ISIT'2013 1569719089

Computation Over Gaussian Networks With Orthogonal Components

Sang-Woon Jeon
Department of Information and Communication Engineering
Andong National University
Andong, Korea
swjeon@anu.ac.kr

Chien-Yi Wang, Michael Gastpar†
School of Computer and Communication Sciences
EPFL
Lausanne, Switzerland
{chien-yi.wang, michael.gastpar}@epfl.ch

***Abstract*—Function computation of arbitrarily correlated discrete sources over Gaussian networks with multiple access components but no broadcast is studied. Two classes of functions are considered: the arithmetic sum function and the frequency histogram function. The arithmetic sum function in this paper is defined as a set of multiple weighted arithmetic sums, which includes averaging of sources and estimating each of the sources as special cases. The frequency histogram function counts the number of occurrences of each argument, which yields many important statistics such as mean, variance, maximum, minimum, median, and so on. For a class of networks, an approximate computation capacity is characterized. The proposed approach first abstracts Gaussian networks into the corresponding modulo-sum multiple-access channels via lattice codes and linear network coding and then computes the desired function by using linear Slepian–Wolf source coding.**

I. Introduction

In various applications such as distributed averaging, alarm detection, and environmental monitoring, a fusion center or coordination node wishes to learn a *function of the data or measurements* via communication networks, rather than the data or measurements themselves. Due to the many-to-one mapping property between raw data or measurements and a desired function, traditional communication philosophies developed for estimating data or measurements can be quite suboptimal for function computation. In other words, source–channel separation [1] does not hold and, as a result, joint source–channel coding is required to efficiently compute a desired function [2]–[6].

In many cases of interest, a fusion center may wish to estimate the *sample mean* of measurements [7], [8], for example, average temperature for environmental monitoring. For alarm detection, a relevant function will be the *maximum or minimum* value of measurements [7], [8]. A substantial body of work has addressed various cases [2]–[8], but in general, it is still unclear how to compute these statistics efficiently in a distributed manner over Gaussian networks.

In this paper, we study the computation of a more general class of functions over Gaussian networks with orthogonal components. Specifically, we consider *Gaussian network with multiple-access components* (but no broadcast component), which includes the Gaussian multiple-access channel (MAC) and more generally Gaussian tree networks as special cases. We focus on two classes of desired functions: *the arithmetic sum function and the frequency histogram function*. The former is defined as a set of multiple weighted arithmetic sums, which includes averaging of measurements and estimating each of the measurements as special cases. The latter counts the number of occurrences of each argument among measurements. The frequency histogram is very powerful since it yields many important statistics such as mean, variance, maximum, minimum, median, and any symmetric function invariant to permutations of their arguments (see the definition and subclasses of symmetric functions in [7]).

†He is also with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, USA.

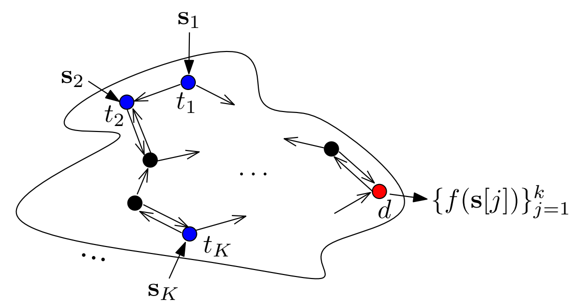

Fig. 1. Computation over a network in which node t_i observes the length-k source $\mathbf{s}_i = [s_i[1], \cdots, s_i[k]]^T$ and node d wishes to compute the desired function $\{f(\mathbf{s}[j])\}_{j=1}^k$, where $\mathbf{s}[j] = [s_1[j], \cdots, s_K[j]]^T$.

II. Problem Formulation

Throughout the paper, we denote $[1:n] := \{1, 2, \cdots, n\}$, $\mathsf{C}(x) := \frac{1}{2}\log(1+x)$, and $\mathsf{C}^+(x) := \max\left\{\frac{1}{2}\log(x), 0\right\}$. For $x_i \in \mathbb{F}_p$, $\bigoplus_{i=1}^n x_i$ denotes the modulo-p sum of $\{x_i\}_{i\in[1:n]}$. Let $\mathrm{card}(\cdot)$ denote the cardinality of a set and $\mathbf{1}_{(\cdot)}$ denote the indicator function of an event.

