Privacy-Preserving Methods for Repeated Measures Designs

Kevin Liou
Netflix
Los Gatos, California, USA
kevinycliou@gmail.com

Wenjing Zheng
Netflix
Los Gatos, California, USA
wzheng@netflix.com

Sathya Anand
Netflix
Los Gatos, California, USA
sathya@netflix.com

ABSTRACT

Evolving privacy practices have led to increasing restrictions around the collection and storage of user level data. In turn, this has resulted in analytical challenges, such as properly estimating experimental statistics, especially in the case of long-running tests with repeated measurements. We propose a method for analyzing A/B tests which avoids aggregating and storing data at the unit-level. The approach utilizes a unit-level hashing mechanism which generates and stores the first and second moments of random subsets of the original population, thus allowing estimation of statistics, such as the variance of the average treatment effect (ATE), by bootstrap. Across a sample of past A/B tests at Netflix, we provide empirical results that demonstrate the effectiveness of the approach, and show how techniques to improve the sensitivity of experiments, such as regression adjustment, are still feasible under this new design.

CCS CONCEPTS

• **Mathematics of computing** → *Probability and statistics*; • **General and reference** → **Experimentation**.

KEYWORDS

privacy, A/B testing, controlled experiments

ACM Reference Format:
Kevin Liou, Wenjing Zheng, and Sathya Anand. 2022. Privacy-Preserving Methods for Repeated Measures Designs. In *Companion Proceedings of the Web Conference 2022 (WWW '22 Companion), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3487553.3524216

1 INTRODUCTION

Measures to protect user data have been increasingly prevalent across the industry. These enacted privacy practices range from more stringent guardrails regarding the collection of data samples to limiting the duration for which data can be stored.

For instance, recently Apple announced at WWDC that iPhone users had the choice to opt out of background data collection. For instances where data was collected, limitations regarding the duration to which data could be stored by third-party companies were also put in place [9, 20]. In a more familiar international example, in 2016 GDPR enacted a set of regulations that made huge effects on how data was treated across various tech companies [1, 8, 10, 17].

In turn, the emphasis in protecting user data has resulted in various analytical challenges that were once simple to execute. One example is in A/B testing. Long considered the gold standard of causal inference, A/B testing is widely used by companies as their main data-driven decision-making tool [11, 12]. However, the increased data privacy scrutiny has resulted in several challenges in experimentation in recent years. For example, there have been more guardrails regarding the duration to which data can be linked to their individual-level identifiers. Such data retention policies can limit both the conclusiveness of experiments as well as the ability to calculate important metrics [7]. This paper focuses on one growing relevant challenge: the inability to calculate key test statistics, such as the variance of the average treatment effect (ATE), for A/B tests with repeated measurements, such as longitudinal experiments.

1.1 The impact of privacy challenges on A/B testing

One example reveals how data retention limitations can affect the ability to calculate key experimental statistics. To begin, recall that for experiments with repeated measurements, to obtain the outcome metric for each experimental unit, one first aggregates all observations of a unit over the duration of the experiment. Estimating the ATE and variance of the ATE are then straightforward, once this outcome metric is obtained, either through bootstrapping or the delta method [3, 6].

However, there is one important nuance behind the execution of the approach above: each experimental unit must be linked and stored with its observations throughout the duration of the A/B test. While this has traditionally been straightforward, the emphasis on protecting user data over recent years will lead to challenges. Specifically, privacy restrictions may limit the storage of user data over an extended time period - in particular, more companies are implementing policies regarding how long data with individual-level identifiers can be stored. Since the same unit may be associated with multiple observations over time, this makes it infeasible to perform standard experimental analysis, one of which is to calculate the variance of the ATE, our primary focus below.

Figure 1 makes this point. On the left is an example experiment, where daily user metrics are observed over the duration of a study. If this were a two week experiment, for instance, then a maximum of 14 observations from each user are made. Here, M_{11}, M_{12}, and M_{13} correspond to metric observations 1, 2, and 3, respectively, for user 1. As denoted in column 3, the outcome metric, M_1, is the sum of all observations for each user 1, and M_1 is then used to

WWW '22 Companion, April 25–29, 2022, Virtual Event, Lyon, France.

