Efficient Kernel UCB for Contextual Bandits

AISTATS 2022

Houssam Zenati [1, 2] Alberto Bietti[3] Eustache Diemert [1]
Julien Mairal [2] Matthieu Martin [1] Pierre Gaillard [2]

[1]Criteo AI Lab

[2]Univ. Grenoble Alpes, Inria

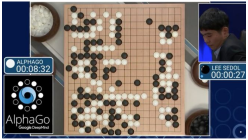
[3]Center for Data Science, New York University

March, 2022

Sequential learning with bandit feedback

Problem: In interactive systems, full information is not always available.

We wish to learn to **take actions** so as to **maximize a reward** signal...



... while **ensuring efficient learning** for large-scale real-world applications.

Contextual Bandits and Kernel UCB

At each round t, a bandit *agent* receives a context $x_t \in \mathcal{X}$ takes an *action* $a_t \in \mathcal{A}$ and receives a *reward* $r_t = r(x_t, a_t)$ from the environment. To evaluate its performance, we use the regret:

$$R_T := \mathbb{E}\left[\sum_{t=1}^{T} \max_{a \in \mathcal{A}} r(x_t, a) - \sum_{t=1}^{T} r_t\right]. \quad (1)$$

In the Kernel UCB setting, we assume

$$r_t = \langle \theta^*, \phi(x_t, a_t) \rangle + \varepsilon_t \,,$$

where $\theta^* \in \mathcal{H}$ the RKHS, ϕ is a feature map associated to $\mathcal{H}$ and the kernel k. The algorithm estimates θ^* with a regularized least squares:

$$\hat{\theta}_t \in \operatorname*{arg\,min}_{\theta \in \mathcal{H}} \left\{ \sum_{s=1}^{t} \left(\langle \theta, \phi(x_s, a_s) \rangle - r_s\right)^2 + \lambda \|\theta\|^2 \right\}. \quad (2)$$

Kernel UCB regret

K-UCB builds a confidence set around the estimate $\hat{\theta}$.

$$\mathcal{C}_t = \{\theta \in \mathcal{H} : \|\theta - \hat{\theta}_{t-1}\|_{V_{t-1}} \leq \beta\}.$$

where V_t is a data-dependent regularized covariance matrix. The algorithm takes the upper bound on C_t and then selects the best action:

$$\text{K-UCB}_t(a) = \max_{\theta \in \mathcal{C}_t} \langle \theta, \phi(x_t, a) \rangle, \quad a_t \in \arg\max_{a \in \mathcal{A}} \text{K-UCB}_t(a) \tag{3}$$

Proposition

The algorithm enjoys the following pseudo-regret bound:

$$R_T \lesssim \sqrt{T}\left(\|\theta^*\|\sqrt{\lambda d_{\text{eff}}} + d_{\text{eff}}\right),$$

The algorithm runs in $\mathcal{O}(T^2)$ space complexity and $\mathcal{O}(T^3)$ time complexity.

d_{eff} is the effective dimension associated λ and the kernel matrix K_t.

Incremental approximations of the RKHS

We use the KORS algorithm [Calandriello et al., 2017] to build incremental Nyström approximations of the RKHS.

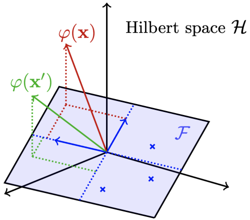


Figure: Nyström projection of a feature map φ

Proposition

The sequence of dictionaries $\mathcal{Z}_1 \subset \mathcal{Z}_2 \subset \dots \mathcal{Z}_T$ learned by KORS with parameters $\mu > 0$ satisfies with high probability that $|\mathcal{Z}_t| \approx d_{\text{eff}}(\mu, T)$.

Efficient Kernel UCB regret bound

EK-UCB estimates the parameter θ^* with a projected parameter $\tilde{\theta}_t$ to build a confidence set $\tilde{\mathcal{C}}_t$.

Theorem

Writing $m := |\mathcal{Z}_T|$, when choosing $\mu = \lambda$ we have $m \lesssim d_{\text{eff}}$ and the regret of the EK-UCB algorithm matches

$$R_T \lesssim \sqrt{T}\left(\|\theta^*\|\sqrt{\lambda d_{\text{eff}}} + d_{\text{eff}}\right).$$

The algorithm runs in $O(Tm)$ space complexity and $O(Tm^2)$ time complexity.

