ESTIMATING UNCERTAINTY FOR MASSIVE DATA STREAMS

By Nicholas Chamandy, Omkar Muralidharan,
Amir Najmi, Siddartha Naidu [*]

Google

We address the problem of estimating the variability of an estimator computed from a massive data stream. While nearly-linear statistics can be computed exactly or approximately from "Google-scale" data, second-order analysis is a challenge. Unfortunately, massive sample sizes do not obviate the need for uncertainty calculations: modern data often have heavy tails, large coefficients of variation, tiny effect sizes, and generally exhibit bad behaviour. We describe in detail this New Frontier in statistics, outline the computing infrastructure required, and motivate the need for modification of existing methods. We introduce two procedures for basic uncertainty estimation, one derived from the bootstrap and the other from a form of subsampling. Their costs and theoretical properties are briefly discussed, and their use is demonstrated using Google data.

1. Introduction.

With the advent of modern computing and the rise of user-generated web content, data structures are becoming increasingly unwieldy. In this paper we investigate the important problem of estimating the variance of a statistic in the context of "Google-scale" data. Three important properties of these data sets necessitate specialized methodology:

1. *Massive scale.* The sample size N in a statistical problem can be very large, at least on the order of 10^9 and often larger.
   E.g. N is the number of visitors to google.com over the course of a week.
2. *Streaming form.* The statistician interacts with the data as a stream. This is true either literally, because the data are analyzed as they are collected, or is effectively true since the data are too large to be fit in memory.
   E.g. Each time a query is issued on google.com, a server emits data to logs.
3. *Sharded units.* Data from one statistical unit are scattered temporally or across multiple data sources or machines ("shards"); it is therefore impractical to retain an entire observation in memory at any given time.
   E.g. A user's queries do not appear consecutively, nor are they guaranteed to be processed by the same server or even the same data center.

There are many practical implications of these properties. Item 1 suggests that we have finally arrived in *Asymptopia*, and yet at the same time precludes computationally intensive methods and even positing structures of size, say, $N \times N$. Computational tools such

[*] N.C. and A.N. are Statisticians, O.M. is Research Scientist, S.N. is Software Engineer; Google, 1600 Amphitheatre Pkwy, Building 40, Mountain View, CA 94043.

as the MapReduce framework (Dean and Ghemawat, 2004), which we discuss briefly in Section 1.2, are necessary. Moreover, web data on even this scale often contain sufficient variability that estimating it is important. An example will be given in Section 5.

Property 2 implies that 'single-pass' methods are preferable to multi-pass or iterative algorithms. For instance, fitting certain classes of linear models to streaming data demands a special algorithm wherein parameter estimates are updated for each new data point. In this paper we assume for the most part that appropriate single-pass estimation procedures exist.

The consequences of Property 3 are most material to this paper, and often counter-intuitive: summary statistics that we normally take for granted, such as sums of squared deviations across all units, are sometimes unavailable. The latter observation, discussed further in Section 1.3, is what makes uncertainty estimation challenging in the context of massive data streams.

1.1. *Mathematical framing.*

In this paper we consider the problem of estimating a parameter θ of the distribution F of an i.i.d. stream of random objects

$$\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N, \ldots \tag{1}$$

and most importantly estimating the uncertainty in our estimator. An important constraint is that we cannot necessarily observe any $\mathbf{X}_i$ as a random variable in memory. This is a fundamental property of massive data streams: the exchangeable unit is often different from the unit used to record the data. We will see an example of this in the next section, where the exchangeable unit is a Google user, while the unit of the data stream is a Google *query* event. In view of this, the observed data stream is more formally

$$\mathbf{X}_{i_1,j_1}, \mathbf{X}_{i_2,j_2}, \ldots, \mathbf{X}_{i_k,j_k}, \ldots \tag{2}$$

where i indexes the exchangeable unit and j the record within the exchangeable unit. The notation in (2) highlights that the ordering of our data stream is completely arbitrary. For the most part, to simplify notation, we stick to representation (1) with the understanding that $\mathbf{X}_i$ may be sharded across different records or machines.

In general, $\mathbf{X}_i$ may be a non-vector, instead having a more complex hierarchical structure with a random dimensionality. For example, it may consist of the set of queries issued by a user on a given day. We denote the space of possible values by $\mathcal{X}$. Without any real loss of generality, we can assume if necessary that each $\mathbf{X}_i$ has a p-dimensional sufficient statistic $\mathbf{Y}_i$. Therefore, for simplicity we assume $\mathcal{X} = \mathbb{R}^p$.

We assume throughout that F has the requisite number of finite absolute moments, and denote its mean by μ and variance matrix by Σ (or σ^2 when $p = 1$). For the most part, we focus our attention on functions of means because their sample versions can be easily computed from one pass over a massive data stream using MapReduce. In such cases, the parameter of interest can be expressed as $\theta = g(\mu)$, where $g : \mathbb{R}^p \to \mathbb{R}$ is a smooth function with gradient $\dot{\mathbf{g}}$ and hessian $\ddot{\mathbf{g}}$. Moreover, letting $\mathbf{X} = (\mathbf{X}_1 \cdots \mathbf{X}_N)$ and

$\bar{\mathbf{X}} = \bar{\mathbf{X}}_N$ denote the sample mean, we typically estimate θ by

$$\hat{\theta}_N(\mathbf{X}) = g(\bar{\mathbf{X}}) = g\left(\sum_{i=1}^{N} \mathbf{X}_i/N\right). \tag{3}$$

In order to keep technical conditions to a minimum, we simply assume that g has continuous derivatives of all orders, each of which – including g itself – can be bounded componentwise by a polynomial in its argument. (Less restrictive assumptions are possible on a results-by-result basis, but they do not add any real insight.)

We are interested in either (a) estimating the variance of $\hat{\theta}_N$ or (b) estimating the distribution of a root

$$R_N = \tau_N(\hat{\theta}_N - \theta), \tag{4}$$

with τ_N an appropriate scaling constant. Typically $\tau_N = N^{1/2}$, but it could be any sample size dependent constant which gives R_N a nondegenerate limiting distribution. We denote by G_N the distribution of R_N. Let $\mathsf{Var}(\hat{\theta}_N) = \xi_N^2$, and write $\mathbb{1} \otimes \bar{\mathbf{X}}$ to denote the $p \times N$ matrix $(\bar{\mathbf{X}}, \ldots, \bar{\mathbf{X}})$. The classical delta method estimator of ξ_N^2 is

$$S_\Delta^2 = \dot{\mathbf{g}}(\bar{\mathbf{X}})'(\mathbf{X} - \mathbb{1} \otimes \bar{\mathbf{X}})(\mathbf{X} - \mathbb{1} \otimes \bar{\mathbf{X}})'\dot{\mathbf{g}}(\bar{\mathbf{X}})/N^2. \tag{5}$$

In the simple case that $p = 1$ and $g(\mu) = \mu$, which we shall refer to from time to time as an illustrative example, (5) reduces to

$$S_\Delta^2 = \frac{1}{N^2}\sum_{i=1}^{N}(X_i - \bar{X})^2, \tag{6}$$

the scaled (slightly biased) sample variance. As the example in the next section demonstrates, Limitation 3 above prevents us from explicitly computing the sample covariance in (5). While it is tempting to calculate S_Δ^2 naïvely by assuming that each data record is independent, doing so typically ignores positive correlation and leads to underestimation of ξ_N^2. S_Δ^2 nonetheless serves as a useful gold standard against which to evaluate other variance estimators, since it works remarkably well when available in massive data settings. Our main contribution will be to describe two feasible alternatives to S_Δ^2 and examine their properties.

Though the emphasis above, and indeed throughout much of the paper, is on parameters which are smooth functions of μ, our results in Section 4 generalize the validity of the methods to more complicated functionals of G_N. These include quantiles, and other parameters for which bootstrap-type inference is valid.

1.2. *A cartoon Google data stream.*

Figure 1 depicts a stylized path that Google data may take from the time of their creation to the time they arrive on the statistician's computer. The example is designed to illustrate why traditional methods, which assume that all data from a given statistical unit are available in memory at the same time, fail for massive data. It is not necessarily the case that such data are analyzed continuously upon arrival, however, it is useful to think in such terms because our most powerful computational tools, including MapReduce, effectively reproduce such a scenario. They do this to achieve the degree of parallelism necessary for doing computation at scale.

We assume that the final summary data to be output is a list of tuples $(k, v) = (k, v(k))$, where $k \in \mathcal{K}$ denotes a categorical 'key', and $v \in \mathcal{V}$ a random vector somehow obtained by aggregation from the raw data. This is typical in the analysis of web data. The cardinality of $\mathcal{K}$ may be large, though it is generally orders of magnitude smaller than the size of the raw data. As a concrete running example, consider the following application involving advertisements displayed on google.com search results pages. We wish to compute the average click-through rate (CTR) for the largest 10,000 advertisers and 100 countries, defined as the total number of ad clicks divided by the number of ads displayed (called "impressions"). In this case, k is the bivariate factor (advertiser ID)×(country), and v a pair consisting of an ad impression count and an ad click count (summed over many queries).

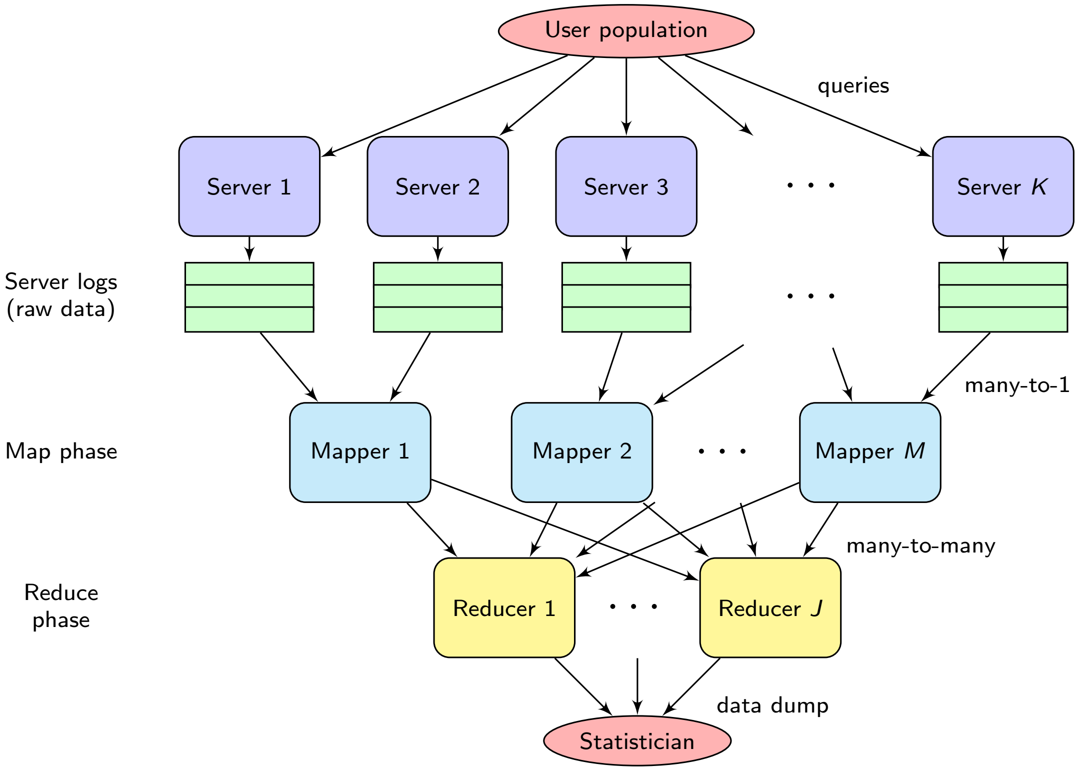


Fig 1: A stylized example of a Google data stream. Every user-generated query is processed by one of K servers, with different queries from the same user possibly hitting different servers. The servers write raw data to logs, one record per query. These records are first processed by M mapper machines, which are shuffled along the key of interest k and mapped in order to J reducer machines. The data are further processed by the reducers into a single summary tuple for each unique value of k. These data are finally analyzed by the statistician. Typically, $K \gg M \gg J$.

