On the Optimality of Batch Policy Optimization Algorithms

Chenjun Xiao [*1 2] **Yifan Wu** [*3] **Tor Littlemore** [4] **Bo Dai** [2] **Jincheng Mei** [1 2] **Lihong Li** [†5] **Csaba Szepesvari** [1 4] **Dale Schuurmans** [1 2]

Abstract

Batch policy optimization considers leveraging existing data for policy construction before interacting with an environment. Although interest in this problem has grown significantly in recent years, its theoretical foundations remain underdeveloped. To advance the understanding of this problem, we provide three results that characterize the limits and possibilities of batch policy optimization in the finite-armed stochastic bandit setting. First, we introduce a class of *confidence-adjusted index* algorithms that unifies optimistic and pessimistic principles in a common framework, which enables a general analysis. For this family, we show that *any* confidence-adjusted index algorithm is minimax optimal, whether it be optimistic, pessimistic or neutral. Our analysis reveals that instance-dependent optimality, commonly used to establish optimality of *on-line* stochastic bandit algorithms, *cannot be achieved by any algorithm* in the batch setting. In particular, for any algorithm that performs optimally in some environment, there exists another environment where the same algorithm suffers arbitrarily larger regret. Therefore, to establish a framework for distinguishing algorithms, we introduce a new *weighted-minimax* criterion that considers the inherent difficulty of optimal value prediction. We demonstrate how this criterion can be used to justify commonly used pessimistic principles for batch policy optimization.

1. Introduction

We consider the problem of *batch policy optimization*, where a learner must infer a behavior policy given only access to a fixed dataset of previously collected experience, with no further environment interaction available. Interest in this problem has grown recently, as effective solutions hold the promise of extracting powerful decision making strategies from years of logged experience, with important applications to many practical problems (Strehl et al., 2011; Swaminathan & Joachims, 2015; Covington et al., 2016; Jaques et al., 2019; Levine et al., 2020).

Despite the prevalence and importance of batch policy optimization, the theoretical understanding of this problem has, until recently, been rather limited. A fundamental challenge in batch policy optimization is the insufficient coverage of the dataset. In online reinforcement learning (RL), the learner is allowed to continually explore the environment to collect useful information for the learning tasks. By contrast, in the batch setting, the learner has to evaluate and optimize over various candidate policies based only on experience that has been collected a priori. The distribution mismatch between the logged experience and agent-environment interaction with a learned policy can cause erroneous value overestimation, which leads to the failure of standard policy optimization methods (Fujimoto et al., 2019). To overcome this problem, recent studies propose to use the *pessimistic principle*, by either learning a pessimistic value function (Swaminathan & Joachims, 2015; Wu et al., 2019; Jaques et al., 2019; Kumar et al., 2019; 2020) or pessimistic surrogate (Buckman et al., 2020), or planning with a pessimistic model (Kidambi et al., 2020; Yu et al., 2020). However, it still remains unclear how to maximally exploit the logged experience without further exploration.

In this paper, we investigate batch policy optimization with finite-armed stochastic bandits, and make three contributions toward better understanding the statistical limits of this problem. *First*, we prove a minimax lower bound of $\Omega(1/\sqrt{\min_i n_i})$ on the simple regret for batch policy optimization with stochastic bandits, where n_i is the number of times arm i was chosen in the dataset. We then introduce the notion of a confidence-adjusted index algorithm that unifies both the optimistic and pessimistic principles in a single algorithmic framework. Our analysis suggests that any index algorithm with an appropriate adjustment, whether pessimistic or optimistic, is minimax optimal.

[*]Equal contribution [†]Work done when Lihong Li was with Google Research [1]University of Alberta [2]Google Research, Brain Team [3]Carnegie Mellon University [4]DeepMind [5]Amazon. Correspondence to: Chenjun Xiao <chenjun@ualberta.ca>, Yifan Wu <yw4@andrew.cmu.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

Second, we analyze the instance-dependent regret of batch policy optimization algorithms. Perhaps surprisingly, our main result shows that instance-dependent optimality, which is commonly used in the literature of minimizing cumulative regret of stochastic bandits, does not exist in the batch setting. Together with our first contribution, this finding challenges recent theoretical findings in batch RL that claim pessimistic algorithms are an optimal choice (e.g., Buckman et al., 2020; Jin et al., 2020). In fact, our analysis suggests that for any algorithm that performs optimally in some environment, there must always exist another environment where the algorithm suffers arbitrarily larger regret than an optimal strategy there. Therefore, any reasonable algorithm is equally optimal, or not optimal, depending on the exact problem instance the algorithm is facing. In this sense, for batch policy optimization, there remains a lack of a well-defined optimality criterion that can be used to choose between algorithms.

Third, we provide a characterization of the pessimistic algorithm by introducing a weighted-minimax objective. In particular, the pessimistic algorithm can be considered to be optimal in the sense that it achieves a regret that is comparable to the inherent difficulty of optimal value prediction on an instance-by-instance basis. Overall, the theoretical study we provide consolidates recent research findings on the impact of being pessimistic in batch policy optimization (Buckman et al., 2020; Jin et al., 2020; Kumar et al., 2020; Kidambi et al., 2020; Yu et al., 2020; Liu et al., 2020).

The remainder of the paper is organized as follows. After defining the problem setup in Sections 2, we present the three main contributions in Sections 3 to 5 as aforementioned. Section 6 discusses the related works. Section 7 gives our conclusions.

2. Problem setup

To simplify the exposition, we express our results for batch policy optimization in the setting of stochastic finite-armed bandits. In particular, assume the action space consists of $k > 0$ arms, where the available data takes the form of $n_i > 0$ real-valued observations $X_{i,1}, \dots, X_{i,n_i}$ for each arm $i \in [k] := \{1, \dots, k\}$. This data represents the outcomes of n_i pulls of each arm i. We assume further that the data for each arm i is *i.i.d.* with $X_{i,j} \sim P_i$ such that P_i is the reward distribution for arm i. Let $\mu_i = \int x P_i(dx)$ denote the mean reward that results from pulling arm i. All observations in the data set $X = (X_{i,j})_{i\in[k], j\in[n_i]}$ are assumed to be independent.

We consider the problem of designing an algorithm that takes the counts $(n_i)_{i\in[k]}$ and observations $X \in \times_{i\in[k]} \mathbb{R}^{n_i}$ as inputs and returns the index of a single arm in $[k]$, where the goal is to select an arm with the highest mean reward. Let $\mathcal{A}(X) \in [k]$ be the output of algorithm $\mathcal{A}$, Then the (simple) regret of $\mathcal{A}$ is defined as

$$\mathcal{R}(\mathcal{A}, \theta) = \mu^* - \mathbb{E}_{X\sim\theta}[\mu_{\mathcal{A}(X)}]\,,$$

where $\mu^* = \max_i \mu_i$ is the maximum reward. Here, the expectation $\mathbb{E}_{X\sim\theta}$ considers the randomness of the data X generated from problem instance θ, and also any randomness in the algorithm $\mathcal{A}$, which together induce the distribution of the random choice $\mathcal{A}(X)$. Note that this definition of regret depends both on the algorithm $\mathcal{A}$ and the problem instance $\theta = ((n_i)_{i\in[k]}, (P_i)_{i\in[k]})$. When θ is fixed, we will use $\mathcal{R}(\mathcal{A})$ to reduce clutter.

