Counting Bits for Distributed Function Computation

Ola Ayaso, Devavrat Shah, and Munther A. Dahleh
Laboratory for Information and Decision Systems
MIT
Cambridge, MA
Email: ayaso@mit.edu

***Abstract*— We consider a network of nodes, each having an initial value or measurement, and seeking to acquire an estimate of a given function of all the nodes' values in the network. Each node may exchange with its neighbors a finite number of bits every time communication is initiated. In this paper, we present an algorithm for computation of separable functions, under the constraint that communicated messages are quantized, so that with some specified probability, all nodes have an estimate of the function value within a desired interval of accuracy. We derive an upper bound on the computation time needed to achieve this goal, and show that the dependence of the computation time on the network topology, via the "conductance" of the graph representing this topology, matches a lower bound derived from Information Theoretic analysis. Hence, the algorithm's running time is optimal with respect to dependence on the graph structure.**

I. Introduction

We consider a network of nodes, each having an initial value or measurement, and seeking to acquire an estimate of a given function of all the nodes' values in the network. Each node may exchange with its neighbors a finite number of bits every time communication is initiated. In this paper, we present an algorithm for computation of separable functions, under the constraint that communicated messages are quantized, so that with some specified probability, all nodes have an estimate of the function value within a desired interval of accuracy.

Our algorithm is based on a distributed algorithm, for computing separable functions, in [3]. It is a simple randomized algorithm that is based on each node generating an exponentially distributed random variable with mean equal to the reciprocal of the node's initial value. The nodes sample from their respective distributions and make use of an information spreading algorithm to make computations and ultimately obtain an estimate of the desired function.

The advantage of this algorithm is that it is completely distributed. Nodes need not keep track of the identity of the nodes from which received information originates. Furthermore, the algorithm is not sensitive to the order in which information is received. In terms of its performance, the algorithm's computation time is almost optimal in its dependence on the network topology, as the computation time scales inversely with conductance of the graph representing the communication topology. For a large class of graphs, conductance grows like $O(1/\text{diameter})$. The drawback of the algorithm in [3], however, is that it requires nodes to exchange real numbers. As such, the algorithm is not practically implementable.

In this paper, we describe how the algorithm of [3] can be applied to the scenario where nodes can only exchange a finite number of bits when they communicate. This involves truncating the exponential distributions generated at the nodes and quantizing the messages that the nodes communicate. The analysis finds the precise scaling of the algorithm's computation time in terms of the number of bits required to maintain the performance guarantees of the original algorithm, specifically the guarantees on the probability of error in the nodes' estimates of the desired function. We find that the effect of our modification of the algorithm of [3] is to slow it down by $\log n$.

Thus, the contribution of this paper includes the non-trivial quantized implementation of the algorithm of [3] and its analysis. As a consequence, we obtain the fastest, in terms of dependence on network topology, quantized distributed algorithm for separable function computation.

II. Problem Formulation

Let an arbitrary connected network of n nodes be represented by the undirected graph $G = (V, E)$. The nodes are arbitrarily enumerated and are the vertices of the graph, $V = \{1, \ldots, n\}$; the enumeration is for the purpose of analysis only as the computation algorithm does not depend on the identities of the nodes. If nodes i and j communicate with each other, then the edge (i, j) belongs to the set E.

Each node i has a measurement or initial value $x_i \in \mathbb{R}$. We let the vector x represent all the initial values in the network, $x = (x_1 \ \ldots \ x_n)$. The goal of the nodes is to each acquire an estimate of a given function,f, of all the initial values. In this paper, the function f is separable, defined as follows. Here, 2^V denotes the power set of V.

Definition II.1. $f : \mathbb{R}^n \times 2^V \to \mathbb{R}$ is separable if there exist functions $f_1, \ldots, f_n$ such that for all $S \subseteq V$,

$$f(x, S) = \sum_{i \in S} f_i(x_i).$$

Furthermore, in this paper we assume $f \in \mathcal{F}$ where $\mathcal{F}$ is the class of all separable functions with $f_i(x_i) \geq 1$ for all $x_i \in \mathbb{R}$ and $i = 1, \ldots, n$.

