

ユーザの選好推定に基づく多目的ベイズ最適化による ハイパーパラメータ最適化

Hyperparameter Optimization by Multi-objective Bayesian Optimization
based on Inference of User Preference

尾崎令拓^{*1}
Ryota Ozaki

高木優介^{*1}
Yusuke Takagi

烏山昌幸^{*1}
Masayuki Karasuyama

竹内一郎^{*1*2}
Ichiro Takeuchi

^{*1}名古屋工業大学
Nagoya Institute of Technology

^{*2}理化学研究所
RIKEN

AutoML considers hyper-parameter optimization (HPO) of machine learning models. However, there often exist multiple evaluation indices for the learned models. For example, both model accuracy and memory size can be objective functions, which are typically in the trade-off relation. In this case, the importance of each objective function depends on the user preference. To incorporate the preference adaptively into HPO, we propose a preference-learning-based multi-objective Bayesian optimization (PL-MBO) method. Since directly specifying the exact preference can be difficult for the user, PL-MBO considers only querying a ‘relative preference’ that the user can give much easier. By combining a Bayesian user preference model and the standard Gaussian process model of objective functions, the expected improvement criterion of the user preference is derived. Our numerical experiments show that the optimal solution based on the user preference can be found efficiently in HPO for neural networks.

1. はじめに

機械学習モデルの自動構築を目的とする AutoML では、モデルのハイパーパラメータ最適化問題を考える必要がある。また、実践的には、モデルの精度やメモリサイズ、多クラス分類におけるクラスごとの精度など、複数の目的関数に関する多目的最適化が必要になることがある。ハイパーパラメータ最適化においては最適化したい対象はブラックボックスであることが多いため、ベイズ最適化など、目的関数を推定しながら探索を行う手法がよく用いられる（特に多目的の場合は、多目的ベイズ最適化とも呼ばれる）。

多目的最適化では、目的関数間のトレードオフにより、一般には解が一意に定まらない。実際、ニューラルネットワークの例では、ハイパーパラメータとなり得るモデルのユニット数を増やすことで、推論精度が向上するかわりにメモリ使用量が増加してしまうという状況は十分に考えられる。あるいは、多クラス分類でのクラスごとの精度を考えると、あるクラスの recall の上昇が他のクラスの recall を犠牲にすることが起こり得る。以上のような性質から、多目的最適化では、パレート解と呼ばれる数学的な上下関係が定められないすべての解を列挙することを目的としている。

実問題においてユーザが多目的最適化を利用する際には、列挙された解の中から必要な解の一つを選んで採用することが多い。しかし、パレート解の全列挙には多くの計算コストがかかる。ここで、パレート解を列挙してからユーザが解を選択するのではなく、多目的最適化の段階でユーザがどのような解を必要としているのかという情報を考慮することで、ユーザが求める解のみを効率的に発見しようという考えが生じる。このような考えに基づく多目的最適化の既存研究として、ユーザが求める目的関数値の範囲をあらかじめ指定するような手法 [Yang 16, Paria 20] や、ユーザがどの目的関数を重視するかをあらかじめ指定するような手法 [He 20] が存在する。しかし、多目的最適化を行うユーザが、目的関数値の範囲や目的関数の重視度合いを事前に指定するのは困難な状況も多い。

本稿では、多目的最適化に関するユーザの好みを、ユーザが具体的な値として直接与えるのではなく、より容易に与えられる情報として「解の良さの比較情報」や「目的関数値の改善要望」から逐次的にベイズ推定し、それを多目的最適化に取り入れることを考える。解の比較や改善要望といった間接的な観測から事後分布がどのように定式化できるかを示し、ユーザの好みの不確かさも反映した獲得関数を提案する。実験として、ベンチマーク関数による評価と、ニューラルネットワークのハイパーパラメータ最適化問題を考えた実データによる評価を行い、本稿で提案するフレームワークによってユーザにとって好ましい解を効率的に探索することができる状況があることを示す。