ACM ISBN 978-1-4503-9130-6/22/04…$15.00
https://doi.org/10.1145/3487553.3524216

calculate final experimental statistics. Specifically, this outcome metric would be necessary if one were to calculate the variance of the ATE of this group of users. However, consider a scenario for which storing a unit associated with its observations over multiple days is prohibited, as mentioned previously. While we can still store all numerical observations for each user such as M_{11}, M_{12}, M_{21}, ..., the identity behind the metric is no longer known. This presents the scenario on the right in Figure 1. All metric observations are still available, but note how the correct outcome metric, which is the sum of a user and all of its observations over the duration of the study, is unknown. Therefore, we cannot obtain correct user-level sums, as required for the standard variance estimator.

The inability to estimate the variance of the ATE is one of many quantities that are intractable under this privacy restriction scenario, but it will be the main statistic that we focus on in our examples below. To solve this issue, we propose two approaches to enable one to estimate such statistics while adhering to privacy guidelines. The first estimator simulates bootstrapping by generating samples of the population probabilistically, while the second aggregates units into clusters, and treats each cluster as a single experimental unit. The key to both approaches, and the primary contribution of this paper, is a unit hashing mechanism which ensures that the sufficient statistics for estimating metrics of interest, such as the variance of the ATE, are stored without aggregating or storing unit-level information. In particular, the first and second moments of each unit metric and the number of treated units of random samples of the population are stored at each time step (e.g., at the end of each day), so that one will never need to store and link observations at the individual unit level.

Moreover, the second estimator—aggregating the moments of unit metrics into clusters via the hashing mechanism—can be used in conjunction with other popular experimental methods such as regression adjustment [4, 14, 18], for variance reduction purposes. We examine the details of this hashing procedure in the following section, and discuss pros and cons for each approach.

The decision on which estimator to use depends primarily on three things: 1) the amount of data in the experiment, 2) the storage capacity, and 3) whether A/B testing extensions such as variance reduction are necessary. We discuss these tradeoffs and alternatives in section 3. Using a sample of A/B tests at Netflix, we evaluate the performance of this privacy-preserving estimator relative to a situation where there are no restrictions on data storage. We find that the approach estimates the true variance of the ATE of an A/B test with high accuracy (bounded by 2% error) while achieving the nominal coverage probability. Moreover, we find that if the second estimator is used, then variance reduction approaches such as regression adjustment and variance-weighted estimators [15] may enable one to achieve nearly the same power compared to situations without privacy restrictions.

2 PRIVACY-PRESERVING EXPERIMENTATION

2.1 Estimator 1: Implementing the Bootstrap via Hashing

The key behind the first privacy-preserving estimator is the probabilistic generation of multiple subsets of the experimental population in order to effectively simulate a bootstrap. To do this, we create N subsets for both test and control, for which inclusion in each subset is based on whether a unit passes a deterministic hash function for that subset (e.g., each user id is hashed to a binary value). The hashed result, which is based on a unit and a subset id, is constant through the duration of the experiment.

Algorithm 1: Estimating the Variance of the ATE in A/B tests using Estimator 1

Result: Variance of ATE of A/B test

1. Initialize N test subset sums, $S_{t_1}, S_{t_2}, ..., S_{t_N}$, to 0.
2. Assign to each test unit i and test subset k pair a Poisson(1) draw, if using the Poisson bootstrap, or Bernoulli(0.5), if using half sampling. Denote this draw as w_{ik}.
3. When each test unit i logs a metric j, m_{ij}, during the experiment, update each subset sum: $S_{t_k} = S_{t_k} + w_{ik} * m_{ij}$
4. After the experiment is complete, for each subset k calculate its mean μ_{t_k}: $\mu_{t_k} = \frac{S_{t_k}}{\sum_i w_{ik}}$
5. Repeat Steps 1-4 for the control group.
6. Using subset means $\mu_{t_1}, \mu_{t_2}, ...$ in test and $\mu_{c_1}, \mu_{c_2}, ...$ in control, one can calculate the variance of the ATE.

For instance, one way to accomplish this would be to make a Bernoulli(0.5) draw for each unit-subset pair, indicating that a unit will belong in a given subset with 50% chance. Note that selecting, on expectation, half of the units in an experiment is well studied in prior literature, and is known as *half-sampling* or the *double or nothing bootstrap* [5, 13]. This hashing function will map each unit ID and subset ID pairing into a binary label, indicating whether the unit ID belongs in the subset or not.