For convenience, we also let $n = \sum_i n_i$ and $n_{\min}$ denote the total number of observations and the minimum number of observations in the data. The optimal arm is a^* and the suboptimality gap is $\Delta_i = \mu^* - \mu_i$. The largest and smallest non-zero gaps are $\Delta_{\max} = \max_i \Delta_i$ and $\Delta_{\min} = \min_{i:\Delta_i>0} \Delta_i$. In what follows, we assume that the distributions P_i are 1-subgaussian with means in the unit interval $[0, 1]$. We denote the set of these distributions by $\mathcal{P}$. The set of all instances where the distributions satisfy these properties is denoted by Θ. The set of instances with $\mathbf{n} = (n_i)_{i\in[k]}$ fixed is denoted by $\Theta_{\mathbf{n}}$. Thus, $\Theta = \cup_{\mathbf{n}} \Theta_{\mathbf{n}}$. Finally, we define $|\mathbf{n}| = \sum_i n_i$ for $\mathbf{n} = (n_i)_{i\in[k]}$.

3. Minimax Analysis

In this section, we introduce the notion of a *confidence-adjusted index algorithm*, and prove that a broad range of such algorithms are minimax optimal up to a logarithmic factor. A confidence-adjusted index algorithm is one that calculates an index for each arm based on the data for that arm only, then chooses an arm that maximizes the index. We consider index algorithms where the index of arm $i \in [k]$ is defined as the sum of the sample mean of this arm, $\hat{\mu}_i = \frac{1}{n_i}\sum_{j=1}^{n_i} X_{i,j}$ plus a bias term of the form $\alpha/\sqrt{n_i}$ with $\alpha \in \mathbb{R}$. That is, given the input data X, the algorithm selects an arm according to

$$\arg\max_{i\in[k]} \; \hat{\mu}_i + \frac{\alpha}{\sqrt{n_i}}\,. \tag{1}$$

The reason we call these confidence-adjusted is because for a given confidence level $\delta > 0$, by Hoeffding's inequality, it follows that

$$\mu_i \in \left[\hat{\mu}_i - \frac{\beta_\delta}{\sqrt{n_i}},\; \hat{\mu}_i + \frac{\beta_\delta}{\sqrt{n_i}}\right] \tag{2}$$

with probability at least $1 - \delta$ for all arms with

$$\beta_\delta = \sqrt{2\log\left(\frac{k}{\delta}\right)}\,.$$

Thus, the family of confidence-adjusted index algorithms consists of all algorithms that follow this strategy, where each particular algorithm is defined by a (data independent) choice of α. For example, an algorithm specified by $\alpha =$

$-\beta_\delta$ chooses the arm with highest lower-confidence bound (highest LCB value), while an algorithm specified by $\alpha = \beta_\delta$ chooses the arm with the highest upper-confidence bound (highest UCB value). Note that $\alpha = 0$ corresponds to what is known as the *greedy* (sample mean maximizing) choice.

Readers familiar with the literature on batch policy optimization will recognize that $\alpha = -\beta_\delta$ implements what is known as the pessimistic algorithm (Jin et al., 2020; Buckman et al., 2020; Kidambi et al., 2020; Yin et al., 2021), or distributionally robust choice, or risk-adverse strategy. It is therefore natural to question the utility of considering batch policy optimization algorithms that *maximize* UCB values (i.e., implement optimism in the presence of uncertainty, or risk-seeking behavior, even when there is no opportunity for exploration). However, our first main result is that for batch policy optimization a risk-seeking (or greedy) algorithm cannot be distinguished from the more commonly proposed pessimistic approach in terms of minimax regret.

To establish this finding, we first provide a lower bound on the minimax regret:

Theorem 1. *Fix $\mathbf{n} = (n_i)_{i\in[k]}$ with $n_1 \le \cdots \le n_k$. Then, there exists a universal constant $c > 0$ such that*

$$\inf_{\mathcal{A}} \sup_{\theta\in\Theta_{\mathbf{n}}} \mathcal{R}(\mathcal{A},\theta) \ge c \max_{m\in[k]} \sqrt{\frac{\max(1,\log(m))}{n_m}}\,.$$

The assumption of increasing counts, $n_1 \le \cdots \le n_k$, is only needed to simplify the statement; the arm indices can always be re-ordered without loss of generality. The proof follows by arguing that the minimax regret is lower bounded by the Bayesian regret of the Bayesian optimal policy for any prior. Then, with a judicious choice of prior, the Bayesian optimal policy has a simple form. Intuitively, the available data permits estimation of the mean of action a with accuracy $O(\sqrt{1/n_a})$. The additional logarithmic factor appears when $n_1, \ldots, n_m$ are relatively close, in which case the lower bound is demonstrating the necessity of a union bound that appears in the upper bound that follows. The full proof appears in the supplementary material.

Next we show that a wide range of confidence-adjusted index algorithms are nearly minimax optimal when their confidence parameter is properly chosen:

Theorem 2. *Fix $\mathbf{n} = (n_i)_{i\in[k]}$. Let δ be the solution of $\delta = \sqrt{32\log(k/\delta)/\min_i n_i}$, and $\mathcal{I}$ be the confidence-adjusted index algorithm with parameter α. Then, for any $\alpha \in [-\beta_\delta, \beta_\delta]$, we have*

$$\sup_{\theta\in\Theta_{\mathbf{n}}} \mathcal{R}(\mathcal{I}(\alpha),\theta) \le 12\sqrt{\frac{\log(k/\delta)}{\min_i n_i}}\,.$$

Remark 1. *Theorem 2 also holds for algorithms that use different $\alpha_i \in [-\beta_\delta, \beta_\delta]$ for different arms.*

Perhaps a little unexpectedly, we see that *regardless* of optimism vs. pessimism, index algorithms with the right amount of adjustment, or *even no adjustment*, are minimax optimal, up to an order $\sqrt{\log(kn)}$ factor. We note that although these algorithms have the same worst case performance, they can behave very differently indeed on individual instances, as we show in the next section.

In effect, what these two results tell us is that minimax optimality is too weak as a criterion to distinguish between pessimistic versus optimistic (or greedy) algorithms when considering the "fixed count" setting of batch policy optimization. This leads us to ask whether more refined optimality criteria are able to provide nontrivial guidance in the selection of batch policy optimization methods. One such criterion, considered next, is known as instance-optimality in the literature of cumulative regret minimization for stochastic bandits.