Briefly, MapReduce is a parallel computing framework which can produce such tuples from petabytes (10^{15}B) of input data in just a few hours, by leveraging a large number of machines. In its simplest form, the MapReduce consists of a single *master* machine, which coordinates the process, and many *mapper* and *reducer* machines. The system is fully parallel in that there is no communication among mappers nor among reducers (though there is some limited communication between the two groups, and both communicate with the master).

There are correspondingly two phases of the MapReduce. During the *map phase*, each mapper processes a set of input records, which may themselves be complex non-vector objects, and produces intermediate tuples (k, u), $u \in \mathcal{U}$. The u tuples are different from the final v because they have been computed from only a subset of the data, and must be further combined. These intermediate data are then sorted by k and sent as input to the reducers, a process known as 'shuffling'. Assume, as in Figure 1, that there are M mappers and J reducers. Shuffling can be thought of simply as a map S from $\mathcal{K}$ into $\{1, \ldots, J\}$. In the *reduce phase*, the jth reducer further aggregates the vectors u to produce a final statistic $v = v(k)$ for every unique $k \in S^{-1}(j)$. The final data, which are of a manageable size, can then be downloaded from the MapReduce network.

In slightly more detail, the output of mapper m is the set of tuples $\{(k, u_m(k)),\ k \in \mathcal{K}_m\}$ for some subset $\mathcal{K}_m \subset \mathcal{K}$ of keys which happen to appear therein. For a fixed key k, $v(k)$ must then be computable from the sequence of arbitrarily ordered vectors $u_m(k)$ via a binary, associative operation. In other words, there exists a function $A : \mathcal{U}^2 \to \mathcal{V}$ such that $A(s, r) = A(r, s)$ and

$$v(k) = A(u_M(k), \ldots, A(u_3(k), A(u_2(k), u_1(k))) \cdots) \tag{7}$$

(in the case that $k \in \mathcal{K}_m$ for every m).

It is the job of the $S(k)$th reducer to perform the calculation (7), in a sequential manner as it scans over its input data. In the CTR example given above, u consists of partial sums of (ad impressions, ad clicks), and the aggregator A is simple addition. More generally, A may be allowed to depend on a small buffer of summary statistics of the local collection of previously seen $u(k)$. This slight complication extends the methodology from sum-like estimands to more complicated statistics such as approximate sample quantiles.

1.3. *Practical example.* Rate parameters are ubiquitous in the statistical analysis of internet data. In addition to the CTR statistic introduced in the last section, other examples include the average revenue per web pageview for display advertising, and the "conversion rate" among users who click through to an e-commerce site. Such scenarios can be modeled within the framework of Section 1.1 by assuming that

$$\mathbf{X}_i = (X_i, Y_i)'; \quad \mathbb{E}X_i = \mu; \quad \mathbb{E}[Y_i \mid X_i] = \theta X_i; \quad g(x, y) = y/x; \quad x > 1. \tag{8}$$

In many Google-scale applications, distributions of statistics can be reasonably well-approximated by a lognormal distribution. Since a ratio of lognormals is again lognormal and has moments of all orders, this is a convenient distribution to posit in the current example.

The delta method variance estimator is

$$S_\Delta^2 = \left(\frac{\bar{Y}}{\bar{X}}\right)^2 \left\{ \frac{\sum_i X_i^2}{N\bar{X}^2} + \frac{\sum_i Y_i^2}{N\bar{Y}^2} - 2\frac{\sum_i X_i Y_i}{N\bar{X}\bar{Y}} \right\}. \tag{9}$$

To see why (9) poses computational problems in a massive data setting, let us return to our running CTR example. X_i denotes the number of ad impressions displayed to User i over some time period, and Y_i denotes the number of times the user clicked on an ad. The exhangeable unit is the Google user, the record unit is a Google query, and the primary output key is an advertiser-country pair. This example illustrates yet a third unit of interest: the finest division of the data which belongs to a unique element of $\mathcal{K}$, which we call the "key unit". In a sense, this becomes the new sharding unit once the MapReduce has begun. In our example, the key unit is an ad impression.

Under the MapReduce framework, and analogously to (2), we can decompose a user's data based on which reducer machine each impression is mapped to. Therefore, $X_i = \sum_{j=1}^{J} X_{ij}$ and $Y_i = \sum_{j=1}^{J} Y_{ij}$ with j indexing the reducer. In this example, there may exist $j \neq j'$ such that both $X_{ij} > 0$ and $X_{ij'} > 0$ for any user seeing ad impressions from multiple advertisers. Thus, while it is a simple matter to compute both $\sum_i X_i = \sum_{i,j} X_{ij}$ and $\sum_{i,j} X_{ij}^2$, we have no hope of computing $\sum_{i,j,j'} X_{ij} X_{ij'}$, nor therefore $\sum_i X_i^2 = \sum_i \left(\sum_j X_{ij} \right)^2$. Similar arguments of course apply to $\sum_i Y_i^2$ and $\sum_i X_i Y_i$. Moreover, blindly using $\sum_{i,j} X_{ij}^2$, $\sum_{i,j} Y_{ij}^2$ and $\sum_{i,j} X_{ij} Y_{ij}$ in their place will systematically underestimate the variability, to the extent that different queries and ad impressions from the same user are correlated.

Put simply, the delta method fails for massive data whenever sharding has been done in a manner inconsistent with the statistical dependence structure of the data—when the exchangeable unit is coarser than the record unit. This phenomenon is illustrated with real data in Section 5.

1.4. *Costs and constraints.* There are numerous costs associated with inference on massive data streams. Loosely, they can be grouped as follows.

- *Iteration cost.* Increases with the number of required passes over the data, and decreases as greater parallelization is acheived.
- *Computation cost.* Related to the complexity of the intermediate calculations performed on each input record.
- *Input/output (I/O) cost.* These can be grouped into the cost associated with
  1. Reading a record into memory.
  2. Updating an estimator (e.g. incrementing a MapReduce sum).
  3. Writing data to file for post-processing.

In this paper we assume that single-pass, highly parallel algorithms are available, and that the computation required per input record is constant with respect to the uncertainty estimation method chosen (from among those proposed). Hence iteration and computation costs are less interesting than I/O cost.

The primary cost with which we shall concern ourselves is the storage and retrieval cost associated with I/O, which we shall later summarize with a parameter b. Suppose, as is often the case, that post-processing of our summary statistics must be performed, using R software say, on a single machine. For example, we may wish to examine fine categorical slices of the advertiser population in order to isolate where an applied treatment has the greatest impact. Typically, our data will have size proportional to $b|\mathcal{K}|$. Put differently, for a fixed amount of disk memory D, we can effectively only afford on the order of $D/|\mathcal{K}|$ 'degrees of freedom' for use in estimating second-order statistics.

1.5. *Looking ahead.* The rest of the paper is organized as follows. In Sections 2 and 3 we describe two competing but related approaches to solving the inference problem described above, and investigate their properties. The first of these is a streaming analogue of subsampling, and the second a streaming analogue of the bootstrap. The main theoretical results of this paper will be to show that (a) the two methods have about equivalent asymptotic performance, and (b) they are both, in a sense to be made more precise, very close to their non-streaming counterparts. We close by illustrating our results via a hybrid real/simulated data example in Section 5.

2. Streaming buckets.

When analyzing stream data we are both cursed and blessed by the typically large, and a priori unknown, sample size N. One simple way to use the sample size to our advantage is to subdivide the statistical units into b approximately equal sized groups, which we call 'buckets'. The estimator computed from the jth bucket is also based on a large number $n \approx N/b$ of observations, and hence its distribution is closely related to that of $\hat{\theta}_N$. In particular, the variability among these b copies of our statistic, which we term *bucket replicates*, can be used to approximate the sampling uncertainty of $\hat{\theta}_N$.

More precisely (and assuming that b divides N for simplicity), consider relabeling the random objects with double indices, using the mapping $\mathbf{X}_{(i-1)b+j} \mapsto \mathbf{X}_{[ij]}$, with $1 \le j \le b$ and $1 \le i \le n$. The jth bucket replicate can be defined as $\hat{\theta}^\bullet_j = g(\bar{\mathbf{X}}_j)$ where $\bar{\mathbf{X}}_j = \sum_{i=1}^n \mathbf{X}_{[ij]}/n$. When N and n are both large, the distribution of $\bar{\mathbf{X}}_j$ is obviously close to that of $\bar{\mathbf{X}}$; it is usually a small leap to conclude that $g(\cdot)$ preserves this relationship. As we shall see, the precision of these estimators improves as the array implied by our 2-D indexing gets 'fatter', but the cost incurred also increases linearly.

2.1. *Variance estimator.* A natural estimator of ξ_N^2 which makes use of bucket replicates is given by

$$S^2_{\mathsf{buck}} = \frac{(\tau_n/\tau_N)^2}{b-1} \sum_{j=1}^{b} (\hat{\theta}^\bullet_j - \hat{\theta}_N)^2. \tag{10}$$

Technically (10) is closer in spirit to a mean-square error, and contains an additional squared bias term. One could eliminate that concern by replacing $\hat{\theta}_N$ by $\bar{\theta}^\bullet_b$, the mean estimate computed over b replicates. However, in typical situations the difference is negligible.

Note that with b fixed, $\xi_n^2 = \mathsf{Var}(\hat{\theta}^\bullet_1)$ and S^2_{buck} is consistent for ξ_N^2 provided

$$\tau_n^2 \xi_n^2 / (\tau_N^2 \xi_N^2) \longrightarrow 1 \tag{11}$$

as $N \to \infty$.

In most cases, $\tau_n = \sqrt{n}$, and assumption (11) is reasonable by virtue of the delta method approximation

$$\mathsf{Var}(g(\bar{\mathbf{X}})) \approx \dot{\mathbf{g}}(\mu)'\mathsf{Var}(\bar{\mathbf{X}})\dot{\mathbf{g}}(\mu) \tag{12}$$

$$= \dot{\mathbf{g}}(\mu)'\Sigma\dot{\mathbf{g}}(\mu)/N \tag{13}$$

$$\approx \frac{n}{N}\mathsf{Var}(g(\bar{\mathbf{X}}_1)). \tag{14}$$

The estimator S^2_{buck} is closely related to that given by Carlstein in the context of stationary timeseries data (Carlstein, 1986). It is also closely related to the delete-h jackknife (for $h = (1 - 1/b)N$) (Efron, 1982), the n out of N bootstrap (Bickel, Götze and van Zwet, 1997), and disjoint block subsampling (Politis, Romano and Wolf, 1999).

2.2. *Plug-in estimator.* A full-fledged application of the plug-in principle yields the bucket empirical distribution

$$G^{\bullet}_N(x) = \frac{1}{b}\sum_{j=1}^{b} \mathbb{1}\{\tau_n(\hat{\theta}^{\bullet}_j - \hat{\theta}_N) \leq x\}, \tag{15}$$

which can be used as an approximation to the sampling distribution function G_N of R_N. The distribution (15) can be used to construct approximate quantile-based confidence intervals for $\hat{\theta}_N$. It is shown to be a consistent estimator of G_N in Section 4.

