Information Theoretic Bounds for Distributed Computation

by

Ola Ayaso

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2008

Author .. ..
Department of Electrical Engineering and Computer Science
May 2008

Certified by ....
Munther A. Dahleh
Professor
Thesis Supervisor

Certified by.. ....
Devavrat Shah
Assistant Professor
Thesis Supervisor

Accepted by ....
Terry P. Orlando
Chairman, Department Committee on Graduate Students

Information Theoretic Bounds for Distributed Computation
by
Ola Ayaso

Submitted to the Department of Electrical Engineering and Computer Science
on May 2008, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

In this thesis, I explore via two formulations the impact of communication constraints on distributed computation. In both formulations, nodes make partial observations of an underlying source. They communicate in order to compute a given function of all the measurements in the network, to within a desired level of error. Such computation in networks arises in various contexts, like wireless and sensor networks, consensus and belief propagation with bit constraints, and estimation of a slowly evolving process. By utilizing Information Theoretic formulations and tools, I obtain code- or algorithm-independent lower bounds that capture fundamental limits imposed by the communication network.

In the first formulation, each node samples a component of a source whose values belong to a field of order q. The nodes utilize their knowledge of the joint probability mass function of the components together with the function to be computed to efficiently compress their messages, which are then broadcast. The question is: how many bits per sample are necessary and sufficient for each node to broadcast in order for the probability of decoding error to approach zero as the number of samples grows.

I find that when there are two nodes in the network seeking to compute the sample-wise modulo-q sum of their measurements, a node compressing so that the other can compute the modulo-q sum is no more efficient than its compressing so that the actual data sequence is decoded. However, when there are more than two nodes, we demonstrate that there exists a joint probability mass function for which nodes can more efficiently compress so that the modulo-q sum is decoded with probability of error asymptotically approaching zero. It is both necessary and sufficient for nodes to send a smaller number of bits per sample than they would have to in order for all nodes to acquire all the data sequences in the network.

In the second formulation, each node has an initial real-valued measurement. Nodes communicate their values via a network with fixed topology and noisy channels between nodes that are linked. The goal is for each node to estimate a given function of all the initial values in the network, so that the mean square error in the estimate is within a prescribed interval. Here, the nodes do not know the distribution of the source, but have unlimited computation power to run whatever algorithm needed to ensure the mean square error criterion. The question is: how does the communication network impact the time until the performance criterion is guaranteed.

Using Information Theoretic inequalities, I derive an algorithm-independent lower bound on the computation time. The bound is a function of the uncertainty in the function to be estimated, via its differential entropy, and the desired accuracy level, as specified by the mean square error criterion. Next, I demonstrate the use of this bound in a scenario where nodes communicate through erasure channels to learn a linear function of all the node's initial values. For this scenario, I describe an algorithm whose running time, until with high probability all nodes' estimates lie within a prescribed interval of the true value, is reciprocally related to the "conductance." Conductance quantifies the information flow "bottle-neck" in the network and hence captures the effect of the topology and capacities. Using the lower bound, I show that the running time of any algorithm that guarantees the aforementioned probability criterion, must scale reciprocally with conductance. Thus, the lower bound is tight in capturing the effect of network topology via conductance; conversely, the running time of our algorithm is optimal with respect to its dependence on conductance.

Thesis Supervisor: Munther A. Dahleh
Title: Professor

Thesis Supervisor: Devavrat Shah
Title: Assistant Professor

Acknowledgments

I am indebted to many many people. But here, I will acknowledge those most directly linked to my experience at MIT. I am thankful to:

My advisors, Professor Munther Dahleh and Professor Devavrat Shah, for the privilege of learning from them. To Professor Dahleh, I am deeply grateful for much more than I can mention here. Everything that I have been able to accomplish is due to him. I am especially thankful for his vital support through out the many years, particularly at critical moments. Two such moments influenced my experience most dramatically and positively. These are (1) the seamless restart in a new research area in April of 2005, and, (2) the connection made with Devavrat.

My interaction with Devavrat has been extremely rewarding. I am convinced that it is thanks to Devavrat that my thesis has progressed as fast and as positively as it has. Not only have I learned much, but I have enjoyed the process immensely. It has been an honor to work with a Rising Star.

Professor Nuno Martins, for giving this thesis life. Not only am I grateful to him for giving me a good start in this research area, but I am also grateful for his crucial advice and encouragement all along the way. In my opinion, another rising star, I am lucky to call a friend.

My committee member, Professor Asuman Ozdaglar, for her interest and kind words of encouragement.

My mentor and friend from the AUB days, Professor Nassir Sabah, and his family, Mrs. Gharid Sabah and Maysa. They took good care of me during my first year in Boston, which coincided with their stay here.

My uncle Souheil Al-Jadda and his family, Aunt Sahar, Souheila, Adel and Omar, for making me feel at home, especially during the first year.

My dear friends, the Jabri's, Aunti Norma and Ammo Salman, Zouhair, Fadi and Farah, for being my family on this continent.

My excellent friends, for their friendship, let alone the support, encouragement, and help offered so generously even when I didn't know that I needed it. I cannot begin to describe how vital a role each has played at some moment, so I will just list their names in alphabetical order. Mukul Agarwal, Mayssam Ali and Waleed Farahat, Monica and Fawaz Al-Malood, Saeed Arida, Chadi El-Chemali, Wassim El-Solh, Husni Idris, Hisham Kassab, Dina Katabi, Hesham Khalfan, Georgios Kotsalis, Pat Kreidl, Sriram Krishnan, Yao Li and Kazutaka Takahashi, Nuno Martins, Basel and Tareq Naffouri, Raj Rao, Sarah Saleh, Sridevi Sarma, Rehan Tahir, Dora Tzianetopoulou.

My immediate and extended academic family, I am proud to be part of the same family. Of those I have not mentioned before, the family includes: Soosan Beheshti, Iahn Cajigas, Jorge Goncalves, Sleiman Itani, Fadi Karameh, Yola Katsargyri, Jerome Le Ny, Paul Njoroge, Mesrob Ohannessian, Mike Rinehart, Keith Santarelli, Parikshit Shah, Sean Warnick.

My caring friend Fifa Monserrate, her presence made LIDS a warm and welcoming place.

My aunts and uncles, the Ayaso's and Al-Jadda's, and their families. I know that they have kept me in their thoughts and prayers.

My parents, Anan and Othman, and sisters, Rasha, Noha and Raghad, for joining me on this ride, even when they might have wanted to get off!

Finally, to those whom it has slipped my mind to mention, I beg your pardon and thank you for your understanding!

Contents

List of Figures

Chapter 1

Introduction

We investigate the impact of communication constraints on distributed computation. Each node or agent in a network has partial information, a measurement of a component of an underlying source. All nodes need to estimate a desired function of all the measurements in the network. To do so, they must communicate; but, the communication is imperfect. In this thesis, we consider two types of communication constraints. The first consists of nodes communicating compressed messages, broadcast over noiseless channels. The second consists of nodes communicating via a fixed topology and noisy channels. The nodes are connected according to some fixed pattern, so nodes do not communicate directly with all other nodes, but whenever two nodes are linked, they communicate through noisy channels.

In the first formulation we consider, each node makes repeated measurements, or samples, of the component it can access. Each node compresses the data stream it has gathered so that all other nodes can reliably decode a given function of the data in the network. Each node knows the joint probability distribution of all the components, which take values in a discrete set. Nodes must take advantage of this in order to efficiently compress their messages. The performance requirement is that the probability of error at any of the nodes approaches zero as the number of samples increases. Our goal is to characterize the necessary and sufficient compression rates, or bits communicated per sample, for reliable computation.

In the second formulation, each node has an initial real-valued message. They communicate via a network with a fixed topology, where nodes that are linked communicate with each other over noisy channels. The goal is to compute a given function of all the data in the network. In this case, nodes have no information about the distribution of the initial messages. The performance requirement is that the mean square error in the nodes' estimates of the desired function is within a desired range. Our goal is to characterize, as a function of the communication constraints, the time at which the performance criterion can be guaranteed.

Our approach is to use Information Theoretic analysis, to which the formulations naturally lend themselves. This type of analysis leads to "algorithm-independent" lower bounds; bounds that must hold regardless of the the communication (encoding/decoding) scheme used by the nodes. For the majority of this work, we make no assumptions on the computation that the nodes perform. Hence, our lower bounds also hold regardless of the computation scheme used by the nodes. We assume that nodes have no computational limitations, like limited memory or power. This assumption enables us to capture the limitations that arise exclusively due to the communication constraints. However, our Information Theoretic techniques do apply to cases where there are computational restrictions, as we show via simple examples. Moreover, the techniques provide tighter bounds in the presence of computational limitations.

1.1 Motivation

Distributed computation in the presence of communication constraints arises is many practical applications. Distributed computation algorithms are likely be core of future distributed estimation algorithms for slowly evolving processes. In wireless or sensor networks, the computation that is demanded of nodes is simple, so the performance of the nodes is primarily limited by the communication constraints. On the other hand, recently popular algorithmic approaches based on belief propagation, in hardware for decoding, or consensus, in a multi-agent system, are examples of distributed computation with stringent computation constraints as well as bit constraints.

We investigate algorithm-independent lower bounds for such scenarios. These are bounds that hold regardless of the scheme that nodes use to communicate. As such, the bounds capture the effect of the physical communication constraints on the performance of the computation algorithm.

In the case of our first formulation, nodes compress their data for reliable computation. One way to achieve the goal of decoding the function they desire is for nodes to communicate to each other their entire data streams, as in the traditional Information Theoretic formulation. Then, each node can compute the function it desires. However, nodes may be able to send less bits per sample, if they compress with the goal of computation, as opposed to communication of the entire data. The ability to compress more efficiently depends on the function that is desired. Compression for computation may be critical in situations where nodes have limited transmission power.

In the case of our second formulation, nodes communicate via noisy channels. The time it takes for the mean square error performance criterion to be guaranteed will depend on the connectivity of the nodes and the magnitudes of the channels between them. It also depends on the desired function and the accuracy, as quantified by the mean square error. The algorithm-independent lower bound captures these dependencies precisely. In addition, it allows for the determination of the optimality of algorithms. Any algorithm that has a convergence time that matches the lower bound is optimal; it is the fastest possible.

1.2 Background

The topic of our interest is related to several areas of previous work. We broadly group the related literature into three groups: Distributed Computation, Information Theory, and Communication. Our investigation lies in the intersection of these areas. Our formulation is amenable to the techniques of Network Information Theory and Rate Distortion Theory. Furthermore, our goal of obtaining lower bounds is compatible with these techniques. But, our context, distributed computation, is more in line with the Distributed Computation and Communication literature. We briefly discuss how our work compares and contrasts with some of the work in these and other areas.

Our context of distributed computation is one that is encountered in the Distributed Computation literature, the Communication literature, and the Complexity Theory literature. By the "Distributed Computation" literature, we mean work where a model of computation is assumed for the nodes at the outset. An example is the early work of Tsitsiklis [33, 32, 18], which analyzes algorithms that perform specific computations in distributed settings. For example, in [32], the goal is for the nodes to estimate the minimum of some function. Each node updates its estimate by computing an affine function of its current estimate, a certain local measurement, and the estimates received from its neighbors.

Recently, there has been a renewed interest in distributed algorithms for computation, for example [3, 24]. Many have revisited one of the problems considered by Tsitsiklis, the so-called "consensus" problem, for example [15, 23, 26]. However, in all these cases, whenever nodes communicate with each other, they exchange real-valued messages. As such, there are no bit constraints.

Very recent related work, however, does indeed explicitly model communication constraints. For example, in [17], Kreidl presents a message-passing algorithm for inference when nodes communicate via discrete memoryless channels. In [31], the authors consider hypothesis testing in a network of nodes connected in a tree configuration. The root, the fusion center, is to make the decision. Again, the nodes are assumed to have limited communication capabilities. For example, they use quantized messages to communicate their observations to the fusion center. The authors analyse the rate of decay of the probability of error as the number of nodes in the network approaches infinity.

By the "Communication" literature, we mean work that explicitly models the communication constraints, but does not assume before hand a computation scheme at the nodes. Suppose that the goal is for nodes to compute a particular function. The typical result would be to provide an algorithm that can be used in the assumed communication network. The algorithm must perform the desired computation while satisfying some desired performance criterion, such as a probability of error condition. The analysis of the algorithm will result in algorithm-specific bounds, for example, on the number of bits that must be communicated, or the transmission power that is needed.

Examples of such work along these lines are [11, 36]. In [11], each node has one bit, that it can broadcast to all other nodes via binary symmetric channels. A dedicated node, a fusion center, needs to compute the modulo-2 sum of all the bits, so that the probability of error is minimized. Gallager describes a communication scheme that can be used in this scenario. He provides an upper bound on the number of bits that need to be exchanged in the network in terms of the number of nodes in the network. Recently, in [14], this bound has been shown to be tight. The authors produce an algorithm-independent lower bound that matches, up to a constant, the upper bound.

In the Complexity Theory literature, function computation is again the objective of nodes. However, the focus is characterizing the difficulty, for example in terms of number of operations or bits needed, in exactly computing the desired function. The thrust is to capture the impact on algorithm performance due to computational requirements, and not communication limitations. The models that are used for communication are not probabilistic, and hence do not allow for formulations where there are noisy channels. For example, see [38, 10, 27].

Probabilistic communication models, "channels," are the main feature in the Information Theory domain. So, this theory naturally provides tools for analyzing formulations that include such communication constraints. However, the objective in classical Information Theory is reliable communication, as opposed to computation. That is, decoders need to estimate the sent messages, rather than a function of them. Much work is done for the setting of a network of nodes. This body of work constitutes an area known as Network Information Theory; for an overview, see [9, 5].

However, the Information Theory literature does contain instances where function computation has been investigated. For example, in [1, 28], an encoder compresses a message and sends the information noiselessly to a decoder that has side information. The decoder must produce an estimate of a given function of the message and the side information. The objective is for the probability of error to approach zero asymptotically as the size of the message grows. This set-up corresponds to a two node case. In Chapters 2 and 3, we consider coding for reliable computation for multiple nodes.

There is another area in Information Theory that deals with nodes computing a function of the data in the network. This is the area of secure communications, for example [7]. Nodes need to generate *some* function of their data. This would then be used as a secret key that is used to encrypt messages. The difference between this work and ours is that here, the function is not known to the nodes *a priori*. Nodes exploit the common randomness in their messages to each generate the same function of the messages in the network.

Finally, our work on the problem of "distributed computation with communication constraints" has been influenced by work done in the area of "control with communication constraints." Such work includes [30, 19, 20, 22, 21]. In particular, Martins' application of Information Theoretic notions, such as mutual information, in a feedback control setting, has led to novel results capturing Bode-like fundamental limits on performance. In that area, Information Theoretic tools have been effective in the analysis of systems.

1.3 Overview of Results

We briefly describe the two main contributions of this thesis, two instances in which we have used Information Theoretic formulations or tools to derive lower bounds for function computation, together with scenarios for which we have shown that our bounds are tight. In the first case, discussed in Chapters 2 and 3, the nodes communicate compressed messages via noiseless channels, so bounds are derived on nodes' compression rates. These bounds are tight when the source has a specific joint probability distribution and nodes compute the sample-wise modulo-q sum of their data. In the second case, discussed in Chapters 4 and 5, nodes communicate via noisy channels. We derive bounds on computation time, the time at which nodes' estimates of the desired function satisfy the performance criterion. This bound is tight when nodes communicate via erasure channels with the objective of computing the sum of their initial values.

1.3.1 Computation via Compressed Data

We consider a network of n nodes. Each node makes repeated measurements of a component of a source. It compresses its message and broadcasts it, over a noiseless channel, to all other nodes. Ultimately each node is to compute, with high reliability, a function K of all the measured data of the network. By high reliability, we mean that the probability, that any of the nodes makes a decoding error, approaches zero asymptotically with the number of samples of data.

We present a Network Information Theoretic formulation for this context of distributed computation. We characterize the compression rates that are necessary for reliable computation. Next we consider the achievability of these necessary rates, that is, the existence of codes with rates that are arbitrarily close to the necessary rates.

When the nodes seek to compute the modulo-q sum of all the measured data in the network, we provide an example (specifically, a joint probability mass function for the source) for which we provide an explicit characterization of the "K-rate region", the rates necessary and sufficient for distributed computation of the function K. For this example, when there are two nodes in the network, the K-rate region coincides with the Slepian-Wolf rate region. However, for more than 2 nodes, compressing for computation can result in lower rates than compressing, as in the Slepian-Wolf formulation, for the entire data sequences in the network. We quantify the savings in rate that are obtained, when compressing for K in comparison to the Slepian-Wolf rates, as the number of nodes in the network grows.

1.3.2 Computation via Noisy Channels

We study a network of n nodes communicating over noisy channels. Each node has an initial value. The objective of each of the nodes is to compute a given function of the initial values in the network. We derive a lower bound to the time at which the mean square error in the nodes' estimates is within a prescribed accuracy interval. The lower bound is a function of the channel capacities, the accuracy specified by the mean square error criterion, and the uncertainty in the function that is to be estimated. The bound reveals that, first, the more randomness in the function to be estimated, the larger the lower bound on the computation time. Second, the smaller the mean square error that is tolerated, the larger the lower bound on the computation time. Hence there is a trade-off captured between computation accuracy and computation time. In addition, the lower bound can be used to capture the dependence of the convergence time on the structure of the underlying communication network.

We consider a network of nodes communicating via erasure channels to compute a sum of the initial values in the network. Each of the nodes is required to acquire an estimate that is, with a specified probability, within a desired interval of the true value of the sum. We apply our Information Theoretic technique to derive a lower bound on the computation time for this scenario. We show that the computation time is inversely related to a property of the network called "conductance." It captures the effect of both the topology and channel capacities by quantifying the bottle-neck of information flow. Next, we describe an algorithm that can be used in this setting of nodes computing a sum via erasure channels, and guarantees that with the specified probability, each of the nodes' estimate is within the desired interval. We determine an upper bound on the algorithm's computation time and show that it too is inversely related to conductance. Hence, we conclude that our lower bound is tight in capturing the effect of the communication network, via conductance. Equivalently, our algorithm's run-time is optimal in its dependence on conductance.

Chapter 2

Computation via Compressed Data

We seek to study "distributed computation with communication constraints", and, in particular, compression rates required for communicating "nodes" to compute a given function of all the data in the network with low probability of error. Every encoder is paired with a decoder, and together, they form a "node". A two node network is shown in Figure 2-1. There is a source, (X_1, X_2), with a given probability mass function; node 1 encodes the X_1^N sequences while node 2 encodes the X_2^N sequences. The nodes communicate over a noiseless channel, after which their decoders are required to produce estimates of a given function of the data. In the Slepian-Wolf (SW) set-up, that function is the identity ($\widehat{X_2^N}$ is to be estimated at the node 1 decoder and $\widehat{X_1^N}$ at the node 2 decoder). Both nodes will ultimately become "omniscient"[1], having an estimate of all the data sequences in the network.

Let $X_i^N = \{X_{i1}, X_{i2}, \ldots, X_{iN}\}$ be IID random variables sampled from a source, X_i, with distribution $p(x_i)$. Then, a consequence of Shannon's Asymptotic Equipartition Principle is that for large enough N, $NH(X_i)$ bits are needed on average to encode the sequence X_i^N. Equivalently, the encoder's compression rate, R_i, or average outgoing codeword bits per incoming message digit, must be larger than $H(X_i)$ in order to guarantee low probability of error, $\{\widehat{X_i^N} \neq X_i^N\}$, at the decoder [5].

The Slepian and Wolf (SW) theorem for distributed source coding establishes the rates at which a joint source (X_1, X_2) can be separately encoded by two encoders, one encoding the sequence X_1^N and the other encoding X_2^N, such that a decoder receiving messages of both the encoders can reliably decode the sequence $\{\widehat{X_1^N, X_2^N}\}$ [5, 9]. The encoder of node 1 has access to the X_1 sequence only, while the encoder of node 2 has access to the X_2 sequence only; the decoder, on the other hand, has access to the compressed messages sent noiselessly by both the encoders (hence we call this "centralized decoding"), as shown in Figure 2-2. The decoder also knows the joint distribution of the source.

[1] We borrow this terminology from [7].

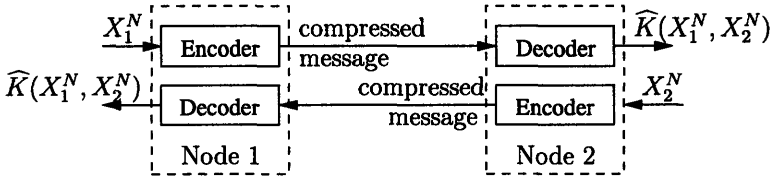

Figure 2-1: Two nodes compressing and communicating for computation of K.

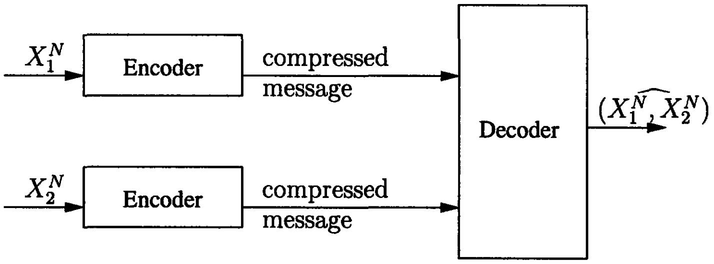

Figure 2-2: Slepian-Wolf distributed source coding with "centralized decoding".

If the encoding was centralized, that is, the encoder knows the (X_1, X_2) sequence, then, on average, $H(X_1, X_2)$ bits per source letter are needed to describe the source. If each encoder was communicating with a separate decoder, $H(X_1)$ (respectively $H(X_2)$) bits would be needed to decode the X_1 (respectively X_2) sources separately. An interesting statement of Slepian-Wolf is that for their set-up, if $H(X_1)$ bits are used to encode X_1, then only $H(X_2|X_1)$ are needed to encode X_2, even though the encoders encode their messages completely separately.

Indeed, their theorem specifies the compression rates at the X_1 and X_2 encoders, R_1 and R_2, that are both needed for and will guarantee low probability of error at the decoder, by the inequalities

$$\begin{aligned} R_1 &\geq H(X_1|X_2) \\ R_2 &\geq H(X_2|X_1) \\ R_1 + R_2 &\geq H(X_1, X_2). \end{aligned} \tag{2.1}$$

These inequalities specify the "achievable" rate pairs (R_1, R_2). Figure 2-4 shows all such rate pairs in the shaded rate region.

Extensions and alternative proofs of the Slepian-Wolf theorem have been suggested. For example, the source samples need not be IID; Cover[4] showed that the theorem holds under certain conditions on the source, such as ergodicity. An interesting proof for the existence of codes with rates that are close to the boundary of the rate region specified by the theorem is proposed by Wyner [34] when the source is binary symmetric. This is shown using results from channel coding for the binary symmetric channel. This idea is going to be used in section 3.1.

The result has also been generalized to networks with multiple sources and networks with various structures [5, 6, 7, 35]. For example, the structure that is of interest in this chapter is one where every encoder is paired with a decoder, and together, they form a "node". A two node network is shown in Figure 2-3. There is a source, (X_1, X_2), with

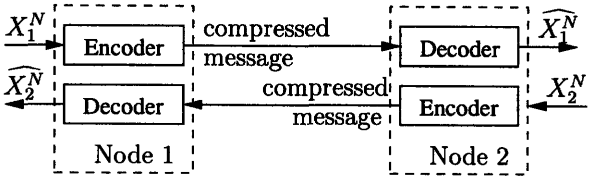

Figure 2-3: Two node SW set up with "decentralized decoding".

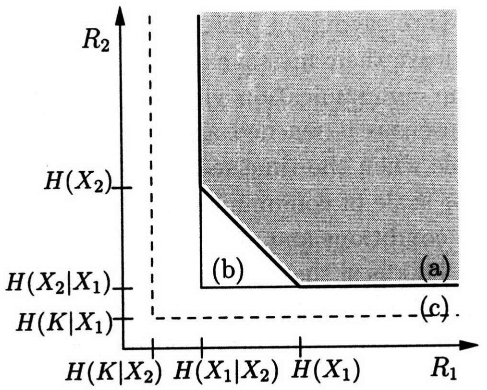

Figure 2-4: The boundaries of the rate regions for two nodes are shown for (a) the SW/omniscience with centralized decoding set-up (thick solid line); (b) the SW/omniscience with decentralized decoding (thin solid line); and (c) compressing for K and decentralized decoding (dashed line). The rate regions are to the upper right of the boundaries; the shaded region, for example, is the SW with centralized decoding rate region.

some joint distribution; node 1 encodes the X_1^N sequences while node 2 encodes the X_2^N sequences independently of node 1. The nodes communicate over a noiseless channel, after which their decoders are required to produce estimates of the other node's sequence $(\widehat{X_2^N}$ at the node 1 decoder and $\widehat{X_1^N}$ at the node 2 decoder). Both nodes will ultimately become "omniscient", having an estimate of all the data sequences in the network. Because decoding occurs separately at every node and no decoder receives compressed messages from all the encoders in the network, we refer to this set-up as a "decentralized decoding" scheme.

The fact that the decoding is "decentralized" affects the shape of the achievable rate region. The ideas used in the Slepian-Wolf theorem proof can be readily applied to this set-up to derive the inequalities that determine the rate region. These are,

$$\begin{aligned} R_1 &\geq H(X_1|X_2) \\ R_2 &\geq H(X_2|X_1) \end{aligned} \tag{2.2}$$

The first inequality ensures that the probability of an error made at the node 2 decoder in determining X_1^N is small. Similarly, the second guarantees small error probability at node 1 when determining X_2^N. Because neither node needs to estimate all of (X_1^N, X_2^N), the last inequality of (2.1) is no longer needed. The boundary of the rate region determined by (2.2) is shown in Figure 2-4.

In this chapter, however, the objective of each of the nodes is to acquire, not necessarily the entire data sequences of all the nodes (omniscience), e.g. $\widehat{(X_1^N, X_2^N)}$, that Slepian-Wolf require, but a given function of that, $\widehat{K}(X_1^N, X_2^N)$. If all the nodes eventually obtain, with low probability of error, estimates of that function then they would have reached a consensus. Clearly, one (uninteresting) way to achieve this consensus is for all nodes to acquire omniscience first, then compute K.

Our formulation is motivated by the problem of estimation, when the sensing is distributed, of a stochastic process that may be generated by a dynamic system. Each source variable represents the measurements collected by one of the nodes. The nodes may communicate their measurements, and, it is desirable to do so after having compressed the outgoing messages (as it leads to savings in power used for transmission, for example). The nodes are assumed to broadcast their messages to all the nodes in the network, which is a situation that may arise in communication via satellites. Even though our formulation (and results) imply that consensus is reached asymptotically, for N sufficiently large, the formulation is still reasonable when the time scale of the evolution of the dynamic system is much slower than the time scale of communication.

In the control literature, conditions and algorithms for communicating nodes to achieve consensus were studied by Tsitsiklis in the 80's [33, 32] and these topics have recently become an active area of research (for example [2, 3, 15, 23, 26]). The idea of achieving consensus on the average of quantities measured and computed by individual nodes is used to suggest a scheme for distributed Kalman filtering and parameter estimation [25, 37]. However, none of this work assumes any communication constraints. By using the Information Theoretic set-up discussed above, we can naturally study the rate requirements that are imposed on the nodes by the need to compress data.

In the next section, we present the formulation of Slepian-Wolf for convenience and to establish notation. In section 2.2, we present the general formulation we use for studying distributed computation when nodes communicate via compressed messages. It is useful for three reasons. First, the inequalities that define the rate region are simple; they involve the conditional entropy rates of the function K, which are easy to compute. Second, the formulation can be easily extended to an arbitrary number of nodes, and hence the rate region can easily be defined for an arbitrary number of nodes. Finally, our set-up inherently involves feedback.

In section 2.2, we also state the main theorem of this chapter, which provides lower bounds on the compression rates necessary for decoding K with probability of error approaching zero asymptotically. We prove this theorem in section 2.3. Next, in section 2.4, we discuss achievability of rates arbitrarily close to the boundary of the rate-region specified by our theorem.

2.1 Slepian-Wolf Source Coding Theorem

In this section, the Slepian-Wolf (SW) theorem is repeated for the three node case, both for convenience and to establish notation, which is then used for the theorem of section 2.2. Two cases are considered. In the first, the no interactive communication case, nodes are not allowed to use their incoming data to generate the outgoing message; that is, the encoded message at a node is a function of that node's information only. In the second case, a node's broadcast message is a function of both its own data and the messages that it has received previously from the other nodes. This is the case of "interactive communication," as it is called in [7]. The SW theorem holds under both cases. The formulation and notation below is taken from [35].

2.1.1 No Interactive Communication

We consider first three sources, (X_1, X_2, X_3), taking values on the finite set $\mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3$, and having some joint probability mass function $p(x_1, x_2, x_3)$. The joint source is sampled and the outcome is represented by the sequence $\{X_{1t}, X_{2t}, X_{3t}\}$ for $t = 1, 2, \ldots$. Each of the samples is drawn independently and is distributed according to $p(x_1, x_2, x_3)$.