2. 多目的最適化とパレート最適性

複数の目的関数を同時に最適化する問題は多目的最適化と呼ばれる。通常、多目的最適化の目的関数間にはトレードオフの関係があり、最適解が一意には定まらず、パレート最適と呼ばれる性質を満たす解が複数存在し得る。 L 個の目的関数 $f^l: \mathcal{X} \mapsto \mathbb{R}$ ($l \in \{1, \dots, L\}, \mathcal{X} \subseteq \mathbb{R}^d$) があり、二つの入力 $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$ について以下が成り立つとき、 \mathbf{x}_i は \mathbf{x}_j を支配しているという。

$$\forall l \in \{1, \dots, L\}, f^l(\mathbf{x}_i) \geq f^l(\mathbf{x}_j),$$

$$\exists k \in \{1, \dots, L\}, f^k(\mathbf{x}_i) > f^k(\mathbf{x}_j).$$

この関係を $\mathbf{x}_i \succ \mathbf{x}_j$ と表す。ここで、入力 \mathbf{x} が他のどの入力にも支配されないとき、つまり以下が成り立つとき、 \mathbf{x} をパレート解と呼ぶ：

$$\forall \mathbf{x}' \in \mathcal{X} \setminus \{\mathbf{x}\}, \mathbf{x}' \not\succ \mathbf{x}.$$

多目的最適化ではこのパレート解の列挙を目指すことが最も一般的である。

連絡先: 尾崎令拓, ozaki.ryota.mllab.nit@gmail.com
烏山昌幸, karasuyama@nitech.ac.jp

3. 選好推定を用いた多目的ベイズ最適化

問題によってはパレート解は大量に存在し、全列挙は必ずしも簡単でないことが知られている。しかし、現実の問題設定では、得られた解の中から人間（ユーザ）が採用したい解を一つ選択することになる場合が多い。そこで、多目的最適化にユーザの好み（選好）を導入することで、ユーザの求める解を効率的に発見することを考える。

観測が高価なブラックボックス関数の最適化を行う有名な手法にベイズ最適化があり、多目的最適化への拡張も行われてきた。ベイズ最適化では確率的なモデルを用いることで、目的関数の不確かさを考慮することができる。本稿ではさらに、相対的な選好比較からユーザの好みを確率的に推定し、ユーザの「好みの不確かさ」も探索に組み込む方法を提案する。

3.1 問題設定

目的関数 $f^l(\mathbf{x})$ について、ユーザがどの目的関数をどの程度重視しているのかを考慮した最大化問題を考える。以降、 $f_{\mathbf{x}}^l = f^l(\mathbf{x})$ と略記し、ある \mathbf{x} における目的関数の値を並べたベクトルを $\mathbf{f}_{\mathbf{x}} = [f_{\mathbf{x}}^1, \dots, f_{\mathbf{x}}^L]^T$ と定義する。目的関数はブラックボックスであり、ガウス過程回帰によりモデル化されているとする。ある \mathbf{x}_i に対する観測は各目的関数に独立なガウスノイズ $\epsilon_i \stackrel{i.i.d.}{\sim} N(0, \sigma^2)$ が加わった $y_i^l = f_{\mathbf{x}_i}^l + \epsilon_i$ として得られ、 \mathbf{x}_i と $\mathbf{y}_i = [y_i^1, \dots, y_i^L]^T$ の t 個の組の集合 $D_t = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^t$ から事後分布 $p(\mathbf{f}_{\mathbf{x}} | D_t)$ が定められるとする [Williams 06]。