Related Work

BKB Calandriello et al. [2019] and BBKB Calandriello et al. [2020] recompute the Nyström dictionary to obtain efficient algorithms in the non contextual setting.

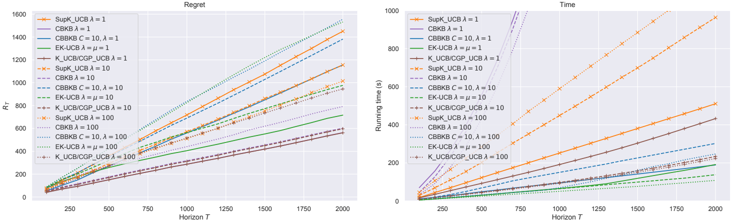
Algorithm	Space	Time Complexity
CGP-UCB [Krause and Ong, 2011]	$\mathcal{O}(T^2)$	$\mathcal{O}(CT^3)$
SupKernelUCB [Valko et al., 2013]	$\mathcal{O}(T^2)$	$\mathcal{O}(CT^3)$
K-UCB (ours)	$\mathcal{O}(T^2)$	$\mathcal{O}(CT^3)$
C-BKB [Calandriello et al., 2019]	$\mathcal{O}\left(Td_{\mathrm{eff}}\right)$	$\mathcal{O}\left(T^2 d_{\mathrm{eff}}^2 + CTd_{\mathrm{eff}}^2\right)$
C-BBKB [Calandriello et al., 2020]	$\mathcal{O}\left(Td_{\mathrm{eff}}\right)$	$\mathcal{O}\left(Td_{\mathrm{eff}}^3 + CTd_{\mathrm{eff}}^2\right)$
EK-UCB (ours)	$\mathcal{O}\left(Td_{\mathrm{eff}}\right)$	$\mathcal{O}(CTd_{\mathrm{eff}}^2)$

Table: Comparison of regrets, space and time complexities

Here, C is a constant related to optimizing the UCB rule.

Numerical Experiments

We consider synthetic environments with contexts and compare to K-UCB, SupK-UCB and to works which focus on improving the $\mathcal{O}(T^3)$ time-complexity as CBKB and CBBKB.



- Smaller μ induce a better regret but a higher time complexity.
- CGP-UCB/K-UCB obtain best regret
- SupK-UCB performs poorly due to its over-exploring elimination strategy, even though it has a tighter regret.

Take Home Message

- The EK-UCB algorithm runs in $\mathcal{O}(Td_{\mathrm{eff}})$ space and $\mathcal{O}(CTd_{\mathrm{eff}}^2)$ time complexity, which significantly improves over the standard contextual kernel UCB method.
- The incremental projection updates are crucial to perform efficient approximations in the joint context-action space.
- A natural question is whether we may obtain algorithms with better regret guarantees similar to Valko et al. [2013] in the finite action case, while also achieving gains in computational efficiency as in our work.

References

Daniele Calandriello, Alessandro Lazaric, and Michal Valko. Efficient second-order online kernel learning with adaptive embedding. In *Adv. Neural Information Processing Systems (NIPS)*, 2017.

Daniele Calandriello, Luigi Carratino, Alessandro Lazaric, Michal Valko, and Lorenzo Rosasco. Gaussian process optimization with adaptive sketching: Scalable and no regret. In *Conference on Learning Theory (COLT)*, 2019.

Daniele Calandriello, Luigi Carratino, Alessandro Lazaric, Michal Valko, and Lorenzo Rosasco. Near-linear time Gaussian process optimization with adaptive batching and resparsification. In *International Conference on Machine Learning (ICML)*, 2020.

Andreas Krause and Cheng S Ong. Contextual gaussian process bandit optimization. In *Adv. Neural Information Processing Systems (NIPS)*, 2011.

Michal Valko, Nathan Korda, Rémi Munos, Ilias Flaounas, and Nello Cristianini. Finite-time analysis of kernelised contextual bandits. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2013.