

Exact Bayesian A/B testing using distributed fault-tolerant Moore rejection sampler

Benny Avelin¹, and Raazesh Sainudiin^{1,2}

¹ AI & Analytics Centre of Excellence, Combient AB, Stockholm, Sweden

² Department of Mathematics, Uppsala University, Uppsala, Sweden

Box 480, Department of Mathematics, Uppsala University, 751 06

UPPSALA, Sweden

`raazesh.sainudiin@math.uu.se`

Keywords: trans-dimensional posterior samples, locally Lipschitz arithmetical expression, interval extended von Neumann rejection sampler, Apache Spark

A/B testing is ubiquitous in industrial data science problems today, whether it be comparing the efficacy of drug treatments or hit-rates for website alternatives. The traditional way to phrase an A/B test is to compare estimators of two populations. The estimators can be in the case of hit-rates, just estimating the probability θ in a binomial distribution, or it can be the mean value given some assumptions on the population distribution (usually Gaussian). The down-side to going down this traditional path is that the frequentist approach amounts to testing whether the two populations are the same or not. The Bayesian approach is another way of thinking about the problem. In this setting we are more concerned with which model is more probable given the data. That is, we compare the models in one class with those in another class by integrating the *a posteriori* probability of each class given *a priori* assumptions for each model and their parameters.

Let us begin with the simplest of the A/B tests, i.e. to determine whether the hit-rate differs between models A and B. Modeling hit-rate is usually done with the Binomial distribution, i.e. there is a probability θ_A of hit under model A and a possibly distinct probability θ_B under model B. The question now becomes: does the model which assumes that $\theta_A = \theta_B$ fit better or worse than the model with θ_A and θ_B estimated from trials under treatments A and B separately. This turns the problem into a so called trans-dimensional model selection problem. The reason for calling it trans-dimensional is simply that the model space for assuming $\theta_A = \theta_B$ is one dimensional when parametrized by the probability in $[0, 1]$. The other model space, i.e. θ_A and θ_B are possibly distinct, with $(\theta_A, \theta_B) \in [0, 1] \times [0, 1]$ is two dimensional. The Bayesian way of handling this would be to compute the posterior distribution and then sample from this distribution and compute

the estimated probabilities for the different models. The problem is now, how do we sample from this trans-dimensional model space? To do this we will be using the Moore rejection sampler developed by the second author in [1] with a non-distributed C++ implementation in `mrs2` [2]. This provides us with a way to perform exact posterior sampling from the trans-dimensional model space even when the posterior does not have analytic solution (under conjugacy assumptions on prior and posterior distributions) using interval enclosures of the posterior densities over all the models.

In this work we extend this Moore rejection sampler to the distributed fault-tolerant setting of Apache Spark, a unified engine for big data processing [3]. This extension allows us to obtain posterior samples in parallel for realistic industrial-scale workloads and naturally integrate with Apache Spark’s SparkSQL, MLlib and GraphX libraries for structured querying, machine learning and distributed vertex programming, respectively. Moreover, we can not only obtain posterior samples from more than two treatments but also from trans-dimensional posterior densities that can be specified by any locally Lipschitz arithmetical expression. Furthermore, depending on the memory and processor constraints on each commodity compute node in the Spark cluster, we can handle data streaming in bursts through time.

References

- [1] R. Sainudiin and T. York. An auto-validating, trans-dimensional, universal rejection sampler for locally Lipschitz arithmetical expressions. *Reliable Computing*, 18:15–54, 2013.
- [2] Raazesh Sainudiin, Thomas York, Jennifer Harlow, Gloria Teng, Warwick Tucker, and Dillon George. MRS 2.0, a C++ class library for statistical set processing and computer-aided proofs in statistics. <https://github.com/raazesh-sainudiin/mrs2>, 2008–2018.
- [3] Matei Zaharia, Reynold S. Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J. Franklin, Ali Ghodsi, Joseph Gonzalez, Scott Shenker, and Ion Stoica. Apache Spark: A unified engine for big data processing. *Commun. ACM*, 59(11):56–65, October 2016.