A. Network Model

Consider a network represented by a directed graph $G = (V, E)$ depicted in Fig. 1. Denote the set of incoming and outgoing nodes at node $v \in V$ by $\Gamma_{in}(v) = \{u : (u, v) \in E\}$ and $\Gamma_{out}(v) = \{u : (v, u) \in E\}$, respectively. Denote the ith sender, $i \in [1:K]$, by $t_i \in V$ and suppose that it observes a length-k source vector $\mathbf{s}_i = [s_i[1], \cdots, s_i[k]]^T \in [0:p-1]^k$. The receiver $d \in V$ wishes to compute a symbol-by-symbol function of K sources, i.e., $f(\mathbf{s}[j])$ for all $j \in [1:k]$, where $\mathbf{s}[j] = [s_1[j], \cdots, s_K[j]]^T$ denotes the set of K sources at time j. We assume that $d \notin \{t_i\}_{i\in[1:K]}$ and G contains a directed path from all nodes in V to the receiver d. Without loss of generality, the nodes with no incoming edges are sources.

Definition 1 (Sources): Let $\mathbf{S} = [S_1, \cdots, S_K]^T \in [0:p-1]^K$ be a random vector with a joint probability mass function $p_{\mathbf{S}}(\cdot)$. We assume that at each time $j \in [1:k]$ the set of K sources $\mathbf{s}[j]$ is drawn independently from $p_{\mathbf{S}}(\cdot)$.

We will consider two desired functions: the arithmetic sum function and the frequency histogram or type function, whose formal definitions are given below.

Definition 2 (Arithmetic Sum): For arithmetic sum computation, the desired function is given by $f(\mathbf{s}[j]) = \{\sum_{i=1}^K a_{li}s_i[j]\}_{l=1}^L$, where $a_{li} \in [0:p-1]$. Hence $f(\mathbf{s}[j]) \in [0:(p-1)^2K]^L$ for the arithmetic sum function.

Definition 3 (Frequency Histogram): For frequency histogram or type computation, the desired function is given by $f(\mathbf{s}[j]) = \{b_0(\mathbf{s}[j]), \cdots, b_{p-1}(\mathbf{s}[j])\}$, where $b_l(\mathbf{s}[j]) = \mathrm{card}(\{i \in [1:K] | s_i[j] = l\})$ for $l \in [0:p-1]$. Hence $f(\mathbf{s}[j]) \in [0:K]^p$ for the frequency histogram function.

Let $f(\mathbf{S})$ denote the desired function induced by the random source vector $\mathbf{S}$. The following two definitions define random variables associated with the desired function, which will be used throughout the paper.

Definition 4 (Arithmetic Sum Induced by $\mathbf{S}$*):* Define $U_l = \sum_{i=1}^K a_{li}S_i$ for $l \in [1:L]$, which are the random variables associated with the arithmetic sum function. Then $f(\mathbf{S}) = (U_1, \cdots, U_L)$ for the arithmetic sum function.

Definition 5 (Frequency Histogram Induced by $\mathbf{S}$*):* Define $B_l = \mathrm{card}(\{i \in [1:K] | S_i = l\})$ for $l \in [0:p-1]$, which are the random variables associated with the frequency histogram function. Then $f(\mathbf{S}) = (B_0, \cdots, B_{p-1})$ for the frequency histogram function.

Associated with $G = (V, E)$, we consider the following Gaussian network with multiple access components.

Definition 6 (Gaussian Networks With Multiple-Access): Associated with $G = (V, E)$, the length-n time-extended received signal vector of node v is given by

$$\mathbf{y}_v = \sum_{u \in \Gamma_{in}(v)} h_{u,v}\mathbf{x}_{u,v} + \mathbf{z}_v, \qquad (1)$$

where $\mathbf{x}_{u,v}$ and $h_{u,v}$ denote the length-n time-extended transmit signal vector and the scalar channel coefficient from node u to node v, respectively. The elements of $\mathbf{z}_v$ are i.i.d. drawn from $\mathcal{N}(0,1)$ and $\frac{1}{n}\|\mathbf{x}_{u,v}\|^2 \le P$ should be satisfied for all $(u,v) \in E$.

B. Computation Capacity

Based on the above network and channel model, the length-n block code for Gaussian networks with multiple-access is defined as follows:

- (Sender Encoding) The ith sender t_i transmits $x_{t_i,w}^{(t)} = \varepsilon_{t_i,w}^{(t)}\left(\mathbf{s}_i, \mathbf{y}_{t_i}^{t-1}\right)$ for $t \in [1:n]$ to node $w \in \Gamma_{out}(t_i)$.
- (Relay Encoding) Node $v \notin \{t_i\}_{i\in[1:K]} \bigcup \{\mathrm{d}\}$ transmits $x_{v,w}^{(t)} = \varepsilon_{v,w}^{(t)}\left(\mathbf{y}_v^{t-1}\right)$ for $t \in [1:n]$ to node $w \in \Gamma_{out}(v)$.
- (Decoding) The receiver d estimates $\hat{f}(\mathbf{s}[j]) = \psi^{(j)}\left(\mathbf{y}_d\right)$ for $j \in [1:k]$.

