Off-Policy Confidence Sequences

Nikos Karampatziakis [1] **Paul Mineiro** [2] **Aaditya Ramdas** [3]

Abstract

We develop confidence bounds that hold uniformly over time for off-policy evaluation in the contextual bandit setting. These confidence sequences are based on recent ideas from martingale analysis and are non-asymptotic, non-parametric, and valid at arbitrary stopping times. We provide algorithms for computing these confidence sequences that strike a good balance between computational and statistical efficiency. We empirically demonstrate the tightness of our approach in terms of failure probability and width and apply it to the "gated deployment" problem of safely upgrading a production contextual bandit system.

1. Introduction

Reasoning about the reward that a new policy π would have achieved if it had been deployed, a task known as Off-Policy Evaluation (OPE), is one of the key challenges in modern Contextual Bandits (CBs) (Langford and Zhang, 2007) and Reinforcement Learning (RL). A typical OPE use case is the validation of new modeling ideas by data scientists. If OPE suggests that π is better, this can then be validated online by deploying the new policy to the real world.

The classic way to answer whether π has better reward than the current policy h is via a confidence interval (CI). Unfortunately, CIs take a very static view of the world. Suppose that π is better than h and our OPE shows a higher but not significantly better estimated reward. What should we do? We could collect more data, but since a CI holds for a particular (fixed) sample size and is not designed to handle interactive/adaptive data collection, simply recalculating the CI at a larger sample size invalidates its coverage guarantee.

While there are ways to fix this, such as a crude union bound, the proper statistical tool for such cases is called a Confidence Sequence (CS). A CS is a sequence of CIs such that the probability that they ever exclude the true value is bounded by a prespecified quantity. In other words, they retain validity under optional (early) stopping and optional continuation (collecting more data).

In this work we develop CSs for OPE using recent insights from martingale analysis (for simpler problems). Besides the aforementioned high probability uniformly over time guarantee, these CSs make no parametric assumptions and are easy to compute. We use them to create a "gated deployment" primitive: instead of deploying π directly we keep it in a staging area where we compute its off-policy CS as h is collecting data. Then π can replace h as soon as (if ever) we can reject the hypothesis that h is better than π.

We now introduce some notation to give context to our contributions. We have iid CB data of the form (x, a, r) collected by a historical policy h in the following way. First a context x was sampled from an unknown distribution D. Then h assigns a probability to each action. An action a is sampled with probability $h(a; x)$ and performed. A reward r associated with performing a in situation x is sampled from an unknown distribution $R(x, a)$. Afterwards, we wish to estimate the reward of another policy π that is absolutely continuous wrt h. We have

$$V(\pi) = \mathbb{E}_{\substack{x\sim D\\ a\sim\pi(x)\\ r\sim R(x,a)}}[r] = \mathbb{E}_{\substack{x\sim D\\ a\sim h(x)\\ r\sim R(x,a)}}\left[\frac{\pi(a;x)}{h(a;x)} r\right] \qquad (1)$$

where the last quantity can be estimated from data.

Letting $w = \frac{\pi(a;x)}{h(a;x)}$ we see that $\mathbb{E}_{x\sim D, a\sim h}[w] = 1$, where we write w instead of $w(x, a)$ to reduce notation clutter. More generally for any function $q(x, a)$ — which is typically a predictor of the reward of a at x — we have

$$\mathbb{E}_{x\sim D, a\sim h}[wq(x, a)] = \sum_{a'} \pi(a'; x) q(x, a'), \qquad (2)$$

which reduces to $\mathbb{E}[w] = 1$ when $q(x, a) = 1$ always. Eq. (1) and (2) are the building blocks of OPE estimators. The IPS estimator (Horvitz and Thompson, 1952) estimates (1) via Monte Carlo: $\hat{V}^{\text{IPS}}(\pi) = 1/n \sum_{i=1}^{n} w_i r_i$. A plethora of other OPE estimators are discussed in Section 6. In general there is a tension between the desirability of an unbiased estimator like $\hat{V}^{\text{IPS}}$ and the difficulty of working with it in finite samples due to its excessive variance.

[1] Microsoft Azure AI [2] Microsoft Research [3] Carnegie Mellon University. Correspondence to: Nikos Karampatziakis <nikosk@microsoft.com>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

Recently, Kallus and Uehara (2019) proposed an OPE estimator based on Empirical Likelihood (Owen, 2001) with several desirable properties. Empirical Likelihood (EL) has also been used to derive CIs for OPE in CBs (Karampatziakis et al., 2020) and RL (Dai et al., 2020). Our CSs can be thought of as a natural extension to the online setting of the CIs for OPE in the batch setting; its advantages include

- Our CSs hold non-asymptotically, unlike most existing CIs mentioned above which are either asymptotically valid (or nonasymptotic but overly conservative).
- Our CSs are not unnecessarily conservative due to naive union bounds or peeling techniques.
- We do not make any assumptions, either parametric or about the support of w and r, beyond boundedness.
- Our validity guarantees are time-uniform, meaning that they remain valid under optional continuation (collecting more data) and/or at stopping times, both of which are not true for all aforementioned CIs.

2. Background: OPE Confidence Intervals

We start by reviewing OPE CIs from the perspective of Karampatziakis et al. (2020). Their CI is constructed by considering plausible distributions from a nonparametric family $\mathcal{Q}$ of distributions Q for random vectors $(w, r) \in [0, w_{\max}] \times [0, 1]$ under the constraint $\mathbb{E}_Q[w] = 1$. Let Q_{wr} be the probability that $Q \in \mathcal{Q}$ assigns to the event where the importance weight is w and the reward is r. Then there exists $Q^* \in \mathcal{Q}$ such that

$$Q^*_{wr} = \mathbb{E}_{x\sim D, a\sim h, \rho\sim R(x,a)} \left[\mathbb{I}\left[\frac{\pi(a;x)}{h(a;x)} = w \right] \cdot \mathbb{I}\left[\rho = r\right] \right]$$

and $V(\pi) = \mathbb{E}_{Q^*}[wr]$. To estimate $V(\pi)$ we can find $Q^{\text{mle}} \in \mathcal{Q}$ that maximizes the data likelihood. To find a CI we minimize/maximize $\mathbb{E}_Q[wr]$ over plausible $Q \in \mathcal{Q}$ so the data likelihood is not far off from that of Q^{mle}.

