Distributed Statistical Learning under Communication Constraints

by

Mostafa El Gamal

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Doctor of Philosophy

in

Electrical and Computer Engineering

by

June 2017

APPROVED:

Professor Lifeng Lai, Major Thesis Advisor

Professor Alexander Wyglinski, Department of Electrical and Computer Engineering

Professor Randy Paffenroth, Department of Mathematical Sciences

Abstract

In this thesis, we study distributed statistical learning, in which multiple terminals, connected by links with limited capacity, cooperate to perform a learning task. As the links connecting the terminals have limited capacity, the messages exchanged between the terminals have to be compressed. The goal of this thesis is to investigate how to compress the data observations at multiple terminals and how to use the compressed data for inference.

We first focus on the distributed parameter estimation problem, in which terminals send messages related to their local observations using limited rates to a fusion center that will obtain an estimate of a parameter related to the observations of all terminals. It is well known that if the transmission rates are in the Slepian-Wolf region, the fusion center can fully recover all observations and hence can construct an estimator having the same performance as that of the centralized case. One natural question is whether Slepian-Wolf rates are necessary to achieve the same estimation performance as that of the centralized case. In this thesis, we show that the answer to this question is negative. We establish our result by explicitly constructing an asymptotically minimum variance unbiased estimator (MVUE) that has the same performance as that of the optimal estimator in the centralized case while requiring information rates less than the conditions required in the Slepian-Wolf rate region. The key idea is that, instead of aiming to recover the observations at the fusion center, we design universal schemes enabling the fusion center to compute a sufficient statistic using rates outside of the Selpian-Wolf region.

We then examine the optimality of data dimensionality reduction via sufficient statistics compression in distributed parameter estimation problems. The data dimensionality reduction step is often needed especially if the data has a very high dimension and the

communication rate is not as high as the one characterized above. We show that reducing the dimensionality by extracting sufficient statistics of the parameter to be estimated does not degrade the overall estimation performance in the presence of communication constraints. We establish this result by comparing two system models, one applies the compression scheme to raw observations, and the other applies the compression scheme to the extracted sufficient statistics. We prove that both system models have the same performance measured by the Bayesian risk. We further analyze the optimal estimation performance in the presence of communication constraints and we verify the derived bound using simulations.

Finally, we study distributed optimization problems, for which we examine the randomized distributed coordinate descent algorithm with quantized updates. In the literature, the iteration complexity of the randomized distributed coordinate descent algorithm has been characterized under the assumption that machines can exchange updates with an infinite precision. We consider a practical scenario in which the messages exchange occurs over channels with finite capacity, and hence the updates have to be quantized. We derive sufficient conditions on the quantization error such that the algorithm with quantized update still converge. We extend our results to the general case of block coordinate descent, and we analyze the convergence rate for the parallel scenario whether the machines are synchronized or not. We further verify our theoretical results by running an experiment, where we apply the algorithm with quantized updates to solve a linear regression problem.

Acknowledgements

First of all I would like thank my advisor Professor Lifeng Lai for his guidance and support throughout my research. He encouraged and motivated me to improve my analytical thinking and research quality.

I would also like to thank Professor Alexander Wyglinski and Professor Randy Paffenroth for not only serving as committee members for this thesis, but also taking personal interest to help me grow as a researcher.

I would like to thank my peers and my friends specially Ain-ul-Aisha, Wenwen Tu, Wenwen Zhao, Bingwen Zhang, Jun Geng, and Bo Jiang, for keeping a healthy competitive environment around me and making sure that I push myself harder.

I would like to thank the WPI ECE department office personnel, facilities, students and professors for maintaining a creative environment. I would like to personally thank Cathy and Colleen from the WPI ECE office for their utmost support.

Most importantly I would like to thank my parents and brothers for supporting me especially in tough times. I would not reach this point in my life without their amazing and unconditional support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Increased data sources in the recent years has made it important to find efficient methods to analyze the data. Statistical learning provides a number of methods that process the data to estimate an unknown parameter, find a function that relates different data variables, classify the data points into different categories, and more.

Statistical learning is playing an increasingly important role in multiple areas, such as artificial intelligence, biology, finance, and marketing. Some examples of learning tasks are:

- Estimating the location of an object using the received measurements from multiple sensors.

- Predicting whether a patient will have a heart attack based on clinical variables for the patient.

- Teaching a robot to read a handwritten ZIP code.

- Predicting the price of a stock based on the company performance and other economic variables.

To perform a learning task, it is typical to collect measurements related to the task. These measurements are divided into a training dataset and a test dataset. The training dataset is used to train a statistical model to solve the learning problem. The model is then tested using the test dataset to verify its accuracy. Different statistical models go through the same process until a satisfying test accuracy is reached.

It is often required to analyze the measurements data in a distributed fashion, where multiple machines cooperate to perform a specific learning task. The data can be distributed over a number of machines for multiple reasons that include, but are not limited to:

- The large size of data prohibits locating all data on a single machine, which is the case for the big data problem.
- The data are collected and hence are naturally distributed over multiple geographical locations.
- It is required to process the data in parallel to speed up the analysis.

Figure 1.1: The channels connecting the machines are capacity limited.

To solve the problem efficiently, multiple rounds of communication is often required between the machines in order to perform the learning task. The major challenge for that setup is that the communication channels between the machines have limited capacity. This requires each machine to compress its message before sending it to other machines.

In this thesis, we study how to compress data for statistical inference purpose and how to perform inference using compressed data.

1.1 Motivation and Literature Review

The key role of distributed statistical learning has motivated many researchers to study the problem rigorously [1–16]. In this thesis, we focus on two types of distributed statistical learning, namely distributed parameter estimation and distributed optimization.

1.1.1 Distributed Parameter Estimation

Motivated by applications in sensor networks and other areas, the problem of distributed estimation has been extensively investigated from various perspective [17–32]. As observations are distributed over multiple terminals in the distributed setting, the performances of distributed estimators are no better than those of centralized estimators who have access to all observations. The questions we address in this thesis are: 1) to achieve the same performance as that of the centralized setup, how much information has to be exchanged in the distributed setting; 2) for a given rate constraints, what is the optimal data reduction method. In particular, we study the rate requirements for the information exchange, and whether it is optimal or not to reduce the dimension of the data prior to applying the compression scheme.

A. Rate Requirements

We consider this problem for the following setup. There are two random variables (X, Y) with a joint probability mass function (PMF) $P_\theta(X, Y)$ parameterized by an unknown parameter θ. Two terminals A and B observe X^n and Y^n respectively and send messages related to their own local observations with limited rates to terminal C, which will then obtain an estimate of the unknown parameter. It is well known that if the transmission rates from the terminals are inside the Slepian-Wolf rate region [33], there exists a universal coding scheme [34] that enables terminal C to fully recover (X^n, Y^n). Hence, once the transmission rates are inside the Slepian-Wolf rate region, the performance of the best estimator for the distributed setup is the same as that of the best estimator for the centralized case. In this thesis, we focus on unbiased estimators, and we define the best centralized estimator as the unbiased estimator that achieves the minimum variance index, whose precise definition will be provided in Section 2.1, in the centralized setup. We use the centralized performance to refer to the performance of the best centralized estimator.

One natural question is: are Slepian-Wolf rates *necessary* to achieve the same estimation performance as that of the centralized case? The answer to this question has significant implications in the distributed estimation. If the answer is yes, then to obtain the best estimate of the unknown parameter requires transmission rates to be so high that they are sufficient to fully recover the observations at the decoder, hence no rate reduction is possible. On the other hand, if the answer is no, then the observations can be compressed beyond the limits of source coding for full observation recovery. At a first glance, the answer to this question should be no as we are only interested in estimating a parameter related to the observations and are not interested in recovering the observations themselves. However, all existing related works indicate otherwise. For example, [35] addressed the same question and suggested that Slepian-Wolf rates might be necessary. In addition, the performance of the best known estimator by Han and Amari [36] does not match that of the centralized case when the information rates are outside of the Slepian-Wolf rate region. Furthermore, [37] showed that, under certain conditions, extracting even one bit of information from distributed sources is as hard as recovering full observations and hence requires the information rates to be in the Slepian-Wolf rate region.

In this thesis, we compare our results to the best known estimator by Han and Amari [36]. In [36], the authors established their estimation algorithm by introducing auxiliary random variables and solving the maximum-likelihood equation. They showed that their estimation algorithm achieves a smaller variance than the estimator by Zhang and Berger [38].

B. Dimensionality Reduction

In some cases, where the dimension of the observations is very high and the communication rate is low, it is necessary to reduce the data dimensionality before applying the compression scheme [39, 40]. Similar to the concept of transductive inference [41], it is preferable to solve a simple version of the problem rather than solving the general one. In this case, one simple version of the problem is to apply the compression scheme after reducing the dimensionality of the observations, while the general problem is to compress high dimensional observations.

For distributed parameter estimation, a sufficient statistic contains all relevant information of the unknown parameter [42–44]. Hence, it is natural to reduce the dimension of the observations by extracting a sufficient statistic. One question that follows is whether this process degrades the estimation performance of the network in the presence of communication constraints. In other words, is it optimal to extract a sufficient statistic from the observations before applying the compression scheme?

For a set of n observations X^n generated according to the probability distribution $f(X|\theta)$ with θ as the unknown parameter, a sufficient statistic $T(X^n)$ is defined such that

$$f(X^n|(T(X^n),\theta)) = f(X^n|T(X^n)). \tag{1.1}$$

Some examples of sufficient statistics are:

- In the coin flip experiment, the number of heads is a sufficient statistic of the coin bias.
- In digital communications, the output of a matched filter at the receiver side is a sufficient statistic of the transmitted symbol [45].