The performance of an algorithm, $\mathcal{C}$, used by the nodes to compute an estimate of $f(x, V)$ at each node, is measured by the algorithm's (ϵ, δ)-computation time, $T_{\mathcal{C}}^{\text{cmp}}(\epsilon, \delta)$. It is the time until the estimates at all nodes are within a factor

978-1-4244-2571-6/08/$25.00 ©2008 IEEE

of $1 \pm \epsilon$ of $f(x, V)$, with probability larger than $1 - \delta$. The definition follows, where $\hat{y}_i(t)$ denotes the estimate of $f(x, V)$ at node i at time t.

Definition II.2. For $\epsilon > 0$ and $\delta \in (0, 1)$, the (ϵ, δ)-computing time of an algorithm, $\mathcal{C}$, denoted as $T_{\mathcal{C}}^{\text{cmp}}(\epsilon, \delta)$ is defined as

$$T_{\mathcal{C}}^{\text{cmp}}(\epsilon, \delta) = \sup_{f \in \mathcal{F}} \sup_{x \in \mathbb{R}^n} \inf \{t : \mathbf{P}(\cup_{i=1}^n \{\hat{y}_i(t) \notin [(1-\epsilon)f(x,V), (1+\epsilon)f(x,V)]\}) \leq \delta\}.$$

The algorithm described here depends on the nodes' use of an information spreading algorithm, $\mathcal{D}$, as a subroutine to communicate to each other their messages. The performance of this algorithm is captured by the δ-information-spreading time, $T_{\mathcal{D}}^{\text{spr}}(\delta)$, at which with probability larger than $1 - \delta$ all nodes have all messages. More formally, let $S_i(t)$ is the set of nodes that have node i's message at time t, and V is the set of nodes, the definition of $T_{\mathcal{D}}^{\text{spr}}(\delta)$ is the following.

Definition II.3. For a given $\delta \in (0, 1)$, the δ-information-spreading time, of the algorithm $\mathcal{D}$, $T_{\mathcal{D}}^{\text{spr}}(\delta)$, is

$$T_{\mathcal{D}}^{\text{spr}}(\delta) = \inf\{t : \mathbf{P}(\cup_{i=1}^n \{S_i(t) \neq V\}) \leq \delta\}.$$

A. Main Result

Consider a model where each node may contact one of its neighbors once in each time slot. If the edge (i, j) belongs to E, node i sends its messages to node j with probability p_{ij} and with probability p_{ii} sends its messages to no other nodes; if $(i, j) \notin E$, $p_{ij} = 0$. So, the matrix $P = [p_{ij}]$ is a stochastic matrix that describes the information spreading algorithm. The information spreading time if this algorithm is derived in terms of the "conductance" of P.

Definition II.4. For a stochastic matrix P, the conductance of P, denoted $\Phi(P)$, is

$$\Phi(P) = \min_{\substack{S \subset V \\ 0 < |S| \leq n/2}} \frac{\sum_{i \in S, j \notin S} p_{ij}}{|S|}.$$

For this model, the main result of this paper is stated in the following theorem.

Theorem II.5. *Let P be a stochastic and symmetric matrix for which if $(i, j) \notin E$, $p_{ij} = 0$. There exists an algorithm $\mathcal{AP}^{\mathcal{Q}}$ for computing separable functions $f \in \mathcal{F}$ via communication of quantized messages, with quantization error no more than a given $\gamma = \Theta(\frac{1}{n})$, such that for any $\epsilon \in (\gamma f(x, V), \gamma f(x, V) + \frac{1}{2})$ and $\delta \in (0, 1)$,*

$$T_{\mathcal{AP}^{\mathcal{Q}}}^{cmp}(\epsilon, \delta) = O\left(\epsilon^{-2}(1 + \log \delta^{-1}) \frac{(\log n + \log \delta^{-1}) \log n}{\Phi(P)}\right).$$

Setting $\delta = \frac{1}{n^2}$ in the above bound, we have

$$T_{\mathcal{AP}^{\mathcal{Q}}}^{\text{cmp}}\left(\epsilon, \frac{1}{n^2}\right) = O\left(\epsilon^{-2} \frac{\log^3 n}{\Phi(P)}\right). \quad (1)$$

We note here that for this case, by an Information Theoretic lower bounded derived in [2] we have that the computation time is lower bounded as

$$T \geq \frac{\log \frac{1}{K\epsilon^2 + (\frac{1}{K})^{\frac{1}{n}}}}{\Phi(P)},$$

where K is a constant such that for all i, $f_i(x_i) \leq K$. Thus, the bound in (1) is tight in capturing the scaling of the computation time with respect to the graph conductance.