2.3. *Streaming allocation.* Until now in this section, we have assumed that each bucket replicate is formed from exactly $n = N/b$ independent units. When the data are streaming, this can rarely be guaranteed; indeed, N is not even known in advance. Instead, each new unit is “allocated” to replicate j with probability $1/b$, analogously to a b-arm clinical trial with continuous patient recruitment (and equal-sized treatment arms). When units are sharded, as for web data, this must be done carefully. One approach is to introduce a hash function

$$f : \mathcal{S} = \{\text{user ID strings}\} \longrightarrow \{0, 1\}^{64} \tag{16}$$

mapping the space of unique user ID strings uniformly to the 64-bit integers. For User i with ID $s_i \in \mathcal{S}$, bucket allocation is then done using a pseudo random number generator seeded with $f(s_i)$. This ensures that all of User i's data will be mapped to the same bucket, while not requiring any communication between two machines simulateneously processing fragments of $\mathbf{X}_i$.

The consequence of this random allocation is a mild inflation of our uncertainty estimators. This is because replicate j is based on k_j observations, where $(k_j)_{j=1}^b$ is a random vector. If the total sample size N is fixed, then $(k_j)_{j=1}^b \sim \mathsf{Multinom}_N(1/b, \ldots, 1/b)$. We later demonstrate that this added variance is negligible for large N.

Alternatively, one could model N as a Poisson random variable with mean λ, whereby $k_j \sim \mathsf{Pois}(n = \lambda/b)$ are i.i.d. In addition to being mathematically convenient, such a model is arguably more realistic in applied problems where N represents a quantity, such as the number of users, which has day-to-day fluctuation and is of interest in its own right. It also provides a natural segue to the streaming bootstrap introduced in the next section. Nevertheless, in order to make minimal assumptions when stating later results, we maintain a fixed sample size view of the streaming buckets procedure – equivalently, we choose to make inference conditional on N.

3. The Poisson bootstrap.

The standard nonparametric bootstrap procedure involves repeated generation of i.i.d. $\mathsf{Multinom}_N(1/N, \ldots, 1/N)$ random weight vectors. This is infeasible when analyzing stream data, both because N is large and because N is not even known until all data have been processed. Modifications to the bootstrap using alternative weight vectors have been considered elsewhere both in generality (Praestgaard and Wellner, 1993), and for specific choices (Rubin, 1981; Owen and Eckles, 2012).

One such approach uses i.i.d. Poisson random weights with mean 1, and is particularly convenient mathematically. We discuss some properties of the Poisson bootstrap in later sections, and its implementation in the MapReduce framework will be outlined in detail in a future paper (Najmi and Naidu, 2012). The basic method has appeared sporadically in the literature in various contexts (Hanley and MacGibbon, 2006; Oza and Russell, 2001; Lee and Clyde, 2004), but has not gained significant traction. It has been viewed primarily as a computational trick, since drawing Poisson random variables can be easier than drawing multinomial vectors. In the streaming setting the latter is actually not only difficult, but impossible. The procedure can also be viewed as a bootstrap with random resample size, which is appealing in its own right.

We briefly describe how to carry out the Poisson bootstrap. For each observation $\mathbf{X}_i$, one draws b independent $\mathsf{Pois}(1)$ random variables $m_{i1}, \ldots, m_{ib}$. The weight m_{ij} describes the number of times that unit i contributes to bootstrap resample j. For the class of statistics described above, the jth bootstrap replicate of T is given by

$$\hat{\theta}_j^\star(\mathbf{X}) = g(\bar{\mathbf{X}}_j^\star) = g\left(\sum_{i=1}^{N} m_{ij}\mathbf{X}_i \Big/ \sum_{i=1}^{N} m_{ij} \right). \tag{17}$$

In order to compute (17) over a streaming data source, it is necessary to ensure that all fragments of data from the ith observation are given the same weight in each resample. As in the previous section, we accomplish this in practice by seeding the random weight sequence $(m_{ij})_{j\geq 1}$ with the user ID hash $f(s_i)$. The low-probability event $\{\sum_i m_{ij} = 0\}$ causes obvious problems in (17); we ignore that technicality for the moment but consider it in some detail in Section 4.

3.1. *Variance estimator.*

The estimator of ξ_N^2 analogous to (10) is given by

$$S_{\mathsf{boot}}^2 = \frac{1}{b} \sum_{j=1}^{b} (\hat{\theta}_j^\star(\mathbf{X}) - \hat{\theta}_N)^2. \tag{18}$$

Note that we do not need a normalization constant in (18) as we did in (10). This may be seen as an advantage of the bootstrap in the rare application where τ_N cannot be posited. An alternative to S_{boot}^2 would replace $\hat{\theta}_N$ with the mean among bootstrap replicates, $\bar{\theta}_b^\star$.

3.2. *Plug-in estimator.* Just as in the streaming buckets procedure, we can approximate the distribution of R_N via the empirical bootstrap replicate distribution

$$G_N^\star(x) = \frac{1}{b}\sum_{j=1}^{b} \mathbb{1}\{\tau_N(\hat\theta_j^\star - \hat\theta_N) \le x\}. \tag{19}$$

As is usual with bootstrap distributions, $G_N^\star(x)$ can be used to estimate more complicated functionals of the distribution of R_N, such as quantiles. It is worth pointing out here that R_N is usually not a pivotal quantity in our applications. This is because while τ_N normalizes the distribution for asymptotic order, it does not remove dependence on ξ_N, which is of course unknown. As a consequence, some higher order properties often associated with the bootstrap cannot be guaranteed in our problems (Hall, 1992).

4. Comparison with non-streaming methods. In this section we consider two important comparisons. Firstly, we examine the properties of the variance estimators S^2_{buck} and S^2_{boot} as they compare to the elusive (for massive data) delta method estimator S^2_Δ. We show that the replicate-based methods perform similarly, and are n times more variable than the delta method. Next, we examine the proximity, in a stochastic sense, of our streaming replication procedures to their non-streaming counterparts, namely subsampling and the multinomial bootstrap. We show that in both cases the streaming and non-streaming versions are closer to each other than either estimator is to the true underlying empirical process defined by G_N.

In what follows we make a slight modification to the algorithms described above. Specifically, if for any replicate j of the bootstrap (respectively, buckets) we obtain by chance an empty resample (respectively, bucket), then the entire procedure should be rejected and started again. The proofs of our main results demonstrate why this is necessary: it boils down to the fact that for a Poisson variate M, $\mathbb{E}[1/M]$ is infinite unless we condition on the event $\{M > 0\}$ (and similarly for a binomial). Though abandoning the analysis in this fashion after computing estimates would seem to violate the spirit of single-pass algorithms, we shall see that this issue is not of practical import because of its exponentially tiny probablity. In the Poisson bootstrap, rejection is equivalent to observing 0 for one of b i.i.d. $\mathsf{Pois}(N)$ random variables; in streaming buckets, it corresponds to a 0 value for the minimum of b independent $\mathsf{Binom}(N, 1/b)$ variates. We could alternatively only reject those replicates for which 0's are obseverved, but this would still require a second pass over the data (or else lead to a random number of replicates b).

4.1. *Bias of the variance.* In the exceptional case that g is a linear function, the variance estimators introduced above are all unbiased. In general, elementary calculations show that the delta method variance estimator is consistent for ξ_N^2 with its bias decaying as $N^{-3/2}$ (i.e. a relative error of order $N^{-1/2}$). There are two sources of this bias: the first is the linearization of g, viz. the approximation $g(\bar{\mathbf{X}}) \approx g(\mu) + \dot{\mathbf{g}}(\mu)(\bar{\mathbf{X}} - \mu)$; the second is a substitution of $\bar{\mathbf{X}}$ for μ inside $\dot{\mathbf{g}}$. The latter error is typically of order N^{-2}, so it is (relatively) unimportant. The following proposition asserts that on average, the Poisson bootstrap does not do significantly worse than the delta method, while the streaming buckets procedure does only slightly worse.

PROPOSITION 1. *Let $B_N(S^2) = B_N(S^2, \xi_N^2) = \mathbb{E}[S^2] - \xi_N^2$ denote the bias of an estimator S^2 of ξ_N^2. Suppose that b, N, and $n = N/b$ all tend to infinity. Then:*

1. $B_N(S^2_{\mathsf{boot}}) = B_N(S^2_\Delta) = O(N^{-3/2})$
2. $B_N(S^2_{\mathsf{buck}}) = O(b^{1/2}N^{-3/2})$.

The extra factor of $b^{1/2}$ in the bias of the buckets procedure is precisely the penalty paid for assuming that $g(\bar{\mathbf{X}}_j)$ has approximately the same distribution, suitably rescaled, as $g(\bar{\mathbf{X}})$. Clearly, this assumption gets better as b approaches 1, and can be very poor when $b \sim N$, so that each observation is its own bucket. Nevertheless, Proposition 1 asserts that S^2_{buck} is consistent (in a relative error sense), as long as $b = o(N)$. In the example presented in Section 5, the additional bias in S^2_{buck} is negligible.

4.2. *Variance of the variance.* As seen in the previous section, the delta method does not necessarily offer any advantage over replication-based methods in terms of asymptotic consistency. Where its merits become evident is in the stability of the variance estimates produced for massive data. Note that for the sample mean case, where $g(x) = x$, it is well known (Kendall and Stuart, 1969) that $\mathsf{Var}(S^2_\Delta) = [(N-1)/N]^2(2\mathsf{Var}(\bar{X})^2/(N-1) + \kappa_4(\bar{X}))$. The following proposition begins by generalizing that result.

PROPOSITION 2.

$$\mathsf{Var}\left(S^2_\Delta\right) = \left(\frac{2\xi_N^4}{N} + \kappa_4(g(\bar{\mathbf{X}})) + \frac{R_g(F)}{N^3}\right)\left[1 + O\left(N^{-1/2}\right)\right], \tag{20}$$

where $R_g(F)$ depends on the first three derivatives of g at μ, and on moments of $\mathbf{X}$ up to order 3. In particular, if g is a linear mapping then $R_g(F) = 0$.

Suppose that N, b, and $n = N/b$ all tend to infinity. Then

$$\mathsf{Var}\left(S^2_{\mathsf{boot}}\right) = \frac{2\xi_N^4}{b}[1 + o(1)]. \tag{21}$$

If, in addition, $b = o(N^{1/2})$, then

$$\mathsf{Var}\left(S^2_{\mathsf{buck}}\right) = \frac{2\xi_N^4}{b}[1 + o(1)]. \tag{22}$$

Hence the relative efficiency of the delta method vs. replicate-based variance estimation is given by

$$\mathsf{eff}\left(S^2_\Delta, S^2_{\mathsf{boot\ (buck)}}\right) = O\left(n\right) \text{ as } N, b\ (\textit{and } n/b) \to \infty. \tag{23}$$

More details are given in the Appendix, but if $b \geq cN^{1/2}$ and g is non-linear, then the relative error in (22) is of order b/n, and therefore $\mathsf{Var}(S^2_{\mathsf{buck}}) = O(bN^{-3})$ instead of $O(b^{-1}N^{-2})$. While S^2_{boot} can only get better as $b \to \infty$, Propositions 1 and 2 suggest that a qualitative change in the behaviour of S^2_{buck} occurs if b grows too quickly relative to N. The following corollary makes this more precise. It simply says that we can gain precision in the bucketed variance estimator by increasing the number of buckets, but not beyond that attained at the square root of the sample size.

Corollary 3. *For a given g, G_N and N, let b_{min} denote the value of $b = b(N)$ yielding the smallest possible mean-square error for the bucket variance estimator S^2_{buck}. Then $b_{\mathsf{min}} = O(N^{1/2})$ as $N \to \infty$.*

It is worth noting that the theoretical turning point of $b \sim N^{1/2}$ is unlikely to be surpassed in the applications we have in mind. Even if $N = 10^9$, this corresponds to more than 30,000 bucket replicates. Surely, in the vast majority of problems one would obtain adequate inference, even accurate percentile-based confidence intervals, with many fewer replicates than this. From a storage cost perspective then, any additional accuracy would rarely justify such a large b.