The above question has been answered for the centralized scenario and a specific setup of the decentralized scenario in [17]. The author assumed that the observations are distributed, such that each machine has all observations about a single random variable. For that setup, it was shown that sufficiency based data reduction is only optimal if the observations are conditionally independent or if the data has a specific structure (HCI structure). In this thesis, we answer the above question for the decentralized scenario, where each machine has a few observations about all random variables.

1.1.2 Distributed Optimization

Similar to distributed parameter estimation, distributed optimization problems naturally arise in various scenarios. For example, in solving regression problems, the training dataset might be too large to be stored in a single machine, or the data might be collected (and hence is naturally located) at multiple locations. This motivated many researchers to develop algorithms to solve distributed optimization problems. Distributed algorithms are also useful to harness parallel processing capabilities of multiple machines.

In distributed optimization, it is essential for machines involved to exchange messages. As communication links between machines have limited capacity and have significantly longer delay, many recent papers focus on developing algorithms that are communication efficient. In [46], an algorithm was proposed to reduce the amount of necessary communication by using the local computation in a primal-dual setting. Another communication efficient algorithm for empirical risk minimization was introduced in [47]. ADMM was considered in [48–50] to handle the communication bottleneck.

Most of the existing studies analyze how many rounds of communications are required for the convergence of the developed algorithms. In each communication round, it is typically assumed that machines can exchange messages with an infinite precision. However, in practice, these data exchanges occur over physical channels that have limited capacity. As a result, machines cannot exchange messages with an infinite precision and need to quantize messages before sending them to other machines. A natural question to ask is whether these distributed algorithms will still converge if the exchanged messages are quantized. If these algorithms still converge, one can further ask what are the effects of the quantization on the converge rate.

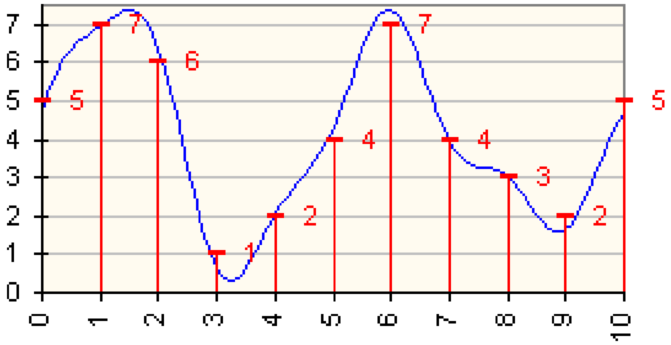

Figure 1.2: The machines need to quantize their messages.

In this thesis, we answer these questions for a particular optimization algorithm, namely randomized coordinate descent [51]. This algorithm is easily implementable to solve distributed optimization problems since each machine can compute a single coordinate of the gradient. In each iteration of the randomized coordinate descent, the algorithm takes a step in the direction of a randomly chosen coordinate in order to decrease the function value. This is done by computing the partial derivatives, which is much cheaper computationally than taking a full gradient step. The iteration complexities of the randomized coordinate descent algorithms are analyzed in [52,53] under a very general setup. In [54], a hybrid coordinate descent method (Hydra) was presented to speed up the coordinate descent algorithm. Asynchronous parallel processing was analyzed in [55] for a number of optimization algorithms including the randomized coordinate descent.

1.2 Summary of Contributions

1.2.1 On Rate Requirements for Distributed Parameter Estimation

We show that the answer to the question (are Slepian-Wolf rates *necessary* to achieve the same estimation performance as that of the centralized case?) is indeed *no*. We establish our result by explicitly constructing a distributed estimation algorithm that achieves the same performance as that of the optimal estimator for the centralized case while using information rates outside of the Slepian-Wolf region. The main observation is that, to construct an estimator that has the same performance as that of the centralized case, the fusion center needs only sufficient statistics not full data. Based on this observation, the key idea of our algorithm is that, instead of trying to fully recover the source observations, we design schemes that enable the fusion center to recover sufficient statistics using less information rates. We study the case, in which each terminal has all observations related to a single random variable, and we compute a global sufficient statistic efficiently at the fusion center using compressed versions of the local observations while using rates outside of the Slepian-Wolf rate region.

To illustrate the idea, we first consider binary symmetric sources (i.e., both X^n and Y^n are binary sequences) parameterized by an unknown parameter θ. For this model, in our algorithm, we first design a universal coding/decoding scheme that enables terminal C to compute component-wise module-two sum $Z^n = X^n \oplus Y^n$, which can be achieved using rates outside of the Slepian-Wolf rate region, and then construct an estimator using Z^n. Here $\oplus$ denotes element-wise xor. We show that our estimator is an asymptotically minimum variance unbiased estimator (MVUE) [56] and achieves the same variance index as that of the best estimator in the centralized case. We then generalize our study to general binary sources models that are not necessarily symmetric anymore. Compared with the symmetric case, there are two additional challenges: 1) Z^n alone is not a sufficient statistic anymore; and 2) We do not have an MVUE to compare the performance to anymore, as it is not clear whether an MVUE exists and even if it exists its form is model dependent. To address the first issue, we modify our scheme and ask the transmitters to send additional information (more specifically, empirical marginal PMF) that requires diminishing rate. Combining Z^n with these additional information, the fusion center can then construct the empirical joint PMF, which is a sufficient statistic. To address the second issue, we show a stronger result that for any centralized estimator, we can construct a plugin estimator with the same performance by using the only decoded information at terminal C. We further extend our results to a more general class of non-binary sources and show that our algorithm can also achieve the same performance as that of the best estimator in the centralized case while using transmission rates less than the conditions required in the Slepian-Wolf rate region. Finally, although our estimation algorithm achieves the centralized performance at rates less than Slepian-Wolf rates, there is no optimality guarantee at very low rates which can be the case for a number of practical applications. To address this, we propose a practical design of our estimation algorithm and show that it outperforms the best known estimator by Han and Amari [36] at all rates.

1.2.2 Sufficiency Based Data Reduction

We answer the question of whether it is optimal or not to extract a sufficient statistic from the observations before applying the compression scheme. We show that the answer is positive. We establish this result by considering a set of n observations X_1^n related to the random variable $X \in \mathcal{X}$. The observations are distributed between two nodes, such that node 1 has access to the observations $X_1^{n_1}$ and node 2 has access to $X_{n_1+1}^n$. The observations are independent and identically distributed (i.i.d.) and generated according to the parametric probability distribution $f(X|\theta)$, where $\theta \in \Theta$ is the unknown parameter.

To answer this question, we compare the performance of two system models (a) and (b), while applying the same compression rate pair (R_1, R_2) to both models. In system model (a), node 1 and node 2 compress the observations $X_1^{n_1}$ and $X_{n_1+1}^n$ using the encoding functions $g_1(\cdot) \in \mathcal{G}_1$ and $g_2(\cdot) \in \mathcal{G}_2$, respectively. The fusion center receives the compressed messages and applies the decoding function ϕ_a to get an estimate $\hat{\theta}_a$ of the unknown parameter θ. In (b), an additional step is added to extract the sufficient statistics $T_1(X_1^{n_1})$ and $T_2(X_{n_1+1}^n)$ from the observations, then compress them using the encoding functions $h_1(\cdot) \in \mathcal{H}_1$ and $h_2(\cdot) \in \mathcal{H}_2$, respectively. The fusion center uses the decoding function ϕ_b to get an estimate $\hat{\theta}_b$ of the unknown parameter.

We show that the two system models have the same estimation performance using the Bayesian risk as the performance metric. We also analyze the asymptotic optimal Bayesian performance in the presence of communication constraints. We further verify our results through simulations as we plot the simulated distortion for system model (b) and we compare it the asymptotic optimal Bayesian performance.

1.2.3 Distributed Optimization with Quantized Updates

We answer the question of whether distributed optimization algorithms can converge in the presence of quantization error by first modifying a distributed version of the coordinate descent algorithm to fit the paradigm of capacity limited communication. We then determine sufficient conditions on the quantization error such that the algorithm converges to the optimal solution. In particular, we apply our algorithm to an unconstrained minimization problem of a function f that is L-smooth and m-strongly convex. We show that for an accuracy level ϵ and a confidence level ρ, our algorithm converges to the optimal solution if the quantization error Δ is upper bounded by a function of ϵ, ρ, L, m, and d, where d is the number of features.

We further extend our results to the general case of randomized block coordinate descent, where each machine can update a block of coordinates. We consider two scenarios: First, the selected machine sends the update for all its coordinates. Second, the selected machine samples a subset of its coordinates and sends the update for that subset. It is obvious that the first scenario converges faster than the second, but it requires more computational power per machine.

We also analyze the convergence rate when all machines can send their updates in parallel. We consider two scenarios: First, all machines are synchronized to process the same update, which requires the fast machines to wait for the slow ones before they can process the next update. Second, we consider the asynchronous scenario, where different machines can process different updates depending on their individual speeds. The convergence analysis of the second scenario can be challenging especially in the presence of quantization error. Therefore, we analyze a special case of two machines, where machine 1 is twice as fast as machine 2. We compare both results in the synchronous and asynchronous scenarios.

We verify the results by running an experiment, where we apply our algorithm to solve a linear regression problem. The dataset we use is collected from a power plant and consists of one output and four predictors. We show that our algorithm converges to the optimal solution if the quantization error is relatively small, which coincides with our theoretical results.

Chapter 2

On Rate Requirements for Distributed Parameter Estimation

In this chapter, we study the rate requirements for distributed parameter estimation schemes to achieve the optimal centralized performance. The chapter is organized as follows. We introduce the problem formulation in Section 2.1. In Section 2.2, we establish our main results for binary symmetric sources, then we generalize it to non-symmetric binary sources in Section 2.3. We extend our work to a more general class of information sources in Section 2.4, and to multiple source networks in Section 2.5. We propose a practical design of our estimation algorithm in Section 2.6. We present the simulation results in Section 2.7.

