A Simple Dynamic Bandit Algorithm for Hyper-parameter Tuning

Xuedong Shang[1,2], Emilie Kaufmann,[1,2,3] Michal Valko[4,1]

[1]Inria SequeL [2]Univ. Lille [3]CNRS [4]DeepMind Paris

Problem and Objectives

We treat the **hyper-parameter tuning** problem for *supervised learning* tasks.

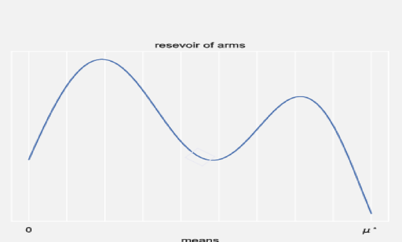
- global optimisation task: $\min\{f(\boldsymbol{\lambda}) : \boldsymbol{\lambda} \in \Omega\}$;
- $f(\boldsymbol{\lambda}) \triangleq \mathbb{E}\left[\ell\left(\mathbf{Y}, g_{\boldsymbol{\lambda}}^{(n)}(\mathbf{X})\right)\right]$ measures the generalization power;

Our contribution: a simple, robust, (almost) parameter-free bandit algorithm.

How and Why

How?

We see the problem as *best arm identification* in a *stochastic infinitely-armed bandit*: arms' means are drawn from some *reservoir distribution* ν_0.



In each round:
→ (optional) query a new arm from ν_0
→ sample an arm that was previously queried

Goal: output an arm with mean close to $\mu^\star$

`D-TTTS` ⇝ a dynamic algorithm built on `TTTS` [1]

Why?

→ `TTTS` is *anytime* for finitely-armed bandits
→ the flexibility of this Bayesian algorithm allows to propose a **dynamic** version for the infinite BAI
→ unlike previous approaches, `D-TTTS` **does not** need to fix the number of arms queried in advance, and naturally **adapts** to the difficulty of the task

`HPO` as a `BAI` problem

BAI	HPO
query ν_0	pick a new configuration $\boldsymbol{\lambda}$
sample an arm	train the classifier $g_{\boldsymbol{\lambda}}$
reward	cross-validation loss

In the Context of `BAI`...

- `Beta-Bernoulli` Bayesian bandit model
- a uniform prior over the mean of new arms

Posterior distribution on arm i at time t:

$$\texttt{Beta}(1 + S_{t,i}, N_{t,i} - S_{t,i} + 1).$$

`D-TTTS` principle: in each round, query a new arm endowed with a `Beta`(1,1) prior, *without sampling it*, and run `TTTS` on the new set of arms.

Implementation tricks

Binarization trick: When a reward $Y_{t,i} \in [0,1]$ is observed, the algorithm is updated with a fake binary reward $Y'_{t,i} \sim \texttt{Ber}(Y_{t,i}) \in \{0,1\}$.

Order statistic trick: with $\mathcal{L}_{t-1}$ the list of arms that have been effectively sampled at time t, we run `TTTS` on the set $\mathcal{L}_{t-1} \cup \{\mu_0\}$ where μ_0 is a pseudo-arm with posterior $\texttt{Beta}(t - |\mathcal{L}_{t-1}|, 1)$.

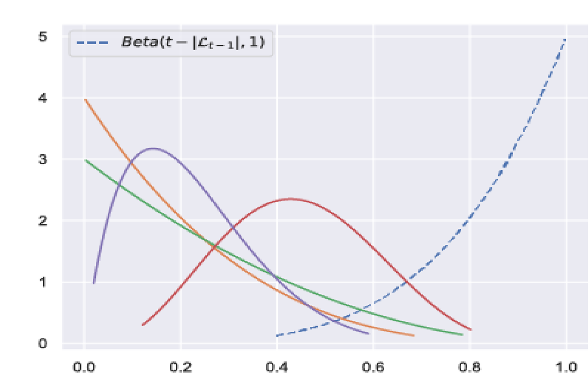


Figure: Posterior distributions of 4 arms and the pseudo-arm

Experimental Setting

Classifier	Hyper-parameter	Type	Bounds
`SVM`	C	$\mathbb{R}^+$	$[10^{-5}, 10^5]$
	γ	$\mathbb{R}^+$	$[10^{-5}, 10^5]$

Table: hyper-parameters to be tuned for `UCI` experiments.