4. Instance-Dependent Analysis

To better distinguish between algorithms we require a much more refined notion of performance that goes beyond merely considering worst-case behavior over all problem instances. Even if two algorithms have the same worst case performance, they can behave very differently on individual instances. Therefore, we consider the instance dependent performance of confidence-adjusted index algorithms.

4.1. Instance-dependent Upper Bound

Our next result provides a regret upper bound for a general form of index algorithm. All upper bounds in this section hold for any $\theta \in \Theta_{\mathbf{n}}$ unless otherwise specified, and we use $\mathcal{R}(\mathcal{A})$ instead of $\mathcal{R}(\mathcal{A},\theta)$ to simplify the notation.

Theorem 3. *Consider a general form of index algorithm, $\mathcal{A}(X) = \arg\max_i \hat{\mu}_i + b_i$, where b_i denotes the bias for arm $i \in [k]$ specified by the algorithm. For $2 \le i \le k$ and $\eta \in \mathbb{R}$, define*

$$g_i(\eta) = \sum_{j\ge i} e^{-\frac{n_j}{2}(\eta-\mu_j-b_j)_+^2} + \min_{j<i} e^{-\frac{n_j}{2}(\mu_j+b_j-\eta)_+^2}$$

and $g_i^* = \min_\eta g_i(\eta)$. Assuming $\mu_1 \ge \mu_2 \ge \cdots \ge \mu_k$, for the index algorithms (1) we have

$$\mathbb{P}\left(\mathcal{A}(X) \ge i\right) \le \min\{1, g_i^*\} \tag{3}$$

and

$$\mathcal{R}(\mathcal{A}) \le \sum_{2\le i\le k} \Delta_i \left(\min\{1,g_i^*\} - \min\{1,g_{i+1}^*\}\right) \tag{4}$$

where we define $g_{k+1}^* = 0$.

The assumption $\mu_1 \ge \mu_2 \ge \cdots \ge \mu_k$ is only required to express the statement simply; the indices can be reordered without loss of generality. The expression in equation 3 is a bit difficult to work with, so to make the subsequent analysis

simpler we provide a looser but more interpretable bound for general index algorithms as follows.

Corollary 1. *Following the setting of Theorem 3, consider any index algorithm and any $\delta \in (0,1)$. Define $U_i = \mu_i + b_i + \beta_\delta/\sqrt{n_i}$ and $L_i = \mu_i + b_i - \beta_\delta/\sqrt{n_i}$. Let $h = \max\{i \in [k] : \max_{j<i} L_j < \max_{j' \geq i} U_{j'}\}$. Then we have*

$$\mathcal{R}(\mathcal{A}) \leq \Delta_h + \frac{\delta}{k}\Delta_{\max} + \frac{\delta}{k}\sum_{i>h}(\Delta_i - \Delta_{i-1})\sum_{j \geq i} e^{-\frac{n_j}{2}\left(\max_{j'<i} L_{j'} - U_j\right)^2}.$$

Remark 2. *The upper bound in Corollary 1 can be further relaxed as $\mathcal{R}(\mathcal{A}) \leq \Delta_h + \delta\Delta_{\max}$.*

Remark 3. *The minimax regret upper bound (Theorem 2) can be recovered a result of Corollary 1 (see supplement).*

Corollary 1 highlights an inherent optimization property of index algorithms: they work by designing an additive adjustment for each arm, such that all of the bad arms ($i > h$) can be eliminated efficiently, i.e., it is desirable to make h as small as possible. We note that although one can directly plug in the specific choices of $\{b_i\}_{i\in[k]}$ to get instance-dependent upper bounds for different algorithms, it is not clear how their performance compares to one another. Therefore, we provide simpler relaxed upper bounds for the three specific cases, greedy, LCB and UCB, to allow us to better differentiate their performance across different problem instances (see supplement for details).

Corollary 2 (Regret Upper bound for Greedy)**.** *Following the setting of Theorem 3, for any $0 < \delta < 1$, the regret of greedy ($\alpha = 0$) on any problem instance is upper bounded by*

$$\mathcal{R}(\mathcal{A}) \leq \min_{i\in[k]}\left(\Delta_i + \sqrt{\frac{2}{n_i}\log\frac{k}{\delta}} + \max_{j>i}\sqrt{\frac{2}{n_j}\log\frac{k}{\delta}}\right) + \delta\,.$$

Corollary 3 (Regret Upper bound for LCB)**.** *Following the setting of Theorem 3, for any $0 < \delta < 1$, the regret of LCB ($\alpha = -\beta_\delta$) on any problem instance is upper bounded by*

$$\mathcal{R}(\mathcal{A}) \leq \min_{i\in[k]}\Delta_i + \sqrt{\frac{8}{n_i}\log\frac{k}{\delta}} + \delta\,.$$

Corollary 4 (Regret Upper bound for UCB)**.** *Following the setting of Theorem 3, for any $0 < \delta < 1$, the regret of UCB ($\alpha = \beta_\delta$) on any problem instance is upper bounded by*

$$\mathcal{R}(\mathcal{A}) \leq \min_{i\in[k]}\left(\Delta_i + \max_{j>i}\sqrt{\frac{8}{n_j}\log\frac{k}{\delta}}\right) + \delta\,.$$

Remark 4. *The results in these corollaries sacrifice the tightness of instance-dependence to obtain cleaner bounds for the different algorithms. The tightest instance dependent bounds can be derived from Theorem 3 by optimizing η.*

Discussion. The regret upper bounds presented above suggest that although they are all nearly minimax optimal, UCB, LCB and greedy exhibit distinct behavior on individual instances. Each will eventually select the best arm with high probability when n_i gets large for *all* $i \in [k]$, but their performance can be very different when n_i gets large for only a *subset* of arms $S \subset [k]$. For example, LCB performs well whenever S contains a good arm (i.e., with small Δ_i and large n_i). UCB performs well when there is a good arm i such that all worse arms are in S (n_j large for all $j > i$). For the greedy algorithm, the regret upper bound is small only when there is a good arm i where n_j is large for all $j \geq i$, in which situation both LCB and UCB perform well.

Clearly there are instances where LCB performs much better than UCB and vice versa. Consider an environment where there are two groups of arms: one with higher rewards and another with lower rewards. The behavior policy plays a subset of the arms $S \subset [k]$ a large number of times and ignores the rest. If S contains at least one good arm but no bad arm, LCB will select a good played arm (with high probability) while UCB will select a bad unplayed arm. If S consists of all bad arms, then LCB will select a bad arm by being pessimistic about the unobserved good arms while UCB is guaranteed to select a good arm by being optimistic.