2.1 Problem Formulation

Consider two information sources X and Y taking values from the discrete alphabets $\mathcal{X}$ and $\mathcal{Y}$, respectively. $(X^n, Y^n) = \{(X_i, Y_i)\}_{i=1}^n$ are n independent and identically distributed (i.i.d.) observations drawn according to the parametric joint PMF $P_\theta(X, Y)$

where $\theta \in \Theta$ is the unknown parameter. We assume that the range of Θ is bounded and hence $\theta_u \triangleq \max\{|\inf(\Theta)|, |\sup(\Theta)|\}$ is finite. We consider a distributed setup in which X^n are observed at terminal A and Y^n are observed at terminal B. Using limited rates, these two terminals send messages related to their own local observations to a fusion center (terminal C), which will then obtain an estimate $\hat{\theta}$ of θ using these messages. The setup is illustrated in Fig. 2.1.

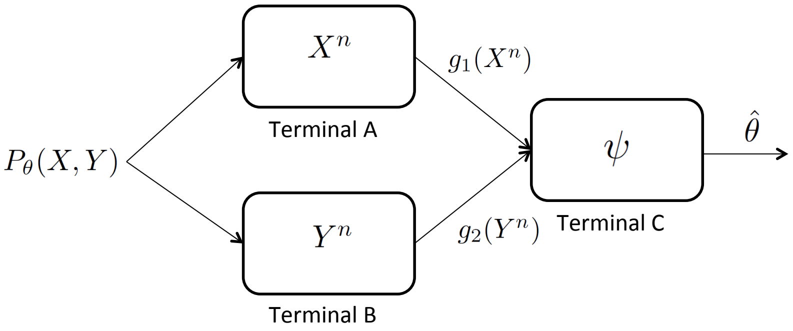

Figure 2.1: System Model.

In particular, terminal A employs an encoding function $g_1 : X^n \to g_1(X^n)$, while terminal B employs an encoding function $g_2 : Y^n \to g_2(Y^n)$. The code rates are

$$R_X = \frac{\log ||g_1||}{n}, R_Y = \frac{\log ||g_2||}{n}, \tag{2.1}$$

where $||g_i||$ is the cardinality of the encoding function g_i.

From $g_1(X^n)$ and $g_2(Y^n)$, the decoder obtains an estimate $\hat{\theta}$ of the unknown parameter θ using estimator ψ:

$$\hat{\theta} = \psi(g_1(X^n), g_2(Y^n)). \tag{2.2}$$

To evaluate the quality of the estimator, we use the variance index that is defined as[1]

$$V_\theta[\hat{\theta}] = \lim_{n\to\infty} n\text{Var}_\theta[\hat{\theta}] = \lim_{n\to\infty} n\mathbb{E}_\theta[(\hat{\theta} - \mathbb{E}[\hat{\theta}])^2]. \tag{2.3}$$

It is desirable to have an estimator that is asymptotically unbiased, i.e., $\mathbb{E}_\theta[\hat{\theta}] \to \theta$ as $n \to \infty$, range-preserving, i.e., the range of the estimation function ψ is Θ, and has a small variance index.

It is well-known that, if the coding rates satisfy (will be called Slepian-Wolf rates in the sequel)

$$R_X \geq H_\theta(X|Y), \tag{2.4}$$

$$R_Y \geq H_\theta(Y|X), \tag{2.5}$$

$$R_X + R_Y \geq H_\theta(X,Y), \tag{2.6}$$

there exists universal source coding schemes [34] (i.e., the coding scheme does not depend on the value of the unknown parameter θ) such that the decoder can reconstruct X^n and Y^n with a diminishing error probability. Here, $H_\theta(\cdot)$ and $H_\theta(\cdot|\cdot)$ denote the entropy and conditional entropy respectively. Hence, if (2.4)-(2.6) are satisfied, we can obtain the same estimation performance as that of the centralized case.

The question we ask in this chapter is: are Slepian-Wolf rates *necessary* to achieve the same estimation performance as that of the centralized case? [35] investigated the same question and suggested that Slepian-Wolf rates appear to be necessary for achieving the centralized estimation performance. We show that Slepian-Wolf rates are *not* necessary. In particular, we show that there indeed exists a class of PMFs and the corresponding distributed estimators that require communication rates less than the Slepian-Wolf rates

[1] Throughout the chapter, we use the subscript θ to emphasize the fact that value of the quantity of interest depends on the parameter θ.

while still achieving the same performance as that of the best estimator for the centralized case.

Throughout the chapter, we use an upper case letter U to denote a random variable, a lower case letter u to denote a realization of U, and $\mathcal{U}$ to denote the discrete alphabet from which U takes values. For any sequence $u^n = (u(1), \cdots, u(n)) \in \mathcal{U}^n$, the relative frequencies (empirical PMF) $\pi(a|u^n) \triangleq n(a|u^n)/n, \forall a \in \mathcal{U}$ of the components of u^n is called the type of u^n. Here $n(a|u^n)$ is the total number of indices t at which $u(t) = a$. Chapter 11 of [57] contains a comprehensive overview of useful properties of the type.

2.2 Binary Symmetric Case

In this section, to illustrate our main idea, we first consider the case of binary symmetric sources with $|\mathcal{X}| = |\mathcal{Y}| = 2$ and a joint PMF of (X, Y) as given in Table 2.1, in which the unknown parameter $\theta \in \Theta = (0, 1)$. The insights obtained here will be generalized to more general models in later sections.

X/Y	0	1
0	$\theta/2$	$(1-\theta)/2$
1	$(1-\theta)/2$	$\theta/2$

Table 2.1: The joint PMF of binary symmetric sources.

Note that for this model, neither terminal A nor terminal B alone will be able to obtain a meaningful estimation of the value of θ, as the marginal distributions of X and Y are independent of θ. On the other hand, to estimate θ, the fusion center does not need to know (X^n, Y^n) fully. It is easy to check that the component-wise module-two sum $Z^n = X^n \oplus Y^n \triangleq [X_1 \oplus Y_1, \cdots, X_i \oplus Y_i, \cdots, X_n \oplus Y_n]$ is a sufficient statistic for estimating θ. Hence, as long as the fusion center can compute Z^n, it can construct an estimator that has the same performance as that of the centralized case. Based on this observation, we show that, to estimate θ for this class of PMFs, we can achieve the centralized estimation performance using rates that do not satisfy (2.4)-(2.6). We establish this result using two steps: 1) in the first step, we design a universal encoder at terminals A and B and universal decoder at terminal C to compute the modulo-two sum $Z^n = X^n \oplus Y^n$; 2) in the second step, we construct an estimator using Z^n and analyze its performance.

2.2.1 Step 1: Computing Z^n

Here, we discuss how to universally compute $Z^n = X^n \oplus Y^n$ at terminal C. Towards this goal, we will use the same linear code at both encoders and use a minimum entropy decoder at terminal C.

Since the encoders at terminals A and B are the same, we use the following simplified notation

$$\begin{aligned} f &= g_1 = g_2, \\ R &= R_X = R_Y. \end{aligned} \tag{2.7}$$

The following theorem shows that as long as $R \geq H_\theta(X|Y) = H_\theta(Y|X)$, the decoder can reconstruct Z^n with a diminishing error probability.

Theorem 1. *If*

$$R > H_\theta(X|Y) = H_\theta(Y|X), \tag{2.8}$$

there exist universal encoding/decoding functions to reconstruct $Z^n = X^n \oplus Y^n$ at terminal C with an exponentially decreasing error probability.

Proof. The proof follows a similar structure as the proofs in [58] and [34]. In particular, using the ideas in [34], we modify the proof of [58] to make it universal.

Random Code Generation: We use a linear code f with an encoding matrix A of size $n \times nR$ to map $\{0,1\}^n$ to $\{0,1\}^{nR}$. Hence $||f|| = 2^{nR}$. We independently generate each entry of A using a uniform binary distribution, i.e., each entry of A is 0 or 1 with probability 0.5.

Encoding: The encoded messages of the realizations $x^n \in \{0,1\}^n$ and $y^n \in \{0,1\}^n$ are

$$\begin{aligned} f(x^n) &= x^n A, \\ f(y^n) &= y^n A, \end{aligned} \tag{2.9}$$

in which the operations are all in binary field.

Decoding: The decoder first combines the messages into a single message as

$$f(x^n) \oplus f(y^n). \tag{2.10}$$

It follows from the code linearity that

$$f(x^n) \oplus f(y^n) = f(x^n \oplus y^n) = f(z^n). \tag{2.11}$$

From $f(x^n \oplus y^n)$, terminal C uses a minimum entropy decoder to obtain $\hat{z}^n$. In particular, for each $\bar{z}^n$ such that $f(\bar{z}^n) = f(x^n \oplus y^n)$, the minimum entropy decoder first calculates the entropy of its type, then picks the one that has the least entropy to be the decoded sequence. In the following, to simplify the notation, we use $\bar{Z}^{(n)}$ and $Z^{(n)}$ to denote dummy random variables whose PMFs $P_{\bar{Z}^{(n)}}$ and $P_{Z^{(n)}}$ are the same as the types of $\bar{z}^n$ and z^n, respectively. The final decoded message is denoted as

$$\hat{z}^n = \phi(f(z^n)), \tag{2.12}$$

where ϕ denotes the minimum entropy decoding function.

