Cascaded Algorithm-Selection and Hyper-Parameter Optimization with Extreme-Region Upper Confidence Bound Bandit *

Yi-Qi Hu, Yang Yu and Jun-Da Liao
National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China {huyq, yuy}@lamda.nju.edu.cn, liaojd98@gmail.com

Abstract
An automatic machine learning (AutoML) task is to select the best algorithm and its hyper-parameters simultaneously. Previously, the hyper-parameters of all algorithms are joint as a single search space, which is not only huge but also redundant, because many dimensions of hyper-parameters are irrelevant with the selected algorithms. In this paper, we propose a cascaded approach for algorithm selection and hyper-parameter optimization. While a search procedure is employed at the level of hyper-parameter optimization, a bandit strategy runs at the level of algorithm selection to allocate the budget based on the search feedbacks. Since the bandit is required to select the algorithm with the maximum performance, instead of the average performance, we thus propose the extreme-region upper confidence bound (ER-UCB) strategy, which focuses on the extreme region of the underlying feedback distribution. We show theoretically that the ER-UCB has a regret upper bound $O(K \ln n)$ with independent feedbacks, which is as efficient as the classical UCB bandit. We also conduct experiments on a synthetic problem as well as a set of AutoML tasks. The results verify the effectiveness of the proposed method.

1 Introduction
Algorithm selection and hyper-parameter optimization are core parts of automatic machine learning (AutoML). Previously, AutoML approaches often define the search space as the algorithm selection space [Brazdil et al., 2003; Adankon and Cheriet, 2009; Biem, 2003], hyper-parameter space [Hu et al., 2018; Hu et al., 2019], or the joint of the both spaces (CASH problem) [Feurer et al., 2015; Thornton et al., 2013]. While the joint space allows a more thorough search that could cover potentially better configurations, the huge space is a barrier to effective search in limited time. Moreover, the joint space can be quite redundant when considering only one of the algorithms, since the hyper-parameters of the other algorithms are irrelevant. Therefore, the joint space contains redundancy or even can be misleading.

The cascaded algorithm selection can have levels [Jamieson and Talwalkar, 2016]. The first level is on the hyper-parameter optimization. It only needs to focus on the selected algorithm, but not the hyper-parameters of all algorithms. The second level is on the algorithm selection. However, previous methods in this kind commonly carry out a full hyper-parameter optimization on the candidate algorithms, making the slow and expensive algorithm evaluations.

In this paper, we propose a cascaded algorithm selection approach to avoid a full-space hyper-parameter optimization. The hyper-parameter optimization usually employs some stepping search methods, which can be paused after every search step, and can also be resumed. The selection receives feedback and allocates the next search step to one of the algorithms. Thus, the cascaded algorithm selection is naturally to be modeled as a multi-armed bandit problem [Auer et al., 2002]. However, most of the classical bandits maximize the average feedbacks. In the AutoML, however, only the best feedback matters. A variant of the bandit, the extreme bandit [Carpentier and Valko, 2014], can model this situation, which tries to identify the arm with the maximize (or equivalently minimize) feedback value. However, as the extreme bandit follows the extreme distribution, it is not only unstable but often require to known the distribution type, making the extreme bandit approach unpractical.

In this paper, we propose the extreme-region UCB bandit (ER-UCB), which focuses on the extreme region of the feedback distributions. Unlike the extreme bandit, ER-UCB considers a region instead of the extreme point, which can lead to a better mathematical condition. Moreover, in machine learning where the test data is commonly different from the train data, the extreme region can be more robust for generalization. In $K$-arms and $n$ trials, our analysis proves that ER-UCB has the $O(K \ln n)$ regret upper bound, which has the same order with the classical UCB strategy. The experiments on synthetic and real AutoML tasks reveal that the ER-UCB can find the best algorithm precisely, and exploit it with the majority of the trial budget.

---

*This work is supported by the National Key R&D Program of China (2017YFB1001903), NSFC (61876077), Jiangsu SF (BK20170013), and Collaborative Innovation Center of Novel Software Technology and Industrialization. Yang Yu is the corresponding author.
The rest sections present background & related works, extreme-region UCB bandit, experiments, and conclusion.