Using convex duality the MLE is $Q^{\text{mle}}_{wr} = \frac{1}{n(1+\lambda_1^{\text{mle}}(w-1))}$ where λ_1^{mle} is a dual variable solving

$$\lambda_1^{\text{mle}} = \underset{\lambda_1}{\operatorname{argmax}} \sum_{i=1}^{n} \log(1 + \lambda_1(w_i - 1))$$

subject to $1 + \lambda_1(w_{\max} - 1) \geq 0$, $1 - \lambda_1 \geq 0$. The profile likelihood $L(v) = \sup_{Q:\mathbb{E}_Q[w]=1, \mathbb{E}_Q[wr]=v} \prod_{i=1}^n Q_{w_i,r_i}$ is used for CIs. From EL theory, an asymptotic $1-\alpha$-CI is

$$\left\{ v : -2\ln\left(\frac{\prod_{i=1}^n Q^{\text{mle}}_{w_i,r_i}}{L(v)} \right) \leq \chi_1^{2,1-\alpha} \right\}$$

where $\chi_1^{2,1-\alpha}$ is the $1-\alpha$ quantile of a χ^2 distribution with one degree of freedom. Using convex duality the CI is

$$\left\{ v : B(v) - \sum_{i=1}^{n} \log(1 + \lambda_1^{\text{mle}}(w_i - 1)) \leq \chi_1^{2,1-\alpha} \right\}$$

where the dual profile log likelihood $B(v)$ is

$$B(v) = \sup_{\lambda_1,\lambda_2} \sum_{i=1}^{n} \log(1 + \lambda_1(w_i - 1) + \lambda_2(w_i r_i - v)) \quad (3)$$

subject to $(\lambda_1, \lambda_2) \in \mathcal{D}_v^0$ where

$$\begin{aligned}\mathcal{D}_v^m = \{(\lambda_1,\lambda_2) : & 1 + \lambda_1(w-1) + \lambda_2(wr - v) \geq m \\ & \forall (w,r) \in \{0, w_{\max}\} \times \{0,1\}\}. \quad (4)\end{aligned}$$

The CI endpoints can be found via bisection on v.

3. Off-Policy Confidence Sequences

We now move from the batch setting and asymptotics to online procedures and finite sample, time-uniform results. We adapt and extend ideas from Waudby-Smith and Ramdas (2020) which constructs CSs for the means of random variables in $[0, 1]$. Our key insight is to combine their construction with an interpretation of (3) as the log wealth accrued by a skeptic who is betting against the hypotheses

$$\mathbb{E}_{Q^*}[w] = 1 \text{ and } \mathbb{E}_{Q^*}[wr] = v.$$

In particular, the skeptic starts with a wealth of 1 and wants to maximize her wealth. Her bet on the outcome $w - 1$ is captured by λ_1, while λ_2 represents the bet on the outcome of $wr - v$ so that the wealth after the i-th sample is multiplied by $1 + \lambda_1(w_i - 1) + \lambda_2(w_i r_i - v)$. If the outcomes had been in $[-1, 1]$ then $|\lambda_1|$ and $|\lambda_2|$ would have an interpretation as the fraction of the skeptic's wealth that is being risked on each step. The bets can be positive or negative, and their signs represent the directions of the bet. For example, $\lambda_2 < 0$ means the skeptic will make money if $w_i r_i - v < 0$. Enforcing the constraints (4) from the batch setting here means that the resulting wealth cannot be negative.

The first benefit of this framing is that we have mapped the abstract concepts of dual likelihood, dual variables, and dual constraints to more familiar concepts of wealth, bets, and avoiding bankruptcy. We now formalize our constructions and show how they lead to always valid, finite sample, CSs. We introduce a family of processes

$$K_t(v) = \prod_{i=1}^{t} (1 + \lambda_{1,i}(w_i - 1) + \lambda_{2,i}(w_i r_i - v))$$

where $\lambda_{1,i}$ and $\lambda_{2,i}$ are predictable, i.e. based on past data (formally, measurable with respect to the sigma field $\sigma(\{(w_j, r_j)\}_{j=1}^{i-1})$). We also formalize CIs and CSs below.

Definition 1. *Given data $S_n = \{(x_i, a_i, r_i)\}_{i=1}^n$, where $x_i \sim D$, $a_i \sim h(\cdot; x_i)$, $r_i \sim R(x_i, a_i)$, a $(1-\alpha)$-confidence interval for $V(\pi)$ is a set $C_n = C(h, \pi, S_n)$ such that*

$$\sup_{D,R} \Pr(V(\pi) \notin C_n) \leq \alpha.$$

In contrast, a $(1-\alpha)$-confidence sequence for $V(\pi)$ is a sequence of confidence intervals $(C_t)_{t\in\mathbb{N}}$ such that

$$\sup_{D,R} \Pr(\exists t \in \mathbb{N} : V(\pi) \notin C_t) \leq \alpha.$$

We now have the setup to state our first theoretical result.

Theorem 1. *$K_t(V(\pi))$ is a nonnegative martingale. Moreover, the sequences $C_t = \{v : K_t(v) \leq \frac{1}{\alpha}\}$ and $\mathfrak{C}_t = \bigcap_{i=1}^t C_i$ are $(1-\alpha)$-confidence sequences for $V(\pi)$.*

All proofs are in the appendix. The process $K_t(v)$ tracks the wealth of a skeptic betting against $V(\pi) = v$. The process $K_t(V(\pi))$ is a nonnegative martingale so it has a small probability of attaining large values (formally, Ville's inequality states that the probability of ever exceeding $1/\alpha$ is at most α). Of course, we don't know $V(\pi)$, but if we retain all values of v where the wealth is below $1/\alpha$, and reject the values of v for which it has crossed $1/\alpha$ at some point, this set will always contain $V(\pi)$ with high probability; this is the basis of our construction. The strength of our approach comes from this result, as it guarantees always-valid bounds for $V(\pi)$ using only martingale arguments crucially avoiding parametric or other assumptions.

What about $v \neq V(\pi)$? Can we be sure that C_t does not contain values v very far from $V(\pi)$? That's where the betting strategy, quantified by the predictable sequences $(\lambda_{1,i})$ and $(\lambda_{2,i})$, enters. The hope is the skeptic can eventually force $K_t(v)$ to be large via a series of good bets. Importantly, Theorem 1 holds regardless of how the bets are set, but good bets will lead to "small" C_t. How to smartly bet is the subject of what follows.

4. Main Betting Strategy: MOPE

We develop our main betting strategy, MOPE (Martingale OPE) in steps starting with a slow but effective algorithm and making changes that gradually lead to a computationally efficient algorithm.