Error Probability Analysis: A decoding error occurs if and only if there exists a sequence $\hat{z}^n \neq z^n$ such that

$$f(\hat{z}^n) = f(z^n) \ \text{ and } \ H(\hat{Z}^{(n)}) \leq H(Z^{(n)}). \tag{2.13}$$

The error probability, averaging over all possible codebooks, is

$$P_e^{(n)} = \sum_{z^n \in \{0,1\}^n} P_\theta(z^n)\text{Pr}(\hat{z}^n \neq z^n) = \sum_f \text{Pr}(f) P_{e,f}^{(n)}, \tag{2.14}$$

in which $P_\theta(z^n) \triangleq \text{Pr}(Z^n = z^n)$, and $P_{e,f}^{(n)}$ denotes the error probability if a particular codebook f is used. By analyzing (2.14), we show that there exists a particular codebook f^* such that $P_{e,f^*}^{(n)} \to 0$ exponentially as $n \to \infty$ as long as the conditions in the theorem are satisfied. Detailed analysis can be found in Appendix A. This implies that if we use f^*, then the fusion center will be able to compute Z^n with an exponentially decreasing error probability.

□

Theorem 1 implies that the required rates to decode $Z^n = X^n \oplus Y^n$ with a small error probability is

$$R_X > H_\theta(X|Y), \tag{2.15}$$

$$R_Y > H_\theta(Y|X). \tag{2.16}$$

This rate region is larger than the Slepian-Wolf region in (2.4)-(2.6), as the condition $R_X + R_Y \geq H_\theta(X, Y)$ is not necessary anymore.

2.2.2 Step 2: Estimation

After obtaining $\hat{Z}^n$, which is equal to Z^n with a probability converging to 1 exponentially, we then design an asymptotically MVUE of θ. Our estimator is

$$\hat{\theta} = \frac{n(0|\hat{Z}^n)}{n}, \tag{2.17}$$

in which the notation $n(\cdot|\cdot)$ is defined in Section 2.1.

Theorem 2. *If the conditions in Theorem 1 are satisfied, the estimator in* (2.17) *is an asymptotically MVUE and achieves the optimal variance index as that of the centralized case.*

Proof. We establish this result by showing that the estimator (2.17) achieves the same performance as that of the optimal estimator in the centralized case. Detailed analysis can be found in Appendix B. □

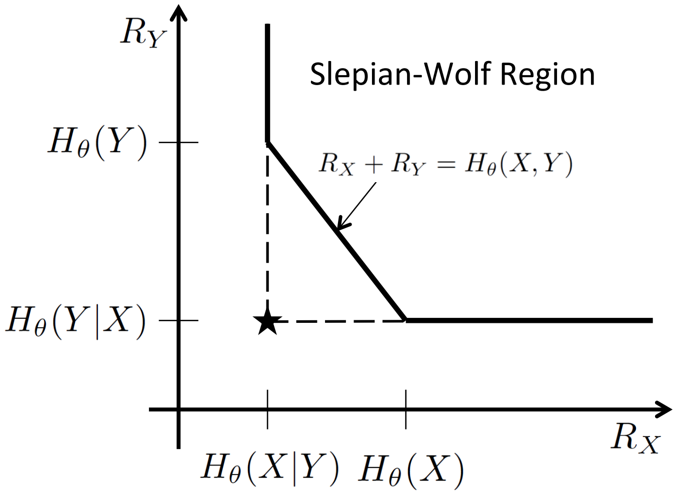

Figure 2.2: Our estimator is optimal at any rate larger than the rate pair indicated by (★), which is outside of the Slepian-Wolf rate region.

Combining Theorems 1 and 2, we conclude that, in the distributed parameter estimation, the Slepian-Wolf rates are not necessary to achieve the same optimal estimation performance as that of the centralized case. Fig. 2.2 illustrates the comparison between the Slepian-Wolf rate region and the rate pair used in our estimator.

2.3 General Binary Case

In this section, we extend our study to the general binary source models $P_\theta(X, Y)$. Here, we do not make any particular assumption of the form of $P_\theta(X, Y)$. For example, $P_\theta(X, Y)$ could be a nonlinear function of θ. Similar to the previous section, we assume that $P_\theta(X = i, Y = j) > 0$ for all $\theta \in \Theta$ and $i, j \in \{0, 1\}$. Compared with the binary symmetric source model considered in Section 2.2, there are two additional challenges. First, the component-wise module-two sum Z^n is not a sufficient statistic in general, hence recovering Z^n alone is not enough. Second, unlike the symmetric case in which we have an MVUE centralized estimator to compare to, we cannot do that anymore as we are considering general models whose optimal centralized is model specific (and in some cases, MVUE may not exist). Despite these challenges, we prove the following result:

Theorem 3. *For any binary source with a parametric PMF $P_\theta(X, Y)$, where $\theta \in \Theta$ is the unknown parameter and Θ is a bounded set, there exits an unbiased estimator $\hat{F}$ based on $Z^n = X^n \oplus Y^n$ that achieves the centralized performance asymptotically and requires communication rates of*

$$R_X = R_Y > H_\theta(Z). \tag{2.18}$$

Proof. The proof consists of two main steps: 1) in the first step, we construct a scheme to enable the fusion center to compute a sufficient statistic with exponentially diminishing error probability; 2) in the second step, we establish an estimator using the computed statistics and show that the estimator achieves the performance of the centralized estimator. Detailed analysis can be found in Appendix C. □

Depending on the PMF of the binary source, the required sum rate to achieve the optimal centralized performance $2H_\theta(Z)$ as obtained using our algorithm can be less than Slepian-Wolf sum rate $H_\theta(X,Y)$. As an example, consider a non-symmetric nonlinear binary source with the PMF shown in Table 2.2.

X/Y	0	1
0	$1/4+\theta^2$	$1/4-\theta^2$
1	$1/4-\theta$	$1/4+\theta$

Table 2.2: An example of a joint PMF of a non-symmetric binary source with $\theta \in \Theta = (0, 1/4)$.

Although the joint PMF given in Table 2.2 is not symmetric and nonlinear in θ, the required rates to obtain an unbiased estimator that achieves the centralized performance are still lower than Slepian-Wolf rates as shown in Fig. 2.3.

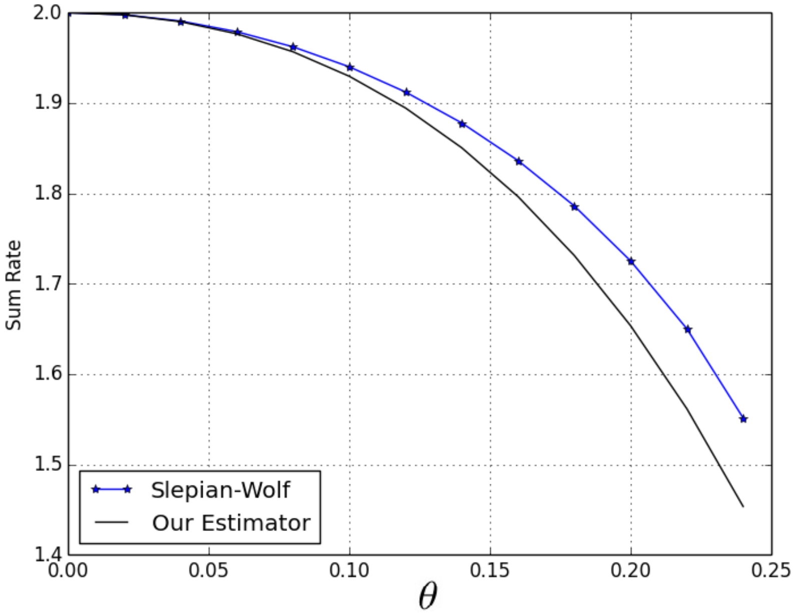

Figure 2.3: The required rates to achieve the optimal centralized performance for the binary source given in Table 2.2 is lower than Slepian-Wolf rates.

2.4 Non-Binary Models

In this section, we extend our results for binary models to more general class of non-binary models. Let $\mathcal{X} = \mathcal{Y} = \{0, 1, ..., M-1\}$ and consider the class of PMFs

$$P_\theta(X = i, Y = j) = \begin{cases} \frac{\theta}{M} \text{ , if } (i+j) \neq M-1 \\ \frac{1-\theta(M-1)}{M} \text{ , otherwise,} \end{cases} \tag{2.19}$$

where $\theta \in \Theta = (0, \frac{1}{(M-1)})$. Note that each information source has a uniform marginal PMF and setting $M = 2$ recovers the binary case.

Similar to the binary case, we first use a linear code and minimum entropy decoder to reconstruct $Z^n = (X^n + Y^n) \mod M$ at the decoder and then design an estimator from Z^n. In this section, we use $\mod M$ to denote element-wise mod operation,

In particular, we use a linear code f that maps $\{0, 1, ..., M-1\}^n$ to $\{0, 1, ..., M-1\}^k$. The encoded messages of the realizations $x^n \in \{0, 1, ..., M-1\}^n$ and $y^n \in \{0, 1, ..., M-1\}^n$ are

$$\begin{aligned} f(x^n) &= x^n A, \\ f(y^n) &= y^n A, \end{aligned} \tag{2.20}$$

in which the code matrix A has n rows and k columns with each entry taking values from $\{0, 1, ..., M-1\}$. The coding rate is

$$R = \frac{k}{n} \log M. \tag{2.21}$$

The decoder first combines the encoded messages into a single message as

$$f(x^n) + f(y^n) \mod M. \tag{2.22}$$

The final decoded message is given by

$$\hat{z}^n = \phi(f(z^n)), \tag{2.23}$$

where ϕ the the minimum entropy decoding function. Following the same error probability analysis for the binary case, we can show that there exists a codebook f^* (and hence a particular encoding matrix A) that achieves a probability of decoding error $P_{e,f^*}^{(n)} \to 0$ exponentially as $n \to \infty$ if

$$R \geq H_\theta(Z) = H_\theta(X|Y) = H_\theta(Y|X). \tag{2.24}$$

Therefore, as long as

$$R_X > H_\theta(X|Y), \tag{2.25}$$

$$R_Y > H_\theta(Y|X), \tag{2.26}$$

we can reconstruct $Z^n = X^n + Y^n \mod M$ at the decoder with an exponentially diminishing error probability.