This example actually raises a potential reason to favor LCB, since the condition for UCB to outperform LCB is stricter: requiring the behavior policy to play all bad arms while ignoring all good arms. To formalize this, we compare the upper bounds for the two algorithms by taking the n_i for a subset of arms $i \in S \subset [k]$ to infinity. For $\mathcal{A} \in \{\text{greedy}, \text{LCB}, \text{UCB}\}$, let $\hat{\mathcal{R}}_S(\mathcal{A})$ be the regret upper bounds with $\{n_i\}_{i\in S} \to \infty$ and $\{n_i\}_{i\notin S} = 1$ while fixing $\mu_1, ..., \mu_k$ in Corollary 2, 3, and 4 respectively. Then LCB dominates the three algorithms with high probability under a uniform prior for S:

Proposition 1. *Suppose $\mu_1 > \mu_2 > ... > \mu_k$ and $S \subset [k]$ is uniformly sampled from all subsets with size $m < k$, then*

$$\mathbb{P}\left(\hat{\mathcal{R}}_S(\text{LCB}) < \hat{\mathcal{R}}_S(\text{UCB})\right) \geq 1 - \frac{(k-m)!m!}{k!}\,.$$

This lower bound is $1/2$ when $k = 2$ and approaches 1 when k increases for any $0 < m < k$ since it is always lower bounded by $1 - 1/k$. The same argument applies when comparing LCB to greedy.

To summarize, when comparing different algorithms by their upper bounds, we have the following observations: (i) These algorithms behave differently on different instances, and none of them outperforms the others on all instances. (ii) Both scenarios where LCB is better and scenarios where UCB is better exist. (iii) LCB is more favorable when k is not too small because it is the best option among these algorithms on most of the instances.

Simulation results. Since our discussion is based on comparing only the upper bounds (instead of the exact regret) for different algorithms, it is a question that whether these statements still hold in terms of their actual performance. To answer this question, we verify these statements through experiments on synthetic problems. The details of these synthetic experiments can be found in the supplementary material.

We first verify that there exist instances where LCB is the best among the three algorithms as well as instances where UCB is the best. For LCB to perform well, we construct two ϵ-greedy behavior policies on a 100-arm bandit where the best arm or a near-optimal arm is selected to be played with a high frequency while the other arms are uniformly played with a low frequency. Figure 1(a) and 1(b) show that LCB outperforms UCB and greedy on these two instances, verifying our observation from the upper bound (Corollary 3) that LCB only requires a good behavior policy while UCB and greedy require bad arms to be eliminated (which is not the case for ϵ-greedy policies). For UCB to outperform LCB, we set the behavior policy to play a set of near-optimal arms with only a small number of times and play the rest of the arms uniformly. Figure 1(c) and 1(d) show that UCB outperforms LCB and greedy on these two instances, verifying our observation from the upper bound (Corollary 4) that UCB only requires all worse arms to be identified.

We now verify the statement that LCB is the best option on most of the instances when k is not too small. We verify this statement in two aspects: First, we show that when $k = 2$, LCB and UCB have an equal chance to be the better algorithm. More specifically, we fix $n_1 > n_2$ (note that if $n_1 = n_2$ all index algorithms are the same as greedy) and vary $\mu_1 - \mu_2$ from -1 to 1. Intuitively, when $|\mu_1 - \mu_2|$ is large, the problem is relatively easy for all algorithms. For $\mu_1 - \mu_2$ in the medium range, as it becomes larger, the good arm is tried more often, thus the problem becomes easier for LCB and harder for UCB. Figure 2(a) and 2(b) confirm this and show that both LCB and UCB are the best option on half of the instances. Second, we show that as k grows, LCB quickly becomes the more favorable algorithm, outperforming UCB and greedy on an increasing fraction of instances. More specifically, we vary k and sample a set of instances from the prior distribution introduced in Proposition 1 with $|S| = k/2$ and $|S| = k/4$. Figure 2(c) and 2(d) shows that the fraction of instances where LCB is the best quickly approaches 1 as k increases.

4.2. Instance-dependent Lower Bound

We have established that, despite all being minimax optimal, index algorithms with different adjustment can exhibit very different performance on specific problem instances. One might therefore wonder if instance optimal algorithms exist for batch policy optimization with finite-armed stochastic bandits. To answer this question, we next show that there is no instance optimal algorithm in the batch optimization setting for stochastic bandits, which is a very different outcome from the setting of cumulative regret minimization for online stochastic bandits.

For cumulative regret minimization, Lai & Robbins (1985) introduced an asymptotic notion of instance optimality (Lattimore & Szepesvári, 2020). The idea is to first remove algorithms that are insufficiently adaptive, then define a yardstick (or benchmark) for each instance as the best (normalized) asymptotic performance that can be achieved with the remaining adaptive algorithms. An algorithm that meets this benchmark over all instances is then considered to be an instance optimal algorithm.

When adapting this notion of instance optimality to the batch setting there are two decisions that need to be made: what is an appropriate notion of "sufficient adaptivity" and whether, of course, a similar asymptotic notion is sought or optimality can be adapted to the finite sample setting. Here, we consider the asymptotic case, as one usually expects this to be easier.

We consider the 2-armed bandit case ($k = 2$) with Gaussian reward distributions $\mathcal{N}(\mu_1, 1)$ and $\mathcal{N}(\mu_2, 1)$ for each arm respectively. Recall that, in this setting, fixing $\mathbf{n} = (n_1, n_2)$ each instance $\theta \in \Theta_{\mathbf{n}}$ is defined by (μ_1, μ_2). We assume that algorithms only make decisions based on the sufficient statistic — empirical means for each arm, which in this case reduces to $X = (X_1, X_2, \mathbf{n})$ with $X_i \sim \mathcal{N}(\mu_i, 1/n_i)$.

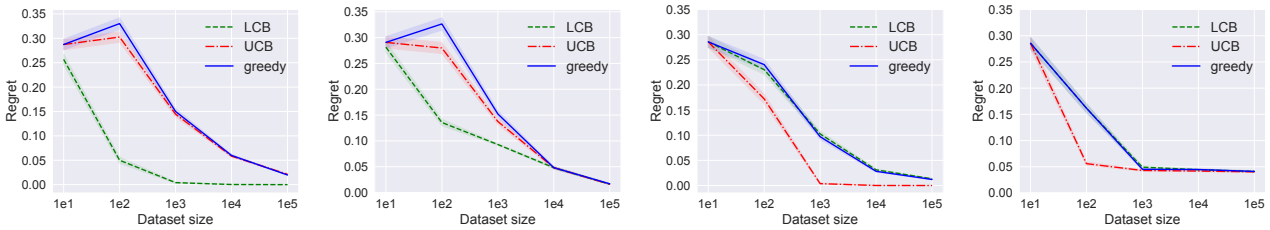
To introduce an asymptotic notion, we further denote $n = n_1 + n_2$, $\pi_1 = n_1/n$, and $\pi_2 = n_2/n = 1 - \pi_1$. Assume $\pi_1, \pi_2 > 0$; then each $\mathbf{n}$ can be uniquely defined by (n, π_1) for $\pi_1 \in (0, 1)$. We also ignore the fact that n_1 and n_2 should be integers since we assume the algorithms can only make decisions based on the sufficient statistic $X_i \sim \mathcal{N}(\mu_i, 1/n_i)$, which is well defined even when n_i is not an integer.

