Fast Averaging

Shreeshankar Bodas, Devavrat Shah
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
{bodas, devavrat}@mit.edu

***Abstract*—We are interested in the following question: given n numbers $x_1, \dots, x_n$, what sorts of approximation of average $x_{\text{ave}} = \frac{1}{n}(x_1 + \cdots + x_n)$ can be achieved by knowing only r of these n numbers. Indeed the answer depends on the *variation* in these n numbers. As the main result, we show that if the vector of these n numbers satisfies certain regularity properties captured in the form of finiteness of their empirical moments (third or higher), then it is possible to compute approximation of x_{ave} that is within $1 \pm \varepsilon$ multiplicative factor with probability at least $1 - \delta$ by choosing, on an average, $r = r(\varepsilon, \delta, \sigma)$ of the n numbers at random with r is dependent only on ε, δ and the amount of variation σ in the vector and is independent of n.**

The task of computing average has a variety of applications such as distributed estimation and optimization, a model for reaching consensus and computing symmetric functions. We discuss implications of the result in the context of two applications: load-balancing in a computational facility running MapReduce, and fast distributed averaging.

***Index Terms*—Averaging, Probabilistic approximation.**

I. Introduction

We are interested in computing the average or the arithmetic mean of a long vector of nonnegative real numbers. Mathematically, the task could not be simpler: all we need is $n - 1$ additions and one division. This provides us with the exact answer, requires n operations and the knowledge of all the n numbers. The task of averaging has become of great interest across a variety of applications in the context of distributed optimization [1], [2], estimation [3], control and robotics [4], [5], database operations [6], peer-to-peer networks [7] and so on. On one hand, this seemingly simple problem is precisely what is required in various applications. On the other hand, it has served as the simplest non-trivial example to gather insights for complex scenarios such as social learning [8] and flocking of birds [9]. In most of the above mentioned scenarios, the interest is in understanding how to obtain a good approximate estimate of the average of the n numbers with minimal effort, usually in a distributed manner.

A. Main result

Motivated to design a fast algorithm for computing the average in a distributed manner, we study the properties of a simple randomized estimator for average of n numbers based on sampling fewer than n numbers: choose r samples at random (with replacement[1]) and use their average as surrogate for the average of the n numbers. Specifically, the goal is to understand the dependence of the approximation error (and confidence) on the (average) number of samples r.

[1]If the samples are chosen by different, distributed agents or if the algorithm operates with little memory, then sampling without replacement is not feasible.

First suppose all the n numbers are equal in value, say 5. Then for any $r \geq 1$, the above stated algorithm produces the exact estimate! At the other extreme, consider the sequence of numbers with $x_1 = C$ (for some $C > 0$) and $x_i = 0$ for $i \geq 2$. Then we require $r = \Omega(n)$ in order to sample x_1 even once and hence to estimate the mean within (constant) multiplicative error with probability close to 1.

These two extreme examples suggest that in general the number of samples needed to obtain a good estimate of the average ought to depend upon the *regularity* of the sequence of n numbers. As the main result of this paper, we obtain bound on the number of samples needed to learn the average within multiplicative error $1 \pm \varepsilon$ with probability at least $1 - \delta$ for given $\varepsilon, \delta \in (0, 1)$. To establish these results, we use the following notion of *regularity*: the sequence of n numbers must have uniformly bounded empirical third moment. If the sequence is such that it has the exponential moment (defined in Section II-A) uniformly bounded then it leads to tighter bounds on approximation error and the number of samples required. This result along with problem statement is described in Section II. Note the difference between our result and the classic results on the large deviations analysis of sampling with/without replacement [10], where the rate-function results are derived in terms of the entries of the given vector (as opposed to its statistics).

B. Related work

There has been a very long line of research, especially in recent years, to design fast distributed algorithms for computing the average of numbers in a given network represented by a graph. This line of inquiry was initiated by Tsitsiklis [1] and co-authors [11] where a linear iterative algorithm was proposed with updates constrained by the network graph. It was utilized as an important *sub-routine* for designing parallel, distributed algorithms for a class of optimization problems [2]. More recently, inspired by applications in sensor and peer-to-peer networks, robust and randomized variants of such algorithms have been studied [7], [12]. In these and other works such as [13], the algorithm's running time was tied with topological property of the *communication* network graph. Various topology-specific improvements have been proposed to improve the running time, e.g. use of geometry for graphs like Geometric random graphs [14], or use of non-reversible random walks for general graph [15]. The work by Mosk-

Aoyama and Shah [16] provides a simple algorithm with seemingly optimal dependence on the graph topology (in the form of inverse conductance). Information theoretic approach has been developed in [17] to capture such a qualitative aspect in the lower bound. A more direct lower bound for linear (or Lipschitz) iterative algorithms has been developed for ring(like) graphs [18].

