- 既存手法の EI,MP と比較して同等以上の性能を示した

Next Section

結論

- GP-UCB というベイズ最適化の獲得関数を新たに提案
  - GP-UCB は探索と活用を両立する獲得関数
- GP-UCB は既存の手法と同等以上の性能を示した
  - Mean Average Regret の観点で実データと合成データにおける性能を発揮