After obtaining $\hat{Z}^n$, which is equal to Z^n with a probability converging to 1 exponentially, our estimator is

$$\hat{\theta} = \frac{n - n(M-1|\hat{Z}^n)}{n(M-1)}. \tag{2.27}$$

Following similar steps as those in the binary case, we can show that, if (2.25)-(2.26) are satisfied, the estimator in (2.27) is asymptotically unbiased and achieves a variance index

$$V_\theta[\hat{\theta}] = \frac{\theta[1-\theta(M-1)]}{M-1}. \tag{2.28}$$

We can further show that (2.28) is the best variance index that can be achieved even in the centralized case. This implies that our algorithm achieves the centralized performance using rates outside the Slepian-Wolf region.

2.5 Multiple Source Networks

In this section we extend our results to the case of multiple source networks. We consider a network that consists of N binary information sources $(X_1, X_2, ..., X_N)$. The observations $(X_1^n, X_2^n, ..., X_N^n)$ are (i.i.d.) and drawn according to the parametric joint PMF $P_\theta(X_1, X_2, ..., X_N)$ given by

$$P_\theta(x_1, ..., x_N) = \begin{cases} \frac{\theta}{2^{N-1}}, \text{ if}(x_1 \oplus ... \oplus x_N) = 0 \\ \frac{1-\theta}{2^{N-1}}, \text{ otherwise}, \end{cases} \tag{2.29}$$

where $\theta \in (0, 1)$. To establish our estimator, we first use a linear code f that has a rate R, and a minimum entropy decoder ϕ to reconstruct $Z^n = (X_1^n \oplus, ..., \oplus X_N^n)$. The encoded messages for the realizations $(x_1^n, x_2^n, ..., x_N^n)$ are given by

$$\begin{aligned} f(x_1^n) &= x_1^n A, \\ f(x_2^n) &= x_2^n A, \\ &. \\ &. \\ f(x_N^n) &= x_N^n A, \end{aligned} \tag{2.30}$$

where A is a binary matrix that has n rows and nR columns. At the fusion center, the decoder first combines the encoded messages into a single message $f(x_1^n)\oplus, ..., \oplus f(x_N^n)$, then reconstructs $\hat{z}^n$ as following

$$\hat{z}^n = \phi(f(z^n)). \tag{2.31}$$

Similar to the error probability analysis for the two source case, we can show that there exists a codebook f^*, such that $\hat{z}^n$ can be reconstructed efficiently with a probability of decoding error $P_{e,f^*}^{(n)} \to 0$ as $n \to \infty$ if

$$\begin{aligned} R &> H_\theta(Z) \\ &= H_\theta(X_1|X_2, ..., X_N) \\ &. \\ &. \\ &= H_\theta(X_N|X_1, ..., X_{N-1}). \end{aligned} \tag{2.32}$$

Since there is no additional constraints on the sum rates, then Z^n can be reconstructed efficiently at rates less than Slepian-Wolf rates. After obtaining $\hat{Z}^n$, we construct our estimator as following

$$\hat{\theta} = \frac{n(0|\hat{Z}^n)}{n}. \tag{2.33}$$

Following similar steps to the two source case, we can show that our estimator is asymptotically unbiased and achieves the minimum variance index given by

$$V_\theta[\hat{\theta}] = \theta(1-\theta). \tag{2.34}$$

Hence, our estimator is an asymptotically MVUE using rates outside the Slepian-Wolf region.

2.6 Practical Approach

In the previous sections, we established an unbiased estimator that achieves the centralized performance for a number of information sources, while requires less rates than Slepian-Wolf rates. For binary symmetric sources and its extension, our estimator achieves the CRLB within the combined regions of Slepian-Wolf and the dotted region as shown in Fig. 2.4, where

$$\begin{aligned} R_X &> H_\theta(X|Y), \\ R_Y &> H_\theta(Y|X). \end{aligned} \tag{2.35}$$

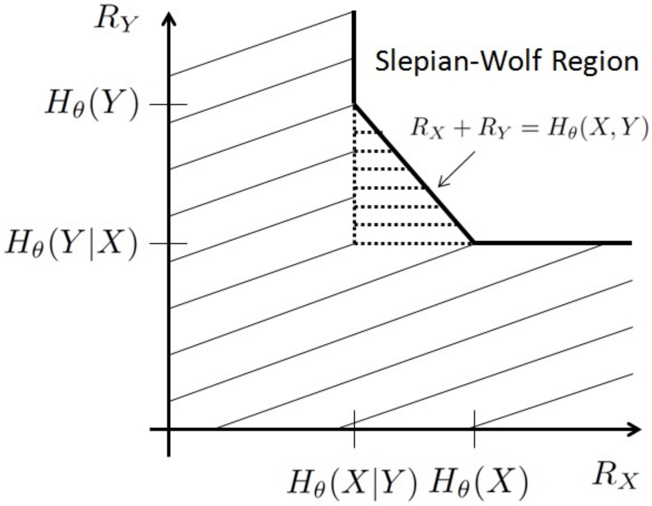

Figure 2.4: The low rates inside the dashed region are considered in this section.

Our estimator is optimal if $Z^n = X^n \oplus Y^n$ is decoded with a vanishing probability of error. Otherwise, there is no optimality guarantee. In practical applications, the communications rates can be lower than our conditions (2.35). Therefore, we modify the design of our estimation algorithm in this section to ensure a good performance at all rates including the low rates inside the dashed region as shown in Fig. 2.4. We start with the case of binary symmetric sources then we extend the results to the general class of PMFs as presented in Section 2.4. For binary symmetric sources, we assume that the unknown parameter θ takes values in $(0, t)$, where $t \in (0, 0.5)$ is known.

First, we apply the encoding/decoding scheme introduced in Section 2.2 to encode p observations (x^p, y^p) and decode $\hat{z}^p = x^p \oplus y^p$, where

$$p = \begin{cases} n, \text{ if } R \geq H(t) \\ \lfloor \frac{nR}{H(t)} \rfloor, \text{ otherwise,} \end{cases} \tag{2.36}$$

where $\lfloor \cdot \rfloor$ is an operator that maps its argument to the largest previous integer, and $H(t) = -t \log t - (1-t) \log(1-t)$. Then, we modify our estimator as following:

$$\hat{\theta} = \frac{n(0|\hat{Z}^p)}{p}. \tag{2.37}$$

The following Theorem states the performance bounds of our estimator.

Theorem 4. *If*

$$R \geq H(t), \tag{2.38}$$

our estimator is an asymptotically MVUE. Otherwise, our estimator is asymptotically unbiased and its variance index is bounded as

$$V_\theta[\hat{\theta}] \leq \frac{H(t)\theta(1-\theta)}{R}. \tag{2.39}$$

Proof. The proof of this theorem can be found in Appendix D. □

For the general class of PMFs given in (2.19), we assume that θ takes values in $(0, t)$, and $t \in (0, \frac{1}{2(M-1)})$. We establish our estimator as

$$\hat{\theta} = \frac{p - n(M-1|\hat{Z}^p)}{p(M-1)}. \tag{2.40}$$

Following similar steps to the proof of Theorem 4, we have that our estimator is an asymptotically MVUE if $R \geq H(t)$. Otherwise, our estimator is asymptotically unbiased and its variance index is bounded as

$$V_\theta[\hat{\theta}] \leq \frac{H(t)\theta[1-\theta(M-1)]}{R(M-1)}. \tag{2.41}$$

For binary symmetric sources and its extension, we guarantee a worst case performance that is a function of the communication rate R. In the following section, we show that despite of a small performance degradation in the rate region $H(\theta) \leq R < H(t)$ as compared to our estimator in Section 2.4, we managed to achieve a very good performance at low rates.

2.7 Numerical Results

In this section, we use several numerical examples to illustrate the comparison between our estimators to the best known estimator by Han and Amari [36]. In the simulation, we fix the unknown parameter θ and change the encoding rates R_X and R_Y such that

$$R_X = R_Y = R. \tag{2.42}$$

We conduct the comparison for $M = 2$ and $M = 4$, respectively.

For our estimator in Section 2.4 and $M = 2$, the variance index of our estimator is (B.16), while the variance index of the estimator by Han and Amari is calculated in example 3 of [36]

$$(V_\theta[\hat{\theta}])_{HA} \simeq \frac{1}{16a^2b^2}\left\{\frac{1}{4} - \left(\theta - \frac{1}{2}\right)^2 [1 - (1 - 4a^2)(1 - 4b^2)]\right\}, \tag{2.43}$$

where a and b are functions of R_X and R_Y, whose expressions are given in (14.12) and (14.13) of [36], respectively.