In all of the above work, the primary goal has been to understand the effect of network structure on computing the average of an arbitrary sequences of n numbers. Here the focus is to capture the effect of regularity of the sequence of n numbers on the amount of information needed (the number of samples required) to learn a good estimate of the average.

C. Application

While the viewpoint of this work differs from the literature as explained, it does help in designing better distributed averaging algorithms. In particular, we describe the natural adaptation for computing average of numbers in a distributed manner in the network. We relate the computation time with the graph structure using the main result mentioned earlier. This is discussed in Section III-B.

The task of averaging is computationally equivalent to computing a symmetric function. The recently emerging computational framework for large data centers, the MapReduce introduced by researchers at Google, Inc. [19] is precisely attempting to compute symmetric functions. In this sense, averaging provides a convenient abstract model to study various aspects of system design for the data centers utilizing MapReduce. Specifically, we study the question of load-balancing on the commodity servers in the context of task assignment in MapReduce. In Section III-A, we provide a simple randomized algorithm for this, based on the main result.

II. Main result

In this section, we establish Theorem 1 as the main result of the paper. This result provides an upper-bound on the number of samples required to get an (ε, δ)-approximation for the arithmetic mean of a given vector.

A. Problem statement

For the purpose of the empirical mean calculation, we assume that $\{x_j\}_{j=1}^n$ are the first n numbers of a fixed sequence $\{x_j\}_{j=1}^\infty$. Given the n non-negative real numbers $x_1, \dots, x_n$, we wish to compute their average or arithmetic mean $x_{\text{ave}} = (1/n)\sum_{j=1}^n x_j$. We assume that the sequence satisfies the following regularity conditions:

[C1.] *Bounded mean, variance:* There exist real numbers $0 < \nu < \mu$ and $\sigma > 0$ so that for all n, $\nu \le x_{\text{ave}} \le \mu$, and $\frac{1}{n}\sum_{j=1}^n (x_j - x_{\text{ave}})^2 \le \sigma^2$.
The mean and variance of the given n numbers depend upon the parameter n. We interchangeably use the notation $\mu_n := x_{\text{ave}}$ and $\sigma_n := \left(\frac{1}{n}\sum_{j=1}^n (x_j - x_{\text{ave}})^2\right)^{1/2}$.

[C2a.] *Bounded k^{th} moment:* There exist an integer $k \ge 3$ and $\alpha > 0$ such that for all n, $\left(\frac{1}{n}\sum\limits_{j=1}^n x_j^k\right)^{1/k} \le \alpha$.

[C2b.] *Bounded exponential moment:* There exist $\rho > 0$, $\beta > 0$ so that for all n, $\dfrac{1}{n}\sum\limits_{j=1}^n x_j^3 e^{\rho x_j} \le \beta$.

Note: the condition [C2b] implies [C2a] for all $k < \infty$.

B. Proposed Algorithm and its properties

Algorithm: A given entry x_i in the vector is sampled Y_i times, where $Y_i \sim B(r, 1/n)$ is a binomial random variable. All the Y_is are mutually independent. Declare $\hat{Y} = \frac{\sum_j x_j Y_j}{\sum_j Y_j}$ as the estimate of x_{ave}. $\diamond$

Note that the average number of samples under the algorithm is r, as the vector has n entries. We now analyze the approximation error in the estimator $\hat{Y}$.

Theorem 1 (Probability of Error): Given a sequence of n numbers $x_1, \dots, x_n$ satisfying regularity conditions [C1]-[C2a] or [C1]-[C2b], let $\hat{Y}$ be the estimation of $x_{\text{ave}} = \frac{1}{n}\left(\sum_{i=1}^n x_i\right)$ under the sampling algorithm as described above. Then for any given $\varepsilon > 0$ small enough and $\delta > 0$,

$$\mathbb{P}\left(\left|\frac{x_{\text{ave}} - \hat{Y}}{x_{\text{ave}}}\right| \ge \varepsilon\right) \le \delta, \tag{1}$$

as long as

1. $r \ge \left(\frac{1}{\varepsilon^2}\log\frac{2}{\delta}\right) \cdot \frac{20(\mu^2+\sigma^2)^2}{\nu^4}$ under [C1]-[C2a].
2. $r \ge \left(\frac{1}{\varepsilon^2}\log\frac{2}{\delta}\right) \cdot \frac{20(\mu^2+\sigma^2)}{\nu^2}$ under [C1]-[C2b].