ある解 $\mathbf{f}_{\mathbf{x}}$ についてのユーザにとっての好ましさを表す関数 $U_{\mathbf{w}}: \mathbb{R}^L \mapsto \mathbb{R}$ が存在するとする。ただし、 $\mathbf{w} \in \mathbb{R}^L$ は選好の度合いを調整するパラメータであり、詳細は後述する。このとき、ユーザの選好最適化は、 $U_{\mathbf{w}}$ の \mathbf{x} に関する最大化として解釈できる：

$$\mathbf{x}_{\star} := \arg \max_{\mathbf{x} \in \mathcal{X}} U_{\mathbf{w}}(\mathbf{f}_{\mathbf{x}}).$$

3.2 提案フレームワークの構成

本稿では、直接与えることが困難であるユーザの選好を推定し、それを用いた多目的ベイズ最適化を提案する。提案手法は主に 3 つの部分から構成される。フレームワークの全体像を表す図を図 1 に示す。構成要素の一つ目は目的関数の推定であり、これは各目的関数に対して独立なガウス過程回帰によって行う（図 1①）。二つ目はユーザへの選好に関する情報の問い合わせである（図 1②）。本稿では、選好に関する情報として、ユーザがある 2 つの解についてどちらを好むのかという情報と、ある解においてユーザがどの目的関数値を改善したいかという情報の 2 種類の情報を考える。本稿の計算機実験においては簡単な設定として、多目的ベイズ最適化のイテレーションごとに、1) 観測済みの入力のうち最も良い入力と 2) その入力に対して最も改善したい目的関数がユーザによって与えられるという設定とするが、実際にはこれらの 2 種類の情報は任意のタイミングで任意の量与えることが可能である。三つ目の構成要素は、選好の推定と、選好を用いた獲得関数の計算である（図 1③）。ユーザから得られる選好に関する情報（選好推定の訓練データとなる）から選好をベイズ的に推定し、ベイズ最適化における獲得関数の計算に用いる。以下では、選好の表現方法とその推定、獲得関数の設計について述べる。

3.3 選好表現

多目的最適化に選好を導入する手法として、選好を超直方体として表現する手法 [Yang 16, Paria 20] や、選好をベクトルとして表現する手法 [He 20] がある。選好を超直方体として表現する手法は、パレート解のうち、事前情報として与えられた

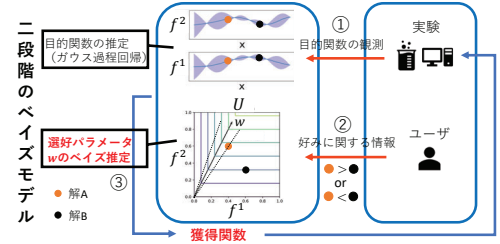


図 1: 提案フレームワークの全体像。

超直方体の内部に存在する解を列挙するものである。しかし、事前に目的関数の形状が分からない場合、超直方体の適切な定義が困難である。また、パレート解が超直方体の内部に存在するとは限らない点、解の良さの評価が超直方体の領域内であるか領域外であるかという 2 値でしかできない点が欠点として考えられる。選好をベクトルとして表現する手法は、パレート解のうち、与えられたベクトルの方向に近い解を良い解とするものである。ベクトルによる選好表現では、ベクトルとの近さを定義することで解の良さを連続的に評価することができるという利点がある。しかし、ユーザがベクトルを直接定義することは、特に高次元の問題において困難となりうる。

本稿では、ユーザの選好を表現する手法として、解の良さを連続的に評価できる、ベクトル表現に基づく効用関数を考える。まず、選好を表すベクトル（選好ベクトル）を $\mathbf{w} = [w_1, \dots, w_L]^T$ と定義し、制約を $w_i > 0 (i = 1, \dots, L), \|\mathbf{w}\|_1 = 1$ とする。効用関数は、多目的最適化のある入力 \mathbf{x} に対する $\mathbf{f}_{\mathbf{x}}$ を入力にとり、その \mathbf{x} に対する好ましさを度合いをスカラー値として返すものとし、以下のように定義する。

$$U_{\mathbf{w}}(\mathbf{f}_{\mathbf{x}}) = \min \left(\frac{f_{\mathbf{x}}^1}{w_1}, \dots, \frac{f_{\mathbf{x}}^L}{w_L} \right).$$