2 Background & Related Works

We consider the algorithm selection and hyper-parameter optimization on classification tasks. Let \( D_{train} \) and \( D_{valid} \) denote the training and testing datasets. Let \( C = \{ C_1, C_2, \ldots, C_K \} \) denote the algorithm set with \( K \) candidates. For \( C_i \in C \), \( \delta_i \in \Delta_i \) denotes a hyper-parameter setting, where \( \Delta_i \) is the hyper-parameter space of \( C_i \). Let \( f(\cdot) \) denote a performance criterion for a configuration \( C_{i}^{\delta_i} \), e.g., accuracy, AUC score, etc. The AutoML problem can be formulated as follows:

\[
C_{i}^{\delta_i} = \arg\max_{C_i \in C, \delta_i \in \Delta_i} \frac{1}{k} \sum_{j=1}^{k} f\left(C_{i}^{\delta_i}, D_{train}^{j}, D_{valid}^{j}\right),
\]

where \( D_{valid}^{j} \in D_{train} \) and \( D_{train}^{j} = D_{train} - D_{valid}^{j} \). It is also called the CASH problem formulation [Feurer et al., 2015].

Because of the non-convex, non-continuous and non-differentiable properties, derivative-free optimization [Yu et al., 2016; Hu et al., 2017] is usually applied to solve it. For example, a tree-structure based Bayesian optimization (SMAC) [Hutter et al., 2011] is employed on AutoWEKA [Thornton et al., 2013] and AutoSKLEARN [Feurer et al., 2015], the popular open-source tools. Derivative-free optimization explores search space by sampling and evaluating. But the high time-cost restrains the total number of evaluations on AutoML. With the limited trials, the performance of derivative-free optimization is very sensitive to search space. However, in above formulation, the search space \( \Delta = \Delta_1 \times \Delta_2 \times \ldots \times \Delta_K \). Obviously, \( \Delta \) is redundant, because the best configuration is only relevant to the hyper-parameter space of the best algorithm.

Hence, we consider an easier formulation, i.e., optimizing hyper-parameters of algorithms separately:

\[
C_{i}^{\delta_i} = \arg\max_{C_i \in C} \frac{1}{k} \sum_{j=1}^{k} f\left(C_{i}^{\delta_i}, D_{train}^{j}, D_{valid}^{j}\right),
\]

where \( \delta_i^* = \arg\max_{\delta_i \in \Delta_i} \frac{1}{k} \sum_{j=1}^{k} f\left(C_{i}^{\delta_i}, D_{train}^{j}, D_{valid}^{j}\right). \) (2)

The hyper-parameter processes can be seen as arms. The algorithm selection level is a multi-armed bandit problem. The bandit is a classical formulation of the resource allocation problem. In [Felício et al., 2017], the authors formulated the cold-start user recommendation as a multi-armed bandit problem, which user information was unavailable at the beginning. The feedbacks of users has to be obtained by trials. In this situation, the bandit concerns more about the average feedback of arms. In [Cicirello and Smith, 2005], the authors proposed the max K-armed bandit, which focused on the maximum feedback of trials. But it assumed that the reward distribution was a Gaussian distribution, and it was designed for the heuristic search, in which more than one arms can be selected at a trial step.

In this paper, we customize the extreme-region UCB (ER-UCB) bandit for AutoML problems.

3 Extreme-Region UCB Bandit

In this section, we present details of the ER-UCB: the bandit formulation for AutoML, the deduction of the ER-UCB strategy and the theoretical analysis on the ER-UCB strategy.

3.1 Bandit Formulation for AutoML

In the classical multi-armed bandit, feedbacks of an arm obey an underlying distribution. In this paper, we employ the random search on the hyper-parameter optimization. A trial in a model \( C_i \) is uniformly sampling hyper-parameters from \( \Delta_i \), and its performance is the feedback of this trial. Thus, \( X_i \sim D_i (\mu_{X_i}, \sigma_{X_i}^2) \), where \( X_i \) denote a feedback of a trial on \( C_i \), and \( D_i (\mu_{X_i}, \sigma_{X_i}^2) \) is the underlying performance distribution of \( C_i \). Because of the random search, \( D_i \) is fixed. With \( K \) algorithm candidates, let \( D = \{ D_1, D_2, \ldots, D_K \} \) denote the performance distribution set. The \( K \)-armed bandit formulation for AutoML is: at the \( t \)-th trial, the \( C_{i_t} \), selected from \( K \) algorithm candidates, and get a feedback \( X_{i_t} \), independently from \( D_{i_t} \).

3.2 Deduction

In AutoML tasks, the selected algorithm is required to have maximum performances. For this requirement, we present the extreme-region target for the proposed bandit. Then, we show the deduction details of extreme-region UCB strategy.