Throughout the experiment, for each subset one will store two rolling statistics. The first is the the sum of all observations for units in the subset. To do this, given subset k, metric observation j, and unit i, we have subset sum S_k, metric observation j for unit i, denoted m_{ij}, and weight w_{ik} denoting the binary hash for $\mathcal{H}$(unit id, subset id). Next, each subset sum is calculated as $S_k = S_k + w_{ik} * m_{ij}$. Note that if a weight is 0, that means the unit observation does not belong in that subset, so the subset sum will not be updated. The second metric we store is the total number of distinct units in the subset. Based on these two values, one can calculate the mean μ_k of each subset k as $\mu_k = \sum_{i=1}^{M} w_{ik}(\sum_{j \in i} m_{ij} / \sum_{i=1}^{M} w_{ik})$, given units $i = 1, ..., M$, j metrics m_{ij} for each unit i, and weights w_{ik} for each unit-subset pair obtained through a hash function $\mathcal{H}$. In the case of using binary weights, the weights simply denote whether a unit belongs in a subset, and $\sum_{i=1}^{M} w_{ik}$ will output the number of units in subset k. After obtaining these subset means, one can calculate N differences in means with the N means obtained for both test and control subsets. Estimating both the average treatment effect and its variance are then straightforward, as we have simulated sampling with replacement - the essence behind bootstrapping.

Extending to Poisson weights. While the estimator above provides a method to simulate a bootstrap, it does have one important shortcoming. A bootstrap samples all units with replacement from a population, but the approach above simply selects a percentage of units in an experiment at random without replacement to belong in each subset, based on the pre-selected binomial probability given

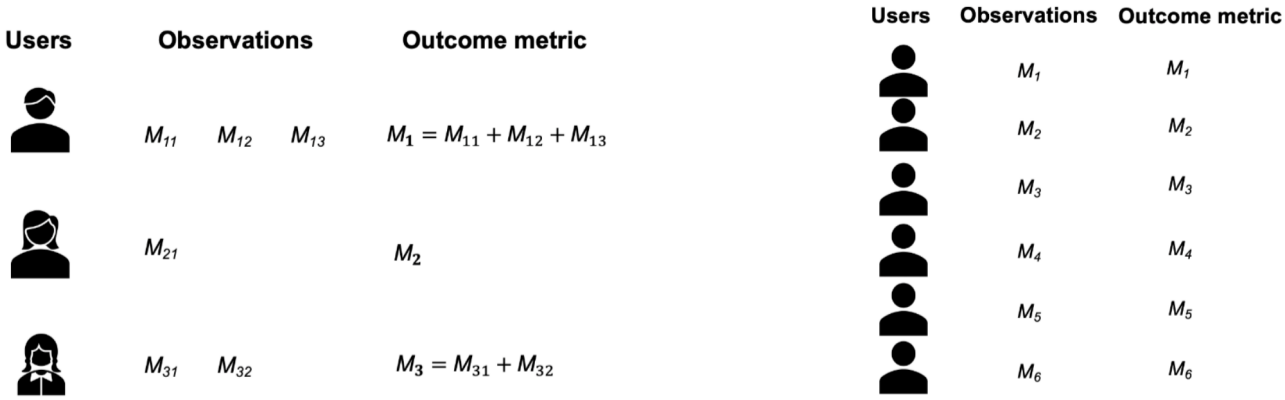

Figure 1: left: example A/B test, right: A/B test where user observations are not linked to individual-level identifiers

in (1). When using half-sampling, for instance, then 50% of the population is selected.

One option that solves this issue, and importantly also works nicely with streaming data, is a Poisson bootstrap [2, 16]. Here, instead of a hash function that outputs a binary label indicating whether a unit belongs in a subset, the hash value is now drawn from a Poisson(1) distribution, i.e., each distinct value of $\mathcal{H}$(unit ID, subset ID) is assigned a draw from a $Poisson(1)$ distribution. If we assume M users in each subset, this approximates a $Binomial(M, 1/M)$ well, since $\lim_{M\to\infty} Binomial(M, \frac{1}{M}) = Poisson(1)$. The reason why we opt not to directly draw from the binomial distribution $Binomial(M, 1/M)$ instead of $Poisson(1)$ is because for most experiments we don't know the sample size M in advance, and instead are dealing with streaming data. See Algorithm 1 below for a step-by-step outline of estimator 1.