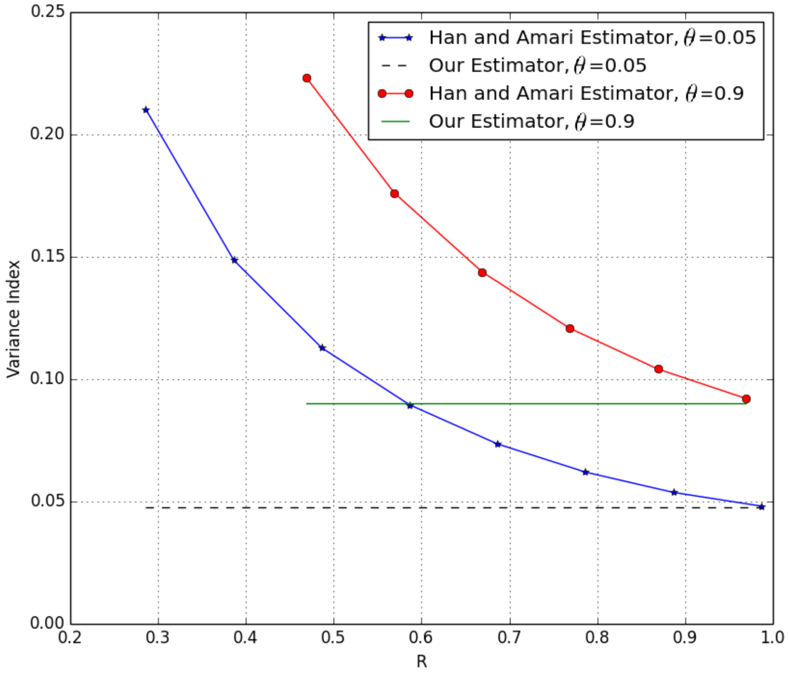

Figure 2.5: Performance Comparison: $M = 2$

Fig. 2.5 shows the performance gain, in terms of the variance index, of our estimator over Han and Amari's estimator for binary symmetric sources ($M = 2$) at two different values of the unknown parameter, $\theta = 0.05$ and $\theta = 0.9$, respectively. The performance difference is more noticeable at low rates. For $\theta = 0.05$, the Slepian-Wolf sum rate is

$R_X + R_Y = 1.29$ bits, while our estimator requires a sum rate of $R_X + R_Y = 2R = 0.57$ bits. For $\theta = 0.9$, the Slepian-Wolf sum rate is 1.47 bits, while our estimator requires a sum rate of 0.94 bits. Furthermore, for Han and Amari's estimator to achieve the centralized performance, the required sum-rate is 2 bits for both cases, which is not only much larger than the sum rate required in our estimator but also much larger than the sum-rate required by conditions specified in the Slepian-Wolf rate region.

For our estimator in Section 2.4 and $M = 4$, the variance index of our estimator is given in (2.28). The performance of Han and Amari's estimator relies on the choice of the test channels. The authors did not specify an optimal choice of the test channels in order to extend example 3 in [36] to the case of $M = 4$. We find the following mapping to be a natural extension:

$$Q = \begin{cases} 0, \text{if } \ X \in \{0, 1\} \\ 1, \text{if } \ X \in \{2, 3\}, \end{cases} \qquad T = \begin{cases} 0, \text{if } \ Y \in \{0, 1\} \\ 1, \text{if } \ Y \in \{2, 3\}. \end{cases} \tag{2.44}$$

Notice that (Q, T) are distributed according to a binary symmetric PMF with an unknown parameter $\alpha = 2\theta$. Using an estimator $\hat{\theta} = \frac{\hat{\alpha}}{2}$ leads to the following expression for the variance index:

$$(V_\theta[\hat{\theta}])_{HA} \simeq \frac{1}{64a^2b^2}\left\{\frac{1}{4} - \left(2\theta - \frac{1}{2}\right)^2 [1 - (1 - 4a^2)(1 - 4b^2)]\right\}. \tag{2.45}$$

Fig. 2.6 compares the variance indices achieved using our estimator and Han and Amari's estimator for $M = 4$ and $\theta = 0.01$. It is clear that our estimator outperforms that of Han and Amari's estimator. Furthermore, the performance difference is more noticeable at low rates. The Slepian-Wolf sum rate is 2.24 bits, while our estimator requires a sum rate of 0.48 bits.

For our practical estimator in Section 2.6 and $M = 2$, the variance index of our

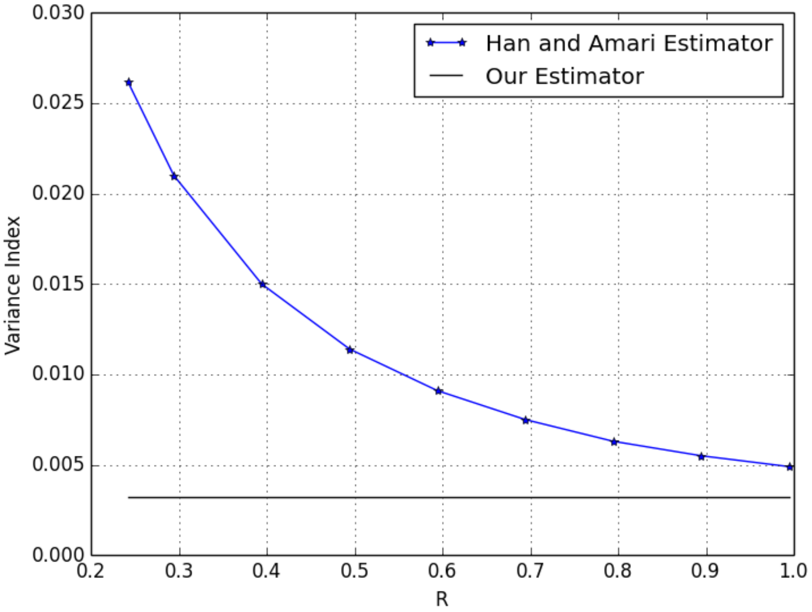

Figure 2.6: Performance Comparison: $\theta = 0.01$, $M = 4$

estimator is bounded as in (2.41) if $R < H(t)$. Otherwise, it achieves the CRLB. The variance index of Han and Amari's estimator is (2.43).

For our practical estimator in Section 2.6 and $M = 4$, the variance index of our estimator is bounded as in (D.12) if $R < H(t)$. Otherwise, it achieves the CRLB. The variance index of Han and Amari's estimator is (2.45).

Fig. 2.7 and Fig. 2.8 show that our estimator outperforms Han and Amari's estimator at all rates. The performance difference is more noticeable at very low rates, which makes our estimator a good choice for applications with low rate constraints. Our estimator performs better for smaller values of the range of θ, which is determined by t.

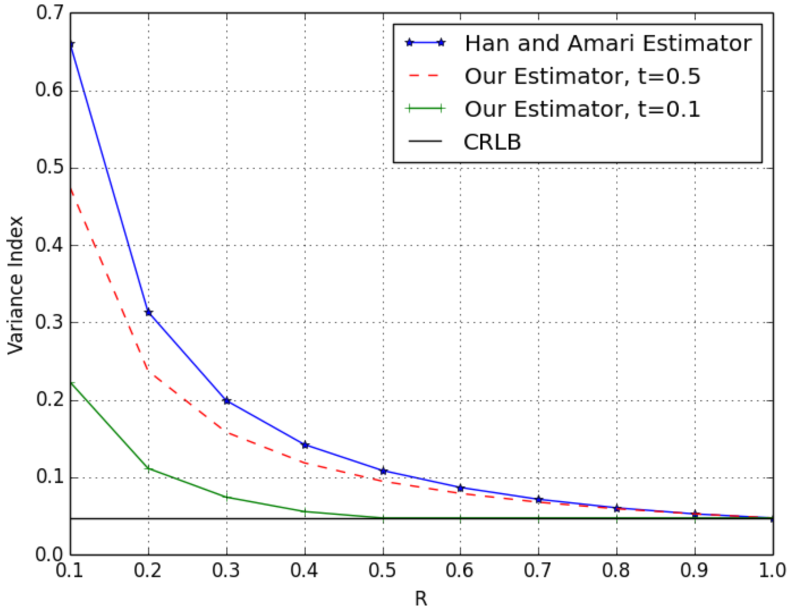

Figure 2.7: Performance Comparison: $\theta = 0.05$, $M = 2$, $t = 0.5$ and 0.1

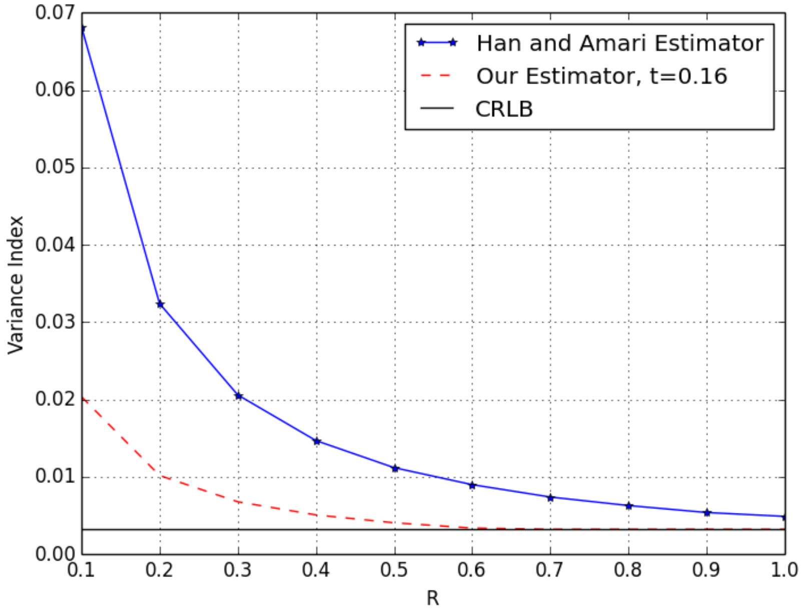

Figure 2.8: Performance Comparison: $\theta = 0.01$, $M = 4$, $t = 0.16$

Chapter 3

Sufficiency Based Data Reduction

In this chapter, we examine the optimality of data dimensionality reduction via sufficient statistics compression in distributed parameter estimation problems for the scenario where the communication rates are not sufficient to achieve the same inference performance as that of the centralized case as characterized in Chapter 2. The chapter is organized as follows. We give a formal statement of the problem in Section 3.1. In Section 3.2 we prove that sufficiency based data reduction is optimal, then we extend our result to the multiple sources scenario and the discrete case in Section 3.3. In Section 3.4 we analyze the asymptotic optimal Bayesian performance in the presence of communication constraints. We verify our results through simulations in Section 3.5.

