Bootstrap Thompson Sampling and sequential decision problems in the behavioral sciences

Eckles, Dean; Kaptein, Maurits

Published in:
Sage Open

DOI:
10.1177/2158244019851675

Publication date:
2019

Document Version
Publisher's PDF, also known as Version of record

SAGE Open
April-June 2019: 1–12
© The Author(s) 2019
DOI: 10.1177/2158244019851675
journals.sagepub.com/home/sgo
SAGE

Bootstrap Thompson Sampling and Sequential Decision Problems in the Behavioral Sciences

Dean Eckles[1] and Maurits Kaptein[2]

Abstract

Behavioral scientists are increasingly able to conduct randomized experiments in settings that enable rapidly updating probabilities of assignment to treatments (i.e., arms). Thus, many behavioral science experiments can be usefully formulated as sequential decision problems. This article reviews versions of the multiarmed bandit problem with an emphasis on behavioral science applications. One popular method for such problems is Thompson sampling, which is appealing for randomizing assignment and being asymptoticly consistent in selecting the best arm. Here, we show the utility of bootstrap Thompson sampling (BTS), which replaces the posterior distribution with the bootstrap distribution. This often has computational and practical advantages. We illustrate its robustness to model misspecification, which is a common concern in behavioral science applications. We show how BTS can be readily adapted to be robust to dependent data, such as repeated observations of the same units, which is common in behavioral science applications. We use simulations to illustrate parametric Thompson sampling and BTS for Bernoulli bandits, factorial Gaussian bandits, and bandits with repeated observations of the same units.

Keywords

multiarmed bandits, Thompson sampling, online learning, bootstrapping, model misspecification, dependent data

Introduction

Many investigations in and applications of behavioral science involve assigning units (e.g., people, organisms, groups) to treatments. For example, experiments frequently randomly assign people to different treatments with the aim of learning about the effects of those treatments and thereby testing theory. The resulting scientific knowledge is often subsequently applied by changing policies (i.e., treatment assignment rules, guidelines) employed by practitioners, firms, and governments. These are often quite distinct steps, whereby behavioral scientists conduct multiple experiments, publish papers reporting on these, and thereby influence policies. However, behavioral scientists are increasing able to more rapidly iterate on experiments—including field experiments—such that what has been learned from an experiment so far may be used to modify that experiment moving forward to, for example, focus on increasing precision where it is most needed or to allocate units according to a policy that appears better. Such experiments in the behavioral sciences can be seen as instances of a *sequential decision problem* in which units are allocated to treatments over time, with the possibility of information from earlier units outcomes being used for later allocations.

In this article, we review a particular class of sequential decision problems, multiarmed bandit problems, and contextual multiarmed bandit problems, in which units are each assigned to one of a number of distinct treatments (i.e., arms), perhaps according to covariates (i.e., context) that is observed about those units, with the aim of maximizing a random outcome affected by treatment (i.e., the reward). We present a pair of methods for solving these problems, Thompson sampling and bootstrap Thompson sampling (BTS), which have appealing properties for behavioral science applications. This article is structured as follows. We first formalize these problems and give some examples from applied behavioral sciences. We then introduce the algorithms for solving these problems and illustrate them applied to simulated data having characteristics

[1]Massachusetts Institute of Technology, Cambridge, USA
[2]Jheronimus Academy of Data Science, 's-Hertogenbosch, The Netherlands

Corresponding Author:
Dean Eckles, MIT Sloan School of Management, Massachusetts Institute of Technology, 77 Massachusetts Ave, Cambridge, MA 02139, USA.
Email: eckles@mit.edu

common in behavioral science applications (e.g., model misspecification, dependent data).

Multiarmed Bandit Problems

In multiarmed bandit problems, a set of actions have potentially heterogeneous stochastic payoffs and an experimenter aims to maximize the payoff over a sequence of selected actions (Macready & Wolpert, 1998; Whittle, 1980). Multiarmed bandit problems can be formalized as follows. At each time $t = 1,\ldots,T$, we have a set of possible actions (i.e., arms, treatments) $\mathcal{A}$. After choosing $a_t \in \mathcal{A}$ we observe reward r_t. The aim is to select actions so as to maximize the cumulative reward $\mathcal{R}_c = \Sigma_{t=1}^{T} r_t$. Solutions typically involve balancing learning about the distribution of rewards for each arm (exploration) and choosing the arms that appear better given prior learning (exploitation; Audibert, Munos, & Szepesvári, 2009; Garivier & Cappé, 2011; Scott, 2010). Methods that tilt toward the latter are often called "greedy."

Contextual Bandit Problems

In many social science applications, the outcome distribution likely depends on observable characteristics of the units being allocated. In the contextual bandit problem (Beygelzimer, Langford, Li, Reyzin, & Schapire, 2011; Bubeck, 2012; Dudík, Erhan, Langford, & Li, 2014; Langford & Zhang, 2008), the set of past observations $\mathcal{D}$ is composed of a triplet $\{x, a, r\}$, where the x denotes the context (i.e., covariates): additional information that is observed prior to the decision, rather than assigned by the experimenter. There is currently substantial interest in heterogeneous treatment effects in the behavioral sciences (Grimmer, Messing, & Westwood, 2017; Kaptein & Eckles, 2012; Künzel, Sekhon, Bickel, & Yu, 2019; Wager & Athey, 2018); when such heterogeneity is large enough that different units have different best arms, then it can be advantageous to think of the problem as a contextual bandit problem—or at least one of learning a policy (i.e., treatment rule or regime) that assigns different units to different arms (Manski, 2004).

Examples

Bandit problems appear throughout the behavioral sciences. Here, we briefly highlight examples in political science, medicine, and educational psychology.

Donation requests to political campaign email list. A political campaign has a list of email addresses of likely supporters and is trying to raise money for the campaign. Staffers have written several versions of emails to send to supporters and there are many different photos of the candidate to use in those emails. The campaign can randomize which variant on the email is sent to a supporter and observe how much they donate.

Chemotherapy following surgery. Following surgery for colon cancer, some guidelines recommend adjuvant chemotherapy, but there is substantial uncertainty about which patients should be given chemotherapy (Gray et al., 2007; Verhoeff, van Erning, Lemmens, de Wilt, & Pruijt, 2016). For example, should older patients still be given chemotherapy? Continuing to randomize treatment of some types of patients even as the best treatment for other types is known could help discover improved treatment guidelines that reduce, for example, 5-year mortality.

Psychological interventions in online courses. Interventions designed to increase motivation and planning for overcoming obstacles are sometimes used in educational settings, including online courses where students begin and complete the course at their own pace. There are many variations on these interventions and students may respond differently to these variations. For example, motivational interventions might work differently for students from collectivist versus individualist cultures (Kizilcec & Cohen, 2017). The learning software can randomize students to these interventions while learning which interventions work for (e.g., result in successful course completion) different types of students.

How should we solve these bandit problems? Many possible solutions to bandit problems have been suggested (Auer & Ortner, 2010; Garivier & Cappé, 2011; Gittins, 1979; Whittle, 1980) and, while in some cases there are exactly optimal methods available, these are often not readily extensible to realistic versions of these problems. Thus, popular solutions are fundamentally heuristic methods, though many are approximately (e.g., asymptotically) optimal. We now turn to the two methods we focus on in this review: Thompson sampling and a recent variant, BTS.