III. Unquantized Function Computation

In [3], a randomized algorithm is proposed for distributed computation of a separable function of the data in the network, so that with some specified probability, all nodes have an estimate of the function value within the desired interval of accuracy. The computation algorithm depends on

- the properties of exponentially distributed random variables, and,
- an information spreading algorithm used as a subroutine for the nodes to communicate their messages and determine the minimum of the messages.

The first of the two main theorems of [3] provides an upper bound on the computing time of the proposed computation algorithm and the second provides an upper bound on the information spreading time of a randomized gossip algorithm. These theorems are repeated below for convenience as our results build on those of [3].

Theorem III.1. *Given an information spreading algorithm $\mathcal{D}$ with δ-spreading time $T_{\mathcal{D}}^{spr}(\delta)$ for $\delta \in (0, 1)$, there exists an algorithm $\mathcal{A}$ for computing separable functions $f \in \mathcal{F}$ such that for any $\epsilon \in (0, 1)$ and $\delta \in (0, 1)$,*

$$T_{\mathcal{A}}^{cmp}(\epsilon, \delta) = O\left(\epsilon^{-2}(1 + \log \delta^{-1}) T_{\mathcal{D}}^{spr}\left(\frac{\delta}{2}\right)\right).$$

In the next section, we state a theorem analogous to this one, but for the case where the nodes are required to communicate a finite number of bits.

Next, the upper bound on the information spreading time is derived for the communication scheme, or equivalently, the randomized gossip algorithm, described in section II-A. We refer the reader to [3] for further details on the information spreading algorithm, including an analysis of the case of asynchronous communication. The theorem relevant to this paper follows.

Theorem III.2. *Consider any stochastic and symmetric matrix P such that if $(i, j) \notin E$, $p_{ij} = 0$. There exists an information spreading algorithm, $\mathcal{P}$, such that for any $\delta \in (0, 1)$,*

$$T_{\mathcal{P}}^{spr}(\delta) = O\left(\frac{\log n + \log \delta^{-1}}{\Phi(P)}\right).$$

IV. Quantized Function Computation

The nodes need to each acquire an estimate of $f(x,V) = \sum_{i=1}^{n} f_i(x_i)$. For convenience, we denote $f_i(x_i)$ by θ_i, and $y = f(x,V) = \sum_{i=1}^{n} \theta_i$ is the quantity to be estimated by the nodes. We denote the estimate of y at node i by $\hat{y}_i^Q$. The Q is added to emphasize that this estimate was obtained using an algorithm for nodes that can only communicate quantized values using messages consisting a finite number of bits.

We assume that node i can compute θ_i without any communication. Further, we assume that there exists a K for which: for all i, $\theta_i \in [1,K]$.

Recall that the goal is to design an algorithm such that, for large enough t, $\mathbf{P}\left\{\cap_{i=1}^{n}\{|\hat{y}_i^Q(t) - y| \leq \epsilon y\}\right\} \geq 1-\delta$, while communicating only a finite number of bits between the nodes. Again, we take advantage of the properties of exponentially distributed random variables, and an information spreading algorithm used as a subroutine for the nodes to determine the minimum of their values.

A. Computation of Minima Using Information Spreading

The computation of the minimum using the information spreading algorithm occurs as follows. Suppose that each node i has an initial vector $W^i = (W_1^i, \ldots, W_r^i)$ and needs to obtain $\bar{W} = (\bar{W}_1, \ldots, \bar{W}_r)$, where $\bar{W}_l = \min_{i=1,\ldots,n} W_l^i$. To compute $\bar{W}$, each node maintains an r-dimensional vector, $\hat{\bar{w}}^i = (\hat{\bar{w}}_1^i, \ldots, \hat{\bar{w}}_r^i)$, which is initially $\hat{\bar{w}}^i(0) = W^i$, and evolves such that $\hat{\bar{w}}^i(t)$ contains node $i's$ estimate of $\bar{W}$ at time t. Node i communicates this vector to its neighbors; and when it receives a message from a neighbor j at time t containing $\hat{\bar{w}}^j(t^-)$, node i will update its vector by setting $\hat{\bar{w}}_l^i(t^+) = \min(\hat{\bar{w}}_l^i(t^-), \hat{\bar{w}}_l^j(t^-))$, for $l = 1, \ldots, r$.