3.1 Problem Formulation

We study the problem of distributed parameter estimation when the n observations X_1^n related to the random variable $X \in \mathcal{X}$ are distributed between two nodes, such that node 1 has access to the observations $X_1^{n_1}$ and node 2 has access to $X_{n_1+1}^n$. The observations are independent and identically distributed (i.i.d.) and generated according to the parametric probability distribution function $f(X|\theta)$, where $\theta \in \Theta$ is an unknown parameter that follows a probability distribution function $f(\theta)$. The question we answer is whether sufficiency based data reduction is optimal or not.

To answer this question, we compare the performance of the two system models (a) and (b) as shown in Fig. 3.1, while applying the same compression rate pair (R_1, R_2) to both models. In system model (a), node 1 and node 2 compress the observations $X_1^{n_1}$ and $X_{n_1+1}^{n}$ using the encoding functions $g_1(\cdot) \in \mathcal{G}_1$ and $g_2(\cdot) \in \mathcal{G}_2$, respectively. The fusion center receives the compressed messages and applies the decoding function ϕ_a to get an estimate $\hat{\theta}_a$ of the unknown parameter θ. In (b), an additional step is added to extract the sufficient statistics $T_1(X_1^{n_1})$ and $T_2(X_{n_1+1}^{n})$ from the observations, then compress them using the encoding functions $h_1(\cdot) \in \mathcal{H}_1$ and $h_2(\cdot) \in \mathcal{H}_2$, respectively. The fusion center uses the decoding function ϕ_b to get an estimate $\hat{\theta}_b$ of the unknown parameter.

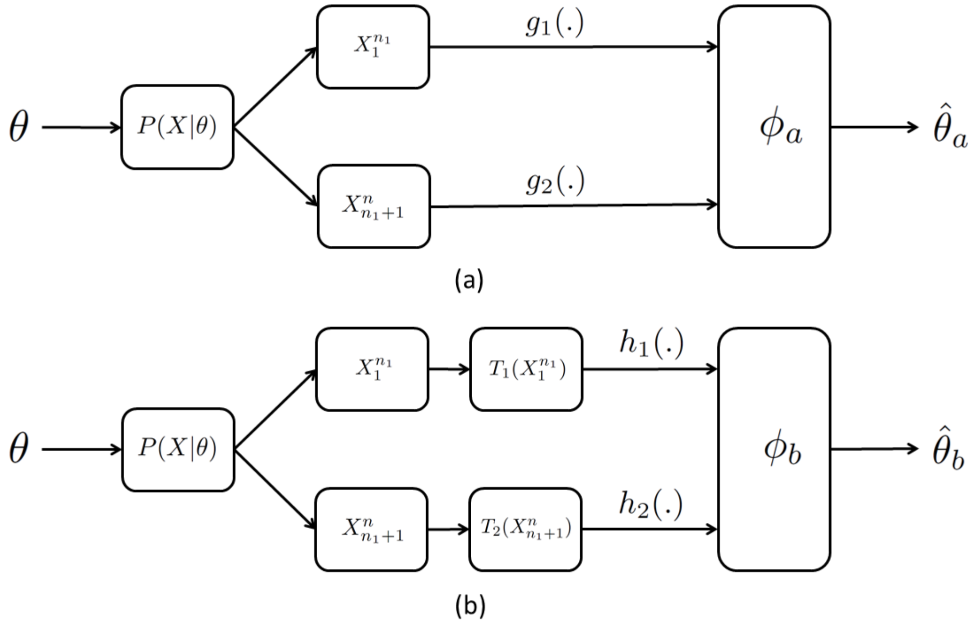

Figure 3.1: System models (a) and (b).

The encoding rates are given by

$$R_1 = \frac{1}{n_1} \log |\mathcal{G}_1| = \frac{1}{n_1} \log |\mathcal{H}_1|, \tag{3.1}$$

$$R_2 = \frac{1}{(n-n_1)} \log |\mathcal{G}_2| = \frac{1}{(n-n_1)} \log |\mathcal{H}_2|. \tag{3.2}$$

According to the definition, a sufficient statistic $T(Y)$ carries as much information about the unknown parameter θ as the observation Y. Hence, we have the following Markov chains

$$\begin{aligned} \theta &- T_1(X_1^{n_1}) - X_1^{n_1}, \\ \theta &- T_2(X_{n_1+1}^{n}) - X_{n_1+1}^{n}. \end{aligned} \tag{3.3}$$

The performance metric we use is the Bayesian risk, which is expressed as

$$B_a = \inf_{g_1, g_2, \phi_a} \mathbb{E}_{\theta, X}[(\theta - \hat{\theta}_a)^2], \tag{3.4}$$

$$B_b = \inf_{h_1, h_2, \phi_b} \mathbb{E}_{\theta, X}[(\theta - \hat{\theta}_b)^2]. \tag{3.5}$$

In the following section, we answer the main question of the chapter for the Bayesian risk by analyzing the relationship between the pair B_a and B_b. An easy observation about this relationship is that $B_a \leq B_b$, which can be shown by choosing $g_1(x_1^{n_1}) = h_1(T_1(x_1^{n_1}))$ and $g_2(x_{n_1+1}^{n}) = h_2(T_2(x_{n_1+1}^{n}))$.

Throughout the chapter, we use an upper case letter U to denote a random variable, a lower case letter u to denote a realization of U, and $\mathcal{U}$ to denote the set from which U takes values.

3.2 Data Reduction

Theorem 5. *The optimal Bayesian performance can be achieved by compressing the sufficient statistics extracted from the observations. Hence*

$$B_b = B_a.$$

Proof. Let (g_1^*, g_2^*, ϕ_a^*) be the encoding/decoding functions used to obtain B_a. Therefore

$$\begin{aligned} B_a &= \int_\Theta \int_{\mathcal{X}^n} (\theta - \phi_a^*(g_1^*(x_1^{n_1}), g_2^*(x_{n_1+1}^n)))^2 f(x^n|\theta) f(\theta) dx^n d\theta \\ &= \int_\Theta \int_{\mathcal{X}_1^{n_1}} \int_{\mathcal{X}_{n_1+1}^n} (\theta - \phi_a^*(g_1^*(x_1^{n_1}), g_2^*(x_{n_1+1}^n)))^2 f(x_1^{n_1}, x_{n_1+1}^n|\theta) \\ &\times f(\theta) dx_1^{n_1} dx_{n_1+1}^n d\theta \\ &= \int_\Theta \int_{\mathcal{X}_1^{n_1}} \int_{\mathcal{X}_{n_1+1}^n} (\theta - \phi_a^*(g_1^*(x_1^{n_1}), g_2^*(x_{n_1+1}^n)))^2 f(x_1^{n_1}|\theta) f(x_{n_1+1}^n|\theta) \\ &\times f(\theta) dx_1^{n_1} dx_{n_1+1}^n d\theta, \end{aligned} \tag{3.6}$$

where the last equality follows from the fact that the observations are (i.i.d.). We have that

$$\begin{aligned} f(x_1^{n_1}|\theta) f(\theta) &= f(x_1^{n_1}) f(\theta|x_1^{n_1}) \\ &= f(x_1^{n_1}) f(\theta|T_1(x_1^{n_1})), \end{aligned} \tag{3.7}$$

where the second equality follows from (3.3) as explained in Lemma 1 of [1]. By defining $\alpha_1(g_1^*)$ as

$$\begin{aligned} \alpha_1(g_1^*) &= \int_\Theta \int_{\mathcal{X}_{n_1+1}^n} (\theta - \phi_a^*(g_1^*(x_1^{n_1}), g_2^*(x_{n_1+1}^n)))^2 \\ &\times f(x_{n_1+1}^n|\theta) f(\theta|T_1(x_1^{n_1})) dx_{n_1+1}^n d\theta, \end{aligned} \tag{3.8}$$

the Bayesian risk can be rewritten as

$$B_a = \int_{\mathcal{X}_1^{n_1}} \alpha_1(g_1^*) f(x_1^{n_1}) dx_1^{n_1}. \tag{3.9}$$

Given $x_1^{n_1} \in \mathcal{X}_1^{n_1}$, the encoded message $g_1^*(x_1^{n_1}) = s \in \mathcal{G}_1$ is chosen to minimize α_1. Hence, for any $t \in \mathcal{G}_1$, we have that

$$\begin{aligned} 0 &\geq \alpha_1(s) - \alpha_1(t) \\ &= \int_\Theta \int_{\mathcal{X}_{n_1+1}^n} [(\theta - \phi_a^*(s, g_2^*(x_{n_1+1}^n)))^2 - (\theta - \phi_a^*(t, g_2^*(x_{n_1+1}^n)))^2] \\ &\times \ f(x_{n_1+1}^n|\theta) f(\theta|T_1(x_1^{n_1})) dx_{n_1+1}^n d\theta. \end{aligned} \tag{3.10}$$

Condition (E.6) depends on $x_1^{n_1}$ only through $T_1(x_1^{n_1})$. Therefore s can be chosen optimally using information about $T_1(x_1^{n_1})$, which implies that there exists an encoding function $h_1^*(\cdot) \in \mathcal{H}_1$, such that $h_1^*(T_1(x_1^{n_1})) = g_1^*(x_1^{n_1})$ and $\mathcal{H}_1 = \mathcal{G}_1$. Similarly, one can define $\alpha_2(g_2^*)$ and show that there exists an encoding function $h_2^*(\cdot) \in \mathcal{H}_2$, such that $h_2^*(T_2(x_{n_1+1}^n)) = g_2^*(x_{n_1+1}^n)$ and $\mathcal{H}_2 = \mathcal{G}_2$.