Thompson Sampling

Recently, there has been substantial interest in Thompson sampling (Agrawal & Goyal, 2012; Gopalan, Mannor, & Mansour, 2014; Thompson, 1933). The basic idea of Thompson sampling is straightforward: randomly select an action a at time t with the probability that it is optimal (e.g., leading to the greatest expected reward) according to your current beliefs. The set of past observations $\mathcal{D}$ consists of the actions $a_{(1,\ldots,t)}$ and the rewards $r_{(1,\ldots,t)}$. The rewards are modeled using a parametric likelihood function: $P(r \mid a, \theta)$ where θ is a set of parameters. Using Bayes rule, it is, in some problems, easy to compute or sample from $P(\theta \mid \mathcal{D})$. Given that we can work with $P(\theta \mid \mathcal{D})$, we can select an action according to its probability of being optimal:

$$\int \mathbf{1}\left[\mathbb{E}(r \mid a, \theta) = \max_{a'} \mathbb{E}(r \mid a', \theta)\right] P(\theta \mid \mathcal{D}) d\theta \qquad (1)$$

where $\mathbf{1}$ is the indicator function. It is not necessary to compute the above integral: It suffices to draw a random sample θ^* from the posterior at each round and select the action

with the greatest expected reward according to the current draw.

Thompson sampling asymptotically achieves the optimal performance limit for Bernoulli payoffs (Kaufmann, Korda, & Munos, 2012; Lai & Robbins, 1985). Empirical analyses of Thompson sampling, also in problems more complex than the Bernoulli bandit, show performance that is competitive to other solutions (Chapelle & Li, 2011; Scott, 2010).

When it is easy to sample from $P(\theta \mid \mathcal{D})$, Thompson sampling is easy to implement. However, to be practically feasible for many problems, and thus scalable to large T or to complex likelihood functions, Thompson sampling requires computationally efficient sampling from $\mathrm{P}(\theta \mid \mathcal{D})$. In practice $P(\theta \mid \mathcal{D})$ might not always be easy to sample from: already in situations in which a logit or probit model is used to model the expected reward of the actions, $P(\theta \mid \mathcal{D})$ is not available in closed form and is then often computed using Markov chain Monte Carlo (MCMC) methods or otherwise approximated, which can be computationally costly. Furthermore, for many likelihood functions, it is not straightforward to exactly update the posterior online (i.e., row-by-row) thus requiring reinspection of the full data set $\mathcal{D}$ at each iteration.

Thompson Sampling for Contextual Bandit Problems

In the contextual bandit case, Equation 1 becomes

$$\int \mathbf{1}\left[\mathbb{E}\left(r \mid a, x, \theta\right) = \max_{a'} \mathbb{E}\left(r \mid a', x, \theta\right)\right] P\left(\theta \mid \mathcal{D}\right) d\theta \qquad (2)$$

In this specification, and with rewards $r_t \in \{0,1\}$, one would often model $P(\theta \mid \mathcal{D})$ with a probit or logistic regression. No general closed form for $P(\theta \mid \mathcal{D})$ exists. Thus, one would resort to MCMC methods. At each time t when a decision needs to be made, this can be computationally costly as the chain has to converge, but even more cumbersome is the fact that no online update is available. To produce a sample from the posterior at time t the algorithm will revisit all data $\mathcal{D}_{(t=1,\ldots,t=t)}$ giving a computational complexity of $O(t)$ at each update.[1] In particular cases, other methods for approximate Bayesian computation will be available. However, most current presentations of Thompson sampling make use of conjugacy relationships, MCMC, or problem-specific formulations or approximations (Chapelle & Li, 2011; Scott, 2010).

Consider, for example, the problem of adaptively selecting persuasive messages where the outcome of interest is a click or a conversion, such as donation to a political campaign (Gibney, 2018) or purchase of a product (Hauser & Urban, 2011; Hauser, Urban, Liberali, & Braun, 2009; Kaptein & Eckles, 2012; Kaptein, Markopoulos, de Ruyter, & Aarts, 2015; Kaptein, McFarland, & Parvinen, 2018). Such efforts often generate data that have a complex, cross-random effects structure (Ansari & Mela, 2005). In these cases, Thompson sampling is computationally challenging, and effectively implementing Thompson sampling often requires a problem- or model-specific approach and considerable engineering work.

We now turn to a variation on Thompson sampling that is computationally feasible with such models and with dependent data.

Bootstrap Thompson Sampling

BTS replaces samples from the Bayesian posterior $P(\theta \mid \mathcal{D})$ with the bootstrap distribution of the point estimate $\hat{\theta}$ of the mean reward for each arm. This can have practical, computational, and robustness advantages compared with Thompson sampling—both in general and specifically for behavioral science applications, as we describe below. Several recent papers (Bietti, Agarwal, & Langford, 2018; Eckles & Kaptein, 2014; Lu & Van Roy, 2017; Osband & Van Roy, 2015) have considered versions of BTS.

Bootstraps and Bayesian Posteriors

Bootstrap methods are widely used to quantify uncertainty in statistics. The standard bootstrap distribution (Efron, 1979) is the distribution of estimates of θ on samples of size N formed by sampling from the original N data points with replacement. However, contemporary bootstrap methods are especially computationally appealing. In particular, streaming bootstrap methods that involve randomly reweighting data (Praestgaard & Wellner, 1993; Rubin, 1981), rather than resampling data, can be conducted online (Chamandy, Muralidharan, Najmi, & Naidu, 2012; Lee & Clyde, 2004; Owen & Eckles, 2012; Oza, 2001).

Statisticians have long noted relationships between bootstrap distributions and Bayesian posteriors. With a particular weight distribution and nonparametric model of the distribution of observations, the bootstrap distribution and the posterior coincide (Rubin, 1981). In other cases, the bootstrap distribution can be used to approximate a posterior, for example, as a proposal distribution in importance sampling (Efron, 2012; Newton & Raftery, 1994). Moreover, sometimes the difference between the bootstrap distribution and the Bayesian posterior is that the bootstrap distribution is more robust to model misspecification, such that if they differ substantially, the bootstrap distribution may even be preferred (Eckles & Kaptein, 2014; Liu & Rubin, 1994; Szpiro, Rice, & Lumley, 2010).

The computational benefits of reweighting bootstrap methods, their robustness, and their direct link to Bayesian posteriors, suggest the use of reweighting bootstrap methods

in sequential decision problems. In sequential decision problems, uncertainty quantification is a key element that drives the exploration behavior of decision policies.

BTS replaces the posterior $P(\theta \mid \mathcal{D})$ by an online bootstrap distribution of the point estimate $\hat{\theta}$, explained in detail below. Recently, alternative bootstrapping policies for bandit problems have been proposed (Baransi, Maillard, & Mannor, 2014), however, we consider explicitly an online bootstrap of the point estimate as a computationally feasible approach for complex data sets. Following a preprint version of this article, BTS has been used by various researchers as an applied method for balancing exploration and exploitation in a computationally feasible way (Bietti et al., 2018; Osband & Van Roy, 2015); however, the examination of the performance of BTS in cases of model misspecification or dependencies between observations remains otherwise unexplored. In the remainder of this article, we provide further details on BTS, our proposed implementation, and provide a number of empirical evaluations.

BTS Algorithm

In contrast to Thompson sampling, as specified in Equation 1, BTS replaces the posterior $P(\theta \mid \mathcal{D})$ by an online bootstrap distribution of the point estimate $\hat{\theta}$. Specifically, for BTS, we propose a bootstrap method in which, for each bootstrap replicate $j \in \{1,\ldots,J\}$, each observation t gets a weight $w_{tj} \sim 2 \times \text{Bernoulli}(1/2)$. Inline with prior work, we refer to this reweighting bootstrap as the *double-or-nothing bootstrap* (DoNB) or *online half-sampling* (McCarthy, 1969; Owen & Eckles, 2012). As the absolute scale of the weights does not matter for most estimators, it is equivalent to have the weights be 0 or 1, rather than 0 or 2.

Remark 1. Other weight distributions with mean 1 and variance 1 could be used for various reasons. For example, using Exp(1) weights is the so-called Bayesian bootstrap (Lee & Clyde, 2004; Rubin, 1981). In that case, each observation has positive weight in each replicate, which can avoid numerical problems in some settings, but requires updating all replicates for each observation. Poisson(1) weights, as an approximation to the standard bootstrap's multinomial sampling, have been a common choice (Oza, 2001). However, for a given number of replicates, double-or-nothing weights have the highest precision in estimating the variance of a mean (Owen & Eckles, 2012). Compared with Exp(1) and Poisson(1), it also has to update the fewest replicates for each observation, making it computationally preferable to other weight distributions.