2.2 Estimator 2: Unit clusters

While estimator 1 enables one to estimate experimental statistics correctly, such as the variance of the ATE, while adhering to more stringent privacy principles, it does have several major limitations. Specifically, 1) there is a large storage requirement, depending on the number of bootstrap iterations, and 2) the estimator may be suitable only for estimating experiment results, and less flexible to be able to be used in conjunction with other experimental techniques, such as variance reduction [21].

One extension of estimator 1 that improves on these issues is to employ the same hashing trick as estimator 1, but change the hashing mechanism such that units of an A/B test are hashed into distinct clusters. The main difference here is that *each unit belongs in only one cluster*; in effect, all experimental units are partitioned into disjoint clusters. Instead of drawing from a Bernoulli or Poisson(1) value as in estimator 1, across all clusters C_i, $i = 1, ..., N$, only one of $\mathcal{H}(C_i, X_j)$ will output a value of 1 (e.g., unit ID belongs in only one cluster.) And again, whenever a unit logs a metric, the product of the binary hash and the metric will indicate whether an observation is logged in the cluster, and this product will be added to the cluster sum. For example, if 1000 clusters are desired and there are a million units in the test group of the experiment, we would expect to have approximately 1000 units in each cluster. The key of this approach, like estimator 1, is that the hashing mechanism enables us to *not have to ever store any pair of a user and its metric together*, but rather only the cluster-level sums of subsets of the population. Once again, the variance of the ATE can be obtained readily.

Estimator 2 enables one to conduct analysis with more flexibility and apply methods such as regression adjustment, since one can use the identical subset-level hashing functions for each cluster-unit pair pre-treatment. In addition, in the next subsection we show how other tricks, like storing cluster-level variances, are also feasible with this approach. Storing both the mean and variance of clusters allows for the potential for more analysis of the experimental sample, such as estimating the variance of the underlying population.

However, this estimator also has its disadvantages. The first is that information on the unit-level is diminished when we aggregate unit data together, and therefore estimations of variance may be less accurate. Additionally, this approach requires experiments with large sample size. Otherwise, if the number of units in each group is small, unit-level privacy may still be compromised.

2.3 Storing moments of metrics

Note that in the two estimators mentioned above, our main objective was to be able to calculate the variance of the ATE. Doing this required storing the sums of random subsets of units. However, it is important to note that we can store *any function of an observation* without running into privacy issues. One option is to store a higher order of moments of each individual observation. This allows for the possibility of obtaining any statistic that is a function of moments of metrics. One example metric of interest is storing the cluster-level variance. To see how this can be useful, a challenge we often face is the possibility of highly heteroscedastic units. It is common that clusters within an A/B test have high metric variability. As a result, A/B test units become a mixture of highly heterogeneous units.

To solve this issue, ideally we can store the *variance* of each cluster, in addition to the mean. Obtaining this turns out to be

Table 1: Comparing results across estimators

Estimator	% Variance error	% Coverage ($\alpha = 0.05$)
`Estimator 1 (Double or Nothing)`	1.82%	94.1%
`Estimator 1 (Poisson)`	1.01%	95.2%
`Estimator 2`	1.46%	94.7%
`Estimator 2 + inverse-variance weighting`	1.83%	92.3%

Algorithm 2: Reducing variance using inverse-variance weighting

Result: ATE and Variance of ATE of A/B Test

1 Initialize cluster-level first moment sums, $C^1_{t_1}, C^1_{t_2}, ..., C^1_{t_N}$, to 0.

2 Additionally, also initialize cluster-level second moment sums, $C^2_{t_1}, C^2_{t_2}, ..., C^2_{t_N}$, to 0.

3 Assign to each test unit X_i and cluster C_k pair a binary value, $\mathcal{H}(C_i, X_i) \rightarrow [0, 1]$, denoted w_{ik}.