There are 3 nodes. Node X_1 has access to the sequence $\{X_{1t}\}$; similarly, X_2 and X_3 have access to $\{X_{2t}\}$ and $\{X_{3t}\}$ respectively. Each node encodes its data, of block-length

N, and broadcasts it to the other two nodes via a noiseless channel. Upon receiving the encoded data from the other nodes, each node attempts to reconstruct, using its decoder, the data sequences that were sent, such that the probability of error is small.

A code having the parameters $(N, M_{x_1}, M_{x_2}, M_{x_3}, P_{e,N})$ consists of the following mappings.

1. Encoder Maps

$$\begin{aligned} e_{x_1} &: \mathcal{X}_1^N \rightarrow \mathcal{I}_{M_{x_1}} \\ e_{x_2} &: \mathcal{X}_2^N \rightarrow \mathcal{I}_{M_{x_2}} \\ e_{x_3} &: \mathcal{X}_3^N \rightarrow \mathcal{I}_{M_{x_3}}, \end{aligned} \tag{2.3}$$

where, $\mathcal{I}_M$ denotes the set $\{1, 2, \ldots M\}$.

The random variables at the outputs of the encoders are denoted by

$$\begin{aligned} W_{x_1} &= e_{x_1}(X_1^N) \\ W_{x_2} &= e_{x_2}(X_2^N) \\ W_{x_3} &= e_{x_3}(X_3^N), \end{aligned} \tag{2.4}$$

where, X_i^N is a random variable that represents the finite sequence $\{X_{i1}, X_{i2}, \ldots, X_{iN}\}$.

2. Decoder Maps

$$\begin{aligned} d_1 &: \mathcal{X}_1^N \times \mathcal{I}_{M_{x_2}} \times \mathcal{I}_{M_{x_3}} \rightarrow \mathcal{X}_2^N \times \mathcal{X}_3^N \\ d_2 &: \mathcal{X}_2^N \times \mathcal{I}_{M_{x_1}} \times \mathcal{I}_{M_{x_3}} \rightarrow \mathcal{X}_1^N \times \mathcal{X}_3^N \\ d_3 &: \mathcal{X}_3^N \times \mathcal{I}_{M_{x_1}} \times \mathcal{I}_{M_{x_2}} \rightarrow \mathcal{X}_1^N \times \mathcal{X}_2^N. \end{aligned} \tag{2.5}$$

The random variables at the outputs of the decoders are denoted by

$$\begin{aligned} (\widehat{X_2^N, X_3^N}) &= d_1(X_1^N, W_{x_2}, W_{x_3}) \\ (\widehat{X_1^N, X_3^N}) &= d_2(X_2^N, W_{x_1}, W_{x_3}) \\ (\widehat{X_1^N, X_2^N}) &= d_3(X_3^N, W_{x_1}, W_{x_2}). \end{aligned} \tag{2.6}$$

The error probability is defined to be $P_{e,N} = \mathbf{P}(E_1 \cup E_2 \cup E_3)$, where,

$$\begin{aligned} E_1 &= \{(\widehat{X_2^N, X_3^N}) \neq (X_2^N, X_3^N)\} \\ E_2 &= \{(\widehat{X_1^N, X_3^N}) \neq (X_1^N, X_3^N)\} \\ E_3 &= \{(\widehat{X_1^N, X_2^N}) \neq (X_1^N, X_2^N)\}. \end{aligned} \tag{2.7}$$

Definition 2.1.1. A rate triple (R_1, R_2, R_3) is *achievable* if for every $\delta > 0$ and all N sufficiently large, there exists a sequence of codes with parameters $(N, M_{x_1}, M_{x_2}, M_{x_3}, P_{e,N})$

such that,

$$\begin{aligned} R_1 + \delta &\geq \frac{1}{N} \log M_{x_1} \\ R_2 + \delta &\geq \frac{1}{N} \log M_{x_2} \\ R_3 + \delta &\geq \frac{1}{N} \log M_{x_3}, \end{aligned} \tag{2.8}$$

and,

$$P_{e,N} \leq \delta. \tag{2.9}$$

The Slepian-Wolf theorem establishes the achievable rate region when every node is required to decode every other node's data sequence. Following [7], we call this requirement "omniscience" and the rate region is the "omniscience rate region."

Theorem 2.1.2. (R_1, R_2, R_3) *is achievable iff*

$$\begin{aligned} R_1 &\geq H(X_1|X_2, X_3) \\ R_2 &\geq H(X_2|X_1, X_3) \\ R_3 &\geq H(X_3|X_1, X_2) \\ R_1 + R_2 &\geq H(X_1, X_2|X_3) \\ R_2 + R_3 &\geq H(X_2, X_3|X_1) \\ R_1 + R_3 &\geq H(X_1, X_3|X_2). \end{aligned} \tag{2.10}$$

2.1.2 With Interactive Communication

When the nodes are each allowed to communicate for several rounds, using the information received in previous rounds to update the outgoing message accordingly, sufficiency of the conditions (2.10), or what is known as the achievability part of the theorem, follows immediately. Since a rate triple specified by (2.10) is achievable without feedback, it must be achievable when feedback is allowed; just set all feedback to be empty and use the code that worked in the no-interactive communication case. To show that the rates in (2.10) are necessary, however, does not follow so immediately. It is not obvious that when feedback is allowed, lower rates are not possible. Although not obvious, by applying standard techniques, including the use of Fano's inequality, it can be shown that the converse part of Theorem 2.1.2 still holds [35, 7].

Below, we restate the definitions that will be changed when feedback is allowed, such as the encoder and decoder mappings. Also, the definition of the achievable rate triple is modified.

1. Encoder maps for the b^{th} round of communication, $b = 1, 2, \ldots, B$.

$$\begin{aligned} W_{x_1,b} &= e_{x_1,b}(X_1^N, W_{x_2,1}^{b-1}, W_{x_3,1}^{b-1}) && \in \mathcal{I}_{M_{x_1,b}} \\ W_{x_2,b} &= e_{x_2,b}(X_2^N, W_{x_1,1}^{b-1}, W_{x_3,1}^{b-1}) && \in \mathcal{I}_{M_{x_2,b}} \\ W_{x_3,b} &= e_{x_3,b}(X_3^N, W_{x_1,1}^{b-1}, W_{x_2,1}^{b-1}) && \in \mathcal{I}_{M_{x_3,b}}, \end{aligned} \tag{2.11}$$

where, $W_{x_i,1}^{b-1} = \{W_{x_i,1}, W_{x_i,2}, \ldots, W_{x_i,b-1}\}$ is the sequence of previous messages sent by the node X_i encoder. $\mathcal{I}_M$ denotes the set $\{1, 2, \ldots M\}$.

2. Decoder maps applied after B rounds of communication.

$$\begin{aligned}\widehat{(X_2^N, X_3^N)} &= d_1(X_1^N, W_{x2,1}^B, W_{x3,1}^B)\\ \widehat{(X_1^N, X_3^N)} &= d_2(X_2^N, W_{x1,1}^B, W_{x3,1}^B)\\ \widehat{(X_1^N, X_2^N)} &= d_3(X_3^N, W_{x1,1}^B, W_{x2,1}^B). \end{aligned} \tag{2.12}$$

Definition 2.1.3. When interactive communication is allowed, a rate triple (R_1, R_2, R_3) is *achievable* if for every $\delta > 0$ and all N sufficiently large, there exists a sequence of codes with parameters $(N, \{M_{x1,1}, \ldots, M_{x1,B}\}, \{M_{x2,1}, \ldots, M_{x2,B}\}, \{M_{x3,1}, \ldots, M_{x3,B}\}, P_{e,N})$ such that,

$$\begin{aligned}R_1 + \delta &\geq \frac{1}{N} \sum_{b=1,\ldots,B} \log M_{x1,b}\\ R_2 + \delta &\geq \frac{1}{N} \sum_{b=1,\ldots,B} \log M_{x2,b}\\ R_3 + \delta &\geq \frac{1}{N} \sum_{b=1,\ldots,B} \log M_{x3,b}, \end{aligned} \tag{2.13}$$

and, $P_{e,N} \leq \delta$.

2.2 Compression for Computation

We let $K(X_1^N, X_2^N, X_3^N)$ be a function that is known to each of the nodes (X_i^N is a random variable that represents the finite sequence $\{X_{i1}, X_{i2}, \ldots, X_{iN}\}$). More specifically, the function is denoted as $K^N = \{K_1, K_2, K_3, \ldots, K_N\}$ where $K_t = f_t(X_1^t, X_2^t, X_3^t)$, and the f_t are known to all the nodes. Further, let $\mathcal{K}_t$ be the range of f_t, then, K^N takes values from the set $\mathcal{K}_1 \times \mathcal{K}_2 \times \cdots \times \mathcal{K}_N$. The cardinality of this set is $\kappa(N) = |\mathcal{K}_1 \times \mathcal{K}_2 \times \cdots \times \mathcal{K}_N| = |\mathcal{K}_1||\mathcal{K}_2| \ldots |\mathcal{K}_N|$. We assume that $\lim_{N\to\infty} \frac{1}{N} \log \kappa(N) = c$, where c is a constant.

Suppose that interactive communication is allowed, so the encoder maps (2.11) and achievability definition 2.1.3 from above still apply. The decoder maps will now be defined as,

$$\begin{aligned}\widehat{K_{x1}^N} &= d_1(X_1^N, W_{x2,1}^B, W_{x3,1}^B)\\ \widehat{K_{x2}^N} &= d_2(X_2^N, W_{x1,1}^B, W_{x3,1}^B)\\ \widehat{K_{x3}^N} &= d_3(X_3^N, W_{x1,1}^B, W_{x2,1}^B). \end{aligned} \tag{2.14}$$

The error probability is defined as, $P_{e,N} = \mathbf{P}(E_1 \cup E_2 \cup E_3)$, where,

$$\begin{aligned}E_1 &= \{\widehat{K_{x1}^N} \neq K(X_1^N, X_2^N, X_3^N)\}\\ E_2 &= \{\widehat{K_{x2}^N} \neq K(X_1^N, X_2^N, X_3^N)\}\\ E_3 &= \{\widehat{K_{x3}^N} \neq K(X_1^N, X_2^N, X_3^N)\}. \end{aligned} \tag{2.15}$$

Now, an obvious lower bound of the K-rate region boundary is given in the following theorem. A proof is written out below; see also [28].

Theorem 2.2.1. *(Converse) (R_1, R_2, R_3) is achievable only if*

$$\begin{aligned} R_1 &\geq H(K|X_2, X_3) \\ R_2 &\geq H(K|X_1, X_3) \\ R_3 &\geq H(K|X_1, X_2) \\ R_1 + R_2 &\geq H(K|X_3) \\ R_2 + R_3 &\geq H(K|X_1) \\ R_1 + R_3 &\geq H(K|X_2), \end{aligned} \tag{2.16}$$

where,

$$\begin{aligned} H(K|X_i, X_j) &\overset{\text{def}}{=} \lim_{N\to\infty} \frac{1}{N} H(K^N|X_i^N, X_j^N) \\ H(K|X_i) &\overset{\text{def}}{=} \lim_{N\to\infty} \frac{1}{N} H(K^N|X_i^N). \end{aligned} \tag{2.17}$$

Note that K_t is allowed to be a function of all previous (X_1, X_2, X_3) samples. However, if it is a function of only the current samples and if $f_t = f$, that is $K_t = f(X_{1t}, X_{2t}, X_{3t})$, then, $H(K|X_i, X_j) = H(f(X_1, X_2, X_3)|X_i, X_j)$ and $H(K|X_i) = H(f(X_1, X_2, X_3)|X_i)$. Also, note that because $H(f(X)) \leq H(X)$, it can be seen easily that rate region specified in (2.16), which we call the "K-achievability" rate region or "K-rate region" must contain the SW rate region for omniscience. In section 2.4, more is said about the achievability of rates outside the SW region. First, a proof of the converse is presented.

2.3 Proof of Converse

The converse states that if there exists a sequence of codes with rates (R_1, R_2, R_3) and for which $P_{e,N} \to 0$, then these rates must be in the rate region specified by (2.16). The converse can be shown using the same techniques used in the proof of the converse part of the Slepian-Wolf Theorem, namely, application of Fano's inequality. The proof below closely follows the proof in [35].

Proof. We begin with Fano's inequality. Recall that $E_1 = \{\widehat{K^N_{x_1}} \neq K({X_1}^N, {X_2}^N, {X_3}^N)\}$. Now,

$$\begin{aligned} H(K^N|{X_1}^N, W^B_{x_2,1}, W^B_{x_3,1}) &\leq H(K^N|\widehat{K^N_{x_1}}) \\ &\leq h(\mathbf{P}(E_1)) + \mathbf{P}(E_1) \log \kappa(N), \end{aligned}$$

where, $h(p) = -p \log p - (1-p)\log(1-p)$.

But $h(\mathbf{P}(E_1)) \leq \log 2$ and $\mathbf{P}(E_1) \leq \mathbf{P}(E_1 \cup E_2 \cup E_3) = P_{e,N}$. So,

$$H(K^N|{X_1}^N, W^B_{x_2,1}, W^B_{x_3,1}) \leq N\Delta(P_{e,N}, N), \tag{2.18}$$

where $\Delta(P_{e,N}, N) = \frac{\log 2}{N} + P_{e,N} \frac{\log \kappa(N)}{N}$. Now,

$$\sum_{b=1,\ldots,B} \log M_{x_1,b} \geq \sum_{b=1,\ldots,B} H(W_{x_1,b}) \geq H(W^B_{x_1,1}).$$

And,

$$\begin{aligned} H(W^B_{x1,1}) &\geq H(W^B_{x1,1}|X_2{}^N, X_3{}^N) \\ &\overset{(a)}{\geq} I(W^B_{x1,1}; K^N|X_2{}^N, X_3{}^N) \\ &= H(K^N|X_2{}^N, X_3{}^N) - H(K^N|W^B_{x1,1}, X_2{}^N, X_3{}^N), \end{aligned}$$

where (a) holds because

$$I(W^B_{x1,1}; K^N|X_2{}^N, X_3{}^N) = H(W^B_{x1,1}|X_2{}^N, X_3{}^N) - H(W^B_{x1,1}|K^N, X_2{}^N, X_3{}^N)$$

and $H(W^B_{x1,1}|K^N, X_2{}^N, X_3{}^N) \geq 0$.

Now,

$$\begin{aligned} H(K^N|W^B_{x1,1}, X_2{}^N, X_3{}^N) &\overset{(a)}{=} H(K^N|W^B_{x1,1}, X_2{}^N, X_3{}^N, W^B_{x2,1}, W^B_{x3,1}) \\ &\leq H(K^N|X_2{}^N, W^B_{x1,1}, W^B_{x3,1}) \\ &\overset{(b)}{\leq} N\Delta(P_{e,N}, N), \end{aligned}$$

where (a) holds because of (2.11), specifically, $W^B_{x2,1}$ and $W^B_{x3,1}$ are functions of $(W^B_{x1,1}, X_2{}^N, X_3{}^N)$, and, (b) follows from Fano's inequality as did equation 2.18. Putting the above inequalities together, we have that

$$\sum_{b=1,\ldots,B} \log M_{x_1,b} \geq H(K^N|X_2{}^N, X_3{}^N) - N\Delta(P_{e,N}, N).$$

By assumption, we have that for every $\delta > 0$ and N large enough,

$$R_1 + \delta \geq \frac{1}{N} \sum_{b=1,\ldots,B} \log M_{x_1,b},$$

so,

$$R_1 + \delta \geq \lim_{N\to\infty} \left(\frac{1}{N} H(K^N|X_2{}^N, X_3{}^N) - \Delta(P_{e,N}, N) \right).$$

Now since, by assumption, $\lim_{N\to\infty} \frac{1}{N} \log \kappa(N) = c$, we have $\Delta(P_{e,N}, N) \to 0$ as $N \to \infty$ and $\delta \to 0$; hence,

$$R_1 \geq \lim_{N\to\infty} \frac{1}{N} H(K^N|X_2{}^N, X_3{}^N).$$

Similarly,

$$\begin{aligned} \sum_{b=1,\ldots,B} \log M_{x_2,b} M_{x_3,b} &\geq \sum_{b=1,\ldots,B} H(W_{x_2,b}, W_{x_3,b}) \\ &\geq H(W^B_{x_2,1}, W^B_{x_3,1}). \end{aligned}$$

And,

$$\begin{aligned} H(W^B_{x_2,1}, W^B_{x_3,1}) &\geq H(W^B_{x_2,1}, W^B_{x_3,1} | {X_1}^N) \\ &\geq I(W^B_{x_2,1}, W^B_{x_3,1}; K^N | {X_1}^N) \\ &= H(K^N | {X_1}^N) - H(K^N | {X_1}^N, W^B_{x_2,1}, W^B_{x_3,1}) \\ &\geq H(K^N | {X_1}^N) - N\Delta(P_{e,N}, N). \end{aligned}$$

Now,

$$\begin{aligned} R_2 + R_3 + 2\delta &\geq \frac{1}{N} \sum_{b=1,\ldots,B} \log M_{x_2,b} \log M_{x_3,b} \\ &\geq \frac{1}{N} H(K^N | {X_1}^N) - \Delta(P_{e,N}, N). \end{aligned}$$

So, as $N \to \infty$ and $\delta \to 0$,

$$R_2 + R_3 \geq \lim_{N \to \infty} \frac{1}{N} H(K^N | {X_1}^N).$$

Necessity of the remaining conditions (2.16) in Theorem 2.2.1 follows in the same way. □

2.4 Achievability of Rates Outside the SW Rate Region

To show that the inequalities (2.10) of Theorem 2.1.2 and (2.16) of Theorem 2.2.1 are also sufficient conditions for the achievability of a rate triple, it must be shown that there exist codes with rates that are arbitrarily close to the boundary of the rate regions described by equations (2.10) and (2.16) respectively, for which $P_{e,N} \to 0$. In the proof of the Slepian-Wolf Theorem [5], a "random binning" encoding scheme is used whereby each encoder randomly assigns a codeword/index (for example from 1 to M_{x_1} at node 1) to each of its messages (${X_1}^N$ at node 1). Decoding occurs by joint typicality. However, in the case where decoding of $K({X_1}^N, {X_2}^N, {X_3}^N)$ is required, and in the absence of feedback, it is not clear how this encoding scheme can be used as the nodes do not know what value K takes until after the nodes exchange some information.

In this section, we seek to understand the scenarios for which it is advantageous to use compression for reliable computation, that is receiving nodes decode a function of the nodes data streams, $K({X_1}^N, {X_2}^N, {X_3}^N)$, versus compression for reliable communication, that is receiving nodes decode the entire data streams, $({X_1}^N, {X_2}^N, {X_3}^N)$, as in the Slepian-Wolf formulation. Recall that the K-rate region contains the SW-rate region. So, when there is an achievable rate vector that belongs to the K-rate region, but does not belong to the SW-rate region, then, compressing for K is advantageous because less bits per sample need to be communicated. We discuss this via two specific cases.

In section 2.4.1, we show that when one node is selected amongst all the nodes to act as a fusion center, the random binning argument can be used to show that there are codes with rates that do not belong to the omniscience rate region. Indeed, these rates may even be on the boundary of the K-rate region. However, in section 2.4.2 we observe that in the two node case, the omniscience rate region and the K-rate region coincide for a class of functions, K. So, nodes may as well compress their messages as if receiving nodes were to decode the entire data streams, and not a function of the data streams. In the next chapter, however, we will see that as nodes are added, this may no longer remain the case.

2.4.1 A Fusion Center Scheme

Suppose there are two nodes, corresponding to X_1 and X_2, and the X_2 node is designated the fusion center. Then, the X_1 node uses random binning encoding to send its ${X_1}^N$ sequence to the X_2 node. The X_2 node will be able to decode X_1's message with arbitrarily small probability of error, $\mathbf{P}(\widehat{{X_1}^N} \neq {X_1}^N)$, for sufficiently large N if $R_1 \geq H(X_1|X_2)$. Now, the X_2 node can use the $\widehat{{X_1}^N}$ it has decoded to compute $K(\widehat{{X_1}^N}, {X_2}^N)$, which it then sends back to the X_1 node. The X_2 node can use random binning to encode its message by randomly partitioning the range of K, $\mathcal{K}_1 \times \mathcal{K}_2 \times \cdots \times \mathcal{K}_N$. Now, the X_1 decoder uses joint typicality to decode X_2's message, where it chooses the $\widehat{K^N}$ such that

1. $\widehat{K^N}$ is associated with the index that was received from the X_2 node, and,

2. $({x_1}^N, \widehat{k^N}) \in A_\epsilon^N(K, X_1)$, where $A_\epsilon^N(K, X_1)$ is the restriction of $A_\epsilon^N(K, X_1, X_2)$ to (K, X_1) and $A_\epsilon^N(K, X_1, X_2)$ is defined as (see [5] page 384),

$$
\begin{aligned}
A_\epsilon^N(K, X_1, X_2) = \Bigg\{ & ({x_1}^N, {x_2}^N, k^N) : \\
& \left| -\frac{1}{N} \log p({x_1}^N, {x_2}^N) - H(X_1, X_2) \right| < \epsilon \\
& \left| -\frac{1}{N} \log p({x_1}^N, k^N) - H(X_1, K) \right| < \epsilon \\
& \left| -\frac{1}{N} \log p({x_2}^N, k^N) - H(X_2, K) \right| < \epsilon \\
& \left| -\frac{1}{N} \log p({x_1}^N) - H(X_1) \right| < \epsilon \\
& \left| -\frac{1}{N} \log p({x_2}^N) - H(X_2) \right| < \epsilon \\
& \left| -\frac{1}{N} \log p(k^N) - H(K) \right| < \epsilon \Bigg\}.
\end{aligned}
$$

Node 1 will be able to decode node 2's message with arbitrarily small probability of error, $\mathbf{P}(\widehat{K^N} \neq K(\widehat{{X_1}^N}, {X_2}^N))$, for sufficiently large N if $R_2 \geq H(K|X_1)$. Note that the point $(R_1, R_2) = (H(X_1|X_2), H(K|X_1))$ is on the boundary of the rate region specified by the equivalent of (2.16) for the two node case, namely, $R_1 \geq H(K|X_2)$ and $R_2 \geq H(K|X_1)$. Further, by choosing node 1 to be the fusion center, the point $(H(K|X_2), H(X_2|X_1))$ can be achieved. However, it is not obvious how an arbitrary point on the boundary may be achieved with this scheme.

This idea can be extended to many nodes/sources. For instance, suppose there are three nodes, X_1, X_2, X_3. If node 1 is chosen to be the fusion center, then nodes 2 and 3 send ${X_2}^N$ and ${X_3}^N$ respectively to node 1 which subsequently decodes $(\widehat{{X_2}^N, {X_3}^N})$.

$\mathbf{P}(\widehat{(X_2{}^N, X_3{}^N)} \neq (X_2{}^N, X_3{}^N)) < 2\epsilon$ is guaranteed at node 1 when

$$\begin{aligned} R_2 &\geq H(X_2|X_3, X_1) + 3\epsilon \\ R_3 &\geq H(X_3|X_1, X_2) + 3\epsilon \\ R_2 + R_3 &\geq H(X_2, X_3|X_1) + 3\epsilon \end{aligned}$$

Now, X_1 computes K and broadcasts its value to the X_2 and X_3 nodes. Decoding at X_2 (X_3) occurs by choosing the $\widehat{K^N}$ for which $(x_2{}^N, \widehat{k^N}) \in A_\epsilon^N(K, X_2)$ $((x_3{}^N, \widehat{k^N}) \in A_\epsilon^N(K, X_3))$ where $A_\epsilon^N(K, X_2)$ $(A_\epsilon^N(K, X_3))$ is the restriction of $A_\epsilon^N(K, X_1, X_2, X_3)$ to (K, X_2) (respectively, (K, X_3)). Now, $\mathbf{P}(\widehat{K_{x_2}^N} \neq K^N) \to 0$ at node 2 if $R_1 \geq H(K|X_2)$ and $\mathbf{P}(\widehat{K_{x_3}^N} \neq K^N) \to 0$ at node 3 if $R_1 \geq H(K|X_3)$.

So, under the fusion center scheme, we have that (R_1, R_2, R_3) is achievable if

$$\begin{aligned} R_1 &\geq \max\{H(K|X_2), H(K|X_3)\} \\ R_2 &\geq H(X_2|X_3, X_1) \\ R_3 &\geq H(X_3|X_1, X_2) \\ R_2 + R_3 &\geq H(X_2, X_3|X_1). \end{aligned}$$

Finally, we note that these conditions are similar to those derived in [35]. There, three nodes broadcast their data sequences to each other via a satellite. In our case, node 1 can be thought of as the satellite. Our set-up is a somewhat more straight-forward application of SW-like ideas, however, because in [35], the satellite is not required to be able to decode the data sequences of the communicating nodes.

Optimal Choice of Fusion Center

Once a node is designated as a fusion center, the rate region, $\mathcal{R}$, for the achievable rate vectors, $\mathbf{R} \in \mathbb{R}^n$ for n nodes, can be determined. Given that rate region, one can choose the rate vector that minimizes a certain function, $u(\mathbf{R})$, depending on the application, subject to $\mathbf{R} \in \mathcal{R}$. Following [7], in section 3.1, we find it convenient to take that function as the sum of the entries of the rate vector, $u(\mathbf{R}) = \mathbf{1}'\mathbf{R}$, where $\mathbf{1}$ is the vector of ones.

Now, if any of the nodes can be freely picked to be the fusion center, then an optimal choice of fusion center can be made. Number the nodes $1, 2, \ldots, n$ and for each i let node i act as the fusion center, and determine the corresponding rate region, $\mathcal{R}_i$. Compute $R_i^* = \text{argmin}_{\mathbf{R} \in \mathcal{R}_i} u(\mathbf{R})$. The optimal choice of fusion center is $i^* = \text{argmin}_{i \in \{1,2,\ldots,n\}} R_i^*$.

2.4.2 No Gain for Two Nodes

For the two node case of our set-up, Figure 2-1, the K-achievability rate region coincides with the omniscience rate region in many interesting cases, because in these cases it turns out that $H(K|X_2) = H(X_1|X_2)$ and $H(K|X_1) = H(K|X_2)$. Thus, to get information about K, the nodes exchange as much information as they would have to exchange to communicate their entire data sequences. Functions, K, for which this is true can be derived using the following proposition.

Proposition 2.4.1. Suppose that the function $f(X_1, X_2)$ is either

(1) injective, or,

(2) both the mappings

$$\{f(X_1,X_2)\times X_1\}\to X_2$$
$$\{f(X_1,X_2)\times X_2\}\to X_1$$

are injective.

Then,

$$H(f(X_1,X_2)|X_1)=H(X_2|X_1)$$
$$H(f(X_1,X_2)|X_2)=H(X_1|X_2)$$

Note that (1) implies (2) but the converse is not true. As an example, consider $f(X_1,X_2) = X_1 \oplus X_2$, the modulo 2 sum, and $X_1,X_2 \in \{0,1\}$. This function is not injective, but (2) holds.

Proof. This proof follows exercise 5 in [5], page 43. Using the chain rule, we can write

$$\begin{aligned}H(f(X_1,X_2),X_2|X_1) &= H(f(X_1,X_2)|X_1)+H(X_2|f(X_1,X_2),X_1)\\ &= H(X_2|X_1)+H(f(X_1,X_2)|X_1,X_2).\end{aligned}$$

But $H(f(X_1,X_2)|X_1,X_2)=0$. Further, if either (1) or (2) holds, then $H(X_2|f(X_1,X_2),X_1)=0$. □

Example 2.4.2. An example of an interesting function K for which the omniscience region and the K-achievability region coincide in the two node case has each f_t equal to a linear function of ${X_1}^t$ and ${X_2}^t$, $f_t=\sum_{j=1}^{t}\alpha_j X_{1j}+\beta_j X_{2j}$. Then, $H(K^N|{X_1}^N)=H({X_2}^N|{X_1}^N)$ $\forall N$. To see this, it suffices to consider $H(K_1,K_2|X_{11},X_{12})$, where $K_1=X_{11}+X_{21}$ and $K_2=(X_{11}+X_{12})+(X_{21}+X_{22})$.

$$\begin{aligned}H(K_1,K_2|X_{11},X_{12}) &= H(X_{11}+X_{21},(X_{11}+X_{12})+(X_{21}+X_{22})|X_{11},X_{12})\\ &\overset{(a)}{=} H(X_{21},X_{21}+X_{22}|X_{11},X_{12})\\ &\overset{(b)}{=} H(X_{21},X_{22}|X_{11},X_{12}),\end{aligned}$$

where (a) follows by proposition 2.4.1 and (b) can be seen by an application of the chain rule:

$$\begin{aligned}H(Y_1,Y_1+Y_2) &= H(Y_1)+H(Y_1+Y_2|Y_1)\\ &= H(Y_1)+H(Y_2|Y_1)\\ &= H(Y_1,Y_2).\end{aligned}$$

Finally, we note that the K-achievability and omniscience rate regions no longer coincide when there are three sources (nodes), even in the simple case when K is linear. For example, let $f_t=X_{1i}+X_{2i}+X_{3i}$. While on one hand we have that $H(K|X_2,X_3)=H(X_1|X_2,X_3)$, on the other hand, for the set of constraints of the other form,

$$H(K|X_3)=H(X_1+X_2|X_3)\le H(X_1,X_2|X_3). \tag{2.19}$$

Further, the inequality in equation (2.19) may be strict for certain joint distributions of (X_1,X_2,X_3). For example, consider (X_1,X_2), which are independent and $\mathbf{P}(X_1=0)=$

$\mathbf{P}(X_1 = 1) = 1/2$ and $\mathbf{P}(X_2 = -1) = \mathbf{P}(X_2 = 0) = 1/2$, and $K = X_1 + X_2$. Then, $H(X_1) = H(X_2) = 1$ and $H(X_1, X_2) = H(X_1) + H(X_2) = 2$, while $H(K) = 3/2$. So, $H(X_1 + X_2) < H(X_1, X_2)$.