**Extreme-Region Target**

The target of the hyper-parameter optimization is to find the hyper-parameters which have the maximum performance. In the bandit, with a fixed \( \epsilon \), we want the probability \( \Pr[X_i \geq \mu_{X_i} + \epsilon] \) as large as possible. With the Chebyshev inequality: \( \Pr[X_i \geq \mu_{X_i} + \epsilon] \leq \frac{\sigma_{X_i}^2}{\epsilon^2} \), let \( \frac{\sigma_{X_i}^2}{\epsilon^2} = \theta \),

\[
\Pr[X_i \geq \mu_{X_i} + \sqrt{\frac{1}{\theta} \sigma_{X_i}}] \leq \theta.
\] (3)

In other words, with the same fixed probability upper bound \( \theta \), the best arm selection is:

\[
I_t = \arg\max_{i \in \{1, 2, \ldots, K\}} \mu_{X_i} + \sqrt{\frac{1}{\theta} \sigma_{X_i}}.
\] (4)

With the given \( \mu_{X_i} \) and \( \sigma_{X_i} \), the ground-truth selection strategy is (4). But, when facing the unknown distributions, we have to estimate the expectation and variance based on the observations. With the Markov inequality, it is easy to relate the expectation \( \mu_{X_i} \) with its estimation. But for variance, it is hard to find the relationship. With the variance definition:

\[
E[X_i^2] = \sigma_{X_i}^2 + \mu_{X_i}^2.
\] (5)

Because \( E[X_i^2] \) is the expectation of the random variable \( X_i^2 \). The Markov inequality can be applied to it easily. And \( E[X_i^2] \) can partly represent \( \sigma_{X_i}^2 \) according to (5). Thus, we try to replace \( \sigma_{X_i} \), with \( \sqrt{E[X_i^2]} \):

\[
I_t = \arg\max_{i \in \{1, 2, \ldots, K\}} \mu_{X_i} + \sqrt{\frac{1}{\theta} E[X_i^2]}. \] (6)
Comparing with (4), (6) magnifies the effect of expectation item on selection strategy. To tackle this issue, we introduce a hyper-parameter $\beta \geq 0$, and construct a new random variable $Y_i = X_i - \beta$. Furthermore, let $\mu_{Y_i} = \mathbb{E}[Y_i]$, $Z_i = (X_i - \beta)^2$ and $\mu_{Z_i} = \mathbb{E}[Z_i]$. Thus, the extreme-region target is:

$$I_i = \arg\max_{i \in \{1, 2, \ldots, K\}} \mu_{Y_i} + \sqrt{\frac{1}{\theta} \mu_{Z_i}}. \quad (7)$$

We prove that it can reduce the effect of expectation on algorithm selection by introducing $\beta$.

**Proof.** According to definitions of $X_i$, $Y_i$ and $Z_i$,

$$\mu_{Y_i} + \sqrt{\frac{1}{\theta} \mu_{Z_i}} = \mu_{X_i} - \beta + \sqrt{\frac{1}{\theta} \left( \sigma^2_{X_i} + (\mu_{X_i} - \beta)^2 \right)}.$$

Comparing with (4), because of $\beta \geq 0$, the item of expectation is reduced, but the item of variance stays the same. It concludes the proof. \hfill \Box

**Extreme-Region UCB Strategy**

We apply the upper confidence bound (UCB) strategy on the extreme-region target. In this paper, we assume that the random variables satisfy the following moment condition. There exists a convex function $\psi$ on the reals, for all $\lambda \leq 0$,

$$\ln \mathbb{E} e^{\lambda |X - \mathbb{E}[X]|} \geq \psi(\lambda). \quad (9)$$

If we let $X \in [0, 1]$ and $\psi(\lambda) = \frac{\lambda^2}{8}$, (9) is known as Hoeffding’s lemma. We apply this assumption to construct an upper bound for the estimated expectations at some fixed confidence level. Let $\psi^*$ denote the Legendre-Fenchel transform of $\psi$. With $s$ observations of $X_i$, let $\hat{\mu}_{Y_i} = \frac{1}{s} \sum_{t=1}^{s} Y_{i,t}$ and $\hat{\mu}_{Z_i} = \frac{1}{s} \sum_{t=1}^{s} Z_{i,t}$ denote the estimated expectations of $Y_i$ and $Z_i$. Only for $Y_i$ with a fixed $\epsilon_Y$, using the Markov inequality:

$$\Pr \left[ \mu_{Y_i} \geq \hat{\mu}_{Y_i} + \epsilon_Y \right] \leq e^{-s\psi^*(\epsilon_Y)}. \quad (10)$$

The same deduction for $Z_i$, and $f(x) = \sqrt{x}$ is a monotonically increasing function:

$$\Pr \left[ \frac{1}{\theta} \mu_{Z_i} \geq \frac{1}{\theta} \left( \mu_{Z_i}^{\psi} + \epsilon_Z \right) \right] \leq e^{-s\psi^*(\epsilon_Z)}. \quad (11)$$

