Advancements Blending Reinforcement Learning and Design of Experiments

Peter P. Norwood
Department of Statistics, North Carolina State University

Abstract

In sequential experiments, researchers must assign interventions to incoming groups of subjects. As more groups complete the experiment, researchers can analyze the accruing data to understand which interventions cause some numeric reward (favorable outcome) and which need more attention to reduce uncertainty. The exploration-exploitation trade-off is a fundamental problem in sequential experimentation. This trade-off describes the choice to exploit accruing information from the experiment to gain some immediate reward or to explore the system in search an optimal strategy that maximizes reward globally. Special consideration is required when data are expensive because exploration can be costly or unethical. In this article, we review (a) multi-armed bandits and Markov decision processes, two topics fundamental to sequential decision making, (b) experimental design work that quantifies information gain in an experiment, and (c) approaches that carefully consider if, when, and how to explore. A companion simulation study compares these approaches with classical methods in sequential experiments using linear models with multiple interventions and multiple covariates – a contextual linear bandit. The review and simulation study highlight the opportunity to blend reinforcement learning and classical design of experiments to build more data-efficient sequential designs that balance exploration and exploitation.

Keywords: Sequential decision making, reinforcement learning, design of experiments, multi-armed bandits

1 Introduction

Sequential decision making in the face of uncertainty describes a variety of scenarios ranging from choosing a restaurant to searching for efficacious treatments in a clinical trial. The fundamental trade-off is whether to exploit the accruing information from the experiment to choose an action (e.g., treatment, intervention, etc.) that appears to be favorable or to explore other actions in search of the optimum. This paper reviews the statistical framework for sequential decision making and surveys methodology that studies the exploration-exploitation trade-off. Particular attention is given to methodology that focuses on if, when, and how to explore.

Reinforcement learning (RL) is a subfield of machine learning and artificial intelligence that formalizes sequential decision making. In their introductory book on RL, Sutton and Barto (2018) first describe RL as learning what to do to maximize a numeric reward over time. A *reward* is a random variable that depends on the action, coded so higher values indicate a more favorable response (e.g., number of advertisement clicks). A *policy* is a strategy for selecting actions based on available information. Formalizing this problem and developing methods to find policies that maximize reward have tremendous scientific interest. Applications include clinical trials, (Williamson and Villar, 2019; Korn and Freidlin, 2017; Wathen and Thall, 2017; Berry, 2012), mobile health (mHealth) (Lei et al., 2017; Luckett et al., 2020; Tomkins et al., 2020), hiring practices (Li et al., 2020; Schumann et al., 2019), online advertising (Zeng et al., 2016; Nguyen-Thanh et al., 2019), and elsewhere.

Design of experiments (DOE) is a subfield of Statistics as old as the field itself. A major topic of interest in DOE is optimal designs, first considered by Smith (1918). Creating a design can be viewed as selecting actions, or design points. Optimal designs select the actions that maximize some utility in an experiment given its constraints (e.g., total

sample size, cost of actions). Often, this utility is information gain, which can be defined as reduction in uncertainty of parameter estimates. With more precise estimates, more accurate decisions can be made that lead to higher reward.

In RL, heuristics often determine if, when, and how to explore. An example is ϵ-greedy, an algorithm that chooses the greedy (estimated optimal) action with probability $1 - \epsilon_t$ or randomly chooses a non-greedy action with probability ϵ_t for the t-th action. Because DOE examines how best to explore, there is an opportunity to combine aspects from both fields to create more data-efficient RL algorithms. These algorithms are particularly important to research where data are expensive and exploration can be costly or unethical.

Section 2 formalizes sequential decision making by covering multi-armed bandits (MABs) and Markov decision processes (MDPs). Brief overviews of the methods to identify optimal policies, Q and V-learning, conclude Section 2. Section 3 provides a high-level overview of optimal DOE. Methodology in sequential experimentation to balance exploration and exploitation is covered in Section 4. The article concludes with a discussion about future research in Section 5.

2 Sequential Decision Making Fundamentals

2.1 Multi-Armed Bandit Problems

A MAB is a sequential decision problem where the *agent*, or decision-maker, repeatedly makes the decision on which of the K *actions*, or arms, to choose (Sutton and Barto, 2018). The statistical problem is designing and evaluating bandit algorithms that simultaneously learn optimal strategies to maximize reward and use that information to maximize cumulative reward over time. Often, constraints like the number of total actions one can take or

the cost of exploration need to be considered. An example is a clinical trial where funding or time constrains the experiment to T total patients. Additionally, if a treatment appears effective, it can be unethical to explore other treatments later in the experiment.

MABs are sometimes framed as sequential experiments where the actions are interventions and the goal is to find effective intervention strategies. This framing is a reason why MABs are of such interest to statisticians, notably in adaptive clinical trials (Williamson and Villar, 2019; Korn and Freidlin, 2017; Wathen and Thall, 2017; Berry, 2012).

Many algorithms exist to balance exploitation with exploration. We provide a high-level overview of three popular ones: ϵ-greedy, Thompson sampling, and upper confidence bound (UCB). In Section 4 we discuss more sophisticated algorithms for MABs.

We use the following example to aid explanation. We model the t-th outcome with a linear model:

$$Y_t = \phi(\mathbf{X}_t, A_t)^T \theta^* + \varepsilon_t. \tag{2.1}$$

The reward is simply the outcome, where higher values are coded to be more favorable. Here, $\phi(\cdot, \cdot) \in \mathbb{R}^p$ is a function that creates a feature vector based on the context $\mathbf{X}_t \in \mathbb{X}$, where $\mathbb{X}$ is the entire domain of the context, and discrete action $A_t \in (1, \ldots, K)$. A simple example is $\phi(X_t, A_t) = (1, X_t, A_t, X_t A_t)^T$, where X_t is the scalar context. The mean parameter vector is denoted $\theta^* \in \mathbb{R}^p$ and the residual error is ε_t. Let $\widehat{\theta}_t$ be the ordinary least squares (OLS) estimator after t observations: $\widehat{\theta}_t = \arg\min_{\theta \in \Theta} \sum_{i=1}^{t} \left\{ Y_i - \phi(\mathbf{X}_i, A_i)^T \theta \right\}^2$, where Θ is the parameter space.