4.1. Follow The Leader

We begin with a Follow-The-Leader (FTL) strategy that is known to work very well for iid problems (De Rooij et al., 2014). We define $\ell_i^v(\lambda) = \ln(1+\lambda_1(w_i-1)+\lambda_2(w_i r_i - v))$ and set $\lambda = [\lambda_1, \lambda_2]$ to maximize wealth in hindsight

$$\lambda_t^{\text{ftl}}(v) = \operatorname*{argmax}_{\lambda} \sum_{i=1}^{t-1} \ell_i^v(\lambda) \tag{5}$$

for every step of betting in $K_t(v)$. The problem (5) is convex and can be solved in polynomial time leading to an overall polynomial time algorithm. However, this approach has three undesirable properties. First, the algorithm needs to store the whole history of (w, r) samples. Second the overall algorithm is tractable but slow. Finally, we need to solve (5) for all values of v that we have not yet rejected. We address these issues by replacing the wealth with a lower bound that is a quadratic polynomial, introducing a hedged process, and using common bets for all v. We detail these below.

4.2. Maximizing a Lower Bound on Wealth

We can avoid having to store all history by optimizing an easy-to-maintain lower bound of (5).

Lemma 1. *For all $x \geq -\frac{1}{2}$ and $\psi = 2 - 4\ln(2)$, we have*

$$\ln(1+x) \geq x + \psi x^2.$$

Observe that if we restrict our bets to lie in the convex set $\mathcal{D}_v^{1/2}$ (cf. eq. (4)) then for all $\lambda \in \mathcal{D}_v^{1/2}$

$$\sum_{i=1}^{t-1} \ell_i^v(\lambda) \geq \lambda^\top \sum_{i=1}^{t-1} b_i(v) + \psi \lambda^\top \left(\sum_{i=1}^{t-1} A_i(v)\right) \lambda$$

where $b_i(v) = \begin{bmatrix} w_i - 1 \\ w_i r_i - v \end{bmatrix}$ and $A_i(v) = b_i(v) b_i(v)^\top$. The first step towards a more efficient algorithm is to set our bets at time t as

$$\lambda_t(v) = \operatorname*{argmax}_{\lambda \in \mathcal{D}_v^{1/2}} \psi \lambda^\top \left(\sum_{i=1}^{t-1} A_i(v)\right) \lambda + \lambda^\top \sum_{i=1}^{t-1} b_i(v) \tag{6}$$

The restriction $\lambda \in \mathcal{D}_v^{1/2}$ is very mild: it does not allow the skeptic to lose more than half of her wealth from any single outcome. The first advantage of this formulation is that $\sum_i A_i(v)$ and $\sum_i b_i(v)$ are low degree polynomials of v and can share the coefficients

$$\sum_{i=1}^{t-1} A_i(v) = A_t^{(0)} + v A_t^{(1)} + v^2 A_t^{(2)}$$
$$\sum_{i=1}^{t-1} b_i(v) = b_t^{(0)} + v b_t^{(1)}.$$

Secondly, the coefficients can be updated incrementally

$$A_t^{(0)} = \sum_{i=1}^{t-1} \begin{bmatrix} (w_i-1)^2 & (w_i-1)w_i r_i \\ (w_i-1)w_i r_i & w_i^2 r_i^2 \end{bmatrix}, \tag{7}$$

$$A_t^{(1)} = \sum_{i=1}^{t-1} \begin{bmatrix} 0 & -(w_i-1) \\ -(w_i-1) & -2w_i r_i \end{bmatrix}, \tag{8}$$

$$A_t^{(2)} = \sum_{i=1}^{t-1} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \tag{9}$$

$$b_t^{(0)} = \sum_{i=1}^{t-1} \left[\begin{array}{c} w_i - 1 \\ w_i r_i \end{array} \right], \tag{10}$$

$$b_t^{(1)} = \sum_{i=1}^{t-1} \left[\begin{array}{c} 0 \\ -1 \end{array} \right]. \tag{11}$$

Finally, we can solve (6) exactly in $O(1)$ time. Section 4.4 will elaborate on this using a slight variation of eq. (6).

4.3. Common Bets and Hedging

So far we have made betting simpler, but we still need to reason about betting per value of v. While the most competitive betting sequences for the process $K(v)$ will take advantage of the knowledge of v, placing different bets for different values of v creates two problems: First, the resulting confidence set need not be an interval and second makes it hard to implement Theorem 1 in a computationally efficient way. Indeed, even in the simpler setup of Waudby-Smith and Ramdas (2020) the authors maintain a grid of test values for the quantity of interest (here v) and at least keep track of the wealth separately. This is because tracking the wealth for each value in the grid is not straightforward when the bets are different.

To make wealth tracking easy and obtain algorithms that do not require the discretization of the domain of v, a natural proposal would be to use a common bet for all v in each timestep. Unfortunately, this is not adequate because we do need $\lambda_2 > 0$ for $v < \mathbb{E}_{Q^*}[wr]$ and $\lambda_2 < 0$ for $v > \mathbb{E}_{Q^*}[wr]$. A simple fix is to use a hedged strategy as in Waudby-Smith and Ramdas (2020). First, we split the initial wealth equally. The first half is used to bet against low v's via the process

$$K_t^+(v) = \prod_{i=1}^{t} \left(1 + \lambda_{1,i}^+(w_i - 1) + \lambda_{2,i}^+(w_i r_i - v)\right)$$

and the second half to bet against high v's via a separate process $K_t^-(v)$ which for symmetry we parametrize as

$$K_t^-(v) = \prod_{i=1}^{t} \left(1 + \lambda_{1,i}^-(w_i - 1) + \lambda_{2,i}^-(w_i r_i' - v')\right).$$

where $r_i' = 1 - r_i$ and $v' = 1 - v$. This can be seen as the wealth process for betting against $1 - v$ in an world where r has been remapped to $1 - r$. Thus betting against high values of v reduces to betting against low values of v in a modified process. The total wealth of the hedged process is

$$K_t^{\pm}(v) = \frac{1}{2}(K_t^+(v) + K_t^-(v)), \tag{12}$$

and it can be used for CSs in the same way as $K_t(v)$:

Theorem 2. *The sequence $C_t^{\pm} = \{v : K_t^{\pm}(v) \leq \frac{1}{\alpha}\}$ and its running intersection $\bigcap_{i=1}^{t} C_i^{\pm}$ are $1 - \alpha$ CSs for $V(\pi)$.*

