Online Bootstrapping for Efficient Exploration

Dean Eckles[1], **and Maurits Kaptein**[2]

[1] MIT Sloan School of Management, Marketing

[2] Tilburg University, Statistics and Research Methods

Abstract

Bootstrapping, in its various forms, often provides a computationally feasible approach to quantifying estimator uncertainty. Recent online (e.g., row-by-row) bootstrapping procedures are of special interest computationally, and the online clustered bootstrap has proven highly effective in dealing with dependent data. In this paper we explore the utility of using online, clustered, bootstrapping procedures in sequential decision problems; uncertainty quantification is a key element that drives the exploration behavior of many sequential decision policies. Specifically, we explore the utility of using an online bootstrap distribution as an alternative to the—often computationally challenging—full posterior distribution in a often used sequential decision policy called Thompson sampling. Thompson sampling provides a solution to bandit problems in which new observations are allocated to arms with the posterior probability that each arm is optimal. While sometimes easy to implement and asymptotically optimal, Thompson sampling is often computationally demanding in large scale bandit problems, and its performance is dependent on the parametric model fit to the observed data. We introduce *bootstrap Thompson sampling* (BTS), a generic black-box, heuristic method for solving bandit problems which modifies Thompson sampling by replacing the posterior distribution used in Thompson sampling by a bootstrap distribution.

1. Bootstrap Thompson sampling (BTS)

Bandit problems, in which a set of actions have varied stochastic payoff and an experimenter aims to maximize the payoff over a sequence of selected actions, are prevalent. For example, in online advertising the action of displaying a specific ad out of a set of multiple ads for the current visitor of the website can be regarded as a bandit problem: each ad has an uncertain payoff, and *a priori* the ad with the highest pay-off is unknown. The experimenter has to trade off exploration and exploitation: displaying ads — and observing the subsequent response — about which little is known (exploration) increases one's knowledge about the success rate of that ad. However, displaying ads which one already believes to be effective (exploitation) likely increases the overall payoff. Exploration and exploitation need to be balanced over the course of multiple interactions. Formally, bandit problems can be described as follows: at each time $t = 1, \ldots, T$, we have a set of possible actions $\mathcal{A}$. After choosing $a_t \in \mathcal{A}$ we observe reward r_t. The aim is to find a policy to select actions such that the cumulative reward $\mathcal{R}_c = \sum_{t=1}^{T} r_t$ is as large as possible.

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**
  $M_i = M_{\text{init}}$
end for
// Do sequential allocation:
for $t = 1, \ldots, T$ **do**
  // Select the best arm according to one replicate:
  Sample $j \sim \text{Uniform}(1, \ldots, J)$
  for $i = 1, \ldots, K$ **do**
    Compute predicted reward $M_j(a_i)$
  end for
  Play arm $\hat{\imath} = \arg\max_i M_j(a_i)$ and observe r_t
  // Update random subset of replicates:
  for $j = 1, \ldots, J$ **do**
    Sample $d_j \sim \text{Bernoulli}(1/2)$
    if $d_j = 1$ **then**
      Update M_j with $(a_{\hat{\imath}}, r_t)$
    end if
  end for
end for

The BTS Algrotihm for bandit problems, is presented in above. At each step t, the algorithm first chooses the best arm according to a single, uniformly randomly selected, bootstrap replicate. Then the algorithm updates a random subset of the replicate models using the observed data $(a_{\hat{\imath}}, r_t)$, which can involve routing subsets of observed data to each replicate. This is one concrete way that the algorithm can be implemented in parallel.

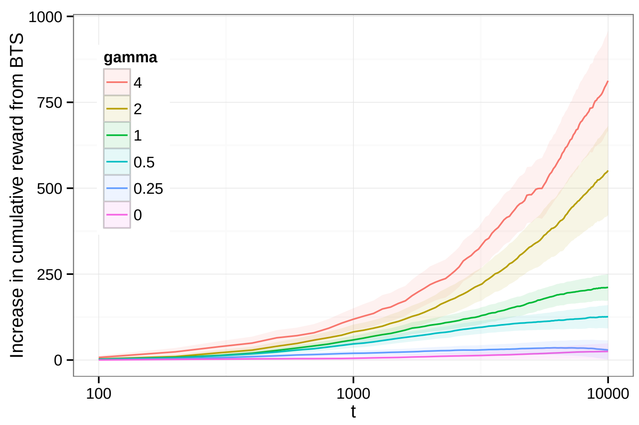
2. Heteroscedastic Errors

We expected BTS to be more robust to some kinds of model misspecification, given the robustness of the bootstrap for statistical inference. To examine this benefit of BTS we 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, $z_t = \{z_1, z_2, z_3\}$, 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

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

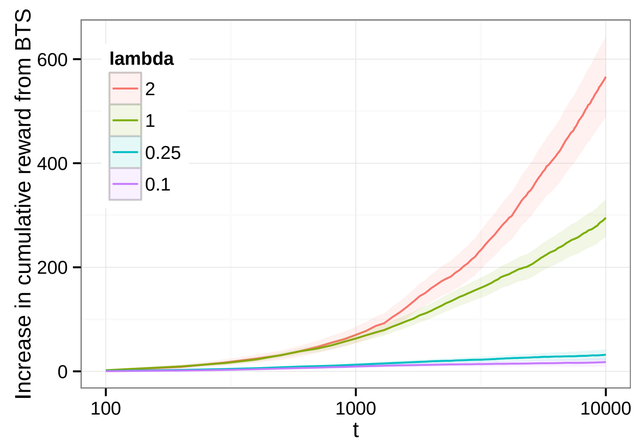
and $\sigma^2 = \{1, 0, 0, \gamma, 0, 0, 0, 0\}$. Here, $\mathbf{Z}$ is the design matrix, with each row corresponding to one of the 8 arms, β is the vector of coefficients for the linear model including all interactions. Finally, we use σ^2 to denote the vector of variance components for each column of $\mathbf{Z}$. We vary γ to create different degrees of heteroscedasticity.

The Figure 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 significantly greater cumulative reward than Thompson sampling. As expected, this difference increases with γ.



3. 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. For an initial examination of value of BTS in cases in which the observations are dependent, we replicate the previous simulation study, 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 1, and $\Sigma = \text{diag}(\lambda^2)$ is the diagonal co-variance 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 $\lambda \in \{0.10, 0.25, 1.00, 2.00\}$, this is approximately 65, 52, 31, 22, 17, and 15% respectively.



The Figure presents the results of our simulations. Thompson sampling is implemented as before, 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.

Conclusions

We have presented BTS as a computationally attractive and extremely flexible alternative to Thompson sampling to introduce exploration in sequential decision problems. The main advantages of BTS over competing policies are: I) BTS, by relying on the large body of theory on scalable and robust bootstrapping methods can deal with complex data structure (heteroscedasticity, dependencies, etc.) in a computationally attractive way, and II) BTS allows one to use *any* predictive model for which point estimates can efficiently be computed in large-scale bandit tests.

For more information please contact dean@deaneckles.com or m.c.kaptein@uvt.nl.
For details see: `https://arxiv.org/abs/1410.4009`