ここで、選好ベクトル \mathbf{w} は、多目的最適化の各目的関数に対する重みパラメータとなっている。選好ベクトルを用いる際の効用関数 $U_{\mathbf{w}}$ には様々な定義が考えられるが、本稿における U は、多目的最適化における入力同士の支配関係を崩さずに選好を表現できるという点から定義されるものである。実際、

$$\mathbf{x} \succ \mathbf{x}' \Rightarrow U_{\mathbf{w}}(\mathbf{x}) \geq U_{\mathbf{w}}(\mathbf{x}'),$$

が成り立っている。なお、このような、入力に対して重みづけされた値の最小値（最大値）を関数値とするスカラーゼーション関数は、Chebyshev scalarizing function [Giagkiozis 15] として知られている。

以上のように、効用関数を定義することで、多目的最適化の解に対するユーザの選好を表現することができた。しかし、選好ベクトルはユーザが事前に直接定義しなければならず、これは目的関数の形状や多目的最適化において取りうる解の分布が分からない状態では、特に目的関数の数が大きい場合に困難である。よって、ユーザから選好に関する情報を訓練データとして受け取ることで、効用関数のパラメータ \mathbf{w} を逐次的に推定することを考える。

3.4 選好のベイズ推定

ユーザから得られる相対的な選好情報からパラメータ \mathbf{w} をベイズ的に推定することを考える。事前分布として、 $w_i > 0 (i = 1, \dots, L), \|\mathbf{w}\|_1 = 1$ の制約を定めたことから、この制

約を満たす確率分布である Dirichlet 分布を採用する:

$$p(\mathbf{w}; \boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{i=1}^L \mathbf{w}_i^{\alpha_i - 1}.$$

ただし, $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_L]^\top$ は集中パラメータ, B はベータ関数である. 本稿では, ユーザーが与える選好表現として, 1) 二つの解に対する好ましきの比較 (一対比較) と, 2) ある入力について, どの目的関数値を改善したいかを選択する情報 (勾配比較) の2種類を考え, このとき \mathbf{w} の事後分布がどのように得られるかを示す.

まず, ユーザから一対比較として解 $\mathbf{f}_{\mathbf{x}_i}$ が解 $\mathbf{f}_{\mathbf{x}'_i}$ よりも良い解であると提示した場合を考える. これを $\mathbf{f}_{\mathbf{x}_i} \succ \mathbf{f}_{\mathbf{x}'_i}$ と表すとする. 一対比較が, 効用関数値にガウスノイズ $\varepsilon_i, \varepsilon'_i \stackrel{i.i.d.}{\sim} N(0, \sigma^2)$ が加わった後の大小関係として観測されたと考えると

$$\mathbf{f}_{\mathbf{x}_i} \succ \mathbf{f}_{\mathbf{x}'_i} \Leftrightarrow U_{\mathbf{w}}(\mathbf{f}_{\mathbf{x}_i}) + \varepsilon_i > U_{\mathbf{w}}(\mathbf{f}_{\mathbf{x}'_i}) + \varepsilon'_i,$$

のように表現することができる. 訓練データとして, n 個の独立な比較 $\{\mathbf{f}_{\mathbf{x}_i} \succ \mathbf{f}_{\mathbf{x}'_i}\}_{i=1}^n$ が得られたとする. このとき, \mathbf{w} の尤度は以下のように書くことができる:

$$L(\mathbf{w} \mid \{\mathbf{f}_{\mathbf{x}_i} \succ \mathbf{f}_{\mathbf{x}'_i}\}_{i=1}^n) = \prod_{i=1}^n \Phi \left(\frac{U_{\mathbf{w}}(\mathbf{f}_{\mathbf{x}_i}) - U_{\mathbf{w}}(\mathbf{f}_{\mathbf{x}'_i})}{\sqrt{2}\sigma} \right),$$

ただし, Φ は標準正規分布の累積分布関数である. この尤度の導出はガウス過程回帰の選好観測拡張 [Chu 05] で使われているものと同様の考え方である (ただし U のモデルが異なる).