Because $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$, and let $\epsilon_Y = \epsilon_Z = \epsilon$. With the union bound, we combine $Y_i$ and $Z_i$ as follows:

$$\Pr \left[ \mu_{Y_i} + \sqrt{\frac{1}{\theta} \mu_{Z_i}} \geq \mu_{Y_i}^{\psi} + \frac{1}{\theta} \mu_{Z_i}^{\psi} + \epsilon + \frac{1}{\theta} \epsilon \right] \leq 2e^{-s\psi^*(\epsilon)}. \quad (12)$$

Let $2e^{-s\psi^*(\epsilon)} = \delta$. With the probability at least $1 - \delta$,

$$\hat{\mu}_{Y_i} + \frac{1}{\theta} \hat{\mu}_{Z_i} + (\psi^*)^{-1} \left( \frac{1}{s} \ln \frac{2}{\delta} \right)$$

$$+ \frac{1}{\theta} (\psi^*)^{-1} \left( \frac{1}{s} \ln \frac{2}{\delta} \right) > \mu_{Y_i} + \sqrt{\frac{1}{\theta} \mu_{Z_i}}. \quad (13)$$

**Algorithm 1 Extreme-region UCB Bandit**

**Input:**

- $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K$: $K$ model candidates;
- $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_K$: hyper-parameter spaces of models;
- $\gamma, \theta, \beta$: hyper-parameters;
- $n$: trial budget;
- $\mathcal{D}_{train}$: train dataset of task;
- Sample$_{\delta_1}$: uniform sample sub-procedure;
- Evaluate: evaluation sub-procedure.

**Procedure:**

1: for $t = 1$ to $K$
2: \hspace{1em} $X_i = \text{Evaluate} \left( \mathcal{C}_i, \text{Sample}_{\delta_1} (\mathcal{D}_i), \mathcal{D}_{\text{train}} \right)$
3: \hspace{1em} $\hat{\mu}_{Y_i} = X_i - \beta, \hat{\mu}_{Z_i} = (X_i - \beta)^2, T_i = 1$
4: end for
5: for $t = K + 1$ to $n$
6: \hspace{1em} get index $I_t$ according to (17)
7: \hspace{1em} $X_{I_t} = \text{Evaluate} \left( \mathcal{C}_{I_t}, \text{Sample}_{\delta_1} (\mathcal{D}_{I_t}), \mathcal{D}_{\text{train}} \right)$
8: \hspace{1em} $\hat{\mu}_{Y_{I_t}} = \frac{T_{I_t}(t) \hat{\mu}_{Y_{I_t}} + X_{I_t} - \beta}{T_{I_t}(t)+1}$
9: \hspace{1em} $\hat{\mu}_{Z_{I_t}} = \frac{T_{I_t}(t) \hat{\mu}_{Z_{I_t}} + (X_{I_t} - \beta)^2}{T_{I_t}(t)+1}$
10: \hspace{1em} $T_{I_t} = T_{I_t} + 1$
11: end for
12: return the hyper-parameters with the best $X$.

Within total $s$ trials, let $T_i(s) = \sum_{t=1}^{s} 1_{I_t=i}$ denote the number that the $i$-th arm is selected, and $\frac{2}{\theta} = t^\alpha$. ($\alpha, \psi$)-ER-UCB strategy is:

$$I_t = \arg\max_{i \in \{1, 2, \ldots, K\}} \Omega_i (Y_i, Z_i, T_i (t)) + \Psi_i (T_i (t), t),$$

where

$$\Omega_i = \mu_{Y_i}^{\psi} + \frac{1}{\theta} \mu_{Z_i}^{\psi},$$

$$\Psi_i = (\psi^*)^{-1} \left( \alpha \ln t \right) + \frac{1}{\theta} (\psi^*)^{-1} \left( \alpha \ln t \right) \frac{2 \ln t}{T_i(t)}.$$

(14)

$\Omega_i$ and $\Psi_i$ are the exploitation and exploration items. With Hoeffding’s lemma, taking $\psi(\lambda) = \frac{\lambda^2}{8}$, then, $\psi^*(\epsilon) = 2\epsilon^2$. And let $\alpha = 4$. The exploration can be re-written as:

$$\Psi'_i = \sqrt{\frac{2 \ln t}{T_i(t)}} \left( \alpha \frac{\ln t}{T_i(t)} \right). \quad (15)$$

Thus, the Hoeffding’s ER-UCB strategy is:

$$I_t = \arg\max_{i \in \{1, 2, \ldots, K\}} \Omega_i (Y_i, Z_i, T_i (t)) + \Psi'_i (T_i (t), t). \quad (16)$$

