Cold Start SBL - Bypass

Zikun Ye

April 2021

1 Algorithm

$a \in [K]$ is the index of ads. $\hat{c}_{ia}^{m}$ is the predicted CTR of context i and ad a at the epoch m, which is computed by offline regression model $\hat{h}_m(i,a) = \hat{c}_{ia}^{m}$.

Shadow Bidding with Learning - FAst Least-squares-regression-oracle CONtextual bandits (SBL - FALCON)

Parameters: Epoch schedule $0 = \tau_0 < \tau_1 < \dots$. Cold start value coefficient $\boldsymbol{\beta}$. Target conversion parameter α. Confidence parameter δ.

Initialization: $\boldsymbol{\lambda}^0 = 0$, $t = 0$, $m = 0$

For $t = 1, 2, \dots, T$ **do**

Step 1: Observes the context $x_t = i$ at period t. Compute click through rate $\hat{c}_{ia}^{m}$ for each arm j. Let $\hat{a}_t = \arg\max_a \hat{c}_{ia}^{m}(b_a + \lambda_j^m)$. Define

$$p_t(a) = \begin{cases} \frac{1}{K+\gamma_m(\hat{c}_{i,\hat{a}_t}^{m}(b_{\hat{a}_t}+\lambda_{\hat{a}_t}^{m}) - \hat{c}_{ia}^{m}(b_a+\lambda_a^m))} & \forall a \neq \hat{a}_t \\ 1 - \sum_{a \neq \hat{a}_t} p_t(a) & a = \hat{a}_t \end{cases} \tag{1}$$

Then sample $a_t \sim p_t(\cdot)$ and observe the reward $v_t(a_t)$. where $v_t(a_t)$ represents whether the ad is clicked, the learning rate $\gamma_m = (1/30)\sqrt{(K\tau_{m-1})/\log(|\mathcal{F}|\tau_{m-1}/\delta)}$ (for epoch 1, $\gamma_1 = 1$).

Step 2: If $t = \tau_m$, we solve the following dual model and update $\boldsymbol{\lambda}^m$ by the sub-gradient descent algorithm. Assume updating λ without computation error.

$$\mathsf{OPT}^t = \min_{0 \le \lambda_a \le \beta_a} \sum_{i \in X} p_i^t \max_{a=1,2,\dots,K} \left(\hat{c}_{ia}^{t}(b_a + \lambda_a)\right) + \alpha \sum_{a=1}^{K} (\beta_a - \lambda_a), \tag{2}$$

Step 3: If $t = \tau_m$, update the epoch index $m = m + 1$. And update the offline regression model by computing $\hat{h}_m = \arg\min_{h \in \mathcal{F}} \sum_{t=1}^{\tau_m - 1} (h(x_t, a_t) - v_t(a_t))^2$.

Definitions

For the ease of the proof, we assume context distribution p_i^t is known and i.i.d over time periods. We make some definitions. Some definitions are defined in the main paper. $\mathcal{R}$ is the dual reward function.

$$V(p,\pi) = \mathbb{E}_{x\sim D_x}\frac{1}{p(\pi(x)|x)}$$

$$m(t) = \min\{m \in \mathbb{N} : t \leq \tau_m\}$$

$$\mathcal{V}_t(\pi) = \max_{1\leq m\leq m(t)-1}\{V(p_m,\pi)\}$$

$$\mathcal{R}(\pi) = \mathbb{E}_{x\sim D_x}[h^*(x,\pi(x))(b_{\pi(x)} + \lambda^*_{\pi(x)}) + \alpha\sum_{a=1}^{K}(\beta_a - \lambda^*_a)]$$

$$\pi^* = \arg\max_{\pi\in\Pi}\mathcal{R}(\pi)$$

$$\hat{\mathcal{R}}_t(\pi) = \mathbb{E}_{x\sim D_x}[\hat{h}_{m(t)}(x,\pi(x))(b_{\pi(x)} + \lambda^{m(t)}_{\pi(x)}) + \alpha\sum_{a=1}^{K}(\beta_a - \lambda^{m(t)}_a)]$$

$$\mathsf{Reg}(\pi) = \mathcal{R}(\pi^*) - \mathcal{R}(\pi)$$

$$\hat{\mathsf{Reg}}_t(\pi) = \hat{\mathcal{R}}_t(\pi^*) - \hat{\mathcal{R}}_t(\pi)$$

2 Regret Analysis

The key steps of proof is building the connection between our algorithm with implicit algorithm in [Agarwal et al., 2014], see the following lemmas.