As previously mentioned, ϵ-greedy explores with probability ϵ_t and chooses the greedy-action with probability $1 - \epsilon_t$. For (2.1), the t-th action is selected as:

$$P(A_t = a) = \begin{cases} 1 - \epsilon_t, & \text{if } a = \arg\max_{a' \in (1,\ldots,K)} \phi(\mathbf{X}_t, A_t = a')^T \widehat{\theta}_{t-1} \\ \epsilon_t/(K-1), & \text{otherwise.} \end{cases}$$

The exploration probability need not be stationary. If ϵ_t decreases to zero with t, then ϵ-greedy converges to a greedy algorithm. Similarly, an explore-then-commit strategy starts with a period of uniform randomization, then switches to a greedy algorithm after t_0 subjects (Yekkehkhany et al., 2019). In general, the choice of ϵ_t depends on problem-specific goals, e.g., in a more traditional experiment, where the data will be used to make statistical inference, more exploration may be warranted.

Thompson sampling randomizes according to the probability that an action is optimal. This approach dates back to Thompson (1933) and is now widely popular, notably in adaptive clinical trials (Thall and Wathen, 2007; Wathen and Thall, 2017). Typically, Thompson sampling is a Bayesian approach. Let $\rho(\theta^*)$ be the prior distribution of θ^* and $\mathbf{H}_t = (\mathbf{X}_1, A_1, \ldots \mathbf{X}_t, A_t)$ be the accrued data after t observations. Then $\rho(\theta^* \mid \mathbf{H}_t)$ is the posterior distribution of θ^*. Based on this posterior, the randomization probability is:

$$P(A_t = a) = \mu \left[P \left\{ a = \underset{a' \in (1,\ldots,K)}{\arg\max} \, \phi(\mathbf{X}_t, A_t = a')^T \theta^* \mid \mathbf{H}_{t-1} \right\} \right],$$

where $\mu(\tilde{p})$ is some function relating the randomization probability to the posterior probability, $\tilde{p}$. One example is to threshold the randomization probabilities to some maximum and minimum; e.g., $\mu(\tilde{p}) = \tilde{p}\mathbb{I}(0.1 \leq \tilde{p} \leq 0.9) + 0.1\mathbb{I}(\tilde{p} < 0.1) + 0.9\mathbb{I}(\tilde{p} > 0.9)$, where $\mathbb{I}(\cdot)$ is the indicator function (Zhang et al., 2020). Russo et al. (2017) provide an overview of Thompson sampling and its many applications.

The next approach is UCB, which selects the action that maximizes the estimated reward plus some measure of uncertainty – an upper confidence bound. An obvious example

for the linear model is:

$$A_t = \underset{a \in (1,\ldots,K)}{\arg\max} \left[\phi(\mathbf{X}_t, A_t = a)^T \widehat{\theta}_{t-1} + Z_\alpha \left\{ \phi(\mathbf{X}_t, A_t = a)^T \widehat{\Sigma}_{t-1} \phi(\mathbf{X}_t, A_t = a) \right\}^{1/2} \right],$$

where Z_α is the $1-\alpha$ quantile and of the standard normal distribution and $\widehat{\Sigma}_t$ is the OLS estimator for the covariance matrix of $\widehat{\theta}_t$. Sutton and Barto (2018) suggest using $c\{t/N_t(a)\}^{1/2}$ as the measure of uncertainty, where c is some constant and $N_t(a)$ is the number of times action a has been selected.

A popular way to compare different MAB algorithms is through their *regret*, the expected difference in reward from the true optimal action and the action chosen. MAB algorithms that provide lower cumulative regret after T actions are more favorable. For (2.1), the cumulative regret is,

$$\text{Regret}(T) = \sum_{t=1}^{T} \left[\max_{a \in (1,\ldots,K)} \left\{ \phi(\mathbf{X}_t, A_t = a)^T \theta^* \right\} - \phi(\mathbf{X}_t, A_t)^T \theta^* \right].$$

Regret bounds describe the finite-sample behavior of regret for different bandit algorithms. Analytical bounds on Regret(T) can be derived and used to compare different algorithms (Agrawal and Goyal, 2012). An algorithm is said to *rate-optimal* if it achieves a minimal regret bound. For more information on regret bounds and rate optimality, see Lai and Robbins (1985); Slivkins (2019).

2.2 Markov Decision Processes

MDPs provide the general framework to study sequential decision making problems (MABs are a subset of these problems). We base this overview on Alagoz et al. (2009) and Sutton and Barto (2018). In this section, we consider discrete-time MDPs, where decisions are

made on discrete intervals (e.g., a decision to continue or discontinue medication each month). For a more detailed summary of MDPs and their applications, see John et al. (1989).

In an MDP, the agent has $t = 0, 1, 2, \ldots$ decisions to make. At decision point t, the agent knows the *state* $S_t \in \mathcal{S}$, which is all relevant information to the decision process. Let s denote a realized value of S_t. This framework allows a flexible definition of the state; an example of the state for an mHealth application that encourages exercise is the patient characteristics (age, sex, etc.), a count of how many health reminders have been sent that day, and a count of how many times the patient responded (exercised) to the reminder. Based on S_t, the agent selects an action $A_t \in \mathcal{A}(s)$. Let a denote a realized value of A_t. The set $\mathcal{A}(s)$ is the set of possible actions based on s. The reward is $R_t(a_t, s_t) \in \mathcal{R}$, where $\mathcal{R}$ is the support of the reward's distribution.

The following probability defines the *dynamics* of the system:

$$p_t(s', r|s, a) \triangleq P\left(S_{t+1} = s', R_t = r \mid S_t = s, A_t = a\right).$$

This distribution describes the probability of the state and reward based on the previous state and action. Based on the dynamics, we define the *transition probability* as $p_t(s'|s, a) \triangleq P\left(S_t = s' \mid S_{t-1} = s, A_{t-1} = a\right)$. The transition probability is the probability of the the state at decision point t, conditional on the previous state and action. It follows that $p_t(s'|s, a) = \int_{r \in \mathcal{R}} p_t(s', r|s, a) dr$. The *Markov assumption* (also called as the *Markov property*) states that dynamics are fully described by S_t and A_t, i.e., conditional on (S_t, A_t), the dynamics are independent of all previous information (e.g., S_{t-1}).