Algorithm 1 Solve $\lambda^* = \operatorname{argmax}_{\lambda \in \mathcal{C}} \psi \lambda^\top A \lambda + \lambda^\top b$

```
Input: $A, b$
$\lambda = -(2\psi A)^{-1} b$
if $\lambda \in \mathcal{C}$ then
   Return $\lambda$
end if
$\Lambda = \left\{ \left[\frac{1}{2(1-w_{\max})}, 0\right], \left[\frac{1}{2}, 0\right], \left[0, \frac{1}{2}\right] \right\}$ {vertices of $\mathcal{C}$}
for $c, d \in \{([0,1], 0), ([1,1], \frac{1}{2}), ([1 - w_{\max}, 1], \frac{1}{2})\}$ do
   $\mu = -\frac{c^\top (2\psi A)^{-1} b + d}{c^\top (2\psi A)^{-1} c}$ {Lagrange multiplier}
   $\lambda = -(2\psi A)^{-1}(b + \mu c)$
   if $\lambda \in \mathcal{C}$ then
      $\Lambda = \Lambda \cup \{\lambda\}$ {Add feasible solutions on faces}
   end if
end for
Return $\operatorname{argmax}_{\lambda \in \Lambda} \psi \lambda^\top A \lambda + \lambda^\top b$
```

It remains to design a common bet for $K_t^+(v)$. Betting against any fixed v_0 will not work well when $V(\pi) = v_0$ since the optimal bet for $V(\pi)$ is 0 but such a bet cannot help us reject those v that are far from $V(\pi)$. Therefore we propose to adaptively choose the bets against the smallest v that has not been rejected. As we construct the CS, we have access to the values of v that constitute the endpoints of the CS at the last time step. These values are on the cusp of plausibility given the available data and confidence level which means the bets are neither too conservative nor too detached from what can be estimated.

4.4. Avoiding Grid Search

Once we have determined v for the current step we could choose λ via (6). For reasons that will become apparent shortly, we can also consider a more restrictive criterion

$$\lambda_t = \operatorname*{argmax}_{\lambda \in \mathcal{C}} \psi \lambda^\top \left(\sum_{i=1}^{t-1} A_i(v) \right) \lambda + \lambda^\top \sum_{i=1}^{t-1} b_i(v), \tag{13}$$

where $\mathcal{C} = \{\lambda : \lambda_2 \geq 0\} \cap \bigcap_{v \in [0,1]} \mathcal{D}_v^{1/2}$ or more succintly

$$\mathcal{C} = \left\{ \lambda : \lambda_2 \geq 0, \lambda_1 + \lambda_2 \leq \frac{1}{2}, \lambda_1 (1 - w_{\max}) + \lambda_2 \leq \frac{1}{2} \right\}.$$

The constraint $\lambda_2 \geq 0$ is expected for good bets in $K_t^+(v)$ (and by reduction in $K_t^-(v)$) since we are eliminating v's with $\mathbb{E}[wr - v] > 0$. Since there are only three constraints and two variables we can exactly solve (13) very efficiently. Our implementation first tries to return the unconstrained maximizer, if feasible. If not, we evaluate the objective on up to 6 candidates: up to one candidate per face of $\mathcal{C}$ (obtained via maximizing the objective subject to one equality constraint) and its 3 vertices. Algorithm 1 summarizes this.

Algorithm 2 MOPE: Martingale Off-Policy Evaluation

```
Input: process $Z = (w_i, r_i)_{i=1}^{\infty}, w_{\max}, \alpha$
Let $Z' = (w_i, 1 - r_i)$ for $(w_i, r_i)$ in $Z$
for $v_i, v_i'$ in zip(LCS($Z$), LCS($Z'$)) do
    Output($v_i, 1 - v_i'$)
end for
function LCS($Z$)
    $\lambda_1 = [0, 0]^\top, v = 0$
    for $i = 1, \dots$ do
        Observe $(w_i, r_i)$ from $Z$
        Update statistics via (7)-(11) and (18)-(22).
        if (14) has real roots then
            $v = \max(v$, largest root of (14))
        end if
        yield $v$ {execution suspends/resumes here}
        $A = A_i^{(0)} + vA_i^{(1)} + v^2 A_i^{(2)}$
        $b = b_i^{(0)} + vb_i^{(1)}$
        $\lambda_{i+1} = \text{argmax}_{\lambda \in C}\, \psi \lambda^\top A \lambda + b^\top \lambda.$
    end for
end function
```

Given $\lambda_1, \dots, \lambda_{t-1}$, from (13) we get from Lemma 1

$$\sum_{i=1}^{t-1} \ell_i^v(\lambda_i) \geq \psi \sum_{i=1}^{t-1} \lambda_i^\top A_i(v) \lambda_i + \sum_{i=1}^{t-1} \lambda_i^\top b_i(v)$$

for all $v \in [0, 1]$. Thus, if the lower bound exceeds $\ln(1/\alpha)$ for a particular v, the log wealth will also exceed it. Furthermore, the lower bound is quadratic in v so we can easily find those values $v \in [0, 1]$ such that

$$\psi \sum_{i=1}^{t-1} \lambda_i^\top A_i(v) \lambda_i + \sum_{i=1}^{t-1} \lambda_i^\top b_i(v) = \ln\left(\frac{2}{\alpha}\right). \tag{14}$$

The extra 2 is due to the hedged process. Appendix B explains this and the details of how to incrementally maintain statistics for solving (14) via eqs. (18)-(22). The advantage of (13) over (6) is that the latter cannot ensure that old bets will produce values in $\mathcal{D}_v^{1/2}$ for future values of v while the former always does because $\mathcal{C} \subseteq \mathcal{D}_v^{1/2}$, $\forall v \in [0, 1]$.

The whole process of updating the statistics, tightening the lower bound v via (14) and computing the new bets via (13) is summarized in Algorithm 2.

Confidence Intervals. If one only desires a single CI using a fixed batch of data, then a CI can be formed by returning the last set from the CS on any permutation of the data. To reduce variance, we can average the wealth of several independent permutations without violating validity.

Alternative Betting Algorithms An obvious question is why develop this strategy and not just feed the convex functions $-\ell_i^v(\lambda)$ to an online learning algorithm? The Online Newton Step (ONS) (Hazan et al., 2007) is particularly well-suited as $-\ell_i^v(\lambda)$ is exp-concave. While ONS does not require storing all history and needs small per-step computation, we could not find an efficient way to efficiently reason about $K_t(v)$ for every $v \in [0, 1]$. While the ONS bounds the log wealth in terms of the gradients observed at each bet, these gradients depend on v in a way that makes it hard to efficiently reuse for different values of v. Our approach on the other hand maintains a lower bound on the wealth as a second degree polynomial in v, enabling us to reason about all values of v in constant time.