Lemma 2.1 (Lemma 3 in [Simchi-Levi and Xu, 2020]) *Fix any epoch m. The action selection scheme $p_m(\cdot|\cdot)$ is a valid probability kernel. There exist a probability measure Q_m such that,*

$$\forall a, \forall x, p_m(a|x) = \sum_{\pi\in\Pi}\mathbb{I}_{\pi(x)=a}Q_m(\pi)$$

Lemma 2.1 is essentially built upon the Kolmogorov extension theorem with discrete actions $A = [K]$ and finite context space X. The universal policy space $\Pi = A^X$ contains all possible policies. $Q_m(\cdot)$ is a dense distribution over all policies in the universal policy space. By the Lemma 2.1, there is an unique "randomized policy" $p_m(\cdot|\cdot)$ over the actions in the action space A. It builds the connection between our algorithm and the algorithm ILOVETOCONBANDITS in [Agarwal et al., 2014], the key step of which is solving an "Optimization Problem" to balance the exploration and exploitation. The next three lemmas justify that randomized policy $p_m(\cdot|\cdot)$ implicitly is the solution of the "Optimization Problem".

Lemma 2.2 (Adapted from Lemma 4 in [Simchi-Levi and Xu, 2020]) *Fix any epoch m, for any round t in the epoch m, we have*

$$\mathbb{E}_{x_t,v_t,a_t}[v_t(\pi^*)(b_{a_{\pi^*}} + \lambda^*_{\pi^*}) - v_t(a_t)(b_{a_t} + \lambda^*_{a_t})|\mathcal{H}_{t-1}] = \sum_{\pi\in\Pi}Q_m(\pi)\mathsf{Reg}(\pi)$$

Proof of Lemma 2.2. By Lemma 2.1, we have

$$
\begin{aligned}
&\mathbb{E}_{x_t,v_t,a_t}[v_t(\pi^*)(b_{a_{\pi^*}}+\lambda^*_{\pi^*})-v_t(a_t)(b_{a_t}+\lambda^*_{a_t})|\mathcal{H}_{t-1}]\\
=&\mathbb{E}_{x_t,a_t}[h^*(x_t,\pi^*(x_t))(b_{a_{\pi^*}}+\lambda^*_{\pi^*})-h^*(x_t,a_t)(b_{a_t}+\lambda^*_{a_t})|\mathcal{H}_{t-1}]\\
=&\mathbb{E}_{x\sim D_x,a\sim p_m(\cdot|x)}[h^*(x,\pi^*(x))(b_{a_{\pi^*}}+\lambda^*_{\pi^*})-h^*(x,a)(b_a+\lambda^*_a)]\\
=&\mathbb{E}_x[\sum_{a\in[K]}\sum_{\pi\in\Pi}\mathbb{I}_{\pi(x)=a}Q_m(\pi)(h^*(x,\pi^*(x))(b_{a_{\pi^*}}+\lambda^*_{\pi^*})-h^*(x,a)(b_a+\lambda^*_a))]\\
=&\sum_{\pi\in\Pi}Q_m(\pi)\mathbb{E}_x[\sum_{a\in[K]}(h^*(x,\pi^*(x))(b_{a_{\pi^*}}+\lambda^*_{\pi^*})-h^*(x,a)(b_a+\lambda^*_a))]\\
=&\sum_{\pi\in\Pi}Q_m(\pi)\mathbb{E}_x[h^*(x,\pi^*(x))(b_{a_{\pi^*}}+\lambda^*_{\pi^*})-h^*(x,a)(b_a+\lambda^*_a)]\\
=&\sum_{\pi\in\Pi}Q_m(\pi)\mathsf{Reg}(\pi)
\end{aligned}
$$

The next two lemmas state key property of Q_m, which control the implicit regret and decisional divergence. Those two properties balance the exploration and exploitation, see the optimization problem in [Agarwal et al., 2014].