The probability of error is defined by $P_e^{(n)} = \Pr\left[\bigcup_{j=1}^k \{\hat{f}(\mathbf{s}[j]) \neq f(\mathbf{s}[j])\}\right]$. We then define the computation capacity as the follow.

Definition 7 (Computation Capacity): We say that the computation rate $R := \frac{k}{n}$ is achievable if there exists a sequence of length-n block codes such that $P_e^{(n)}$ converges to zero as n increases. The computation capacity is the maximum over all achievable computation rates.

III. Main Results

In this section, we state our main results. We introduce an upper bound on the computation capacity in Section III-A and then state our achievable computation rate and establish an approximate computation capacity for a class of networks in Section III-B.

To state the main results, we first explain the notation for the cut-set bound. For a subset $\Sigma \subseteq [1:K]$, define $G(\Sigma) = (V(\Sigma), E(\Sigma))$ as the subgraph of G consisting of a subset of nodes in V such that for any node $v \in V(\Sigma)$ there exists a directed path from at least one sender in $\{t_i\}_{i\in\Sigma}$ to node v. Let $\Lambda(\Sigma)$ denote the set of all cuts dividing all of the senders in $\{t_i\}_{i\in\Sigma}$ from the receiver d on $G(\Sigma)$. Then denote

$$\bar{C}(\Sigma) = \min_{\Omega \in \Lambda(\Sigma)} \sum_{v \in V(\Sigma)\setminus\Omega} \mathsf{C}\left(\left(\sum_{u \in \Omega \cap \Gamma_{in}(v)} h_{u,v}\right)^2 P\right). \qquad (2)$$

Here a (vertex) cut is a partitioning of $V(\Sigma)$ into two disjoint sets Ω and $V(\Sigma)\setminus\Omega$. To keep notation simple, we use the term "cut" to represent the subset Ω of $V(\Sigma)$. From the definition of $\Lambda(\Sigma)$, we only consider cuts where $\{t_i\}_{i\in\Sigma} \subseteq \Omega$ and $d \in V(\Sigma)\setminus\Omega$. Note that $\bar{C}(\Sigma)$ is an upper bound on the minimum-cut value over all cuts in $\Lambda(\Sigma)$ on $G(\Sigma)$ for Gaussian networks with multiple-access.

A. Upper Bound

From the cut-set upper bound (2), we derive a simple upper bound on the computation capacity. By assuming that a genie provides $\{\mathbf{s}_i\}_{i\in[1:K]\setminus\Sigma}$ to the receiver, computing $\{f(\mathbf{s}[j]) | \{s_i[j]\}_{i\in[1:K]\setminus\Sigma}\}_{j=1}^k$ at the receiver is enough to recover the desired function $\{f(\mathbf{s}[j])\}_{j=1}^k$. Hence assuming full cooperation between the nodes in Ω and between the nodes in Ω^c, from the source–channel separation theorem [1],

$$R \le \min_{\Sigma \subseteq [1:K]} \frac{\bar{C}(\Sigma)}{H(f(\mathbf{S}) | \{S_i\}_{i\in[1:K]\setminus\Sigma})} \qquad (3)$$

for Gaussian networks with multiple-access.

B. Constructive Lower Bound

For the achievability, we first abstract each multiple-access component by compute-and-forward and then apply linear network coding at each relay node in order to convert the original network into the corresponding modulo-q sum channel [6], [9]. Then we apply linear Slepain–Wolf source coding to compute the desired function over the converted modulo-q sum

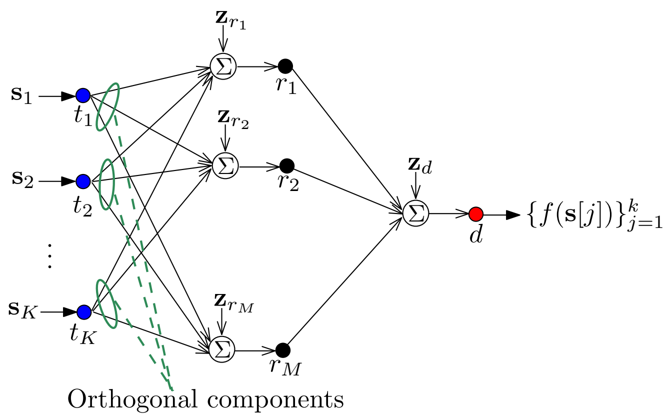

Fig. 2. Example of Gaussian networks with multiple-access that satisfies the condition in Corollary 1, where the channel coefficients are equal to one on all links.

channel [4], [10]. The following theorem shows an achievable computation rate for Gaussian networks with multiple-access.