Remark 2. The replicates can be initialized and updated in other ways, including diverging from weights with mean 1 and variance 1. Lu and Van Roy (2017) analyze initializing each replicate with a draw from a prior. Bietti et al. (2018) propose including one replicate fit to the full data set to make BTS greedier. More generally, we can expect that using lower variance weights would make BTS greedier.

Algorithm 1 details BTS for the contextual bandit problem. At each step t, the algorithm first chooses the best arm according to a single, uniformly randomly selected, bootstrap replicate.

Note that in large-scale applications, requests for which arm to use can be randomly routed to different servers with different replicates. Then the algorithm updates a random subset of the replicate models using the observed data (a_i, r_t), which can involve routing subsets of observed data to each replicate. This is one concrete way that Algorithm 1 can be implemented in parallel. The model updating in practice will often constitute an online update of point estimates of the model parameters (e.g., using stochastic gradient descent to update row-by-row the point estimates of the parameters in of a logistic regression model); BTS will be particularly useful if such an online update is easy to compute, while the posterior is hard to sample from (which would render Thompson sampling infeasible).

For BTS, as long as $\hat{\theta}$ can be computed online, which is often possible even when $P(\theta \mid \mathcal{D})$ cannot be updated and sampled from online, the computational complexity of getting an updated sample at time t is $O(J) = O(1)$ as J is a constant. This contrasts favorably with the analysis above of Thompson sampling when sampling from the posterior using MCMC is necessary.

Remark 3. In contextual bandit problems, it is often desirable to compute the probability that a unit is assigned to a given arm; these design-based propensity scores may then be used in subsequent analysis using inverse-probability-weighted or doubly robust estimators (Dudík et al., 2014). In BTS, the propensity score is the fraction of replicates in which a has the highest expected reward. This can be easily accomplished by modifying to Algorithm 1 to compute predicted rewards for all arms for each of the J replicates.

Properties and Performance of BTS

What can we say about the performance of BTS analytically? Osband and Van Roy (2015) have shown an equivalence of BTS to Thompson sampling when using Exp(1) weights with a Bernoulli bandit; such a version of BTS would thus inherit established results for Thompson sampling described above. Lu and Van Roy (2017) prove that in a linear bandit problem the expected regret BTS is larger than that of Thompson sampling by a term that can be made arbitrarily small by increasing J.

There are empirical evaluations of BTS that show its promise in comparison with Thompson sampling and other methods. Eckles and Kaptein (2014) compared BTS with

Algorithm 1 Bootstrap Thompson Sampling

Require: M_{init} : initial model state, where the model is a function from an action $a \in \mathcal{A}$ to a predicted reward. J : Number of bootstrap replicates.

// Initialize:
for $j = 1, \ldots, J$ **do**
  $M_j = M_{\text{init}}$
end for
// Do sequential allocation:
for $t = 1, \ldots, T$ **do**
  Observe context x_t
  Sample $j \sim \text{Uniform}(1, \ldots, J)$
  for $i = 1, \ldots, K$ **do**
    Compute predicted reward $M_j(x_t, a_i)$
  end for
  Play arm $\hat{i} = \text{argmax}_i M_j(x_t, a_i)$ and observe r_t
  for $j = 1, \ldots, J$ **do**
    Sample $d_j \sim \text{Bernoulli}(1/2)$
    if $d_j = 1$ **then**
      Update M_j with (x_t, a_i, r_t)
    end if
  end for
end for

Thompson sampling, illustrating its robustness to model misspecification. Bietti et al. (2018) included BTS and greedier variants in an empirical evaluation using conversion of classification problems to contextual bandit problems, where it performed competitively. We further illustrate the performance of BTS below. In particular, a novel contribution of the present article is to illustrate the straightforward use of versions of BTS for dependent data, which frequently arises in the behavioral science, as suggested by Eckles and Kaptein (2014).

Alternatives

Here, we briefly discuss some alternative solutions to bandit problems besides "vanilla" Thompson sampling and BTS. Other papers (Dudík et al., 2014) and textbooks (Sutton & Barto, 2018) provide a broader overview.

Thompson Sampling With Intractable Likelihoods

When Thompson sampling is computationally impractical, such as when sampling from the posterior would require using MCMC, there are sometimes other alternatives available. In some specific cases, sequential Monte Carlo and other online methods (Carvalho, Johannes, Lopes, & Polson, 2010) may be available (Chapelle & Li, 2011; Scott, 2010) when complex models are considered. However, substantial engineering work may still be required to develop custom Monte Carlo algorithms to efficiently sample from the posterior distribution—above that for just computing a point estimate of the expected reward. Particular problems may involve one-off engineering to, for example, combine MCMC with other approximations for parts of the likelihood, as when Laber et al. (2018) consider a sequential decision problem in the presence of spatial contagion.

Greedier Methods

Solutions to sequential decision problems often try to balance exploration and exploitation. Thompson sampling may sometimes overexplore; that is, it is dominated in some settings by other "greedier" methods that exploit existing information more readily. Bastani and Bayati (2015) argue that many contextual bandit problems provide enough implicit exploration through randomness in the context that explicit exploration is often suboptimal. Closer to this review, Bietti et al. (2018) evaluate contextual bandit algorithms, noting the benefits of being greedy compared with typically used algorithms. In particular, they evaluate versions of BTS that include among the bootstrap replicates estimates using the full data set; this makes BTS greedier.[2] Bietti et al. (2018) find that such versions of BTS perform well, notably with very few bootstrap replicates (e.g., $J \in \{4, 8, 16\}$).

Illustration via Simulations

To empirically examine the performance of BTS, we run a number of simulations studies. First, for didactic purposes, we demonstrate how BTS is implemented for the simple k-armed Bernoulli bandit problem. Next, we demonstrate the robustness of BTS to model misspecification and illustrate its usefulness when there are dependencies between observations.

Bernoulli Bandit

A commonly used example of a bandit problem is the K-armed Bernoulli bandit problem, where $r_t \in \{0, 1\}$, and the action a is to select an arm $i \in \{1, \ldots, K\}$ at time t. The reward of the i-th arm follows a Bernoulli distribution with true mean θ_i. The implementation of standard Thompson sampling using Beta priors for each arm is straightforward: For each arm i one initializes a beta-Bernoulli model and at each round one obtains a single draw θ_i^* from each of the beta posteriors, plays the arm $\hat{i} = \text{argmax}_i \theta_i^*$, and subsequently uses the observed reward r_t to update the Beta posterior; for a full description and examination of the empirical performance of this policy, see Chapelle and Li (2011). The BTS implementation is similar, but instead of using a beta-Bernoulli model to compute $P(\theta \mid \mathcal{D})$, we use the DoNB to obtain a sample from the bootstrap distribution $\tilde{\theta}$; that is, from the bootstrap distribution for each arm, $\tilde{\theta}_i$, we obtain a draw $\tilde{\theta}_i^*$ and again play the arm $\hat{i} = \text{argmax}_i \tilde{\theta}_i^*$.