Lemma 2.3 (Adapted from Lemma 5 in [Simchi-Levi and Xu, 2020]) *Fix any epoch m, for any round t in the epoch m, we have*

$$\sum_{\pi\in\Pi}Q_m(\pi)\hat{\mathsf{Reg}}_t(\pi)\leq\frac{K}{\gamma_m}$$

Proof of Lemma 2.3. We have

$$
\begin{aligned}
&\sum_{\pi\in\Pi}Q_m(\pi)\hat{\mathsf{Reg}}_t(\pi)\\
&=\sum_{\pi\in\Pi}Q_m(\pi)\mathbb{E}_{x\sim D_x}[\hat{h}_m(x,\hat{a}_m(x))(b_{\hat{a}}+\lambda^{m(t)}_{\hat{a}})-\hat{h}_m(x,\pi(x))(b_{\pi(x)}+\lambda^{m(t)}_{\pi(x)})]\\
=&\mathbb{E}_{x\sim D_x}[\sum_{a\in[K]}\sum_{\pi\in\Pi}\mathbb{I}_{\pi(x)=a}Q_m(\pi)(\hat{h}_m(x,\hat{a}_m(x))(b_{\hat{a}}+\lambda^{m(t)}_{\hat{a}})-\hat{h}_m(x,a)(b_a+\lambda^{m(t)}_a))]\\
=&\mathbb{E}_{x\sim D_x}[\sum_{a\in[K]}p_m(a|x)(\hat{h}_m(x,\hat{a}_m(x))(b_{\hat{a}}+\lambda^{m(t)}_{\hat{a}})-\hat{h}_m(x,a)(b_a+\lambda^{m(t)}_a))]
\end{aligned}
$$

Given any context $x\in X$,

$$
\begin{aligned}
&\sum_{a\in[K]}p_m(a|x)(\hat{h}_m(x,\hat{a}_m(x))(b_{\hat{a}}+\lambda^{m(t)}_{\hat{a}})-\hat{h}_m(x,a)(b_a+\lambda^{m(t)}_a))\\
&=\sum_{a\neq\hat{a}_m(x)}\frac{\hat{h}_m(x,\hat{a}_m(x))(b_{\hat{a}}+\lambda^{m(t)}_{\hat{a}})-\hat{h}_m(x,a)(b_a+\lambda^{m(t)}_a)}{K+\gamma_m(\hat{h}_m(x,\hat{a}_m(x))(b_{\hat{a}}+\lambda^{m(t)}_{\hat{a}})-\hat{h}_m(x,a)(b_a+\lambda^{m(t)}_a))}\leq\frac{K-1}{\gamma_m}
\end{aligned}
$$

Lemma 2.4 (Adapted from Lemma 6 in [Simchi-Levi and Xu, 2020]) *Fix any epoch m, for any round t in the epoch m, for all policy π, we have*

$$V(p_m,\pi)\leq K+\gamma_m\hat{\mathsf{Reg}}_t(\pi)$$

Proof of Lemma 2.4. For any policy $\pi\in\Pi$, given any context $x\in X$,

$$
\frac{1}{p_m(\pi(x)|x)}\begin{cases}=K+\gamma_m(\hat{h}_m(x,\hat{a}_m(x))(b_{\hat{a}}+\lambda^{m(t)}_{\hat{a}})-\hat{h}_m(x,a)(b_a+\lambda^{m(t)}_a)) & if\pi(x)\neq\hat{a}_m(x)\\ \leq\frac{1}{1/K}\leq K+\gamma_m(\hat{h}_m(x,\hat{a}_m(x))(b_{\hat{a}}+\lambda^{m(t)}_{\hat{a}})-\hat{h}_m(x,a)(b_a+\lambda^{m(t)}_a)) & if\pi(x)=\hat{a}_m(x)\end{cases}
$$

Thus

$$V(p_m,\pi)=\mathbb{E}_{x\sim D_x}[\frac{1}{p_m(\pi(x)|x)}]\leq K+\gamma_m\mathbb{E}_{x\sim D_x}[\hat{h}_m(x,\hat{a}_m(x))(b_{\hat{a}}+\lambda^{m(t)}_{\hat{a}})-\hat{h}_m(x,a)(b_a+\lambda^{m(t)}_a)]=K+\gamma_m\hat{\mathsf{Reg}}_t(\pi)$$