Proof: Due to space limitations, we only prove the result assuming the conditions [C1]-[C2a] (the first part). The proof of the second part is similar, the only difference is the constant terms. The proof is based on the Chernoff bound and the union bound. The main idea is to show that $(1/n)\sum_j x_j Y_j \to r\mu_n/n$ and $(1/n)\sum_j Y_j \to r/n$, so that their ratio converges to μ_n. In the process, we obtain the respective rates of convergence. The proof proceeds according to the following steps:

Step 1: Let $W_n := \frac{1}{n}\sum_{j=1}^n Y_j$. Then given $\epsilon \in (0,1)$ there exists $\delta > 0$ such that for all n,

$$\mathbb{P}\left(|W_n - r/n| \ge \epsilon r/n\right) \le e^{-n\delta}.$$

Further, the choice

$$\delta = (r/n)\min\left((1-\epsilon)\log(1-\epsilon) + \epsilon, (1+\epsilon)\log(1+\epsilon) - \epsilon\right)$$

satisfies the stated bound.

Proof: (of Step 1) Fix any $\epsilon \in (0,1)$. By the Chernoff bound, $\mathbb{P}\left(|W_n - c| \ge \epsilon c\right) \le \exp\left(-\inf\limits_{|t-c|\ge\epsilon c}\Lambda^*_{W_n}(t)\right)$, where

$$\Lambda^*_{W_n}(y) = \sup_{\theta\in\Re}\left(y\theta - \Lambda_{W_n}(\theta)\right)$$

and (with $p = 1/n$)

$$\begin{aligned}\Lambda_{W_n}(\theta) &= \log\mathbb{E}(e^{\theta W_n}) &&= \log\left(1 + p(e^{\theta/n} - 1)\right)^{rn} \\ &\overset{(a)}{\le} \log\left(e^{p(e^{\theta/n}-1)}\right)^{rn} &&= r(e^{\theta/n} - 1),\end{aligned}$$

where (a) holds because $e^x \ge 1 + x$ for all $x \in \Re$. Thus,

$$\Lambda^*_{W_n}(y) \ge \sup_{\theta\in\Re}\left(y\theta - r(e^{\theta/n} - 1)\right) = n\left(y\log\frac{yn}{r} - y + \frac{r}{n}\right),$$

after some calculus. The function $f(y) = y\log(yn/r) - y + r/n$ is nonnegative and convex for $y > 0$, and has the minimum at $y = r/n$ with $f(r/n) = 0$. It follows that for all $\epsilon \in (0,1)$, $f((r/n)(1 \pm \epsilon)) > 0$. Therefore,

$$\mathbb{P}\left(|W_n - r/n| \geq \epsilon r/n\right) \leq e^{[-r\min(f((r/n)(1-\epsilon)),f((r/n)(1+\epsilon)))]},$$

completing the proof of Step 1. ■

Step 2: Let $V_n := \frac{1}{n}\sum_{j=1}^n x_j Y_j$. Then there exists $\epsilon_0 > 0$ such that for all $\epsilon \in (0, \epsilon_0)$ there exists $\delta > 0$ such that

$$\mathbb{P}\left(|V_n - r\mu_n/n| \geq \epsilon\right) \leq e^{-n\delta}.$$

Proof: (of Step 2) Fix any $\epsilon \in (0, r\nu/n)$. By the Chernoff bound,

$$\mathbb{P}\left(|V_n - r\mu_n/n| \geq \epsilon\right) \leq \exp\left(-\inf_{|t-r\mu_n/n|\geq\epsilon}\Lambda^*_{V_n}(t)\right),$$

where
$$\Lambda^*_{V_n}(y) = \sup_{\theta\in\Re}\left(y\theta - \Lambda_{V_n}(\theta)\right)$$

and
$$\Lambda_{V_n}(\theta) = \log\mathbb{E}\left(e^{\theta V_n}\right) \leq \frac{r}{n}\sum_{j=1}^n (e^{\theta x_j/n} - 1),$$

after some calculations and using (as before) $e^x \geq 1 + x$ for all $x \in \Re$. Define $f(\theta) := y\theta - \frac{r}{n}\sum_{j=1}^n e^{\theta x_j/n} + r$. Then

$$\Lambda^*_{V_n}(y) \geq \sup_{\theta\in\Re} f(\theta) \geq f(\theta_0), \quad \text{for any } \theta_0 \in \Re.$$