2.5 Summary

In this chapter, we looked at compression rates required for communicating nodes to compute a function of all the data in the network with low probability of error. The joint probability mass function of the source was known at all nodes; the task of the nodes was to take advantage of the joint probability mass function in compressing their messages efficiently for decoding K. We have seen that the lower bound to the boundary of the rate region for K achievability also is a lower bound to the SW omniscience rate region, which means that if the objective of the nodes is to reliably acquire a function of all the data in the network, they may only need to exchange information at rates lower than would be required for each of them to acquire all other nodes' data.

We have seen that, for a class of interesting functions, this is not true for two nodes. But, in the next chapter, we will see that when there are more than 2 nodes, things may change. Indeed, for the modulo-2 sum computation example, the K-rate region and the omniscience rate regions coincide for two nodes. However, for more than two nodes, we present a source, described by a joint probability mass function, for which we will show achievability of a point that does not belong to the omniscience rate region. This point is the vertex of the cone defining the lower bound to the K-rate region boundary, defined by the inequalities in Theorem 2.2.1. Furthermore, the example will generalize to modulo-q summation when the source random variables take values on a finite field of order q.

Chapter 3

A Tight Bound: Computation of the Modulo-q Sum

To show that the lower bounds of Theorem 2.2.1 are also sufficient conditions for the achievability of a rate triple, it must be shown that there exist codes with rates that are arbitrarily close to the boundary of the rate region described by the lower bounds, for which the probability of error approaches zero asymptotically. In the proof of the Slepian-Wolf Theorem [5], a "random binning" encoding scheme is used whereby each encoder randomly assigns a codeword/index to each of its messages. Decoding occurs by joint typicality. However, in the case where decoding of a function of the data streams is required, and in the absence of feedback, it is not clear how the random-binning encoding scheme can be used as the nodes do not know what value K takes until after the nodes exchange some information.

In this chapter, it will be shown that when K is the sample-wise modulo-q sum of the nodes' data streams, there exists a specific joint distribution of the source for which the rates specified by Theorem 2.2.1 are also sufficient, even when no interactive communication is used. For our scenario, when there are only two nodes in the network, the omniscience rate region and the K-rate region coincide. However, this is no longer the case when there are more than two nodes.

Ahlswede and Csiszar [1], and later, Orlitsky and Roche [28], studied compression rate requirements to decode $K(X_1^N, X_2^N)$ correctly with high probability; but, they limited their study to two sources (nodes) only, (X_1, X_2). In [1], the authors determine a class of functions ("highly sensitive" and "sensitive") for which, when there is no interactive communication between the nodes, the region of rates necessary and sufficient for decoding K with high reliability coincides with the Slepian-Wolf rate region. Thus, they conclude that "To Get a Bit of Information May Be As Hard As to Get Full Information," as their paper is entitled.

We provide an example for which Ahlswede and Csiszar's conclusion no longer holds when an additional node is added to the network. In particular, in keeping with the results in [1], when $K = X_1 + X_2 \mod 2$, and $x_1, x_2 \in \{0, 1\}$, the "Omniscience/SW rate region" (rates required for omniscience) coincides with the "K-rate region" (the rates required for K-computation). However, when there are three nodes, $x_i \in \{0, 1\}$, $i = 1, 2, 3$, and $K = \sum X_i \mod 2$, for a certain probability mass function on the joint source (X_1, X_2, X_3), the Omniscience and K-rate regions, no longer coincide. We generalize this example to (1) an arbitrary number of nodes, $i = 1, 2, \ldots, n$ and (2) when the source random variables X_i take values in a finite field of order q, $x_i \in GF(q)$, and $K = \sum \alpha_i X_i \mod q$, $\alpha_i \in GF(q)$.

We provide an explicit characterization for the K-rate region as a function of n and q.

3.1 Main Results

Suppose that $K_t = X_{1t} \oplus X_{2t}$ and $x_1, x_2 \in \{0,1\}$, where $\oplus$ denotes modulo-2 addition. The omniscience rate region and the K-rate regions coincide for the two node case because $H(K|X_i) = H(X_j|X_i)$ for $i, j = 1, 2$ and $i \neq j$. However, in this chapter we show that this is not necessarily the case for three nodes. We show that the rate triple $\mathbf{R}_K^* = [h(p), h(p), h(p)]$ is achievable for a specific joint probability mass function of the source. It turns out that $\mathbf{R}_K^*$ is also on the boundary of the K-rate region, and does not belong to the omniscience rate region. Furthermore, we show that the rate triple $\mathbf{R}_K^*$ minimizes the $\mathcal{L}_1$ norm of the rate vector, $R = [R_1, R_2, R_3]$, subject to the rate vector belonging to the K-rate region. We compare $\mathbf{R}_K^*$, to $\mathbf{R}_O^*$, the rate vector belonging to the omniscience rate region, with the smallest $\mathcal{L}_1$ norm. The rate advantage in compressing for K rather than for omniscience is captured by the ratio of the $\mathcal{L}_1$ norms of the minimizing rate vectors, $R(n) = \|\mathbf{R}_O^*\|_1 / \|\mathbf{R}_K^*\|_1$, which depends on the number of nodes, n. Interestingly, for our particular set up, as n grows, $R(n) \to 1/h(p)$.

3.2 The Korner-Marton/Wyner Lemma

In this section, we describe the key lemma that we use to show achievability of rates arbitrarily close to the boundary of the K-rate region, when nodes compute the sample-wise modulo-q sum of their data streams. The existence of these codes can be shown because in that particular case, an analogy can be made between the joint source and the binary symmetric channel (BSC), so the channel code can be used as a source code in our problem. The Channel Coding Theorem assures the existence of codes that achieve capacity and parity check codes are linear codes that do so for the BSC [12, 34, 16].

In the papers of Wyner [34] and Korner and Marton [16], (X_1, X_2) is a discrete memoryless source with binary symmetric joint distribution,

$$\begin{aligned} \mathbf{P}(X_1 = X_2 = 0) = \mathbf{P}(X_1 = X_2 = 1) &= \frac{1}{2}(1-p) \\ \mathbf{P}(X_1 = 0, X_2 = 1) = \mathbf{P}(X_1 = 1, X_2 = 0) &= \frac{1}{2}p. \end{aligned} \tag{3.1}$$

The X_1 source sequence and the X_2 source sequence are encoded separately at two encoders. The codewords are transmitted noiselessly to a decoder that is to produce the X_1 and X_2 sequences with low probability of error. Wyner draws an analogy between coding for this source and coding for a binary symmetric channel. He uses established results for the BSC to conclude existence of a source code with achievable (R_1, R_2) if the rate pair (R_1, R_2) is in the rate region specified by the SW inequalities. Thus, he provides a new proof for the SW Theorem for a very special case.

Korner and Marton, on the other hand, use the same idea to establish the existence of distributed codes with the rate pair $(h(p), h(p))$ which guarantee decoding $K = X_1 \oplus X_2$ with high reliability. Note that for this K, $H(K|X_2) = H(X_1|X_2)$ and $H(K|X_1) = H(X_2|X_1)$; further, for the chosen source distribution, $H(K) = H(K|X_2) = H(K|X_1) = h(p)$. We also note that the Korner-Marton statement is interesting because this achievable rate pair does not belong to the SW rate region for the same set up (Figure 2-2). Because decoding is "centralized", there is the additional condition that $R_1 + R_2 \geq H(X_1, X_2)$, where $H(X_1, X_2) = 1 + h(p)$. Clearly, this condition does not hold for $(h(p), h(p))$, $p \neq \frac{1}{2}$.

For convenience, we repeat below a lemma from [16] that is attributed to results of Elias in [12]. This lemma contains the needed result of the analogy with the BSC, which can be found in [34], and is explained below.

Lemma 3.2.1. Let a sequence $\{K_i\}_{i=1}^{\infty}$ of i.i.d. binary random variables be given. For fixed $\epsilon > 0$ and for sufficiently large N there exists a 0-1 matrix A of $M \times N$ entries and a function $\psi : \{0,1\}^M \to \{0,1\}^N$ such that

(C1) $M < N(H(K) + \epsilon)$,

(C2) letting $A(k^N)$ denote the modulo-two product of the matrix A with the binary column vector k^N of length N, we have $\mathbf{P}(\psi(A(K^N)) \neq K^N) < \epsilon$.

Now, the idea used by Korner and Marton is to use A at both the X_1 encoder, to encode X_1^N, and the X_2 encoder, to encode X_2^N. Now at the decoder, ψ is applied to $A(X_1^N) \oplus A(X_2^N)$. By linearity of A and (C2) of lemma 3.2.1,

$$\begin{aligned}\mathbf{P}(\psi(A(X^N) \oplus A(Y^N)) \neq K^N) &= \mathbf{P}(\psi(A(X^N \oplus Y^N)) \neq K^N)\\ &= \mathbf{P}(\psi(A(K^N)) \neq K^N) < \epsilon.\end{aligned}$$

Since A maps the length N binary messages of X_1 to binary codewords of length M, there are 2^M possible codewords. The rate of this source code is therefore $\frac{1}{N}\log 2^M = M/N$, and we have by (C1) of Lemma 3.2.1 that $h(p)+\epsilon > \frac{1}{N}\log 2^M$. Similarly for the X_2 encoder. Hence, $(h(p), h(p))$ is an achievable pair (recall definition 2.1.1).

The proof of the Lemma that is given in [16, 34] and below uses an analogy to channel coding for the BSC, however, there is an intuitive interpretation of the Lemma that relates to the typical set, A_ϵ^N, with respect to the distribution of K[20]. The idea is also found in the proof of Theorem 3.2.1 ([5] p.54) where it is shown that, on average, $NH(K)$ bits are needed to represent a sequence of IID random variables, K^N. Because for N sufficiently large, $\mathbf{Pr}(A_\epsilon^N) \geq 1 - \epsilon$, what is effectively needed in order to encode K^N and decode it with high reliability is to enumerate a subset of the possible realizations of K^N, the elements of the typical set,

$$A_\epsilon^N = \{(k_1, k_2, \ldots, k_N) : 2^{-N(H(K)+\epsilon)} \leq p(k_1, k_2, \ldots, k_N) \leq 2^{-N(H(K)-\epsilon)}\},$$

of which there are no more than $2^{N(H(K)+\epsilon)}$, so no more than $N(H(K)+\epsilon)$ bits are needed.

The outputs of the linear operator A are vectors of length $M < N$, which means that there are 2^M possible sequences that can be mapped to the elements of the typical set. For M large enough, that map will be injective. But M need not be any larger than $N(H(K)+\epsilon)$ to ensure low probability of decoding error.

Proof. (Lemma 3.2.1) The existence of A and ψ such that (C1) and (C2) of Lemma 3.2.1 holds for N sufficiently large is guaranteed by the Channel Coding Theorem and standard results for the BSC (see [12] page 206). Let U be the input of a BSC and V its output. Note that if the channel crossover probability is p, $V = U \oplus K$, where K and U are independent and $\mathbf{P}(K = 0) = 1 - p$. Now, when U is uniformly distributed, the joint distribution of (U, V) is given by (3.1).

By the Channel Coding Theorem, there exist a sequence of codes, with rate R, such that if $R < C$, for sufficiently large block length N, the average probability of error is arbitrarily small. For the above BSC, $C = 1 - h(p) = 1 - H(K)$; and when the channel input is uniformly distributed that capacity is achieved, that is, there exist codes with rate R arbitrarily close to capacity, $R + \epsilon > C$, for which the probability of error goes to zero as block length increases. Furthermore, for the BSC, such codes include parity check codes and decoding is accomplished by maximum likelihood decoding.

Decoding the channel output occurs by multiplying (modulo 2) the parity check matrix, A, consisting of zeros and ones, by V^N. Since the choice of A must be such that U^N belongs to the null space of A, $A(V^N) = A(K^N)$. Now, decoding can be performed by ψ as described in [12] page 201, such that $\psi(A(K^N)) = \widehat{K^N}$, and $\widehat{U^N} = \widehat{K^N} + V^N$. An error occurs when $\{\widehat{U^N} \neq U^N\}$, or equivalently, $\{\widehat{K^N} \neq K^N\}$. So, for codes with $R < C$, we have $\mathbf{P}(\{\widehat{K^N} \neq K^N\}) \to 0$, which is (C2) of Lemma 3.2.1.

Finally, note that the codewords at the input of the channel have length N. Since they belong to the null space of A, there are at most $N - M$ messages (and $N - M$ information digits). So, the rate of the code is $R = (N - M)/N$. For the capacity achieving code with this rate, substituting in $R + \epsilon > C$, and using $C = 1 - H(K)$, we obtain (C1) of Lemma 3.2.1. □

3.3 Modulo-2 Sum Computation in an n-Node Network

For the three node case, when $K = X_1 \oplus X_2 \oplus X_3$ and for the appropriate probability distribution on (X_1, X_2, X_3), the achievability of the rate triple $(h(p), h(p), h(p))$ follows due to a simple extension of of the above lemma to three sources. Furthermore, the rate triple is on the boundary of the K-achievability rate region. The idea can also be extended to more than three nodes, as we show below.

$\underline{n = 3.}$ We determine the K-rate region under the following assumptions. Let

1. $n = 3$,
2. $K = X_1 \oplus X_2 \oplus X_3$, and
3. the source variables (X_1, X_2, X_3) have the following joint probability mass function.

$$\begin{aligned} &\mathbf{P}(X_1 = X_2 = 0 | X_3 = 0) \\ &= \mathbf{P}(X_1 = X_2 = 1 | X_3 = 0) = \frac{1}{2}(1-p), \\ &\mathbf{P}(X_1 = 0, X_2 = 1 | X_3 = 0) \\ &= \mathbf{P}(X_1 = 1, X_2 = 0 | X_3 = 0) = \frac{1}{2}p. \end{aligned} \tag{3.2}$$

and

$$\mathbf{P}(X_3 = 0) = \mathbf{P}(X_3 = 1) = \frac{1}{2}. \tag{3.3}$$

We assume that X_1, X_2, and X_3 are pair-wise independent.

Then, $H(K|X_i, X_j) = H(K|X_i) = h(p)$, and indeed, by symmetry, all entropies in equation (2.16) are equal to $h(p)$. In fact, $H(K|X_i, X_j) = H(\{X_i, X_j\}^c | X_i, X_j)$, where $\{X_i, X_j\}^c$

is $\{X_1, X_2, X_3\} \backslash \{X_i, X_j\}$. Finally, $H(X_1, X_2|X_3) = H(X_1|X_3) + H(X_2|X_1, X_3) = 1 + h(p)$, and, by symmetry, so are the remaining entropies of the same form that define the SW rate region.

Now, we have that,the omniscience rate region, $\mathcal{R}_O$, is determined by the equations

$$\begin{aligned}R_1 &\geq h(p)\\ R_2 &\geq h(p)\\ R_3 &\geq h(p)\\ R_1 + R_2 &\geq 1 + h(p)\\ R_2 + R_3 &\geq 1 + h(p)\\ R_1 + R_3 &\geq 1 + h(p),\end{aligned} \tag{3.4}$$

Furthermore, we have the following theorem, which is the core of the main result of this chapter.

Theorem 3.3.1. *For $K = X_1 \oplus X_2 \oplus X_3$ and for the probability mass function on (X_1, X_2, X_3) defined in equations (3.2) and (3.3), (R_1, R_2, R_3) is achievable iff (2.16) holds. Specifically,*

$$\begin{aligned}R_1 &\geq h(p)\\ R_2 &\geq h(p)\\ R_3 &\geq h(p)\\ R_1 + R_2 &\geq h(p)\\ R_2 + R_3 &\geq h(p)\\ R_1 + R_3 &\geq h(p),\end{aligned} \tag{3.5}$$

where $h(p) = -p \log p - (1-p) \log(1-p)$.

Proof. The converse part of Theorem 3.3.1 was shown in section 2.3. To show the achievability part of the Theorem, it suffices to show that by an extension of Lemma 3.2.1, $(h(p), h(p), h(p))$ is also an achievable rate triple, so it belongs to the K-rate region, $\mathcal{R}_K$. This is true because here the K-rate region is fully specified by the first three inequalities of (2.16); any rate triple satisfying the first three inequalities will also satisfy the last three. So, the rate region is a cone whose vertex is $(h(p), h(p), h(p))$.

By Lemma 3.2.1, there exists A and ψ such that (C1) and (C2) hold. For our source coding problem, we apply A to X_3^N at the X_3 encoder and use ψ at the decoders of the receiving nodes. By symmetry, we can use the same A and ψ at all the nodes. The decoding works by applying ψ to $A(X_1^N) \oplus A(X_2^N) \oplus A(X_3^N)$ because, by linearity of the operator A, $\psi(A(X_1^N) \oplus A(X_2^N) \oplus A(X_3^N)) = \psi(A(X_1^N \oplus X_2^N \oplus X_3^N)) = \psi(A(K^N)) = \widehat{K^N}$. Now, since $H(K) = h(p)$, $(h(p), h(p), h(p))$ is an achievable rate triple. □

Two important details should be mentioned here. These will be used later in extending our argument to more than 3 nodes.

1. For the achievable triple to be $(h(p), h(p), h(p))$, we must have $H(K) = h(p)$.
2. The point $(h(p), h(p), h(p))$ will be the vertex of the cone that is the K-achievable rate region when $H(K|X_1, X_2) = H(K|X_1) = h(p)$, and, moreover, all the conditional entropies defining $\mathcal{R}_K$ are equal to $h(p)$. This constraint imposes structure on the joint distribution of the source.

Finally, it is clear that $(h(p), h(p), h(p)) \notin \mathcal{R}_O$ for $p \neq \frac{1}{2}$, since in this case $h(p) < 1$. So, the last three conditions in 3.4 do not hold.

$\underline{n \geq 4.}$ Lemma 3.2.1 indeed holds for any n, source $(X_1, X_2, \ldots, X_n)$ and $K_t = \sum_{i=1}^{n} X_{it}$ mod 2. Thus, achievability of $h(p)\mathbf{1}$ for $\mathbf{1} \in \mathbb{R}^n$ for the n node case follows in the same way that was argued above for the three node case, as long as $(X_1, X_2, \ldots, X_n)$ has the appropriate joint probability mass function.

Corollary 3.3.2. For any n, there exists a joint source $(X_1, \ldots, X_n)$ with appropriate joint probability mass function such that, when $K_t = \sum_{i=1}^{n} X_{it}$ mod 2, the rate vector $(h(p), \ldots, h(p)) \in \mathbb{R}^n$ is

- the vertex of the cone defining the lower bound to the K-rate region boundary, $R_i \geq h(p)$ for $i = 1, \ldots, n$,
- and, is achievable.

Example 3.3.3. Let n=4. Let the joint probability mass function on (X_1, X_2, X_3) defined by (3.2) and (3.3) be equal to the conditional probability measure, $\mathbf{P}(X_1, X_2, X_3 | X_4 = 0)$. Let $\mathbf{P}(X_4 = 0) = \frac{1}{2}$, and let any 3 random variables taken from X_1, X_2, X_3, X_4 be independent, e.g. $\mathbf{P}(X_1 = x_1, X_2 = x_2, X_3 = x_3) = \mathbf{P}(X_1 = x_1)\mathbf{P}(X_2 = x_2)\mathbf{P}(X_3 = x_3)$, and let the marginals all be uniform. Then, $h(p)\mathbf{1} \in \mathbb{R}^4$ is achievable. Note that,

1. For $K = X_1 \oplus X_2 \oplus X_3 \oplus X_4$, $H(K) = H(K|X_1, X_2, X_3) = H(K|X_1, X_2) = H(K|X_1) = h(p)$, so, the right hand side of all the inequalities defining $\mathcal{R}_K$ (4-node equivalent to 2.16) is $h(p)$.
2. $H(X_4|X_1, X_2, X_3) = h(p)$. Also, $H(X_1, X_2|X_3, X_4) = 1+h(p)$ and $H(X_1, X_2, X_3|X_4) = 2 + h(p)$, as can be computed by applying the chain rule.

So, the SW constraints defining the rate region for omniscience, $\mathcal{R}_O$, are,

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \end{bmatrix} \geq \begin{bmatrix} h(p) \\ h(p) \\ h(p) \\ h(p) \\ 1+h(p) \\ 1+h(p) \\ 1+h(p) \\ 1+h(p) \\ 1+h(p) \\ 1+h(p) \\ 2+h(p) \\ 2+h(p) \\ 2+h(p) \\ 2+h(p) \end{bmatrix} \tag{3.6}$$

3.4 Minimum Achievable Rates: Omniscience versus K-achievability

We compare the omniscience and K-rate regions to quantify how much is gained, in terms of being able to compress more and communicate at lower rates, by compressing for K rather than compressing for omniscience. For this, we choose to characterize the rate region by a single number as in [7], namely,

$$u(\mathbf{R}^*) = \min_{\mathbf{R}\in\mathcal{R}} u(\mathbf{R}), \tag{3.7}$$

where $u(\mathbf{R}) = \mathbf{1}'\mathbf{R}$, $\mathbf{R} = [R_1, R_2, \ldots, R_n]'$, and $\mathcal{R}$ is either the K-rate region, $\mathcal{R}_K$, or the omniscience rate region, $\mathcal{R}_O$. We quantify the change as n grows of the rate savings gained due to compressing for K, by computing $R(n) = u(\mathbf{R}_O^*)/u(\mathbf{R}_K^*)$.

$\underline{\mathbf{R}_K^*}$. For the chosen joint probability mass function of the source, $(X_1, X_2, \ldots, X_n)$, the right-hand side of all the inequalities defining the K-rate region is $h(p)$, for any n. It is obvious that any solution satisfying the first n constraints, $R_i \geq h(p)$ for $i = 1, \ldots n$, will simultaneously satisfy all the rest, namely, that the sum of any pair, triple, ..., n-tuple of rates must be greater or equal to $h(p)$. Now, since any permutation of the indicies of the rates, R_i, yields the exact same minimization problem (3.7), we must have that, for every i, $R_i^* = c$, where c is a constant. Clearly, $c = h(p)$, and $u(\mathbf{R}_K^*) = nh(p)$.

$\underline{\mathbf{R}_O^*}$. In general, there are $\sum_{i=1}^{n-1} \binom{n}{i}$ inequalities that define the SW rate region. The first n are constraints on R_i; in our example, the right-hand side of all the inequalities is $h(p)$. The next $\binom{n}{2}$ inequalities are on all the possible ways to sum two pairs of rates out of n; in our example, the right-hand side of all the inequalities is $1 + h(p)$. The m^{th} set of constraints are on the $\binom{n}{m}$ possible ways to sum m out of n rates; in our example, the right-hand side of all the inequalities is $m - 1 + h(p)$.

For our example, only the last n equations determine the solution to the minimization problem; if a rate vector satisfies the last n constraints, it simultaneously satisfies all the remaining inequalities. To see this, first consider the last n constraints that have the form

$$R_1 + R_2 + \cdots + R_{n-1} \geq n - 2 + h(p).$$

Assume, $\forall i \in \{1, 2, \ldots, n\}$,

$$R_i \geq \frac{n - 2 + h(p)}{n - 1}. \tag{3.8}$$

Now, consider the m^{th} set of constraints for $m \leq n - 2$, for example,

$$R_1 + R_2 + \cdots + R_m \geq m - 1 + h(p).$$

Then, for the choice of R_i in (3.8) we have that these constraints are satisfied.

$$\begin{aligned}
R_1 + R_2 + \cdots + R_m &\geq \frac{m(n-2)}{n-1} + \frac{m}{n-1}h(p) \\
&= \frac{m(n-2)}{n-1} + h(p) - \frac{n-1-m}{n-1}h(p) \\
&\overset{(a)}{\geq} \frac{m(n-2)}{n-1} - \frac{n-1-m}{n-1} + h(p) \\
&= m - 1 + h(p),
\end{aligned}$$

where (a) holds because $h(p) \leq 1$; in fact, equality holds for $p = 1/2$ and inequality is strict otherwise.

Finally, any permutation of the indices of the rates, R_i, in the minimization problem (3.7) will result in the same problem. Therefore, the minimizing solution must consist of a vector with the same entries, for every i, $R_i = c$, where c is some constant. So, we must have that $c = \frac{n-2+h(p)}{n-1}$, and $u(\mathbf{R}_O^*) = \frac{n}{n-1}(n-2+h(p))$.

<u>$u(\mathbf{R}_O^*)/u(\mathbf{R}_K^*)$.</u> Substituting the results of the above computations we have that

$$\begin{aligned} R(n) &= \frac{u(\mathbf{R}_O^*)}{u(\mathbf{R}_K^*)} \\ &= \frac{n-2+h(p)}{(n-1)h(p)} \\ &\xrightarrow[n\to\infty]{} \frac{1}{h(p)}. \end{aligned}$$

We note here that this result is not surprising in view of the fact that the number of constraints for both minimization problems, (3.7) with $\mathcal{R} = \mathcal{R}_K$ and with $\mathcal{R} = \mathcal{R}_O$, is linear in n. So, neither rate region, $\mathcal{R}_K$ or $\mathcal{R}_O$, shrinks more rapidly than the other as n increases.

3.5 Generalizing to Finite Fields: Modulo-q Sum Computation

Lemma 3.2.1 can be generalized to the case when the sources $X_1, X_2, \dots, X_n$, take values from a finite field of order q, $GF(q)$, and $K = \sum_{i=1}^n \alpha_i X_i \mod q$, where α_i's are constants in $GF(q)$.

Lemma 3.5.1. Let a sequence $\{K_i\}_{i=1}^\infty$ of i.i.d. random variables taking values in $GF(q)$ be given. For fixed $\epsilon > 0$ and for sufficiently large N there exists a matrix A of $M \times N$ entries which are elements of $GF(q)$ and a function $\psi : GF(q)^M \to GF(q)^N$ such that

(C1) $M \log q < N(H(K) + \epsilon)$,

(C2) letting $A(k^N)$ denote the product in $GF(q)$ of the matrix A with the column vector k^N of length N, we have $\mathbf{P}(\psi(A(K^N)) \neq K^N) < \epsilon$.

This lemma then establishes the existence of a decentralized coding scheme with the achievable rate vector $H(K)\mathbf{1}$, where $\mathbf{1} \in \mathbb{R}^n$ is the vector of ones. In particular, the encoder of source $i = 1, \dots, n$ applies the matrix A to its sequence X_i^N. The decoder at j, upon receiving $A(X_k^N)$ for $k \neq j$, and having its own $A(X_j^N)$, forms the sum $\sum_{i=1}^n \alpha_i A(X_i^N)$ mod q. Now, by linearity of A,

$$\begin{aligned} \sum_{i=1}^n \alpha_i A(X_i^N) \mod q &= A(\sum_{i=1}^n \alpha_i X_i^N \mod q) \\ &= A(K^N) \end{aligned}$$

So, application of the decoding function ψ to the sum will result in $\psi(A(K^N)) = \widehat{K^N}$, for which, by lemma 3.5.1, $\mathbf{P}(\psi(A(K^N)) \neq K^N) < \epsilon$.

Again, the lemma can be interpreted in terms of the typical set of K^N, A_ϵ^N. As before, for N sufficiently large, $\mathbf{P}(A_\epsilon^N) \geq 1-\epsilon$, so, it is possible to get low probability of decoding error by enumerating the elements of the typical set. Again, there are no more than $2^{N(H(K)+\epsilon)}$ of those elements. Now, A generates q^M sequences, so our code can enumerate as many sequences. Since we need not enumerate more than $2^{N(H(K)+\epsilon)}$ sequences, we have that $q^M < 2^{N(H(K)+\epsilon)}$.

The lemma can also be shown by analogy with a symmetric discrete memoryless channel (DMC), as was done above for $q = 2$. To see this, consider $q = 3$. Let U be the input of a DMC, and V its output. Let the channel transition probability be given by the following matrix, P, where the entries are $p_{ij} = \mathbf{P}(V = j-1|U = i-1)$, for $i, j \in \{1, 2, 3\}$,

$$P = \begin{bmatrix} p_1 & p_2 & 1-p_1-p_2 \\ 1-p_1-p_2 & p_1 & p_2 \\ p_2 & 1-p_1-p_2 & p_1 \end{bmatrix}.$$

Note that for this channel, the output can also be written as $V = U + K \mod q$, where U and K are independent and K has the distribution: $\mathbf{P}(K = 0) = p_1$ and $\mathbf{P}(K = 1) = p_2$. Consequently, $H(K) = -p_1 \log p_1 - p_2 \log p_2 - (1-p_1-p_2)\log(1-p_1-p_2)$.