次に, ユーザーがある解 $\mathbf{f}_{\mathbf{x}_j}$ について, 目的関数 $f_{\mathbf{x}_j}^{l_j}$ を目的関数 $f_{\mathbf{x}_j}^{l'_j}$ より改善したいと提示する場合を考える. これを $l_j \succ l'_j$ と書くこととする (l_j, l'_j は目的関数の番号 $1, \dots, L$ のいずれかを表す). 例えば, ユーザーが最も改善したい目的関数を指定するような場合も, 選ばれた目的関数とその他の目的関数の比較としてこの方法で表現できる. ここで, $f_{\mathbf{x}_j}^{l_j}$ を $f_{\mathbf{x}_j}^{l'_j}$ よりも改善したいという情報は, $U_{\mathbf{w}}(\mathbf{f}_{\mathbf{x}_j})$ において $f_{\mathbf{x}_j}^{l_j}$ 方向の傾きが $f_{\mathbf{x}_j}^{l'_j}$ 方向の傾きより大きいと解釈できる (選好がより増加すると考えられるため). 一対比較と同様, ユーザの判断にはノイズ $\varepsilon_j \stackrel{i.i.d.}{\sim} N(0, \sigma^2)$ が加わると仮定すると, 勾配比較は以下のように表現できる:

$$l_j \succ l'_j \Leftrightarrow g^{l_j}(\mathbf{f}_{\mathbf{x}_j}) - g^{l'_j}(\mathbf{f}_{\mathbf{x}_j}) > \varepsilon_j,$$

ただし, $g^l(\mathbf{f}_{\mathbf{x}_j}) = \frac{\partial U_{\mathbf{w}}(\mathbf{f}_{\mathbf{x}})}{\partial f_{\mathbf{x}}^l} \Big|_{\mathbf{f}_{\mathbf{x}}=\mathbf{f}_{\mathbf{x}_j}}$ は $\mathbf{f}_{\mathbf{x}_j}$ における U の次元 l での勾配である. 勾配比較が m 個独立に得られたときの訓練データは $\{l_j \succ l'_j\}_{j=1}^m$ となる. このとき, 一対比較の場合と同様の考え方により, 尤度は以下のように表現することができる:

$$L(\mathbf{w} \mid \{l_j \succ l'_j\}_{j=1}^m) = \prod_{j=1}^m \Phi \left(\frac{g^{l_j}(\mathbf{f}_{\mathbf{x}_j}) - g^{l'_j}(\mathbf{f}_{\mathbf{x}_j})}{\sqrt{2}\sigma} \right).$$

パラメータ \mathbf{w} のベイズ推定における \mathbf{w} の事後分布は, $D_t^{\text{pre}} = \{(\mathbf{f}_{\mathbf{x}_i} \succ \mathbf{f}_{\mathbf{x}'_i})\}_{i=1}^n \cup \{l_j \succ l'_j\}_{j=1}^m$ とすると,

$$p(\mathbf{w} \mid D_t^{\text{pre}}) \propto p(\mathbf{w}; \boldsymbol{\alpha}) \times L(\mathbf{w} \mid \{\mathbf{f}_{\mathbf{x}_i} \succ \mathbf{f}_{\mathbf{x}'_i}\}_{i=1}^n) L(\mathbf{w} \mid \{l_j \succ l'_j\}_{j=1}^m),$$

と表せる. 事後分布を解析的に計算することは困難であるが, マルコフ連鎖モンテカルロ法 (MCMC) などを用いることで, 事後分布からのサンプリングが可能である.