4.3. *Empirical processes.* In this section we demonstate that as random processes, and subject to rejection in the case of zero sample size replicates, the Poisson bootstrap "agrees" with the multinomial bootstrap to within $o_p(N^{-1/2})$, and streaming buckets with a flavour of subsampling to within $o_p(n^{-1/2})$. These facts, made more precise in Theorem 4, imply that the proposed streaming methods immediately inherit the first-order properties of their predecessors. The need for a rejection clause stems from the fact that the exchangeable sampling weights cannot otherwise be normalized to sum to 1 with probability 1. We note here that the powerful main result in Praestgaard and Wellner (1993) can be used to show consistency of the Poisson bootstrap and streaming buckets empirical processes. Our Theorem 4 is a stronger result, proved indirectly by appealing to the ordinary bootstrap and subsampling empirical processes.

We first define the Wasserstein distance between two probability distributions having distribution functions G_1 and G_2. This is defined relative to some Donsker class of functions $\mathcal{F}$ (see Van der Vaart (1998) for a gentle introduction), as

$$\|G_1 - G_2\|_{\mathcal{W}} = \sup_{H \in \mathsf{BL}_1(\mathcal{F})} \left|\mathbb{E}[H(G_1) \mid \mathbf{X}] - \mathbb{E}[H(G_2) \mid \mathbf{X}]\right|, \tag{24}$$

Where $\mathsf{BL}_1(\mathcal{F})$ denotes the space of functions $H : \ell^\infty(\mathcal{F}) \to \mathbb{R}$ which are uniformly Lipschitz with constant 1. Here we can think of $\mathcal{F}$ as the set of indicator functions $\mathbb{1}_{(-\infty,x]}$, $x \in \mathbb{R}$.

Let $\hat{F} = \hat{F}_N$ denote the empirical distribution function of F. We also introduce the following notation. Let $F^\star_N$ denote the multinomial bootstrap distribution, $F^{\star\star}_N$ the Poisson bootstrap distribution, $F^\bullet_N$ the disjoint block subsampling distribution, and $F^{\bullet\bullet}_N$ the streaming buckets distribution. More explicitly, given a random sample $\mathbf{X}$ from F, letting $\rho \sim \mathsf{Unif}(S_N)$ denote a random permutation of $\{1, \dots, N\}$, and assuming for notational convenience that $N/b = n \in \mathbb{N}$, we have:

$$d\hat{F}_N(\mathbf{x}) = \sum_{i=1}^N \delta_{\mathbf{X}_i}(\mathbf{x})/N; \tag{25}$$

$$dF^\star_N(\mathbf{x}) = \sum_{i=1}^N w^\star_i \delta_{\mathbf{X}_i}(\mathbf{x}),\ \ w^\star \sim \mathsf{Multinom}_N(\mathbf{1}/N)/N; \tag{26}$$

$$dF^{\star\star}_N(\mathbf{x}) = \sum_{i=1}^N w^{\star\star}_i \delta_{\mathbf{X}_i}(\mathbf{x}),\ \ w^{\star\star}_i = m_i / \sum_{i=1}^N m_i,\ \ m_i \overset{\text{i.i.d.}}{\sim} \mathsf{Pois}(1); \tag{27}$$

$$dF_N^{\bullet}(\mathbf{x}) = \sum_{i=1}^{N} w_i^{\bullet}\delta_{\mathbf{X}_i}(\mathbf{x}),\ \ w_i^{\bullet} = \mathbb{1}\{\rho(i) \le n\}/n; \tag{28}$$

$$dF_N^{\bullet\bullet}(\mathbf{x}) = \sum_{i=1}^{N} w_i^{\bullet\bullet}\delta_{\mathbf{X}_i}(\mathbf{x}),\ \ w_i^{\bullet\bullet} = \mathbb{1}\{\rho(i) \le k\}/k,\ \ k = \sum_{i=1}^{N} k_i, k_i \overset{\text{i.i.d.}}{\sim} \mathsf{Bern}(1/b). \tag{29}$$

Note that definitions (27) and (29) are subject to the same rejection constraints described earlier in this section; in other words, they are conditional on $\{\sum_i m_i > 0\}$ and $\{\sum_i k_i > 0\}$, respectively. Applying the consistency result in Praestgaard and Wellner (1993) to the processes (27) or (29) would require (in their notation) setting $W_i = Nw_i^{\star\star}$ or $W_i = nw_i^{\bullet\bullet}$, respectively.

Statements about the asymptotic properties of the bootstrap (respectively, subsampling) are generally proved by bounding the difference, suitably scaled, between $\hat{F} - F$ and $F^{\star} - \hat{F}$ (respectively, $F^{\bullet} - \hat{F}$) asymptotically. Specifically, these converge to a particular Gaussian process at the rate of $\sqrt{N}$ (respectively, $\sqrt{n}$) (Van der Vaart, 1998; Politis, Romano and Wolf, 1999) . The following result piggy-backs on that basic method of proof by approximating the difference between $F^{\star\star} - \hat{F}$ and $F^{\star} - \hat{F}$, and that between $F^{\bullet\bullet} - \hat{F}$ and $F^{\bullet} - \hat{F}$.

THEOREM 4. *Let* $\mathbf{X}$ *be a random sample of size* N *drawn from* F*, with the derived distributions defined by* (25)-(29)*. Take* $F_N^{\star\star}$ *to be conditional on the event* $\{\sum_{i=1}^N m_i > 0\}$*, and* $F_N^{\bullet\bullet}$ *to be conditional on* $\{\sum_{i=1}^N k_i > 0\}$*. For simplicity assume that* $\tau_N = N^{1/2}$*.*

1. *(Proximity of Poisson and multinomial bootstrap)*

$$\left\|\sqrt{N}(F_N^{\star\star} - \hat{F}_N) - \sqrt{N}(F_N^{\star} - \hat{F}_N)\right\|_{\mathcal{W}} = O_p(N^{-1/4}) \tag{30}$$

2. *(Proximity of streaming buckets and disjoint block subsampling)*

$$\left\|\sqrt{n}(F_N^{\bullet\bullet} - \hat{F}_N) - \sqrt{n}(F_N^{\bullet} - \hat{F}_N)\right\|_{\mathcal{W}} = O_p(n^{-1/4}) \tag{31}$$

An equivalent statement of (30) and (31) is that the differences $\left\|\sqrt{N}(F_N^{\star\star} - F_N^{\star})\right\|_{\mathcal{W}}$ and $\|\sqrt{n}(F_N^{\bullet\bullet} - F_N^{\bullet})\|_{\mathcal{W}}$ are also of order $N^{-1/4}$ in probability. Thus Theorem 4 settles the issue of the first-order validity of the streaming methods introduced in this paper, by demonstrating that they are in general strictly closer to their non-streaming counterparts than the latter procedures are to the 'truth', where truth is defined as the distribution of R_N. Note that since $\sum_i m_{ij} \sim \mathsf{Pois}(N)$ and $\sum_i k_{ij} \sim \mathsf{Binom}(N, 1/b) \approx \mathsf{Pois}(n)$, the probablity of having to make extra passes through the data is bounded by be^{-N} for the Poisson bootstrap and be^{-n} for streaming buckets. Even if $b \sim N^c$ for some arbitrary $c > 0$, this probability is exponentially small.

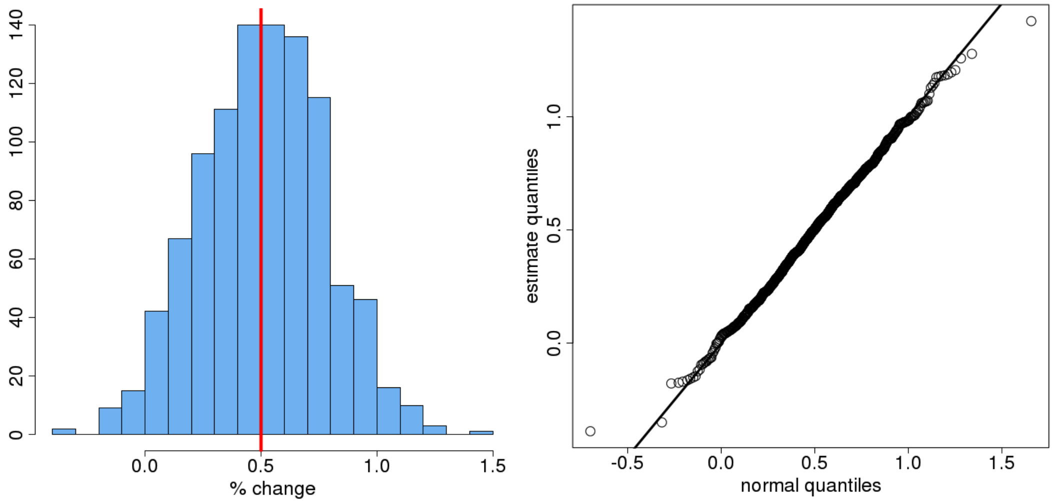
5. Simulations and data. Here we present the results of a hybrid simulation and real data example. Because simulating massive data streams on a single machine is infeasible, we used actual Google data and injected randomness by (a) creating synthetic 'experiments' based on a hash of the user ID, and (b) randomly perturbing the data for users in the 'treatment' group. Assignment to treatment was done orthogonally to the hash function f used for our replication algorithms, similarly to how real Google experiments are run (Tang et al., 2010). By construction, the record unit in this application is a query, while the correct exchangeable unit is the user. We created independent 'iterations' of this simulation using yet another orthogonal hash function. The result was 1000 identically distributed copies of a synthetic experiment data set, with average size equal to 1% of daily Google traffic, drawn from a single pass over the data. There was slight dependence between these copies, so the observed variability in our confidence intervals is only a lower bound for what would typically occur in practice. Note that 1% is on the low end of the range of sample sizes that we encounter in real applications; sample sizes 3 orders of magnitude higher are not uncommon.

The parameter ρ estimated in these simulations is a relative treatment difference of ratio metrics, i.e. $\mathbf{x} = (x, y, z, w)$ and $g(\mathbf{x}) = (y/x)/(w/z) - 1$. Thus, $\bar{Y}/\bar{X}$ (respectively $\bar{W}/\bar{Z}$) is our estimate of the parameter of interest in the treatment group (respectively, control group). Confidentiality prevents us from giving full details, but each ratio is a measure of user satisfaction with their interaction with ads on Google search. The treatment effect was induced by multiplication of Y by an indpendent random variable with mean 1.005; thus $\rho = 0.5\%$. In each iteration, the streaming buckets and Poisson boostrap procedures were applied with $b = 1000$. This allowed us to easily compute replicate-based inference for any b which is either less than 1000 (in the case of bootstrap), or a divisor of 1000 (in the case of buckets). We show results for $b = 20, 100, 1000$ here.

Note that since (X_i, Y_i) describes outcomes in the treatment, and (Z_i, W_i) describes outcomes in the control, either $X_i = Y_i = 0$ or $Z_i = W_i = 0$ for every unit i—a unit cannot be simultaneously in the two groups. We therefore considered both the naïve variance and plug-in estimators (which needlessly join pairs of treatment and control replicates), and the unpaired refinement in which the sample variance is computed from all b^2 combinations of control and treatment replicates. The latter estimator is not discussed in detail here, but can be shown to be about twice as efficient in certain situations (Chamandy and Muralidharan, 2012). In all cases we constructed 90% confidence intervals for ρ on the percent scale. A null ($\rho = 0$) simulation yielded similar results, and is not shown.

Figure 3 presents a visualization of the 'standard' symmetric, equal-tailed intervals computed using the simple variance estimators (10) and (18), and the normal quantile $\Phi^{-1}(0.95)$. Similar plots appearing in Figure 4 show 'percentile' intervals, computed based on the 5th and 95th percentiles of the replicate distributions (15) and (19) (save for the delta method panels in the top row, which assume normality and thus are computed symmetrically in the standard way). Two variants of the delta method procedure are also shown for comparison. The first is easlily computable, but naïvely assumes independence between queries issued by the same user, and underestimates the variance accordingly. The second is computed with the aid of an expensive intermediate data join, and assigns data to replicates at the user-level. Effectively this join creates a new data source whose record unit is the exchangeable one, that is the user—such a procedure is too expensive for most applications.