Because $X_i \in [0, 1]$ on AutoML, the exploitation item is often much smaller than the exploration item. To further exploration and exploitation trade-off, we introduce a hyper-parameter $\gamma$. The practical Hoeffding’s ER-UCB strategy is:

$$I_t = \arg\max_{i \in \{1, 2, \ldots, K\}} \gamma \Omega_i (Y_i, Z_i, T_i (t)) + \Psi'_i (T_i (t), t). \quad (17)$$
The cascaded algorithm selection and hyper-parameter optimization with ER-UCB bandit is presented at Algorithm 1. Line 2 and 7 are the procedures of uniformly sampling hyper-parameters for the selected algorithm and obtaining the feedbacks. Line 1 to 4 are the initialization steps. In the main loop (line 5 to 10), the algorithm is selected by the ER-UCB strategy (line 6). Line 7 to 9 are the procedures for updating the exploitation item for the selected algorithm.

We have to discuss the hyper-parameters, i.e., $\theta$, $\gamma$ and $\beta$ for the ER-UCB bandit. $\theta$ is employed to control the space size of the extreme region. It is usually a small real number, e.g., 0.1 or 0.01. $\gamma$ is the exploration-and-exploitation trade-off hyper-parameter. In AutoML tasks, $\gamma$ is used to magnify the exploitation item. Thus, it is usually a big number such as $10$ or $20$. $\beta$ is applied to reduce the impact of expectation item in the selection strategy. It should be tuned according to tasks. In experiments, we will investigate them empirically.

### 3.3 Theoretical Analysis

We present the analysis of the upper bound for $(\alpha, \psi)$-ER-UCB strategy (14) and the Hoeffding’s ER-UCB strategy (15) on the extreme-region regret. For the arbitrary arm $i$ and a fixed $p$, we define $\Pr[X_i \geq \rho] = p_i$. Thus, $p^* = \arg\max_{i \in \{1, 2, \ldots, K\}} p_i$. According to (7), let $i^* = \arg\max_{i \in \{1, 2, \ldots, K\}} \mu_Y + \sqrt{\frac{2\mu^2}{n}}$, thus $\mu^* + \sqrt{\frac{1}{2\mu^2}}$ is a distribution of arbitrary arm satisfy (9). With fixed two hyper-parameters, we can easily prove the extreme-region regret of the Hoeffding’s ER-UCB strategy:

**Corollary 1 (Regret of Hoeffding’s ER-UCB).** Assume the feedback distribution of arbitrary arm satisfy (9). With $\alpha > 2$, Hoeffding’s ER-UCB satisfies:

$$R_n \leq \sum_{i: \Gamma_i \geq \theta} \Theta_i \left( \frac{8\alpha \ln n}{\Gamma^2_i / (1 - \theta^{-1})^4} + \frac{\alpha + 2}{\alpha - 2} \right).$$

According to the theoretical analysis, the ER-UCB bandit has $O(K \ln n)$ upper bound on the extreme-region regret.

### 4 Experiments

In the experiment section, we empirically investigate the effectiveness of the ER-UCB bandit on some synthetic and real-world AutoML tasks. Some state-of-the-art bandit strategies are selected as the compared methods, including the classical UCB (C-UCB) [Bubeck et al., 2012], $\epsilon$-greedy [Sutton and Barto, 2018], softmax strategy [Taki and Palm, 2011] and random strategy which allocates the budget by selecting arms randomly. In addition, we apply the random search on the joint hyper-parameter spaces of all algorithms (Joint) to compare with the cascaded hyper-parameter optimization.

#### 4.1 Synthetic Problem

We construct a 7-armed bandit problem in this section. The feedbacks obey Gaussian distributions with different expectations and variances: $G_{1}$ $(0.84, 0.07^2)$, $G_{2}$ $(0.84, 0.01^2)$, $G_{3}$ $(0.85, 0.04^2)$, $G_{4}$ $(0.85, 0.02^2)$, $G_{5}$ $(0.88, 0.01^2)$, $G_{6}$ $(0.88, 0.02^2)$, $G_{7}$ $(0.89, 0.01^2)$). The best arm is not only related with the expectation, but also influenced by the variance. Obviously, it is more likely to obtain the best feedback by exploiting in $G_{1}$, in other words, $i^* = 1$. We study on the three hyper-parameters of ER-UCB firstly, and then compare the ER-UCB with other methods.