As argued in [3], when an information spreading algorithm $\mathcal{D}$ is used where one real-number is transferred between two nodes every time there is a communication, then with probability larger than $1-\delta$, for all i, $\hat{\bar{w}}^i(t) = \bar{W}$ when $t = rT_{\mathcal{D}}^{\text{spr}}(\delta)$, because the nodes propagate in the network an evolving estimate of the minimum, an r-vector, as opposed to the n r-vectors $W^1, \ldots, W^n$.

Now, suppose that node i quantizes a value $\hat{\bar{w}}_l^i$ that it needs to communicate to its neighbor, j, where node i maps the value $\hat{\bar{w}}_l^i$ to a finite set $\{1, \ldots M\}$ according to some quantization scheme. Then, $\log M$ bits have to be communicated between the nodes before j can decode the message and update its $\hat{\bar{w}}_l^j$. So, when each communication between nodes is a single bit, the time until all nodes' estimates are equal to $\bar{W}$ with probability larger than $1-\delta$ will increase by a factor of $\log M$, to $t = rT_{\mathcal{D}}^{\text{spr}}(\delta)\log M$.

B. Summary of Algorithm & Main Theorem

The proposed algorithm, $\mathcal{A}^{\mathcal{Q}}$ is summarized below.

1) Independently from all other nodes, node i generates r independent samples from an exponential distribution, with parameter θ_i. If a sample is larger than an m (which we will specify later), the node discards the sample and regenerates it.
2) The node quantizes each of the samples according to a scheme we describe in section IV-D. The quantizer maps points in the interval $[0,m]$ to the set $\{1,2,\ldots,M\}$.
3) Each of the nodes performs steps 1 and 2 and communicates its messages via the information spreading algorithm, $\mathcal{D}$, to the nodes with which it is connected. The nodes use the information spreading algorithm to determine the minimum of each of the r sets of messages. After $rT_{\mathcal{D}}^{\text{spr}}(\delta)\log M$ time has elapsed, each node has obtained the r minima with probability larger than $1-\delta$.
4) Node i sets its estimate of y, $\hat{y}_i^Q$, to be the reciprocal of the average of the r minima that it has computed.

Here, r is a parameter that will be designed so that $\mathbf{P}\left\{\cap_{i=1}^{n}\{|\hat{y}_i^Q - y| \leq \epsilon y\}\right\} \geq 1-\delta$ is achieved. Determining how large r and M must be leads to the main theorem of this paper.

Theorem IV.1. *Given an information spreading algorithm $\mathcal{D}$ with δ-spreading time $T_{\mathcal{D}}^{spr}(\delta)$ for $\delta \in (0,1)$, there exists an algorithm $\mathcal{A}^{\mathcal{Q}}$ for computing separable functions $f \in \mathcal{F}$ via communication of quantized messages, with quantization error no more than a given $\gamma = \Theta(\frac{1}{n})$, such that for any $\epsilon \in (\gamma f(x,V), \gamma f(x,V) + \frac{1}{2})$ and $\delta \in (0,1)$,*

$$T_{\mathcal{A}^{\mathcal{Q}}}^{cmp}(\epsilon,\delta) = O\left(\epsilon^{-2}(1+\log\delta^{-1})(\log n)T_{\mathcal{D}}^{spr}\left(\frac{\delta}{2}\right)\right).$$

Remark Here, we point out that the condition in the theorem that $\epsilon \in (y\gamma, y\gamma + 1/2)$ reflects the fact that due to quantization, $\hat{y}_i^Q$ can never get arbitrarily close to y, no matter how large r is chosen.