4 When each test unit i logs a metric j, m_{ij}, during the experiment, update each subset sum:
1) $C^1_{t_k} = C^1_{t_k} + w_{ik} * m_{ij}$ 2) $C^2_{t_k} = C^2_{t_k} + w_{ik} * m^2_{ij}$

5 After the experiment is complete, for each cluster k calculate its:
1) mean: $\mu_{t_k} = \frac{C^1_{t_k}}{\sum_i w_{ik}}$ 2) variance: $\sigma^2_{t_k} = \frac{C^2_{t_k} - (C^1_{t_k})^2}{\sum_i w_{ik}}$

6 One can now calculate the variance of the ATE in two ways: the first is the standard unweighted estimator, and the second applies inverse-variance weights to each cluster:

$$\Delta = \frac{\sum_i \sigma^2_{t_k} \mu_{t_k}}{\sum_i \sigma^2_{t_k}} - \frac{\sum_i \sigma^2_{c_k} \mu_{c_k}}{\sum_i \sigma^2_{c_k}}$$

straightforward. One adjustment to the approach detailed in section 2.2 is to store the second moment of each observation as well, so that at the conclusion of the experiment we have both the sum of the first and second moments of all observations, still without linking each observation to its unit ID. Obtaining the variance of clusters now enables us to perform more techniques on top of the aforementioned privacy-preserving estimator. For instance, one practical technique utilizing the variance of clusters is to increase the overall power of an A/B test, using an inverse-variance weighting approach [15, 19]. This is outlined in Algorithm 2.

3 EMPIRICAL RESULTS

3.1 Case Studies

To measure the accuracy of both estimators from section 2, we used a set of A/B tests at Netflix. Each test was analyzed in two ways: the first is the classical case where users observations are recorded with no data storage restrictions. The ATE and its variance can be estimated straightforwardly. In the second scenario, we assume that a user cannot be linked to its metric after 24 hours. As a result, the estimators in section 2 are necessary in order to accurately estimate statistics of interest, such as the variance of the ATE.

To evaluate the performance of the privacy-preserving estimators, at the end of each time step, the sum of all observations for users within a subset are aggregated, such that no user-level data is stored. We then calculated two statistics: the variance error, defined as the percentage difference between the variance estimated assuming no privacy restrictions, versus the variance estimated in the estimators above. We also simulated A/A tests and calculated the coverage rate. Four estimators were compared: the double-or-nothing (half sampling) method for estimator 1, Poisson weights for estimator 1, estimator 2, and estimator 2 + cluster-level variances.

The results are shown in Table 1. In general, all three estimators have similar performances, where the % error in variance is less than 2%. Using estimator 1 and Poisson(1) weights provides the optimal performance, in terms of achieving the nominal coverage probability and estimating the true variance accurately. Moreover, using estimator 2 along with storing the second moments of individual observations allows us to store and estimate cluster-level variances, as detailed in Algorithm 3. This provides us with the option of using an inverse-variance weighted (IVW) estimator. In addition to being able to accurately estimate variance, we find that using IVW enables us to achieve, on average, a **5.6%** decrease in the variance of the ATE estimator, adding another benefit on top of our privacy-preserving estimator.

4 CONCLUSION

Limiting the storage of individual-level identifiers with their observations is an increasingly common privacy principle adopted in the industry. For experiments with repeated measurements, this creates a situation where calculating key experimental statistics, like the variance of the ATE, becomes intractable. The estimator proposed in this work enables one to bypass this issue by utilizing a unit-level hashing mechanism that stores moments of random subsets of the original test population. Two variants of the estimator are introduced - each with different degrees of accuracy, privacy, and storage requirements. We also show that methods such as variance reduction via either regression adjustment or cluster-level variances are still possible under this estimator. Finally, empirical results across a set of past A/B tests show that the proposed estimators are able to estimate the variance of the ATE with high accuracy, while achieving nominal coverage.

We conclude by noting that while this paper focuses on estimating key metrics in A/B tests, such as the variance of the ATE or cluster-level variances, experimenters can also leverage this approach to obtain a larger set of statistics for their purposes, in non-experimentation settings as well. The only requirement is that the statistic of interest is a function of moments of the metric observations of interest.