Lemma 2.5 (Adapted from Lemma 10 in [Simchi-Levi and Xu, 2020]) *With an epoch schedule such that $\tau_m \leq 2\tau_{m-1}$ for $m \geq 1$ and $\tau_1 = O(1)$. $b, \beta = O(1)$. For any $T \in \mathbb{N}$, with probability at least $1-\delta$, the accumulated dual regret of* SBL-FALCON *after T rounds is at most*

$$O(\sqrt{KT\log(|\mathcal{F}|T/\delta)})$$

This regret matches the lower bound in the [Agarwal et al., 2012] up to logarithmic factors. The proof can be found in [Simchi-Levi and Xu, 2020] and [Agarwal et al., 2014], and is omitted here. this lemma is a conjugate, in addition to the error of h offline regression, the error od the λ^m is not incorporated yet. i.e., the lemma holds when $\lambda_m = \lambda^*$

However, we aim at bounding the primal regret, which is $T \cdot \mathsf{OPT} - \mathbb{E}_{\mathcal{D},\pi}\left[\Gamma(\boldsymbol{V})\right]$,

$$\begin{aligned}
&T \cdot \mathsf{OPT} - \mathbb{E}_{\mathcal{D},\pi}\left[\Gamma(\boldsymbol{V})\right] \\
&= T \cdot \mathsf{OPT} - \mathbb{E}\left[\sum_{t=1}^{T} r_t(\pi)\right] \\
&\leq \left|T \cdot \mathsf{OPT} - \mathbb{E}\left[\sum_{t=1}^{T} \bar{r}_t(\pi^*)\right]\right| + \left|\mathbb{E}\left[\sum_{t=1}^{T} \bar{r}_t(\pi^*)\right] - \mathbb{E}\left[\sum_{t=1}^{T} \bar{r}_t(\pi)\right]\right| + \left|\mathbb{E}\left[\sum_{t=1}^{T} \bar{r}_t(\pi)\right] - \mathbb{E}\left[\sum_{t=1}^{T} r_t(\pi)\right]\right| \\
&\leq TK^2 O(T^{-1/2}(\log T)^{1/3}K^{-5/3}) + O(\sqrt{KT}) + \left|\mathbb{E}\left[\sum_{t=1}^{T} \bar{r}_t(\pi)\right] - \mathbb{E}\left[\sum_{t=1}^{T} r_t(\pi)\right]\right|
\end{aligned}$$

where $r_t(\pi)$ is the primal reward process, which is defined as $r(x_t, a_t)$ in the main paper. $\bar{r}_t(\pi)$ is the dual reward process (i.e., $\mathbb{I}_{v_t(a_t)=1}(b_{a_t} + \lambda^*_{a_t}) + \alpha\sum_{a=1}^{K}(\beta_a - \lambda^*_a)$). By Lemma 2.5, we bound the dual reward regret $\left|\mathbb{E}\left[\sum_{t=1}^{T} \bar{r}_t(\pi^*)\right] - \mathbb{E}\left[\sum_{t=1}^{T} \bar{r}_t(\pi)\right]\right|$ by $O(\sqrt{KT})$ ignoring logarithmic factors. By Assumption 1 (revised) and Lemma 2, the primal dual gap of the optimal policy $\pi^* = \arg\max_{\pi\in\Pi} \mathcal{R}(\pi)$ is bounded by $O(T^{1/2}(\log T)^{1/3}K^{1/3})$.

The remaining is to show the primal dual gap of the reward $\left|\mathbb{E}\left[\sum_{t=1}^{T} \bar{r}_t(\pi)\right] - \mathbb{E}\left[\sum_{t=1}^{T} r_t(\pi)\right]\right|$ is bounded within $O(\sqrt{T})$ under our algorithm.