Note that in this simple k-armed Bernoulli bandit case Thompson sampling is extremely easy to carry out using conjugate beta priors for each arm and Thompson sampling is computationally more efficient than BTS. We use the

Algorithm 2 BTS for the K -armed Bernoulli bandit

Require: α , β : prior parameters.
J: Number of bootstrap replicates.
// Initialize:
for $j = 1,\ldots,J$ **do**
  for $i = 1,\ldots,K$ **do**
    $(\alpha_{ij},\beta_{ij}) = (\alpha,\beta)$
  end for
end for
// Do sequential allocation:
for $t = 1,\ldots,T$ **do**
  // Select the best arm according to one replicate:
  for $i = 1,\ldots,K$ **do**
    Sample $j_i \sim \text{Uniform}(1,\ldots,J)$
    Retrieve $\alpha_{ij_i},\beta_{ij_i}$ and compute
    $\hat{\theta}_{ij_i} = \alpha_{ij_i} / (\alpha_{ij_i} + \beta_{ij_i})$
  end for
  Play arm $\hat{i} = \text{argmax}_i \hat{\theta}_{ij_i}$ and observe reward r_t
  // Update random subset of replicates (DoNB):
  for $j = 1,\ldots,J$ **do**
    Sample $d_j \sim \text{Bernoulli}(1/2)$
    if $d_j = 1$ **then**
      $\alpha_{\hat{i}j} = \alpha_{\hat{i}j} + r_t$
      $\beta_{\hat{i}j} = \beta_{\hat{i}j} + (1 - r_t)$
    end if
  end for
end for

k-arm Bernoulli bandit here because it provides a convenient didactic problem to illustrate both Thompson sampling and BTS.

BTS for the k-armed Bernoulli bandit is given in Algorithm 2; we write this version out explicitly to demonstrate its implementation. We start with an initial belief about the parameter θ_i by setting $\alpha_{ij} = 1$ and $\beta_{ij} = 1$ for each arm i and each bootstrap replicate j. To decide on an action, for each arm i, we uniformly randomly draw one of the J replicates j_i, and play arm $\hat{i}$ with the largest point estimate $\hat{\theta}_{ij_i} = \alpha_{ij_i} / (\alpha_{ij_i} + \beta_{ij_i})$, breaking ties randomly.[3]

After observing reward r_t , we update each bootstrap replicate j with probability 0.5. Note that here each α can be considered a counter of the number of successes, and β of failures, which are kept in memory; this is a convenient parametrization for this specific problem. The same results are, however, obtained by reparametrisizing the problem into the point estimate $\hat{\theta} = \alpha / \alpha + \beta$ and $n = \alpha + \beta$, and by updating each of these online using a DoNB scheme. This latter parametrization is closer to our general algorithm introduced above.

The choice of J limits the number of samples we have from the bootstrap distribution $\tilde{\theta}$. For small J, BTS is expected to become greedy: if in all combinations of the J replicates, some arm i does not have the largest point estimate, then this arm has zero probability of being played until this changes. In such a case, BTS could tend to overexploit. A large J more accurately approximates the true bootstrap distribution, and while potentially more computationally demanding, avoids this degree of overexploitation.

To examine the performance of BTS, we present an empirical comparison of Thompson sampling to BTS in the k-armed Bernoulli bandit case. In our simulations, the best arm has a reward probability of 0.5, and the k-1 other arms have a probability of 0.5-ϵ. We examine cases with $K \in \{10,100\}$ and $\epsilon \in \{0.02,0.1\}$. Figure 1 presents the empirical regret over time, $R_t = 0.5t - \Sigma_{t'=1}^{t}(r_{t'})$, of Thompson and BTS.[4] The results presented here are for $t = 1,\ldots,T = 10^6$ and averaged over 1000 simulation runs with $J = 1000$ bootstrap replicates. Also included is a version of BTS in which effectively the number of bootstrap replicates J is infinite; we obtain this by resampling from the full data set at each round.[5]

Clearly, the mean regret of BTS is similar to that of standard Thompson sampling. In some sets of simulations, BTS appears to have a lower empirical regret than Thompson sampling (and even lower than the optimal performance limit); however, this is because the use of a finite J makes BTS greedy, and this particular set of the number of simulations(1000) is insufficient to properly illustrate the expected performance of the method in this greedy case: in each of the 1000 replications, BTS became stuck on the "right" arm. The "infinite" version of BTS, however, has performance very closely comparable to Thompson sampling.

To further examine the importance of the number of bootstrap replicates J, Figure 2 presents the cumulative regret for the $K = 10$ and $\epsilon = 0.1$ with $J \in \{10,100,1000,10000,\infty\}$. Here, it is clear that in cases when J is small, the algorithm becomes too greedy and thus, in the worst case, suffers linear regret (which in this setting is clearly identified). J can thus be thought of as a tuning parameter for the BTS algorithm: with large ϵ , one might settle for lower values of J as arms are more easily separable and the chance of getting "stuck" on an incorrect arm is small (albeit still positive). If small values of ϵ are of interest, then a larger number of bootstrap replicates will be necessary. Similarly, if in a practical application, the horizon T is comparatively small, a small number of bootstrap replicates suffices: the performance of BTS before becoming greedy is similar to Thompson sampling. The tuning parameter J can thus also be evaluated in relation to the expected T in applied problems where for large T more replicates are necessary.

Robustness to Model Misspecification

We expected BTS to be more robust to some kinds of model misspecification, given the robustness of the bootstrap for statistical inference.[6] Bootstrap methods are often used in statistical inference for regression coefficients and predictions when errors may be heteroscedastic because the model fit for the conditional mean may be incorrect (Freedman, 1981). To examine this benefit of BTS, we

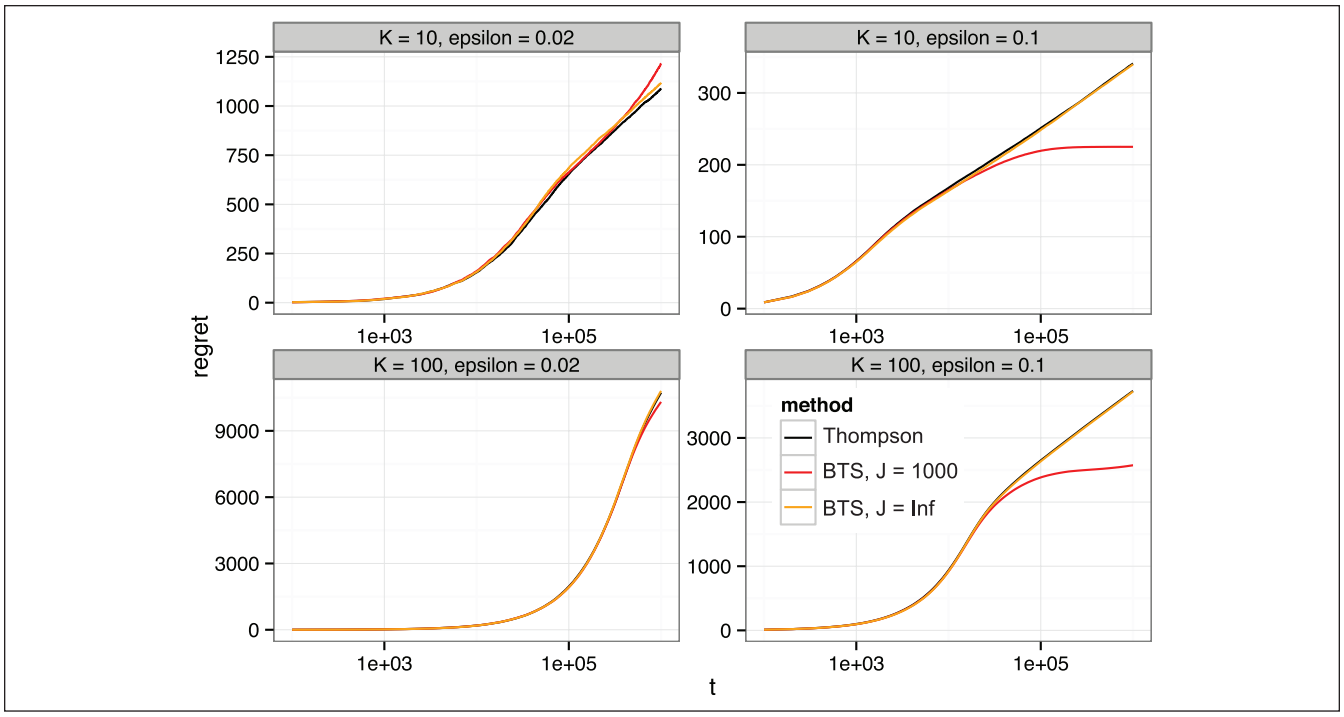

Figure 1. Empirical regret for Thompson sampling and BTS in a *K*-armed binomial bandit problem with varied differences between the optimal arm and all others ϵ.