Choosing $\theta_0 = n^2(y - r\mu_n/n)/(r(\mu_n^2 + \sigma_n^2))$, we get

$$f(\theta_0) = h(y) := \frac{n^2 y(y - r\mu_n/n)}{r(\mu_n^2 + \sigma_n^2)} - \frac{r}{n}\sum_{j=1}^n e^{\frac{x_j(y-r\mu_n/n)}{r(\mu_n^2+\sigma_n^2)/n}} + r.$$

We have $h(r\mu_n/n) = 0, h'(r\mu_n/n) = 0$, and

$$h''(r\mu_n/n) = \frac{n}{r(\mu_n^2 + \sigma_n^2)/n} \geq \frac{n}{r(\mu^2 + \sigma^2)/n} > 0.$$

Therefore (using Taylor's Theorem), there exists $\epsilon_0 > 0$ such that for all $\epsilon \in (0, \epsilon_0)$ there exists $\delta > 0$ such that $\Lambda^*_{V_n}(r\mu_n/n \pm \epsilon) > n\delta$, completing the proof of Step 2. ■

Step 3: Define $\tau := (\mu^2 + \sigma^2)/(\alpha\nu), \gamma := r\nu^2/(2n\alpha^2)$. Then for all $a, \delta > 0$ there exists an integer n_0 such that for all $n \geq n_0$,

$$\mathbb{P}\Big(\Big|\frac{1}{n}\sum_{j=1}^n x_j Y_j - \frac{r\mu_n}{n}\Big| \geq \frac{r\mu_n a\tau}{n\sqrt[k]{n}}\Big) \leq e^{-(a^2\gamma-\delta)n^{1-2/k}}.$$

Proof: (of Step 3:) We only prove the claim for the case $a = 1$, because the more general case is an immediate consequence. We have $\tau \geq \dfrac{\mu_n^2 + \sigma_n^2}{\alpha\mu_n}$, thus $\dfrac{r\mu_n\tau}{n\sqrt[k]{n}} \geq \dfrac{r(\mu_n^2 + \sigma_n^2)}{n\sqrt[k]{n}\alpha}$, implying

$$\begin{aligned}
&\mathbb{P}\left(\left|\frac{1}{n}\sum_{j=1}^n x_j Y_j - \frac{r\mu_n}{n}\right| \geq \frac{r\mu_n\tau}{n\sqrt[k]{n}}\right) \\
&\quad\leq \mathbb{P}\left(\left|\frac{1}{n}\sum_{j=1}^n x_j Y_j - \frac{r\mu_n}{n}\right| \geq \frac{r(\mu_n^2 + \sigma_n^2)}{n\sqrt[k]{n}\alpha}\right). \qquad (2)
\end{aligned}$$

Define $\epsilon_n = r(\mu_n^2 + \sigma_n^2)/(n\sqrt[k]{n}\alpha) \leq r(\mu^2 + \sigma^2)/(n\sqrt[k]{n}\alpha)$ and consider $y = r\mu_n/n + \epsilon_n$ and fix an integer $w \leq k$. From the proof of Step 2, we have $\Lambda^*_{V_n}(y) \geq h(y)$ for all y and our objective now is to lower-bound the value of $h(y)$. For any $\zeta \in [0,1]$, we get

$$\begin{aligned}
|h^{(w)}(r\mu_n/n + \zeta\epsilon_n)| &= n\left[\frac{r}{n^2}\sum_{j=1}^n \frac{x_j^w e^{\frac{x_j\zeta\epsilon_n}{r(\mu_n^2+\sigma_n^2)/n}}}{r^w(\mu_n^2 + \sigma_n^2)^w/n^w}\right] \\
&\overset{(a)}{\leq} n\left[\frac{r}{n^2}\sum_{j=1}^n \frac{x_j^w e^{\zeta}}{r^w(\mu_n^2 + \sigma_n^2)^w/n^w}\right] \\
&\overset{(b)}{\leq} \frac{r\alpha^w e}{r^w(\mu_n^2 + \sigma_n^2)^w/n^w}, \qquad (3)
\end{aligned}$$