The visualization technique is a modification of that of Franz (2007). In the plots, each



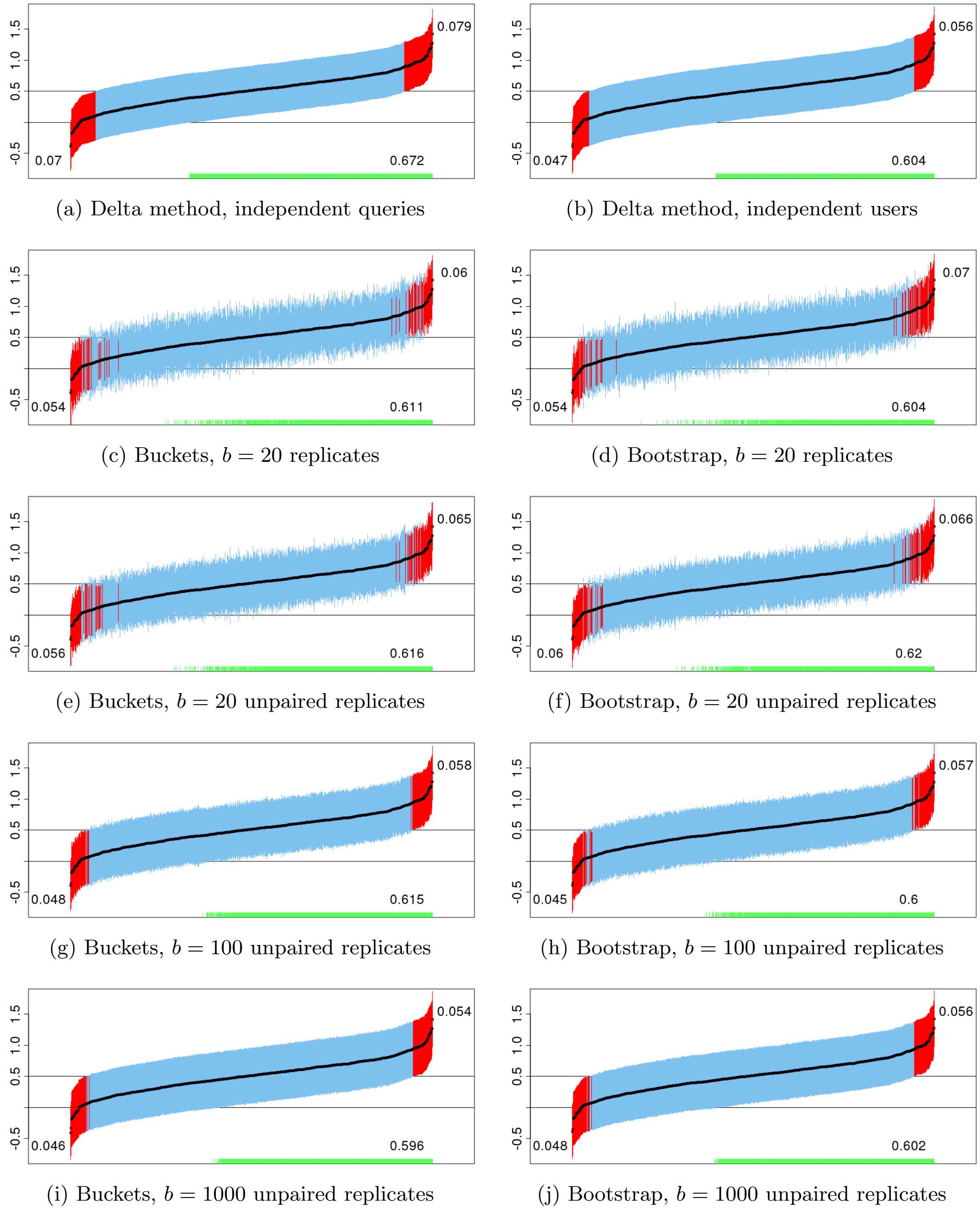
(a) Histogram of simulated parameter estimates (b) Quantile-quantile plot of simulated estimates

Fig 2: The empirical distribution of 1000 simulated estimates $\hat{\rho} = [(\bar{Y}_1/\bar{X}_1)/(\bar{Y}_0/\bar{X}_0) - 1] \times 100\%$. Both the histogram (a) and q-q plot (b) indicate that a normal approximation may be adequate.

vertical segment represents a single iteration of the specified interval estimation procedure, with red indicating non-coverage of the true parameter value. Green indicates detection of a non-zero treatment effect. The intervals are sorted by their point estimate, so that jaggedness in the envelope of the segments illustrates variability in the procedure. The efficiency gained in estimating variance via the delta method is evident in the top row of the figures, where the curves traced by upper and lower interval endpoints are very smooth. However, its practical advantage is more or less gone once we have reached $b = 1000$ replicates in this example. The intervals built from only 20 replicates perform poorly for these data, especially when the percentile method is used—although unpairing the replicates noticeably improves the results, as it does for larger b (not shown).

The statistic chosen for this example is relatively well-behaved—see Figure 2—and overall 1000 replicates appears adequate for interval estimation. It is worth noting that because the statistic is approximately normal, the 'fight' between the delta and replicate-based methods in this example is not a fair one. The latter estimators can at best hope to approach the smoothness of the delta method by paying for more replicates, but cannot really improve in terms of average interval length or coverage. In problems where normality is a poor approximation, the delta method is very stable, but its confidence interval is biased by construction since it ignores higher order cumulants. Therefore without modification, say using Edgeworth expansion, the delta method cannot hope to compete with the percentile replicate-based intervals since they have asymptotically correct coverage as $b \to \infty$.

6. Future work. As mentioned in Section 4, the streaming buckets and bootstrap procedures introduced in this paper are but two instances in a richer class of algorithms.



(a) Delta method, independent queries

(b) Delta method, independent users

(c) Buckets, $b = 20$ replicates

(d) Bootstrap, $b = 20$ replicates

(e) Buckets, $b = 20$ unpaired replicates

(f) Bootstrap, $b = 20$ unpaired replicates

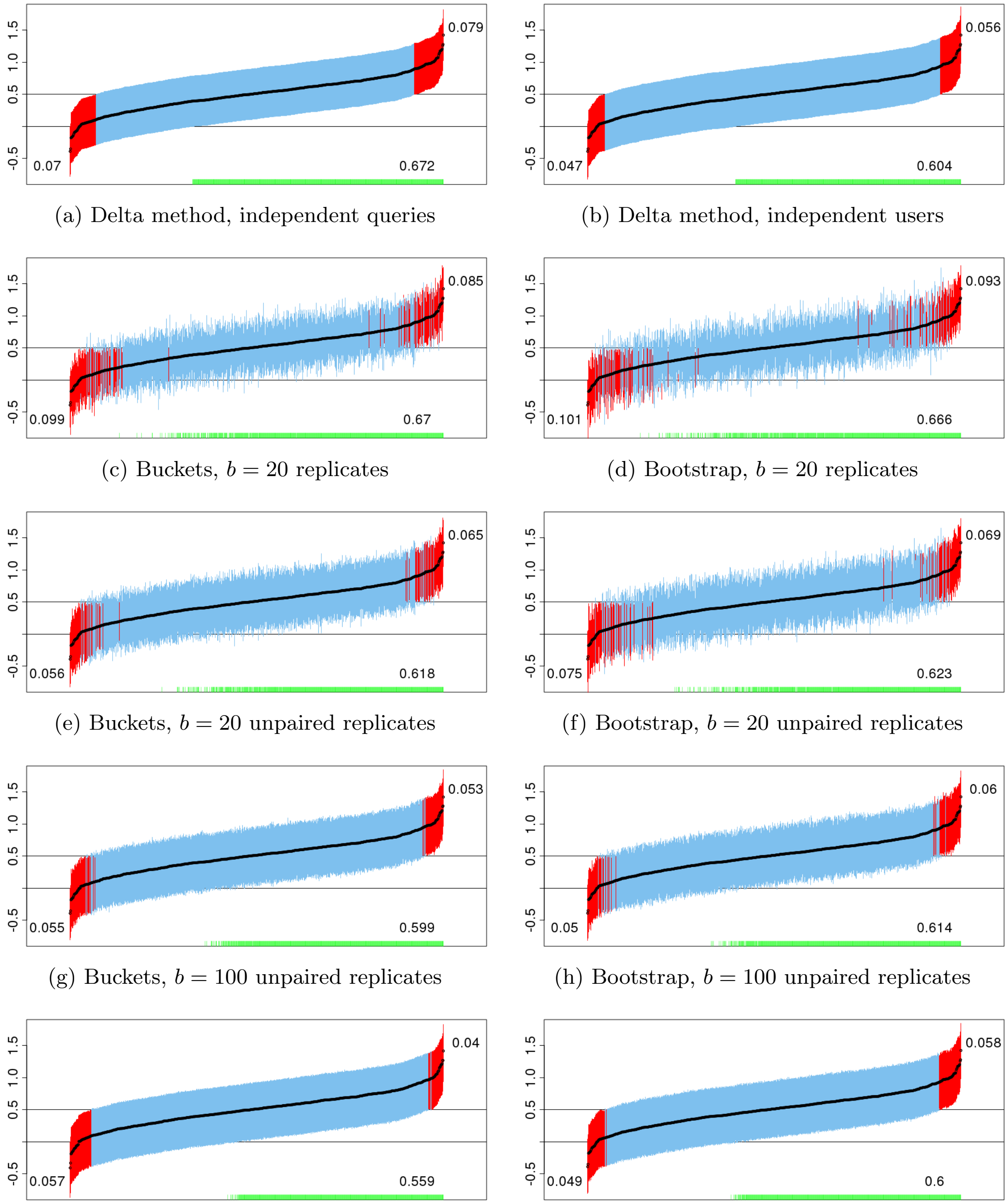
(g) Buckets, $b = 100$ unpaired replicates

(h) Bootstrap, $b = 100$ unpaired replicates

(i) Buckets, $b = 1000$ unpaired replicates

(j) Bootstrap, $b = 1000$ unpaired replicates

Fig 3: Confidence intervals with nominal 90% coverage, constructed by both the delta method and ‘standard’ method, assuming a normal distribution for the replicates. Each vertical segment is one of 1000 iterations. (Red) Blue indicates (non-)coverage of the true experimental effect $\rho = 0.5\%$. Green rug indicates detection of a non-zero effect. Empirical non-coverage in each tail and power are printed in each panel. Vertical axis is on the percent scale.



(a) Delta method, independent queries

(b) Delta method, independent users

(c) Buckets, $b = 20$ replicates

(d) Bootstrap, $b = 20$ replicates

(e) Buckets, $b = 20$ unpaired replicates

(f) Bootstrap, $b = 20$ unpaired replicates

(g) Buckets, $b = 100$ unpaired replicates

(h) Bootstrap, $b = 100$ unpaired replicates

(i) Buckets, $b = 1000$ unpaired replicates

(j) Bootstrap, $b = 1000$ unpaired replicates

Fig 4: Confidence intervals with nominal 90% coverage, constructed by both the delta method and basic percentile method. Each vertical segment is one of 1000 iterations. (Red) Blue indicates (non-)coverage of the true experimental effect $\rho = 0.5\%$. Green rug indicates detection of a non-zero effect. Empirical non-coverage in each tail and power are printed in each panel. Vertical axis is on the percent scale.