Before proving this theorem, it is convenient to consider the algorithm described above, excluding step 2; that is, with no sample quantization. In section IV-C, the derivation of the computation time of this modified algorithm will lead to determining the appropriate truncation parameter, m. In section IV-D we introduce a quantization scheme and determine the number of bits to use in order to guarantee that the node estimates of y converge with desired probability; we find that this number of bits is of the order of $\log n$.

C. Determining m

Before we state the lemma of this section, we describe the modified computation algorithm, $\mathcal{A}_{\mathcal{M}}^{\mathcal{Q}}$, which consists of steps 1 to 4 above excluding 2, and we introduce the necessary variables.

First, node i, independently from all other nodes, generates r samples drawn independently from an exponential distribution, with parameter θ_i. If a sample is larger than m, the node discards the sample and regenerates it. This is equivalent to drawing the samples from an exponential distribution truncated at m.

Let $(W_l^i)_T$ be the random variable representing the l^{th} sample at node i, where the subscript "T" emphasizes that the distribution is truncated. Then, the probability density function

of $(W_l^i)_T$ is that of an exponentially distributed random variable, W_l^i, with probability density function $f_{W_l^i}(w) = \theta_i e^{-\theta_i w}$ for $w \geq 0$, conditioned on the the event $A_l^i = \{W_l^i \leq m\}$. For $w \in [0, m]$,

$$f_{(W_l^i)_T}(w) = \frac{\theta_i e^{-\theta_i w}}{1 - e^{\theta_i m}},$$

and $f_{(W_l^i)_T}(w) = 0$ elsewhere.

Second, the nodes use a spreading algorithm, $\mathcal{D}$, so that each determines the minimum over all n for each set of samples, $l = 1, \ldots, r$. Recall that we consider the random variables at this stage as if there was no quantization. In this case, the nodes compute an estimate of $\bar{W}_l = \min_{i=1\ldots n}(W_l^i)_T$; we denote the estimate of $\bar{W}_l$ at node i by $\widehat{\bar{W}_l}^i$. Furthermore, we denote the estimates at node i of the minimum of each of each of the r set of samples by $\widehat{\bar{W}}^i = (\widehat{\bar{W}_1}^i, \ldots, \widehat{\bar{W}_r}^i)$, and the actual minima of the r set of samples by $\bar{W} = (\bar{W}_1, \ldots, \bar{W}_r)$.

It it is shown in [3] that by the aforementioned spreading algorithm, with probability at least $1 - \delta/2$, the estimates of the r minima, $\widehat{\bar{W}}^i$, will be be equal to the actual minima, $\bar{W}$, for all nodes, $i = 1, \ldots, n$, in $rT_{\mathcal{D}}^{spr}(\delta/2)$ time slots.

Last, each of the nodes computes its estimate, $\hat{y}_i$, of y by summing the r minimum values it has computed, inverting the sum, and multiplying by r:

$$\hat{y}_i = \frac{r}{\sum_{l=1}^r \widehat{\bar{W}_l}^i}.$$

The following lemma will be needed in the proof of Theorem IV.1.

Lemma IV.2. Let $\theta_1, \ldots, \theta_n$ be real numbers such that for all i, $\theta_i \geq 1$, $y = \sum_{i=1}^n \theta_i$ and $\bar{W} = (\bar{W}_1, \ldots, \bar{W}_r)$. Furthermore, let $\widehat{\bar{W}}^i = (\widehat{\bar{W}_1}^i, \ldots, \widehat{\bar{W}_r}^i)$ and let $\hat{y}_i$ denote node i's estimate of y using the modified algorithm of this section, $\mathcal{A}_{\mathcal{M}}^{\mathcal{Q}}$.

For any $\mu \in (0, 1/2)$, and for $I = ((1-\mu)\frac{1}{y}, (1+\mu)\frac{1}{y})$, if $m \geq \ln n - \ln(1 - e^{-\frac{\mu^2}{6}})$,

$$\mathbf{P}\left(\cup_{i=1}^n \{\hat{y}_i^{-1} \notin I\} | \forall i \in V, \widehat{\bar{W}}^i = \bar{W}\right) \leq e^{-r\frac{\mu^2}{6}},$$

where, $\hat{y}_i^{-1} = \frac{1}{r}\sum_{l=1}^r \widehat{\bar{W}_l}^i$.