Theorem 1: Consider the Gaussian network with multiple-access in Definition 6. Suppose that the receiver wishes to compute the arithmetic sum in Definition 2 or the frequency histogram in Definition 3. Then the computation rate of

$$R = \frac{\min_{i\in[1:K]} C^+(\{i\})}{H(f(\mathbf{S}))} \tag{4}$$

is achievable, where

$$C^+(\{i\}) = \min_{\Omega\in\Lambda(\{i\})} \sum_{v\in\Omega^c} \Bigg(\mathbf{1}_{\{u\in\Omega\cap\Gamma_{in}(v)\}\neq\emptyset} \cdot \mathsf{C}^+\left(\frac{1}{\mathrm{card}(\Gamma_{in}(v))} + \min_{u\in\Gamma_{in}(v)} h_{u,v}^2 P\right)\Bigg) \tag{5}$$

and $f(\mathbf{S})$ is given in Definition 4 for the arithmetic sum and in Definition 5 for the frequency histogram, respectively.

Proof: Due to page limitation, we briefly explain the main idea of the achievability in Section IV and refer to the full paper at Arxiv for the detailed proof. ■

Corollary 1 (Approximate Computation Capacity): If, for each $v \in V$, $h_{u,v}$ are the same for all $u \in \Gamma_{in}(v)$ and $\bar{C}([1:K]) - \min_{i\in[1:K]} \bar{C}(\{i\}) \leq c_1|V|\log|V|$ for some constant $c_1 > 0$, then the computation rate in Theorem 1 achieves the computation capacity to within $c_2\frac{|V|\log|V|}{H(f(\mathbf{S}))}$ functions per channel use for any power P, where $c_2 > 0$ is some constant.

Proof Outline: Assuming $\Sigma = [1:K]$ in (3) provides $R \leq \frac{\bar{C}([1:K])}{H(f(\mathbf{S}))}$. After some manipulation, we can show that the gap between the above upper bound and (4) is upper bounded by $c_2\frac{|V|\log|V|}{H(f(\mathbf{S}))}$ under the condition in Corollary 1. ■

Figure 2 is an example network satisfying the condition in Corollary 1. Basically, any network having the same channel coefficients at each multiple-access component satisfies the condition in Corollary 1.

C. Orthogonal Gaussian Networks

We also study orthogonal Gaussian networks with no broadcast and no multiple-access component. That is, the length-n time-extended inout–output is given by $\mathbf{y}_{u,v} = h_{u,v}\mathbf{x}_{u,v} + \mathbf{z}_{u,v}$ for every $(u,v) \in E$ in this case. We show that

$$R = \frac{\min_{i\in[1:K]} \min_{\Omega\in\Lambda(\{i\})} \sum_{\substack{u\in\Omega, v\in V(\{i\})\setminus\Omega \\ s.t. (u,v)\in E(\{i\})}} \mathsf{C}\left(h_{u,v}^2 P\right)}{H(f(\mathbf{S}))} \tag{6}$$

is achievable and then prove that (6) provides the computation capacity for a class of networks. We refer to the full paper at Arxiv for the detailed proof.

IV. Computation Over Gaussian Networks With Multiple-Access

In this section, we provide the main ingredients in order to achieve the computation rate presented in Theorem 1. Due to page limitation, the discussion in Sections IV-A and IV-B is restricted to the computation of a single arithmetic sum $\sum_{i=1}^K \mathbf{s}_i$ based on the layered network in Fig. 2. Then we will briefly explain how to extend to the computation of multiple (weighted) arithmetic sums and the frequency histogram in Section IV-C.

From (1), the length-n output of relay node r_j, $j \in [1:M]$, and receiver d in Fig. 2 respectively are given by

$$\mathbf{y}_{r_j} = \sum_{i=1}^K \mathbf{x}_{t_i,r_j} + \mathbf{z}_{r_j}, \tag{7}$$

$$\mathbf{y}_d = \sum_{j=1}^M \mathbf{x}_{r_j,d} + \mathbf{z}_d. \tag{8}$$

From Theorem 1, the computation rate of $R = \frac{\mathsf{C}^+\left(\frac{1}{K}+P\right)}{H(\sum_{i=1}^K S_i)}$ is achievable for a single arithmetic sum computation, i.e., $\{f(\mathbf{s}[j])\}_{j=1}^k = \sum_{i=1}^K \mathbf{s}_i$. In the following two subsections, we will show how to achieve this computation rate.