Note. For BTS with $J = 1000$ bootstrap replicates, Algorithm 2 is used. BTS with $J = 1000$ sometimes shows lower mean regret than Thompson sampling when the arms are more different (i.e., $\epsilon = 0.1$). The lower empirical regret in this finite sample of simulations occurs as BTS with a finite J is greedier than Thompson sampling. For comparison, we also show the performance when J is effectively infinite (see main text); in this case, BTS and Thompson sampling have very similar performance. BTS = bootstrap Thompson sampling.

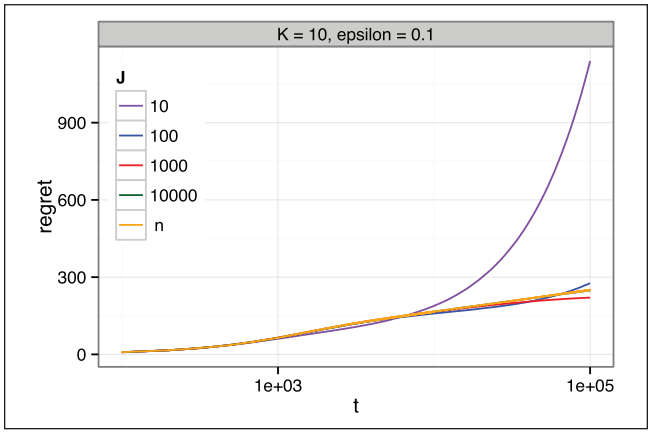

Figure 2. Comparison of empirical regret for BTS with varied number of bootstrap replicates J.

Note. Using a very small J (e.g., 10) results in an overgreedy method that can get "stuck" on nonoptimal arms, as the optimal arm may not win in any of the J replicates. Larger values of J give performance much more similar to that when J is effectively infinite. BTS = bootstrap Thompson sampling.

compare the performance of BTS and Thompson sampling in simulations of a factorial Gaussian bandit with heteroscedastic errors.

The data-generating process we consider here has three factors, $z_l = \{z_1, z_2, z_3\}$, with two levels $l \in \{0,1\}$ each. Thus, in our simulation $a \in \{1,\ldots,8\}$ referring to all 2^3 possible configurations. The true data generating model is $r = \mathbf{Z}\beta + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \mathbf{Z}\sigma^2)$. We use

$$\mathbf{Z} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}, \quad \beta = \begin{pmatrix} 1.00 \\ -0.20 \\ 0.10 \\ 0.20 \\ 0.10 \\ 0.05 \\ 0.10 \\ 0.01 \end{pmatrix}, \tag{3}$$

and $\sigma^2 = \{1,0,0,\gamma,0,0,0,0\}$. Here, $\mathbf{Z}$ is the design matrix, with each row corresponding to one of the eight arms, β is the vector of coefficients for the linear model including all interactions. Finally, we use σ^2 to denote the vector of

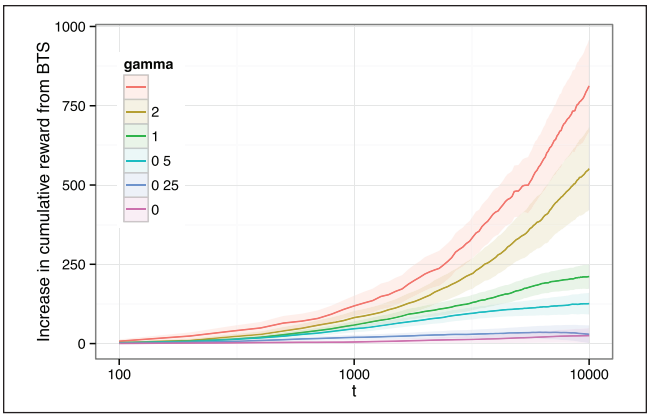

Figure 3. Comparison of Thompson sampling and BTS (with $J = 1000$) with a factorial design and continuous rewards with a heteroscedastic error distribution.
Note. Lines correspond to differing degrees of heteroscedasticity for $\gamma \in \{0, .25, .5, 1, 2, 4\}$. Increasing heteroscedasticity produces a larger differences between Thompson sampling and BTS. (Bands are point-wise 95% confidence intervals using a normal approximation.) BTS = bootstrap Thompson sampling.

variance components for each column of $\mathbf{Z}$. We vary γ to create different degrees of heteroscedasticity. Note that Arm 7, with an expected reward 1.40, is the optimal arm. The next best arm is Arm 8 with expected reward 1.36, while Arm 2, with an expected reward 0.8, is the worst arm.

We then compare the performance of Thompson sampling and BTS. For both Thompson sampling and BTS, we fit a full factorial linear model. For Thompson sampling, we fit a Bayesian linear model with Gaussian errors and a Gaussian prior with variance 1. We fit the linear model at each step to the full data set $\mathcal{D}$ consisting of $r_1,\ldots,r_{t'},x_1,\ldots,x_{t'}$ where x_t denotes the feature vector at time t. A draw from $P(\theta \mid \mathcal{D})$ to used to determine which action to take. For BTS, we use the online, summation-form implementation of obtaining point estimates of the parameters of a linear model (Chu et al., 2007) with a ridge penalty $\lambda = 1$.[7] We then follow Algorithm 1 with $J = 1000$ bootstrap replicates.

Figure 3 presents the difference in cumulative reward between BTS and Thompson sampling for $t = 1,\ldots,10^4$ for varying degrees of heteroscedasticity, $\gamma \in \{0, .25, .5, 1, 2, 4\}$, with 100 simulations. Even with a relatively small degree of misspecification (e.g., $\gamma = 0.5$) and with small t (e.g., $t = 1000$), BTS has substantially greater cumulative reward than Thompson sampling. As expected, this difference increases with γ.

Remark 4. It is well known that neglecting such heteroscedasticity can result in anticonservative inference; in this Bayesian context, the posterior for some arms would often be more concentrated than with a more flexible model.[8] Although here we have worked with a data-generating process that has heteroscedastic errors, similar anticonservative inference occurs when a parametric model is incorrectly specified (e.g., $\mathrm{E}[r \mid z]$ is nonlinear in z, but the model fit is linear in z). Such situations are also expected to be advantageous for BTS over Thompson sampling. Note that the performance of Thompson sampling can likely be improved in this setting by adopting a very conservative, large, prior variance, making the procedure less greedy. However, the current comparison shows that BTS attains this robustness without additional tuning.

Dependent Data

Bootstrap methods are easily adapted to use with dependent observations (e.g., repeated measures, time series, spatial dependence), and so are widely used for statistical inference in these settings, especially when this dependence is otherwise difficult to account for in inference (Cameron & Miller, 2015). Similarly, when observed rewards are dependent, BTS can easily be adapted to use an appropriate bootstrap method for dependent data. For example, if there are multiple observations of the same person, BTS can use a cluster-robust bootstrap (Davison & Hinkley, 1997) that reweights entire units. On the contrary, a typical Bayesian approach to such dependent data is a hierarchical (or random effects) model (Gelman et al., 2013). Not only can this present computational challenges, but these models only guarantee valid statistical inference when the specific model for dependence posited is correct, unlike the very general forms of dependence for which cluster-robust bootstraps are valid (Cameron & Miller, 2015; Owen & Eckles, 2012). In the case of multiple sources of dependence (i.e., crossed random effects), fitting crossed random effects models remains especially difficult to parallelize (Gao & Owen, 2016), but bootstrap methods remain mildly conservative (McCullagh, 2000; Owen, 2007; Owen & Eckles, 2012).