The goal (scientific, business, etc.) in sequential decision making problems is to search for action strategies that maximize cumulative reward over time. We define

$G_t \triangleq \sum_{k \geq 0} \gamma^k R_{t+k+1}$ as the *return*, the sum of rewards after t. Here $\gamma \in [0, 1]$ is the *discounting factor* which reduces the importance of reward over time (reward now can be more important than reward later). Discounting is especially popular in *infinite-horizon* MDPs ($T = \infty$), but can also be used in *finite-horizon* MDPs ($T < \infty$). An example of an infinite-horizon MDP is the aforementioned mHealth application; here, there is no clear end point and the number of decision points is best treated as infinite. An example of a finite-horizon MDP is a cancer treatment regime where each month for six months patients receive increasingly more invasive therapies if they fail to respond to the previous therapy.

A policy, π, is the strategy for selecting actions based on the state. We define $\pi(a|s) \triangleq P^\pi(A_t = a|S_t = s)$, the probability we select $A_t = a$ given $S_t = s$ if we are following the policy π. A simple policy could be if the patient responded at time point t ($S_t = 1$), then give the treatment $A_t = 1$; otherwise randomize to either treatment with equal probability. We compare different policies through their *values*; higher values correspond with more cumulative reward. Define the *state-value function* for a policy, which is the expected return if we follow π after t, given the current state s: $V^\pi(s) \triangleq \mathbb{E}^\pi \left(G_t \mid S_t = s \right)$. Similarly, define the *action-value function* for a policy to be the value of taking action a, then following π: $q^\pi(s, a) \triangleq \mathbb{E}^\pi \left(G_t \mid S_t = s, A_t = a \right)$

A policy is optimal, π_{opt}, if and only if $V^{\pi_{opt}}(s) \geq V^\pi(s)$ for any other π and all $s \in \mathcal{S}$. Similarly, it can be shown that a policy is optimal if and only if $q^{\pi_{opt}}(s, a) \geq q^\pi(s, a)$.

An obvious question is: how can we use data to find, or estimate an optimal policy? The *Belman equation* is useful for stating V^π. For finite MDPs (where $\mathcal{S}, \mathcal{A}$, and $\mathcal{R}$ are finite), the Bellman equation is:

$$V^\pi(s) = \sum_{a \in \mathcal{A}(s)} \pi(a \mid s) \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r \mid, a) \left\{ r + \gamma V^\pi(s') \right\}.$$

The Bellman equation is also useful for stating $V^{\pi_{opt}}$, specifically the *Bellman optimality equation*:

$$V^{\pi_{opt}}(s) = \max_{a \in \mathcal{A}(s)} \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r \mid s, a) \left\{ r + \gamma V^{\pi_{opt}}(S_{t+1}) \right\}.$$

These representations are crucial for estimating V^{π} and learning $V^{\pi_{opt}}$ (Sutton and Barto, 2018). Notably, these provide the basis for variants of Q and V-learning, two methods to learn optimal policies discussed in the subsequent subsections (Ertefaie and Strawderman, 2018; Luckett et al., 2020).

2.3 Q-Learning

A popular approach for learning an optimal policy is *Q-learning*. Clifton and Laber (2020) describe Q-learning as the class of methods that focus on optimizing the *Q-function* (formally defined below). At decision point t, the Q-function models the return of each possible action, given the decision-maker will choose the optimal action at all subsequent decision points. Seminal works in Q-learning include Bellman (1957), Watkins and Dayan (1992), and Murphy (2005).

For finite-horizon decision problems, Q-learning finds an optimal policy through backward-induction of the Q-functions, i.e., model the outcome at time T, then based on the model at T, we model for $T-1$, and so on. At the final decision point T, the Q-function is $Q_T(s,a) \triangleq \mathbb{E}\left(R_T \mid S_T = s, A_T = a\right)$. Then, for $t < T$, the Q-function is:

$$Q_t(s,a) \triangleq \mathbb{E}\left\{ R_t + \max_{a \in \mathcal{A}(S_{t+1})} \gamma Q_{t+1}(S_{t+1}, a) \mid S_t = s, A_t = a \right\}. \tag{2.2}$$

Sutton and Barto (2018) call $Q_t(\cdot,\cdot)$ as the *learned* action-value function. The learned action-value function approximates the optimal action-value function, $q_t^{\pi_{opt}}(s,a)$. Given a

class of possible Q-functions $\{\mathcal{Q}_t\}_{t=1}^T$, we can estimate the optimal Q-functions by minimizing a loss function. Consider the following example with the squared loss function. Given data on $i = 1, \ldots, n$ subjects for $t = 1, \ldots, T$ time points:

$$\widehat{Q}_T = \underset{Q_T \in \mathcal{Q}_T}{\arg\min} \sum_{i=1}^{n} \{R_{T,i} - Q_T(S_{T,i}, A_{T,i})\}^2,$$

$$\ldots$$

$$\widehat{Q}_t = \underset{Q_t \in \mathcal{Q}_t}{\arg\min} \sum_{i=1}^{n} \left[\left\{ R_{t,i} + \max_{a \in \mathcal{A}(S_{t+1,i})} \widehat{Q}_{t+1}(S_{t+1,i}, A_{t+1,i} = a) \right\} - Q_t(S_{t,i}, A_{t,i}) \right]^2,$$

Then, the optimal policy at time t is $\widehat{\pi}_{t,opt} \triangleq \arg\max_{a \in \mathcal{A}_t(S_{t+1,i})} \widehat{Q}_t(S_t = s, A_t = a)$ and $\widehat{\pi}_{opt} = (\widehat{\pi}_{1,opt}, \ldots, \widehat{\pi}_{T,opt})$.

Q-learning in the finite-horizon context is popular in precision medicine, notably to estimate optimal treatment regimes (Tsiatis et al., 2020; Schulte et al., 2014). Q-learning also serves as the basis for developing adaptive randomization procedures for Sequential Multiple Assignment Randomized Trials (SMARTs) (Cheung et al., 2015), and sample-size calculations for SMARTs (Rose et al., 2019).