where the inequality (a) holds because $x_j \leq \alpha\sqrt[k]{n}$ for all $1 \leq j \leq n$ for all n, and the inequality (b) holds because for all $1 \leq w \leq k$, $\sqrt[w]{\frac{1}{n}\sum\limits_{j=1}^n x_j^w} \leq \sqrt[k]{\frac{1}{n}\sum\limits_{j=1}^n x_j^k} \leq \alpha$. Using Taylor's theorem and with $\gamma_n := r\mu_n/n$, we get

$$h(\gamma_n + \epsilon_n) = h(\gamma_n) + \epsilon_n h'(\gamma_n) + \frac{\epsilon_n^2 h''(\gamma_n)}{2!} + \frac{\epsilon_n^3 h^{(3)}(\xi)}{3!}$$

for some $\xi \in [r\mu_n/n, r\mu_n/n + \epsilon_n]$. Using the bound (3) on the absolute value of $h^{(w)}(r\mu_n/n + \zeta\epsilon_n)$, we see that

$$h(r\mu_n/n + \epsilon_n) = \gamma n^{1-2/k} + o(n^{1-2/k})$$

for some constant γ with $0 < \gamma = \dfrac{r\nu^2}{2n\alpha^2} \leq \dfrac{r(\mu_n^2 + \sigma_n^2)}{2n\alpha^2}$. Since $\Lambda^*_{V_n}(y) \geq h(y)$ for all y (from the proof of Step 2),

$$\Lambda^*_{V_n}(r\mu_n/n + \epsilon_n) \gtrsim \gamma n^{1-2/k}.$$

More formally, for all $\delta > 0$ there exists n_1 large enough such that for all $n \geq n_1$,

$$\Lambda^*_{V_n}(r\mu_n/n + \epsilon_n) \geq (\gamma - \delta)n^{1-2/k}.$$

Using essentially the same argument, we can show that

$$h(r\mu_n/n - \epsilon_n) = \gamma n^{1-2/k} + o(n^{1-2/k}),$$

and for all $\delta > 0$ there exists n_2 large enough such that for all $n \geq n_2$, $\Lambda^*_{V_n}(r\mu_n/n - \epsilon_n) \geq (\gamma - \delta)n^{1-2/k}$.

Thus, the choice $\tau = (\mu^2 + \sigma^2)/(\alpha\nu)$ and $\gamma = r\nu^2/(2n\alpha^2)$, along with an application of the Chernoff bound, gives us (for $n \geq \max(n_1, n_2)$)

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{j=1}^n x_j Y_j - r\mu_n/n\right| \geq \frac{r\mu_n a\tau}{n\sqrt[k]{n}}\right) \leq e^{-(a^2\gamma-\delta)n^{1-2/k}},$$

completing the proof of Step 3. ■

Step 4: For $\gamma = r\nu^2/(2n\alpha^2), \tau = (\mu^2 + \sigma^2)/(\alpha\nu)$ and any $a, \delta > 0$, for n large enough,

$$\begin{aligned}
\mathbb{P}\left(\frac{\sum_{j=1}^n x_j Y_j}{\sum_{j=1}^n Y_j} \notin \left(\mu_n\frac{1 - \frac{a\tau}{\sqrt[k]{n}}}{1 + \frac{a\tau}{\sqrt[k]{n}}}, \mu_n\frac{1 + \frac{a\tau}{\sqrt[k]{n}}}{1 - \frac{a\tau}{\sqrt[k]{n}}}\right)\right) \\
\leq 2\exp\left[-(a^2\gamma - \delta)n^{1-2/k}\right].
\end{aligned}$$

Proof: Fix $\epsilon = \epsilon_n = a\tau/\sqrt[k]{n}$ for some constant $a > 0$. Step 1 implies that for some $\delta > 0$ and for all n,

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{j=1}^n Y_j - \frac{r}{n}\right| \geq \epsilon r/n\right) \leq e^{-n\delta},$$

with

$$\delta(\epsilon) = \frac{r}{n}\min\left((1-\epsilon)\log(1-\epsilon)+\epsilon, (1+\epsilon)\log(1+\epsilon)-\epsilon\right).$$

For n large (i.e., $\epsilon = \epsilon_n$ small), $\delta \approx r\epsilon^2/(2n)$ using the approximation $\log(1+x) \approx x$. More formally, for any $\theta > 0$ there exists n_0 such that for all $n \geq n_0$,

$$\delta(\epsilon) \geq \frac{(1-\theta)ra^2\tau^2}{2n^{2/k}}.$$

From Step 3, for all $\delta > 0$ and all n large enough,

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{j=1}^{n} x_j Y_j - r\mu_n/n\right| \geq \frac{r\mu_n a\tau}{n\sqrt[k]{n}}\right) \leq e^{-(a^2\gamma-\delta)n^{1-2/k}}.$$