Proof: First, note that when $\{\forall i \in V, \widehat{\bar{W}}^i = \bar{W}\}$, we have that for all i, $\hat{y}_i^{-1} = \frac{1}{r}\sum_{l=1}^r \bar{W}_l$. So, it is sufficient to show that

$$\mathbf{P}\left(\frac{1}{r}\sum_{l=1}^r \bar{W}_l \notin I\right) \leq e^{-r\frac{\mu^2}{6}}.$$

Let $W_l^* = \min_{i=1,\ldots,n} W_l^i$, the minimum of independent exponentially distributed random variables, W_l^i, with parameters $\theta_1, \ldots, \theta_n$ respectively, then W_l^* will itself be exponentially distributed with parameter $y = \sum_i \theta_i$. Observe that the cumulative distribution function of $\bar{W}_l$, $\mathbf{P}(\bar{W}_l \leq w)$, is identical to that of W_l^*, conditioned on the event $A_l = \{\cap_{i=1}^n A_l^i\}$, where $A_l^i = \{W_l^i \leq m\}$, $\mathbf{P}(W_l^* \leq w | A_l)$. Hence, we have that

$$\mathbf{P}\left(\frac{1}{r}\sum_{l=1}^r \bar{W}_l \notin I\right) = \mathbf{P}\left(\frac{1}{r}\sum_{l=1}^r W_l^* \notin I | \cap_{l=1}^r A_l\right).$$

Now, because $\mathbf{P}(A \cap B) \leq \mathbf{P}(A)$, it follows that

$$\mathbf{P}\left(\frac{1}{r}\sum_{l=1}^r W_l^* \notin I | \cap_{l=1}^r A_l\right)\mathbf{P}\left(\cap_{l=1}^r A_l\right) \leq \mathbf{P}\left(\frac{1}{r}\sum_{l=1}^r W_l^* \notin I\right).$$

From Cramer's Theorem and the properties of exponential distributions, we have that

$$\mathbf{P}\left(\frac{1}{r}\sum_{l=1}^r W_l^* \notin I\right) \leq e^{-r(\mu - \ln(1+\mu))}$$

and for $\mu \in (0, 1/2)$, $e^{-r(\mu-\ln(1+\mu))} \leq e^{-r\frac{\mu^2}{3}}$.

Next, we have that $\mathbf{P}\left(\cap_{l=1}^r A_l\right) = \left(\mathbf{P}\left(A_l\right)\right)^r$, because the $A_1, \ldots, A_r$ are mutually independent. Furthermore, $\mathbf{P}\left(A_l\right) \geq 1 - ne^{-m}$. To see this, note that the complement of A_l is $A_l^c = \{\cup_{i=1}^n \{W_l^i > m\}\}$, and $\mathbf{P}\left(W_l^i > m\right) = e^{-\theta_i m}$. So, by the union bound, we have

$$\mathbf{P}\left(A_l^c\right) \leq \sum_{i=1}^n e^{-\theta_i m} \leq ne^{-m},$$

where the last inequality follows because $\forall i, \theta_i \geq 1$.

Finally, putting all this together, we have that

$$\mathbf{P}\left(\frac{1}{r}\sum_{l=1}^r \bar{W}_l \notin I\right) \leq (1 - ne^{-m})^{-r} e^{-r\frac{\mu^2}{3}}.$$

Letting $1 - ne^{-m} \geq e^{-\frac{\mu^2}{6}}$ completes the proof. ■

D. Proof of Theorem IV.1

Before we proceed with the proof of the Theorem, we describe the quantization scheme. In step 2 of the algorithm $\mathcal{A}^{\mathcal{Q}}$, node i quantizes the sample it draws, a realization of $(W_l^i)_T$ denoted by w_l^i. The quantizer Q maps points in the interval $[0, m]$ to the set $\{1, 2, \ldots, M\}$. Each node also has a "codebook," Q^{-1}, a bijection that maps $\{1, 2, \ldots, M\}$ to $\{w_{q_1}, w_{q_2}, \ldots, w_{q_M}\}$, chosen such that for a given γ, $|w_l^i - Q^{-1}Q(w_l^i)| \leq \gamma$. We will denote $Q^{-1}Q(w_l^i)$ by $(w_l^i)_Q$.