Definition 1 (Minimax Optimality)**.** *Given a constant $c \geq 1$, an algorithm is said to be minimax optimal if its worst case regret is bounded by the minimax value of the problem up to a multiplicative factor c. We define the set of minimax optimal algorithms as*

$$\mathcal{M}_{\mathbf{n},c} = \left\{ \mathcal{A} : \sup_{\theta \in \Theta_{\mathbf{n}}} \mathcal{R}(\mathcal{A}, \theta) \leq c \cdot \inf_{\mathcal{A}'} \sup_{\theta \in \Theta_{\mathbf{n}}} \mathcal{R}(\mathcal{A}', \theta) \right\}.$$

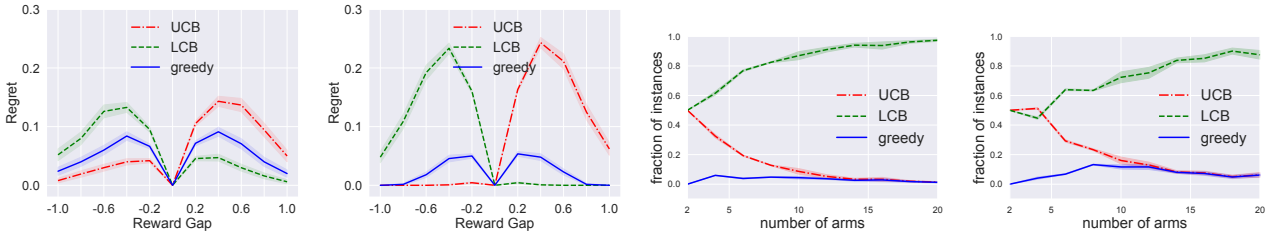
Definition 2 (Instance-dependent Lower Bound)**.** *Given a set of algorithms $\mathcal{M}$, for each $\theta \in \Theta_{\mathbf{n}}$, we define the instance-dependent lower bound as $\mathcal{R}^*_{\mathcal{M}}(\theta) = \inf_{\mathcal{A} \in \mathcal{M}} \mathcal{R}(\mathcal{A}, \theta)$.*

The following theorem states the non-existence of instance optimal algorithms up to a constant multiplicative factor.



(a) LCB-1 (b) LCB-2 (c) UCB-1 (d) UCB-2

Figure 1. Comparing UCB, LCB and greedy on synthetic problems (with $k = 100$). (a) and (b): Problem instances where LCB has the best performance. The data set is generated by a behavior policy that pulls an arm i with *high* frequency and the other arms uniformly. In (a) i is the best arm while in (b) i is the 10th-best arm. (c) and (d): Problem instances where UCB has the best performance. The data set is generated by a behavior policy that pulls a set of good arms $\{j : j \leq i\}$ with very *small* frequency and the other arms uniformly. In (c) we use $i = 1$ while in (d) we use $i = 10$. Experiment details are provided in the supplementary material.



(a) $k = 2, n_1/n_2 = 2$ (b) $k = 2, n_1/n_2 = 10$ (c) $|S| = k/2$ (d) $|S| = k/4$

Figure 2. Comparing UCB, LCB and greedy on synthetic problems. (a) and (b): A set of two-armed bandit instances where both LCB and UCB dominate half of the instances. (c) and (d): For each k, we first sample 100 vectors $\vec{\mu} = [\mu_1, ..., \mu_k]$ and for each $\vec{\mu}$ we uniformly sample 100 (if exist) subsets $S \subset k, |S| = m$ ($m = k/2$ in (c) and $m = k/4$ in (d)), to generate up to 10k instances. We then count the fraction of instances where each algorithm performs better than the other two algorithms among the randomly sampled set of instances. Experiment details are provided in the supplementary material.

Theorem 4. *Let c_0 be the constant in minimax lower bound such that $\inf_{\mathcal{A}} \sup_{\theta \in \Theta_{\mathbf{n}}} \mathcal{R}(\mathcal{A}, \theta) \geq c_0/\sqrt{n_{\min}}$. Then for any $c > 2/c_0$ and any algorithm $\mathcal{A}$, we have*

$$\sup_{\theta \in \Theta_{\mathbf{n}}} \frac{\mathcal{R}(\mathcal{A}, \theta)}{\mathcal{R}^*_{\mathcal{M}_{\mathbf{n},c}}(\theta)} \geq \frac{n_{\min}}{n_{\min} + 4} e^{\frac{\beta^2}{4} + \frac{\beta}{4}\sqrt{n_{\min}}}$$

where $\beta = cc_0 - 2$.

Corollary 5. *There is no algorithm that is instance optimal up to a constant multiplicative factor. That is, fixing $\pi_1 \in (0, 1)$, given any $c > 2/c_0$ and for any algorithm $\mathcal{A}$, we have*

$$\limsup_{n \to \infty} \sup_{\theta \in \Theta_{\mathbf{n}}} \frac{\mathcal{R}(\mathcal{A}, \theta)}{\mathcal{R}^*_{\mathcal{M}_{\mathbf{n},c}}(\theta)} = +\infty \,.$$

The proof of Theorem 4 follows by constructing two competing instances where the performance of any single algorithm cannot simultaneously match the performance of the adapted algorithm on each specific instance. Here we briefly discuss the proof idea – the detailed analysis is provided in the supplementary material.

Step 1, define the algorithm $\mathcal{A}_\beta$ as

$$\mathcal{A}_\beta(X) = \begin{cases} 1 & \text{if } X_1 - X_2 \geq \frac{\beta}{\sqrt{n_{\min}}} \\ 2 & \text{otherwise} \end{cases} .$$

For any β within a certain range, it can be shown that $\mathcal{A}_\beta \in \mathcal{M}_{\mathbf{n},c}$, hence $\mathcal{R}^*_{\mathcal{M}_{\mathbf{n},c}}(\theta) \leq \mathcal{R}(\mathcal{A}_\beta, \theta)$.

Step 2, construct two problem instances as follows. Fix a $\lambda \in \mathbb{R}$ and $\eta > 0$, and define

$$\theta_1 = (\mu_1, \mu_2) = (\lambda + \frac{\eta}{n_1}, \lambda - \frac{\eta}{n_2}) \,,$$

$$\theta_2 = (\mu_1', \mu_2') = (\lambda - \frac{\eta}{n_1}, \lambda + \frac{\eta}{n_2}) \,.$$

Since we have $X_1 - X_2 \sim \mathcal{N}(\Delta, \sigma^2)$ on instance θ_1 and $X_1 - X_2 \sim \mathcal{N}(-\Delta, \sigma^2)$ on instance θ_2, where $\Delta = (\frac{1}{n_1} + \frac{1}{n_2})\eta$ and $\sigma^2 = \frac{1}{n_1} + \frac{1}{n_2}$, the regret of $\mathcal{A}_\beta$ on both instances can be computed using the CDF of Gaussian distributions. Note that $\mathcal{R}(\mathcal{A}_{-\beta}, \theta_1) = \mathcal{R}(\mathcal{A}_\beta, \theta_2)$. We now chose a $\beta_1 < 0$ for θ_1 to upper bound $\mathcal{R}^*_{\mathcal{M}_{\mathbf{n},c}}(\theta_1)$ by $\mathcal{R}(\mathcal{A}_{\beta_1}, \theta_1)$ and use $\beta_2 = -\beta_1 > 0$ to upper bound $\mathcal{R}^*_{\mathcal{M}_{\mathbf{n},c}}(\theta_2)$ by $\mathcal{R}(\mathcal{A}_{\beta_2}, \theta_1)$.