These can generally be described in the following steps:

1. Choose an array of non-negative weights $\mathbf{W}$ of dimensions $N \times b$, both tending to infinity, as well as a sequence of triangular arrays of scaling constants $(\tau^b_{j,N})^b_{j=1}$;
2. Determine a streaming method for generating the weights $\mathbf{W}$;
3. Pass over the data stream, and for any record belonging to Unit i, update the jth replicate of the estimator, $\hat{\theta}_{j,N}$, exactly w_{ij} times;
4. Approximate the target distribution of $\tau_N(\hat{\theta}_N - \theta)$ by the empirical distribution of $\tau^b_{j,N}(\hat{\theta}_{j,N} - \hat{\theta}_N), j = 1, \dots, b$.

The weight array $\mathbf{W}$ can be quite general, and as we have illustrated can be determinstic (in the case of buckets) or i.i.d. (Poisson bootstrap). One fundamental property which must be satisfied for massive data streams is independence of the rows of $\mathbf{W}$, which avoids the need for cross-machine communication, and makes Item 2 feasible. Note that the columns of $\mathbf{W}$ need not be independent; indeed they are not in the streaming buckets algorithm. An interesting open statistical problem is to enumerate a minimal set of weak conditions on $\mathbf{W}$ in order to make the streaming replication algorithm described above 'work', perhaps in the spirit of Theorem 4.

We believe that massive data streams have brought with them a New Frontier in statistics, and indeed a shift in the way statisticians must think about inference. Such structures make certain operations almost trivial (e.g. density estimation, Bayesian priors), while rendering others hopelessly infeasible (e.g. the delta method, multinomial sampling). But in some cases, even when a traditional method fails badly for streaming data, a subtle modification thereof may be sufficient to save the day.

References.

Bickel, P. J., Götze, F. and van Zwet, W. R. (1997). Resampling fewer than n observations: Gains, losses, and remedies for losses. *Statistica Sinica* **7** 1–31.

Brillinger, D. R. (1969). The calculation of cumulants via conditioning. *Annals of the Institute of Statistical Mathematics* **21** 215–218.

Carlstein, E. (1986). The use of subseries values for estimating the variance of a general statistic from a stationary sequence. *The Annals of Statistics* **14** 1171–1179.

Chamandy, N. and Muralidharan, O. (2012). More bang for your buckets. *Technical Report, Google.*

Dean, J. and Ghemawat, S. (2004). MapReduce: simplified data processing on large clusters. *Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation* 137–149.

Efron, B. (1982). *The Jackknife, the Bootstrap, and other Resampling Plans.* SIAM Monograph #38, CBMS-NSF.

Franz, V. H. (2007). Ratios: A short guide to confidence limits and proper use. [arXiv:0710.2024v1].

Glynn, P. W. (1986). Upper bounds on Poisson tail probabilities. *Operations Research Letters* **6** 9–14.

Hall, P. (1992). *The Bootstrap and Edgeworth Expansion.* Springer, New York.

Hanley, J. A. and MacGibbon, B. (2006). Creating non-parametric bootstrap samples using Poisson frequencies. *Computer Methods and Programs in Biomedicine* **83** 57–62.

Hurt, J. (1976). Asymptotic Expansions of Functions of Statistics. *Aplikace Matematiky* **21** 444–456.

Kendall, M. G. and Stuart, A. (1969). *The Advanced Theory of Statistics, Vol. 1.* Griffin, New York.

Kleiner, A., Talwakar, A., Sarkar, P. and I., J. M. (2011). A Scalable Bootstrap for Massive Data. [arXiv:1112.5016v1].

Lee, H. K. H. and Clyde, M. A. (2004). Online Bayesian Bagging. *The Journal of Maching Learning Research* **5** 143–151.

Mudholkar, G. S. and Trivedi, M. C. (1981). A Gaussian Approximation to the Distribution of the Sample Variance for Nonnormal Populations. *Journal of the American Statistical Association* **76** 479–485.

Najmi, A. (2010). Introduction to the Poisson bootstrap. *Technical report, Google.*

Najmi, A. and Naidu, S. (2012). Implementing the Poisson bootstrap. *In preparation.*

Oehlert, G. W. (1992). A Note on the Delta Method. *The American Statistician* **46** 27–29.

Owen, A. B. and Eckles, D. (2012). Bootstrapping data arrays of arbitrary order. [arXiv:1106.2125].

Oza, N. C. and Russell, S. (2001). Online bagging and boosting. *Artificial Intelligence and Statistics* 105-112.

Politis, D. N., Romano, J. P. and Wolf, M. (1999). *Subsampling.* Springer, New York.

Praestgaard, J. and Wellner, J. A. (1993). Exchangeably weighted bootstraps of the general empirical process. *The Annals of Probability* **21** 2053–2056.

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

Tang, D., Agarwal, A., O'Brien, D. and Meyer, M. (2010). Overlapping Experiment Infrastructure: More, Better, Faster Experimentation. In *Proceedings 16th Conference on Knowledge Discovery and Data Mining* 17–26.

Van der Vaart, A. W. (1998). *Asymptotic statistics.* Cambridge University Press, New York.

Appendix. Throughout this section we use the term *delta method* (often just DM) to refer to the variant of Theorem 1 or 2 in Hurt (1976) which replaces his boundedness condition on h with the polynomial boundedness requirement described in Oehlert (1992), which is weaker than our own assumptions. (We write h in the place of Hurt's g, since for us g is already reserved for the original function of interest.)

We state the following lemma for use in the proofs below.

Lemma 5. *Let $M \sim \mathsf{Pois}(N)$ and $k \sim \mathsf{Binom}(N, 1/b)$. Suppose N, b, and $n = N/b$ all tend to infinity. Then*

1. $\mathbb{E}\left[|(N-1)/M - 1| | M > 0\right] = O(N^{-1/2})$
2. $\mathbb{E}\left[|N/M - 1|^{1/2} | M > 0\right] = O(N^{-1/4})$
3. $\mathbb{E}\left[|n/k - 1| | k > 0\right] = O(n^{-1/2})$
4. $\mathbb{E}\left[|n/k - 1|^{1/2} | k > 0\right] = O(n^{-1/4})$

Proof of Lemma 5. We first prove that $\mathbb{E}\left[|N/M - 1| | M > 0\right] = O(N^{-1/2})$. This is sufficient to establish 1 and 2 of the lemma by Jensen's inequality and the observation $|(N-1)/M - 1| \leq [(N-1)/N]|N/M - 1| + 1/N$. We start by evaluating the expectation

$$\mathbb{E}\left[\left|\frac{N}{M} - 1 + \frac{M-N}{N}\right| |M > 0\right] = \mathbb{E}\left[|Y_N| | M > 0\right]. \tag{32}$$

Note that $Y_N = (N-M)^2/(NM)$. We partition the possible values of M into three cases by defining $\delta = \sqrt{3 \log N / N} \to 0$: $A = \{1 \leq M < (1-\delta)N\}$; $B = \{(1-\delta)N \leq M \leq (1+\delta)N\}$; $C = \{(1+\delta)N < M\}$. We then apply the results of Glynn (1986) to bound the expectations on A and C, as follows. Using Propositions 1(i) and 2(i) from Glynn (1986), we have

$$\mathbb{P}\{A\} \leq \mathbb{P}\{1 \leq M \leq \lfloor (1-\delta)N \rfloor\} \tag{33}$$
$$\leq \mathbb{P}\{M = \lfloor (1-\delta)N \rfloor\}/\delta \tag{34}$$

$$\le \mathbb{P}\{M = N - (\delta N - 1)\}/\delta \tag{35}$$
$$\le (2\pi N)^{-1/2} \exp\{-(\delta N - 1)(\delta N - 2)/2N\}/\delta \tag{36}$$
$$= (8\pi \log N)^{-1/2} \exp\{-\delta^2 N/2 + 3\delta/2 - 1/N\} \tag{37}$$
$$\le K_{A,1}(\log N)^{-1/2} \exp\{-\delta^2 N/2\} \tag{38}$$
$$= K_{A,1}(\log N)^{-1/2} N^{-3/2} \tag{39}$$

for some constant $K_{A,1}$ and large enough N. Note that on A, since $M \ge 1$, we can also bound $|Y_N|$ by $(N-1)^2/N \le K_{A,2}N$. Thus, for large enough N, $\mathbb{E}[|Y_N|\mathbb{1}_A] \le K_A(\log N)^{-1/2}N^{-1/2}$. Similarly, we can use Propositions 1(ii) and 2(ii) from Glynn (1986) to establish that

$$\mathbb{P}\{C\} \le P\{M \ge \lceil (1+\delta)N \rceil\} \tag{40}$$
$$\le \mathbb{P}\{M = \lceil (1+\delta)N \rceil\} \left(1 - \frac{N}{\lceil (1+\delta)N \rceil + 1}\right)^{-1} \tag{41}$$
$$\begin{aligned} &\le (2\pi N)^{-1/2} \frac{1+\delta+1/N}{\delta + 1/N} \exp\{-\lceil \delta N \rceil (\lceil \delta N \rceil - 1)/2N \\ &\qquad\qquad + \lceil \delta N \rceil (\lceil \delta N \rceil - 1)(2\lceil \delta N \rceil - 1)/(12N^2)\} \end{aligned} \tag{42}$$
$$\le K_{C,1}\frac{1+\delta}{\delta} N^{-1/2} \exp\{-\delta^2 N/2 + \delta/2\} \tag{43}$$
$$\le K_{C,2}(\log N)^{-1/2} N^{-3/2}. \tag{44}$$

Use $|Y_N| \le |N/M - 1| + |(M-N)/N| \le 1 + N^{-1/2}|N^{-1/2}(M-N)|$ on C. Since $\mathbb{E}|N^{-1/2}(M-N)| \le 1$, it follows that $\mathbb{E}\{|Y_N|\mathbb{1}_C\} \le K_{C,2}(\log N)^{-1/2}N^{-3/2} + N^{-1/2} \le K_{C,3}N^{-1/2}$. Finally, on B we have $Y_N = (N-M)^2/(NM) \le \delta^2 N^2/[(1-\delta)N^2] = \delta^2/(1-\delta) \le k_B \log N/N$. Thus $\mathbb{E}\{|Y_N|\mathbb{1}_B\} \le k_B \log N/N$. Putting this all together, we have $\mathbb{E}\left[|Y_N|\middle| M > 0\right] = \mathbb{P}\{M > 0\}^{-1}\mathbb{E}[|Y_N|\mathbb{1}_{\{M>0\}}] = O(N^{-1/2})$. Therefore

$$\mathbb{E}\left[|N/M - 1| | M > 0\right] \le E\{|Y_N| | M > 0\} + N^{-1/2}\mathbb{E}\{|M-N|/N\} \le KN^{-1/2} \tag{45}$$

for some K, establishing 1 and 2 of the lemma.

Parts 3 and 4 of the lemma are proved very similarly, except that we can make use of the well-known Chernoff bounds for a sum of Bernoulli random variables rather than the somewhat clumsier Poisson bounds. That is, we set $\delta = \sqrt{3\log n/n}$, partition the expectation by comparing k to $(1 \pm \delta)n$, and use

$$\mathbb{P}\{k < (1-\delta)n\} \le e^{-\delta^2 n/2} = n^{-3/2}; \quad \mathbb{P}\{k > (1+\delta)n\} \le e^{-\delta^2 n/3} = n^{-1}. \tag{46}$$

We omit the remaining details. □

PROOF OF PROPOSITION 1. We assume that $p = 1$ for simplicity of exposition, but the proof of the general case is similar. As mentioned, we make heavy use of the results outlined by Oehlert (1992), who cites Hurt (1976).