While we do not further specify the choice of the quantization points, w_{q_k}, we will use the fact that the quantization error criterion can be achieved by a quantizer that divides the interval $[0, m]$ to no more than M intervals of length γ each. Then, the number of messages will be $M = m/\gamma$, and the number of bits that the nodes communicate is $\log M$.

Proof: We seek an upper bound on the (ϵ, δ)-computation time of the algorithm A^Q, the time until, with probability at least $1 - \delta$, all nodes $i = 1, \ldots, n$ have estimates $\hat{y}_i^Q$ that are within a factor of $1 \pm \epsilon$ of y. That is,

$$\mathbf{P}(\cup_{i=1}^n \{\hat{y}_i^Q \notin [(1-\epsilon)y, (1+\epsilon)y]\}) \leq \delta.$$

First, suppose that we may communicate real-valued messages between the nodes. We analyse the effect of quantization on the convergence of the node estimates to the desired $1 \pm \epsilon$ factor of y. For this, we compare the quantized algorithm, $\mathcal{A}^{\mathcal{Q}}$, with the modified algorithm $\mathcal{A}^{\mathcal{Q}}_{\mathcal{M}}$.

Note that for the above quantization scheme, for all i, l and any realization of $(W_l^i)_T$ denoted by w_l^i,

$$(w_l^i)_Q \in \left[w_l^i - \gamma, w_l^i + \gamma\right],$$

hence,

$$\min_{i=1,\ldots,n} (w_l^i)_Q \in \left[\min_{i=1,\ldots,n} w_l^i - \gamma, \min_{i=1,\ldots,n} w_l^i + \gamma\right],$$

and,

$$\frac{1}{r}\sum_{l=1}^{r} \min_{i=1,\ldots,n} (w_l^i)_Q \in \left[\frac{1}{r}\sum_{l=1}^{r} \min_{i=1,\ldots,n} w_l^i - \gamma, \frac{1}{r}\sum_{l=1}^{r} \min_{i=1,\ldots,n} w_l^i + \gamma\right]. \quad (2)$$

Note that $\frac{1}{r}\sum_{l=1}^{r} \min(w_l^i)_Q$ is a realization of $(\hat{y}_i^Q)^{-1}$.

Now, suppose that the information spreading algorithm, $\mathcal{D}$, is used so that in $O(rT_{\mathcal{D}}^{\text{spr}}(\delta/2))$ time,

$$\mathbf{P}\left(\cup_{i=1}^{n} \{\widehat{\bar{W}}^i \neq \bar{W}\}\right) \leq \frac{\delta}{2}. \quad (3)$$

Consider the case where $\{\cap_{i=1}^{n} \{\widehat{\bar{W}}^i = \bar{W}\}\}$, we have from Lemma IV.2 that, for any $\mu \in (0, 1/2)$, if $m = \ln n - \ln\left(1 - e^{-\frac{\mu^2}{6}}\right)$,

$$\mathbf{P}\left(\frac{1}{r}\sum_{l=1}^{r} \bar{W}_l \notin \left((1-\mu)\frac{1}{y}, (1+\mu)\frac{1}{y}\right)\right) \leq e^{-r\frac{\mu^2}{6}}.$$

Combining with (2), we have that

$$\mathbf{P}\left(\cup_{i=1}^{n} \left\{(\hat{y}_i^Q)^{-1} \notin \left((1-\mu)\frac{1}{y} - \gamma, (1+\mu)\frac{1}{y} + \gamma\right)\right\} \mid \cap_{i=1}^{n} \{\widehat{\bar{W}}^i = \bar{W}\}\right) \leq e^{-r\frac{\mu^2}{6}},$$

But the event

$$\left\{(\hat{y}_i^Q)^{-1} \notin \left((1-\mu)\frac{1}{y} - \gamma, (1+\mu)\frac{1}{y} + \gamma\right)\right\}$$

is equivalent to

$$\left\{(\hat{y}_i^Q) \notin \left((1+(\mu+y\gamma))^{-1}y, (1-(\mu+y\gamma))^{-1}y\right)\right\}.$$