Then applying the Neyman-Pearson Lemma (Neyman & Pearson, 1933) to this scenario gives that $\mathcal{A}_0$ is the optimal algorithm in terms of balancing the regret on θ_1 and θ_2:

$$\mathcal{R}(\mathcal{A}_0, \theta_1) = \mathcal{R}(\mathcal{A}_0, \theta_2) = \min_{\mathcal{A}} \max\{\mathcal{R}(\mathcal{A}, \theta_1), \mathcal{R}(\mathcal{A}, \theta_2)\}\,.$$

Step 3, combining the above results gives

$$\begin{aligned}
\sup_{\theta \in \Theta_{\mathbf{n}}} \frac{\mathcal{R}(\mathcal{A}, \theta)}{\mathcal{R}^*_{\mathcal{M}_{\mathbf{n},c}}(\theta)} &\geq \max\left\{ \frac{\mathcal{R}(\mathcal{A}, \theta_1)}{\mathcal{R}^*_{\mathcal{M}_{\mathbf{n},c}}(\theta_1)}, \frac{\mathcal{R}(\mathcal{A}, \theta_2)}{\mathcal{R}^*_{\mathcal{M}_{\mathbf{n},c}}(\theta_2)} \right\} \\
&\geq \max\left\{ \frac{\mathcal{R}(\mathcal{A}, \theta_1)}{\mathcal{R}(\mathcal{A}_{\beta_1}, \theta_1)}, \frac{\mathcal{R}(\mathcal{A}, \theta_2)}{\mathcal{R}(\mathcal{A}_{\beta_2}, \theta_2)} \right\} \\
&= \frac{\max\{\mathcal{R}(\mathcal{A}, \theta_1), \mathcal{R}(\mathcal{A}, \theta_2)\}}{\mathcal{R}(\mathcal{A}_{\beta_1}, \theta_1)} \\
&\geq \frac{\mathcal{R}(\mathcal{A}_0, \theta_1)}{\mathcal{R}(\mathcal{A}_{\beta_1}, \theta_1)}\,.
\end{aligned}$$

Note that both the regret $\mathcal{R}(\mathcal{A}_0, \theta_1)$ and $\mathcal{R}(\mathcal{A}_{\beta_1}, \theta_1)$ can be exact expressed as CDFs of Gaussian distributions: $\mathcal{R}(\mathcal{A}_0, \theta_1) = \Phi(-\Delta/\sigma)$ and $\mathcal{R}(\mathcal{A}_{\beta_1}, \theta_1) = \Phi\left(-\beta/(\sigma\sqrt{n_{\min}}) - \Delta/\sigma\right)$ where Φ is the CDF of the standard normal distribution.

Now we can conclude the proof by picking $\lambda = 1/2$ and $\eta = n_{\min}/2$ such that $\theta_1, \theta_2 \in [0,1]^2$. Then the result in Theorem 4 can then be proved by applying an approximation of Φ and setting $\beta_1 = -\beta_2 = 2 - cc_0$ such that both β_1 and β_2 are within the range that makes $\mathcal{A}_\beta \in \mathcal{M}_{\mathbf{n},c}$.

To summarize, in batch problems, unlike in online learning, there is no universally adaptive algorithm, and in fact all the confidence-adjusted algorithms have a niche where they outperform the others. Thus, any reasonable algorithm is equally optimal, or not optimal, depending on whether the minimax or instance optimality is considered. In this sense, there remains a lack of a well-defined optimality criterion that can be used to choose between algorithms for batch policy optimization.

5. A Characterization of Pessimism

It is known that the pessimistic algorithm, maximizing a lower confidence bound on the value, satisfies many desirable properties: it is consistent with rational decision making using preferences that satisfy uncertainty aversion and certainty-independence (Gilboa & Schmeidler, 1989), it avoids the optimizer's curse (Smith & Winkler, 2006a), it allows for optimal inference in an asymptotic sense (Lam, 2019), and in a certain sense it is the unique strategy that achieves these properties (Van Parys et al., 2017; Sutter et al., 2020). However, a pure statistical decision theoretic justification (in the sense of Berger (1985)) is still lacking.

The instance-dependent lower bound presented above attempts to characterize the optimal performance of an algorithm on an instance-by-instance basis. In particular, one can interpret the objective $\mathcal{R}(\mathcal{A}, \theta)/\mathcal{R}^*_{\mathcal{M}_{\mathbf{n},c}}(\theta)$ defined in Theorem 4 as weighting each instance θ by $1/\mathcal{R}^*_{\mathcal{M}_{\mathbf{n},c}}(\theta)$, where this can be interpreted as a measure of instance difficulty. It is natural to consider an algorithm to be optimal if it can perform well relative to this weighted criteria. However, given that the performance of an algorithm can be arbitrarily different across instances, no such optimal algorithm can exist under this criterion. The question we address here is whether other measures of instance difficulty might be used to distinguish some algorithms as naturally advantageous over others.

In a recent study, Jin et al. (2020) show that the pessimistic algorithm is minimax optimal when weighting each instance by the variance induced by the optimal policy. In another recent paper, Buckman et al. (2020) point out that the pessimistic choice has the property that its regret improves whenever the optimal choice's value is easier to predict. In particular, with our notation, their most relevant result (Theorem 3) implies the following: if b_i defines an interval such that $\mu_i \in [\hat{\mu}_i - b_i, \hat{\mu}_i + b_i]$ for all $i \in [k]$, then for $i' = \arg\max_i \hat{\mu}_i - b_i$ one obtains [1]

$$\mu^* - \mu_{i'} \leq 2b_{a^*}\,. \tag{5}$$

If we (liberally) interpret b_{a^*} as a measure of how hard it is to predict the value of the optimal choice, this inequality suggests that the pessimistic choice could be justified as the choice that makes the regret comparable to the error of predicting the optimal value.

To make this intuition precise, consider the same problem setup as discussed in Section 2. Suppose that the reward distribution for each arm $i \in [k]$ is a Gaussian with unit variance. Consider the problem of estimating the optimal value μ^* where the optimal arm a^* is also provided to the estimator. We define the set of minimax optimal estimators.