A. Transformation Into Modulo-q Sum Channel

We first apply lattice codes [6], [11] to transform the length-n Gaussian multiple-access components in (7) and (8) respectively into the corresponding length-m modulo-q sum components: $\mathbf{y}'_{r_j} = \bigoplus_{i=1}^K \mathbf{x}'_{t_i,r_j}$ and $\mathbf{y}'_d = \bigoplus_{j=1}^M \mathbf{x}'_{r_j,d}$ for n sufficiently large, where $\mathbf{x}'_{t_i,r_j} \in \mathbb{F}_q^m$ and $\mathbf{x}'_{r_j,d} \in \mathbb{F}_q^m$. Here q is set to be the largest prime number between $[1:n\log n]$ and

$$m = n\mathsf{C}^+\left(\frac{1}{K}+P\right)(\log q)^{-1}. \tag{9}$$

Note that we treated $\mathbf{y}'_{r_j}$ and $\mathbf{y}'_d$ as the desired functions in [6, Theorem 2], see also [6, Example 1].

We then apply linear network coding at each sender and relay node over the above transformed network. Specifically, sender t_i transmits $\mathbf{x}'_{t_i,r_j} = \mathbf{F}_{t_i,r_j}\mathbf{x}_{t_i}$ to relay node r_j, where $\mathbf{F}_{t_i,r_j} \in \mathbb{F}_q^{m\times m}$ and $\mathbf{x}_{t_i} \in \mathbb{F}_q^m$. Relay node r_j transmits $\mathbf{x}'_{r_i,d} = \mathbf{F}_{r_j,d}\mathbf{y}_{r_j}$ to receiver d, where $\mathbf{F}_{r_j,d} \in \mathbb{F}_q^{m\times m}$. Then

$$\mathbf{y}'_d = \bigoplus_{i=1}^K \left(\bigoplus_{j=1}^M \mathbf{F}_{r_j,d}\mathbf{F}_{t_i,r_j}\right)\mathbf{x}_{t_i}. \tag{10}$$

If we set the elements of $\mathbf{F}_{t_i,r_j}$ and $\mathbf{F}_{r_j,d}$ i.i.d. uniformly at random from $\mathbb{F}_q$, the $m \times m$ matrix $\bigoplus_{j=1}^{M} \mathbf{F}_{r_j,d}\mathbf{F}_{t_i,r_j}$ has rank m for all $i \in [1:K]$ with probability approaching one as m increases (equivalently, as n increases) [9]. Hence, by setting $\mathbf{x}_{t_i} = (\bigoplus_{j=1}^{M} \mathbf{F}_{r_j,d}\mathbf{F}_{t_i,r_j})^{-1}\mathbf{x}'_{t_i}$, $\mathbf{x}'_{t_i} \in \mathbb{F}_q^m$, we can transform the length-n original Gaussian network into the following length-m modulo-q sum channel:

$$\mathbf{y}'_d = \bigoplus_{i=1}^{K} \mathbf{x}'_{t_i}. \tag{11}$$

B. Arithmetic Sum Over Modulo-q Sum Channel

Now consider the arithmetic sum computation over the transformed modulo-q sum channel in (11). The key observation is that utilizing a small portion of finite field elements and then computing the corresponding modulo-q sum can attain the desired arithmetic sum function. Furthermore, linear source coding in [10] can compensate the inefficiency of utilizing only a small portion of finite field elements. In the end, this approach optimally computes the arithmetic sum over the modulo-q sum channel in (11). Therefore, rate loss only occurs during the transformation procedure described in Section IV-A.

The following lemma shows the existence of a linear compression matrix that optimally compresses finite field sources.

Lemma 1 (Csiszár): Let $\mathbf{u} \in \mathbb{F}_q^k$ be the length-k source i.i.d. drawn from a probability mass function $p_U(\cdot)$. Then there exists a linear compression matrix $\mathbf{H} \in \mathbb{F}_q^{\frac{H(U)}{\log q}k \times k}$ such that $\Pr[\hat{\mathbf{u}}(\mathbf{Hu}) \neq \mathbf{u}] \to 0$ as k increases.

Proof: We refer to [10] for the proof. ■

The receiver first computes the modulo-q sum of the K sources of size p instead of the desired arithmetic sum, where q is the finite field size of the transformed modulo-q sum channel in (11). The linear compression matrix $\mathbf{H}$ in Lemma 1 is used to compress the modulo-q sum of the K sources. Notice that this modulo-q sum can be compressed in a distributed manner by multiplying $\mathbf{H}$ at each sender, see (12) and (13). Since q is arbitrarily large as the block length n increases, the computed modulo-q sum becomes the desired arithmetic sum for n sufficiently large. Therefore, the receiver can attain the arithmetic sum from the computed modulo-q sum. The following is the formal proof of how to achieve the computation rate in Theorem 1 over the transformed channel in (11) for a single arithmetic sum.