Classifier	Hyper-parameter	Type	Bounds
`MLP`	`hidden_layer_size`	Integer	$[5, 50]$
	`alpha`	$\mathbb{R}^+$	$[0, 0.9]$
	`learning_rate_init`	$\mathbb{R}^+$	$[10^{-5}, 10^{-1}]$

Table: hyper-parameters to be tuned for `MNIST` experiments.

Sampling Rule

Input: β
Initialization: $\mu_1 \sim \nu_0$; $\mathcal{A} = \{\mu_1\}$; $m = 1$; $S_1, N_1 = 0$
while budget still available **do**
  $\mu_{m+1} \sim \nu_0$; $\mathcal{A} \leftarrow \mathcal{A} \cup \{\mu_{m+1}\}$
  $S_{m+1}, N_{m+1} \leftarrow 0$; $m \leftarrow m + 1$
  $\forall i \in \mathcal{A}, \theta_i \sim \texttt{Beta}(S_i + 1, N_i - S_i + 1)$
  $I^{(1)} = \arg\max_{i=0,\dots,m} \theta_i$
  if $U(\sim \mathcal{U}([0,1])) > \beta$ **then**
   while $I^{(2)} \neq I^{(1)}$ **do**
    $\forall i \in \mathcal{A}, \theta'_i \sim \texttt{Beta}(S_i + 1, N_i - S_i + 1)$
    $I^{(2)} \leftarrow \arg\max_{i=0,\dots,m} \theta'_i$
   end while
   $I^{(1)} \leftarrow I^{(2)}$
  end if
  $Y \leftarrow$ `evaluate arm` $I^{(1)}$; $X \sim \texttt{Ber}(Y)$
  $S_{I^{(1)}} \leftarrow S_{I^{(1)}} + X$; $N_{I^{(1)}} \leftarrow N_{I^{(1)}} + 1$
end while

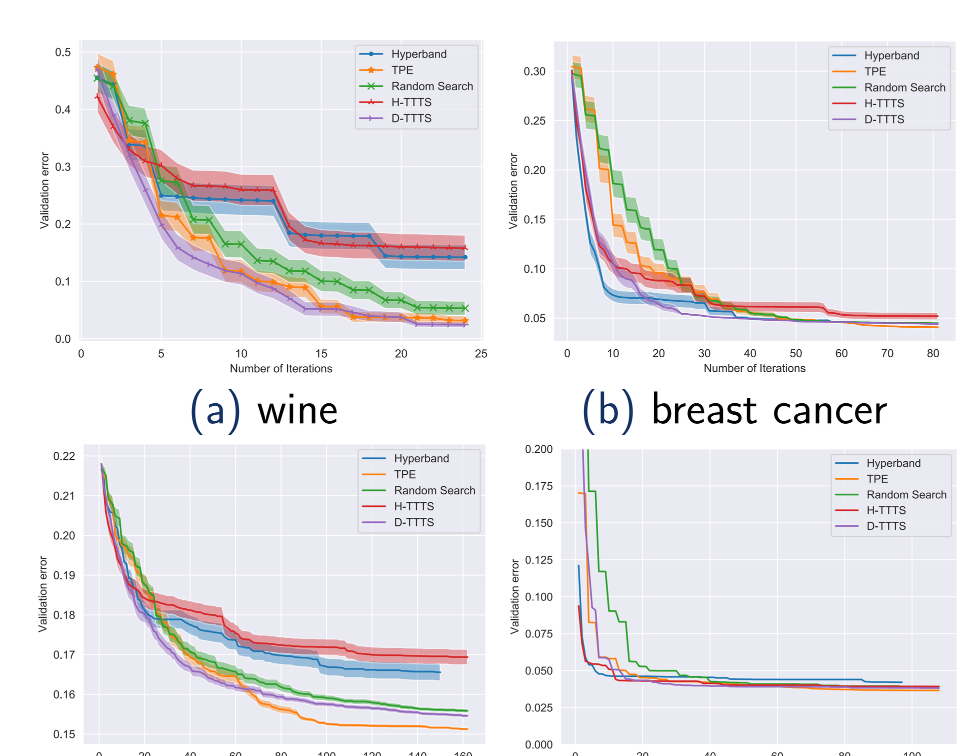
Recommendation Rule

We recommend the arm with the largest *posterior probability of being optimal*:

$$\bar{I}_n \triangleq \arg\max_{i \in \mathcal{A}} \Pi_n(\Theta_i),$$

where $\Theta_i \triangleq \{\boldsymbol{\theta} \in \Theta \,|\, \theta_i > \max_{j \neq i} \theta_j\}$.

Results for HPO



(a) wine (b) breast cancer (c) adult (d) `MNIST`

Understanding the Algorithm

Adaptation to the difficulty: for a 'difficult' reservoir, the pseudo-arm μ_0 is sampled more often (i.e. more arms are effectively sampled)

⇝ efficiently sampled arms for $\texttt{Beta}(\alpha, 1)$ reservoirs:

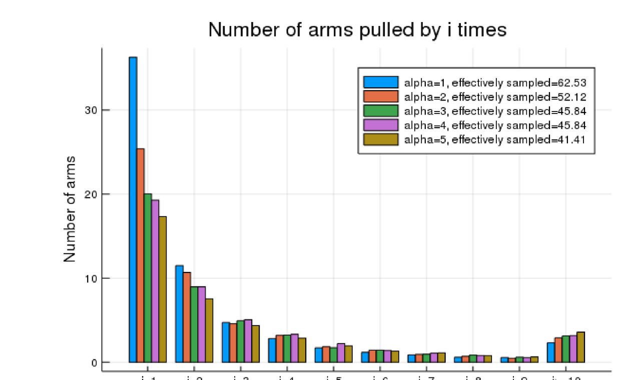


Figure: α

⇝ efficiently sampled arms for $\texttt{Beta}(1, \beta)$ reservoirs:

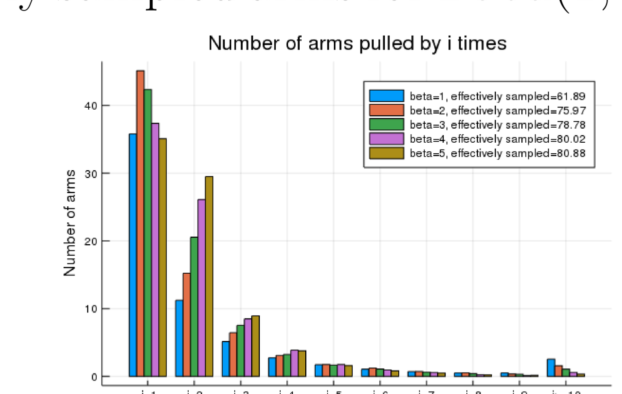


Figure: β

⇝ efficiently sampled arms for shifted `Beta` reservoirs:

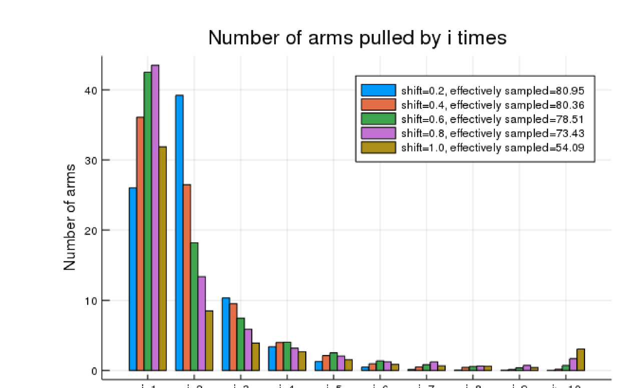


Figure: shift

References

[1] `TTTS`: D. Russo, *Simple Bayesian algorithms for best arm identification.* In CoLT, 2016.
[2] `Hyperband`: L. Li et al., *Hyperband: Bandit-based configuration evaluation for hyperparameter optimization.* In ICLR, 2017.