5 ACKNOWLEDGMENTS

The authors are grateful to Ian Yohai from Netflix, and Felipe Coirolo, Riccardo Tortul, Ian Yohai, Daniel Haimovich, Kenneth Hung, Nihar Shah, Dominic Coey, and Peter Straka from Facebook, for contributions to previous editions of the paper and useful comments and discussions.

REFERENCES

[1] Jan Philipp Albrecht. 2016. How the GDPR will change the world. *Eur. Data Prot. L. Rev.* 2 (2016), 287.
[2] Nicholas Chamandy, Omkar Muralidharan, Amir Najmi, and Siddartha Naidu. 2012. *Estimating Uncertainty for Massive Data Streams.* Technical Report. Google.
[3] Alex Deng, Ulf Knoblich, and Jiannan Lu. 2018. Applying the Delta method in metric analytics: A practical guide with novel ideas. *arXiv preprint arXiv:1803.06336* (2018).
[4] Alex Deng, Ya Xu, Ron Kohavi, and Toby Walker. 2013. Improving the sensitivity of online controlled experiments by utilizing pre-experiment data. In *Proceedings of the sixth ACM international conference on Web search and data mining*. 123–132.
[5] Bradley Efron. 1981. Nonparametric estimates of standard error: the jackknife, the bootstrap and other methods. *Biometrika* 68, 3 (1981), 589–599.
[6] Bradley Efron and Robert J Tibshirani. 1994. *An introduction to the bootstrap.* CRC press.
[7] Aleksander Fabijan, Pavel Dmitriev, Colin McFarland, Lukas Vermeer, Helena Holmström Olsson, and Jan Bosch. 2018. Experimentation growth: Evolving trustworthy A/B testing capabilities in online software companies. *Journal of Software: Evolution and Process* 30, 12 (2018), e2113.
[8] Michelle Goddard. 2017. The EU General Data Protection Regulation (GDPR): European regulation that has a global impact. *International Journal of Market Research* 59, 6 (2017), 703–705.
[9] Allegra Hobbs. 2021. Facebook Battles Apple Over User Privacy. (2021).
[10] Kimberly A Houser and W Gregory Voss. 2018. GDPR: The end of Google and Facebook or a new paradigm in data privacy. *Rich. JL & Tech.* 25 (2018), 1.
[11] Ron Kohavi, Alex Deng, Brian Frasca, Toby Walker, Ya Xu, and Nils Pohlmann. 2013. Online controlled experiments at large scale. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1168–1176.
[12] Ron Kohavi, Diane Tang, and Ya Xu. 2020. *Trustworthy online controlled experiments: A practical guide to a/b testing.* Cambridge University Press.
[13] David Letson and BD McCullough. 1998. Better confidence intervals: The double bootstrap with no pivot. *American journal of agricultural economics* 80, 3 (1998), 552–559.
[14] Winston Lin. 2013. Agnostic notes on regression adjustments to experimental data: Reexamining Freedman's critique. *The Annals of Applied Statistics* 7, 1 (2013), 295–318.
[15] Kevin Liou and Sean J Taylor. 2020. Variance-Weighted Estimators to Improve Sensitivity in Online Experiments. In *Proceedings of the 21st ACM Conference on Economics and Computation*. 837–850.
[16] Art B Owen, Dean Eckles, et al. 2012. Bootstrapping data arrays of arbitrary order. *The Annals of Applied Statistics* 6, 3 (2012), 895–927.
[17] Giovanni Maria Riva, Alexandr Vasenev, and Nicola Zannone. 2020. SoK: Engineering privacy-aware high-tech systems. In *Proceedings of the 15th International Conference on Availability, Reliability and Security*. 1–10.
[18] Donald B Rubin. 1973. The use of matched sampling and regression adjustment to remove bias in observational studies. *Biometrics* (1973), 185–203.
[19] Julio Sánchez-Meca and Fulgencio Marin-Martinez. 1998. Weighting by inverse variance or by sample size in meta-analysis: A simulation study. *Educational and Psychological Measurement* 58, 2 (1998), 211–220.
[20] David Stites and Katie Skinner. 2014. User privacy on iOS and OS X. In *The Apple Worldwide Developers Conference*.
[21] Huizhi Xie and Juliette Aurisset. 2016. Improving the sensitivity of online controlled experiments: Case studies at netflix. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 645–654.