Proof of Theorem 1 for a single arithmetic sum: Let $g(\cdot)$ denote the mapping from a subset of integers $[0:q-1]$ to the corresponding finite field $\mathbb{F}_q$. Define $s'_i[j] = g(s_i[j])$ and $U' = \bigoplus_{i=1}^{K} g(S_i)$. Suppose that sender t_i observes the length-k source $\mathbf{s}'_i = [s'_i[1], \cdots, s'_i[k]]^T$, which can be obtained from the original source $\mathbf{s}_i$, and receiver d wishes to compute $\mathbf{u}' = \bigoplus_{i=1}^{K} \mathbf{s}'_i$ over the transformed length-m modulo-q sum channel in (11).

Denote $\mathbf{H} \in \mathbb{F}_q^{\frac{H(U')}{\log q}k \times k}$ as the linear compression matrix in Lemma 1 for the length-k source $\mathbf{u}'$. Then sender t_i transmits

$$\mathbf{x}'_{t_i} = \mathbf{H}\mathbf{s}'_i \tag{12}$$

for $i \in [1:K]$, where we set $k = \frac{m \log q}{H(U')}$. Then (11) and (12) yield

$$\mathbf{y}'_d = \bigoplus_{i=1}^{K} \mathbf{x}'_{t_i} = \mathbf{H} \bigoplus_{i=1}^{K} \mathbf{s}'_i. \tag{13}$$

Hence, from Lemma 1, receiver d can estimate $\bigoplus_{i=1}^{K} \mathbf{s}'_i$ with an arbitrarily small error as n increases. Therefore, the computation rate of

$$R = \frac{k}{n} = \frac{m \log q}{nH(U')} = \frac{\mathsf{C}^+\left(\frac{1}{K} + P\right)}{H(U')} \tag{14}$$

is achievable for the desired function $\mathbf{u}'$, where the last equality follows from (9). Since there exists $n_0 \geq 0$ such that $q > (p-1)K$ for all $n \geq n_0$ (q is the largest prime number between $[1 : n \log n]$), we have

$$\mathbf{u}' = \left[g\left(\sum_{i=1}^{K} s_i[1]\right), \cdots, g\left(\sum_{i=1}^{K} s_i[k]\right) \right]^T, \tag{15}$$

$$U' = g\left(\sum_{i=1}^{K} S_i\right) \tag{16}$$

for n sufficiently large. Since $g(\cdot)$ is a one-to-one correspondence, from (15), the receiver can compute the arithmetic sum $\sum_{i=1}^{K} \mathbf{s}_i$ from $\mathbf{u}'$ for n sufficiently large. Finally, from (14), (16), and the fact that $H(g(\sum_{i=1}^{K} S_i)) = H(\sum_{i=1}^{K} S_i)$, the achievable computation rate for the arithmetic sum $\sum_{i=1}^{K} \mathbf{s}_i$ is given by $R = \frac{\mathsf{C}^+\left(\frac{1}{K}+P\right)}{H(\sum_{i=1}^{K} S_i)}$, which completes the proof.

C. Extension to Frequency Histogram Computation

Now consider the frequency histogram computation over (11). Let $b_{li}[j] = \mathbf{1}_{s_i[j]=l}$. Then the desired function at time j is given as

$$f(\mathbf{s}[j]) = \left(\sum_{i=1}^{K} b_{0i}[j], \sum_{i=1}^{K} b_{1i}[j], \cdots, \sum_{i=1}^{K} b_{(p-1)i}[j] \right). \tag{17}$$

Hence, by treating $b_{li}[j]$ as a binary source, the frequency histogram can be represented as multiple arithmetic sums of binary sources. Therefore we can apply the previous computation scheme in Section IV-B for computing each of the arithmetic sums. However, we can further improve a computation rate by using the linear Slepian–Wolf source coding [4], [10], which exploits correlation between the arithmetic sums in (17).