5. Extensions

5.1. Adding a Reward Predictor

When considering just the task of Off-Policy Estimation, a well-known improvement over eq. (1) is the doubly robust estimator (Dudík et al., 2011; Jiang and Li, 2016). Suppose we have access to a reward predictor $q(x, a)$ mapping context and action to an estimated reward. If this estimated reward is well correlated with the actual reward then we should expect that subtracting the zero mean quantity

$$c_i = w_i q(x_i, a_i) - \sum_{a'} \pi(a'; x_i) q(x_i, a'). \tag{15}$$

from $w_i r_i$ will not affect the mean but will reduce the variance of our estimator. We can follow the same rationale for defining our wealth process. Let

$$K_t^q(v) = \prod_{i=1}^{t} (1 + \lambda_{1,i}(w - 1) + \lambda_{2,i}(w_i r_i - c_i - v))$$

for predictable sequences $(\lambda_{1,i}, \lambda_{2,i}) \in \mathcal{E}_v^0$, where

$$\mathcal{E}_v^m = \{(\lambda_1, \lambda_2) : 1 + \lambda_1(w - 1) + \lambda_2(wr - c - v) \geq m \\ \forall (x, a, r, q) \in \text{supp}(D) \times \mathcal{A} \times \{0, 1\} \times [0, 1]^{|\mathcal{A}|}\}.$$

Note that $w = w(x, a)$ and $c = c(x, a, q)$ so all quantities are well defined. This set looks daunting but without loss of generality it suffices to only consider two actions: a, which is sampled by h, and an alternative one a', $h(a) \in \{1/w_{\max}, 1\}$, and $\pi(a), q(x, a), q(x, a') \in \{0, 1\}$. Considering all these combinations and removing redundant constraints leads to the equivalent description for $\mathcal{E}_v^m$ as

$$\left\{\lambda : \begin{bmatrix} -1 & -1 & W & W \\ -v & v' & -W - v & W + v' \end{bmatrix}^\top \lambda \geq m - 1\right\}, \tag{16}$$

where $W = w_{\max} - 1$ and $v' = 1 - v$.

For an efficient procedure we introduce the set $\mathcal{C}^q = \bigcap_{v \in [0,1]} \mathcal{E}_v^{1/2}$ to enable the use of our lower bound and common bets for all v. This set can be shown to be the same as (16) but with $v = 1$ and $v' = 1$. For predictable sequences of bets $\lambda_t^{+q}, \lambda_t^{-q} \in \mathcal{C}^q$ define the processes

$$K_t^{+q}(v) = \prod_{i=1}^{t} \left(1 + \lambda_{1,i}^{+q}(w_i - 1) + \lambda_{2,i}^{+q}(w_i r_i - c_i - v)\right),$$

$$K_t^{-q}(v) = \prod_{i=1}^{t} \left(1 + \lambda_{1,i}^{-q}(w_i - 1) + \lambda_{2,i}^{-q}(w_i r_i' - c_i' - v')\right),$$

where $r_i' = 1 - r_i$, $v' = 1 - v$. For the definition of c_i' we reason as follows: If $q(x, a)$ is a good reward predictor for r_i then $q'(x, a) = 1 - q(x, a)$ is a good reward predictor for r_i'. Plugging $q'(x, a)$ in place of $q(x, a)$ in (15) leads to $c_i' = w_i - 1 - c_i$. Finally, the hedged process is just $K_t^{\pm q}(v) = \frac{1}{2}(K_t^{+q}(v) + K_t^{-q}(v))$ and we have

Theorem 3. *The sequences $C_t^q = \{v : K_t^q(v) \le \frac{1}{\alpha}\}$ and $C_t^{\pm q} = \{v : K_t^{\pm q}(v) \le \frac{1}{\alpha}\}$ as well as their running intersections $\bigcap_{i=1}^t C_i^q$ and $\bigcap_{i=1}^t C_i^{\pm q}$ are $1 - \alpha$ CSs for $V(\pi)$.*

Appendix D contains the details on how to bet. We close this section with two remarks. First, a “bad” $q(x, a)$ can make $wr - c$ have larger variance than wr. To protect against this case we can run a *doubly hedged* process: $K_t^{\pm^2}(v) = \frac{1}{2}(K_t^{\pm q}(v) + K_t^{\pm}(v))$ which will accrue wealth almost as well as the best of its two components. Second our framework allows for $q(x, a)$ to be updated in every step as long as the updates are predictable.

5.2. Scalar Betting

Since $\mathbb{E}[w] = 1$, it would seem that the λ_1 bet cannot have any long term benefits. While this will be shown to be false in our experiments we nevertheless develop a betting strategy that only bets on $w_i r_i - v$. The advantages of this strategy are computational and conceptual simplicity. Similarly to Section 4.3 we use a hedged process $K_t^{\gtrless}(v) = \frac{1}{2}\left(K_t^{>}(v) + K_t^{<}(v)\right)$ where

$$K_t^{>}(v) = \prod_{i=1}^{} \left(1 + \lambda_{2,i}^{>}(w_i r_i - v)\right),$$

$$K_t^{<}(v) = \prod_{i=1}^{} \left(1 + \lambda_{2,i}^{<}\left(w_i(1 - r_i) - (1 - v)\right)\right).$$

The definition of $K_t^{<}(v)$ is similar to that of $K_t^{-}(v)$ and in Appendix C.1 we provide an alternative justification for using $1 - r$ and $1 - v$ via a worst case argument. The upshot is that if we start from our original proposal for $K_t(v)$ and try to eliminate λ_1 to ensure a worst case wealth we end up with $\lambda_1 = \max(0, -\lambda_2)$. Plugging this λ_1 in $K_t(v)$ leads to the above processes.