If $\frac{1}{n}\sum_{j=1}^{n} x_j Y_j \in ((r\mu_n/n)(1-\epsilon), (r\mu_n/n)(1+\epsilon))$ and $\frac{1}{n}\sum_{j=1}^{n} Y_j \in ((r/n)(1-\epsilon), (r/n)(1+\epsilon))$, then (deterministically)

$$\frac{\sum_{j=1}^{n} x_j Y_j}{\sum_{j=1}^{n} Y_j} \in \left(\mu_n\frac{1-\epsilon}{1+\epsilon}, \mu_n\frac{1+\epsilon}{1-\epsilon}\right).$$

Therefore, by the union bound,

$$\begin{aligned}
&\mathbb{P}\left(\frac{\sum_{j=1}^{n} x_j Y_j}{\sum_{j=1}^{n} Y_j} \notin \left(\mu_n\frac{1-\epsilon}{1+\epsilon}, \mu_n\frac{1+\epsilon}{1-\epsilon}\right)\right)\\
\leq\ & \mathbb{P}\left(\frac{1}{n}\sum_{j=1}^{n} x_j Y_j \notin ((r\mu_n/n)(1-\epsilon), (r\mu_n/n)(1+\epsilon))\right)\\
&+ \mathbb{P}\left(\frac{1}{n}\sum_{j=1}^{n} Y_j \notin ((r\mu_n/n)(1-\epsilon), (r\mu_n/n)(1+\epsilon))\right)\\
\leq\ & e^{-(1-\theta)((r/n)a^2\tau^2/2)n^{1-2/k}} + e^{-(a^2\gamma-\delta)n^{1-2/k}}\\
\leq\ & 2e^{-(a^2\gamma-\delta)n^{1-2/k}}
\end{aligned}$$

for n large enough, since $r\tau^2 \geq 2n\gamma$. This completes the proof of Step 4. ■

Step 5: $\left(\frac{1-\varepsilon}{1+\varepsilon}, \frac{1+\varepsilon}{1-\varepsilon}\right) \subseteq (1-3\varepsilon, 1+3\varepsilon)$ for $\varepsilon \leq 1/3$, implying (after simple manipulations), for all $\delta > 0$ and all n large enough, that

$$\mathbb{P}\left(\left|(x_{\text{ave}} - \hat{Y})/x_{\text{ave}}\right| > \varepsilon\right) \leq 2e^{-\frac{r\varepsilon^2\nu^4}{20(\mu^2+\sigma^2)^2}}.$$

Thus for given $\delta > 0, \varepsilon > 0$, it is enough to take $r = \left\lceil\left(\frac{1}{\varepsilon^2}\log\frac{2}{\delta}\right)\cdot\frac{20(\mu^2+\sigma^2)^2}{\nu^4}\right\rceil$ samples of the given vector $\{x_j\}_{j=1}^{n}$ (as specified by the algorithm) to estimate the arithmetic mean x_{ave} within the stated error bounds. This completes the proof of Theorem 1. ■

III. Applications

A. MapReduce

As a first application, we look at the problem of allocating servers to jobs in a large data center. The data centers, consisting of hundreds of interconnected servers and disks, have emerged as canonical solutions for processing large volumes of data in real time, and are used by internet content and services providers such as Google, Microsoft, Amazon, Facebook, etc. Many of the computational tasks that are performed in a data center (e.g., searching for a string of text in a collection of files, counting URL access frequency, etc.) are simple mathematical operations, but the sheer volume of data makes the problem nontrivial. MapReduce [20], [19], patented by Google, Inc. is a computational framework (and its implementation) that is de facto architecture for these data center applications.

In MapReduce, there are two phases of computation: the *Map* phase and the *Reduce* phase, where the Map and the Reduce function are user-defined. The Map phase consists of the individual servers processing large volumes of data in parallel, producing intermediate outputs. This is the computationally intensive phase. The Reduce phase combines the intermediate outputs to produce the final output. The Reduce function processes a list of the intermediate outputs and is insensitive to the order in which the inputs are presented, and also to which server processed which part of the data. Thus, the Reduce function depends only on the *histogram* of the input (i.e., the number of times each of the input values is present in the list). If the application allows for an approximate answer and a small probability of error (which includes many applications of interest, like a web search query), then Theorem 1 can be used to design an algorithm for allocating the different computational jobs to the individual servers in the Map phase.