Let $b'_{li}[j] = g(b_{li}[j])$ and $\mathbf{b}'_{li} = [b'_{li}[1], \cdots, b'_{li}[k]]^T$. Suppose that the ith sender observes $\mathbf{b}'_{li}$, which can be obtained from $\mathbf{s}_i$, and the receiver wishes to compute $\{\mathbf{u}''_l = \bigoplus_{i=1}^{K} \mathbf{b}'_{li}\}_{l=0}^{p-1}$ over (11). Similar to Lemma 1, denote the linear Slepian–Wolf compression matrices by $\mathbf{H}_0, \cdots, \mathbf{H}_{p-1}$ that correspond to some point in the Slepian–Wolf rate region with the sum rate $H(B'_0, \cdots, B'_{p-1})$, where $B'_l = \bigoplus_{i=1}^{K} B'_{li}$ and $B'_{li} = g(\mathbf{1}_{S_i=l})$. Then sender t_i transmits

$\mathbf{x}_i = [(\mathbf{H}_0\mathbf{b}'_{0i})^T, \cdots, (\mathbf{H}_{p-1}\mathbf{b}'_{(p-1)i})^T]^T$, where we set $k = \frac{m \log q}{H(B'_0,\cdots,B'_{p-1})}$. Then (11) yields

$$\mathbf{y}'_d = \left[\left(\mathbf{H}_0\left(\bigoplus_{i=1}^{K}\mathbf{b}'_{0i}\right)\right)^T, \cdots, \left(\mathbf{H}_{p-1}\left(\bigoplus_{i=1}^{K}\mathbf{b}'_{(p-1)i}\right)\right)^T\right]^T. \tag{18}$$

Therefore, for n sufficiently large, receiver d can estimate $(\bigoplus_{i=1}^{K}\mathbf{b}'_{0i}, \cdots, \bigoplus_{i=1}^{K}\mathbf{b}'_{(p-1)i})$ from $\mathbf{y}'_d$. From the same arguments in (14) to (16), the receiver can compute the desired frequency histogram with the rate of $R = \frac{\mathsf{C}^+(\frac{1}{K}+P)}{H(B_0,\cdots,B_{p-1})}$.

V. Comparison With Separation-Based Computation

A naive approach to computation is to communicate all the data to the receiver. A first improvement is to use clever source coding techniques, taking into account the specific computation that needs to be executed, but to still use the channel network at capacity. In line with [4, Section III], we refer to this as separation-based computation. In this section, we compare the proposed computation scheme with the separation-based computation.

For simple explanation, consider the arithmetic sum computation $\sum_{i=1}^{K}\mathbf{s}_i$ over the Gaussian MAC assuming that the channel coefficients are equal to one from now on. Let $\mathbf{R}_{\mathsf{f}}$ denote the distributed compression rate region for computing $\{f(\mathbf{s}[j])\}_{j=1}^{k}$ (see [4, Definition 8]) and $\mathbf{C}_{\mathsf{mac}}$ denote the capacity region of the Gaussian MAC, which can be represented as the set of all rate tuples $(R_1, \cdots, R_K)$ satisfying

$$\sum_{i\in\Sigma} R_i \le \mathsf{C}(\mathrm{card}(\Sigma)P) \text{ for all } \Sigma \subseteq [1:K]. \tag{19}$$

Then a computation rate $R^{(\mathsf{sep})}$ is achievable by separation if

$$\mathbf{R}_{\mathsf{f}} \cap \mathbf{C}'_{\mathsf{mac}} \neq \emptyset, \tag{20}$$

where $\mathbf{C}'_{\mathsf{mac}} = \left\{\left(\frac{R_1}{R^{(\mathsf{sep})}}, \cdots, \frac{R_K}{R^{(\mathsf{sep})}}\right) : (R_1, \cdots, R_K) \in \mathbf{C}_{\mathsf{mac}}\right\}$.

Example 1 (i.i.d. Binary Sources): Suppose that the sources are binary and i.i.d. drawn from $\mathrm{Bern}(1/2)$. Then, $\mathbf{R}_{\mathsf{f}}$ is given by all rate tuples $(R_{\mathsf{f},1}, \cdots, R_{\mathsf{f},K})$ satisfying $R_{\mathsf{f},i} \ge H(S_i) = 1$ for all $i \in [1:K]$. Hence, an achievable computation rate by separation is given by $R^{(\mathsf{sep})} = \frac{\mathsf{C}(KP)}{K}$ from (19) and (20). Let $U = \sum_{i=1}^{K} S_i$. Then, from Theorem 1, $R^{(\mathsf{comp})} := \frac{\mathsf{C}^+(\frac{1}{K}+P)}{H(U)}$ is achievable, where $H(U) = -\sum_{x=0}^{K} p_U(x)\log(p_U(x))$ and $p_U(x) = \binom{K}{x}2^{-K}$. Lastly, the cut-set upper bound in (3) assuming $\Sigma = [1:K]$ is given by $R^{(\mathsf{upper})} := \frac{\mathsf{C}(K^2P)}{H(U)}$. Figure 3 plots $R^{(\mathsf{upper})}$, $R^{(\mathsf{comp})}$, and $R^{(\mathsf{sep})}$ with respect to the number of sources K. Since $\frac{1}{2}\log(\pi K/2) \le H(U) \le \frac{1}{2}\log(\pi e K/2)$ [12, Lemma 2.1], $R^{(\mathsf{upper})} = \Theta(1)$, $R^{(\mathsf{comp})} = \Theta(\frac{1}{\log K})$, and $R^{(\mathsf{sep})} = \Theta(\frac{\log K}{K})$ as K increases.