Definition 3 (Minimax Estimator)**.** *For fixed $\mathbf{n} = (n_i)_{i\in[k]}$, an estimator is said to be minimax optimal if its worst case error is bounded by the minimax estimate error of the problem up to some constant. We define the set of minimax optimal estimators as*

$$\mathcal{V}^*_{\mathbf{n}} = \left\{ \nu : \sup_{\theta \in \Theta_{\mathbf{n}}} \mathbb{E}_\theta[|\mu^* - \nu|] \leq c \inf_{\nu' \in \mathcal{V}} \sup_{\theta \in \Theta_{\mathbf{n}}} \mathbb{E}_\theta[|\mu^* - \nu'|] \right\}$$

where c is a universal constant, and $\mathcal{V}$ is the set of all

[1] This inequality follows directly from the definitions: $\mu^* - \mu_{i'} \leq \mu^* - (\hat{\mu}_{i'} - b_{i'}) \leq \mu^* - (\hat{\mu}_{a^*} - b_{a^*}) \leq 2b_{a^*}$ and we believe this was known as a folklore result, although we are not able to point to a previous paper that includes this inequality. The logic of this inequality is the same as that used in proving regret bounds for UCB policies (Lai & Robbins, 1985; Lattimore & Szepesvári, 2020). It is also clear that the result holds for any data-driven stochastic optimization problem regardless of the structure of the problem. Theorem 3 of Buckman et al. (2020) with this notation states that $\mu^* - \mu_{i'} \leq \min_i \mu^* - \mu_i + 2b_i$.

possible estimators.

Now consider using this optimal value estimation problem as a measure of how difficult a problem instance is, and then use this to weight each problem instance as in the definition of instance-dependent lower bound. In particular, let

$$\mathcal{E}^*(\theta) = \inf_{\nu \in \mathcal{V}^*_{\mathbf{n}}} \mathbb{E}_\theta[|\mu^* - \nu|]$$

be the inherent difficulty of estimating the optimal value μ^* on problem instance θ. The previous result (5) suggests (but does not prove) that $\sup_\theta \frac{\mathcal{R}(\mathrm{LCB},\theta)}{\mathcal{E}^*(\theta)} < +\infty$. We now show that not only does this hold, but up to a constant factor, the LCB algorithm is nearly weighted minimax optimal with the weighting given by $\mathcal{E}^*(\theta)$.

Proposition 2. *For any* $\mathbf{n} = (n_i)_{i\in[k]}$,

$$\sup_{\theta\in\Theta_{\mathbf{n}}} \frac{\mathcal{R}(\mathrm{LCB},\theta)}{\mathcal{E}^*(\theta)} < c\sqrt{\log|\mathbf{n}|}\,,$$

where c *is some universal constant.*

Proposition 3. *There exists a sequence* $\{\mathbf{n}_j\}$ *such that*

$$\limsup_{j\to\infty} \sup_{\theta\in\Theta_{\mathbf{n}_j}} \frac{\mathcal{R}(\mathrm{UCB},\theta)}{\sqrt{\log|\mathbf{n}_j|}\cdot\mathcal{E}^*(\theta)} = +\infty$$

$$\limsup_{j\to\infty} \sup_{\theta\in\Theta_{\mathbf{n}_j}} \frac{\mathcal{R}(\mathrm{greedy},\theta)}{\sqrt{\log|\mathbf{n}_j|}\cdot\mathcal{E}^*(\theta)} = +\infty$$

That is, the pessimistic algorithm can be justified by weighting each instance using the difficulty of predicting the optimal value. We note that this result does not contradict the no-instance-optimality property of batch policy optimization with stochastic bandits (Corollary 5). In fact, it only provides a characterization of pessimism: the pessimistic choice is beneficial when the batch dataset contains enough information that is good for predicting the optimal value.

6. Related work

In the context of offline bandit and RL, a number of approaches based on the pessimistic principle have been proposed and demonstrate great success in practical problems (Swaminathan & Joachims, 2015; Wu et al., 2019; Jaques et al., 2019; Kumar et al., 2019; 2020; Buckman et al., 2020; Kidambi et al., 2020; Yu et al., 2020; Siegel et al., 2020). We refer interested readers to the survey by Levine et al. (2020) for recent developments on this topic. To implement the pessimistic principle, the distributional robust optimization (DRO) becomes one powerful tool in bandit (Faury et al., 2019; Karampatziakis et al., 2019) and RL (Xu & Mannor, 2010; Yu & Xu, 2015; Yang, 2017; Chen et al., 2019; Dai et al., 2020; Derman & Mannor, 2020).

In terms of theoretical perspective, the statistical properties of general DRO, *e.g.*, the consistency and asymptotic expansion of DRO, is analyzed in (Duchi et al., 2016). Liu et al. (2020) provides regret analysis for a pessimistic algorithm based on stationary distribution estimation in offline RL with insufficient data coverage. Jin et al. (2020) and Kidambi et al. (2020) recently prove that the pessimistic algorithm is nearly minimax optimal for batch policy optimization. However, the theoretical justification of the benefits of pessimitic principle vs. alternatives are missing in offline RL.

Decision theory motivates DRO with an axiomatic characterization of min-max (or distributionally robust) utility: Preferences of decision makers who face an uncertain decision problem and whose preference relationships over their choices satisfy certain axioms follow an ordering given by assigning max-min utility to these preferences (Gilboa & Schmeidler, 1989). Thus, if we believe that the preferences of the user follow the axioms stated in the above work, one must use a distributionally optimal (pessimistic) choice. On the other hand, Smith & Winkler (2006b) raise the "optimizer's curse" due to statistical effect, which describes the phenomena that the resulting decision policy may disappoint on unseen out-of-sample data, *i.e.*, the actual value of the candidate decision is below the predicted value. Van Parys et al. (2017); Sutter et al. (2020) justify the optimality of DRO in combating with such an overfitting issue to avoid the optimizer's curse. Moreover, Delage et al. (2019) demonstrate the benefits of randomized policy from DRO in the face of uncertainty comparing with deterministic policy. While reassuring, these still leave open the question whether there is a justification for the pessimistic choice dictated by some alternate logic, or perhaps a more direct logic reasoning in terms of regret in decision problem itself (Lattimore & Szepesvári, 2020).

Our theoretical analysis answer this question, and provides a complete and direct justification for all confidence-based index algorithms. Specifically, we show all confidence-based index algorithms are nearly minimax optimal in terms of regret. More importantly, our instance-dependent analysis show that for any algorithm one can always find some problem instance where the algorithm will suffer arbitrarily large regret. Therefore, one cannot directly compare the performance of two algorithms without specifying the problem instance. Buckman et al. (2020) state that for the pessimistic choice to be a good one, it suffices to have data that makes predicting the value of the optimal policy feasible. We provide a formal analysis to support this intuition: the pessimistic algorithm is nearly minimax optimal when weighting individual instance by its inherent difficulty of estimating the optimal value. This weighted criterion can be used to distinguish pessimistic algorithm from other confidence-adjusted index algorithms.