We explain betting for $K_t^{>}(v)$, since betting for $K_t^{<}(v)$ reduces to that. We use a result by Fan et al. (2015):

$$\ln(1 + \lambda\xi) \ge \lambda\xi + (\ln(1 - \lambda) + \lambda) \cdot \xi^2$$

for all $\xi \ge -1$ and $\lambda \in [0, 1)$, which we reproduce in Appendix C.2. We apply it in our case with $\xi_i = w_i r_i - v \ge -1$ and consider the log wealth lower bound for a fixed λ_2

$$\ln(K_t^{>}(v)) \ge \lambda_2 \sum_{i=1}^{t-1} \xi_i + (\ln(1 - \lambda_2) + \lambda_2) \sum_{i=1}^{t-1} \xi_i^2.$$

When $\sum_{i=1}^{t-1} \xi_i^2 > 0$ the lower bound is concave and can be maximized in λ_2 by setting its derivative to 0. This gives

$$\lambda_{2,t}^{>}(v) = \frac{\sum_{i=1}^{t-1}(w_i r_i - v)}{\sum_{i=1}^{t-1}(w_i r_i - v) + \sum_{i=1}^{t-1}(w_i r_i - v)^2}.$$

When $\sum_{i=1}^{t-1} \xi_i^2 = 0$ we can set $\lambda_{2,t}^{>}(v) = 0$. Finally, employing the same ideas as Section 4.4 we can adaptively choose the v to bet against and avoid maintaining a grid of values for v. Details are in Appendix C.3

5.3. Gated Deployment

A common OPE use case is to estimate the difference $V(\pi) - V(h)$. If we can reject all negative values (i.e. the lower CS crosses 0) then π should be deployed. Conversely, rejecting all positive values (i.e. the upper CS crosses 0) means π should be discarded. Since h is the policy collecting the data we have $V(h) = \mathbb{E}[r]$. Thus we can form a CS around $V(\pi) - V(h)$ by considering the process:

$$K_t^{gd}(v) = \prod_{i=1}^{t} \left(1 + \lambda_{1,i}(w_i - 1) + \lambda_{2,i}(w_i r_i - r_i - v)\right)$$

for predictable $\lambda_{1,i}, \lambda_{2,i}$ subject to $(\lambda_{1,i}, \lambda_{2,i}) \in \mathcal{G}_v^0$ where

$$\mathcal{G}_v^m = \{(\lambda_1, \lambda_2) : 1 + \lambda_1(w - 1) + \lambda_2(wr - r) \ge m \\ \forall (w, r) \in \{0, w_{\max}\} \times \{0, 1\}\}. \quad (17)$$

As before, we can form a hedged process and restrict bets to a set that enables the use of our lower bound. We defer these details to appendix E. We can then show

Theorem 4. *The sequences $C_t^{gd} = \{v : K_t^{gd}(v) \le \frac{1}{\alpha}\}$ and $\bigcap_{i=1}^t C_i^{gd}$ are $1 - \alpha$ CSs for $V(\pi) - V(h)$.*

This CS has two advantages over a classical A/B test. First, we don’t have to choose a stopping time in advance. The CS can run for as little or as long as necessary. Second, if π were worse than h the A/B test would have an adverse effect on the quality of the overall system, while here we can reason about this degradation without deploying π.

6. Related Work

Behind our CSs there is always a concentration inequality for martingales. Such inequalities are in a sense dual to regret guarantees online learning: Every online learning regret bound gives rise to a concentration inequality for martingales and vice versa (Rakhlin and Sridharan, 2017). At the same time minimizing regret in online learning can be achieved by betting algorithms (Cutkosky and Orabona, 2018; Orabona and Pál, 2016). These two ideas together imply that betting algorithms can also lead to CSs via their regret bounds. In particular, (Jun and Orabona, 2019) develop regret bounds via a betting view and derive interesting concentration inequalities in the spirit of the law of iterated logarithm. Our approach here, as well as that of (Waudby-Smith and Ramdas, 2020) is related and uses similar techniques. However there is a key difference: we do not rely on a regret bound. In other words, our CSs are driven by the actual regret instead of the regret bound.

Apart from IPS, other popular OPE estimators include Doubly Robust (Robins and Rotnitzky, 1995; Dudík et al., 2011) which incorporates (2) as an additive control variate and SNIPS (Swaminathan and Joachims, 2015) which incorporates $\mathbb{E}[w] = 1$ as a multiplicative control variate. The quest to balance the bias-variance tradeoff in OPE has led to many different proposals (Wang et al., 2017; Vlassis et al., 2019). EL-based estimators are proposed in Kallus and Uehara (2019) and Karampatziakis et al. (2020).

CIs for OPE include both finite-sample (Thomas et al., 2015) and asymptotic (Li et al., 2015; Karampatziakis et al., 2020) ones. Some works that propose both types are Bottou et al. (2013) and Dai et al. (2020). The latter obtains CIs without knowledge of w, a much more challenging scenario that requires additional assumptions.

We are not aware of any CSs for OPE. For on-policy setups, the most competitive CSs all rely on exploiting (super)martingales, and in some sense all admissible CSs have to (Ramdas et al., 2020). Examples include Robbins' mixture martingale (Robbins, 1970) and the techniques of Howard et al. (2020). The recent work of Waudby-Smith and Ramdas (2020) substantially increases the scope of these techniques, while simplifying and tightening the constructions.

7. Experiments

Code to reproduce all experiment results is available at `https://github.com/n17s/mope`

7.1. Coverage

While any predictable betting sequence guarantees correct coverage, some will overcover more than others. Here we investigate the coverage properties of MOPE and the strategy of Section 5.2. We generate 1000 sequences of 100000 (w, r) pairs each from a different distribution. All distributions are maximum entropy distributions subject to $(w, r) \in \{0, 0.5, 2, 100\} \times \{0, 1\}$, $\mathbb{E}[w] = 1$, $\mathbb{E}[w^2] = 10$ and $V(\pi)$ sampled uniformly in $[0, 1]$. In Figure 1 we show the empirical mean coverage of the two CSs for $\alpha = 0.05$. MOPE approaches nominal coverage from above, a property rarely seen with standard confidence bounds.

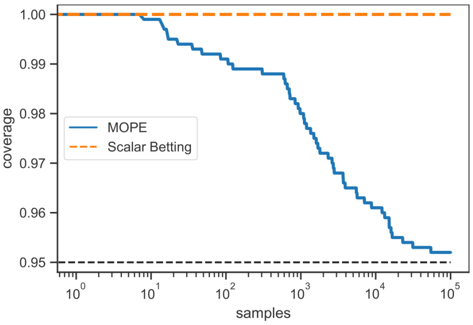

Figure 1. Empirical coverage for two proposed CSs. The CS that bets on both $w - 1$ and $wr - v$ converges to nominal coverage while the CS that does not bet on $w - 1$ overcovers.

7.2. Computational vs. Statistical Efficiency

We run an ablation study for the three ingredients of MOPE, where $-$Vector is the scalar betting technique of section 5.2; $-$Common solves (6) over a grid of 200 v values at each timestep; and $-$Bound optimizes the log wealth exactly rather than the bound of Lemma 1, i.e., Algorithm 2 with equation (5) in lieu of equation (13).