**Hyper-Parameter Study**

We investigate the $\theta$, $\gamma$ and $\beta$ for the ER-UCB. With fixed two of them, we study another one: with fixed $\gamma = 20$, $\beta = 0.85$, we study $\theta \in [0.0001, 0.5]$; with fixed $\gamma = 0.85$, $\theta = 0.01$, we study $\gamma \in [0, 50]$; with fixed $\theta = 0.01$, $\gamma = 20$, we study $\beta \in [0, 1.5]$. For every hyper-parameter, we evenly sample 1000 settings from the setting region. The core problem we care about is how the methods allocate budget to arms. Let $R_n^{\text{exp}} = T_i(n)$ define the exploitation rate for arm $i$. Large $R_n^{\text{exp}}$ means the large number of trials that the arm $i$ is selected. The trial budget is set as 1000. The experiment for every hyper-parameter setting is repeated for 3 times independently, and the average results are presented.

Figure 1:a.1, 2 and 3 show the study results of $\theta$, $\gamma$ and $\beta$. The arm 1 is the best selection. Thus, the larger $R_n^{\text{exp}}$ the better. For $\theta$ (Figure 1:a.1), the green line ($R_n^{\text{exp}}$) is approaching 1 when $\theta$ nears by 0. In practice, $\theta$ should be set as a small value. For $\gamma$ (Figure 1:a.2), when $\gamma$ is small, the exploitation rates of arms are similar. And the green line is increasing during $\gamma$ is increasing. It means that the small $\gamma$ encourages exploration and the large $\gamma$ encourages exploitation according to the observations. For $\beta$ (Figure 1:a.3), the exploitation rates are sensitive to $\beta$ when $\beta$ is around the expectations of reward distributions. Thus, $\beta$ should be carefully tuned according to different tasks.
Investigation with Compared Methods

According to the hyper-parameter study results of the ER-UCB, we set $\theta = 0.01$, $\gamma = 20$, $\beta = 0.85$, and compare it with the C-UCB, $\epsilon$-greedy ($\epsilon = 0.1$), Softmax strategy ($\tau = 0.1$) and random selection strategy. The trial budget is 1000. Every experiment is repeated for 3 times independently. The average performances are presented in Table 1. Table 1 shows that the ER-UCB outperforms the compared methods. Furthermore, the ER-UCB can find the best arm (arm 1) and allocate most of budget to it ($\max_i R^{xi}_{\text{i}} = 1.1, 1$) and average $R^{xi}_{\text{max}}$ is 0.9. Because the C-UCB depends only on mean observations to make decisions. It wrongly allocates budget to arm 7 ($\max_i R^{xi}_{\text{i}} = 7, 7, 7$). The $R^{xi}_{\text{i}}$ of $\epsilon$-greedy is very unstable. It means $\epsilon$-greedy can’t find the best arm effectively. In general, the ER-UCB can effectively discover the best-arm and reasonably allocate budget to exploration and exploitation in this synthetic problem.

4.2 Real-World AutoML Tasks

We apply the ER-UCB to solve the real-world classification tasks. We select 10 frequently-used algorithms as the candidates from SKLEARN [Pedregosa et al., 2011], including DecisionTree (DT), AdaBoost (Ada), QuadraticDiscriminantAnalysis (QDA), GaussianNB (GNB), BernoulliNB (BNB), K-Neighbors (KN), ExtraTree (ET), PassiveAggressive (PA), RandomForest (RF) and SGD. And 12 classification datasets from UCI are selected as AutoML tasks. The evaluation criterion of each configuration is the accuracy score. The compared methods are C-UCB, $\epsilon$-greedy ($\epsilon = 0.1$), Softmax strategy ($\tau = 0.1$), random strategy and Joint. The trial budget is 1000. We set $\theta = 0.01$, $\gamma = 20$ for the ER-UCB on all datasets. The $\beta$ is set according to the tasks, and showed in Table 2. For each method and each dataset, we run every experiment 3 times independently, and the average performances of our experiment are presented. In addition, we apply the random search with 1000 trials to explore on every algorithm candidate. According to (2), we can find out the best ground-truth algorithm for the datasets.

The average performances of the compared methods on all 12 datasets are presented in Table 2. From Table 2, we can get the following empirical conclusions:

- "No free lunch" has been proved again in those experiments. The best performance algorithms are different in different datasets. Particularly, tree-based ensemble algorithms, e.g., AdaBoost, RandomForest, etc, show the outstanding performance in most of the datasets. It indicates that the algorithm selection is necessary for making search hyper-parameters easier.

- The cascaded algorithm selection and hyper-parameter optimization are necessary for making the search problem easier to solve. Comparing the random strategy with the Joint, the random strategy beats the Joint on most of the datasets (8/12). It indicates that the large search space provides more difficult for optimization.

- It will mislead the strategy to select wrong algorithms only according to the average performance. In Table 2, the random strategy is not always bad in datasets. The strategies, such as C-UCB, $\epsilon$-greedy and Softmax, which focus on the average performance are easy to select wrong algorithms which average performances are good.