By the Channel Coding Theorem, there exist codes for which, when the block length is sufficiently large, the probability of decoding error can be made arbitrarily small as long as the code rate, R, is smaller than the channel capacity, C. Further, for a DMC with inputs belonging to $GF(q)$, there exist *linear* codes which are capacity achieving. If the messages are sequences of length $N - M$, and the channel inputs and outputs are of length N, then the rate of the channel code (the number of information bits divided by the block length) is $R = \frac{log 3^{N-M}}{N}$, and the parity check matrix that is used to decode the outputs of the channel has dimension $M \times N$.

Now, it can be shown (Theorem 4.5.2, [12] p.94) that for a symmetric DMC, capacity is achieved when the inputs have equal probabilities. Using this fact in Theorem 4.5.1 [12] p.91, the capacity of the channel with the above transition probability matrix is $C = \log 3 - H(K)$. Since a uniformly distributed input can achieve capacity, we have that $R + \epsilon > C$. Substituting the expressions for C and R, we obtain (C1) of Lemma 3.5.1.

Having established that there exists a decentralized source code with the achievable rate vector $H(K)\mathbf{1}$, we proceed, as before, to construct a source joint probability mass function for which the rate region defined by the K-achievability inequalities is a cone whose vertex is the point $H(K)\mathbf{1}$. Since it has already been established that the vertex is achievable, it follows that every point in the cone is also achievable. Constructing such source probability mass functions can be done for any q, but, again we illustrate for $q = 3$. First, we have the general statement of our result in the following corollary.

Corollary 3.5.2. For any n, there exists a joint source $(X_1, \ldots, X_n)$ with appropriate joint probability mass function such that, when $K_t = \sum_{i=1}^{n} \alpha_i X_{it} \mod q$, the rate vector $(H(K), \ldots, H(K)) \in \mathbb{R}^n$ is

- the vertex of the cone defining the lower bound to the K-rate region boundary, $R_i \geq H(K)$ for $i = 1, \ldots, n$,
- and, is achievable.

Example 3.5.3. For this example, let $n = 3$ and $K = \sum_{i=1}^{3} X_i \mod 3$. Let the sources (X_1, X_2, X_3) have the following joint probability mass function. Let P_0 be the matrix whose elements are $p^0_{ij} = \mathbf{P}(X_1 = i-1, X_2 = j-1|X_3 = 0)$ for $i, j \in \{1, 2, 3\}$, P_1 be the matrix whose elements are $p^1_{ij} = \mathbf{P}(X_1 = i-1, X_2 = j-1|X_3 = 1)$ for $i, j \in \{1, 2, 3\}$, and P_2 be the matrix whose elements are $p^2_{ij} = \mathbf{P}(X_1 = i-1, X_2 = j-1|X_3 = 2)$ for $i, j \in \{1, 2, 3\}$.

The matrices are

$$P_0 = \frac{1}{3}\begin{bmatrix} p_1 & p_2 & 1-p_1-p_2 \\ p_2 & 1-p_1-p_2 & p_1 \\ 1-p_1-p_2 & p_1 & p_2 \end{bmatrix},$$

$$P_1 = \frac{1}{3}\begin{bmatrix} p_2 & 1-p_1-p_2 & p_1 \\ 1-p_1-p_2 & p_1 & p_2 \\ p_1 & p_2 & 1-p_1-p_2 \end{bmatrix},$$

$$P_2 = \frac{1}{3}\begin{bmatrix} 1-p_1-p_2 & p_1 & p_2 \\ p_1 & p_2 & 1-p_1-p_2 \\ p_2 & 1-p_1-p_2 & p_1 \end{bmatrix}.$$

Furthermore, we assume that X_3 is uniformly distributed.

Then, we have that $H(K) = H(K|X_i) = H(K|X_i, X_j)$, $H(X_i|\{X_i\}^c) = H(K)$, where $\{X_i\}^c$ is $\{X_1, X_2, X_3\}\backslash\{X_i\}$, and $H(\{X_i\}^c|X_i) = \log 3 + H(K)$. So, the K-achievability rate region is defined by

$$\begin{aligned} R_1 &\geq H(K) \\ R_2 &\geq H(K) \\ R_3 &\geq H(K) \end{aligned}$$

and the Slepian-Wolf rate region is defined by

$$\begin{aligned} R_1 + R_2 &\geq \log 3 + H(K) \\ R_2 + R_3 &\geq \log 3 + H(K) \\ R_1 + R_3 &\geq \log 3 + H(K). \end{aligned}$$

The construction of a source joint distribution can be done for 4 nodes starting form the 3 node joint distribution as was done in Example 3.3.3 for the binary case. Indeed, if we generalize the above to n nodes when the sources take values in $GF(q)$, we have that the rates in the K-rate region satisfy $R_i \geq H(K)$ and the rates in the SW-rate region satisfy $R_i \geq \frac{1}{n-1}((n-2)\log q + H(K))$. Thus, we have that

$$\begin{aligned} R(n) &= \frac{u(\mathbf{R}_O^*)}{u(\mathbf{R}_K^*)} \\ &= \frac{(n-2)\log q + H(K)}{(n-1)H(K)} \\ &\xrightarrow[n\to\infty]{} \frac{\log q}{H(K)}. \end{aligned}$$

3.6 Summary

In this chapter, we have demonstrated that when nodes need to compress their data in order to reliably compute the sample-wise modulo-q sum of their data, they need to exchange information at rates lower than would be required for each of them to acquire all other nodes' data. This is not true for the two node case, but, when there are more than 2 nodes, things change. Indeed, for the modulo-2 sum computation example, we presented a source for which we showed achievability of a point that does not belong to the omniscience rate region. This point turned out to be the vertex of the cone defining the lower bound to the K-rate region boundary, defined by the inequalities in Theorem 2.2.1. Furthermore, the example was generalized to modulo-q summation when the source random variables take values on a finite field of order q.

Chapter 4

Computation via Noisy Channels

We consider a network of n nodes communicating via noisy channels. Each node has some real-valued initial measurement or message. The goal of each of the nodes is to acquire an estimate of a given function of all the initial measurements in the network.

We seek to understand the limitations imposed by the communication constraints on the nodes' performance in computing the desired function. The performance is measured by the mean square error in the nodes' estimates of the desired function. The communication constraints consist of (1) the topology of the network, that is, the connectivity of the nodes, and (2) the noisy channels between nodes that communicate. In order to capture the limitation due to the communication constraints, we assume that that the nodes have unlimited computation capability. Each node can perform any amount of computation as well as encoding and decoding for communication.

In this formulation, each node has an initial value from a source having some distribution; each node knows nothing of the source distributions of other nodes. This is unlike the "source coding" formulation of Chapter 2 where we assumed that the joint probability mass function of the source was known at all nodes. In that chapter, the task of the nodes was to take advantage of the joint probability mass function in compressing their messages efficiently for decoding a function of the data in the network. Another difference between the two formulations is that here, each node has some real-valued initial measurement or message. In Chapter 2, nodes generated their messages, a data stream, by sampling from their respective sources which were discrete.

As was the case in Chapter 2, the formulation of this chapter lends itself to Information Theoretic techniques. We use Information Theoretic inequalities to derive lower bounds on information exchange necessary between nodes for the mean square error in the nodes' estimates to converge to zero. One way we use this bound is to determine an upper bound on the rate of the channel code in terms of the capacity of the channel and the required rate of convergence of the mean square error to zero. Another way we use the Information Theoretic technique is to determine a lower bound on computation time that must be satisfied by any algorithm used by the nodes to communicate and compute, so that the mean square error in the nodes' estimates is within a given interval around zero. In the latter case, the bound is in terms of the channel capacities, the size of the desired interval, and the uncertainty in the function to be computed.

Existing results include algorithms with upper and/or lower bounds on the time for the nodes to reach agreement or compute a certain quantity with given accuracy, when communication is subject to topological constraints, but perfect when present [2, 3, 33, 32]. Another set of work investigates algorithms for computation when communication is unreliable. The channels in the network are explicitly modelled. The researchers propose an algorithm that will perform the desired computation while satisfying some performance criterion. For example, in [11], each node in the network has one bit. Nodes broadcast messages to each other via binary symmetric channels. The goal is for a fusion center to compute the parity of all the bits in the network. Gallager proposes an algorithm that can be used while guaranteeing a desired probability of error. He exhibits an upper bound that is a constant multiple of the bits that must be transmitted per node. Recently, it has been shown in [14] that this algorithm is optimal. The authors produce an algorithm-independent lower bound that is of the same order as the upper bound.

Many different formulations and corresponding bounds can be found in the literature. Two examples are [10, 18]. In [10], the authors derive Information Theoretic bounds on the number of bits that must be exchanged for nodes communicating via noiseless channels to acquire each other's data. In [18], the authors present lower bounds to the number of messages that must be communicated by two sensors to a fusion center that must determine a given function of the nodes' data. However, the transmitted messages are real-valued vectors and the lower bound is on the sum of the dimensions of the message functions. Several formulations and results relevant to computation in wireless sensor networks can be found in a detailed survey by Giridhar and Kumar [13].

Our approach and, hence results, are quite different. We capitalize on Martins' successful use of Information Theoretic tools in [19, 20, 21, 22] to characterize fundamental performance limits of feedback control systems with communication constraints. We use Information Theoretic inequalities, reminiscent of those of Rate-Distortion theory, in a different setting with different objectives. In particular, we have a network of nodes whose objective is to compute a given function of the nodes' data, rather than to communicate reliably to each other their data.

The Information Theoretic approach captures fundamental performance limitations that arise in the network due to the communication constraints. This happens because the analysis is independent of the communication algorithm used by the nodes. The lower bound we derive in this chapter enables us to characterize the effect of the network structure on algorithm running time. As we will further discuss in the next chapter, for nodes exchanging information over erasure channels to compute the sum of their initial conditions, the lower bound is indeed tight in capturing the network constraints.

In the next section, we describe the problem formulation and state the main theorem of this chapter. In section 4.2 we recall the Information Theoretic definitions and properties that we will need. In section 4.3 we prove our main theorem. Finally, we end by discussing a technical difficulty that arises in applying our lower bound and we suggest a technique to circumvent this difficulty.

4.1 Problem Formulation

A network consists of n nodes, each having a random initial condition or value. We represent the initial condition at node i by the random variable $X_i(0)$. Let $X(0)$ represent the vector of all the initial condition random variables, $[X_1(0) \ \dots \ X_n(0)]'$. Each node is required to compute a given function of all the initial conditions. That is, node i is required to estimate $C_i = f_i(X(0))$. We let $C = [C_1 \ \dots \ C_n]'$. Suppose that nodes 1 to m belong to set S. Whenever we use a set as a subscript to a variable, we mean the vector whose entries are that variable subscripted by the elements of the set. For example, $C_S = [C_1 \ \ldots \ C_m]'$.

We assume that time is discretized into intervals, and enumerated by positive integers, $\{1, 2, \ldots\}$. During each time step, a node can communicate with its neighbors. At the end of time-slot k, node i uses the information it has received thus far to form an estimate of C_i. We denote this estimate by $X_i(k)$. Let, X_i^k denote the finite sequence of estimates at node i, $\{X_i(1), X_i(2), \ldots X_i(k)\}$. The estimates of all nodes in the network at the end of time slot k are denoted by the vector $X(k) = [X_1(k) \ \ldots \ X_n(k)]'$. And, the estimates of nodes in set S are denoted by $X_S(k) = [X_1(k) \ \ldots \ X_m(k)]'$.

The nodes communicate via noisy channels. The network structure is described by a graph, $G = (V, E)$, where V is the set of nodes and E is the set of edges, (i, j). If node i communicates with node j via channel with capacity $\mathbf{C}_{ij} > 0$, then $(i, j) \in E$. If $(i, j) \notin E$, we set $\mathbf{C}_{ij} = 0$. We assume that all channels in the network are independent memoryless discrete-time. Further, for each of the channels, one channel symbol is sent per $\tau_c = 1$ second. Each node generates an input for its encoder every τ seconds, and by the k^{th} input generated, $X_i(k)$, N channel digits have been sent; so, $N\tau_c = k\tau$. When $\tau_c = 1$, the time T until the k^{th} node estimate, $X_i(k)$, has been generated is $T = k\tau$. With no loss of generality, we assume in what follows that $\tau = 1$. So, $X_i(k) = X_i(T)$.

We consider two mean square error criteria. The operator $\|\cdot\|$ is to be interpreted, when the argument is a vector, C, as $\|C\|^2 = \sum C_i^2$.

C1. $E(\|X(T) - C\|^2) \leq \beta 2^{-\alpha}$, and,

C2. $E(X_i(T) - C_i)^2 \leq \beta 2^{-\alpha}$, for all $i \in \{1, \ldots, n\}$,

where $\beta, \alpha \in \mathbb{R}^+ \backslash \{0\}$.

The first criterion requires that as the number of nodes increases, the per node error is also smaller. It suggests that as the number of nodes, n, increases, we require the mean square errors at each of the nodes, $E(X_i(k) - C_i)^2$ to decrease like $1/n$. This criterion is appropriate if, for example, the initial values at the nodes are independent and each node is to estimate the average of the initial values in the network. As the number of nodes increases, the variance of the average decreases. In circumstances where this does not happen, the second criterion may be more appropriate.

The "computation time" is the first time at which the desired performance criterion holds. We seek a lower bound on the computation time, T, that holds if the desired mean square error criterion, C1 or C2, is satisfied. That is, if C1 or C2 holds, then how large must T be?

4.1.1 Main Result

The main theorem of this chapter provides a lower bound to computation time as a function of the accuracy desired, as specified by the mean square error, and the uncertainty in the function that nodes must learn, as captured by the differential entropy.

Theorem 4.1.1. *For the communication network described above, if at time, T, the mean square error is in an interval prescribed by α, $E(X_i(T) - C_i)^2 \leq \beta 2^{-\alpha}$, for every node, then T is lower bounded by*

$$T \geq \max_{S \subset V} \frac{\bar{L}(S)}{\sum_{i \in S^c} \sum_{j \in S} \mathbf{C}_{ij}},$$

where $S^c = V \backslash S$ *and,*

$$\bar{L}(S) = h(C_S|X_S(0)) - \frac{|S|}{2}\log 2\pi e\beta + |S|\frac{\alpha}{2}.$$

This theorem captures the fact that the larger the uncertainty in the function to be estimated, or the larger the desired accuracy, the longer it must take for any algorithm to converge.

4.2 Information Theoretic Preliminaries

The differential entropy of C is denoted by $h(C)$. The mutual information between X and C is denoted by $I(X;C)$.

When indicated, we will need to use the most general definition of mutual information. It can be used when the random variables are arbitrary ensembles, not necessarily both continuous or both discrete. We repeat this definition from [29, p.9]. The conditional mutual information is similarly defined; see [29, Ch. 3].

Suppose X and C are random variables that take values in the measurable spaces (Ω_X, S_X) and (Ω_C, S_C), respectively. S_X denotes the sigma algebra of subsets of Ω_X. Let the probability distributions of X and C be $\mathbf{P}_X$ and $\mathbf{P}_C$. Let $\mathbf{P}_{XC}$ be the joint distribution of X and C.

The mutual information between X and C is

$$I(X;C) = \sup \sum_{i,j} \mathbf{P}_{XC}(E_i \times F_j) \log \frac{\mathbf{P}_{XC}(E_i \times F_j)}{\mathbf{P}_X(E_i)\mathbf{P}_C(F_j)},$$

where the supremum is taken over all partitions $\{E_i\}$ of Ω_X and partitions $\{F_i\}$ of Ω_C.

In the remainder of this section, let C and X be continuous random variables. The definitions for mutual information and differential entropy for this case are repeated from [5, Ch.9].

Let X and C have the probability densities $p(x)$ and $p(c)$. Let their joint density be $p(x, c)$. Then their mutual information is defined as

$$I(X;C) = \int p(x,c) \log \frac{p(x,c)}{p(x)p(c)}\, dx\, dc.$$

The differential entropy of C is defined as

$$h(C) = -\int p(c) \log p(c)\, dc.$$

The conditional differential entropy $h(C|X)$ is

$$h(C|X) = -\int p(x,c) \log p(c|x)\, dx\, dc.$$

The following properties of differential entropy will be used.

(1) Conditioning reduces entropy, $h(C|X) \leq h(C)$. Equality holds if C and X are independent.

(2) Differential entropy, $h(X)$, is maximized, over all distributions with the variance $Var(X) = \sigma^2$, by the normal distribution. If X had a Normal distribution, it would have entropy $\frac{1}{2}\log 2\pi e\sigma^2$. Hence, for any distribution of X with $Var(X) = \sigma^2$,

$$h(X) \leq \frac{1}{2}\log 2\pi e\sigma^2.$$

Further, if X is a vector of random variables, $X = [X_1 \ \ldots \ X_n]'$, then

$$h(X_1, \ldots, X_n) \leq \frac{1}{2}\log(2\pi e)^n|Z|,$$

where Z is the covariance matrix of X and $|Z|$ is the determinant of Z.

Next, the following properties of mutual information will be needed.

(1) Mutual information can be written in terms of differential entropies as

$$I(X; C) = h(C) - h(C|X).$$

(2) By the chain rule for mutual information,

$$I(X_1, X_2, \ldots X_n; C) = \sum_{i=1}^{n} I(X_i; C|X_1, \ldots X_{i-1}).$$

(3) By the data processing inequality, if $Y = f(C)$ for any (measurable) function f, then $I(C; X) \geq I(Y; X)$.

Finally, when the argument in $h(\cdot)$ is a vector of length n, for example, $C = [C_1, \ldots, C_n]'$, it is interpreted as the joint differential entropy $h(C_1, \ldots, C_n)$. Similarly, when the arguments in $I(\cdot;\cdot)$ are vectors of length n, for example C and X, it is to be interpreted as $I(C_1, \ldots, C_n; X_1, \ldots, X_n)$.

4.3 Proof of Main Theorem

In this section, we present the proof of Theorem 4.1.1. The core idea is to characterize the information flow between arbitrary "cut-sets" of the network. A cut divides the network into two sets, S and $S^c = \{1, \ldots n\}\backslash S$. Suppose that nodes 1 to m belong to set S and nodes $m + 1$ to n belong to set S^c. So, the estimates of the nodes in set S at time T are $X_S(T) = [X_1(T) \ \ldots \ X_m(T)]'$. The initial conditions of the nodes in sets S and S^c are denoted by $X_S(0) = [X_1(0) \ \ldots \ X_m(0)]'$ and $X_{S^c}(0) = [X_{m+1}(0) \ \ldots \ X_n(0)]'$.

The quantity that will play a central role in the proof of Theorem 4.1.1 is the mutual information term, $I(X_S(T); X_{S^c}(0)|X_S(0))$. This is mutual information between the estimates of the nodes in set S and the initial conditions of the nodes in set S^c, assuming that all nodes in S have each other's initial conditions. Leading up to the proof of Theorem 4.1.1, we prove 3 lemmas related to $I(X_S(T); X_{S^c}(0)|X_S(0))$.

In the first of our series of lemmas, we bound from above $I(X_S(T); X_{S^c}(0)|X_S(0))$, by the mutual information between the the inputs and the outputs of the channels that traverse the cut.

Lemma 4.3.1. For a given cut in the network, and corresponding cut-sets S^c and S,

$$I(X_S(T); X_{S^c}(0)|X_S(0)) \leq \sum_{l=1}^{N} I(V_S(l); U_{S^c}(l)|U_S(l)),$$

where N is the channel code block length, U_{S^c} is a vector of the variables transmitted by the encoders of the nodes in S^c and V_S is a vector of the variables received via channels by the decoders of the nodes in S. The (l) refers to the l^{th} channel use.

In the second lemma, we bound from above $I(V_S(l); U_{S^c}(l)|U_S(l))$ by the sum of the capacities of the channels traversing the cut.

Lemma 4.3.2. Suppose a network is represented by the graph $G = (V, E)$. The edges of the graph represent channels with positive capacity. If the channels connecting the nodes are memoryless and independent, then,

$$I(V_S(l); U_{S^c}(l)|U_S(l)) \leq \sum_{i \in S^c} \sum_{j \in S} \mathbf{C}_{ij}.$$

The proof of this lemma makes apparent the value of the conditioning in the mutual information terms. This conditioning is equivalent to assuming that all nodes in S have access to all information that is available at the nodes of the set S, including information about $X_S(0)$. In this way, we capture the information that is traversing the cut, without including the effect of information exchanged between nodes in the same set.

Finally, in the third lemma, we bound from below the term $I(X_S(T); X_{S^c}(0)|X_S(0))$. We show that this term is bounded from below by the information that must be communicated from the nodes of S^c to the nodes of S in order for the nodes of S to compute their estimates, $I(X_S(T); C_S|X_S(0))$. We then bound this from below by an expression that involves the desired performance criterion and the desired function.

For the mean square error criterion C1, we have the following lemma.

Lemma 4.3.3. If $E(\|X(T) - C\|^2) \leq \beta 2^{-\alpha}$ then

$$I(X_S(T); X_{S^c}(0)|X_S(0)) \geq L(S)$$

where,

$$L(S) = h(C_S|X_S(0)) - \frac{|S|}{2} \log 2\pi e \beta + \frac{|S|}{2} \log |S| + |S| \frac{\alpha}{2},$$

and, $|S|$ is the size of the set S, specifically, $|S| = m$.

The lower bound involves two terms. These are (1) the desired accuracy in the nodes' estimates, specified by the mean square error criterion, and (2) the uncertainty in the function to be estimated, C_S, quantified by its differential entropy. The larger the desired accuracy, the larger the α in the mean square error criterion. This implies a larger lower bound on the information that must be conveyed. Also, the larger the uncertainty in the function to be learned by the nodes in set S, the larger the differential entropy term. Hence, the lower bound is larger.

For the mean square error criterion C2, we have the following corollary.

Corollary 4.3.4. If, for all $i \in \{1, \ldots, n\}$, $E(X_i(T) - C_i)^2 \leq \beta 2^{-\alpha}$, then,

$$I(X_S(T); X_{S^c}(0)|X_S(0)) \geq \bar{L}(S),$$

where $\bar{L}(S) = h(C_S|X_S(0)) - \frac{|S|}{2}\log 2\pi e\beta + |S|\frac{\alpha}{2}$.

When, for all i, $E(X_i(T) - C_i)^2 \leq \beta 2^{-\alpha}$, we again have a lower bound that depends on the desired accuracy and the uncertainty in the function to be estimated. However, $\bar{L}(S)$ is smaller than $L(S)$ due to the weaker error requirement of C2.

In the next sections, we prove the above lemmas. Then, we prove Theorem 4.1.1.

4.3.1 Proof of Lemma 4.3.1

We prove the following inequality:

$$I(X_S(T); X_{S^c}(0)|X_S(0)) \leq \sum_{l=1}^{N} I(V_S(l); U_{S^c}(l)|U_S(l)), \tag{4.1}$$

where N is the channel code block length, U_{S^c} is a vector of the variables transmitted by the encoders of the nodes in S^c and V_S is a vector of the variables received via channels by the decoders of the nodes in S.

Recall that by the assumptions we made in our problem formulation, the end of the k^{th} time slot corresponds to time T. So, $X_S(T) = X_S(k)$. In this proof, it is convenient to use $X_S(k)$. We need to refer to the sequence of estimates at node i up until time T. The most natural way is to enumerate using integers: $X_i^k = \{X_i(1), X_i(2), \ldots X_i(k)\}$.

For this proof, we use the general formulation for multi-terminal networks of [5, section 14.10]. let U_i be transmitted by the node i encoder and V_i be received by the node i decoder. We denote a sequence of length N transmitted by i as $U_i^N = (U_i(1), U_i(2), \ldots U_i(N))$. The indices in brackets represent channel use. As before, if nodes 1 to m belong to S, we have that $V_S = (V_1, \ldots, V_m)$. Similarly, we have that $V_S(l) = (V_1(l), \ldots, V_m(l))$, representing the variables received after the l-th use of the channel.

We assume that the the estimates at node i, X_i^k, are a function of the received messages at that node, V_i^N and its own data, $X_i(0)$, $X_i^k = \phi_i(V_i^N, X_i(0))$. The message transmitted by i in the l^{th} channel use, $U_i(l)$, is also a function of of the received messages at that node, V_i^{l-1} and its own data, $X_i(0)$, $U_i(l) = \psi_i(V_i^{l-1}, X_i(0))$.

As in [5], the channel is a memoryless discrete-time channel. In our case, for convenience, we assume the channel to be continuous, represented by the conditional probability distribution function $p(v_1, \ldots, v_n|u_1, \ldots, u_n)$. However, we note that the inequalities below hold even in the case that the channel is discrete. In this case, the random variable arguments of $I(\cdot;\cdot|\cdot)$ would be arbitrary ensembles, and so we use the general definition for $I(\cdot;\cdot|\cdot)$ as the "average conditional information" in [29, Ch.3], and for the conditional entropy, $h(X|Y)$, we use $h(X|Y) = I(X;X|Y)$. All the equalities and inequalities below will continue to hold. We refer the reader to [29, Ch.3] for technical details.

The following inequalities proceed in the same manner as Theorem 14.10.1 in [5]. For convenience, we repeat the steps here using our notation.

$$\begin{aligned}
&I(X_S(k); X_{S^c}(0)|X_S(0)) \\
&\overset{(a)}{\leq} I(X_S(1), \ldots, X_S(k); X_{S^c}(0)|X_S(0)) \\
&= I(X_S(1), \ldots, X_S(k), X_S(0); X_{S^c}(0)|X_S(0))
\end{aligned}$$

$$
\begin{aligned}
&\overset{(b)}{\leq} I(V_1^N, \ldots, V_m^N, X_S(0); X_{S^c}(0)|X_S(0)) \\
&= I(V_S(1), \ldots, V_S(N); X_{S^c}(0)|X_S(0)) \\
&\overset{(c)}{=} \sum_{l=1}^{N} I(V_S(l); X_{S^c}(0)|X_S(0), V_S(l-1), \ldots, V_S(1)) \\
&\overset{(d)}{=} \sum_{l=1}^{N} h(V_S(l)|X_S(0), V_S(l-1), \ldots, V_S(1)) \\
&\qquad - h(V_S(l)|X_{S^c}(0), X_S(0), V_S(l-1), \ldots, V_S(1)) \\
&\overset{(e)}{\leq} \sum_{l=1}^{N} h(V_S(l)|X_S(0), V_S(l-1), \ldots V_S(1), U_S(l)) \\
&\qquad - h(V_S(l)|X_{S^c}(0), X_S(0), V_S(l-1), \ldots, V_S(1), U_S(l), U_{S^c}(l)) \\
&\overset{(f)}{\leq} \sum_{l=1}^{N} h(V_S(l)|U_S(l)) - h(V_S(l)|U_S(l), U_{S^c}(l)) \\
&\overset{(g)}{=} \sum_{l=1}^{N} I(V_S(l); U_{S^c}(l)|U_S(l)).
\end{aligned}
$$

Above,

(a) holds by the data processing inequality,

(b) holds again by the data processing inequality, because $X_i^k = \phi_i(V_i^N, X_i(0))$,

(c) follows by the chain rule for mutual information,

(d) follows by the definition of mutual information, (or, in the discrete channel case, it follows by Kolmogorov's formula [29, Ch.3] and by noting that the entropy term is well-defined since V_i would take values in a discrete set),

(e) follows, for the first term, because $U_i(l) = \psi_i(V_i^{l-1}, X_i(0))$, so it does not change the conditioning; and the second part follows because conditioning reduces entropy,

(f) holds, for the first term, because conditioning reduces entropy, and for the second term, because the channel output depends only on the current input symbols,

(g) from the definition of mutual information.

□

4.3.2 Proof of Lemma 4.3.2

In this lemma, we consider a network that is represented by the graph $G = (V, E)$. The edges of the graph represent channels with positive capacity. If the channels connecting the nodes are memoryless and independent, we show that,

$$
I(V_S(l); U_{S^c}(l)|U_S(l)) \leq \sum_{i \in S^c} \sum_{j \in S} \mathbf{C}_{ij}.
$$

For simplicity of notation in the rest of the proof, we omit the braces after the random variables, (l). For example, instead of $V_S(l)$ we write V_S.

As we had in the previous lemma, U_i is transmitted by the node i encoder. Previously, we had not specified which nodes will receive this code letter. In our set up, however, there is a dedicated channel between every two nodes that have an edge between them. So, the transmitter at node i will send out codewords to each of the neighbors of i, that is all j, such that $(i,j) \in E$. We denote the encoder's code letter from i to j as U_{ij}. U_i represents all messages transmitted by the encoder of node i. So, $U_i = \{U_{ij}\}$, for all j, such that $(i,j) \in E$.

Similarly, V_i is received by the node i decoder. It consists of all the digits received by i from its neighbors, all j such that $(j,i) \in E$. If there is a link from node j to i, the code letter from node j arrives at the decoder of i through a channel. We denote the digit received at i from j as V_{ji}. V_i represents all the received messages; so, $V_i = \{V_{ji}\}$, for all j, such that $(j,i) \in E$.

In order to make our notation in the proof simpler, we introduce dummy random variables. In particular, we will use U_{ij} and V_{ij} even if $(i,j) \notin E$. Effectively, we are introducing a link between nodes i and j. But, in this case, we set $\mathbf{C}_{ij} = 0$. So now, we let $U_i = \{U_{i1}, \ldots, U_{in}\}$ and $V_i = \{V_{1i}, \ldots, V_{ni}\}$.