Algorithm 1: 選好推定を用いた多目的ベイズ最適化 (PL-MBO)

```

function PL-MBO( $D_0, D_0^{\text{pre}}$ )
  for  $t = 1, \dots, T$  do
     $\alpha_{t-1}(\mathbf{x}) \leftarrow \text{PreferenceEI}(D_{t-1}, D_{t-1}^{\text{pre}})$ 
     $\mathbf{x}_t \leftarrow \arg \max_{\mathbf{x} \in \mathcal{X}_*} \alpha_{t-1}(\mathbf{x})$ 
     $\mathbf{y}_t \leftarrow \mathbf{f}_{\mathbf{x}_t} + \boldsymbol{\varepsilon}_t, \boldsymbol{\varepsilon}_t \sim N_L(\mathbf{0}, \sigma^2 I_L)$ 
      ( $I_L$ :  $L$ -dimensional identity matrix)
     $D_t \leftarrow D_{t-1} \cup \{(\mathbf{x}_t, \mathbf{y}_t)\}$ 
     $D_t^{\text{pre}} \leftarrow D_{t-1}^{\text{pre}} \cup \{\text{New user preferences}\}$ 
  end
end

function PreferenceEI( $D_t, D_t^{\text{pre}}$ )
  for  $k = 1, \dots, K$  do
    Sample  $\mathbf{f}_{\mathbf{x}}^{(k)}$  from  $p(\mathbf{f}_{\mathbf{x}} \mid D_t)$ 
      ( $\mathbf{f}_{\mathbf{x}}^{(k)}$ : the  $k$ -th sample of  $\mathbf{f}_{\mathbf{x}}$ )
    Sample  $\mathbf{w}^{(k)}$  from  $p(\mathbf{w} \mid D_t^{\text{pre}})$ 
  end
   $\alpha_t(\mathbf{x}) \leftarrow \frac{1}{K} \sum_k \max\{U_{\mathbf{w}^{(k)}}(\mathbf{f}_{\mathbf{x}}^{(k)}) - U_{\mathbf{w}^{(k)}}(\mathbf{f}_{\mathbf{x}_{\text{best}}}), 0\}$ 
  return  $\alpha_t(\mathbf{x})$ 
end

```

3.5 獲得関数

選好を考慮した多目的最適化の目的は, U の値が最大となる入力をできるだけ少ない観測の数で求めることであった. よってこれを反映した獲得関数として, 効用関数 U の値の改善量が大きくなるような解を探索する指標である, U に関する Expected Improvement (EI) [Jones 98] を考える. U に関する EI は, 観測済みの入力の中でユーザーが最も好む入力を $\mathbf{x}_{\text{best}} = \arg \max_{\mathbf{x} \in \mathcal{X}_s} U_{\mathbf{w}}(\mathbf{f}_{\mathbf{x}})$ (\mathcal{X}_s : 観測済みの入力の集合) とすると, 以下のように求めることができる.

$$\alpha_{\text{preEI}}(\mathbf{x}) = \mathbb{E}_{\mathbf{f}_{\mathbf{x}}, \mathbf{w}} [\max\{U_{\mathbf{w}}(\mathbf{f}_{\mathbf{x}}) - U_{\mathbf{w}}(\mathbf{f}_{\mathbf{x}_{\text{best}}}), 0\}]. \quad (1)$$

獲得関数値は, 目的関数 \mathbf{f} と選好パラメータ \mathbf{w} を事後分布からサンプリングすることで, モンテカルロ法により近似的に計算が可能である. 本稿で提案した選好を考慮した多目的ベイズ最適化では, イテレーションごとに $\mathbf{x}_{\text{next}} = \arg \max_{\mathbf{x} \in \mathcal{X}_*} \alpha_{\text{preEI}}(\mathbf{x})$ (\mathcal{X}_* は未観測の入力点の集合) となる入力 \mathbf{x}_{next} を求めて次の観測点とすることで, 効率よくユーザにとって好ましい入力 (多目的最適化の解) を求めることが期待される. 本稿で提案するフレームワークのアルゴリズムの概要を Algorithm1 に示す. T は最大イテレーション数, K は式 (1) のモンテカルロ近似のサンプル数である.