3 Literature Review

Our proposed algorithm builds upon ϵ-greedy contextual bandits and is related to four different bandit algorithms previously studied in the literature: (1) ϵ-greedy bandits. ϵ-greedy is one of the simplest exploration strategies and is equipped with an $O(T^{\frac{2}{3}})$ regret sutton2018reinforcement. One approach to deal with a complicated online learning problem is to reduce a bandit problem to supervised learning with simple exploration strategies, such as the epoch-greedy algorithm langford2007epoch. Although this approach has a sub-optimal regret upper bound of $O((K\log|\Pi|)^{\frac{1}{3}}T^{\frac{2}{3}})$, where Π is the explored policy set, it makes minimum changes to a practical online advertising system. (2) Linear contextual bandits with upper confident bound (UCB) exploration. Linear bandits are successful in both theory and practice chu2011contextual. However, in a real advertising system, the underlying models for predicting CTR and CVR are complicated neural networks, so the linear payoff assumption does not hold and LinearUCB can hardly be implemented. One remedy, called neural linear bandits, is to only explore the last linear layer of the neural networks [e.g.,][]riquelme2018deep. However, this method also makes significant changes to the online system and has no theoretical performance guarantee. (3) General contextual bandits. The recent advancement in general contextual bandits by [Agarwal et al., 2014] covers a wide range of contextual bandits with optimal regret guarantee of $O(\sqrt{KT\log(T|\Pi|)})$, however, it requires solving optimization problems with a given *empirical risk minimization oracle* to balance the exploration and exploitation trade-off, with computation complexity $O((KT)^{\frac{3}{2}})$, the computation time of which is intensive in the online advertising system without the online oracle to solve the embedded optimization. This makes it infeasible to implement this algorithm and its follow-ups in a real online advertising platform. (4) Bandits with knapsacks. This variant of bandits is first proposed by [Badanidiyuru et al., 2013]. Recent works extend this algorithm to a general contextual setting with concave objectives [e.g.,][]agrawal2016efficient. Our work is aligned with knapsack bandits on solving the primal-dual linear program to obtain the optimal policy. The regret of the bandit algorithms in this literature is benchmarked with the best policy in a policy set, and dependent on the cardinality of the policy set. Furthermore, the expected reward of the algorithm is estimated with *Inverse Propensity Score.* Though theoretically convenient, such approaches make it difficult to implement the bandit algorithm on a practical large-scale advertising platform with the neural-network-based estimation of CTR/CVR. Following the idea of [Foster et al., 2018] and [Foster and Rakhlin, 2020], our SBL algorithm reduces the problem into an ϵ-greedy bandit with the underlying prediction model, which could be quite general and takes the form of, e.g., linear regression, regression tree, and neural network. Our theoretical regret analysis with the underlying machine learning oracle being neural networks is based on the recent progress of *Neural Tangent Kernel* jacot2018neural,arora2019exact. Our main algorithmic contribution towards the MAB literature is that we bridge the gap between theory and practice by developing the SBL algorithm with provable performance guarantee and straightforward implementation on real online advertising platforms.

Following [Langford and Zhang, 2007], the majority of recent contextual bandit algorithms are based on a specific underlying optimization model, whose offline solution oracle is given. The recent progress on contextual bandits develops both computationally efficient and optimal learning algorithms based on the empirical risk minimization oracle agarwal2014taming, agrawal2016efficient which leads to a regret of $\tilde{O}(\sqrt{KT\log(|\Pi|)})$. Specifically, under a fixed policy set Π and the set $\mathcal{S}$ of context-loss pairs $(x, \ell) \in X \times \mathbb{R}^K$, the oracle returns the loss-minimization policy $\pi^* = \arg\min_{\pi\in\Pi} \sum_{(x,\ell)\in\mathcal{S}} \ell(\pi(x))$. Our problem setting is fundamentally different from those in the literature in two aspects. First, we explicitly construct a very general (and very large) policy set Π for developing our algorithm. Commonly used policy sets in practice such as linear predictors, regression trees, and neural networks are typically with an extremely large cardinality. For example, if the policy set Π is the collection of neural networks with fixed structure, depth, and width, even under the proper parameter discretization, the cardinality $|\Pi|$ grows exponentially with the number of parameters. Second, in our setting, a policy π contains two sequential decisions: the CTR prediction and ad allocation decisions, which brings challenges in both computation and analysis. This two-step procedure combining data-driven estimation and optimization model is widely used in practice-based research in the operations literature [e.g.,][]glaeser2019optimal, he2020customer, bimpikis2020managing. In this regard, the learning algorithms established in the literature [e.g.,][with a regret depending on $\log(|\Pi|)$ and benchmarked with the best policy in set Π]agarwal2014taming, agrawal2016efficient do not apply in our setting. Instead, the optimal policy we benchmark with is the (relaxed) optimal primal allocation policy (the primal integer decisions $\{0,1\}^K$ relaxed to $[0,1]^K$) with the optimal CTR prediction model.