We can decompose the target ξ_N^2 as follows:

$$\mathsf{Var}(g(\bar{X})) = \mathbb{E}\left[\left(g(\bar{X}) - g(\mu)\right)^2\right] - \left(\mathbb{E}g(\bar{X}) - g(\mu)\right)^2. \tag{47}$$

The second term in (47) is of order $O(N^{-2})$ by Equation (2) in Oehlert (1992) (a single term delta method). The first term can be handled by applying Equation (3) in Oehlert (1992) with the function $h(x) = (g(x) - g(\mu))^2$, and taking $q = 2$. This yields

$$\mathsf{Var}(g(\bar{X})) = h(\mu) + h'(\mu)\mathbb{E}(\bar{X} - \mu) + h''(\mu)\mathsf{Var}(\bar{X})/2 + O(N^{-3/2}) \tag{48}$$

$$= g'(\mu)^2\mathsf{Var}(\bar{X}) + O(N^{-3/2}) \tag{49}$$

$$= S^2_\Delta + [g'(\mu)^2 - g'(\bar{X})^2]\mathsf{Var}(\bar{X}) + O(N^{-3/2}). \tag{50}$$

Applying the same result to the function $h(x) = g'(x)^2$, and $q = 1$, we see that $g'(\bar{X})^2 - g'(\mu)^2 = O(N^{-1})$, whereby $\mathsf{Var}(g(\bar{X})) = S^2_\Delta + O(N^{-3/2})$.

Next, we consider $\mathbb{E}(S^2_{\mathsf{boot}}) = \mathbb{E}\left[(g(\bar{X}^\star) - g(\bar{X}))^2\right]$, where $\bar{X}^\star$ denotes a generic resampled sample mean. Note that $\sqrt{N}((\bar{X}^\star, \bar{X})' - (\mu, \mu)') \xrightarrow{\mathcal{D}} \mathsf{N}(0, \Sigma)$, where

$$\Sigma = \frac{\sigma^2}{N}\begin{pmatrix} 1 + (N-1)\mathbb{E}[1/M] & 1 \\ 1 & 1 \end{pmatrix}. \tag{51}$$

We can therefore apply the multivariate DM (see Theorem 2 in Hurt (1976)), with the function of interest being $h(a, b) = (g(a)-g(b))^2$. Note that $h(\mu, \mu) = 0$ and its derivatives are given by

$$\begin{aligned} h_a(a, b) &= 2g'(a)(g(a) - g(b)) \\ h_b(a, b) &= -2g'(b)(g(a) - g(b)) \\ h_{aa}(a, b) &= 2[g''(a)(g(a) - g(b)) + g'(a)^2] \\ h_{bb}(a, b) &= 2[g''(b)(g(a) - g(b)) - g'(b)^2] \\ h_{ab}(a, b) &= -2g'(a)g'(b). \end{aligned}$$

Thus $h_a(\mu, \mu) = h_b(\mu, \mu) = 0$, and $h_{aa} = h_{bb} = 2g'(\mu)^2 = -h_{ab}$ at (μ, μ).

The conclusion of DM is therefore that

$$\mathbb{E}\left[(g(\bar{X}^\star) - g(\bar{X}))^2\right] = \mathbb{E}\left[h(\bar{X}^\star, \bar{X}) - h(\mu, \mu)\right] \tag{52}$$

$$= \frac{g'(\mu)^2\sigma^2}{N}(N-1)\mathbb{E}[1/M] + O(N^{-3/2}) \tag{53}$$

$$= \left\{\mathsf{Var}(g(\bar{X})) + O(N^{-3/2})\right\}(N-1)\mathbb{E}[1/M] + O(N^{-3/2}). \tag{54}$$

[Note that going to $q = 3$ in the above application of Theorem 2 in Hurt (1976) shows the error in (53) is actually $O(N^{-2})$, a fact which we use later.] Now we need only to apply, from Lemma 5, that $(N-1)\mathbb{E}[1/M] = 1 + O(N^{-1/2})$ to complete the proof of part 1.

For part 2, consider the jth term of S^2_{buck}. We apply the same trick as above, but to the random vector $\left(\bar{X}_j, \sqrt{b}\bar{X}\right) \sim \left((\mu, \sqrt{b}\mu), \frac{\sigma^2}{n}\begin{pmatrix} \mathbb{E}[n/k_j] & b^{-1/2} \\ b^{-1/2} & 1 \end{pmatrix}\right)$, and the function $h(u, v) = \left(g(u) - g(v/\sqrt{b})\right)^2$. We have

$$\mathbb{E}\left[(g(\bar{X}_j) - g(\bar{X}))^2\right] = \mathbb{E}[h(\bar{X}_j, \bar{X}))] \tag{55}$$

$$= \frac{g'(\mu)^2\sigma^2}{n}\left\{\mathbb{E}\left[\frac{n}{k_j}\right] - \frac{1}{b}\right\} + O(n^{-3/2}) \tag{56}$$

$$= \frac{g'(\mu)^2\sigma^2}{n}\left\{1 - \frac{1}{b}\right\} + O(n^{-3/2}). \tag{57}$$

Summing over j and dividing by $b(b-1)$ gives

$$\mathbb{E}[S^2_{\mathsf{buck}}] = \frac{g'(\mu)^2\sigma^2}{N}\frac{b}{b-1}\left\{1 - \frac{1}{b}\right\} + O(b^{-1}n^{-3/2}) \tag{58}$$

$$= \frac{g'(\mu)^2\sigma^2}{N} + O(b^{1/2}N^{-3/2}) \tag{59}$$

$$= \mathsf{Var}(\bar{X}) + O(b^{1/2}N^{-3/2}). \tag{60}$$

□

Proof of Proposition 2. Once again we work in one dimension for simplicity. The first part of the proposition is proved by expanding each of the following with an application of DM: $\mathsf{Var}(S^2_\Delta)$, $\mathsf{Var}(g(\bar{X}))^2$, and $\kappa_4(g(\bar{X})) = \mathbb{E}[(g(\bar{X}) - \mathbb{E}[g(\bar{X})])^4] - 3\mathsf{Var}(g(\bar{X}))^2$. For the first of these we apply the multivariate version in Theorem 2 of Hurt (1976), with the function $h(\bar{X}, R) = g'(\bar{X})^2 R$, with $R = (N-1)/NS^2$ (the biased sample variance), and $q = 1$. Since $h_{1,0}(\mu, (N-1)/N\sigma^2) = 2[(N-1)/N]g'(\mu)g''(\mu)\sigma^2$, $h_{0,1}(\mu, (N-1)/N\sigma^2) = g'(\mu)^2$ and $\mathsf{Cov}(\bar{X}, S^2) = N\kappa_3/(N-1)^2$, this results in

$$\mathsf{Var}(S^2_\Delta) = \mathsf{Var}(h(\bar{X}, R))/N^2 \tag{61}$$

$$= \frac{g'(\mu)^4(2\sigma^4 + \kappa_4)}{N^3} + \frac{4g'(\mu)^2 g''(\mu)^2\sigma^6}{N^3} + \frac{4g'(\mu)^3 g''(\mu)\sigma^3\kappa_3}{N^3} + O(N^{-7/2}). \tag{62}$$

Next, we can write

(63)

$$\mathsf{Var}(g(\bar{X}))^2 = \left\{\frac{g'(\mu)\sigma^2}{N} + \frac{g'(\mu)g''(\mu)\kappa_3}{N^2} + \frac{g'(\mu)g^{(3)}(\mu)\sigma^2}{N^2} + \frac{g''(\mu)^2\sigma^4}{2N^2} + O(N^{-3})\right\}^2$$

$$= \frac{g'(\mu)^4\sigma^4}{N^2} + \frac{2g'(\mu)^3 g''(\mu)\sigma^2\kappa_3}{N^3} + \frac{2g'(\mu)^3 g^{(3)}(\mu)\sigma^6}{N^3} + \frac{g'(\mu)^2 g''(\mu)^2\sigma^6}{N^3} + O(N^{-4}). \tag{64}$$

For the last piece, we can expand the function $h(x) = (g(x) - c)^4$ with $c = \mathbb{E}g(\bar{X})$, which gives

$$\mathbb{E}[(g(\bar{X}) - \mathbb{E}g(\bar{X}))^4] = \frac{-15g'(\mu)^2 g''(\mu)^2\sigma^6}{2N^3} + \frac{18g'(\mu)^3 g''(\mu)\sigma^2\kappa_3}{N^3} + g'(\mu)^4\left[\frac{3\sigma^4}{N^2} + \frac{\kappa_4}{N^3}\right] + O(N^{-4}). \tag{65}$$

Combining these, we have after some algebra that

(66)

$$R_g(F) = \lim_{N\to\infty} N^3\left\{\mathsf{Var}(S^2_\Delta) - \frac{2\mathsf{Var}(g(\bar{X}))^2}{N} - \mathbb{E}[(g(\bar{X}) - \mathbb{E}g(\bar{X}))^4] + 3\mathsf{Var}(g(\bar{X}))^2\right\}$$

$$= \frac{29}{2} g'(\mu) g''(\mu)^2 \sigma^6 - 11 g'(\mu)^3 g''(\mu) \sigma^2 \kappa_3 + 6 g'(\mu)^3 g^{(3)}(\mu) \sigma^6. \tag{67}$$

This establishes the result for S^2_Δ. Note that the function $R_g(F)$ may look different if $p > 1$, but it will still vanish for linear functions g.

Next we treat the bootstrap case in some detail, leaving the (similar) buckets arguments to the reader. Let $\eta_N = \mathbb{E}[(N-1)/M]$ and recall that $\mathsf{Var}(\bar{X}^\star) = (1+\eta_N)\sigma^2/N$. This proof uses similar ideas to that of Proposition 1, but requires more terms in the various DM expansions. We must compute terms of the form

$$C_N^{jk} = \mathsf{Cov}\left(\left(g(\bar{X}) - g(\bar{X}_j^\star)\right)^2, \left(g(\bar{X}) - g(\bar{X}_k^\star)\right)^2 \right) \tag{68}$$

$$= \mathbb{E}\left[\left(g(\bar{X}) - g(\bar{X}_j^\star)\right)^2 \left(g(\bar{X}) - g(\bar{X}_k^\star)\right)^2 \right] - \mathbb{E}\left[\left(g(\bar{X}) - g(\bar{X}^\star)\right)^2 \right]^2 \tag{69}$$

$$= C_{N,1}^{jk} + C_{N,2}. \tag{70}$$

The square root of $C_{N,2}$ was expanded in the previous proof. A more careful analysis, using the fact that $\mathbb{E}[(\bar{X}-\mu)^3] = \kappa_3(\bar{X}) = O(N^{-2})$ along with some conditioning arguments, shows that the $O(N^{-3/2})$ term in (53) is in fact $O(N^{-2})$. Hence

$$C_{N,2} = -\frac{g'(\mu)^4 \sigma^4}{N^2} \eta_N^2 \left(1 + O(N^{-1})\right). \tag{71}$$

Likewise, for the same reason the following also holds:

$$\xi_N^4 = \mathsf{Var}(g(\bar{X}))^2 = \frac{g'(\mu)^4 \sigma^4}{N^2} \left(1 + O(N^{-1})\right). \tag{72}$$

Using another application of DM, we can write

$$C_{N,1}^{jk} = g'(\mu)^4 \left\{ \mathbb{E}[(\bar{X}-\mu)^4] - 4\mathbb{E}[(\bar{X}-\mu)^3(\bar{X}_j^\star - \mu)] \right. \tag{73}$$

$$+ 2\mathbb{E}[(\bar{X}-\mu)^2(\bar{X}_j^\star-\mu)^2] + \mathbb{E}[(\bar{X}_j^\star-\mu)^2(\bar{X}_k^\star-\mu)^2] \tag{74}$$

$$+ 4\mathbb{E}[(\bar{X}-\mu)^2(\bar{X}_j^\star-\mu)(\bar{X}_k^\star-\mu)] \tag{75}$$

$$\left. - 4\mathbb{E}[(\bar{X}-\mu)(\bar{X}_j^\star-\mu)^2(\bar{X}_k^\star-\mu)] \right\} \tag{76}$$

$$+ (\text{degree 5 terms}) + O(N^{-3}). \tag{77}$$

Each degree 4 term can be examined using the relation

$$\mathbb{E}[XYZW] = \kappa_4(X,Y,Z,W) + \mathbb{E}[XY]\mathbb{E}[ZW] + \mathbb{E}[XZ]\mathbb{E}[YW] + \mathbb{E}[XW]\mathbb{E}[YZ]. \tag{78}$$