The key to the proof is the memorylessness and independence of the channels. That is, the output of a channel at any instant, $V_{ij}(l)$, depends only on the channel input at that instant, $U_{ij}(l)$. Because of this, we have that

$$I(V_S; U_{S^c}|U_S) \leq \sum_{i \in S^c} \sum_{j \in S} I(V_{ij}; U_{ij}).$$

To obtain this expression, we express the mutual information in terms of the entropy,

$$I(V_S; U_{S^c}|U_S) = h(V_S|U_S) - h(V_S|U_{S^c}, U_S).$$

Next, we express the entropy terms using the chain rule. We assume that nodes 1 to m belong to set S and nodes $m+1$ to n belong to S^c. Then,

$$h(V_S|U_S) = \sum_{j=1}^{m} h(V_j|V_{j-1}, \ldots, V_1, U_S),$$

and,

$$h(V_S|U_{S^c}, U_S) = \sum_{j=1}^{m} h(V_j|V_{j-1}, \ldots, V_1, U_{S^c}, U_S).$$

Because conditioning reduces entropy, we have that

$$h(V_S|U_S) \leq \sum_{j=1}^{m} h(V_j|U_S).$$

For every channel, given its input, the channel output is independent of all other channel outputs. So,

$$h(V_S|U_{S^c}, U_S) = \sum_{j=1}^{m} h(V_j|U_{S^c}, U_S).$$

Combining the two inequalities, we have,

$$I(V_S; U_{S^c}|U_S) \leq \sum_{j=1}^{m} h(V_j|U_S) - h(V_j|U_{S^c}, U_S).$$

Now, let $j = 1$ and consider the expression $h(V_1|U_S) - h(V_1|U_{S^c}, U_S)$. Recall that we have assumed that $V_1 = \{V_{11}, \ldots, V_{n1}\}$. Also, we have that $U_i = \{U_{i1}, \ldots, U_{in}\}$. So, U_S includes $\{U_{11}, \ldots, U_{m1}\}$.

For the first differential entropy term we have the following sequence of inequalities.

$$\begin{aligned} h(V_1|U_S) &\overset{(a)}{=} \sum_{i=1}^{n} h(V_{i1}|V_{(i-1)1}, \ldots, V_{11}, U_S) \\ &\overset{(b)}{=} \sum_{i=1}^{m} h(V_{i1}|U_{i1}) + \sum_{i=m+1}^{n} h(V_{i1}|V_{(i-1)1}, \ldots, V_{11}, U_S) \\ &\overset{(c)}{\leq} \sum_{i=1}^{m} h(V_{i1}|U_{i1}) + \sum_{i=m+1}^{n} h(V_{i1}), \end{aligned}$$

where,

(a) follows by the chain rule,

(b) follows because the channels are independent; so, given U_{i1}, V_{i1} is independent of all of the other random variables,

(c) holds because conditioning reduces entropy.

Next, observe that

$$\begin{aligned} h(V_1|U_{S^c}, U_S) &\overset{(d)}{=} \sum_{i=1}^{n} h(V_{i1}|V_{(i-1)1}, \ldots, V_{11}, U_{S^c}, U_S) \\ &\overset{(e)}{=} \sum_{i=1}^{n} h(V_{i1}|U_{i1}), \end{aligned}$$

where,

(d) follows by the chain rule,

(e) follows because the channels are independent; so, given U_{i1}, V_{i1} is independent of all of the other random variables.

Finally, combining these inequalities,

$$\begin{aligned} h(V_1|U_S) - h(V_1|U_{S^c}, U_S) &\leq \sum_{i=m+1}^{n} h(V_{i1}) - h(V_{i1}|U_{i1}) \\ &= \sum_{i=m+1}^{n} I(V_{i1}; U_{i1}). \end{aligned}$$

Hence we have the desired expression,

$$I(V_S; U_{S^c}|U_S) \leq \sum_{i \in S^c} \sum_{j \in S} I(V_{ij}; U_{ij}).$$

Finally, to complete the proof, we note that

$$I(V_{ij}; U_{ij}) \leq \mathbf{C}_{ij}.$$

This is because, by definition,

$$\mathbf{C}_{ij} = \max I(V_{ij}; U_{ij}),$$

where the maximum is taken over all distributions of the channel input, U_{ij}. □

4.3.3 Proof of Lemma 4.3.3 and Corollary 4.3.4

Recall that the lemma stated that if $E(\|X(T) - C\|^2) \leq \beta 2^{-\alpha}$ then

$$I(X_S(T); X_{S^c}(0)|X_S(0)) \geq L(S)$$

where,

$$L(S) = h(C_S|X_S(0)) - \frac{|S|}{2} \log 2\pi e\beta + \frac{|S|}{2} \log |S| + |S|\frac{\alpha}{2},$$

and, $|S|$ is the size of the set S, specifically, $|S| = m$.

We start the proof by observing the following.

$$\begin{aligned} &I(X_S(T); X_{S^c}(0)|X_S(0)) \\ &\overset{(a)}{=} I(X_S(T); X(0)|X_S(0)) \\ &\overset{(b)}{\geq} I(X_S(T); C_S|X_S(0)) \end{aligned}$$

where

(a) that is, $I(W; Y, U|U) = I(W; Y|U)$, can be verified by the chain rule for mutual information:

$$\begin{aligned} I(W; Y, U|U) &= I(W; Y|U) + I(W; U|U, Y) \\ &= I(W; Y|U), \end{aligned}$$

because $I(W; U|U, Y) = 0$.

(b) follows by the data processing inequality, because $C_i = f_i(X(0))$.

Second, we obtain a lower bound on $I(X_S(T); C_S|X_S(0))$ in terms of the desired mean square criterion. We have the following series of inequalities.

$$\begin{aligned} I(X_S(T); C_S|X_S(0)) &= h(C_S|X_S(0)) - h(C_S|X_S(T), X_S(0)) \\ &= h(C_S|X_S(0)) - h(C_S - X_S(T)|X_S(T), X_S(0)) \\ &\overset{(c)}{\geq} h(C_S|X_S(0)) - h(C_S - X_S(T)) \end{aligned} \quad (4.2)$$

where, (c) follows because conditioning reduces entropy.

Now, because the multivariate normal maximizes entropy over all distributions with the same covariance,

$$h(X_S(T) - C_S) \leq \frac{1}{2}\log(2\pi e)^m|Z|, \tag{4.3}$$

where, Z is a covariance matrix whose diagonal elements are $Z_{ii} = Var(X_i(T) - C_i)$, and $|Z|$ denotes the determinant. Recall that S is the set containing nodes 1 to m, so it has size m. Also, $X_S(T) - C_S$ is a vector of length m. So, Z is an m by m matrix. Now,

$$\begin{aligned} |Z| &\overset{(d)}{\leq} \prod_{i=1}^{m} Var(X_i(T) - C_i) \\ &\leq \prod_{i=1}^{m} E(X_i(T) - C_i)^2 \\ &\overset{(e)}{\leq} \left(\frac{\beta 2^{-\alpha}}{m}\right)^m. \end{aligned} \tag{4.4}$$

Here, (d) follows due to Hadamard's inequality. To see (e), we have the following proposition, which is shown in the Appendix.

Proposition A.1.1 For $\gamma > 0$, subject to $\sum_{i=1}^{m} y_i \leq \gamma$ and $y_i \geq 0$, $\prod_{i=1}^{m} y_i$ is maximized when $y_i = \frac{\gamma}{m}$.

Now, (e) follows by setting $y_i = E(X_i(T) - C_i)^2$ and observing that

$$\begin{aligned} \sum_{i=1}^{m} y_i &= E(\|X_S(T) - C_S\|^2) \\ &\leq E(\|X(T) - C\|^2) \\ &\leq \beta 2^{-\alpha}, \end{aligned}$$

where the last inequality follows by the assumption of our lemma.

Finally, using (4.4) and (4.3), we bound (4.2) from below and obtain $L(S)$. □

Proof of Corollary 4.3.4. Recall that in this corollary, we had the weaker condition that for all $i \in \{1, \ldots, n\}$, $E(X_i(T) - C_i)^2 \leq \beta 2^{-\alpha}$. In this case, we show that we have the smaller lower bound,

$$\bar{L}(S) = h(C_S|X_S(0)) - \frac{|S|}{2}\log 2\pi e\beta + |S|\frac{\alpha}{2}.$$

To see this, observe that $E(X_i(T) - C_i)^2 \leq \beta 2^{-\alpha}$ implies $E(\|X_S(T) - C_S\|^2) \leq |S|\beta 2^{-\alpha}$. So, replacing β in $L(S)$ of the previous lemma by $|S|\beta$ yields the desired result. □

4.3.4 Proof of Theorem 4.1.1

The proof proceeds in several steps. First, as shown in Lemma 4.3.1, for a given cut in the network and corresponding cut-sets S^c and S,

$$I(X_S(T); X_{S^c}(0)|X_S(0)) \leq \sum_{l=1}^{N} I(V_S(l); U_{S^c}(l)|U_S(l)), \tag{4.5}$$

where N is the channel code block length, U_{S^c} is a vector of the variables transmitted by the encoders of the nodes in S^c and V_S is a vector of the variables received via channel by the decoders of the nodes in S.

Second, by Lemma 4.3.2, because we have assumed that the channels connecting the nodes are memoryless and independent,

$$I(V_S(l); U_{S^c}(l)|U_S(l)) \leq \sum_{i \in S^c} \sum_{j \in S} \mathbf{C}_{ij}. \tag{4.6}$$

Third, we combine equations (4.5) and (4.6) with Corollary 4.3.4 to obtain

$$N \geq \frac{\bar{L}(S)}{\sum_{i \in S^c} \sum_{j \in S} \mathbf{C}_{ij}}, \tag{4.7}$$

Finally, we have that

$$T \geq \max_{S \subset V} \frac{\bar{L}(S)}{\sum_{i \in S^c} \sum_{j \in S} \mathbf{C}_{ij}},$$

because (i) (4.7) holds for any cut, and, (ii) by assumption, each of the channels transmits one symbol per second, so $T = N$.

4.4 A Technical Difficulty and its Resolution

Making use of the lower bounds derived above involves computing the differential entropy of the random variables to be learned in the network, specifically, $h(C_S|X_S(0))$, where $C_S = [C_1 \ \ldots \ C_m].'$ If the C_i's are different random variables, then the differential entropy term is well-defined. However, if two entries of C_S are the same random variable, for example if both are $f(X(0))$, then $h(C_S|X_S(0))$ will be $-\infty$.

One way to avoid this looseness is by viewing all the nodes in the set S as one node. Then we can use the Information Theoretic technique to derive a new lower bound for this scenario. This will result in the new lower bound not recognizing that the information traversing the cut is destined to $|S|$ nodes.

However, the lower bound derived in this chapter can be used capture the effect of $|S|$. To do this, we introduce auxiliary random variables associated with the nodes of set S^c, to be learned by nodes in S. This technique will be used in the next chapter. In the examples below, we demonstrate the computation of $h(C_S|X_S(0))$ when we introduce the auxiliary random variables.

Example 4.4.1 (The Solution). Let nodes $\{1, \ldots, m\}$, $m \leq n/2$, belong to set S, so that $C_S = [C_1 \ \ldots \ C_m].'$ Let $C_1 = f(X(0))$ and $C_i = f(X(0)) + a_i \epsilon_{j_i}$ for $i \in \{2, \ldots, m\}$. One can think of ϵ_{j_i} being associated with a node in set S^c, that is, $j_i \in \{m+1, \ldots n\}$. So, node j_i's initial condition would be $(X_{j_i}(0), \epsilon_{j_i})$.

Furthermore, we assume that f is separable, meaning $f(X(0)) = f_S(X_S(0)) + f_{S^c}(X_{S^c}(0))$. Finally, we assume that the $X_i(0)$'s and ϵ_i's are mutually independent. Then,

$$\begin{aligned}
&h(C_S|X_S(0)) \\
=&h\left(f_{S^c}(X_{S^c}(0)), f_{S^c}(X_{S^c}(0)) + a_2\epsilon_{j_2}, \ldots, f_{S^c}(X_{S^c}(0)) + a_m\epsilon_{j_m}|X_S(0)\right) \\
\overset{(a)}{=}&h(f_{S^c}(X_{S^c}(0)), f_{S^c}(X_{S^c}(0)) + a_2\epsilon_{j_2}, \ldots, f_{S^c}(X_{S^c}(0)) + a_m\epsilon_{j_m}) \\
\overset{(b)}{=}&h(f_{S^c}(X_{S^c}(0))) + \sum_{i=2}^{m} h(a_i\epsilon_{j_i}) \\
\overset{(c)}{=}&h(f_{S^c}(X_{S^c}(0))) + \sum_{i=2}^{m} h(\epsilon_{j_i}) + \log \prod_{i=2}^{m} |a_i|,
\end{aligned}$$

where,

(a) follows because we have assumed that the $X_i(0)$'s and ϵ_i's are mutually independent,

(b) follows by the chain rule for differential entropy, and again using the fact that the $X_i(0)$'s and ϵ_i's are mutually independent,

(c) follows using the fact that $h(a_i\epsilon_{j_i}) = h(\epsilon_{j_i}) + \log|a_i|$, as shown in shown in [5, Ch.9].

In the next example, we assume that the function f is a linear function and that the auxiliary random variables are independent Gaussian random variables. For this scenario, we then obtain the expression for the lower bound of Corollary 4.3.4.

Example 4.4.2 (Using the Solution for a Linear Function). In addition to the assumptions in Example 4.4.1, let $f(X(0)) = \sum_{j=1}^{n} \beta_j X_j(0)$. We assume that $\epsilon_{j2}, \ldots, \epsilon_{j_m}$ are independent and identically distributed Gaussian random variables, with mean zero and variance η. Then, the differential entropy of ϵ_{j_i} is $h(\epsilon_{j_i}) = \frac{1}{2}\log 2\pi e\eta$.

So, substituting in the expression from Example 4.4.1, we have that

$$h(C_S|X_S(0)) = h\left(\sum_{j\in S^c} \beta_j X_j(0)\right) + \frac{m-1}{2}\log 2\pi e\eta + \log\prod_{i=2}^{m}|a_i|. \tag{4.8}$$

To evaluate $h\left(\sum_{j\in S^c} \beta_j X_j(0)\right)$, we use the Entropy Power Inequality, namely, for independent $X_i(0)$'s,

$$2^{2h\left(\sum_{j\in S^c}\beta_j X_j(0)\right)} \geq \sum_{j\in S^c} 2^{2h(\beta_j X_j(0))},$$

which implies that

$$h\left(\sum_{j\in S^c} \beta_j X_j(0)\right) \geq \frac{1}{2}\log\left(\sum_{j\in S^c} 2^{2h(\beta_j X_j(0))}\right).$$

Now, if we assume that each $X_i(0)$ is uniformly distributed in the interval between 1 and $B+1$, $X_i(0) \sim U[1, B+1]$, then,

$$h\left(\beta_j X_j(0)\right) = \log|\beta_j|B.$$

So,

$$h\left(\sum_{j\in S^c}\beta_j X_j(0)\right) \geq \frac{1}{2}\log\left(B^2\sum_{j\in S^c}\beta_j^2\right) = \log B + \frac{1}{2}\log\sum_{j\in S^c}\beta_j^2. \tag{4.9}$$

Finally, we evaluate the lower bound of Corollary 4.3.4 for this scenario. Recall that we had

$$\bar{L}(S) = h(C_S|X_S(0)) - \frac{|S|}{2}\log 2\pi e\beta + |S|\frac{\alpha}{2},$$

and $|S| = m$. Assuming that $\beta = 1$ and using equation (4.8) together with the inequality of equation (4.9), we have that

$$\bar{L}(S) \geq \log\frac{B\left(\sum_{j\in S^c}\beta_j^2\right)^{\frac{1}{2}}\prod_{i=2}^{m}|a_i|}{\sqrt{2\pi e\eta}} + \frac{m}{2}\left(\alpha + \log\eta\right). \tag{4.10}$$

4.5 Summary

In summary, our use of basic Information Theoretic definitions and inequalities has led to a lower bound that we have applied to a formulation for distributed function computation. The lower bound on information consists of a term that arises due to the mean square error criterion and a term due to the function that is to be estimated. Using techniques of Network Information Theory, we have shown how the bound on information can be used to obtain a lower bound on computation time time.

In the next chapter, we use the techniques of this chapter to find a lower bound on computation time when nodes compute a sum via erasure channels. We present a distributed algorithm for computation in this scenario and provide an upper bound the run-time of the algorithm. Both bounds depend inversely on conductance, which captures the limitations due to the network topology. Therefore, we conclude that our lower bound is tight in capturing the effect of the network topology via the conductance.

Chapter 5

A Tight Bound: Computation of the Sum via Erasure Channels

In this chapter we apply the lower bound developed in Chapter 4 to a specific scenario where we find our bounds to be asymptotically tight. Specifically, we consider a scenario nodes are required to learn a linear combination of the initial values in the network while communicating over erasure channels. Our lower bound suggests that in this scenario, the computation time depends reciprocally on "conductance." Conductance essentially captures the information-flow bottle-neck that arises due to topology and channel capacities. The more severe the communication limitations, the smaller the conductance.

To establish the tightness of our lower bound, we describe an algorithm whose computation time matches the lower bound. The algorithm that we describe here can in fact be more generally used for distributed computation of separable functions, a special case of which is the sum. The desired function, a sum, is simple, and the algorithm that we describe has computational demands that are not severe. So, the time until the performance criterion is met using this algorithm is primarily constrained by the limitations on communication.

Indeed, we show that the upper bound, on the time until this algorithm guarantees the performance criterion, depends reciprocally on conductance. Hence, we conclude that that a lower bound we derive using Information Theoretic analysis is tight in capturing the limitations due to the network topology. Alternatively, one can interpret this tightness as the fact that the algorithm we describe here is the fastest with respect to its dependence on the network topology, as quantified by the conductance.

In the next section, we describe the problem formulation and the main results of this chapter. In section 5.3 we derive the lower bound that scales reciprocally with conductance. In section 5.4 we describe an algorithm that can be used to compute the sum via erasure channels. We derive an upper bound on its computation time; we show that this upper bound also scales inversely with conductance.

5.1 Problem Formulation

A network consists of n nodes, each having a random initial condition, denoted by the random variable $X_i(0)$. Suppose the initial values at the nodes are independent and uniformly distributed, $X_i(0) \sim U[1, B+1]$. Each node is required to compute a linear function of all the initial conditions, $C = \sum_{j=1}^{n} \beta_j X_j(0)$. Node i's estimate of C at time k is denoted as $X_i(k)$.

The nodes communicate via noisy channels is described by a graph, $G = (V, E)$, where V is the set of nodes and E is the set of edges, (i, j). If node i communicates with node j via channel with capacity $\mathbf{C}_{ij} > 0$, then $(i, j) \in E$. If $(i, j) \notin E$, we set $\mathbf{C}_{ij} = 0$. We assume that the graph is connected.

We assume that all channels in the network are independent memoryless discrete-time. Further, for each of the channels, one channel symbol is sent per second. We assume that the channels are symmetric, $\mathbf{C}_{ij} = \mathbf{C}_{ji}$. Furthermore, they are erasure channels, so that $C_{ij} = p_{ij}$, where p_{ij} is the probability node j receives node i's bit without error. The matrix $P = [p_{ij}]$ is a doubly stochastic matrix that captures the communication limitations due to the channels.

The conductance of a graph, $\Phi(G)$, is a property that captures the bottle-neck of information flow. It depends on the the connectivity, or topology, of the graph, and the magnitudes of the channel capacities. It is defined as

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

In the case of erasure channels, we substitute in $C_{ij} = p_{ij}$, and we denote the conductance as $\Phi(P)$.

5.1.1 Main Results

Consider any algorithm that guarantees that for any realization of the initial values, with high probability each node has an estimate within $1 \pm \varepsilon$ of the true value of C, at time T. The Information Theoretic lower bound maintains that such algorithm must have a computation time, T, that is inversely proportional to conductance.

Theorem 5.1.1. *Nodes communicate in order for each node to compute a linear combination of all initial values in the network. Let A represent a realization of the initial conditions, $A = \{X_1(0) = x_1, \dots, X_n(0) = x_n\}$. Any algorithm that guarantees that for all $i \in \{1, \dots, n\}$,*

$$\mathbf{P}\left(|X_i(T) - C| \leq \varepsilon C \middle| A\right) \geq 1 - \delta,$$

must have

$$T \geq \frac{1}{\Phi(G)} \log \frac{1}{B\varepsilon^2 + \frac{1}{B}^{\frac{2}{n}}},$$

where, $B\varepsilon^2 \in \left[0, 1 - \frac{1}{B}^{\frac{2}{n}}\right]$.

Next, we provide an algorithm that guarantees, with high probability, the nodes' estimates are within the desired ε-error interval around the true value of the sum. We provide an upper bound on this algorithm's computation time. The computation time is inversely proportional to conductance.

Theorem 5.1.2. *Let P be a stochastic and symmetric matrix for which if $(i, j) \notin E$, $p_{ij} = 0$. Suppose that node i has an initial condition, x_i. There exists a distributed algorithm $\mathcal{AP}^{\mathcal{Q}}$ by which nodes compute a linear sum, $f(x, V) = \sum_{j=1}^{n} \beta_j x_j$, via communication of quantized messages. The quantization error will be no more than a given $\gamma = \Theta(\frac{1}{n})$, and for any*

$\varepsilon \in (\gamma f(x,V), \gamma f(x,V) + \frac{1}{2})$ and $\delta \in (0,1)$, the computation time of the algorithm will be

$$T^{cmp}_{\mathcal{APQ}}(\varepsilon, \delta) = O\left(\varepsilon^{-2}(1 + \log \delta^{-1}) \frac{(\log n + \log \delta^{-1}) \log n}{\Phi(P)}\right),$$

where, $\Phi(P)$ is the conductance evaluated for $\mathbf{C}_{ij} = p_{ij}$.

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

$$T^{\text{cmp}}_{\mathcal{APQ}}\left(\varepsilon, \frac{1}{n^2}\right) = O\left(\varepsilon^{-2} \frac{\log^3 n}{\Phi(P)}\right).$$

The computation time of this algorithm depends on the network topology, via the conductance of the graph, in the same reciprocal manner manifested by the lower bound. Thus, we conclude that the lower bound is tight in capturing the effect of the network topology on computation time. Conversely, the algorithm's running time is optimal with respect to its dependence on the network topology, as captured by the conductance.

5.2 Motivation: Capturing the Effect of Topology

The conductance of a graph, $\Phi(G)$, is a property that captures the bottle-neck of information flow. It depends on the the connectivity, or topology, of the graph, and the magnitudes of the channel capacities. The more severe the network constraints, the smaller the conductance. As we will see later in this chapter, it is also related to time it takes for information to spread in a network; the smaller the conductance, the longer it takes.

Example 5.2.1 (Conductance for Two Topologies). Consider two networks, each has n nodes. We calculate conductance for two extreme cases of connectivity shown in Figure 5-1. On the one hand, we have severe topological constraints: a ring graph. Each node may contact only the node on its left or the node on its right. On the other hand, we have a case of virtually no topological constraints: a fully connected graph. Each node may contact every other node in the network.

For the purpose of illustrating the computation of conductance for the two topologies, suppose that in both cases, the links from a given node to different nodes are equally weighted. So, for the ring graph, let $\mathbf{C}_{ij} = \mathbf{C} = \frac{1}{4}$, for all $i \neq j$; for the fully connected graph, let $\mathbf{C}_{ij} = \mathbf{C} = \frac{1}{n}$, for all $i \neq j$. Assume that for the ring graph, $\mathbf{C}_{ii} = \frac{1}{2}$. If the channels were erasure channels, this would be the probability that node i makes contact with no other nodes. For the fully connected graph, let $\mathbf{C}_{ij} = \frac{1}{n}$. So, in both cases, we have that the sum of the capacities of channels leaving a node is 1, $\sum_j \mathbf{C}_{ij} = 1$.

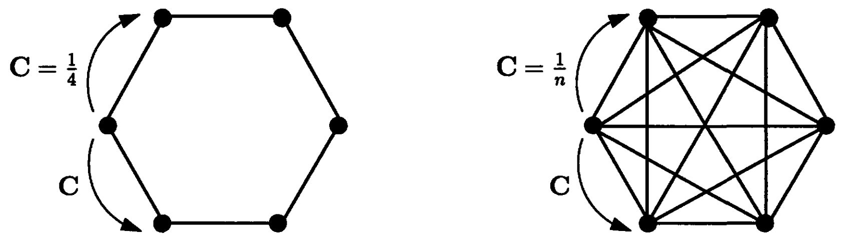

Figure 5-1: Two ways to connect six nodes: a ring graph and a fully connected graph.

Now, we compute the conductance of the ring graph. Recall that conductance is

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

Consider any cut that divides the ring graph into two sets, S and S^c. For any such cut, there will be exactly two links crossing the cut, going from S to S^c. So, $\sum_{i\in S,j\notin S} \mathbf{C}_{ij} = \frac{1}{2}$, and

$$\Phi(G) = \min_{\substack{S\subset V \\ 0<|S|\leq n/2}} \frac{\frac{1}{2}}{|S|}.$$

Since we minimize over all cuts such that $|S| \leq n/2$, the ratio is minimized when the cut divides the ring into two sets of equal size, and $|S| = n/2$. So, $\Phi(G) = \frac{1}{n}$.

Next, we compute the conductance of the fully connected graph. Consider any cut that divides the graph into two sets, S and S^c. For any such cut, there will be $|S||S^c|$ links crossing the cut, going from S to S^c. So,

$$\begin{aligned} \frac{\sum_{i\in S,j\notin S} \mathbf{C}_{ij}}{|S|} &= \frac{|S||S^c|\frac{1}{n}}{|S|} \\ &= \frac{|S^c|}{n} \\ &= \frac{n-|S|}{n} \end{aligned}$$

The last equality is minimized where $|S| = n/2$, so, $\Phi(G) = \frac{1}{2}$.

So, for two networks with the same number of nodes, the network with the more severe topological constraints has smaller conductance. In general, for a ring graph, we have $\Phi(G) = O(\frac{1}{n})$, while for a fully connected graph we have $\Phi(G) = O(1)$.

The popular algorithms for computing a linear function of initial conditions, such as averaging and consensus, are based on linear iterations. The convergence of such iterative algorithms depends on a reversible (or symmetric) and graph conformant matrix P. Usually, the running time of these algorithms scales like $\frac{1}{\Phi(P)^2}$. Specifically, for a ring graph it is $\frac{1}{\Phi(P)^2} \approx n^2$, which means roughly n^2 iterations are needed for the algorithm to converge. In section 5.4, we describe an algorithm that does not use linear computations, and its run-time scales like $\frac{1}{\Phi(P)}$. So, for a ring, roughly n iterations are needed. In the next section, we show that the run-time necessarily scales like $\frac{1}{\Phi(P)}$. So, for computation over a ring graph, n iterations are both sufficient and necessary. More generally, our algorithm scales optimally with respect to the graph topology.

5.3 The Information Theoretic Lower Bound

In this section, we provide the proof of Theorem 5.1.1. We will use the techniques that we have developed in Chapter 4. In particular, we will use the results of Examples 4.4.1 and 4.4.2, namely equation (4.10).

Proof of Theorem 5.1.1. Recall that $C = \sum_{j=1}^{n} \beta_j X_j(0)$. Suppose that we have any realization of the initial conditions, $A = \{X_1(0) = x_1, \ldots, X_n(0) = x_n\}$. We are given an algorithm that guarantees, for every such realization, that at time T each node, i, has an estimate, $X_i(T)$, of C: $\sum_{j=1}^{n} \beta_j x_j$. Furthermore, for this algorithm, the estimate $X_i(T)$ is within an ε-interval of the true value of C, with desired probability. That is,

$$\mathbf{P}\left(|X_i(T) - C| \leq \varepsilon C \middle| A\right) \geq 1 - \delta. \tag{5.1}$$

The proof proceeds in several steps. The proofs for steps 1 and 2 follow this proof.

1. Any algorithm that satisfies the probability condition of equation (5.1) must satisfy, for small enough δ, a mean square error criterion:

$$E(X_i(T) - C)^2 \leq \varepsilon^2 E(C^2).$$

2. Let $C_1 = C$ and $C_i = C + a\epsilon_{j_i}$ for $i \in \{2, \dots, m\}$, where $\epsilon_{j_2}, \dots, \epsilon_{j_m}$ are independent and identically distributed Gaussian random variables, with mean zero and variance η. Let the ϵ_{j_i}'s be independent of the initial conditions, $X_i(0)$. Then,

$$E(X_i(T) - C_i)^2 \leq \varepsilon^2 E(C^2) + a^2\eta.$$

3. Next, let S^* and $(S^*)^c$ be the sets for which

$$\frac{\sum_{i \in S, j \notin S} \mathbf{C}_{ij}}{|S|}$$

is minimized, and assume S^* is the set with smaller size, $|S^*| \leq \frac{n}{2}$. For purposes of this proof, we enumerate the nodes in set S^* from 1 to m. Then, let $C_{S^*} = [C_1 \ \dots \ C_m]'$, where the C_i's are those of Step 2.