4. 計算機実験

計算機実験により, 提案手法の評価を行う. 最初にベンチマーク関数を用いた人工データ実験による評価を行い, その後ニューラルネットワーク (NN) に関する実データを用いた実験結果を示す. 評価指標は, 真の選好パラメータ \mathbf{w}_{true} を用いた効用関数 $U_{\mathbf{w}_{\text{true}}}$ に関する simple regret とし, $\max_{\mathbf{x} \in \mathcal{X}} U_{\mathbf{w}_{\text{true}}}(\mathbf{f}_{\mathbf{x}}) - \max_{\mathbf{x} \in \mathcal{X}_s} U_{\mathbf{w}_{\text{true}}}(\mathbf{f}_{\mathbf{x}})$ で表す. この simple regret が小さいほどユーザにとって良い解を発見できているということになる. 提

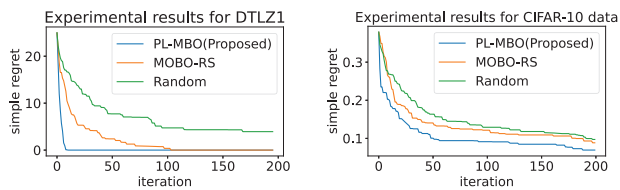


図 2: Simple regret の推移比較. 左図がベンチマーク関数を用いた実験, 右図が CIFAR-10 を用いた NN データに対する実験である.

案手法に対する比較手法は, 候補点の中から一様ランダムに次の探索点を定める Random と, 選択を用いない通常の多目的最適化の手法である, Multi-Objective Bayesian Optimization using Random Scalarizations (MOBO-RS)[Paria 20] とする. 多目的最適化の実行にあたっては, 探索対象となる入力候補点の中からランダムに 4 点を選んで初期点とし, 乱数シードを変えて各手法とも 20 回ずつ実験を行った. 実験結果として示す simple regret は, 20 回の実験で得られた simple regret の各イテレーションでの平均値である.

4.1 人工データ実験

人工データ実験として, 多目的最適化の入力 \mathbf{x} が 3 次元, 出力 (目的関数) $\mathbf{f}_{\mathbf{x}}$ が 3 次元となるデータを考える. 目的関数として, 多目的最適化のベンチマーク関数として使われる DTLZ1[Deb 05] を用いる. 入力 \mathbf{x} の候補点の数は 1000 点であり, 具体的には x_1, x_2, x_3 についてそれぞれ $(0, 1)$ の範囲で等間隔に 10 点のグリッドに切って候補点としている. 真のユーザの選好として, 効用関数における真の重み $\mathbf{w}_{\text{true}} = [\frac{1}{4}, \frac{1}{4}, \frac{2}{4}]^T$ を仮定する. これは, ユーザが目的関数 f^3 を重視しているということと対応している. 実験結果として得られる regret の推移は図 2 (左) の通りである. 図 2 から, 提案手法が最も早く simple regret の値が小さくなっていることが分かる. つまり, あらかじめ仮定したユーザの真の選好に沿った好ましい解が, 提案手法によって効率的に探索できていることが分かる.