- The proposed ER-UCB bandit strategy can effectively find out the best performance algorithm (B. Alg. is the ground-truth algorithm on 9/12 datasets), and reasonably allocate the trial budget to the best algorithm (ER-UCB gets the highest $R^{xi}_{\text{i}}$ on 12/12 datasets).
Dataset Methods V-Eval $R_{\text{eval}}$ B. Alg. T-Eval Dataset Methods V-Eval $R_{\text{eval}}$ B. Alg. T-Eval

Balance (SGD) $\beta = 0.5$

<table>
<thead>
<tr>
<th>Method</th>
<th>V-Eval</th>
<th>$R_{\text{eval}}$</th>
<th>B. Alg.</th>
<th>T-Eval</th>
</tr>
</thead>
<tbody>
<tr>
<td>ER-UCB</td>
<td>.9025</td>
<td>.5677</td>
<td>SGD</td>
<td>.9074</td>
</tr>
<tr>
<td>C-UCB</td>
<td>.8924</td>
<td>.1067</td>
<td>PA</td>
<td>.8339</td>
</tr>
<tr>
<td>$\epsilon$-Greedy</td>
<td>.8931</td>
<td>.0693</td>
<td>SGD</td>
<td>.8227</td>
</tr>
<tr>
<td>Softmax</td>
<td>.9004</td>
<td>.1287</td>
<td>SGD</td>
<td>.8809</td>
</tr>
<tr>
<td>Random</td>
<td>.8978</td>
<td>.1097</td>
<td>SGD</td>
<td>.8597</td>
</tr>
<tr>
<td>Joint</td>
<td>.8978</td>
<td>-</td>
<td>SGD</td>
<td>.8994</td>
</tr>
</tbody>
</table>

Chess (Ada) $\beta = 0.5$

<table>
<thead>
<tr>
<th>Method</th>
<th>V-Eval</th>
<th>$R_{\text{eval}}$</th>
<th>B. Alg.</th>
<th>T-Eval</th>
</tr>
</thead>
<tbody>
<tr>
<td>ER-UCB</td>
<td>.9557</td>
<td>.6137</td>
<td>Ada</td>
<td>.8515</td>
</tr>
<tr>
<td>C-UCB</td>
<td>.9414</td>
<td>.1693</td>
<td>Ada</td>
<td>.8860</td>
</tr>
<tr>
<td>$\epsilon$-Greedy</td>
<td>.9492</td>
<td>.3593</td>
<td>Ada</td>
<td>.7510</td>
</tr>
<tr>
<td>Softmax</td>
<td>.9414</td>
<td>.1290</td>
<td>PA</td>
<td>.8229</td>
</tr>
<tr>
<td>Random</td>
<td>.9464</td>
<td>.1070</td>
<td>SGD</td>
<td>.6999</td>
</tr>
<tr>
<td>Joint</td>
<td>.9457</td>
<td>-</td>
<td>SGD</td>
<td>.8479</td>
</tr>
</tbody>
</table>

Ecoli (RF) $\beta = 0.5$

<table>
<thead>
<tr>
<th>Method</th>
<th>V-Eval</th>
<th>$R_{\text{eval}}$</th>
<th>B. Alg.</th>
<th>T-Eval</th>
</tr>
</thead>
<tbody>
<tr>
<td>ER-UCB</td>
<td>.8763</td>
<td>.7013</td>
<td>RF</td>
<td>.8904</td>
</tr>
<tr>
<td>C-UCB</td>
<td>.8745</td>
<td>.2440</td>
<td>RF</td>
<td>.8333</td>
</tr>
<tr>
<td>$\epsilon$-Greedy</td>
<td>.8129</td>
<td>.3023</td>
<td>RF</td>
<td>.8809</td>
</tr>
<tr>
<td>Softmax</td>
<td>.8728</td>
<td>.1630</td>
<td>RF</td>
<td>.8809</td>
</tr>
<tr>
<td>Random</td>
<td>.8695</td>
<td>.1037</td>
<td>RF</td>
<td>.8762</td>
</tr>
<tr>
<td>Joint</td>
<td>.8549</td>
<td>-</td>
<td>RF</td>
<td>.8714</td>
</tr>
</tbody>
</table>