Q-learning can be extended to the infinite-horizon case. Given data with n subjects and up to T time points, Ertefaie and Strawderman (2018) propose positing a linear model $Q(s, a, \theta) = \phi(s, a)^T \theta$ and solving the estimating equation:

$$\frac{1}{n} \sum_{i=1}^{n} \left(\sum_{t=1}^{T} \left[\left\{ R_{t,i} + \gamma \max_{a \in \mathcal{A}(S_{t+1,i})} \phi(S_{t+1,i}, A_{t+1,i} = a)^T \theta \right\} - \phi(S_{t,i}, A_{t,i})^T \theta \right] \phi(S_{t,i}, A_{t,i}) \right) = 0.$$

The solution $\widehat{\theta}_n$ can then be used to determine an optimal policy: $\widehat{\pi}^{opt} = \arg\max_{a \in \mathcal{A}(s)} Q(s, a, \widehat{\theta}_n)$. Because of the Markov assumption, we can model only using S_t, A_t. This leads to only one

Q-function, making this tractable with infinite time steps.

Infinite-horizon Q-learning is often used in online learning, where batches of subjects enter sequentially and the model(s) update after each batch's outcomes are observed. In this context, decision-makers have the choice to select the optimal action or to explore. The exploration methods in Section 2.1 can be used. For example, Clifton and Laber (2020) present a simple algorithm to blend Thompson sampling with Q-learning and cite other examples of exploration within online Q-learning.

2.4 V-Learning

An alternative to Q-learning in the infinite-horizon case is V-learning, proposed by Luckett et al. (2020). The primary application is mHealth applications where actions can be taken anywhere from a few times times a day to every minute. Like Q-learning, this approach also is applicable for both offline (modeling is done on the same, previously collected data) and online learning where data come sequentially.

We first consider the offline case. Assume the data generating process satisfies the Markov property. Let $\omega_t(a, s) = P(A_t = a \mid S_t = s)$ be the propensity score, which might be known or can be estimated from the data. Then, let $V^\pi(s, \theta)$ be some model for $V^\pi(s)$, the state-value function, and $\nabla_\theta V^\pi(s, \theta)$ be its gradient. Define

$$\Lambda_n^\pi(\theta) = n^{-1} \sum_{i=1}^{n} \sum_{t=1}^{T_i} \frac{\pi(A_{t,i}, S_{t,i})}{\omega(A_{t,i}, S_{t,i})} \left\{ R_{t,i} + \gamma V^\pi(S_{t+1,i}, \theta) - V^\pi(S_{t,i}, \theta) \right\} \nabla_\theta V^\pi(S_{t,i}, \theta), \quad (2.3)$$

and subsequently, $\widehat{\theta}_n = \arg\min_\theta \left\{ \Lambda_n^\pi(\theta)^T \Omega \Lambda_n^\pi(\theta) + \vartheta_n \mathcal{L}(\theta) \right\}$, where Ω is a positive definite matrix, $\mathcal{L}$ is a penalty function, and ϑ_n is a tuning parameter. The estimated *value* of a policy π is $\widehat{V}_n^\pi = \int_{\mathfrak{R}_*} V^\pi(s, \widehat{\theta}_n) d\mathcal{R}(s)$, where $\mathcal{R}(s)$ is a reference distribution and $\mathfrak{R}$ is its

support. The estimated optimal policy is then $\widehat{\pi} = \arg\max_{\pi \in \Pi} \widehat{V}_n^{\pi}$, where Π is the class of potential policies. As the authors note, the essence of V-learning is using 2.3 to estimate the value of a policy and then to maximize over possible policies.

Online V-learning is similar to online Q-learning. Rather than always following the estimated optimal policy, some exploration strategy is used to learn the system. An important feature of V-learning is the ability to consider randomized policies (e.g., follow the estimated optimal policy with probability 0.95, explore a different policy with probability 0.05), particularly when deterministic policies are of scientific interest. As the authors note, using ϵ-greedy introduces randomness into a deterministic policy and induces a randomized policy. For online learning with randomized policies, equation 2.3 can be replaced with,

$$\Lambda_n^{\pi}(\theta) = n^{-1} \sum_{i=1}^{n} \sum_{v=1}^{t} \frac{\pi(A_{v,i}, S_{v,i})}{\widehat{\pi}_{n,v-1}(A_{v,i}, S_{v,i})} \left\{ R_{v,i} + \gamma V^{\pi}(S_{v+1,i}, \theta) - V^{\pi}(S_{v,i}, \theta) \right\} \nabla_{\theta} V^{\pi}(S_{v,i}, \theta),$$

where $\omega(\cdot, \cdot)$ is replaced with $\widehat{\pi}_{n,t}(a, S_{t+1})$, the probability of action assignment according to the estimated policy at time t from n subjects.

A main issue with infinite-horizon Q-learning is the modeling of the Q-function. A simple parametric model risks misspecification and a flexible model likely results in an uninterruptible policy. Because V-learning only requires modeling of the state-action function and then maximizes over a class of potential policies, it an attractive alternative for infinite-horizon decision problems. Another key advantage is the ability to estimate a randomized decision rule; as we have shown, this can be useful in online learning.

3 Design of Experiments

A vast literature exists for DOE. In this section, rather than a high-level overview of the field, we will highlight some important ideas that are used for sequential decision making problems.

We frequently use (2.1) to illustrate the ideas. With this linear model, under typical assumptions, the approximate large-sample covariance matrix of $\widehat{\theta}_t$ is proportional to $\mathbf{M}_t^{-1}$, where $\mathbf{M}_t = \sum_{i=1}^{t} \phi(\mathbf{X}_i, A_i)\phi(\mathbf{X}_i, A_i)^T$. Similar results exist for maximum likelihood estimators (MLE) in non-linear models.

3.1 Optimal Experimental Design

Optimal experimental design studies how to choose a design to achieve some goal (e.g., maximum reduction in uncertainty). Let $(\mathbf{X}_v, A_v) \in \mathcal{X} \subset \mathbf{R}^{p*}$ be a unique design point, where $\mathcal{X}$ is the set of all possible design points. After t subjects have completed the experiment, each $(\mathbf{X}_v, A_v)$ has been chosen t_i times each and there exist t^* unique points. Based on this, we can define $\eta_v = t_v/t$ and $\mathbf{M}_t = t\sum_{v=1}^{t^*} \eta_i \phi(\mathbf{X}_v, A_v)\phi(\mathbf{X}_i, A_i)^T$. Then, the *design* η is the allocation of design points according to each η_v. Furthermore, η can be viewed as a probability measure on $\mathcal{X}$ (Fedorov, 1972).