4. Now, we can apply our Information Theoretic inequalities to this set-up. We think of ϵ_{j_i} being associated with a node in set $(S^*)^c$, that is, $j_i \in \{m+1, \dots n\}$. So, node j_i's initial condition would be $(X_{j_i}(0), \epsilon_{j_i})$. Denote $[\epsilon_{j_1} \ \dots \ \epsilon_{j_m}]$ by ϵ. Using the derivations of Chapter 4, we have that

$$\begin{aligned} T \sum_{i \in (S^*)^c} \sum_{j \in S^*} \mathbf{C}_{ij} &\geq I(X_{S^*}(T); X_{(S^*)^c}(0) | X_{S^*}(0)) \\ &\overset{(a)}{=} I(X_{S^*}(T); X_{(S^*)^c}(0), \epsilon | X_{S^*}(0)) \\ &\geq I(X_{S^*}(T); C_{S^*} | X_{S^*}(0)) \\ &\geq \bar{L}(S^*), \end{aligned}$$

where, (a) follows because $X_{S^*}(T)$ is the vector of estimates produced by the algorithm, and depends on the initial conditions, $X_i(0)$'s, while the ϵ_{j_i}'s are independent of $X_i(0)$'s.

Recall that

$$\bar{L}(S^*) = h(C_{S^*} | X_{S^*}(0)) - \frac{|S^*|}{2} \log 2\pi e\beta + |S^*| \frac{\alpha}{2}.$$

Note that from Step 2, we have that $\beta = 1$ and $\alpha = -\log(\varepsilon^2 E(C^2) + a^2\eta)$.

5. Next, we compute $h(C_{S^*} | X_{S^*}(0))$ given the assumptions of our formulation. Recall

that we have performed these computations in Example 4.4.2. We obtained the following:

$$\bar{L}(S^*) \geq \log \frac{B\left(\sum_{j\in S^c}\beta_j^2\right)^{\frac{1}{2}}|a|^{m-1}}{\sqrt{2\pi e\eta}} + \frac{|S^*|}{2}\left(\log\frac{\eta}{\varepsilon^2 E(C^2)+a^2\eta}\right),$$

where we have substituted in $\alpha = -\log(\varepsilon^2 E(C^2) + a^2\eta)$.

6. Finally, we make the appropriate choice of our parameters, a and η. Assume, without loss of generality, that

$$\left(\frac{\sum_{j\in S^c}\beta_j^2}{2\pi e}\right)^{\frac{1}{2}} \geq 1,$$

otherwise, we can just scale our choices for a and η. Let $a = \left(\frac{\eta^{\frac{1}{2}}}{B}\right)^{\frac{1}{m-1}}$, then,

$$\bar{L}(S^*) \geq \frac{|S^*|}{2}\left(\log\frac{1}{\frac{\varepsilon^2 E(C^2)}{\eta}+a^2}\right).$$

Next, let $\eta = B$. Then, because $m - 1 < \frac{n}{2}$,

$$a^2 < \left(\frac{1}{B}\right)^{\frac{2}{n}}.$$

Observe that $E(C^2) \leq MB^2$, where M is some integer. So,

$$\frac{\varepsilon^2 E(C^2)}{\eta} + a^2 \leq \varepsilon^2 MB + \left(\frac{1}{B}\right)^{\frac{2}{n}}.$$

Combining with Step 4, we have that

$$T\sum_{i\in(S^*)^c}\sum_{j\in S^*}\mathbf{C}_{ij} \geq \frac{|S^*|}{2}\log\frac{1}{\varepsilon^2 MB + \left(\frac{1}{B}\right)^{\frac{2}{n}}}.$$

Rearranging, we have that

$$T \geq \frac{1}{2}\frac{1}{\frac{\sum_{i\in(S^*)^c}\sum_{j\in S^*}\mathbf{C}_{ij}}{|S^*|}}\log\frac{1}{\varepsilon^2 MB + \left(\frac{1}{B}\right)^{\frac{2}{n}}}.$$

Here, we must have $\varepsilon^2 M \in \left[0, \frac{1}{B}\left(1 - \left(\frac{1}{B}\right)^{\frac{2}{n}}\right)\right)$, in order for the lower bound to be positive.

Finally, because we had chose our S^* such that $\frac{\sum_{i\in(S)^c}\sum_{j\in S}\mathbf{C}_{ij}}{|S|}$ is minimized, we have that

$$\phi(G) = \frac{\sum_{i\in(S^*)^c}\sum_{j\in S^*}\mathbf{C}_{ij}}{|S^*|}.$$

□

Remark We show in the next section that out lower bound is tight in its reciprocal dependence on the conductance term. So, for fixed n, we have a scaling law that is tight in the case of severe communication constraints, such as very small channel capacities due to low transmission power.

In the case of increasing number of nodes, however, B must increase exponentially with n for our lower bound to remain valid. The requirement is a by-product of using a formulation based on random variables together with Information Theoretic variables. This requirement ensures that as n increases, our bound properly captures the number of bits that are transferred.

When we consider sums of independent identically distributed random variables, Central Limit Theorem type arguments imply that as the number of the random variables increases, there is some randomness lost, because we know that the distribution of the sum must converge to the Normal distribution. However, in a setting where the initial conditions are fixed values, as in the case of the algorithm we describe below, the addition of a node clearly will not reduce the information that needs to be communicated in the network. To counterbalance the probabilistic effects, we need to have B increase as the number of nodes increases.

Next, we complete the proof of Theorem 5.1.1 by proving the statements of Step 1 and Step 2.

Proof of Step 1. We show that for small enough δ, $\mathbf{P}\left(|X_i(T) - C| \leq \varepsilon C \middle| A\right) \geq 1 - \delta$ implies $E(X_i(T) - C)^2 \leq \varepsilon^2 E(C^2)$.

First, observe that,

$$\mathbf{P}\left(|X_i(T) - C| \geq \varepsilon C \middle| A\right) \leq \delta,$$

is equivalent to

$$\mathbf{P}\left((X_i(T) - C)^2 \geq \varepsilon^2 C^2 \middle| A\right) \leq \delta,$$

Next, when we condition on A, C is a fixed number. So, we have we have that

$$\begin{aligned} E\left((X_i(T) - C)^2 \middle| A\right) &= \int_0^\infty \mathbf{P}\left((X_i(T) - C)^2 \geq x \middle| A\right) dx \\ &= \int_0^{\varepsilon^2 C^2} \mathbf{P}\left((X_i(T) - C)^2 \geq x \middle| A\right) + \int_{\varepsilon^2 C^2}^\infty \mathbf{P}\left((X_i(T) - C)^2 \geq x \middle| A\right) dx \\ &\leq \varepsilon^2 C^2 + \delta\kappa, \end{aligned}$$

where the last inequality follows

- for the first term, because $\mathbf{P}\left((X_i(T) - C)^2 \geq x \middle| A\right) \leq 1$, and,
- for the second term, because $\mathbf{P}\left((X_i(T) - C)^2 \geq x \middle| A\right) \leq \delta$ for all $x \in [\varepsilon^2 C^2, \infty)$. We have also assumed that for every A, $(X_i(T) - C)^2$ is bounded from above.

Finally, we have that

$$E(X_i(T) - C)^2 = E\left(E\left((X_i(T) - C)^2 \middle| A\right)\right),$$

where the outermost expectation is with respect to the joint distribution of the initial conditions. Choosing δ to be very small, we have the desired result. □

Proof of Step 2. We show that if $E(X_i(T) - C)^2 \leq \varepsilon^2 E(C^2)$, then $E(X_i(T) - C_i)^2 \leq \varepsilon^2 E(C^2) + a^2\eta$, where $C_i = C + a\epsilon_{j_i}$, and ϵ_{j_i} has mean zero and variance η and is independent of all the $X_i(0)$'s.

$$\begin{aligned} E(X_i(T) - C_i)^2 &= E(X_i(T) - C - a\epsilon_{j_i})^2 \\ &= E(X_i(T) - C)^2 + E(a\epsilon_{j_i})^2 - 2E(X_i(T) - C)(a\epsilon_{j_i}) \\ &\overset{(a)}{=} E(X_i(T) - C)^2 + E(a\epsilon_{j_i})^2 + 2E(X_i(T) - C)E(a\epsilon_{j_i}) \\ &\overset{(b)}{=} E(X_i(T) - C)^2 + E(a\epsilon_{j_i})^2, \end{aligned}$$

where,

(a) follows because $X_i(T)$ is the estimate produced by the algorithm, and depends on the initial conditions, $X_i(0)$'s, while ϵ_{j_i} is independent of $X_i(0)$'s, and,

(b) follows because ϵ_{j_i} has mean zero.

□

5.4 A Tight Upper Bound: An Algorithm

Next, we describe the algorithm that achieves the lower bound. That is, we exhibit the reciprocal dependence of the algorithm's computation time on the conductance of the graph. Because the function that is to be computed, the sum, is relatively simple, and the algorithm requires little computation overhead, the limitations that arise are due primarily to the communication constraints. In fact, the dependence on the algorithm's run-time on conductance arises due to the fact that the algorithm uses an information spreading algorithm as a subroutine. Information spreading depends reciprocally on conductance: the more severe the connectivity constraints, the smaller the conductance and the longer it takes for information to spread in the network.

We describe in detail the problem formulation in the next section. The algorithm that we describe is based on an algorithm by Mosk-Aoyama and Shah [24]. In section 5.4.2 we discuss this algorithm and its applicability to our formulation. In section 5.4.3 we summarize our main results. In section 5.4.4, we describe the contributions of [24] in the design of an algorithm for distributed computation of a separable function, in a network of nodes using repeated communication of real-valued messages. In section 5.4.5, we describe the algorithm when the communicated messages are quantized, and analyze how the performance of the algorithm changes relative to the performance of the unquantized algorithm of [24].

5.4.1 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 $x_i(0) \in \mathbb{R}$. We let the vector x represent all the initial values in the network, $x = (x_1(0) \ \dots \ x_n(0))$. The goal of the nodes is to each acquire an estimate of a given function,f, of all the initial values. In this section, the function f is separable, defined as follows. Here, 2^V denotes the power set of V.

Definition 5.4.1. $f : \mathbb{R}^n \times 2^V \to \mathbb{R}$ is separable if there exist functions $f_1, \dots, f_n$ such that for all $S \subseteq V$,

$$f(x, S) = \sum_{i \in S} f_i(x_i(0)).$$

Furthermore, we assume $f \in \mathcal{F}$ where $\mathcal{F}$ is the class of all separable functions with $f_i(x_i(0)) \geq 1$ for all $x_i(0) \in \mathbb{R}$ and $i = 1, \dots, n$.

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, $T_{\mathcal{C}}^{\text{cmp}}(\varepsilon, \delta)$. It is the time until the estimates at all nodes are within a factor of $1 \pm \varepsilon$ of $f(x, V)$, with probability larger than $1 - \delta$. The definition follows, where $\hat{y}_i(k)$ denotes the estimate of $f(x, V)$ at node i at time k.

Definition 5.4.2. For $\varepsilon > 0$ and $\delta \in (0, 1)$, the (ε, δ)-computing time of an algorithm, $\mathcal{C}$, denoted as $T_{\mathcal{C}}^{\text{cmp}}(\varepsilon, \delta)$ is defined as

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

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 $S_i(k)$ is the set of nodes that have node i's message at time k, and V is the set of nodes, the definition of $T_{\mathcal{D}}^{\text{spr}}(\delta)$ is the following.

Definition 5.4.3. For a given $\delta \in (0, 1)$, the δ-information-spreading time, of the algorithm $\mathcal{D}$, $T_{\mathcal{D}}^{\text{spr}}(\delta)$, is

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

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

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

5.4.2 Background

The algorithm that we describe is based on an algorithm by Mosk-Aoyama and Shah [24]. In that formulation, each node has a fixed real-valued initial condition, that is bounded away from zero. Nodes compute a separable function [1] of the initial values in the network. The algorithm guarantees that with some specified probability, all nodes have an estimate of the function value within a desired ε-interval of accuracy around the true value. In [24], 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 real-valued message to node j with probability p_{ij} and with probability p_{ii} sends its message to no other nodes; if $(i, j) \notin E$, $p_{ij} = 0$.

The algorithm of [24] 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 [24], however, is that it requires nodes to exchange real numbers. As such, the algorithm is not practically implementable. Below, we quantize this algorithm, so that instead of sending real-valued messages, nodes communicate bits. Now, node i can send to j one bit with probability p_{ij}, and no bits otherwise. This is equivalent to node i communicating to j via an erasure channel with capacity $\mathbf{C}_{ij} = p_{ij}$, so, $\Phi(G) = \Phi(P)$. [2] We will show that the effect of communicating bits instead of real-valued messages is to slow down the original algorithm by $\log n$; however, the dependence of computation time on conductance is unchanged.

Another difference between our formulation and the one in [24], is that we assume that the initial conditions lie in a bounded interval, $[1, B]$, whereas in [24] there is no upper bound. We need this assumption to show that our algorithm will also guarantee that with some specified probability, all nodes have an estimate of the function value within a desired ε-interval of accuracy around the true value. However, due to communicating a finite number of bits, ε cannot be arbitrarily close to zero.

Finally, we recall that in deriving the lower bound of the previous section, we had assumed a joint probability distribution on the initial conditions. However, we will describe the algorithm for fixed initial-values at the nodes. If the initial conditions were in fact distributed according to some joint probability density function, the algorithm that we describe below can be used for any realization of the initial values to guarantee, with the desired probability, the ε-accuracy criterion. So, the algorithm satisfies the "if" condition in the statement of Theorem 5.1.1.

As such, the computation time of the algorithm we describe below must scale reciprocally with conductance. We provide an upper bound on the run-time and show that, indeed, it does scale inversely with conductance. Thus, the contribution of this work includes the non-trivial quantized implementation of the algorithm of [24] and its analysis. As a consequence,

[1] A linear function of the initial conditions is a separable function.

[2] For the purpose of analysis, it is more convenient to consider erasure channels for which $\log M$ bits are sent noiselessly with probability p_{ij}, and there is an erasure otherwise. This avoids the need to account for receivers keeping track of individual bits that make up each message. Furthermore, the reciprocal dependence in our bounds between the computation time and conductance will be unchanged.

we obtain the fastest, in terms of dependence on network topology, quantized distributed algorithm for separable function computation.

5.4.3 Main Result

The main result of this section is stated in the following theorem.

Theorem 5.4.5. *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 $\varepsilon \in (\gamma f(x,V), \gamma f(x,V) + \frac{1}{2})$ and $\delta \in (0,1)$,*

$$T^{cmp}_{\mathcal{AP}^{\mathcal{Q}}}(\varepsilon,\delta) = O\left(\varepsilon^{-2}(1+\log\delta^{-1})\frac{(\log n + \log\delta^{-1})\log n}{\Phi(P)}\right). \tag{5.2}$$

For example, the bound implied by the above theorem when $\delta = \frac{1}{n^2}$ is

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

Recall that by the Information Theoretic lower bound derived in this chapter, we have that the computation time is lower bounded as

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

where B is a constant such that for all i, $f_i(x_i) \leq B$.

Because the computation time and graph conductance are reciprocally related in both this lower bound and the upper bound in (5.2), we conclude that our results are tight in capturing the scaling of the computation time with respect to the graph conductance. So, our algorithm is optimal in its dependence on the network topology.

5.4.4 Unquantized Function Computation

In [24], 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 assumes that the nodes exchange real-valued messages whenever a communication takes place. The 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 Algorithm

The following property of exponential random variables plays a central role in the design of this algorithm. Let $W^1, \ldots, W^n$ be independent exponentially distributed random variables, where W^i has mean $1/\theta_i$. Then, the minimum, $W^* = \min_{i=1,\ldots,n} W^i$, will also be exponentially distributed, and its mean is $1/\sum_{i=1}^n \theta_i$.

Suppose that node i has an initial value θ_i. Each node needs to compute $\sum_{i=1}^n \theta_i$. Node i generates an exponential distribution with mean $1/\theta_i$. It then draws a sample, $W^i = w^i$, from that distribution. All nodes do this. They exchange their samples so that each node knows every sample. Then, each node may compute the minimum of the samples, $w^* = \min_{i=1,\dots,n} w^i$. w^* is a realization of W^*, which is exponentially distributed, with mean $1/\sum_{i=1}^n \theta_i$.

For the algorithm proposed in [24], the nodes perform the above procedure on r samples from each node rather than one. That is, node i draws independently r samples from its exponential distribution, $W_1^i, \dots, W_r^i$. The nodes exchange information using the information spreading algorithm described below. Ultimately, each node acquires $W_1^*, \dots, W_r^*$, where W_l^* is the sample-wise minimum, $W_l^* = \min_{i=1,\dots,n} W_l^i$. Then, for its estimate of $\sum_{i=1}^n \theta_i$, each of the nodes computes

$$\frac{r}{\sum_{l=1}^r W_l^*}.$$

Recall that as r increases, $\frac{1}{r}\sum_{l=1}^r W_l^*$ approaches the mean of W_l^*, namely $1/\sum_{i=1}^n \theta_i$. It is shown that, for large enough r, the nodes' estimates of $\sum_{i=1}^n \theta_i$ will satisfy the desired accuracy criterion with the desired probability.

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 $W^i = (W_1^i, \dots, W_r^i)$ and needs to obtain $\bar{W} = (\bar{W}_1, \dots, \bar{W}_r)$, where $\bar{W}_l = \min_{i=1,\dots,n} W_l^i$. To compute $\bar{W}$, each node maintains an r-dimensional vector, $\hat{w}^i = (\hat{w}_1^i, \dots, \hat{w}_r^i)$, which is initially $\hat{w}^i(0) = W^i$, and evolves such that $\hat{w}^i(k)$ contains node $i's$ estimate of $\bar{W}$ at time k. Node i communicates this vector to its neighbors; and when it receives a message from a neighbor j at time k containing $\hat{w}^j(k^-)$, node i will update its vector by setting $\hat{w}_l^i(k^+) = \min(\hat{w}_l^i(k^-), \hat{w}_l^j(k^-))$, for $l = 1, \dots, r$.

As argued in [24], 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, $\hat{w}^i(k) = \bar{W}$ when $k = rT_{\mathcal{D}}^{\mathrm{spr}}(\delta)$, because the nodes propagate in the network an evolving estimate of the minimum, an r-vector, as opposed to the n r-vectors $W^1, \dots, W^n$.

The Performance

The first of the two main theorems of [24] 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 [24].

Theorem 5.4.6. *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 $\varepsilon \in (0,1)$ and $\delta \in (0,1)$,*

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

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 5.4.3. We refer the reader to [24] for further details on the information spreading algorithm, including an analysis of the case of asynchronous communication. The theorem relevant to this chapter follows.

Theorem 5.4.7. *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 $\delta \in (0,1)$,*

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

5.4.5 Quantized Function Computation

The nodes need to each acquire an estimate of $f(x,V) = \sum_{i=1}^{n} f_i(x_i(0))$. For convenience, we denote $f_i(x_i(0))$ 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 B for which: for all i, $\theta_i \in [1, B]$.

Recall that the goal is to design an algorithm such that, for large enough k,

$$\mathbf{P}\left\{\cap_{i=1}^{n}\{|\hat{y}_i^Q(k) - y| \leq \varepsilon y\}\right\} \geq 1 - \delta,$$

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.

Computation of Minima Using Information Spreading

We use the same scheme that was described in 5.4.4 for computation of minima using information spreading. Now, node i quantizes a value $\hat{\hat{w}}_l^i$ that it needs to communicate to its neighbor, j, where node i maps the value $\hat{\hat{w}}_l^i$ 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 $\hat{\hat{w}}_l^j$. 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 $k = rT^{\text{spr}}_{\mathcal{D}}(\delta) \log M$.

Summary of Algorithm & Main Theorem

The proposed algorithm, $\mathcal{A}^Q$ 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 below. 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 $\mathbf{P}\left\{\cap_{i=1}^{n}\{|\hat{y}_i^Q - y| \leq \varepsilon y\}\right\} \geq 1-\delta$ is achieved. Determining how large r and M must be leads to the main theorem of this section.

Theorem 5.4.8. *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 $\varepsilon \in (\gamma f(x,V), \gamma f(x,V) + \frac{1}{2})$ and $\delta \in (0,1)$,*

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

Remark Here, we point out that the condition in the theorem that $\varepsilon \in (y\gamma, y\gamma + 1/2)$ 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. The derivation of the computation time of this modified algorithm will lead to determining the appropriate truncation parameter, m. Next, 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 $\log n$.

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 [24] 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 5.4.8.

Lemma 5.4.9. 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, $\mathcal{A}_{\mathcal{M}}^{Q}$.

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 $A_l^i = \{W_l^i \leq m\}$, $\mathbf{P}(W_l^* \leq w | A_l)$, (see Appendix for proof). 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, see [8], 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. □

Proof of Theorem 5.4.8

Before we proceed with the proof of the Theorem, we describe the quantization scheme. In step 2 of the algorithm $\mathcal{A}^Q$, 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 $1 \pm \varepsilon$ of y. That is,

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

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 $1 \pm \varepsilon$ factor of y. For this, we compare the quantized algorithm, $\mathcal{A}^Q$, with the modified algorithm $\mathcal{A}_{\mathcal{M}}^Q$.

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,

$$\frac{1}{r}\sum_{l=1}^{r}\min_{i=1,\ldots,n}(w_l^i)_Q \in \left[\frac{1}{r}\sum_{l=1}^{r}\min_{i=1,\ldots,n} w_l^i - \gamma, \frac{1}{r}\sum_{l=1}^{r}\min_{i=1,\ldots,n} w_l^i + \gamma\right]. \tag{5.3}$$

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,

$$\mathbf{P}\left(\cup_{i=1}^{n}\{\widehat{\bar{W}}^i \neq \bar{W}\}\right) \leq \frac{\delta}{2}. \tag{5.4}$$

Consider the case where $\{\cap_{i=1}^{n}\{\widehat{\bar{W}}^i = \bar{W}\}\}$, we have from Lemma 5.4.9 that, for any $\mu \in (0, 1/2)$, if $m = \ln n - \ln(1 - e^{-\frac{\mu^2}{6}})$,

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

Combining with (5.3), 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

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

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

$$((1+\varepsilon)^{-1}, (1-\varepsilon)^{-1}) \subset (1-2\varepsilon, 1+2\varepsilon).$$

So,

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

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 (5.4) in the Total Probability Theorem, we have the desired result,

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

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 $T^{\text{spr}}_{\mathcal{D}}(\delta)$ time slots to disseminate through the network, so $(\log M)T^{\text{spr}}_{\mathcal{D}}(\delta)$ time slots are needed until the quantized messages are disseminated and the minima computed. Consequently, the computation time of the quantized algorithm will be $O((\log M)rT^{\text{spr}}_{\mathcal{D}}(\delta/2))$.

But, $M = m/\gamma$, and by design, for a given μ we choose $m = \ln n - \ln(1 - e^{-\frac{\mu^2}{6}})$; 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, $\mu = \varepsilon - y\gamma$; and, $\gamma = \Theta(1/n)$ so, $y\gamma = O(1)$. We therefore have, for $\varepsilon \in (y\gamma, y\gamma + 1/2)$,

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

□

5.5 Summary

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, especially with respect to its dependence on conductance. This result is stated in Theorem 5.4.8.

Combining the result of Theorem 5.4.8 with that of Theorem 5.4.7 yields Theorem 5.4.5. Comparison with a lower bound obtained via Information Theoretic inequalities in Chapter 4 reveals that the reciprocal dependence between computation time and graph conductance in the upper bound of Theorem 5.4.5 matches the lower bound. Hence the upper bound is tight in capturing the effect of the graph conductance $\Phi(P)$.

Chapter 6

Bounds Capturing Computational Limitations

In our derivation of lower bounds in previous chapters, we assumed no constraints on nodes' computational abilities. They had no limited memory or power. In fact, we assumed that they could run whatever algorithm necessary to make their computations, and they could employ whatever communication algorithm (encoding and decoding) needed. We assumed no constraints on the evolution of the nodes' estimates, $X_i(k)$, apart from the desired mean square error criterion. Because of this, we had to loosen our lower bounds to the point of considering the mutual information between $X_i(T)$ and the initial conditions, rather than the mutual information between the sequence $(X_i(1), \ldots, X_i(T))$ and the initial conditions. As a result, we obtained lower bounds that must hold regardless of the computation or communication algorithm that is used.

In this chapter, we explore, via two examples, lower bounds that capture, in addition to communication constraints, the effect of limited computational resources. Such scenarios arise, for example, in belief propagation algorithms or multi-agent problems when there are bit constraints. In these scenarios, nodes make specific, usually simple, computations.

In section 6.1 we consider limitations that arise due to the computational architecture. Specifically, we consider algorithms that have a tree-based architecture. The computation at any node can depend only on that node's initial condition and messages received from child nodes. We apply techniques of Chapter 4 to a simple 3-node tree where messages flow via channels from the leaf nodes to the root node. The root node is to compute a function of its own initial condition and the initial conditions of the leaf nodes. We obtain lower bounds on channel capacities if at time T the mean square error in the root node's estimate is within a known interval.

In section 6.2 we consider two nodes that can communicate via quantized messages with the goal of computing the average of their initial values. We fix the computation scheme at each of the nodes: the updated estimate of the average at a node is a convex combination of its own data and the data it receives from the other node. We obtain a lower bound on the number of bits that must be communicated between the nodes in order for the norm of the error in the nodes' estimates to converge to zero at a desired rate. The goal is to capture the trade-off between accuracy of the estimates and the communicated bit requirements, and hence resources dedicated for communication. Because we fix the computation scheme, we are able to explicitly incorporate the effects of the computation scheme into our analysis.

6.1 Lower Bounds for Tree-Based Algorithms

In this section, we consider computational limitations that arise for algorithms that run over directed trees. In particular, any computation at a node can depend only on that node's initial value and messages received from that node's children. Messages are received via noisy channels.

Consider the directed tree shown in Figure 6-1. Each node has an initial condition, $X_i(0)$. Node 1, the root, receives messages from its children, nodes 2 and 3. Node 2 sends its messages through a channel with capacity $\mathbf{C}_{21}$; similarly, node 3 sends its messages through a channel with capacity $\mathbf{C}_{31}$. The root needs to acquire an estimate of a given function of the initial conditions in the network, $C = f(X_1(0), X_2(0), X_3(0))$. We denote its estimate of C at time T by $X_1(T)$. Suppose that at time T, the mean square error in the estimate is in an α-interval, $E(X_1(T) - C)^2 \leq 2^{-\alpha}$. Then, the channel capacities are bounded from below, as a function of α, T, and the differential entropy of the desired function.

To derive the lower bounds, we use the techniques developed in Chapter 4. First, we derive a lower bound on $T\mathbf{C}_{21}$. Consider the cut that divides the network to a set containing nodes 1 and 3, $S = \{1, 3\}$, and a set containing node 2, $S^c = \{2\}$. Information traverses this cut via channel $\mathbf{C}_{21}$. In Chapter 4, by using Network Information Theory arguments, we saw that

$$T \sum_{i \in S^c} \sum_{j \in S} \mathbf{C}_{ij} \geq I(X_S(T); X_{S^c}(0) | X_S(0)),$$

which, when substituting for $S = \{1, 3\}$ and $S^c = \{2\}$, becomes

$$T\mathbf{C}_{21} \geq I(X_1(T), X_3(T); X_2(0) | X_1(0), X_3(0)).$$

However, note that in the case of the directed tree, any estimate at node 3, $X_3(T)$, can only depend on node 3's own initial condition and messages received from its children, and hence cannot depend on the initial conditions of node 2, $X_2(0)$. So,

$$I(X_1(T), X_3(T); X_2(0) | X_1(0), X_3(0)) = I(X_1(T); X_2(0) | X_1(0), X_3(0)).$$

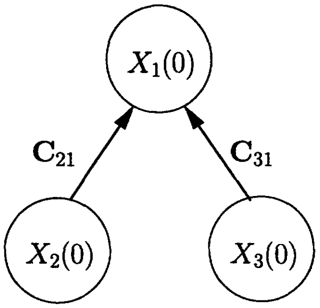

Figure 6-1: The root node receives messages through channels from its children. It computes a function of the initial conditions in the network.

Now, we have that

$$\begin{aligned}
T\mathbf{C}_{21} &\geq I(X_1(T); X_2(0)|X_1(0), X_3(0)) \\
&\geq I(X_1(T); C|X_1(0), X_3(0)) \\
&\geq h(C|X_1(0), X_3(0)) + \frac{\alpha}{2} - \frac{1}{2}\log 2\pi e,
\end{aligned}$$

where the second inequality follows by the Data Processing Inequality and the last inequality follows by the same arguments used in Chapter 4.

Similarly, for $S = \{1, 2\}$ and $S^c = \{3\}$,

$$T\mathbf{C}_{31} \geq h(C|X_1(0), X_2(0)) + \frac{\alpha}{2} - \frac{1}{2}\log 2\pi e.$$

Finally, for $S = \{1\}$ and $S^c = \{2, 3\}$,

$$T(\mathbf{C}_{21} + \mathbf{C}_{31}) \geq h(C|X_1(0)) + \frac{\alpha}{2} - \frac{1}{2}\log 2\pi e.$$

Putting these inequalities together, we have the following theorem.