For an initial examination of value of BTS in cases in which the observations are dependent, we replicate the simulations above, but with the following changes: (a) we set $\gamma = 0$ to make the data-generating process homoscedastic, but (b) we now draw for each unit $u = 1,\ldots,1000$ a unit-specific (e.g., "person-specific") set of parameters $\beta_u \sim \mathcal{N}(\beta, \Sigma)$, where β is the vector of coefficients as given in Equation 3, and $\Sigma = \mathrm{diag}(\lambda^2)$ is the diagonal covariance matrix for the coefficients. We vary the degree of unit-specific heterogeneity by setting $\lambda \in \{0.10, 0.25, 1.00, 2.00\}$. We run 500 simulations for $t = 1,\ldots,10^4$. At each t, we uniformly randomly select unit u, leading to mean of 10 observations per unit per simulation run. Thus, the true generative model is a hierarchical model with unit-specific intercepts and effects.

Varying λ in this data-generating process leads to differences in the fraction of units for which the optimal arm is not the average optimal arm. To illustrate, for $\lambda = 0$, Arm 7 is the best arm for 100% of units, but for

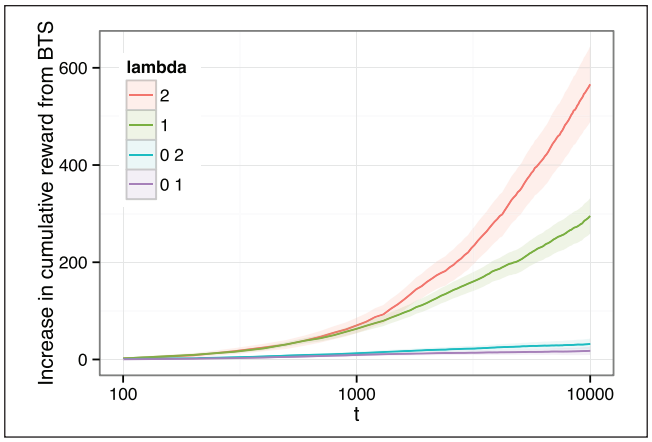

Figure 4. Comparison of Thompson sampling and BTS with dependent data, with BTS using a clustered bootstrap.
Note. As the dependence (controlled by λ) increases, Thompson sampling becomes too greedy, such that BTS with the clustered bootstrap has greater average reward. (Bands are pointwise 95% confidence intervals using a normal approximation.) BTS = bootstrap Thompson sampling.

$\lambda \in \{0.10, 0.25, 1.00, 2.00\}$, this is approximately 65, 52, 31, 22, 17, and 15% respectively.

Here, we implement BTS using a *clustered* DoNB. In particular, all observations of the same unit update the *same subset of the bootstrap replicates*; this can be accomplished by using a hashed unit identifier as the seed to a random number generator as demonstrated in Owen and Eckles (2012). In Algorithm 1, this corresponds to seeding the Bernoulli draw with hash(u). Note that is a very simple modification that provides a scalable and general solution to posited dependency structures in the observed data. Such an online clustered bootstrap is already widely used in the Internet industry for the analysis of nonadaptive experimentation [Bakshy & Eckles, 2013].

Figure 4 presents the results of our simulations. Thompson sampling is implemented as above, and thus does inference assuming the observations are independent, while BTS uses a bootstrap clustered by unit, requiring only that observations of different units are independent. As expected, for moderate and large values of λ, BTS significantly outperforms Thompson sampling. In these cases, Thompson sampling is clearly anticonservative and thus too greedy.

This result is unsurprising in that the model for Thompson sampling does not account for the dependence in the true data-generating process. Although a hierarchical model with varying coefficients could account for this particular case, implementing this with Thompson sampling would in practice require substantial engineering work. The adaptation of BTS for this dependent data, however, is very simple, and provides a very general approach to dealing with dependent observations in bandit problems.

The addition of unit-specific coefficients effectively makes the underlying problem a contextual bandit problem where each unit u specifies a context. This situation often occurs in applications to the Internet services where users may visit a service multiple times and the experimenter can choose multiple versions (possibly with a factorial structure) of various parts of the website (e.g., copy on multiple product pages). In such situations, one could model the context explicitly, either by allowing the model to vary for each user or using available features of the users. In the first case, one might use a hierarchical Bayesian model to estimate coefficients for each user (Cherkassky & Bornn, 2013). Such approaches would often require a more substantial engineering effort (especially with a very large number of unique arm–user pairs), but would be expected to be more effective than BTS, most notably in cases where users are observed many times. In the second case, even conditional on observed features, the data may still be dependent, so that Thompson sampling could still be anticonservative and using cluster-robust BTS would remain appealing.

Discussion

BTS relies on the same idea as Thompson sampling: heuristically optimize the exploration–exploitation trade-off by playing each action a with the probability of it being the best action. However, where Thompson sampling relies on a fully Bayesian specification to sample from $P(\theta \mid \mathcal{D})$, BTS replaces this distribution with a bootstrap distribution $\tilde{\theta}$. By using a reweighting bootstrap, such as the the DoNB used here, BTS can be implemented fully online whenever a point estimate can be updated online.

The practical appeal of BTS is in part motivated by computational considerations. The computational demands of MCMC sampling from $P(\theta \mid \mathcal{D})$ as needed in some cases for Thompson sampling quickly increases as $\mathcal{D}$ grows (e.g., t becomes large). The computation required for each round of BTS, however, need not depend on t and thus can be feasible even when t gets extremely large. This makes online BTS a good candidate for many real explore–exploit problems where a point estimate of θ can be obtained online, but $P(\theta \mid \mathcal{D})$ is hard to sample from. For example, because of the recent success of deep neural networks in prediction and reinforcement learning tasks, the model of rewards might be a network, where sampling from the posterior is difficult; the bootstrap, rather than dropout, is an appropriate method for quantifying uncertainty here (Osband, Blundell, Pritzel, & Van Roy, 2016; Osband & Van Roy, 2015).

We presented a number of empirical simulations of the performance of BTS in the canonical Bernoulli bandit and in Gaussian bandit problems with heteroscedasticity or dependent data. The Bernoulli bandit simulations allowed us to illustrate BTS and highlight the importance of selecting the number of bootstrap replicates J. The Gaussian bandit illustrated how BTS is robust to some forms of model

misspecification. Finally, simulations with dependent data, some versions of which (e.g., repeated observations of the same units) are common in applied problems, demonstrate how BTS can easily be made cluster-robust. Our proposed clustered DoNB scheme here provides a very general alternative when observations are dependent (or nested); a situation that is common in many applied bandit problems. We conclude that BTS is competitive in performance, more amenable to some large-scale problems, and, at least in some cases, more robust than Thompson sampling. The observation that BTS can overexploit when the number of online bootstrap replicates is too small needs further attention. The number of bootstrap replicates J can be regarded a tuning parameter in applied problems—just as the posterior in Thompson sampling can be artificially concentrated (Chapelle & Li, 2011)—or could be adapted dynamically as the estimates of the arms evolve; for example, replicates could be added or removed. Future work could also consider even more computationally appealing variations on the bootstrap, such as streaming subsampling methods (Chamandy et al., 2012) or the bag of little bootstraps (Kleiner, Talwalkar, Sarkar, & Jordan, 2014).

General Discussion

Widespread adoption of information and communication technologies has made it possible to adaptively assign, for example, people to different messages, experiences, interventions, and so on. This makes it felicitous to think of many empirical efforts in the applied behavioral sciences as sequential decision problems and often, more specifically, contextual multiarmed bandit problems. Thompson sampling is a particularly appealing solution for many of these problems and for behavioral scientists in particular; this is largely because, conceptually at least, it is readily extensible to many variations on the standard multiarmed bandit problem. Here, we accompanied our discussion of Thompson sampling with a variant, BTS, largely for this same reason: It allows applying the central idea of Thompson sampling to a wider range of models and settings without some of the computational, practical, and robustness issues that arise for Thompson sampling.