And, letting $\epsilon = \mu + y\gamma$,

$$\left((1+\epsilon)^{-1}, (1-\epsilon)^{-1}\right) \subset (1-2\epsilon, 1+2\epsilon).$$

So,

$$\mathbf{P}\left(\cup_{i=1}^{n} \left\{|\hat{y}_i^Q - y| > 2\epsilon y\right\} \mid \cap_{i=1}^{n} \{\widehat{\bar{W}}^i = \bar{W}\}\right) \leq e^{-r\frac{\mu^2}{6}}.$$

Letting $r \geq 6\mu^{-2}\ln 2\delta^{-1}$, we have that

$$e^{-r\frac{\mu^2}{6}} \leq \frac{\delta}{2}.$$

Combining this with (3) in the Total Probability Theorem, we have the desired result,

$$\mathbf{P}(\cup_{i=1}^{n} \{\hat{y}_i^Q \notin [(1-2\epsilon)y, (1+2\epsilon)y]\}) \leq \delta.$$

Finally, recall that when the nodes communicate their real-valued messages, with high probability all nodes have estimates of the minima that they need in the computation of the estimate of y in $O(rT_{\mathcal{D}}^{\text{spr}}(\delta/2))$ time. So, the computation time is of that order.

Now, when instead the nodes need to communicate $\log M$ bits, as in the quantization algorithm described in this section, the information-spreading algorithm will be slowed down by $\log M$. Each bit requires $T_{\mathcal{D}}^{\text{spr}}(\delta)$ time slots to disseminate through the network, so $(\log M)T_{\mathcal{D}}^{\text{spr}}(\delta)$ time slots are needed until the quantized messages are disseminated and the minima computed. Consequently, the computation time of the quantized algorithm will be $O((\log M)rT_{\mathcal{D}}^{\text{spr}}(\delta/2))$.

But, $M = m/\gamma$, and by design, for a given μ we choose $m = \ln n - \ln\left(1 - e^{-\frac{\mu^2}{6}}\right)$; so $m = O(\log(n))$. Furthermore, we choose γ, such that $\gamma = \Theta(\frac{1}{n})$. Then,

$$\log M \leq \log\log n + \log n,$$

so, $\log M = O(\log n)$ bits are needed.

As we have previously seen, for $\mu \in (0, 1/2)$, $r \geq 6\mu^{-2}\ln 2\delta^{-1}$. But, $\mu = \epsilon - y\gamma$; and, $\gamma = \Theta(1/n)$ so, $y\gamma = O(1)$. We therefore have, for $\epsilon \in (y\gamma, y\gamma + 1/2)$,

$$T_{\mathcal{A}^{\mathcal{Q}}}^{\text{cmp}}(\epsilon, \delta) = O\left((\log n)\epsilon^{-2}(1 + \log\delta^{-1})T_{\mathcal{D}}^{\text{spr}}(\delta/2)\right). \blacksquare$$

V. Discussion and Conclusions

In this paper we have shown how a distributed algorithm for computing separable functions may be quantized so that the effect of the quantization scheme will be to slow down the information spreading by $\log n$, while the remaining performance characteristics of the original algorithm will be virtually unchanged. This result is stated in Theorem IV.1.

Combining the result of Theorem IV.1 with that of Theorem III.2 yields Theorem II.5. Comparison with a lower bound obtained via Information Theoretic inequalities in [2] reveals that the reciprocal dependence between computation time and graph conductance in the upper bound of Theorem II.5 matches the lower bound. Hence the upper bound is tight in capturing the effect of the graph conductance $\Phi(P)$.

References

[1] We refer the reader to a comprehensive list of references in [3].

[2] O. Ayaso, D. Shah, and M.A. Dahleh. "Lower bounds on information rates for distributed computation via noisy channels." Preliminary version in *Forty-Fifth Annual Allerton Conference on Communication, Control and Computing,* September 2007.

[3] D. Mosk-Aoyama and D. Shah. "Computing separable functions via gossip." Preliminary version appeared in *ACM Principles of Distributed Computation,* 2006.