We use four synthetic environments which are distributions over (w, r) generated in the same way as section 7.1 but with $(V(\pi), \mathbb{E}[w^2]) \in \{0.05, 0.5\} \times \{10, 50\}$. Table 1 shows the running times for each method in the environment with the largest variance. We see that directly maximizing wealth and individual betting per v are very slow. MOPE and $-$Vector are computationally efficient. In Figure 2 we show the average CS width over 10 repetitions for 500000 time steps for MOPE and its ablations as well as the asymptotic CI from Karampatziakis et al. (2020) which is only valid *pointwise* and provides a lower bound for all CSs in the figure. MOPE is better than $-$Vector and as good or better than $-$Bound. This may look surprising but there is a simple explanation. In particular, using the bound leads to underestimating the optimal bet size, while using the empirical wealth so far can lead to overestimating the optimal bet size. Moreover there is an asymmetry in overestimating vs. underestimating the optimal bet. Under the approximation $\mathbb{E}[\ln(1 + \lambda X)] \approx \lambda\mathbb{E}[X] - \lambda^2\mathbb{E}[X^2]/2$ (valid for small λX) the optimal bet is $\lambda^* = \mathbb{E}[X]/E[X^2]$. Underestimating $\lambda = \lambda^*/2$ leads to an expected log wealth of 75% of the optimal, while overestimating $\lambda = 2\lambda^*$ leads to an expected log wealth of 0. We believe this asymmetry is why sometimes the bound is better than empirical wealth maximization.

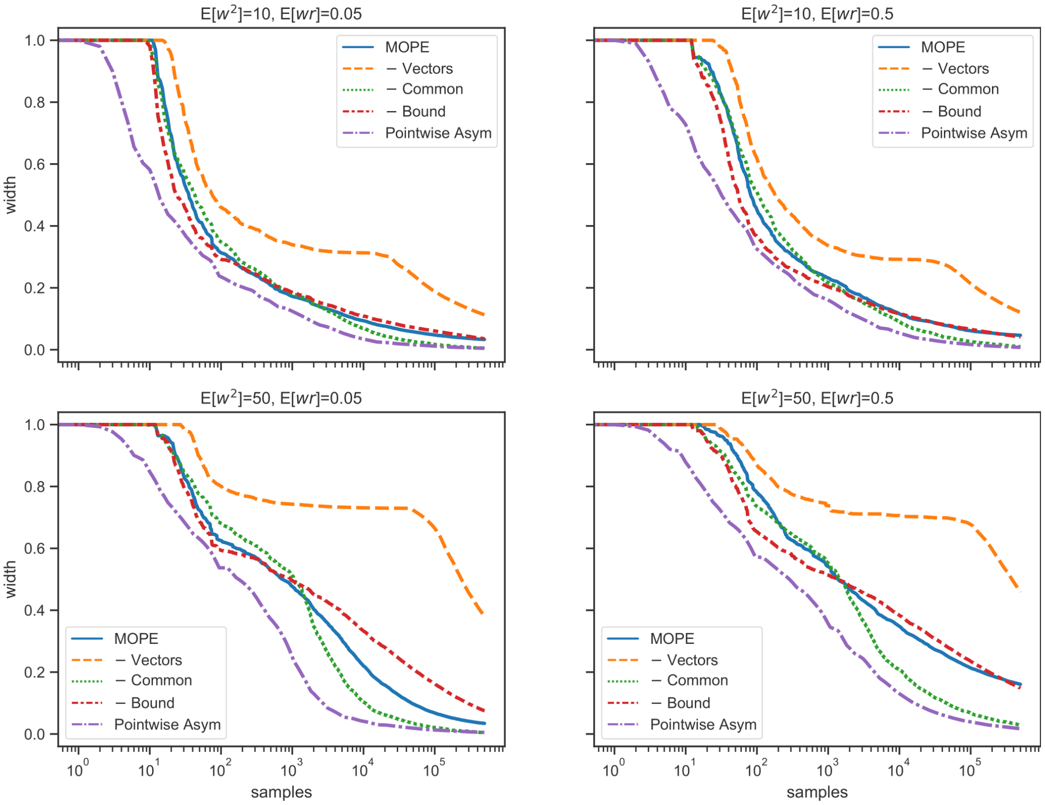

Figure 2. The width of 95% CS produced by MOPE and its three ablations. The pointwise asymptotic curve is *not* a CS.

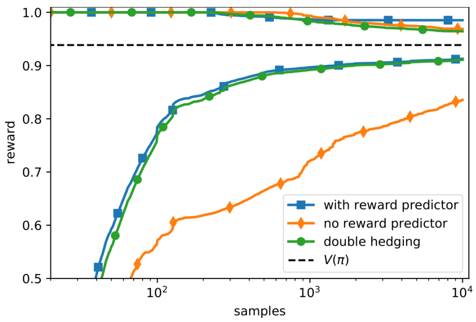

Figure 3. Three 99.9% CSs with/without a reward predictor and a doubly hedged one that achieves the best of both

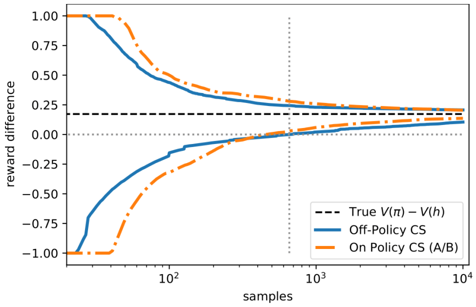

Figure 4. CS for gated deployment and A/B test. π can be deployed as soon as the lower CS crosses 0 (dotted line at $t = 657$).

Table 1. Timings for MOPE and its ablations on 500000 samples

METHOD	MOPE	-VECTOR	-COMMON	-BOUND
TIME (SEC)	32	14.5	10440	15882

Finally, while MOPE is not as tight as the (much more computationally demanding) −Common, the gap is small in all but the most challenging environment.

7.3. Effect of a Reward Predictor

We now investigate the use of reward predictors in our CSs using the processes $K_t^{\pm q}(v)$ and $K_t^{\pm^2}(v)$ of Section 5.1. We use the first 1 million samples from the mnist8m dataset which has 10 classes and train the following functions: h using linear multinomial logistic regression (MLR), π again using MLR but now on 1000 random Fourier features (RFF) (Rahimi and Recht, 2007) that approximate a Gaussian kernel machine, and finally q which uses the same RFF represetation as π but instead its i-th output is independently trained to predict whether the input is the i-th class using 10 binary logistic regressions. We used the rest of the data with the following protocol: for each input/label pair (x_i, y_i), we sample action a_i with probability $0.9h(a_i; x_i) + 0.01$ (so that we can safely set $w_{\max} = 100$), we set $r_i = 1$ if $a_i = y_i$, otherwise $r_i = 0$, and record w_i and c_i. We estimated $V(\pi) \approx 0.9385$ using the next million samples. In Figure 3 we show the CS for $V(\pi)$ averaged over 5 runs each with 10000 different samples using the processes $K_t^{\pm q}(v)$, $K_t^{\pm}(v)$ and $K_t^{\pm^2}(v)$. We see that including a reward predictor dramatically improves the lower bound and somewhat hurts the upper bound. The doubly hedged process on the other hand attains the best of both worlds.

