

# 機械学習のハイパーパラメータ最適化

# 目次

1. ハイパーパラメータ最適化とは
2. Black-box 最適化とは

# 1. ハイパーパラメータ最適化とは

# ハイパーパラメータ最適化の概要

- 中身がブラックボックスな関数の最適化
  - Black-box 最適化

## 2. Black-box 最適化とは

# Black-box 最適化の概要

- 関数の中身がBlack-box で勾配情報等が使えない
- 1回の評価に時間がかかる
- 少ない評価回数で良い解を探索したい
- Black-box 最適化の有力な手法
  - ベイズ最適化
    - GP-EI
    - TPE
  - CMA-ES

# ベイズ最適化の概要

- ベイズ最適で一般的に利用される手法は下記2種類
  - GP-EI
  - TPE

# ベイズ最適化 (GP-EI) の概要

- ガウス過程(GP)によって目的関数をモデル化する
- 評価値の改善量の期待値(EI)が最大となる点を選択



# ベイズ最適化 (GP-EI) のアルゴリズム

1. ガウス過程(GP)により目的関数を予測
2. 期待値(EI)を計算して最適化
3. 2で得られえた点を評価
4. 1～3を繰り返す

# ベイズ最適化 (TPE) の概要

- Hyperopt, Optuna がデフォルトで採用しているアルゴリズム
- 期待値(EI)を使うところまでは、GP-EI と同じ
- GP-EI とは目的関数に対するモデル化が異なる

# ベイズ最適化の比較(GP-EI, TPE):カテゴリーカル変数

- GP-EI
  - one-hot encoding（多くが採用されている）
  - カテゴリーカル変数用のカーネルを設定
- TPE
  - 直接扱うことが可能

# ベイズ最適化の比較(GP-EI, TPE):次元数

- 低次元（～6）の場合
  - GP-EI > TPE
- 高次元（10～）の場合
  - GP-EI < TPE

# ベイズ最適化の比較(GP-EI, TPE):次元数

- GP-EI

- 探索空間全体に対するモデル化が必要
- 次元数の増加で探索空間の体積も指数的に増加
  - スケールしづらい

- TPE

- 既に存在する点から評価値上位を推定
- 有望な点付近の部分空間を探せる

# CMA-ES の概要

- 進化計算における手法の一つ
- 多変量正規分布から解を生成することで最適化を行っている
  - 更新式の一部は自然勾配法に対応している

# CMA-ES のアルゴリズム

1. 正規分布から解を生成
2. 全ての解を評価して重み付けする
3. 正規分布のパラメータを更新する
  - 正規分布からサンプルされる点の期待評価値に対する自然勾配方向への更新に対応
4. 1～3を繰り返す

# ベイズ最適化(GP-EI)とEMA-ESの比較:評価回数

- 評価回数が少ない場合 ( $10 \times$  次元数)
  - GP-EI > CMA-ES
- 評価回数が多い場合 ( $100 \times$  次元数)
  - GP-EI < CMA-ES



# ベイズ最適化(GP-EI, TPE)とEMA-ESの比較: 時間計算量

- iteration 数 $t$  に対する時間計算量
  - GP-EI ;  $O(t^3)$
  - TPE ;  $O(t)$
  - CMA-ES ;  $O(1)$
- 評価回数が膨大にある場合、CMA-ES の方が良い

# Black-box 最適化のPython ライブラリー

- GP-EI ; GpyOpt
- TPE ; Optuna
- CMA-ES ; pycma