Because $\mathbf{M}_t^{-1} \in \mathbb{R}^{p\times p}$, we denote $\psi(\eta)$ as the scalar optimality criterion used to measure the utility from a design. Commonly, $\psi(\eta)$ is a function summarizing the variability of $\widehat{\theta}_t$, with smaller values indicating more reduction in variability; examples of $\psi(\eta)$ follow. A design, η^{opt}, is an optimal design, if and only if $\psi(\eta^{opt}) \leq \psi(\eta)$ for all $\eta \in \mathcal{X}$.

Popular choices for $\psi(\eta)$ include the following (Fedorov and Leonov, 2019). The D-criterion is $\det\left(\mathbf{M}_t^{-1}\right)$, where $\det(\cdot)$ denotes the determinant of a matrix. The E-criterion is the maximum eigenvalue: $\lambda_{max}\left(\mathbf{M}_t^{-1}\right)$. The A-criterion is $\mathrm{tr}\left\{\mathbf{A}\mathbf{M}_t^{-1}\right\}$, where $\mathbf{A}$ is the

utility matrix and can be customized to fit the problem and $\text{tr}(\cdot)$ denotes the trace of a matrix. A special case of the A-criterion is the C-criterion, which minimizes the variance of one linear combination, $\mathbf{c}^T\widehat{\theta}_t$, where $\mathbf{c} \in \mathbb{R}^p$. Naturally, the C-criterion is $\mathbf{c}^T\mathbf{M}_t^{-1}\mathbf{c}$.

Often, the optimal design is decided before an experiment begins. For example, consider simple linear regression where $Y_t = \theta_0^* + \theta_1^* X_t + \varepsilon_t$ and $X_t \in [-1, 1] \cap \mathbb{R}$. With an even T total subjects, it is easy to verify that the D-optimal design is half the design points at -1 and the other half at 1: $\eta_1 = T/2, X_1 = -1, \eta_2 = T/2, X_2 = 1$. Software to find optimal designs in more complicated situations is available, notably the `OptimalDesign` package in `R` and `PROC OPTEX` in `SAS` (Radoslav Harman, 2019; SAS, 2014).

For sequential optimal designs, the next design point is chosen according to the optimality criterion. In fact, Wynn (1970) shows that sequential D-optimal designs and D-optimal designs are equivalent asymptotically. Sometimes the entire design point can be chosen, other times just the action. For example, in linear model example, the sequential D-optimal action is $\arg\min_{a \in (1,\ldots,K)} \det\left\{\mathbf{M}_{t-1} + \phi(\mathbf{X}_t, A_t = a)\phi(\mathbf{X}_t, A_t = a)^T\right\}^{-1}$.

3.2 Bayesian Experimental Design

Bayesian experimental design focusing on choosing η to maximize an expected utility function $U(\eta) = \int_\Theta \int_\mathcal{Y} U(\theta, \eta, \mathbf{Y}) p(\theta|\mathbf{Y}, \eta) p(\mathbf{y}|\eta) d\theta d\mathbf{Y}$, where $\Theta \in \mathbb{R}^p$ is the parameter space and $\mathcal{Y}$ is the domain of the outcome (Chaloner and Verdinelli, 1995). An example is the *Kullback-Leibler divergence* (KL-divergence) between the prior ($\mathcal{P}_0$) and posterior ($\mathcal{P}_1$) distribution of θ:

$$D_{KL}(\mathcal{P}_0||\mathcal{P}_1) = \int_\Theta \int_\mathcal{Y} \log\left\{\frac{p(\theta \mid \mathbf{Y}, \eta)}{p(\theta)}\right\} p(\mathbf{Y}, \theta \mid \eta) d\theta d\mathbf{Y}.$$

We can use the expected *Shannon Information* as our utility function:

$$U(\eta) = \int_{\Theta} \int_{\mathcal{Y}} \log \{p(\theta \mid \mathbf{Y}, \eta)\} \, p(\mathbf{Y}, \theta \mid \eta) d\theta d\mathbf{Y}.$$

Because $p(\theta)$ does not depend on η, maximizing KL-divergence with respect to η is equivalent to maximizing Shannon Information. Intuitively, because KL-divergence measures the "distance" between two distributions, maximizing KL-divergence between the prior and posterior relates to maximizing information from the design.

To gain more intuition behind Bayesian optimal design, consider the example from Chaloner and Verdinelli (1995). Suppose the prior distribution for θ is multivariate normal with mean θ_0 and covariance matrix $\mathbf{R} \in \mathbb{R}^{p \times p}$, $MVN(\theta_0, \mathbf{R})$, and $Y_t \mid \theta, \mathbf{X}_t, A_t \sim MVN\left\{\phi(\mathbf{X}_t, A_t)^T \theta, \sigma^2\right\}$. Then, the posterior distribution of θ is p-dimensional multivariate normal with covariance matrix proportional to $(t\mathbf{M}_t + \mathbf{R})^{-1}$. The Shannon Information is:

$$U(\eta) = -\frac{p}{2} \{1 + \log(2\pi)\} + \frac{1}{2} \log \det \left\{\sigma^{-2}(t\mathbf{M}_t + \mathbf{R})\right\}.$$

Hence, the design that maximizes the KL-divergence and Shannon Information maximizes $\det(t\mathbf{M}_t + \mathbf{R})$, providing a clear connection to the D-optimal design.

Two other definitions of interest are the *entropy* and *mutual information*. For a probability distribution p, the entropy is $H(p) = -\sum_{x \in \mathcal{X}} p(x) \log \{p(x)\}$. For two random variables X_1, X_2, the mutual information is the KL-divergence between their joint distribution, $p(X_1, X_2)$, and the product of their marginal distributions, $p(X_1)p(X_2)$. If two random variables are independent, then they share no information, hence their mutual information is zero. These help quantity information gain in Information-Direct sampling, discussed in detail in Section 4.1 (Russo and Van Roy, 2018).

4 Main Methodology

4.1 Learning to Optimize Via Information-Directed Sampling

Russo and Van Roy (2018) introduce *Information-Directed Sampling* (IDS). IDS is a general design principle for online optimization problems like MABs. With multiple $A_t \in \mathcal{A}(s)$ to choose from, IDS quantifies the information gain from selecting $A_t = a$. They define information gain as the expected reduction in entropy if an action is chosen and the response is observed. Then, actions are sampled based on the ratio of their squared expected regret and their information gain – the cost per bit of information gained. The policy with the lowest cost per bit of information gained is the selected policy.