Theorem 6.1.1. *Consider the network of 3 nodes arranged in a directed tree, in Figure 6-1. The root receives messages from its children via noisy channels and estimates a function of the initial conditions in the network,* $C = f(X_1(0), X_2(0), X_3(0))$.

If at time T *the mean square error in the root's estimate of* C *is within an* α*-interval,* $E(X_1(T) - C)^2 \leq 2^{-\alpha}$*, then, the channel capacities must satisfy*

$$\begin{aligned}
\mathbf{C}_{21} &\geq \frac{1}{T}\left(h(C|X_1(0), X_3(0)) + \frac{\alpha}{2} - \frac{1}{2}\log 2\pi e\right) \\
\mathbf{C}_{31} &\geq \frac{1}{T}\left(h(C|X_1(0), X_2(0)) + \frac{\alpha}{2} - \frac{1}{2}\log 2\pi e\right) \\
\mathbf{C}_{21} + \mathbf{C}_{31} &\geq \frac{1}{T}\left(h(C|X_1(0)) + \frac{\alpha}{2} - \frac{1}{2}\log 2\pi e\right). \qquad (6.1)
\end{aligned}$$

Remark Suppose we define a "capacity region," the region of capacity vectors, $(\mathbf{C}_{21}, \mathbf{C}_{31})$, that are necessary for $E(X_1(T) - C)^2 \leq 2^{-\alpha}$ to hold. This is somewhat analogous to the rate regions we defined in Chapter 2. Recall that, there, the rate region was the set of code rates that are both necessary and sufficient for the probability of decoding error to approach zero asymptotically. Here, in the case of a "capacity region" it only makes sense to consider necessity. If we wanted to consider "tightness," we need to look at lower bounds on T, in terms of the capacities, and produce an algorithm with computation time achieving the lower bound, as we have done in Chapter 5.

The utility of a capacity region is that it provides us with a "negative result." For a given T and α, if a vector, $(\mathbf{C}_{21}, \mathbf{C}_{31})$, is not in the capacity region, then no algorithm will satisfy $E(X_1(T) - C)^2 \leq 2^{-\alpha}$. The set of inequalities (6.1) define the capacity region. Its shape will, in general, depend on the differential entropy terms and the value of α. One possible shape is shown in Figure 6-2.

Example 6.1.2. Consider the 3-node directed tree of Figure 6-1. Suppose the initial conditions were independent and identically distributed, each Normal with mean zero and

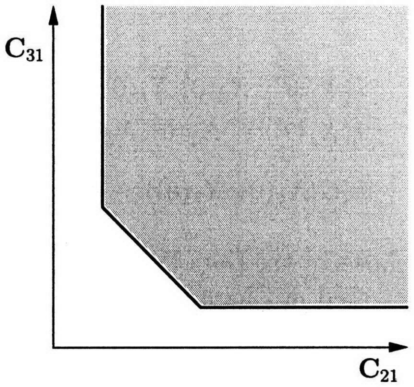

Figure 6-2: The region of capacities that are necessary to ensure the accuracy level α in the root's computation at time T.

variance ν. Let $C = \sum_{i=1}^{3} \beta_j X_j(0)$. Then,

$$\begin{aligned}
h(C|X_1(0), X_3(0)) &= \frac{1}{2}\log 2\pi e \beta_2^2 \nu \\
h(C|X_1(0), X_2(0)) &= \frac{1}{2}\log 2\pi e \beta_3^2 \nu \\
h(C|X_1(0)) &= \frac{1}{2}\log 2\pi e (\beta_2^2 + \beta_3^2) \nu
\end{aligned}$$

Substituting in the lower bounds of the set of inequalities (6.1), we have the following capacity region.

$$\begin{aligned}
C_{21} &\geq \frac{1}{T}\left(\frac{1}{2}(\alpha + \log \nu) + \frac{1}{2}\log {\beta_2}^2\right) \\
C_{31} &\geq \frac{1}{T}\left(\frac{1}{2}(\alpha + \log \nu) + \frac{1}{2}\log {\beta_3}^2\right) \\
C_{21} + C_{31} &\geq \frac{1}{T}\left(\frac{1}{2}(\alpha + \log \nu) + \frac{1}{2}\log({\beta_2}^2 + {\beta_3}^2)\right).
\end{aligned}$$

Therefore, the channels must be such that they are able to accommodate more information reaching the root about the random variable with the larger weight in the sum, C.

6.2 Quantized Consensus: A Two-Node Example

Recently, there has been a resurgence of interest in the consensus/flocking problem, some formulations of which are a special case of work done by Tsitsiklis [32] in the 1980's. For example, in the case of the flocking problem, which models the aggregate behavior of schools of fish or flocks of birds, each agent updates its estimate of a certain parameter, like direction of motion, as a function of information it receives from its set of neighbors, which varies across time. In [15] a discrete-time model of flocking (Viscek model), where for a fixed communication topology agent update equations are linear and communication between pairs of agents is bidirectional, is analyzed using tools from matrix theory. Results provide (sufficient) conditions, involving the frequency of communication of the agents, under which the asymptotic convergence of the estimates of all the agents to the same constant value is guaranteed.

Similar results are provided in [23], for a more general model for the agent dynamics. Now, still in discrete-time, agent dynamics are allowed to be nonlinear, provided that the nonlinear function describing each agent's dynamics (update rule) satisfies a strict convexity assumption. Set Lyapunov theory and graph theory are used to derive necessary and/or sufficient conditions for asymptotic agreement under both unidirectional and bidirectional communication cases.

In the work done by Tsitsiklis [32], also in discrete-time, agents, or "processors," exchange some or all of their estimates, or values of their state variables, with other processors. Each agent updates its estimate by taking a linear convex combination of its estimate and estimates of the same variable that it received from other processors, possibly with delay. However, in addition to using information that is received from other processors to update its estimate, each agent may (but does not necessarily have to) add a quantity that it computes, possibly corrupted by noise. For example, this quantity could be that which causes the agent's state to move such that a cost function of the states is minimized.

Under certain assumptions on the agent update model, the communication topology and frequency, and the cost function, it is shown that the estimates of all the agents asymptotically converge to the same value. Further assumptions are needed to show that the value to which the states converge is a stationary point of the cost function or the global optimum. Key in the method of proof is the ability, due to linearity of the agent update model, to come up with a "global" state that describes the entire system.

The results of the original work [32, 33] are restated for the consensus problem in [2]. Convexity of the update equations, strong connectivity of the nodes, and bounded delays are among the conditions that are needed to guarantee asymptotic consensus. A new result presented is that the bounded intercommunication intervals requirement can be relaxed if the communication between the nodes is bidirectional (symmetry). Further, it is shown that the bidirectional communication need not occur simultaneously; it suffices that the nodes exchange and update their values within a bounded interval of time.

In a special case of the consensus problem, the "average consensus" problem, the nodes each have an initial state and the goal is for each of the nodes to acquire an estimate of an average of all the initial states in the network by receiving information from its neighbors. This problem is the focus of the work by [26]. Here, agent dynamics are linear continuous-time and communication delays are allowed. The conditions on communication topologies that ensure average consensus are presented; it is found that if the graphs formed by the communicating nodes are always strongly connected and "balanced", average consensus will be achieved, and the rate of convergence is related to a property of the graphs. Analysis makes use of Lyapunov equation, matrix theory, and graph theoretic tools. Olfati-Saber goes on to make use of the idea of achieving average consensus by suggesting a scheme for distributed Kalman filtering using an algorithm that depends on average consensus reached on quantities measured and computed by individual sensors [25].

The work in the literature explores the effect on achieving "consensus" of communication topology, that could vary across time, or communication of varying strength, as modeled by weights on the communication links. Some of the work even allows delays. However, when a communication link is present, no further communication constraints are assumed. In this chapter, we study the effect on achieving "average consensus", or computation of the average of the data in the network, that communication constraints impose. Specifically, we start with the simple case in which the nodes are fully interconnected, but communicate over a noiseless finite rate channel, that is, via quantized messages. If there are no communication constraints, each node should have acquired the average value after the first communication occurs. If communication occurs via quantized messages, however, this is no longer true. We seek necessary conditions on the rate of communication of the nodes for the error in each node's estimate of the average to be small beyond some prescribed time.

This problem does not fit into any of the frameworks mentioned above, and those of which we know. Even the most general of the frameworks, [23] which allows nonlinear updates, does not apply in our case because the strict convexity assumption is violated. Further, even if there was a way to fix this, the results presented there still need to be somehow related to the communication constraints imposed by the channels between the nodes.

We present a formulation for 2 interconnected nodes below. The 2-node case is an illustrative exercise. First, even for this simple case, various issues concerning the interplay between computation and communication arise. Second, a correct formulation and results derived for the simple 2-node case can be extended to any n-node fully interconnected network.

The more interesting results possibly lie in further extensions. The first direction is to consider communication topologies that are not fully interconnected; equivalently, we allow the rates of the channels connecting some of the nodes to be zero. The second direction is to consider more general channels between the nodes, and use the techniques of Chapter 4. In this case, however, we make explicit assumptions on the evolution of the node estimates, and use these in our inequalities.

6.2.1 Problem Statement

We consider two nodes, 1 and 2, that communicate via quantized messages. Node 1 sends information to 2 through a channel with rate $R_1 > 0$ and node 2 sends information to 1 through a channel with rate $R_2 > 0$ (R_1 and R_2 are the information capacities of the noiseless channels). There are two components to this system: computation and communication. The computation part refers to the updating that each of the nodes performs on its state, which is an estimate of the average of the initial conditions of the two nodes. The communication component refers to a node's sending of information through an encoder, noiseless channel, and decoder to the other node. Specifically we have,

1 Computation. Assume that node 1, (node 2), performs its updates at times k_i, (l_i), $\in \mathbb{Z}^+$, $i = 1, 2, 3, \ldots$. Then,

$$\begin{aligned} x_1(k_i+1) &= \alpha_1 x_1(k_i) + \alpha_2 \hat{x}_2(k_i) \\ x_2(l_i+1) &= \alpha_1 \hat{x}_1(l_i) + \alpha_2 x_2(l_i), \end{aligned} \tag{6.2}$$

and

$$\begin{aligned} x_1(k_i+m) &= x_1(k_i+1) \quad m = 2, \ldots, k_{i+1} - k_i \\ x_2(l_i+n) &= x_2(l_i+1) \quad n = 2, \ldots, l_{i+1} - l_i, \end{aligned} \tag{6.3}$$

where x_1, $x_2 \in \mathbb{R}^n$ are the states of the nodes which represent their estimate of the network average, $c = \alpha_1 x_1(0) + \alpha_2 x_2(0)$. $\hat{x}_1$, $\hat{x}_2 \in \mathbb{R}^n$ are the inputs to nodes 2 and 1 respectively, that are received through the noiseless channels, at the output of the decoders. For simplicity, we assume $n = 1$ through out, though it is possible to extend arguments to any $n \in \mathbb{Z}^+$. Finally, α_1 and $\alpha_2 \in \mathbb{R}^+ \backslash \{0\}$, and $\alpha_1 + \alpha_2 = 1$.

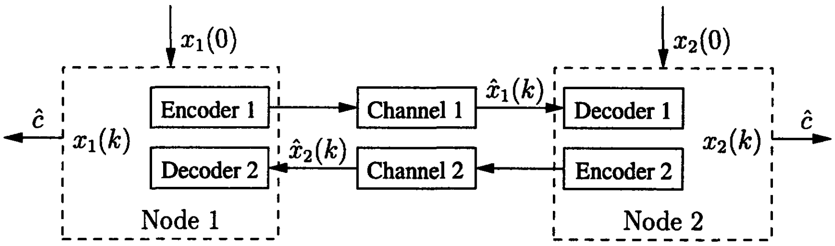

Figure 6-3: Two nodes communicate a finite number of bits for computing the network average.

2 Communication. Suppose that the state at node 1 (node 2) is available at its encoder, E_1 (E_2) which interfaces the node with the communication channel, C_1 (C_2). Then, the outputs of each of the decoders, D_1 and D_2, which interface the channels C_1 and C_2 to nodes 2 and 1, respectively are

$$\begin{aligned}\hat{x}_1(k) &= \phi_1(x_1(k_1), x_1(k_2), \ldots, x_1(k_n), k) \\ \hat{x}_2(k) &= \phi_2(x_2(l_1), x_2(l_2), \ldots, x_2(l_n), k),\end{aligned} \tag{6.4}$$

where k_n and l_n are the latest times at which communication occurs before k, $k_{n+1} > k \geq k_n$ and $l_{n+1} > k \geq l_n$. ϕ_i, $i = 1, 2$, represents the operation of the encoder, channel, and decoder between the nodes.

6.2.2 Convergence to the Network Average

Our goal is to derive a necessary condition on the rate of communication of the nodes for the convergence, with rate r, of each node's state, the estimate of c, to the actual value c. However, to begin, we show that there exists a certain scheme, under which for any $\epsilon > 0$ there exists a time, T, such that $\|x_i(k) - c\| \leq \epsilon$, $\forall\, k \geq T$.

An Illustrative Scheme

Consider the following scheme, in which we assume that the encoders and decoders, (E_i, D_i), $i = 1, 2$, have upper bounds, L_i, of the magnitudes of the initial conditions, $x_1(0)$ and $x_2(0)$. The computation update occurs at both nodes simultaneously once only at time T. At every time $k < T$, node 1 (node 2) sends R_1 (R_2) bits noiselessly to node 2 (node 1), so we have $\hat{x}_1(k) = \phi_1(x_1(0), k)$, ($\hat{x}_2(k) = \phi_2(x_2(0), k)$). The map ϕ_i has the following operation. E_i is composed of a bijection, f_i, of the interval $[-L_i, L_i]$ onto $[0, 1]$, which maps $x_i(0)$ to a number in the unit interval. The binary representation of this is found, and R_i bits are sent over the channel C_i at every time step. These bits are received at the decoder, D_i, which by time k has the first kR_i bits in the binary representation of $f_i(x_i(0))$. This is converted to decimal to obtain a number in the unit interval. This is then mapped back onto $[-L_i, L_i]$, using f_i^{-1}. The output of D_i at time k, $\hat{x}_i(k)$, will be an approximation of $x_i(0)$ with magnitude of error $|e_i(k)| = |\hat{x}_i(k) - x_i(0)| \leq \beta_i 2^{-kR_i}$.

Now, when the update occurs at time T, we have for node 1,

$$\begin{aligned}x_1(T) &= \alpha_1 x_1(T-1) + \alpha_2 \hat{x}_2(T-1) \\ &= \alpha_1 x_1(0) + \alpha_2 x_2(0) + \alpha_2 \hat{x}_2(T-1) - \alpha_2 x_2(0) \\ &= c + \alpha_2(\hat{x}_2(T-1) - x_2(0)).\end{aligned} \tag{6.5}$$

So, at time T, $|x_1(T) - c| \leq \alpha_2\beta_2 2^{-(T-1)R_2}$ and $|x_2(T) - c| \leq \alpha_1\beta_1 2^{-(T-1)R_1}$. So, to ensure that $|x_i(k) - c| \leq \epsilon$, $\forall\, k \geq T$ we must have that $T \geq \max\{1 - \frac{1}{R_1} \log \frac{\epsilon}{\alpha_1\beta_1}, 1 - \frac{1}{R_2} \log \frac{\epsilon}{\alpha_2\beta_2}\}$.

Here, we note that our communication scheme presumes that the decoders have infinite memory to store the received bits. Encoders and decoders are envisioned to be located at the nodes; for example, E_1 and D_2 are located at node 1. So, because we have already assumed infinite memory in the decoders, hence the nodes, we may assume infinite memory at the encoders. In practice the memory of the nodes not only must be finite, but may be required to be small to satisfy design constraints, such as physical size and energy consumption. The more interesting results probably arise in the case that the nodes have finite memory and must therefore encode and decode in *real time*.

Second, we note that the update equations 6.2 add constraints to the problem by explicitly requiring that the computation at node i makes use only of the most current value of x_i. Clearly, if the nodes were assumed to have infinite memory, then this is a rather artificial requirement; it seems inefficient not to allow the nodes to use all information that they have access to, in particular, all past values of their own estimates of c. We have adopted the structure of equations 6.2 in keeping with the formulations the literature, and as a starting point, so that comparison can be made.

Indeed, if the requirement on the structure of the update equations was relaxed, then the optimal strategy when both nodes have infinite memory is trivial. Node i sends its initial value, $x_i(0)$, to node j according to the encoding/decoding strategy described above. As time passes, node j will have a better estimate of $x_i(0)$, $\hat{x}_i(k) = \phi_i(x_i(0), k)$, which it may combine to its own initial value: $x_j(k+1) = \alpha_j x_j(0) + \alpha_i \hat{x}_i(k)$. Thus, the nodes may continuously update their estimates as they receive information. The longer the nodes communicate, the better their estimates of c become.

But this is not true under the update equations 6.2. For example, suppose that the input to E_i by node i is its current state (estimate of c), which it must then send according the scheme described above. Then, once the first update occurs, no further communication between the nodes can improve the nodes' estimates of c. Under the above encoding/decoding scheme, $\hat{x}_i(T-1) - x_i(0) \leq 0$, so, $x_i(T) \leq c$. Now, re-using the above scheme, starting node i at $x_i(T)$ and updating at $T' > T$ will result in $|x_i(T') - c| \geq |x_i(T) - c|$ because (i) the communication scheme consistently underestimates the value sent over the channel, and (ii) the updates are convex combinations. So $\min(x_1(T), x_2(T)) - \delta \leq x_i(T') \leq \max(x_1(T), x_2(T)) \leq c$ [1].

Furthermore, we note that under the naive scheme described above, which does not use the infinite memory to remember past estimates, once the update occurs at time T, no further updates by the nodes can improve the nodes' estimates of c. In particular, even if node 1 had access to $x_2(T)$, it does not know whether $x_2(T)$ is a better or worse estimate of c than $x_1(T)$, because it does not know c. Finally, using similar arguments, one can see that there is no advantage in the first updates of x_1 and x_2 occurring asynchronously.

Necessary Conditions

We wish to show that if there is a computation/communication scheme for which there exists a T such that $|x_i(k) - c| \leq \beta 2^{-Tr}$ for all $k \geq T$, then $h(R_1, R_2) \geq r$ for some h. Note that if the conclusion holds for a scheme that satisfies the stated condition then it must hold for any scheme that satisfies the stronger requirement that the rate of convergence of

[1] The δ arises due to the particular encoding/decoding scheme and the finite rate communication. Recall that in this case, we will have $x_1(T') = \alpha_1 x_1(T) + \alpha_2 \hat{x}_2(T)$ and $x_2(T') = \alpha_2 x_2(T) + \alpha_1 \hat{x}_1(T)$.

the estimate to c is r: $|x_i(k) - c| \leq \beta 2^{-kr}\ \forall k$. Clearly, $|x_i(k) - c| \leq \beta 2^{-kr}\ \forall k$ implies that there exists a T such that $|x_i(k) - c| \leq \beta 2^{-Tr}$ for all $k \geq T$.

We assume that the upper bounds, L_i, of the magnitudes of the initial conditions, $x_1(0)$ and $x_2(0)$ are known, which enables us to make simple counting arguments. We consider two scenarios. The first is where there is a single update at each node, or $|K| = |L| = 1$, where $K = \{k_1, k_2, \dots\}$ and $L = \{l_1, l_2, \dots\}$ are the sets containing the update times of the nodes. For this scenario, because the "computation" part of the scheme is trivial, we can show that if there is a scheme for which there exists a T such that $|x_i(k) - c| \leq \beta 2^{-Tr}$ for all $k \geq T$, then $h(R_1, R_2) \geq r$. Further, we expect that the bound we derive (the function h) is tight.

The second scenario is that for which one or both of $|K|$ and $|L|$ are greater than 1, that is, there can be several updates. Here, without considering the structure of the updates at all, one can show that the weaker statement holds, namely, if there exists a scheme such that $|x_i(k) - c| \leq \beta 2^{-kr}\ \forall k$ then $g(R_1, R_2) \geq r$ for some g. We do not know how tight the bound that we obtain is. Because we do not take into consideration the constraints imposed by the nature of the computation, we expect that it may not be achievable.

First Scenario ($|K| = |L| = 1$) For the first scenario $|K| = |L| = 1$, consider node 1. It follows, as in equations 6.5, that

$$|x_1(k) - c| = \alpha_2 |\hat{x}_2(k^* - 1) - x_2(0)|$$

for all $k \geq k^*$, where $k^* \in K$. Now, assume that $T \geq k^*$, which must be the case if $x_i(0) \neq c$ for $i = 1, 2$ because $x_i(k) = x_i(0)$ for $k < k^*$. This assumption is not restrictive because the effect of communication that occurs for $k < k^*$ can be manifested in $x_i(k)$ only for $k \geq k^*$. These $k \geq k^*$ are the times for which $|x_i(k) - c| \leq \beta 2^{-Tr}$ can hold as a result of communication, which is the concern here. To illustrate this point, suppose $x_1(0) = c$, which implies that $x_2(0) = c$ because $c = \alpha_1 x_1(0) + \alpha_2 x_2(0)$. Then, $|x_i(k) - c| = 0$ for all $k < k^*$, independently of information exchanged, which allows no conclusion concerning the relationship between r and R_1, R_2.

So, by assumption, it then follows that we must have $|\hat{x}_2(k^* - 1) - x_2(0)| \leq \frac{\beta}{\alpha_2} 2^{-Tr}$. Furthermore, this must hold for any $x_2(0)$. But since the channel is noiseless finite rate, at time $k^* - 1$ there can be at most $2^{R_2(k^*-1)}$ choices for $x_2(0)$. That is, the decoder can choose among at most $2^{R_2(k^*-1)}$ values for $\hat{x}_2(k^* - 1)$ and $x_2(0)$ must be within a radius of $\frac{\beta}{\alpha_2} 2^{-Tr}$. So, if $x_2(0) \in [-L_2, L_2]$, then the decoder must be able to cover $[-L_2, L_2]$ with a maximum of $2^{R_2(k^*-1)}$ "cells" each having a maximum radius of $\frac{\beta}{\alpha_2} 2^{-Tr}$. Thus, we must have that

$$2^{R_2(k^*-1)} \frac{\beta}{\alpha_2} 2^{-Tr} \geq L_2.$$

So, rewriting our condition, we have that $r \leq \frac{k^*-1}{T} R_2 + \frac{1}{T} \log \frac{\beta}{\alpha_2 L_2}$. Similarly, from node 2, we get that $r \leq \frac{l^*-1}{T} R_1 + \frac{1}{T} \log \frac{\beta}{\alpha_1 L_1}$. So, we conclude that $r \leq \min\{\frac{l^*-1}{T} R_1 + \frac{1}{T} \log \frac{\beta}{\alpha_1 L_1}, \frac{k^*-1}{T} R_2 + \frac{1}{T} \log \frac{\beta}{\alpha_2 L_2}\}$.

Second Scenario ($|K|$ or $|L| > 1$) First we note that the reason we need to consider this scenario separately is the fact that the node update equations, 6.2, require that $x_i(k_j)$ be used in the computation of $x_i(k_j + 1)$. Indeed, if we were to use $x_i(0)$ at every update instead of $x_i(k_j)$, then the proof of scenario 1 will go through.

However, rather than derive a particular encoding/decoding scheme and make any explicit assumptions on the computation besides those of the communication/computation formulation, we simply assume that there exists a scheme under our formulation such that $|x_i(k) - c| \leq \beta 2^{-kr}\ \forall k$ [2] and show that $g(R_1, R_2) \geq r$ for some g. Thus the derived bounds may not be tight.

Suppose, for simplicity, that $k \in K, L\ \forall k \in \mathbb{N}$. Consider node 1. We have that

$$x_1(k+1) = \alpha_1 x_1(k) + \alpha_2 \hat{x}_2(k).$$

So,

$$\|x_1(k+1) - c\| = \|\alpha_1(x_1(k) - c) - \alpha_2(c - \hat{x}_2(k))\| \leq \beta 2^{-(k+1)r},$$

and hence

$$|\alpha_1 \|x_1(k) - c\| - \alpha_2 \|\hat{x}_2(k) - c\|| \leq \beta 2^{-kr},$$

which we rewrite as

$$\|\hat{x}_2(k) - c\| \leq \frac{1}{\alpha_2}(\alpha_1 \|x_1(k) - c\| + \beta 2^{-kr}).$$

But, $\|x_1(k) - c\| \leq \beta 2^{-kr}$, so,

$$\|\hat{x}_2(k) - c\| \leq \beta \frac{1 + \alpha_1}{\alpha_2} 2^{-kr}.$$

Now, using this inequality together with the assumption that $\|x_2(k) - c\| \leq \beta 2^{-kr}$, and the triangle inequality we have that

$$\begin{aligned} \|x_2(k) - \hat{x}_2(k)\| &= \|x_2(k) - c - \hat{x}_2(k) + c\| \\ &\leq \|x_2(k) - c\| + \|\hat{x}_2(k) - c\| \\ &\leq \beta(1 + \tfrac{1+\alpha_1}{\alpha_2}) 2^{-kr}. \end{aligned} \tag{6.6}$$

Here we use the same counting argument we have used before. Inequality 6.6 must hold for arbitrary $x_2(k)$. Now, because we have assumed that updates may occur at each time step, $x_2(k+1) = \alpha_2 x_2(k) + \alpha_1 \hat{x}_1(k)$, or, $x_2(k) = \alpha_2^k x_2(0) + \alpha_1 \sum_{i=0}^{k-1} \alpha_2^{k-1-i} \hat{x}_1(i)$, we note that at time k, $x_2(k)$ lies in an interval of length $2\alpha_2^k L_2$. However, because the channel is noiseless finite rate, at time k there can be at most $2^{R_2 k}$ choices for $x_2(k)$. So, in order for inequality 6.6 to be true, we must have that

$$2^{R_2 k} \beta (1 + \frac{1 + \alpha_1}{\alpha_2}) 2^{-kr} \geq \alpha_2^k L_2.$$

Rewriting, and using the fact that $\alpha_1 + \alpha_2 = 1$ we have that $r \leq R_2 - \log \alpha_2 - \frac{1}{k} \log \frac{L_2 \alpha_2}{2\beta}$. Similarly, starting the analysis from node 2 we have that $r \leq R_1 - \log \alpha_1 - \frac{1}{k} \log \frac{L_1 \alpha_1}{2\beta}$. Finally, combining these we have that

$$r \leq \min\{R_1 - \log \alpha_1 - \frac{1}{k} \log \frac{L_1 \alpha_1}{2\beta}, R_2 - \log \alpha_2 - \frac{1}{k} \log \frac{L_2 \alpha_2}{2\beta}\}.$$

[2] We believe that we can easily derive a similar result for $|x_i(k_j) - c| \leq \beta 2^{-k_j r}\ \forall j$, where $k_j \in K$ for $i = 1$ and $k_j \in L$ for $i = 2$. But, this is merely an exercise in book-keeping, which we may come back to as time permits.

Thus, our analysis has led to an upper bound on the rate at which the error in node's estimates can converge to zero, if nodes can only communicate via quantization of given resolution. That the bound is tight is not clear. In order to show this, one would need to produce an algorithm that achieves the error rate under the given communication constraints.

Still, the bound captures the effect of the computation algorithm, that is imposed on the nodes, in a qualitatively intuitive way. Specifically, the "$\log \alpha_i$" term arises due to the computation scheme. The closer the α_i to 1, the more heavily node i's own initial condition is weighted in c. Thus, node i's value needs to be communicated to the other node more accurately so that the other node can compute c with the desired accuracy.

6.3 Summary

In this chapter, we incorporated the effect of computational limitations into our lower bounds. We considered two examples. In the first, the computational limits arose from having an underlying tree architecture. We showed how the approach of Chapter 4 is applicable; we characterized the capacity region. A future direction is to generalize the three node example that we considered here.

In the second example, we considered a popular model for distributed computation, where each node updates its estimate by taking a convex combination of its own data and data received from neighboring nodes. We looked at a case where the communication constraint consisted of the requirement that a finite number of bits is exchanged. In this case, we were able to produce a bound using simple inequalities on norms. A future direction would be to use the techniques of Chapter 4 in the more general setting of having channels between nodes. We would make explicit assumptions on the evolution of the process governing the node estimates, $X_i(k)$. It is likely that this would entail the use of more powerful techniques or machinery applied to the sequence of estimates.

Chapter 7

Future Directions

We have set out to study distributed computation under communication constraints, in a setting where there is a network of nodes, each having a piece of information that may be needed by other nodes for computing a given predetermined function. In particular, we sought to understand the limitations imposed by the communication constraints on the computation performance that may be expected of the nodes of the network.

Two Information Theoretic formulations rendered themselves naturally to model "communication constraints." In the first, the subject of Chapters 2 and 3, the communication constraint was that nodes compressed their sequences of data before broadcasting noiselessly to all other nodes. The performance criterion required from each node was that as the sequence length increased, the probability of error, in recovering the entire sequence of corresponding function values, asymptotically approached zero.

In the second formulation, the subject of Chapters 4 and 5, the communication constraint was that nodes had to transmit their messages via noisy channels, some of which may have capacity zero, that is, not all nodes need to be interconnected. The performance criterion for the nodes, in this case, was that the mean square error in nodes' estimates of the function to be computed be bounded after some amount of exchange of information.