Remark 1 (Modulo-p Sum): Although the modulo-p sum computation over MAC has actively been studied in the literature, for example see [2]–[6], it is still unclear how to compute the modulo-p sum over Gaussian MAC for a fixed p.

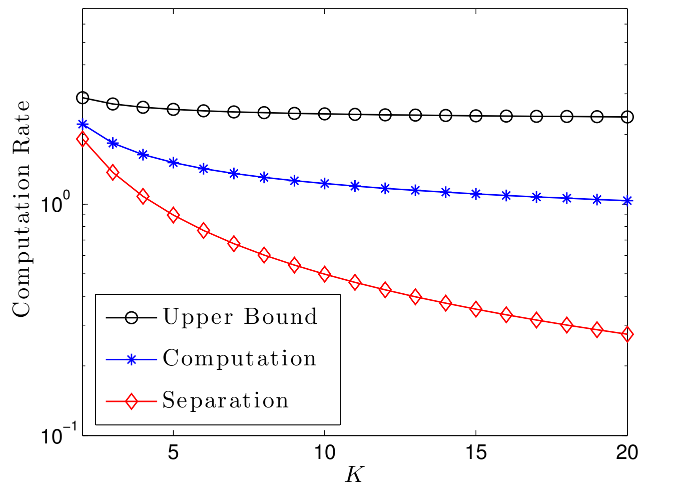

Fig. 3. Computation rates with respect to the number of sources K for $P = 20$ dB.

Since the arithmetic sum implies the modulo-p sum, $R^{(\mathsf{comp})}$ in Example 1 is also an achievable computation rate for the modulo-p sum, which again outperforms the separation-based computation.

Acknowledgement

This work has been supported in part by the European ERC Starting Grant 259530-ComCom. This research was funded in part by the MSIP (Ministry of Science, ICT & Future Planning), Korea in the ICT R&D Program 2013.

References

[1] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 1948.

[2] J. Körner and K. Marton, "How to encode the modulo-two sum of binary sources," *IEEE Trans. Inf. Theory*, vol. IT-25, pp. 219–221, Mar. 1979.

[3] R. Ahlswede and T. S. Han, "On source coding with side information via a multiple-access channel and related problems in multi-user information theory," *IEEE Trans. Inf. Theory*, vol. IT-29, pp. 396–412, May 1983.

[4] B. Nazer and M. Gastpar, "Computation over multiple-access channels," *IEEE Trans. Inf. Theory*, vol. 53, pp. 3498–3516, Oct. 2007.

[5] J. Zhan, S. Y. Park, M. Gastpar, and A. Sahai, "Linear function computation in networks: Duality and constant gap results," *IEEE J. Select. Areas Commun.*, vol. 31, pp. 620–638, Apr. 2013.

[6] B. Nazer and M. Gastpar, "Compute-and-forward: Harnessing interference through structured codes," *IEEE Trans. Inf. Theory*, vol. 57, pp. 6463–6486, Oct. 2011.

[7] A. Giridhar and P. R. Kumar, "Computing and communicating functions over sensor networks," *IEEE J. Select. Areas Commun.*, vol. 23, pp. 755–764, Apr. 2005.

[8] L. Ying, R. Srikant, and G. E. Dullerud, "Distributed symmetric function computation in noisy wireless sensor networks," *IEEE Trans. Inf. Theory*, vol. 53, pp. 4826–4833, Dec. 2007.

[9] A. S. Avestimehr, S. N. Diggavi, and D. N. C. Tse, "Wireless network information flow: A deterministic approach," *IEEE Trans. Inf. Theory*, vol. 57, pp. 1872–1905, Apr. 2011.

[10] I. Csiszár, "Linear codes for sources and source networks: Error exponents, universal coding," *IEEE Trans. Inf. Theory*, vol. IT-28, pp. 585–592, Jul. 1982.

[11] U. Erez, S. Litsyn, and R. Zamir, "Lattices which are good for (almost) everything," *IEEE Trans. Inf. Theory*, vol. 51, pp. 3401–3416, Oct. 2005.

[12] S.-C. Chang and E. Weldon, "Coding for T-user multiple-access channels," *IEEE Trans. Inf. Theory*, vol. IT-25, pp. 684–691, Nov. 1979.