The state at decision t is all accruing information from the experiment, $S_t = \{A_1, Y_1(A_1), \ldots, A_{t-1}, Y_1(A_{t-1})\}$. We define $\pi_t(a \mid s) = P^\pi(A_t = a \mid S_t = s)$. The goal is to find a policy that minimizes the the expected regret. Let A^* denote the optimal action, which yields zero regret.

Let $p_t(a) = P(A^* = a \mid S_t)$ denote the posterior distribution of A^*. Russo and Van Roy define information gain as the mutual information of $p_t(a)$ and $p_{t+1}(a)$. This can be expressed as the expected reduction in entropy of the posterior distribution of A^* due to observing $Y_t(a)$: $g_t(a) \triangleq \mathbb{E}\left\{H\left\{p_t(a)\right\} - H\left\{p_{t+1}(a)\right\} \mid S_t, A_t = a\right\}$. Additionally, let $\Delta_t(a) \triangleq \mathbb{E}\left\{R_t(A^*) - R_t(a)\right\}$ denote the instantaneous regret if $A_t = a$.

Let $p_t(\mathcal{S}_t) \in \mathcal{D}(\mathcal{A})$ denote a probability distribution over $\mathcal{A}$, where $\mathcal{D}(\mathcal{A})$ is the set of all probability distributions over $\mathcal{A}$. For a policy π, the expected information gain is $g_t(\pi) = \sum_{a \in \mathcal{A}} \pi(a|S_t) g_t(a)$. Similarly, $\Delta_t(\pi) = \sum_{a \in \mathcal{A}} \pi(a|S_t)\Delta_t(a)$ denotes the expected regret if π is followed. The IDS policy minimizes the ratio of squared regret and information

gain. That is,

$$\pi_t^{IDS} \in \underset{\pi \in \mathcal{D}(\mathcal{A})}{\arg\min} \left\{ \Psi_t(\pi) \triangleq \frac{\Delta_t(\pi)^2}{g_t(\pi)} \right\}, \tag{4.1}$$

where $\Psi_t(\pi)$ is the *information ratio*. A policy with a small information ratio could have a low regret, high information gain, or ideally a combination of the two.

Russo and Van Roy provide algorithms to compute the information ratio in beta-Bernoulli, independent Gaussian, and linear bandits. In the simulation section, we propose two algorithms to calculate the information ratio for a contextual linear bandit.

Russo and Van Roy argue that IDS improves on other heuristics like Thompson sampling and UCB because (a) it is able to collect information on other actions through the selected action, (b) it can collect feedback that enables higher reward later, and (c) it avoids selecting actions which fail to provide information. They present practical examples where Thompson sampling or UCB fail to satisfy (a), (b), or (c). Additionally, simulation experiments show IDS outperforms the state of the art methods (including Thompson sampling and UCB) in beta-Bernoulli, independent Gaussian, and linear bandit scenarios. In all three scenarios, IDS has lower mean regret compared to all other methods considered.

4.2 Mostly Exploration-Free Algorithms for Contextual Bandits

The previous work mostly considers how to explore, but Bastani et al. (2017) study *if* exploration is necessary in contextual bandits. They show if sufficient randomness exists in the context, then a greedy algorithm has enough *natural exploration* to outperform algorithms with forced exploration. Due to this, the *Greedy-First* (GF) algorithm is presented. GF begins as a greedy algorithm. After selecting A_t, GF checks to see if sufficient information is gathered from the greedy algorithm. If yes, the next action is chosen greedily. If

no, GF switches to a different algorithm with forced exploration.

A key definition is that a random variable $\mathbf{Z} \in \mathbb{R}^p$ has *covariate diversity* if there exists a constant λ_0 such that for every $\mathbf{u} \in \mathbb{R}^p$,

$$\lambda_{min}\left(\mathbb{E}\left[\mathbf{Z}\mathbf{Z}^T\ \mathbb{I}\{\mathbf{Z}^T\mathbf{u} \geq 0\}\right]\right) \geq \lambda_0, \tag{4.2}$$

where λ_{min} denotes the minimum eigenvalue.

The problem formulation model (2.1). Here, the contexts $\mathbf{X}$ are i.i.d. from some distribution. In the case where $A_t \in \{0, 1\}$, Bastani et al. demonstrate that when $\mathbf{X}$ has covariate diversity, greedy optimization has enough natural exploration to ensure parameter estimators converge to their true values. First, in the two-arm case, we can represent $Y_t = \phi(\mathbf{X}_t, A_t)^T\theta^* = (1 - A_t)\mathbf{X}_t^T\theta_0^* + A_t\mathbf{X}_t^T\theta_1^*$ so that $\mathbf{M}_t$ can be represented as a block diagonal matrix with $\mathbf{M}_{t,0}, \mathbf{M}_{t,1}$ on the diagonal. Hence, we can partition the OLS estimator, $\widehat{\theta}_t = (\widehat{\theta}_{t,0}, \widehat{\theta}_{t,1})$. This formulation plus the next result imply that when the context has covariate diversity, each action will be estimated optimal infinitely often.

A key result is if the covariate space is bounded and if $\mathbf{X}$ has covariate diversity, then for λ_0 in (4.2) and any $\mathbf{u} \in \mathbb{R}^p, P\left\{\phi(\mathbf{X}, A)^T\mathbf{u} \geq 0\right\} \geq \lambda_0/x_{max}$, where x_{max} is the maximum entry in $\mathbf{X}$. When taking $\mathbf{u} = \left(\widehat{\theta}_{t,0}, \widehat{\theta}_{t,1}\right)^T$ or $\mathbf{u} = \left(\widehat{\theta}_{t,1}, \widehat{\theta}_{t,0}\right)^T$, we see for all t, there is a non-zero probability that each arm will be the greedy choice when $K = 2$. This implies each action will be selected infinitely often as t approaches infinity.

Based on the previous result, it is shown that $\lambda_{min}(\mathbf{M}_{t,0})$ and $\lambda_{min}(\mathbf{M}_{t,1})$ both grow linearly with t under greedy optimization. This leads to our next key result. If the covariate space is bounded and if $\mathbf{X}$ has covariate diversity, then under greedy optimization, $\widehat{\theta}_t \xrightarrow{p} \theta^*$ using the model (2.1). A specific bound on the rate of convergence is given. This tells us that under common bandit and linear model assumptions, if the context has

sufficient randomness (covariate diversity), then greedy optimization leads to consistent OLS estimators and is a rate optimal solution.