In both of the above formulations, we assumed no computational limitations; our bounds captured limits due to communication constraints. In Chapter 6, however, we began to consider situations where computational limitations arise. We illustrated the use of our lower bound techniques via two examples; and, we suggested that this is an area that is open for further investigation. In the following paragraphs, we recapitulate the main results of this thesis and suggest possible future directions.

7.1 Computation via Compressed Data

In Chapter 2, we looked at compression rates required for communicating nodes to compute a function of all the data in the network with low probability of error. Our formulation is motivated by the problem of estimation, when the sensing is distributed, of a stochastic process that may be generated by a dynamic system. Each source variable represents the measurements collected by one of the nodes. The nodes may communicate their measurements, and, it is desirable to do so after having compressed the outgoing messages (as it leads to savings in power used for transmission, for example). The nodes are assumed to broadcast their messages to all the nodes in the network, which is a situation that may arise in communication via satellites. Even though our formulation (and results) imply that consensus is reached asymptotically, for N sufficiently large, the formulation is still reasonable when the time scale of the evolution of the dynamic system is much slower than the time scale of communication.

The formulation presented in this chapter for studying distributed computation when the communication of the nodes is constrained is useful for three reasons. First, the inequalities that define the rate region are simple; they involve the conditional entropy rates of the function K, which are easy to compute. Second, the formulation can be easily extended to an arbitrary number of nodes, and hence the rate region can easily be defined for an arbitrary number of nodes. Finally, our set-up inherently involves feedback.

We have seen that the lower bound to the boundary of the rate region for K achievability also is a lower bound to the SW omniscience rate region, which means that if the objective of the nodes is to reliably acquire a function of all the data in the network, they may only need to exchange information at rates lower than would be required for each of them to acquire all other nodes' data. We have seen that this is rarely true for the two node case, but, when there are more than 2 nodes, things change. Indeed, in Chapter 3, for the modulo-2 sum computation example, we presented a source for which we showed achievability of a point that does not belong to the omniscience rate region. This point turned out to be the vertex of the cone defining the lower bound to the K-rate region boundary, defined by the inequalities in Theorem 2.2.1. Furthermore, the example was generalized to modulo-q summation when the source random variables take values on a finite field of order q.

The next question that arises naturally is: what are other functions or sources for which we can show that there exist codes with rates arbitrarily close to our lower bound to the "K-rate region" boundary. Another approach is to look for tighter bounds to the "K-rate region" for certain functions of interest. One possibly fruitful avenue is to explore coding schemes that involve feedback, where nodes are allowed to update their estimates of K as they receive messages from other nodes and hence adjust their outgoing messages accordingly.

In another direction, one may seek to characterize the rate region, for the distributed computation set-up of this chapter, under further communication constraints. For example, suppose that the nodes were required to compress their messages, before broadcasting them, but now must do so securely. That is, the encoding must occur such that an eavesdropper cannot compute K. Then, the shape of the rate region must change according to how much secrecy is imposed on the nodes.

7.2 Computation via Noisy Channels

In Chapter 4, our use of basic Information Theoretic definitions and inequalities has led to a lower bound that we have applied to a formulation for distributed function computation. In Chapter 5 we applied the Information Theoretic technique to a scenario where nodes communicate via erasure channels to compute the sum of all the initial values in the network. The lower bound that we obtained is tight in capturing the spectral graph property, conductance, that represents the effect of the topology of the network.

Immediate extensions include generalizing the information rate lower bounds, for example, investigating various distortion measures for the nodes' performance. Another avenue is the use of the tools developed in this thesis to obtain tighter lower bound and performance limitations for restricted computational models, like directional algorithms such as belief propagation. We have illustrated this for two examples in Chapter 6.

More fundamentally, one of the factors that allowed for the derivation of such a simple bound is that we have abstracted away the interdependence of the nodes' estimates as they evolve over time. We believe that this contributes to some "looseness" in the bound; and, if nothing else, it is the reason that the bound does not capture the dynamics of computation. One way that this might be remedied is to consider alternate information quantities, such as directed information.

In a different direction, open is the question of the achievable information rates and corresponding channel codes that can be used in the network setting, especially where the presence of feedback information may be taken into account. We have had the need to consider achievability in this thesis. In particular, in Chapter 5, we presented a simple algorithm for the computation of separable functions via erasure channels. By providing an algorithm that achieves our Information Theoretic lower bound, we showed that our bound is tight. Equivalently, we have produced the fastest algorithm with respect to its dependence on the conductance of the underlying graph. But, because the channels are simple, we did not need to explicitly consider code rates. Further, we did not need to harness the power of feedback in this scenario.

Appendix A

Appendix for Chapter 4

A.1 Proof of Proposition A.1.1

Proposition A.1.1. For $\gamma > 0$, subject to $\sum_{i=1}^n y_i \leq \gamma$ and $y_i \geq 0$, $\prod_{i=1}^n y_i$ is maximized when $y_i = \frac{\gamma}{n}$.

Proof. One way to see this is the following. First, the vector that achieves the maximum, $y^* = [y_1^* \dots y_n^*]'$ must lie on the boundary of the optimization region, meaning that $\sum_{i=1}^n y_i^* = \gamma$; otherwise, if for some $\epsilon > 0$ we have that $\sum_{i=1}^n y_i^* < \gamma - \epsilon$, then for any $\delta \in (0, \epsilon)$, $(y_1^* + \delta) + \sum_{i=2}^n y_i^* < \gamma$ but $(y_1^* + \delta) \prod_{i=2}^n y_i^* > \prod_{i=1}^n y_i^*$, contradicting the optimality of y^*.

Next, substituting $y_1 = \gamma - \sum_{i=2}^n y_i$ in the function we wish to optimize, we obtain $f(y_2, \dots, y_n) = (\gamma - \sum_{i=2}^n y_i) \prod_{i=2}^n y_i$; differentiating f with respect to y_k for $k = 2, \dots n$, we have

$$\frac{d}{dy_k} f = \left(\prod_{\substack{i=2 \\ i \neq k}}^n y_i \right) (\gamma - 2y_k - \sum_{\substack{i=2 \\ i \neq k}}^n y_i).$$

Setting the derivatives to zero for each k, $k = 2, \dots n$ we obtain $n-1$ equations in $n-1$ unknowns, in matrix form:

$$(I_{n-1} + \mathbf{1}_{n-1}\mathbf{1}_{n-1}')\tilde{y} = \gamma \mathbf{1}_{n-1},$$

where I_{n-1} is the identity matrix with dimension $n-1$, $\mathbf{1}_{n-1}$ is the vector of $n-1$ ones and $\tilde{y} = [y_2 \dots y_n]'$.

To solve for $\tilde{y}$, we use the identity, $(I_n + AB)^{-1}A = A(I_m + BA)^{-1}$, for any two matrices $A \in \mathbb{R}^{n \times m}$ and $B \in \mathbb{R}^{m \times n}$. Letting $A = \mathbf{1}_{n-1}$ and $B = A'$,we have that

$$(I_{n-1} + \mathbf{1}_{n-1}\mathbf{1}_{n-1}')^{-1}\mathbf{1}_{n-1} = \mathbf{1}_{n-1}(1 + (n-1))^{-1}.$$

So, $y_i = \frac{\gamma}{n}$.

Finally, computing the second derivatives of f and evaluating at $y_i = \frac{\gamma}{n}$, we have that for $k, j = 2, \dots n$, $\frac{d^2}{dy_k dy_j} f = -2 \left(\frac{\gamma}{n}\right)^{n-2}$. So, the matrix of second derivatives is negative semi-definite. □

A.2 2-Node Network Example

In this section, we illustrate the techniques of Chapter 4 for the special case of 2 nodes. We show an application for which our lower bound can be used to relate the rates of the channel block codes to the channel capacities and the rate of convergence of the mean square error to zero.

Consider two nodes, 1 and 2, communicating via noisy channels. Let $X_1(k)$, $X_2(k)$, and C be continuous scalar random variables. Then, the following lemma will be used to show that if the mean square error in the nodes' estimates is required to converge to zero exponentially, then information flow between the two nodes is lower bounded by a term due to the desired rate of convergence.

Lemma A.2.1. If the mean square error converges to zero with rate r, $E(X_1(k) - C)^2 \leq \beta 2^{-kr}$, where $\beta, r \in \mathbb{R}^+ \backslash \{0\}$, then

$$I(X_1(k); C) \geq h(C) - \frac{1}{2}\log 2\pi e\beta + \frac{kr}{2}.$$

Proof. First, we have that

$$\begin{aligned} I(X_1(k); C) &= h(C) - h(C|X_1(k)) \\ &= h(C) - h(X_1(k) - C|X_1(k)) \\ &\overset{(a)}{\geq} h(C) - h(X_1(k) - C) \\ &\overset{(b)}{\geq} h(C) - \frac{1}{2}\log 2\pi e Var(X_1(k) - C) \\ &\overset{(c)}{\geq} h(C) - \frac{1}{2}\log 2\pi e E(X_1(k) - C)^2, \end{aligned}$$

where,

(a) follows because conditioning reduces entropy,

(b) follows because the Normal distribution maximizes entropy over all distributions with the same variance, and,

(c) follows because for any random variable X, $Var(X) \leq E(X^2)$, and the logarithm is a monotonically increasing function.

□

A similar inequality can be derived for the conditional mutual information, $I(X_1(k); C|W)$, where W is a random variable.

Corollary A.2.2. If $E(X_1(k) - C)^2 \leq \beta 2^{-kr}$ then

$$I(X_1(k); C|W) \geq h(C|W) - \frac{1}{2}\log 2\pi e\beta + \frac{kr}{2}.$$

Proof. We have the same series of inequalities that we encountered in the previous lemma. But, now, we condition all terms on W.

$$\begin{aligned} I(X_1(k); C|W) &= h(C|W) - h(C|X_1(k), W) \\ &= h(C|W) - h(X_1(k) - C|X_1(k), W) \\ &\geq h(C|W) - h(X_1(k) - C). \end{aligned}$$

□

A.2.1 An Application: A Bound for Code Rates

Next, we use the above corollary to relate the rate, R_{12}, of the channel code to the rate of convergence r and the capacity, $\mathbf{C}_{12}$, of a discrete time memoryless channel between nodes 1 and 2. Below, we let U_1 denote the codeword letter at the output of the node 1 encoder, that is transmitted from node 1 through the channel to node 2. V_2 denotes the letter that is received, at the output of the channel, by the node 2 decoder. The relevant variables are shown in Figure A-1.

Each node has access to its own initial condition, $X_i(0)$. The nodes exchange information in order to learn C, a function of both the initial conditions, $C = f(X_1(0), X_2(0))$. We assume that the codeword that is generated by encoder i for the l^{th} transmission depends on node i's initial condition and its past received messages at decoder i; that is, $U_i(l)$ is a function of V_i^{l-1} and $X_i(0)$. Each encoder generates N channel digits or transmissions per node sequence of length k. Finally, we assume that the channels between the two nodes are independent.

The theorem below essentially says that the channel must have enough capacity for both reliable transmission, meaning arbitrarily small probability of decoding error, and exponential convergence of the estimates of the nodes.

Theorem A.2.3. *If node 1 sends information to node 2 via a discrete time memoryless channel with capacity $\mathbf{C}_{12}$, such that the mean square error in node 2's estimate of C converges to zero with rate r, $E(X_2(k) - C)^2 \leq \beta 2^{-kr}$, where $\beta, r \in \mathbb{R}^+ \backslash \{0\}$, then the rate of the channel code, R_{12}, must satisfy*

$$\mathbf{C}_{12} \geq \frac{r}{2} R_{12}.$$

Proof. First, we use the results of Lemmas 4.3.1 and 4.3.2. Setting $S = \{2\}$ and $S^c = \{1\}$,

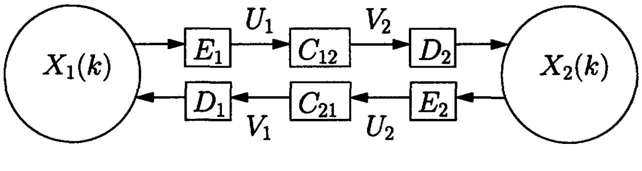

Figure A-1: Set-up for Theorem A.2.3. Two nodes communicating via noisy channels to estimate a function of their initial conditions.

observe that

$$
\begin{aligned}
N\mathbf{C}_{12} &\geq \sum_{l=1}^{N} I(U_1(l); V_2(l)|U_2(l)) \\
&\geq I(X_2(k); X_1(0)|X_2(0)).
\end{aligned}
$$

Next, we have that

$$
\begin{aligned}
I(X_2(k); X_1(0)|X_2(0)) &\overset{(a)}{=} I(X_2(k); X_1(0), X_2(0)|X_2(0)) \\
&\overset{(b)}{\geq} I(X_2(k); C|X_2(0))
\end{aligned}
$$

where,

(a) that is, $I(W;Y,U|U) = I(W;Y|U)$, can be verified by the chain rule for mutual information:

$$
\begin{aligned}
I(W;Y,U|U) &= I(W;Y|U) + I(W;U|U,Y) \\
&= I(W;Y|U),
\end{aligned}
$$

because $I(W;U|U,Y) = 0$.

(b) follows by the data processing inequality, because $C = f(X_1(0), X_2(0))$.

So, from Corollary A.2.2,

$$
\mathbf{C}_{12} \geq \frac{1}{N}\left(h(C|X_2(0)) - \frac{1}{2}\log 2\pi e\beta + \frac{kr}{2}\right),
$$

which holds for all k, and therefore holds as k, hence N, goes to infinity.

Finally, we obtain the desired result by noting the following conditions and facts.

(i) First, $h(C|X_2(0))$ is finite, as long as, given $X_2(0)$, C is random and has a support set of non-zero measure.

(ii) Second, $R_{12} = \lim_{k\to\infty}\frac{k}{N}$ must be finite and smaller than $\mathbf{C}_{12}$ if the probability of decoding error can be made arbitrarily small, according to the Channel Coding Theorem.

□

Appendix B

Appendix for Chapter 5

As in Chapter 5, we 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 from m onwards. 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.

Let $\bar{W}_l = \min_{i=1 \ldots n} (W_l^i)_T$. 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 $y = \sum_i \theta_i$.

Lemma B.0.4. Let $W_l^1, \ldots, W_l^n$ be independent random variables. The cumulative distribution function of $\bar{W}_l$ is identical to that of W_l^*, conditioned on the event $A_l = \{\cap_{i=1}^n A_l^i\}$, where $A_l^i = \{W_l^i \leq m\}$,

$$\mathbf{P}(\bar{W}_l \leq w) = \mathbf{P}(W_l^* \leq w | A_l).$$

Proof.

$$\begin{aligned}
\mathbf{P}(W_l^* \geq w | A_l) &= \mathbf{P}(\min_{i=1,\ldots,n} W_l^i \geq w | \cap_{i=1}^n A_l^i) \\
&\overset{(a)}{=} \mathbf{P}(W_l^1 \geq w, \ldots, W_l^n \geq w | \cap_{i=1}^n A_l^i) \\
&\overset{(b)}{=} \prod_{i=1}^n \mathbf{P}(W_l^i \geq w | A_l^i) \\
&= \prod_{i=1}^n \mathbf{P}((W_l^i)_T \geq w) \\
&\overset{(b)}{=} \mathbf{P}(\cap_{i=1}^n \{(W_l^i)_T \geq w\}) \\
&\overset{(a)}{=} \mathbf{P}(\bar{W}_l \geq w),
\end{aligned}$$

where,

(a) holds because for random variables X_1 and X_2, the events $\{\min\{X_1, X_2\} \geq x\}$ and $\{X_1 \geq x\} \cap \{X_2 \geq x\}$ are equivalent, and,

(b) follows by the independence of the random variables $W_l^1, \ldots, W_l^n$.

□

The above lemma can be used to show the following theorem.

Theorem B.0.5. *Let W_l^i, $i = 1, \ldots, n$, be independent exponentially distributed random variables, each W_l^i with parameter θ_i, where $\theta_i \geq 1$ for all i. Let $(W_l^i)_T$ be the truncated version of W_l^i for which $\mathbf{P}((W_l^i)_T \geq m) = 0$, for some $m > 0$. The expected value of the minimum of the truncated exponentials, $\bar{W}_l = \min_{i=1\ldots n}(W_l^i)_T$, is bounded as follows,*

$$(1 - e^{-ym})^{-1}\left(\frac{1}{y} - \left(\frac{1}{y} + m\right)ne^{-m}\right) \leq E[\bar{W}_l] \leq (1 - ne^{-m})^{-1}\frac{1}{y},$$

where $y = \sum_{i=1}^n \theta_i$.

Remark Note that the above bound implies that for all n, as m approaches infinity, the expected value of $\bar{W}_l$ approaches $\frac{1}{y}$, which is the expected value of W_l^*, as should be the case.

To prove this theorem, we need the following lemma.

Lemma B.0.6. The expectation of W_l^* conditioned on the complement of $A_l = \left\{\cap_{i=1}^n \{W_l^i \leq m\}\right\}$, is upper bounded as,

$$E[W_l^* | A_l^c] \leq \frac{1}{y} + m.$$

Proof. Consider the event $B = \left\{\cap_{i=1}^n \{W_l^i > m\}\right\}$, then it is straightforward to verify that the probability density function of W_l^* conditioned on B is

$$f_{W_l^*|B}(w) = \begin{cases} ye^{-y(w-m)} & \text{for } w > m \\ 0 & \text{otherwise,} \end{cases}$$

which is the probability density function of an exponential random variable with parameter y that is shifted to the right by m. Therefore, $E[W_l^*|B] = \frac{1}{y} + m$.

Next, the result of the theorem follows by noting that

$$E[W_l^*|B] - E[W_l^*|A_l^c] \geq 0.$$

To see this,

$$\begin{aligned} E[W_l^*|B] - E[W_l^*|A_l^c] &= \int_m^\infty w\left(f_{W_l^*|B}(w) - f_{W_l^*|A_l^c}(w)\right)\mathrm{d}w - \int_0^m w f_{W_l^*|A_l^c}(w)\mathrm{d}w \\ &\overset{(a)}{\geq} m\int_m^\infty \left(f_{W_l^*|B}(w) - f_{W_l^*|A_l^c}(w)\right)\mathrm{d}w - m\int_0^m f_{W_l^*|A_l^c}(w)\mathrm{d}w \\ &= m\left(\int_m^\infty f_{W_l^*|B}(w)\mathrm{d}w - \int_0^\infty f_{W_l^*|A_l^c}(w)\mathrm{d}w\right) \\ &\overset{(b)}{=} 0. \end{aligned}$$

Here,

(b) follows because each of the two integrals in the previous line is equal to 1, and,

(a) holds for the first integral because on (m,∞), $w > m$ and $f_{W_l^*|B}(w) - f_{W_l^*|A_l^c}(w) \geq 0$, as will be shown below; and for the second integral because on $[0,m]$, $w \leq m$ and $f_{W_l^*|A_l^c}(w) \geq 0$.

Finally, to show that for $w \in (m,\infty)$, $f_{W_l^*|B}(w) - f_{W_l^*|A_l^c}(w) \geq 0$, we first use the Total Probability Theorem to express $f_{W_l^*|A_l^c}(w)$ as

$$f_{W_l^*|A_l^c}(w) = f_{W_l^*|A_l^c \cap B}(w)\mathbf{P}(B|A_l^c) + f_{W_l^*|A_l^c \cap B^c}(w)\mathbf{P}(B^c|A_l^c).$$

Next, recall that $B = \left\{\cap_{i=1}^n \{W_l^i > m\}\right\}$, and $A_l = \left\{\cap_{i=1}^n \{W_l^i \leq m\}\right\}$. Then,

$$A_l^c \cap B^c = \left\{\{\cap_{i=1}^n \{W_l^i \leq m\}\} \bigcup \{\cap_{i=1}^n \{W_l^i > m\}\}\right\}^c,$$

which says that not all the W_l^i's are larger than m, nor are they all smaller or equal to m; in other words, there is at least one of the W_l^i's that is smaller or equal to m and one that is larger than m. So, conditioned on $A_l^c \cap B^c$, the minimum of the W_l^i's, W_l^*, must be smaller than or equal to m, and $\mathbf{P}(W_l^* \leq m | A_l^c \cap B^c) = 1$. Hence,

$$f_{W_l^*|A_l^c \cap B^c}(w) \begin{cases} \geq 0 & \text{for } 0 \leq w \leq m \\ = 0 & \text{otherwise.} \end{cases}$$

Furthermore, we have that $A_l^c = \left\{\cup_{i=1}^n \{W_l^i > m\}\right\}$, so it is clear that $B \subset A_l^c$ and $B \cap A_l^c = B$ so that $f_{W_l^*|A_l^c \cap B}(w) = f_{W_l^*|B}(w)$. Observing that $\mathbf{P}(B|A_l^c) \leq 1$, we have the desired result, for $w > m$,

$$f_{W_l^*|A_l^c}(w) = f_{W_l^*|A_l^c \cap B}(w)\mathbf{P}(B|A_l^c) \leq f_{W_l^*|B}(w).$$

□

Proof of Theorem B.0.5. It follows from Lemma B.0.4 that $E[\bar{W}_l] = E[W_l^*|A_l]$. By the Total Expectation Theorem,

$$E[W_l^*] = E[W_l^*|A_l]\mathbf{P}(A_l) + E[W_l^*|A_l^c]\mathbf{P}(A_l^c), \tag{B.1}$$

where, because W_l^* is exponentially distributed with parameter y, $E[W_l^*] = 1/y$.

Next, we find bounds on $\mathbf{P}(A_l)$ and $\mathbf{P}(A_l^c)$, where $A_l = \left\{\cap_{i=1}^n \{W_l^i \leq m\}\right\}$ and $A_l^c = \left\{\cup_{i=1}^n \{W_l^i > m\}\right\}$. Although it is clear that

$$\mathbf{P}(A_l) = \prod_{i=1}^n (1 - e^{-\theta_i m}),$$

it is convenient to find upper and lower bounds for $\mathbf{P}(A_l)$. First, we have that by the union bound,

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

where the last inequality follows because, for all i, $\theta_i \geq 1$. So,

$$\mathbf{P}(A_l) \geq 1 - ne^{-m}.$$

Next, observe that $A_l^c \supset \left\{\cap_{i=1}^n \{W_l^i > m\}\right\}$, so

$$\mathbf{P}(A_l^c) \geq \mathbf{P}(\{\cap_{i=1}^n \{W_l^i > m\}\}) = \prod_{i=1}^n e^{-\theta_i m} = e^{-ym}.$$

So,

$$\mathbf{P}(A_l) \leq 1 - e^{-ym}.$$

Finally, for the upper bound on $E[W_l^*|A_l]$, because $E[W_l^*|A_l^c] \geq 0$, we have from equation (B.1),

$$E[W_l^*] \geq E[W_l^*|A_l]\mathbf{P}(A_l),$$

and, rearranging,

$$E[W_l^*|A_l] \leq \frac{1}{\mathbf{P}(A_l)} E[W_l^*] \leq (1 - ne^{-m})^{-1} \frac{1}{y}.$$

For the lower bound on $E[W_l^*|A_l]$, we rearrange equation (B.1),

$$E[W_l^*|A_l] = \frac{1}{\mathbf{P}(A_l)} \left(E[W_l^*] - E[W_l^*|A_l^c]\mathbf{P}(A_l^c)\right).$$

Using the fact that $E[W_l^*|A_l^c] \leq \frac{1}{y} + m$, we obtain the desired result,

$$E[W_l^*|A_l] \geq (1 - e^{-ym})^{-1} \left(\frac{1}{y} - \left(\frac{1}{y} + m\right) ne^{-m}\right).$$

□

Bibliography

[1] R. Ahlswede and I. Csiszar. To get a bit of information may be as hard as to get full information. *IEEE Transactions on Information Theory*, IT-27(4):398–408, July 1981.

[2] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis. Convergence in multiagent coordination, consensus, and flocking. In *Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC'05)*, Seville, Spain, 2005. IEEE.

[3] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6):2508–2530, June 2006.

[4] T.M. Cover. A proof of the data compression theorem of Slepian and Wolf for ergodic sources. *IEEE Transactions on Information Theory*, pages 226–228, March 1975.

[5] T.M. Cover and J.A. Thomas. *Elements of Information Theory.* Wiley-Interscience, 1991.

[6] I. Csiszar and J. Korner. Towards a general theory of source networks. *IEEE Transactions on Information Theory*, IT-26(2):155–165, March 1980.

[7] I. Csiszar and P. Narayan. Secrecy capacities for multiple terminals. *IEEE Transactions on Information Theory*, 50(12):3047–3061, December 2004.

[8] A. Dembo and O. Zeitouni. *Large Deviation Techniques and Applications.* Springer, 1998.

[9] A. El-Gamal and T.M. Cover. Multiple user information theory. *Proceedings of the IEEE*, 68(12):1466–1483, December 1980.

[10] A. El Gamal and A. Orlitsky. Interactive data compression. In *Proceedings of the 25th IEEE Symposium on Foundations of Computer Science*, pages 100–108. IEEE, October 1984.

[11] R. Gallager. Finding parity in a simple broadcast network. *IEEE Transactions on Information Theory*, 34:176–180, 1988.

[12] R.G. Gallager. *Information Theory and Reliable Transmission.* John Wiley and Sons, Inc., 1968.

[13] A. Giridhar and P. R. Kumar. Towards a theory of in-network computation in wireless sensor networks. *IEEE Communications Magazine*, 44(4):98–107, April 2006.

[14] N. Goyal, G. Kindler, and M. Saks. Lower bounds for the noisy broadcast problem. *SIAM Journal on Computing*, 37(6):1806–1841, March 2008.

[15] A. Jadbabaie, J. Lin, and S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

[16] J. Korner and K. Marton. How to encode the modulo-two sum of binary sources. *IEEE Transactions on Information Theory*, IT-25(2):219–221, March 1979.

[17] O. P. Kreidl. *Graphical models and message passing algorithms for network-constrained decision problems.* PhD dissertation, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Laboratory of Information and Decision Systems, 2008.

[18] Z.-Q. Luo and J. N. Tsitsiklis. Data fusion with minimal communication. *IEEE Transactions on Information Theory*, 40(5):1551–1563, September 1994.

[19] N. C. Martins. *Information Theoretic aspects of the control and mode estimation of stochastic systems.* PhD dissertation, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Laboratory of Information and Decision Systems, August 2004.

[20] N. C. Martins. Finite gain lp stabilization requires analog control. *Systems and Control Letters*, 55/1:949–954, Novemeber 2006.

[21] N. C. Martins and M. A. Dahleh. Feedback control in the presence of noisy channels: "bode-like" fundamental limits of performance. *IEEE Transaction on Automatic Control*, May 2008.

[22] N. C. Martins, M. A. Dahleh, and J. C. Doyle. Fundamental limitations of disturbance attenuation in the presence of side information. *IEEE Transaction on Automatic Control*, 52:56–66, January 2007.

[23] L. Moreau. Stability of multiagent systems with time-dependent communication links. *IEEE Transactions on Automatic Control*, 50(2):169–182, 2005.

[24] D. Mosk-Aoyama and D. Shah. Computing separable functions via gossip. In *ACM Principles of Distributed Computation*, 2006.

[25] R. Olfati-Saber. Distributed kalman filter with embedded consenus filters. Technical report ucla-engr-05-256, School of Engineering and Applied Science, University of California, Los Angeles, 2005.

[26] R. Olfati-Saber and R.M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, September 2004.

[27] A. Orlitsky. Worst-case interactive communication I: Two messages are almost optimal. *IEEE Transactions on Information Theory*, 36:1111–1126, September 1990.

[28] A. Orlitsky and J. R. Roche. Coding for computing. *IEEE Transactions on Information Theory*, 47:903–917, March 2001.

[29] M. S. Pinsker. *Information and Information Stability of Random Variables and Processes.* Holden-Day, Inc., San Francisco, 1964.

[30] S. Tatikonda and S. Mitter. Control under communication constraints. *IEEE Transactions on Automatic Control*, 49(7):1056–1068, 2004.

[31] W. P. Tey and J. N. Tsitsiklis. Error exponents for decentralized detection in tree networks. In V. Saligrama, editor, *Networked Sensing Information and Control*, pages 73–92. Springer Verlag, 2008.

[32] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9):803–812, 1986.

[33] J.N. Tsitsiklis. *Problems in Decentralized Decision Making and Computation.* PhD dissertation, Massachusetts Institute of Technology, Laboratory of Information and Decision Systems, Department of Electrical Engineering and Computer Science, 1984.

[34] A.D. Wyner. Recent results in the Shannon theory. *IEEE Transactions on Information Theory*, IT-20(1):2–10, January 1974.

[35] A.D. Wyner, J.K. Wolf, and F.M.J. Willems. Communication via a processing broadcast satellite. *IEEE Transactions on Information Theory*, 48(6):1243–1249, June 2002.

[36] J.-J Xiao and Z.-Q. Luo. Decentralized estimation in an inhomogeneous sensing environment. *IEEE Transactions on Information Theory*, 51(10):3564–3575, October 2005.

[37] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *Fourth International Symposium on Information Processing in Sensor Networks*, pages 63–70. IEEE, 2005.

[38] A. C. Yao. Some complexity questions related to disrtributed computing. In *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing*, pages 209–213, 1979.