The problem: Suppose there are m servers and n files in the system. When a server processes a file F_j for a given job, it produces a number x_j that is specific to the job. The desired output is $(1/n)\sum_i x_i$.

Server allocation algorithm: Each of the servers S_i selects, at random, with a pre-determined probability, a file F_j and processes it. It reports to the central server the following two numbers: (1) The sum of the x_js of the processed files, a_i, and (2) the number of files processed, b_i. The central server (implementing the Reduce function) outputs $(\sum_i a_i)/(\sum_i b_i)$ as the estimate of the mean of x_is. ◇

Some salient features of the algorithm are: (a) robust against node failure due to the randomized nature, (b) totally distributed server allocation, (c) scales well with the system size as it does not require any global knowledge related to system state, and (d) minimal communication between the central server and the other servers.

Performance: The probability of error v/s number of samples trade-off (Theorem 1) applies here, if the vector of x_is obeys the regularity properties. We have analyzed the per-query delay under the algorithm: what is the probability that a typical query takes more than T units of processing time, in terms of the server speeds and the system loading. This probability is $\mathcal{O}(Tn^2\exp(-(n/m)Tf(c,c_0))$, where $f(c,c_0)$ is a simple function of the server loading and the server speed. Due to space limitations, a precise statement of the result and its proof is omitted. Please see [21] for more details.

B. Consensus over Graphs

The problem: We are given a connected, undirected network graph $G = (V, E)$ with n vertices represented as $V = \{1, \ldots, n\}$ and edges $E \subset V \times V$. Each node $i \in V$ has value x_i and the goal is to compute the average of these numbers x_{ave} at all nodes.

In the prior works, various algorithms have been proposed that try to utilize a graph G-conformant, irreducible non-negative valued doubly stochastic matrix $P \in [0,1]^{n\times n}$. The linear iterative algorithm and its variants that has been studied since [1] takes roughly $T_{\text{mix}}(\varepsilon)$ iterations to compute an estimate of x_{ave} that is within $1\pm\varepsilon$ multiplicative error. Here by $T_{\text{mix}}(\varepsilon)$, we mean the ε-mixing time of the probability matrix P which is characterized by the spectral gap of P (see [22] for example, for details). The total number of operations performed by such algorithms scale proportional to the number of edges in the G. Even the randomized or Gossip variant of such algorithm [3] requires $\Omega(n)$ operations per iteration. Therefore, total amount of computation performed is at least $\Omega\big(nT_{\text{mix}}(\varepsilon)\big)$.

The non-reversible Markov chains based on *lifting* of the graph proposed by Jung et al [15] leads to ε-mixing time that scales as $\Theta\big(D\log 1/\varepsilon\big)$ where D is the diameter of the graph. Combined with the above calculation, this leads to a lower bound of $\Omega\big(nD\log 1/\varepsilon\big)$ on total computation performed by best of such an algorithm.

Theorem 1 suggests that if we can sample $r = r(\varepsilon,\delta)$ of the values uniformly at random then if x_is satisfy the regular conditions [C1]-[C2b], then we can obtain desired estimation of average x_{ave}.

Consensus algorithm: Generate a binomial random variable $R \sim B(rn, 1/n)$, where r is specified by Theorem 1. Imagine a token taking a random walk over the graph G as specified by the transition probability matrix P with a uniform stationary distribution (i.e., a doubly stochastic matrix). Take R samples of the x_is, one at the end of each $T_{\text{mix}}(\varepsilon)$ time interval (total time required = $RT_{\text{mix}}(\varepsilon)$). Compute the arithmetic mean of the collected samples and broadcast in the network. $\diamond$

Analysis: Since the random walk takes $\mathcal{O}(1)$ operations per time step (we are assuming selection of random choice as per P at each step is $\mathcal{O}(1)$ operation), the total computation performed is (on an average) $\mathcal{O}(rT_{\text{mix}}(\varepsilon))$. To spread the estimate among all nodes, it could perform broadcast which would require addition at most $2|E|$ information exchanges. Thus, in summary such an algorithm would achieve $1\pm\mathcal{O}(\varepsilon)$ estimation with total of amount of computation/communication scaling as $\mathcal{O}\big(|E| + rT_{\text{mix}}(\varepsilon)\big)$. For constant-degree expanders graph, this is $\mathcal{O}(n + r\log n/\varepsilon)$; for ring graph (with non-reversible random walk) it is $\mathcal{O}\big(n + rn\log 1/\varepsilon\big)$. For example, if the x_is have a uniformly bounded exponential moment, then the computation/communication scaling for the ring graph is $\mathcal{O}(n(1/\varepsilon^2)\log(2/\delta)\log(1/\varepsilon))$, which is better than the earlier-known bounds in terms of their dependence on n or D.