Behavioral and social scientists may find BTS particularly useful, especially as multiple software packages that implement BTS currently exist (Kruijswijk, van Emden, Parvinen, & Kaptein, in press; van Emden & Kaptein, 2018). Bandit problems in the behavioral and social sciences will frequently feature repeated observations of the same units (e.g., people) and possible model misspecification. For example, in economics and political science, there is a norm of using inferential methods, especially cluster-robust sandwich and bootstrap estimators of variance–covariance matrix that are robust to model misspecification and dependence among observations (Cameron & Miller, 2015). Other areas address this same problem in other ways (e.g., use of random effects models in psychology and linguistics [Baayen, Davidson, & Bates, 2008]). BTS makes achieving such robustness in sequential decision problems straightforward.

Acknowledgments

This work benefited from comments from Eytan Bakshy, Thomas Barrios, Daniel Merl, John Langford, Joris Mulder, John Myles White members of the Facebook Core Data Science team, and anonymous reviewers. Dean Eckles contributed to earlier versions of this article, while being an employee of Facebook, Inc.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: Dean Eckles had significant financial interests in Amazon, Netflix, Facebook, and GoFundMe while conducting this work. He was an employed by Facebook and Microsoft during the time he contributed to this article. He has received funding from Amazon, though this was not used for this work.

Notes

1. This specification removes all the constants required, for example, for Markov chain Monte Carlo (MCMC) chains to converge. In some cases, this might also depend on t, potentially increasing complexity.
2. Bietti, Agarwal, and Langford (2018) refer to BTS as "bagging," which is perhaps confusing as bagging stands for "bootstrap aggregation" and is an ensemble learning method that averages *predictions* over models fits to bootstrap replicates.
3. By randomly selecting a replicate for each arm, rather than matching replicate 1 for arm a with replicate 1 for arm a', this algorithm is similar to the "unpaired" version of bootstrapping a difference in means, which decreases Monte Carlo error from having a finite number of replicates J (Chamandy, Muralidharan, Najmi, & Naidu, 2012). In the absence of independence between our estimates for different arms, this strategy could result in a very different distribution than sampling a single replicate for all arms.
4. To decrease simulation error, in this computation we replace $0.5t$ with the observed reward for playing the optimal arm with the same random numbers.
5. We implement this by storing the sufficient statistics (the number of successes and failures) and resampling these at each round according to the *double-or-nothing bootstrap* (DoNB). This is possible in simple bandit problems such as the K-armed Bernoulli bandit case or when the number of unique combinations of arms and rewards is small. This is not intended as a proposed method, but is included as a tool for understanding BTS.
6. See Liu and Rubin (1994) for related remarks on the difference between the Bayesian posterior and the bootstrap distribution.
7. When the error variance is 1, the ridge point estimate is equivalent to the posterior mode with a Gaussian prior with variance 1 (Hastie, Tibshirani, & Friedman, 2008).

8. There is some work in developing Bayesian methods that allow for heteroscedasticity, including that resulting from model misspecification (Szpiro, Rice, & Lumley, 2010).

ORCID iD

Dean Eckles https://orcid.org/0000-0001-8439-442X

References

Agrawal, S., & Goyal, N. (2012). *Thompson sampling for contextual bandits with linear payoffs* (arXiv:1209.3352). Retrieved from https://arxiv.org/abs/1209.3352

Ansari, A., & Mela, C. F. (2005). E-customization. *Journal of Marketing Research*, *40*, 131-145.

Audibert, J.-Y., Munos, R., & Szepesvári, C. (2009). Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, *410*, 1876-1902.

Auer, P., & Ortner, R. (2010). UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, *61*, 55-65.

Baayen, R. H., Davidson, D. J., & Bates, D. M. (2008). Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language*, *59*, 390-412.

Bakshy, E., & Eckles, D. (2013). Uncertainty in online experiments with dependent data: An evaluation of bootstrap methods. In *Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1303-1311). New York, NY: Association for Computing Machinery.

Baransi, A., Maillard, O.-A., & Mannor, S. (2014). Sub-sampling for multi-armed bandits. In T. Calders, F. Esposito, E. Hllermeier, & R. Meo (Eds.), *Lecture Notes in Computer Science: Machine learning and knowledge discovery in databases* (Vol. 8724, pp. 115-131). Berlin, Germany: Springer.

Bastani, H., & Bayati, M. (2015). *Online decision-making with high-dimensional covariates* (Technical report). Retrieved from http://web.stanford.edu/~bayati/papers/lassoBandit.pdf

Beygelzimer, A., Langford, J., Li, L., Reyzin, L., & Schapire, R. E. (2011). Contextual bandit algorithms with supervised learning guarantees. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics (AISTATS 11)* (pp. 19-26). Retrieved from http://proceedings.mlr.press/v15/beygelzimer11a.html

Bietti, A., Agarwal, A., & Langford, J. (2018). *A contextual bandit bake-off* (arXiv:1802.04064). Retrieved from https://arxiv.org/abs/1802.04064

Bubeck, S. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, *5*, 1-122.

Cameron, A. C., & Miller, D. L. (2015). A practitioner's guide to cluster-robust inference. *The Journal of Human Resources*, *50*, 317-372.

Carvalho, C., Johannes, M. S., Lopes, H. F., & Polson, N. (2010). Particle learning and smoothing. *Statistical Science*, *25*, 88-106.

Chamandy, N., Muralidharan, O., Najmi, A., & Naidu, S. (2012). *Estimating uncertainty for massive data streams* (Technical report). Retrieved from https://pub-tools-public-publication-data.storage.googleapis.com/pdf/43157.pdf

Chapelle, O., & Li, L. (2011). An empirical evaluation of Thompson sampling. In *Advances in neural information processing systems* (pp. 2249-2257). Retrieved from https://papers.nips.cc/paper/4321-an-empirical-evaluation-of-thompson-sampling

Cherkassky, M., & Bornn, L. (2013). *Sequential Monte Carlo bandits*. Retrieved from http://arxiv.org/abs/1310.1404

Chu, C.-T., Kim, S. K., Lin, Y.-A., Yu, Y., Bradski, G., Ng, A. Y., & Olukotun, K. (2007). Map-reduce for machine learning on multicore. *Advances in Neural Information Processing Systems*, *19*, 281-288.

Davison, A. C., & Hinkley, D. (1997). *Bootstrap methods and their application*. Cambridge, UK: Cambridge University Press.

Dudík, M., Erhan, D., Langford, J., & Li, L. (2014). Doubly robust policy evaluation and optimization. *Statistical Science*, *29*, 485-511.

Eckles, D., & Kaptein, M. (2014). *Thompson sampling with the online bootstrap* (arXiv:1410.4009). Retrieved from https://arxiv.org/abs/1410.4009

Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, *7*, 1-26.

Efron, B. (2012). Bayesian inference and the parametric bootstrap. *The Annals of Applied Statistics*, *6*, 1971-1997.

Freedman, D. A. (1981). Bootstrapping regression models. *The Annals of Statistics*, *9*, 1218-1228.

Gao, K., & Owen, A. B. (2016). *Estimation and inference for very large linear mixed effects models* (arXiv:1610.08088). Retrieved from https://arxiv.org/abs/1610.08088

Garivier, A., & Cappé, O. (2011). The KL-UCB algorithm for bounded stochastic bandits and beyond. *JMLR: Workshop and Conference Proceedings*, *19*, 359-376.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian data analysis*. Boca Raton, FL: CRC Press.