Messider (SGD) $\beta = 0.5$

<table>
<thead>
<tr>
<th>Method</th>
<th>V-Eval</th>
<th>$R_{\text{eval}}$</th>
<th>B. Alg.</th>
<th>T-Eval</th>
</tr>
</thead>
<tbody>
<tr>
<td>ER-UCB</td>
<td>.7431</td>
<td>.5203</td>
<td>SGD</td>
<td>.7330</td>
</tr>
<tr>
<td>C-UCB</td>
<td>.7297</td>
<td>.1133</td>
<td>SGD</td>
<td>.7272</td>
</tr>
<tr>
<td>$\epsilon$-Greedy</td>
<td>.7362</td>
<td>.0157</td>
<td>SGD</td>
<td>.7272</td>
</tr>
<tr>
<td>Softmax</td>
<td>.7402</td>
<td>.1163</td>
<td>SGD</td>
<td>.7604</td>
</tr>
<tr>
<td>Random</td>
<td>.7406</td>
<td>.1027</td>
<td>SGD</td>
<td>.7330</td>
</tr>
<tr>
<td>Joint</td>
<td>.7399</td>
<td>-</td>
<td>SGD</td>
<td>.7316</td>
</tr>
</tbody>
</table>

Spambase (Ada) $\beta = 0.7$

<table>
<thead>
<tr>
<th>Method</th>
<th>V-Eval</th>
<th>$R_{\text{eval}}$</th>
<th>B. Alg.</th>
<th>T-Eval</th>
</tr>
</thead>
<tbody>
<tr>
<td>ER-UCB</td>
<td>.9328</td>
<td>.9853</td>
<td>Ada</td>
<td>.9449</td>
</tr>
<tr>
<td>C-UCB</td>
<td>.9298</td>
<td>.1757</td>
<td>Ada</td>
<td>.9457</td>
</tr>
<tr>
<td>$\epsilon$-Greedy</td>
<td>.9311</td>
<td>.7333</td>
<td>Ada</td>
<td>.9741</td>
</tr>
<tr>
<td>Softmax</td>
<td>.9298</td>
<td>.1253</td>
<td>Ada</td>
<td>.9471</td>
</tr>
<tr>
<td>Random</td>
<td>.9306</td>
<td>.1057</td>
<td>Ada</td>
<td>.9500</td>
</tr>
<tr>
<td>Joint</td>
<td>.9290</td>
<td>-</td>
<td>RF</td>
<td>.9428</td>
</tr>
</tbody>
</table>

WDBC (Ada) $\beta = 0.6$

<table>
<thead>
<tr>
<th>Method</th>
<th>V-Eval</th>
<th>$R_{\text{eval}}$</th>
<th>B. Alg.</th>
<th>T-Eval</th>
</tr>
</thead>
<tbody>
<tr>
<td>ER-UCB</td>
<td>.9823</td>
<td>.9506</td>
<td>Ada</td>
<td>.9681</td>
</tr>
<tr>
<td>C-UCB</td>
<td>.9808</td>
<td>.1397</td>
<td>Ada</td>
<td>.9710</td>
</tr>
<tr>
<td>$\epsilon$-Greedy</td>
<td>.9816</td>
<td>.8757</td>
<td>Ada</td>
<td>.9681</td>
</tr>
<tr>
<td>Softmax</td>
<td>.9794</td>
<td>.1207</td>
<td>Ada</td>
<td>.9739</td>
</tr>
<tr>
<td>Random</td>
<td>.9794</td>
<td>.1060</td>
<td>Ada</td>
<td>.9739</td>
</tr>
<tr>
<td>Joint</td>
<td>.9794</td>
<td>-</td>
<td>Ada</td>
<td>.9594</td>
</tr>
</tbody>
</table>

Table 2: The average performances on AutoML tasks including the best validation accuracy (V-Eval), the exploitation rate on the ground-truth best algorithm (R_{\text{eval}}), the best-selected algorithm (B. Alg.) and the test accuracy (T-Eval). The items under the dataset name are the ground-truth algorithm and the $\beta$ setting for the ER-UCB. The number in bold means the best performance in compared methods.

## 5 Conclusion

This paper proposes the extreme-region upper confidence bound (ER-UCB) bandit for the cascaded algorithm selection and hyper-parameter optimization. We employ the random search in the hyper-parameter optimization level. The level of algorithm selection is formulated as a multi-armed bandit problem. The bandit strategies are applied to allocate the limited search budget to the hyper-parameter optimization processes on algorithm candidates. However, the algorithm selection focuses on the algorithm with the maximum performance but not the average performance. To tackle this, we propose the extreme-region UCB (ER-UCB) strategy, which selects the arm with the largest extreme region of the underlying distribution. The theoretical study shows that the ER-UCB has $O(K \ln n)$ extreme-region regret upper bound, which has the same order with the classical UCB strategy. The experiments on synthetic and real-world AutoML problems empirically verify that the ER-UCB can precisely discover the algorithm with the best performance, and reasonably allocate the trial budget to the algorithm candidates.
References