IV. Conclusion

We investigated the problem of computing the arithmetic mean of a given (long) vector of real numbers with as few samples as possible. We showed that the regularity properties of the sequence (namely, the existence of uniformly bounded moments) plays a crucial role in determining the number of required samples. We presented a simple randomized algorithm for mean computation that is essentially order-optimal as long as we only allow for a "reasonable" amount of randomness. We showed how the result is useful in designing server allocation algorithms in a large server farm, and also for consensus over graphs.

Acknowledgments

The authors would like to acknowledge the support of DARPA ITMANET project.

References

[1] J. N. Tsitsiklis, "Problems in decentralized decision making and computation," Ph.D. dissertation, Massachusetts Institute of Technology, November 1984.

[2] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[3] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE/ACM Trans. Netw.*, vol. 14, no. SI, pp. 2508–2530, 2006.

[4] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Joint 44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC'05)*, December 2005.

[5] A. Jadbabaie, J. Lin, and A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[6] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "Tag: a tiny aggregation service for ad-hoc sensor networks," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 131–146, 2002.

[7] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *Proc. Ann. IEEE Symposium on Foundations of Computer Science*, Oct. 2003.

[8] B. Golub and M. Jackson, "Naive learning in social networks: Convergence, influence and the wisdom of crowds," *American Economic Journal: Microecnomics*, vol. 2, no. 1, pp. 112 – 149, Feb. 2010.

[9] M. Nagy, Z. Akos, D. Biro, and T. Vicsek, "Hierarchical group dynamics in pigeon flocks," *Nature*, vol. 464, pp. 890 – 893, Apr. 2010.

[10] A. Dembo and O. Zeitouni, *Large Deviations Techniques and Applications*, 2nd ed. Springer-Verlag New York, Inc., 1998.

[11] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803 – 812, Sep. 1986.

[12] S. Samar, S. Boyd, and D. Gorinevsky, "Distributed estimation via dual decomposition," in *European Control Conference*, Jul. 2007.

[13] A. Tahbaz-Salehi and A. Jadbabaie, "Consensus over ergodic stationary graph processes," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 225 – 230, Jan. 2010.

[14] F. Benezit, A. G. Dimakis, P. Thiran, and M. Vetterli, "Order-optimal consensus through randomized path averaging," *IEEE Trans. Inform. Theory*, vol. 56, no. 10, pp. 5150 – 5167, Oct. 2010.

[15] K. Jung, D. Shah, and J. Shin, "Distributed averaging via lifted markov chains," *IEEE Trans. Inform. Theory*, vol. 56, no. 1, pp. 634–647, Jan. 2010.

[16] D. Mosk-Aoyama and D. Shah, "Fast distributed algorithms for computing separable functions," *IEEE Trans. Inform. Theory*, vol. 54, no. 7, pp. 2997 – 3007, Jul. 2008.

[17] O. Ayaso, D. Shah, and M. Dahleh, "Information theoretic bounds for distributed computation over networks of point-to-point channels," *IEEE Trans. Inform. Theory*, vol. 56, no. 12, pp. 6020 – 6039, Dec. 2010.

[18] A. Olshevsky, "Efficient information aggregation strategies for distributed control and signal processing," Ph.D. dissertation, Massachusetts Institute of Technology, Sep. 2010.

[19] J. Dean and S. Ghemawat, "System and method for efficient large-scale data processing," Jan. 2010.

[20] ——, "Mapreduce: Simplified Data Processing on Large Clusters," in *Proc. Operating Systems Design and Implementation*, Dec. 2004.

[21] S. Bodas and D. Shah, "Fast averaging: Tech report," http://www.mit.edu/~bodas/fast_avg_tech_report.pdf, 2011.

[22] R. Montenegro and P. Tetali, "Mathematical aspects of mixing times in markov chains," *Found. Trends Theor. Comput. Sci.*, vol. 1, no. 3, pp. 237–354, 2006.