7. Conclusion

In this paper we study the statistical limits of batch policy optimization with finite-armed bandits. We introduce a family of confidence-adjusted index algorithms that provides a general analysis framework to unify the commonly used optimistic and pessimistic principles. For this family, we show that any index algorithm with an appropriate adjustment is nearly minimax optimal. Our analysis also reveals another important finding, that for any algorithm that performs optimally in some environment, there exists another environment where the same algorithm can suffer arbitrarily large regret. Therefore, the instance-dependent optimality cannot be achieved by any algorithm. To distinguish the algorithms in offline setting, we introduce a weighted minimax objective and justify the pessimistic algorithm is nearly optimal under this criterion.

8. Acknowledgments

Chenjun Xiao and Bo Dai would like to thank Yinlam Chow for providing feedback on a draft of this manuscript. Yifan Wu would like to thank Zachary Lipton for the discussions on this topic. Csaba Szepesvári and Dale Schuurmans gratefully acknowledge funding from the Canada CIFAR AI Chairs Program, Amii and NSERC.

References

Berger, J. O. *Statistical Decision Theory and Bayesian Analysis*. Springer Series in Statistics. Springer New York, New York, NY, January 1985. URL http://link.springer.com/10.1007/978-1-4757-4286-2.

Buckman, J., Gelada, C., and Bellemare, M. G. The importance of pessimism in Fixed-Dataset policy optimization. September 2020. URL http://arxiv.org/abs/2009.06799.

Chen, Z., Yu, P., and Haskell, W. B. Distributionally robust optimization for sequential decision-making. *Optimization*, 68(12):2397–2426, 2019.

Covington, P., Adams, J., and Sargin, E. Deep neural networks for Youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pp. 191–198, 2016.

Dai, B., Nachum, O., Chow, Y., Li, L., Szepesvári, C., and Schuurmans, D. Coindice: Off-policy confidence interval estimation. *arXiv preprint arXiv:2010.11652*, 2020.

Delage, E., Kuhn, D., and Wiesemann, W. "dice"-sion–making under uncertainty: When can a random decision reduce risk? *Management Science*, 65(7):3282–3301, July 2019.

Derman, E. and Mannor, S. Distributional robustness and regularization in reinforcement learning. March 2020. URL http://arxiv.org/abs/2003.02894.

Duchi, J., Glynn, P., and Namkoong, H. Statistics of robust optimization: A generalized empirical likelihood approach. October 2016. URL https://arxiv.org/abs/1610.03425v3.

Faury, L., Tanielian, U., Vasile, F., Smirnova, E., and Dohmatob, E. Distributionally robust counterfactual risk minimization. June 2019. URL http://arxiv.org/abs/1906.06211.

Fujimoto, S., Meger, D., and Precup, D. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pp. 2052–2062. PMLR, 2019.

Gilboa, I. and Schmeidler, D. Maxmin expected utility with non-unique prior. *Journal of Mathematical Economics*, 18(2):141–153, 1989.

Jaques, N., Ghandeharioun, A., Shen, J. H., Ferguson, C., Lapedriza, A., Jones, N., Gu, S., and Picard, R. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. 2019.

Jin, Y., Yang, Z., and Wang, Z. Is pessimism provably efficient for offline RL? 2020.

Karampatziakis, N., Langford, J., and Mineiro, P. Empirical likelihood for contextual bandits. *arXiv preprint arXiv:1906.03323*, 2019.

Kidambi, R., Rajeswaran, A., Netrapalli, P., and Joachims, T. MOReL: Model-based offline reinforcement learning. In *NeurIPS*, 2020.

Kumar, A., Fu, J., Tucker, G., and Levine, S. Stabilizing off-policy q-learning via bootstrapping error reduction. 2019.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. 2020.

Lai, T. L. and Robbins, H. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1): 4–22, 1985.

Lam, H. Recovering best statistical guarantees via the empirical Divergence-Based distributionally robust optimization. *Oper. Res.*, 67(4):1090–1105, July 2019. URL https://doi.org/10.1287/opre.2018.1786.

Lattimore, T. and Szepesvári, C. *Bandit algorithms*. Cambridge University Press, 2020.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. 2020.

Liu, Y., Swaminathan, A., Agarwal, A., and Brunskill, E. Provably good batch reinforcement learning without great exploration. 2020.

Neyman, J. and Pearson, E. S. Ix. on the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706):289–337, 1933.

Siegel, N. Y., Springenberg, J. T., Berkenkamp, F., Abdolmaleki, A., Neunert, M., Lampe, T., Hafner, R., and Riedmiller, M. Keep doing what worked: Behavioral modelling priors for offline reinforcement learning. *arXiv preprint arXiv:2002.08396*, 2020.

Smith, J. E. and Winkler, R. L. The optimizer's curse: Skepticism and postdecision surprise in decision analysis. *Manage. Sci.*, 52(3):311–322, March 2006a. URL https://doi.org/10.1287/mnsc.1050.0451.

Smith, J. E. and Winkler, R. L. The optimizer's curse: Skepticism and postdecision surprise in decision analysis. *Management Science*, 52(3):311–322, 2006b.

Strehl, A. L., Langford, J., Li, L., and Kakade, S. M. Learning from logged implicit exploration data. In *Advances in Neural Information Processing Systems 23*, pp. 2217–2225, 2011.

Sutter, T., Van Parys, B. P. G., and Kuhn, D. A general framework for optimal Data-Driven optimization. October 2020. URL http://arxiv.org/abs/2010.06606.

Swaminathan, A. and Joachims, T. Batch learning from logged bandit feedback through counterfactual risk minimization. *The Journal of Machine Learning Research*, 16(1):1731–1755, 2015.

Van Parys, B. P. G., Esfahani, P. M., and Kuhn, D. From data to decisions: Distributionally robust optimization is optimal. April 2017. URL http://arxiv.org/abs/1704.04118.

Wu, Y., Tucker, G., and Nachum, O. Behavior regularized offline reinforcement learning. 2019.

Xu, H. and Mannor, S. Distributionally robust markov decision processes. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems-Volume 2*, pp. 2505–2513, 2010.

Yang, I. A convex optimization approach to distributionally robust markov decision processes with wasserstein distance. *IEEE control systems letters*, 1(1):164–169, 2017.

Yin, M., Bai, Y., and Wang, Y.-X. Near-optimal offline reinforcement learning via double variance reduction. *arXiv preprint arXiv:2102.01748*, 2021.

Yu, P. and Xu, H. Distributionally robust counterpart in markov decision processes. *IEEE Transactions on Automatic Control*, 61(9):2538–2543, 2015.

Yu, T., Thomas, G., Yu, L., Ermon, S., Zou, J., Levine, S., Finn, C., and Ma, T. Mopo: Model-based offline policy optimization. 2020.