7.4. CSs for Gated Deployment

Here we investigate the use of CSs for gated deployment. We use the same h and π and the same data as in Section 7.3 but now we are using the process $K_t^{gd}(v)$ (or rather a computationally efficient version of this process based on a hedged process with common bets and optimizing a quadratic lower bound c.f. Appendix E). Figure 4 shows the average CS over 5 runs each with 10000 different samples. We see that the CS contains the true difference (about 0.17) and quickly decides that π is better than h at $t = 657$ samples. We also include an on-policy CS (from (Waudby-Smith and Ramdas, 2020)) which *can only be computed if* π *is deployed* e.g. in an A/B test. While this is riskier when π is inferior to h, the on-policy rewards typically have lower variance. Thus the on policy CS can conclude that π is better than h at the same $\alpha = 0.01$ using 440 samples (220 for each policy). If the roles of π and h were swapped, i.e, π was the behavior policy and h was a proposed alternative, the on policy CS would still need to collect 220 samples from h. In contrast, a system using the off-policy CS would never have to experience any regret when the behavior policy is superior. Finally, during the initial phase of the experiment we observe that the off-policy CS is tighter than the on-policy CS. This is because the on-policy CS is splitting the data into two disjoint sets (one for h and one for π) while the off-policy CS is reusing the data for both h and π.

8. Conclusions

We presented a generic way to construct confidence sequences for OPE in the Contextual Bandit setting. The construction leaves a lot of freedom in designing betting strategies and we mostly explored options with an eye towards computational efficiency. Theoretically we achieve finite sample coverage and validity at any time with minimal assumptions. Empirically the resulting sequences are tight and not too far away from asymptotic and pointwise valid existing work. Theoretical results on the width of our CSs remain elusive and are both an interesting area for future work and a key to unlock much stronger analyses of various algorithms in Bandits and RL.

References

John Langford and Tong Zhang. 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, 2007.

Daniel G Horvitz and Donovan J Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*, 47(260):663–685, 1952.

Nathan Kallus and Masatoshi Uehara. Intrinsically efficient, stable, and bounded off-policy evaluation for reinforcement learning. *arXiv preprint arXiv:1906.03735*, 2019.

Art B Owen. *Empirical likelihood*. Chapman and Hall/CRC, 2001.

Nikos Karampatziakis, John Langford, and Paul Mineiro. Empirical likelihood for contextual bandits. *Advances in neural information processing systems*, 33, 2020.

Bo Dai, Ofir Nachum, Yinlam Chow, Lihong Li, Csaba Szepesvari, and Dale Schuurmans. Coindice: Off-policy confidence interval estimation. *Advances in neural information processing systems*, 33, 2020.

Ian Waudby-Smith and Aaditya Ramdas. Variance-adaptive confidence sequences by betting. *arXiv:2010.09686 [math, stat]*, October 2020. URL `http://arxiv.org/abs/2010.09686v1`. arXiv: 2010.09686.

Steven De Rooij, Tim Van Erven, Peter D Grünwald, and Wouter M Koolen. Follow the leader if you can, hedge if you must. *The Journal of Machine Learning Research*, 15(1):1281–1316, 2014.

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. *arXiv preprint arXiv:1103.4601*, 2011.

Nan Jiang and Lihong Li. Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 652–661. PMLR, 2016.

Xiequan Fan, Ion Grama, and Quansheng Liu. Exponential inequalities for martingales with applications. *Electronic Journal of Probability*, 20, 2015.

Alexander Rakhlin and Karthik Sridharan. On equivalence of martingale tail bounds and deterministic regret inequalities. In *Conference on Learning Theory*, pages 1704–1722. PMLR, 2017.

Ashok Cutkosky and Francesco Orabona. Black-box reductions for parameter-free online learning in banach spaces. In *Conference On Learning Theory*, pages 1493–1529. PMLR, 2018.

Francesco Orabona and Dávid Pál. Coin betting and parameter-free online learning. In *NIPS*, pages 577–585, 2016.

Kwang-Sung Jun and Francesco Orabona. Parameter-free online convex optimization with sub-exponential noise. In *Conference on Learning Theory*, pages 1802–1823. PMLR, 2019.

James M. Robins and Andrea Rotnitzky. Semiparametric efficiency in multivariate regression models with missing data. *Journal of the American Statistical Association*, 90 (429):122–129, 1995.

Adith Swaminathan and Thorsten Joachims. The self-normalized estimator for counterfactual learning. In *advances in neural information processing systems*, pages 3231–3239, 2015.

Yu-Xiang Wang, Alekh Agarwal, and Miroslav Dudík. Optimal and adaptive off-policy evaluation in contextual bandits. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 3589–3597, 2017. URL http://proceedings.mlr.press/v70/wang17a.html.

Nikos Vlassis, Aurelien Bibaut, Maria Dimakopoulou, and Tony Jebara. On the design of estimators for bandit off-policy evaluation. In *International Conference on Machine Learning*, pages 6468–6476, 2019.

Philip Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High-confidence off-policy evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Lihong Li, Shunbao Chen, Jim Kleban, and Ankur Gupta. Counterfactual estimation and optimization of click metrics in search engines: A case study. In *Proceedings of the 24th International Conference on World Wide Web*, pages 929–934. ACM, 2015.

Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X Charles, D Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. Counterfactual reasoning and learning systems: The example of computational advertising. *The Journal of Machine Learning Research*, 14(1):3207–3260, 2013.

Aaditya Ramdas, Johannes Ruf, Martin Larsson, and Wouter Koolen. Admissible anytime-valid sequential inference must rely on nonnegative martingales. *arXiv preprint arXiv:2009.03167*, 2020.

Herbert Robbins. Statistical Methods Related to the Law of the Iterated Logarithm. *The Annals of Mathematical Statistics*, 41(5):1397–1409, October 1970. ISSN 0003-4851, 2168-8990.

Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform, nonparametric, nonasymptotic confidence sequences. *The Annals of Statistics*, forthcoming, 2020.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *NIPS*, volume 3, page 5. Citeseer, 2007.

Jean Ville. Etude critique de la notion de collectif. *Bull. Amer. Math. Soc*, 45(11):824, 1939.