Unlike the empirical regret of a policy computed via *Inverse Propensity Score* (IPS) in most of the existing contextual bandit literature agarwal2014taming, agrawal2016efficient, we adopt the *Direct Method* (DM), which uses empirical estimates from the CTR prediction model to evaluate the ad allocation policy. The IPS method gives an unbiased reward estimator of a policy and is, thus, widely used in regret analysis. However, IPS suffers from a high variance when the policy set is large and/or the past sample paths vary significantly, which is indeed the case of our real implementation on Platform O. However, the DM can be robustly implemented on the large-scale DSP and does not affect the current CTR prediction system thereof. As one may expect, under the DM, the total regret depends on the accuracy of the underlying prediction model. In our regret analysis, we demonstrate that as long as the ground-truth CTR can be well captured by the prediction model with a high probability, the SBL algorithm is asymptotically optimal. For more theoretical and computational comparisons between DM and ISP, we refer interested readers to, e.g., [Dudík et al., 2011, Foster et al., 2018, Foster and Rakhlin, 2020].

Notice that in running the algorithm, we are effectively solving

$$\mathsf{OPT}^t = \min_{0\le\lambda_j\le\beta_j} \sum_{i\in X} \hat{p}_i^t \max_{j=1,2,\ldots,K} \left(\hat{c}_{ij}^t(b_j+\lambda_j)\right) + \alpha \sum_{j=1}^{K} (\beta_j - \lambda_j), \tag{3}$$

where $\hat{c}_{ij}^t$ is the estimate of c_{ij} produced by the underlying prediction model prior to round t. $\hat{p}_i^t$ denotes the empirical distribution of contexts prior at round t. Before formally presenting our main regret analysis result, we address two basic issues regrading this formulation. First, we need to bound the gap between optimal empirical *primal* allocation and our optimal empirical *dual* allocation. By strong duality, this gap is induced by tie breaking in Steps 1 and 2 of the SBL algorithm. As we will show in Appendix **??**, adding an arbitrarily small perturbation to the CTR estimate $\hat{c}_{ij}^{t}$ will ensure that the tie-breaking in Step 1 will only induce an arbitrarily small additional regret. To bound the gap from tie breaking in Step 2, we make the following assumption.

References

[Agarwal et al., 2012] Agarwal, A., Dudík, M., Kale, S., Langford, J., and Schapire, R. (2012). Contextual bandit learning with predictable rewards. In Artificial Intelligence and Statistics, pages 19–26. PMLR.

[Agarwal et al., 2014] Agarwal, A., Hsu, D., Kale, S., Langford, J., Li, L., and Schapire, R. (2014). Taming the monster: A fast and simple algorithm for contextual bandits. In International Conference on Machine Learning, pages 1638–1646.

[Badanidiyuru et al., 2013] Badanidiyuru, A., Kleinberg, R., and Slivkins, A. (2013). Bandits with knapsacks. In 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, pages 207–216. IEEE.

[Dudík et al., 2011] Dudík, M., Langford, J., and Li, L. (2011). Doubly robust policy evaluation and learning. In International Conference on Machine Learning.

[Foster et al., 2018] Foster, D., Agarwal, A., Dudik, M., Luo, H., and Schapire, R. (2018). Practical contextual bandits with regression oracles. Proceedings of Machine Learning Research, 80.

[Foster and Rakhlin, 2020] Foster, D. J. and Rakhlin, A. (2020). Beyond ucb: Optimal and efficient contextual bandits with regression oracles. arXiv preprint arXiv:2002.04926.

[Langford and Zhang, 2007] Langford, J. and Zhang, T. (2007). The epoch-greedy algorithm for contextual multi-armed bandits. In Proceedings of the 20th International Conference on Neural Information Processing Systems, pages 817–824. Citeseer.

[Simchi-Levi and Xu, 2020] Simchi-Levi, D. and Xu, Y. (2020). Bypassing the monster: A faster and simpler optimal algorithm for contextual bandits under realizability. Available at SSRN.