Based on these results, the authors present the GF algorithm. For the first t_0 decisions, GF selects the greedy action. After the $t \geq t_0$ action is selected, if $\lambda_{min}(\mathbf{M}_t) > \lambda_0 t/4$, then the greedy algorithm continues for $t+1$. If $\lambda_{min}(\mathbf{M}_t) \leq \lambda_0 t/4$, GF switches to some other bandit algorithm with forced exploration. It is reasonable to run a burn-in period of simple randomization until $\mathbf{M}_t$ is non-singular before starting GF. In GF, t_0 and λ_0 are user defined parameters. The authors suggest using the data-driven $\widehat{\lambda}_0 = (2t_0)^{-1}\lambda_{min}(\mathbf{M}_{t_0})$ for λ_0.

The authors show similar convergence guarantees for generalized linear models and provide a GF algorithm in that case. In multiple simulation experiments, the authors show greedy optimization and GF outperform (measured by minimizing regret) variants of UCB, Thompson sampling, and ϵ-greedy when the context has covariate diversity. When covariate diversity fails, GF has similar performance to the competitors and greedy optimization under-performs.

In this paper, Bastani et al. ponder if we even should explore. They then prove that greedy optimization is preferable to forced exploration in certain contextual bandit scenarios. Because this is not the case in all scenarios, they provide the simple and intuitive GF algorithm that automatically checks to see if exploration is needed. This work has considerable implications for situations where exploration is costly.

4.3 Adaptive Optimization and D-Optimum Experimental Design

Pronzato (2000) blends online learning and optimization with optimal experimental design.

We are interested in choosing actions $\mathbf{A} \in \mathcal{X} \subset \mathbb{R}^{p*}$ to maximize a parametric reward function $R(\mathbf{A}, \theta^*)$. This method is proposed for adaptive control systems, where researchers have complete control of the design, hence $\mathbf{A}$ is the entire design in this case. The outcome is $Y_t = \phi(\mathbf{A}_t)^T\theta^* + \varepsilon_t$. We assume $\phi(\cdot)$ is continuous in $\mathbf{A}$ and ε_t is a martingale difference sequence with respect to an increasing sequence of σ-fields $\{F_t\}$, so $E(\varepsilon_t \mid \mathcal{F}_{t-1}) = 0$ for all t. Let $\widehat{\theta}_t$ be the OLS estimator after t observations and $\mathbf{M}_t = \sum_{i=1}^{t} \phi(\mathbf{A}_t)\phi(\mathbf{A}_t)^T$ (similar to the previous linear model example). We assume a burn-in period of t_{min} of uniform randomization, so $\mathbf{M}_{t_{min}}$ is positive definite and $\widehat{\theta}_t$ can be computed.

For $t > t_{min}$, to simultaneously learn θ^* and choose $\mathbf{A}$ to maximize the reward, Pronzato proposes choosing

$$\mathbf{A}_{t+1} = \arg\max_{\mathbf{A}\in\mathcal{X}} R(\mathbf{A}, \widehat{\theta}_t) + \frac{\alpha_t}{t} d_t(\mathbf{A}),$$

where $d_t(\mathbf{A}) = \phi(\mathbf{A})^T\mathcal{I}_t^{-1}\phi(\mathbf{A})$ and $\mathcal{I}_t = \mathbf{M}_t/t$ and α_t is the weight for the information gain penalty. This represents a compromise between the greedy choice and the sequential D-optimal choice; the equivalence $\arg\max_{\mathbf{A}\in\mathcal{X}} d_t(\mathbf{A}) = \arg\min_{\mathbf{A}\in\mathcal{X}} \det \mathbf{M}_{t+1}^{-1}$ is a result of the matrix determinant lemma.

Pronzato considers how to choose α_t such that $\widehat{\theta}_t \xrightarrow{a.s.} \theta^*$ and $t^{-1}\sum_{i=1}^{t} R(\mathbf{A}_i, \theta^*) \xrightarrow{a.s.} \max_{\mathbf{A}\in\mathcal{X}} R(\mathbf{A}, \theta^*)$ as $t \to \infty$. With a compact design space and a continuous reward function, they show if α_t is chosen such that $(\alpha_t/t)\log(\alpha_t)$ is decreasing and $\alpha_t/(\log t)^{1+\delta}$ is increasing to ∞ for some positive δ then the OLS estimator is consistent and the design maximizes reward. Unfortunately, no practical guidance for selecting α_t is provided.

The idea of choosing $\mathbf{A}_{t+1} = \max_{\mathbf{A}\in\mathcal{X}} R(\mathbf{A}, \widehat{\theta}_t) + d_t(\mathbf{A})$ where larger values of $d_t(\mathbf{A})$ correspond to more reduction in the variability of $\widehat{\theta}_t$ had been studied previously (Aström and Wittenmark, 1995; Wittenmark, 1975; Wittenmark and Elevitch, 1985). Pronzato

improves on those methods by rigorously showing what type of penalty leads to convergence in parameter estimates and reward.

5 Conclusion

This review covers the fundamentals of sequential decision making and optimal experimental design, then presents three modern methods which consider if, when, and how to explore. Continuing to study novel sequential experimentation methods is important, notably in situations where exploration is expensive and minimizing regret is not the only goal (e.g., clinical trials for potentially life-saving therapies). Additionally, studying different exploration methods in online Q and V-learning is relevant.

For the methods discussed in Section 4, future research ideas are the following. For IDS, generalizing the methodology to consider other definitions of information gain is intriguing. Defining information gain through optimal experimental design is an obvious approach to consider. If covariate diversity is satisfied, looking at the efficiency of estimators based on data from greedy optimization versus other algorithms is particularly interesting and relevant to these high-cost situations. For Pronzato (2000), developing data-driven methods to determine α_t that still satisfy the convergence assumptions is of interest.

In the simulation section, we compare these three methods among others in a contextual bandit scenario. We propose a variant of IDS based on D-optimality, address the efficiency of parameter estimates, and examine how well we can assign interventions to new, out-of-experiment subjects with the same context distribution based on the information collected through different policies. In this study, the methods that consider if, when, and how to explore perform well minimizing regret, efficiently learning parameters, and assigning out-of-experiment subjects effective interventions.