Gibney, E. (2018, March 29). The scant science behind Cambridge Analytica's controversial marketing techniques. *Nature*. Retrieved from https://www.nature.com/articles/d41586-018-03880-4

Gittins, J. C. (1979). Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society: Series B (Methodological)*, *41*, 148-177.

Gopalan, A., Mannor, S., & Mansour, Y. (2014). Thompson sampling for complex online problems. In *Proceedings of the 31st international conference on machine learning* (pp. 100-108). Retrieved from http://proceedings.mlr.press/v32/gopalan14.html

Gray, R., Barnwell, J., McConkey, C., Hills, R. K., Williams, N. S., & Kerr, D. J. Quasar Collaborative Group. (2007). Adjuvant chemotherapy versus observation in patients with colorectal cancer: A randomised study. *The Lancet*, *370*, 2020-2029.

Grimmer, J., Messing, S., & Westwood, S. J. (2017). Estimating heterogeneous treatment effects and the effects of heterogeneous treatments with ensemble methods. *Political Analysis*, *25*, 413-434.

Hastie, T., Tibshirani, R., & Friedman, J. (2008). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). New York, NY: Springer.

Hauser, J. R., & Urban, G. L. (2011). *Website morphing 2.0: Technical and implementation advances combined with the first field experiment of website morphing*. Retrieved from https://pdfs.semanticscholar.org/1608/3251080efdcefb4518696f8a64ae09eb4f24.pdf

Hauser, J. R., Urban, G. L., Liberali, G., & Braun, M. (2009). Website morphing. *Marketing Science*, *28*, 202-223.

Kaptein, M., & Eckles, D. (2012). Heterogeneity in the effects of online persuasion. *Journal of Interactive Marketing*, *26*, 176-188.

Kaptein, M., Markopoulos, P., de Ruyter, B., & Aarts, E. (2015). Personalizing persuasive technologies: Explicit and implicit personalization using persuasion profiles. *International Journal of Human-Computer Studies*, *77*, 38-51.

Kaptein, M., McFarland, R., & Parvinen, P. (2018). Automated adaptive selling. *European Journal of Marketing*, *52*, 1037-1059.

Kaufmann, E., Korda, N., & Munos, R. (2012). Thompson sampling: An asymptotically optimal finite-time analysis. In N. H. Bshouty, G. Stoltz, N. Vayatis, & T. Zeugmann (Eds.), *Algorithmic learning theory* (pp. 199-213). Berlin, Germany: Springer.

Kizilcec, R. F., & Cohen, G. L. (2017). Eight-minute self-regulation intervention raises educational attainment at scale in individualist but not collectivist cultures. *Proceedings of the National Academy of Sciences of the United States of America*, *114*, 4348-4353.

Kleiner, A., Talwalkar, A., Sarkar, P., & Jordan, M. I. (2014). A scalable bootstrap for massive data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *76*, 795-816.

Kruijswijk, J., van Emden, R., Parvinen, P., & Kaptein, M. (in press). StreamingBandit; experimenting with bandit policies. *Journal of Statistical Software*.

Künzel, S. R., Sekhon, J. S., Bickel, P. J., & Yu, B. (2019). Meta-learners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences of the United States of America*, *116*, 4156-4165.

Laber, E. B., Meyer, N. J., Reich, B. J., Pacifici, K., Collazo, J. A., & Drake, J. M. (2018). Optimal treatment allocations in space and time for on-line control of an emerging infectious disease. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *67*, 743-770.

Lai, T. L., & Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, *6*, 4-22.

Langford, J., & Zhang, T. (2008). The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in neural information processing systems* (pp. 817-824). Retrieved from https://papers.nips.cc/paper/3178-the-epoch-greedy-algorithm-for-multi-armed-bandits-with-side-information

Lee, H. K. H., & Clyde, M. A. (2004). Lossless online Bayesian bagging. *Journal of Machine Learning Research*, *5*, 143-151.

Liu, J. S., & Rubin, D. B. (1994). Comment on "approximate Bayesian inference with the weighted likelihood bootstrap." *Journal of the Royal Statistical Society: Series B (Methodological)*, *56*, 3-48.

Lu, X., & Van Roy, B. (2017). Ensemble sampling. In *Advances in neural information processing systems* (pp. 3260-3268). Retrieved from https://papers.nips.cc/paper/6918-ensemble-sampling.pdf

Macready, W. G., & Wolpert, D. H. (1998). Bandit problems and the exploration/exploitation tradeoff. *IEEE Transactions on Evolutionary Computation*, *2*, 2-22.

Manski, C. F. (2004). Statistical treatment rules for heterogeneous populations. *Econometrica*, *72*, 1221-1246.

McCarthy, P. J. (1969). Pseudo-replication: Half samples. *Review of the International Statistical Institute*, *37*, 239-264.

McCullagh, P. (2000). Resampling and exchangeable arrays. *Bernoulli*, *6*, 285-301.

Newton, M. A., & Raftery, A. E. (1994). Approximate Bayesian inference with the weighted likelihood bootstrap. *Journal of the Royal Statistical Society: Series B (Methodological)*, *56*, 3-48.

Osband, I., Blundell, C., Pritzel, A., & Van Roy, B. (2016). Deep exploration via bootstrapped DQN. In *Advances in neural information processing systems* (pp. 4026-4034). Retrieved from https://papers.nips.cc/paper/6501-deep-exploration-via-bootstrapped-dqn

Osband, I., & Van Roy, B. (2015). *Bootstrapped thompson sampling and deep exploration* (arXiv:1507.00300). Retrieved from https://arxiv.org/abs/1507.00300

Owen, A. B. (2007). The pigeonhole bootstrap. *The Annals of Applied Statistics*, *1*, 386-411.

Owen, A. B., & Eckles, D. (2012). Bootstrapping data arrays of arbitrary order. *The Annals of Applied Statistics*, *6*, 895-927.

Oza, N. (2001). Online bagging and boosting. In *2005 IEEE international conference on systems, man and cybernetics* (Vol. 3, pp. 2340-2345). New York, NY: Institute of Electrical and Electronics Engineers.

Praestgaard, J., & Wellner, J. A. (1993). Exchangeably weighted bootstraps of the general empirical process. *The Annals of Probability*, *21*, 2053-2086.

Rubin, D. B. (1981). The Bayesian bootstrap. *The Annals of Statistics*, *9*, 130-134.

Scott, S. L. (2010). A modern Bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry*, *26*, 639-658.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.

Szpiro, A. A., Rice, K. M., & Lumley, T. (2010). Model-robust regression and a Bayesian "sandwich" estimator. *The Annals of Applied Statistics*, *4*, 2099-2113.

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

van Emden, R., & Kaptein, M. (2018). *Contextual: Evaluating contextual multi-armed bandit problems in R* (arXiv:1811.01926). Retrieved from https://arxiv.org/abs/1811.01926

Verhoeff, S., van Erning, F., Lemmens, V., de Wilt, J., & Pruijt, J. (2016). Adjuvant chemotherapy is not associated with improved survival for all high-risk factors in Stage II colon cancer. *International Journal of Cancer*, *139*, 187-193.

Wager, S., & Athey, S. (2018). Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, *113*, 1228-1242.

Whittle, P. (1980). Multi-armed bandits and the Gittins index. *Journal of the Royal Statistical Society: Series B (Methodological)*, *42*, 143-149.

Author Biographies

Dean Eckles is the KDD career development professor in Communications and Technology at Massachusetts Institute of Technology (MIT), an assistant professor in the MIT Sloan School of Management, and affiliated faculty at the MIT Institute for Data, Systems & Society.

Maurits Kaptein is professor of Data Science & Health at the University of Tilburg and principal investigator at the Jheronimus Academy of Data Science in Den Bosch. He is the author of multiple books, including Statistics for Data Scientists: An Introduction to Probability, Statistics, and Data Analysis, Persuasion Profiling: How the Internet Knows What Makes You Tick, and Hello World, Hello Computer (Hallo Wereld, Hallo Computer).