4.2 実データ実験

実データ実験として考える多目的最適化問題は, NN のハイパーパラメータ (HP) 最適化問題である. 具体的には, CIFAR-10[Krizhevsky 09] を用いて, クラス数 $C = 10$ の分類を行う NN モデルの Loss 関数におけるクラスごとの重みを HP とする. NN のある入力データに対する重み付き Cross-entropy Loss の式は $\sum_{i=1}^C -\lambda_i y_i \log \hat{y}_i$ で示され, $\lambda_1, \dots, \lambda_{10}$ が HP となる. ただし, y_i が正解ラベルの one hot 表現, \hat{y}_i がクラス i の予測確率とする. また, 目的関数は, テストデータによるモデル評価における各クラスの recall とする ($\mathbf{f}_{\mathbf{x}}$: 10 次元). これは, 多クラス分類問題において, ユーザが各クラスに対する分類の重視度合いを調整したいという要望があるということを仮定した設定である. NN モデルは, Residual Network[He 16] から fine tuning した. 入力 (HP の組み合わせ) の候補点の数は 5005 点であり, 具体的には $\lambda_1, \dots, \lambda_{10}$ が $(0, 1)$ の範囲で 7 通りの値を取り, かつ $\sum_{i=1}^{10} \lambda_i = 1$ という制約を満たすものとした. 真の選好パラメータは, $\mathbf{w}_{\text{true}} = [\frac{1}{10.3}, \frac{1}{10.3}, \frac{1}{10.3}, \frac{1}{10.3}, \frac{1}{10.3}, \frac{1}{10.3}, \frac{1}{10.3}, \frac{1}{10.3}, \frac{1}{10.3}, \frac{1}{10.3}]^T$ とした. これは, 3 番目のクラスの recall を高めたいという要望に相当する. 実験結果として得られる regret の推移は図 2 (右) の通りである. 人工データと同様, 提案手法の優位性が確認できる.

5. まとめ

本稿では, 多目的ベイズ最適化にユーザの好みの推定を導入し, ユーザにとって望ましい解を効率的に探索するフレームワークを提案した. 計算機実験により, 提案手法が有効に働く状況があることを示した.

謝辞:本研究の一部は, 科学研究費 (20H00601,21H03498), JST CREST (JPMJCR21D3), JST ムーンショット型研究開発事業 (JPMJMS2033-05), JST AIP 加速研究 (JPMJCR21U2) NEDO (JPNP18002, JPNP20006), 理化学研究所革新知能統合研究センターの補助を受けて行われた.

参考文献

- [Chu 05] Chu, W. and Ghahramani, Z.: Preference learning with Gaussian processes, in *Proceedings of the 22nd international conference on Machine learning*, pp. 137–144 (2005)
- [Deb 05] Deb, K., Thiele, L., Laumanns, M., and Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization, in *Evolutionary multiobjective optimization*, pp. 105–145, Springer (2005)
- [Giagkiozis 15] Giagkiozis, I. and Fleming, P. J.: Methods for multi-objective optimization: An analysis, *Information Sciences*, Vol. 293, pp. 338–350 (2015)
- [He 16] He, K., Zhang, X., Ren, S., and Sun, J.: Deep residual learning for image recognition, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778 (2016)
- [He 20] He, Y., Sun, J., Song, P., Wang, X., and Usmani, A. S.: Preference-driven Kriging-based multiobjective optimization method with a novel multipoint infill criterion and application to airfoil shape design, *Aerospace Science and Technology*, Vol. 96, p. 105555 (2020)
- [Jones 98] Jones, D. R., Schonlau, M., and Welch, W. J.: Efficient global optimization of expensive black-box functions, *Journal of Global optimization*, Vol. 13, No. 4, pp. 455–492 (1998)
- [Krizhevsky 09] Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
- [Paria 20] Paria, B., Kandasamy, K., and Póczos, B.: A flexible framework for multi-objective Bayesian optimization using random scalarizations, in *Uncertainty in Artificial Intelligence*, pp. 766–776PMLR (2020)
- [Williams 06] Williams, C. K. and Rasmussen, C. E.: *Gaussian processes for machine learning*, Vol. 2, MIT press Cambridge, MA (2006)
- [Yang 16] Yang, K., Li, L., Deutz, A., Back, T., and Emmerich, M.: Preference-based multiobjective optimization using truncated expected hypervolume improvement, in *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pp. 276–281IEEE (2016)