References

Agrawal, S. and Goyal, N. (2012). Further optimal regret bounds for thompson sampling.

Alagoz, O., Hsu, H., Schaefer, A. J., and Roberts, M. S. (2009). Markov decision processes: A tool for sequential decision making under uncertainty. *Medical Decision Making*, 30(4):474–483.

Aström, K. J. and Wittenmark, B. (1995). *Adaptative control*. Addison Wesley.

Bastani, H., Bayati, M., and Khosravi, K. (2017). Mostly exploration-free algorithms for contextual bandits.

Bellman, R. (1957). *Dynamic programming*. Princenton University Press.

Berry, D. (2012). Adaptive clinical trials in oncology. *Nat Rev Clinical Oncology*, 9:199–207.

Chaloner, K. and Verdinelli, I. (1995). Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304.

Cheung, Y. K., Chakraborty, B., and Davidson, K. W. (2015). Sequential multiple assignment randomized trial (smart) with adaptive randomization for quality improvement in depression treatment program. *Biometrics*, 71(2):450–459.

Clifton, J. and Laber, E. B. (2020). Q-learning: Theory and applications. *Annual Review of Statistics and Its Applications*, 7(1):279–301.

Ertefaie, A. and Strawderman, R. L. (2018). Constructing dynamic treatment regimes over indefinite time horizons. *Biometrika*, 105(4):963–977.

Fedorov and Leonov (2019). *Optimal Design for Nonlinear Response Models*. CRC Press.

Fedorov, V. (1972). *The Theory of Optimal Experiments*. Academic Press.

John, F., Marsden, J. E., and Sirovich, L. (1989). *Adaptive Markov Control Processes*. Springer New York.

Korn, E. L. and Freidlin, B. (2017). Adaptive clinical trials: Advantages and disadvantages of various adaptive design elements. *JNCI: Journal of the National Cancer Institute*, 109(6).

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

Lei, H., Tewari, A., and Murphy, S. A. (2017). An actor-critic contextual bandit algorithm for personalized mobile health interventions.

Li, D., Raymond, L. R., and Bergman, P. (2020). Hiring as exploration. Working Paper 27736, National Bureau of Economic Research.

Luckett, D. J., Laber, E. B., Kahkoska, A. R., Maahs, D. M., Mayer-Davis, E., and Kosorok, M. R. (2020). Estimating dynamic treatment regimes in mobile health using v-learning. *Journal of the American Statistical Association*, 115(530):692–706.

Murphy, S. A. (2005). A generalization error for q-learning. *Journal of Machine Learning Research*, 6(37):1073–1097.

Nguyen-Thanh, N., Marinca, D., Khawam, K., Rohde, D., Vasile, F., Lohan, E. S., Martin, S., and Quadri, D. (2019). Recommendation system-based upper confidence bound for online advertising.

Pronzato, L. (2000). Adaptive optimization and d-optimum experimental design. *Ann. Statist.*, 28(6):1743–1761.

Radoslav Harman, L. F. (2019). *OptimalDesign: A Toolbox for Computing Efficient Designs of Experiments*.

Rose, E. J., Laber, E. B., Davidian, M., Tsiatis, A. A., Zhao, Y.-Q., and Kosorok, M. R. (2019). Sample size calculations for smarts.

Russo, D., Roy, B. V., Kazerouni, A., and Osband, I. (2017). A tutorial on thompson sampling. *CoRR*, abs/1707.02038.

Russo, D. and Van Roy, B. (2018). Learning to optimize via information-directed sampling. *Operations Research*, 66(1):230–252.

SAS (2014). *SAS/QC 13.2 User's Guide*.

Schulte, P. J., Tsiatis, A. A., Laber, E. B., and Davidian, M. (2014). Q-and a-learning methods for estimating optimal dynamic treatment regimes. *Statistical Science*, 29(4):640–661.

Schumann, C., Lang, Z., Foster, J. S., and Dickerson, J. P. (2019). Making the cut: A bandit-based approach to tiered interviewing. *CoRR*, abs/1906.09621.

Slivkins, A. (2019). Introduction to multi-armed bandits.

Smith, K. (1918). On the Standard Deviations of Adjusted and Interpolated Values of an Observed Polymonial Function and its Constants and the Guidance They Give Towards a Proper Choice of the Distribution of Observations. *Biometrika*, 12(1-2):1–85.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA.

Thall, P. and Wathen, K. (2007). Practical bayesian adaptive randomization in clinical trials. *European Journal of Cancer*, 43(5):859–866.

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294.

Tomkins, S., Liao, P., Klasnja, P., and Murphy, S. (2020). Intelligentpooling: Practical thompson sampling for mhealth.

Tsiatis, A. A., Davidian, M., Holloway, S. T., and Laber, E. B. (2020). *Dynamic treatment regimes statistical methods for precision medicine*. London.

Wathen, J. K. and Thall, P. F. (2017). A simulation study of outcome adaptive randomization in multi-arm clinical trials. *Clinical Trials*, 14(5):432–440. PMID: 28982263.

Watkins, C. J. C. H. and Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3-4):279–292.

Williamson, S. F. and Villar, S. S. (2019). A response-adaptive randomization procedure for multi-armed clinical trials with normally distributed outcomes. *Biometrics*, 76(1):197–209.

Wittenmark, B. (1975). An active suboptimal dual controller for systems with stochastic parameters. *Automatic Control Theory and Application*, 3:13–19.

Wittenmark, B. and Elevitch, C. (1985). An adaptive control algorithm with dual features. *IFAC Proceedings Volumes*, 18(5):587–592.

Wynn, H. P. (1970). The sequential generation of d-optimum experimental designs. *Ann. Math. Statist.*, 41(5):1655–1664.

Yekkehkhany, A., Arian, E., Hajiesmaili, M., and Nagi, R. (2019). Risk-averse explore-then-commit algorithms for finite-time bandits.

Zeng, C., Wang, Q., Mokhtari, S., and Li, T. (2016). Online context-aware recommendation with time varying multi-armed bandit. page 2025–2034.

Zhang, K. W., Janson, L., and Murphy, S. A. (2020). Inference for batched bandits.