All of the κ_4 terms in $C_{N,1}^{jk}$ can be shown (by conditioning on the bootstrap weights and using the result in Brillinger (1969) to be of order N^{-3}, and can be ignored. When $j \neq k$, it is easily seen that $\mathsf{Cov}(\bar{X}_j^\star, \bar{X}_k^\star) = \sigma^2/N$. After some algebra, we therefore have

$$C_{N,1}^{jk} = \frac{g'(\mu)^4 \sigma^4}{N^2} \left(1 + 2\delta_{jk}\right) \eta_N^2 + O(N^{-3}). \tag{79}$$

Putting this all together, we have

$$\mathsf{Var}(S^2_{\mathsf{boot}}) = \frac{1}{b^2} \sum_{j=1}^{b} \sum_{k=1}^{b} (C_{N,1}^{jk} + C_{N,2}) \tag{80}$$

$$= \frac{1}{b^2}\frac{g'(\mu)^4\sigma^4}{N^2}\left(3b\eta_N^2 + b(b-1)\eta_N^2 - b^2\eta_N^2 + b^2 O(N^{-1})\right) \tag{81}$$

$$= \frac{2g'(\mu)^4\sigma^4}{bN^2}\eta_N^2\left(1 + O(b/N)\right) \tag{82}$$

$$= \frac{2\xi_N^4}{b}\left(1 + O(N^{-1/2})\right)\left(1 + O(b/N)\right). \tag{83}$$

In particular, we have shown that the $o(1)$ relative error in this approximation to the variance of S^2_{boot} is given by $O\left(\max\{N^{-1/2}, n^{-1}\}\right)$. In the case of S^2_{buck} (not derived here) similar calculations lead to a relative error of $O\left(\max\{b/n, b^{-1}\}\right)$. The b/n term involves higher order derivatives of g, and therefore vanishes for linear functions. □

Proof of Corollary 3. The proof is by contradiction. As a shorthand, we write S_b^2 to denote the estimator S^2_{buck} built from b buckets. We can ignore the $O(bN^{-3})$ relative error term in (22) since it is of the same order as $B_N(S_b^2)^2$. Suppose that $b = o(N^{1/2})$. Then there exists a sequence β_N such that $b^{-1}\beta_N \to \infty$ and $N^{-1/2}\beta_N \to 0$. Therefore $B_N(S_b^2)^2$ and $B_N(S^2_{\beta_N})^2$ are both $o(N^{-5/2}) = o(b^{-1}N^{-2})$, whereby $\mathsf{MSE}(S^2_{\beta_N})/\mathsf{MSE}(S_b^2)$ behaves like $\mathsf{Var}(S^2_{\beta_N})/\mathsf{Var}(S_b^2) \sim b/\beta_N \to 0$. Now suppose that $bN^{-1/2} \to \infty$. In that case, the $O(bN^{-3})$ terms dominates the MSE, and therefore we can choose a sequence $\beta_N = o(b)$ such that $\mathsf{MSE}(S^2_{\beta_N})/\mathsf{MSE}(S_b^2) \sim \beta_N/b \to 0$. □

Proof of Theorem 4. We give details for part 1 (the bootstrap), and only a sketch of part 2 (buckets), which is similar.

1. (Poisson bootstrap).
   The proof is done in two parts. We first obtain a bound on the expected Euclidean distance between the multinomial bootstrap weights and the Poisson bootstrap weight. Secondly, we show how this bound carries through to Wasserstein distance between their empirical processes.
   For the first part, we employ the equivalent characterization of the Poisson bootstrap as a multinomial bootstrap where each resample is of a random size $M = \sum_i m_{ij}$, distributed as $\mathsf{Pois}(N)$. Consider the following coupling, for a single bootstrap resample. Let

$$M \sim \mathsf{Pois}(N) \tag{84}$$

$$V \sim \mathsf{Multinom}(M - N\mathbb{1}\{M > N\}, \mathbf{1}/N) \tag{85}$$

$$Y \sim \mathsf{Multinom}(N - M\mathbb{1}\{M \leq N\}, \mathbf{1}/N) \tag{86}$$

$$w^\star = (Y + V\mathbb{1}\{M \leq N\})/N \tag{87}$$

$$w^{\star\star} = (Y\mathbb{1}\{M > N\} + V)/M. \tag{88}$$

   It is readily seen by conditioning on the sign of $M - N$ that $w^\star$ and $w^{\star\star}$ have the correct marginal distributions prescribed by (26) and (27). Without loss of generality suppose that $M \leq N$, so that $V \sim \mathsf{Multinom}(M, \mathbf{1}/N)$ and $Y \sim \mathsf{Multinom}(N - M, \mathbf{1}/N)$ (the $M > N$ case is proved similarly). We have

$$\mathbb{E}\left[\|w^\star - w^{\star\star}\|_2^2 | M\right] = \mathbb{E}\left[\|Y/N + (1/M - 1/N)V\|_2^2 | M\right] \tag{89}$$

$$
\begin{aligned}
&= \mathbb{E}\left[\|(1/N)(Y - [(N-M)/M]\mathbf{1}) + \right. \tag{90}\\
&\qquad \left.(1/M - 1/N)(V - (M/N)\mathbf{1})\|_2^2 | M\right] \tag{91}\\
&= (1/N)^2 \mathrm{tr}(\mathsf{Var}(Y)) + (1/M - 1/N)^2 \mathrm{tr}(\mathsf{Var}(V)) \tag{92}\\
&= \frac{1}{N}\left(1 - \frac{1}{N}\right)\left(\frac{N}{M} - 1\right) \tag{93}\\
&\leq \frac{1}{N}\left|\frac{N}{M} - 1\right|. \tag{94}
\end{aligned}
$$

Thus $\mathbb{E}\left[\|w^\star - w^{\star\star}\|_2 | M\right] \leq N^{-1/2}(N/M - 1)^{1/2}$.
Note that the previous calculation was conditional on M, and made no use of the fact that it is Poisson. Therefore we can replace it with any nonnegative-integer-valued distribution above, in particular the Poisson distribution with mean N left-truncated at 1. By part 2 of Lemma 5, $\mathbb{E}\left[\|w^\star - w^{\star\star}\|_2\right] = \mathbb{E}\left\{\mathbb{E}\left[\|w^\star - w^{\star\star}\|_2 | M\right]\right\} = O(N^{-3/4})$.
Recall (Van der Vaart, 1998) that the $\mathcal{F}$ induces a metric $\|G_1 - G_2\|_{\mathcal{F}} = \sup_{f \in \mathcal{F}} \left(\int f\, dG_1 - \int f\, dG_2\right)$, and also that it has a square-integrable envelope function $B(x) = \sup_{f \in \mathcal{F}} |f(x)|$.
For any $H \in \mathsf{BL}_1(\mathcal{F})$, we have

$$
\left|\mathbb{E}\left[H\left(\sqrt{N}(F^\star - \hat{F})\right)\middle| \mathbf{X}\right] - \mathbb{E}\left[H\left(\sqrt{N}(F^{\star\star} - \hat{F})\right)\middle| \mathbf{X}\right]\right| \tag{95}
$$

$$
\begin{aligned}
&\leq \mathbb{E}\left[\left|H\left(\sqrt{N}(F^\star - \hat{F})\right) - H\left(\sqrt{N}(F^{\star\star} - \hat{F})\right)\right|\middle| \mathbf{X}\right] \tag{96}\\
&\leq \mathbb{E}\left[\left\|\sqrt{N}(F^\star - \hat{F}) - \sqrt{N}(F^{\star\star} - \hat{F})\right\|_{\mathcal{F}}\middle| \mathbf{X}\right] \tag{97}\\
&= \mathbb{E}\left[\left\|\sqrt{N}(F^\star - F^{\star\star})\right\|_{\mathcal{F}}\middle| \mathbf{X}\right] \tag{98}\\
&= \sqrt{N}\,\mathbb{E}\left[\|F^\star - F^{\star\star}\|_{\mathcal{F}} | \mathbf{X}\right] \tag{99}\\
&= \sqrt{N}\,\mathbb{E}\left[\sup_{f \in \mathcal{F}}\left(\int f dF^\star - \int f d\tilde{F}\right)\middle| \mathbf{X}\right] \tag{100}\\
&= \sqrt{N}\,\mathbb{E}\left[\sup_{f \in \mathcal{F}}\left(\sum w_i^\star f(\mathbf{X}_i) - \sum_i w_i^{\star\star} f(\mathbf{X}_i)\right)\middle| \mathbf{X}\right] \tag{101}\\
&\leq \sqrt{N}\,\mathbb{E}\left[\sup_{f \in \mathcal{F}}\left(\sum_i f(\mathbf{X}_i)\,|w_i^\star - w_i^{\star\star}|\right)\middle| \mathbf{X}\right] \tag{102}\\
&\leq \sqrt{N}\,\mathbb{E}\left[\sum_i B(\mathbf{X}_i)\,|w_i^\star - w_i^{\star\star}|\middle| \mathbf{X}\right] \tag{103}\\
&= \sqrt{N}\sum_i B(\mathbf{X}_i)\,\mathbb{E}\left[|w_i^\star - w_i^{\star\star}|\right] \tag{104}\\
&= \sqrt{N}\sum_i \frac{1}{N} B(\mathbf{X}_i)\,\mathbb{E}\left[\|w^\star - w^{\star\star}\|_1\right]. \tag{105}\\
&\leq \sqrt{N}\left(\frac{1}{N}\sum_i B(\mathbf{X}_i)\right)\mathbb{E}\left[\|w^\star - w^{\star\star}\|_2^2\right]^{1/2} \tag{106}
\end{aligned}
$$

$$\leq \left(\frac{1}{N} \sum_i B\left(\mathbf{X}_i\right) \right) O\left(N^{-1/4}\right). \tag{107}$$

Because $\mathbf{x}_i$ are i.i.d. $\sim F$, $(1/N)\sum_i B(\mathbf{x}_i) \to \int BdF$ almost surely, and in particular, $(1/N)\sum_i B(\mathbf{x}_i) = O_P(1)$ is bounded in probability. We have therefore shown that

$$\|\sqrt{N}(F^\star - \hat{F}) - \sqrt{N}(F^{\star\star} - \hat{F})\|_{\mathcal{W}} = O_p(N^{-1/4}). \tag{108}$$

2. (Buckets). The proof of part 2 of the theorem follows along much the same lines. The coupling of $F_N^\bullet$ and $F_N^{\bullet\bullet}$ is constructed as follows. We use S_N to denote the symmetric group of order N.

$$\rho \sim \mathsf{U}(S_N) \tag{109}$$

$$k \sim \mathsf{Binom}(N, 1/b) \tag{110}$$

$$w^\bullet = \rho(\mathbf{1}'_n, \mathbf{0}'_{N-n})/n \tag{111}$$

$$w^{\bullet\bullet} = \rho(\mathbf{1}'_k, \mathbf{0}'_{N-k})/k. \tag{112}$$

It is easy to see that under this construction,

$$\|w^\bullet - w^{\bullet\bullet}\|_2^2 = \frac{1}{n}\left|\frac{n}{k} - 1\right|, \tag{113}$$

which nicely parallels (94). We can therefore use part 4 of Lemma 5 to bound the expected Euclidean distance between the weight vectors. The rest of the proof proceeds much as before.

□