Using the encoding/decoding functions $(h_1^*, h_2^*, \phi_b^* = \phi_a^*)$ together with the fact that $B_a \leq B_b$, we get that

$$B_b = B_a. \tag{3.11}$$

And thus the optimal Bayesian performance can be achieved by compressing the sufficient statistics extracted from the observations. □

This shows that it is optimal to reduce the dimensionality of the observations before compressing them, which greatly simplifies the design of the compression scheme.

3.3 Extensions

3.3.1 Multiple Sources

Our result can be extended to the multiple sources scenario, where the n observations X_1^n are distributed over M nodes. Node $(i+1)$ has access to the observations $X_{n_i+1}^{n(i+1)}$, $i = 0, 1, ..., M-1$, where $n_0 = 0$ and $n_m = n$. In this case, the Bayesian risk can be expressed as

$$\begin{aligned} B_a &= \int_\Theta \int_{\mathcal{X}^n} (\theta - \phi_a^*(g_1^*(x_1^{n_1}), ..., g_m^*(x_{n_{m-1}+1}^n)))^2 f(x^n|\theta) f(\theta) dx^n d\theta \\ &= \int_\Theta \int_{\mathcal{X}_1^{n_1}} ... \int_{\mathcal{X}_{n_{m-1}+1}^n} (\theta - \phi_a^*(g_1^*(x_1^{n_1}), ..., g_m^*(x_{n_{m-1}+1}^n)))^2 f(x_1^{n_1}, ..., x_{n_{m-1}+1}^n|\theta) \\ &\times \ f(\theta) dx_1^{n_1} \times ... \times dx_{n_{m-1}+1}^n d\theta \\ &= \int_\Theta \int_{\mathcal{X}_1^{n_1}} ... \int_{\mathcal{X}_{n_{m-1}+1}^n} (\theta - \phi_a^*(g_1^*(x_1^{n_1}), ..., g_m^*(x_{n_{m-1}+1}^n)))^2 f(x_1^{n_1}|\theta) \times ... \times f(x_{n_{m-1}+1}^n|\theta) \\ &\times \ f(\theta) dx_1^{n_1} \times ... \times dx_{n_{m-1}+1}^n d\theta. \end{aligned} \tag{3.12}$$

By defining $\alpha_1(g_1^*)$ as

$$\begin{aligned} \alpha_1(g_1^*) &= \int_\Theta \int_{\mathcal{X}_{n_1+1}^{n_2}} ... \int_{\mathcal{X}_{n_{m-1}+1}^n} (\theta - \phi_a^*(g_1^*(x_1^{n_1}), ..., g_m^*(x_{n_{m-1}+1}^n)))^2 f(x_{n_1+1}^{n_2}|\theta) \times ... \times f(x_{n_{m-1}+1}^n|\theta) \\ &\times \ f(\theta|T_1(x_1^{n_1})) dx_{n_1+1}^{n_2} \times ... \times dx_{n_{m-1}+1}^n d\theta, \end{aligned} \tag{3.13}$$

and following similar steps to the two sources scenario, it can be shown that $B_a = B_b$.

3.3.2 Discrete Case

In the discrete case, the n observations X_1^n are distributed over M nodes and generated according to the parametric probability mass function $P(X|\theta)$, and $\theta \in \Theta$ follows a probability distribution function $f(\theta)$. The Bayesian risk can be expressed as

$$\begin{aligned}
B_a &= \int_{\Theta} \sum_{\mathcal{X}^n} (\theta - \phi_a^*(g_1^*(x_1^{n_1}), ..., g_m^*(x_{n_{m-1}+1}^n)))^2 P(x^n|\theta) f(\theta) d\theta \\
&= \int_{\Theta} \sum_{\mathcal{X}_1^{n_1}} ... \sum_{\mathcal{X}_{n_{m-1}+1}^n} (\theta - \phi_a^*(g_1^*(x_1^{n_1}), ..., g_m^*(x_{n_{m-1}+1}^n)))^2 P(x_1^{n_1}, ..., x_{n_{m-1}+1}^n|\theta) f(\theta) d\theta \\
&= \int_{\Theta} \sum_{\mathcal{X}_1^{n_1}} ... \sum_{\mathcal{X}_{n_{m-1}+1}^n} (\theta - \phi_a^*(g_1^*(x_1^{n_1}), ..., g_m^*(x_{n_{m-1}+1}^n)))^2 P(x_1^{n_1}|\theta) \times ... \times P(x_{n_{m-1}+1}^n|\theta) \\
&\times\ f(\theta) d\theta. \qquad (3.14)
\end{aligned}$$

We have that

$$\begin{aligned}
P(x_1^{n_1}|\theta) f(\theta) &= P(x_1^{n_1}) f(\theta|x_1^{n_1}) \\
&= P(x_1^{n_1}) f(\theta|T_1(x_1^{n_1})). \qquad (3.15)
\end{aligned}$$

By defining $\alpha_1(g_1^*)$ as

$$\begin{aligned}
\alpha_1(g_1^*) &= \int_{\Theta} \sum_{\mathcal{X}_{n_1+1}^{n_2}} ... \sum_{\mathcal{X}_{n_{m-1}+1}^n} (\theta - \phi_a^*(g_1^*(x_1^{n_1}), ..., g_m^*(x_{n_{m-1}+1}^n)))^2 P(x_{n_1+1}^{n_2}|\theta) \times ... \times P(x_{n_{m-1}+1}^n|\theta) \\
&\times\ f(\theta|T_1(x_1^{n_1})) d\theta, \qquad (3.16)
\end{aligned}$$

and following similar steps to the two sources scenario, it can be shown that $B_a = B_b$.

3.4 Optimal Bayesian Performance

In this section, we study the optimal Bayesian performance in the presence of communication constraints.

Theorem 6. *Let the observations X_1^n be generated according to a Gaussian distribution $X \sim \mathcal{N}(\theta, 1)$, where $\theta \in \mathbb{R}$ follows a Gaussian distribution $\theta \sim \mathcal{N}(0, 1)$. The n observa-*

tions are distributed evenly between two nodes such that $n_1 = n_2 = n/2$. The asymptotic optimal Bayesian (AOB) performance under 1-bit compression constraint is given by

$$B_{AOB-1bit} = 1 - \frac{2}{\pi}.$$

Proof. As shown in Theorem 5, it is optimal that nodes 1 and 2 compress the extracted sufficient statistics $T_1(X_1^{n/2})$ and $T_2(X_{(n/2)+1}^{n})$, respectively, then send them to the fusion center. The fusion center can then establish an optimal Bayesian estimator $\hat{\theta}$ as a function of the received messages. In this case, the sufficient statistics can be calculated as following

$$\begin{aligned} T_1(x_1^{n/2}) &= \frac{2\sum_{i=1}^{n/2} x_i}{n}, \\ T_2(x_{(n/2)+1}^{n}) &= \frac{2\sum_{i=(n/2)+1}^{n} x_i}{n}. \end{aligned} \tag{3.17}$$

Hence $T_1(X_1^{n/2}) \sim \mathcal{N}(\theta, 2/n)$, $T_2(X_{(n/2)+1}^{n}) \sim \mathcal{N}(\theta, 2/n)$, and

$$\begin{aligned} \mathrm{Pr}\big(T_1(X_1^{n/2}) \geq 0\big)|\theta\big) &= 1 - \mathrm{Pr}\big(T_1(X_1^{n/2}) < 0\big)|\theta\big) \\ &= \int_0^{\infty} f(T_1(x_1^{n/2})|\theta) dT_1(x_1^{n/2}) = Q(-\sqrt{\frac{n}{2}}\theta), \\ \mathrm{Pr}\big(T_2(X_{(n/2)+1}^{n}) \geq 0\big)|\theta\big) &= 1 - \mathrm{Pr}\big(T_2(X_{(n/2)+1}^{n}) < 0\big)|\theta\big) \\ &= \int_{-\infty}^{0} f(T_2(x_{(n/2)+1}^{n})|\theta) dT_2(x_{(n/2)+1}^{n}) \\ &= Q(-\sqrt{\frac{n}{2}}\theta), \end{aligned} \tag{3.18}$$

where $Q(\cdot)$ is the Q-function. Due to the symmetry of the problem since both nodes have access to equal number of observations, the encoders at nodes 1 and 2 apply the same compression scheme. A 1-bit compression scheme $h(\cdot)$ maps the value of the sufficient statistic $T(X)$ to one of two levels h_a or h_b with $h_a \geq h_b$. These levels are chosen to minimize the quantization error as given by

$$\begin{aligned}\mathbb{E}[(h(T(X)) - T(X))^2] &= \int_{-\infty}^{\infty} (h(T(x)) - T(x))^2 dT(x) \\ &= \int_{\mathcal{T}_a} (h_a - T(x))^2 dT(x) \\ &+ \int_{\mathcal{T}_b} (h_b - T(x))^2 dT(x), \qquad (3.19)\end{aligned}$$

where $\mathcal{T}_a$ is the range of values of $T(X)$ mapped to h_a, and $\mathcal{T}_b$ is the range of values of $T(X)$ mapped to h_b. To minimize the quantization error, a point $T(x)$ is mapped to its nearest quantization level. Therefore, each of the quantization ranges $\mathcal{T}_a$ and $\mathcal{T}_b$ is continuous, and hence the encoding function is given by

$$h(T(x)) = \begin{cases} h_a, \text{ if } T(x) \geq \frac{h_a - h_b}{2} \\ h_b, \text{ if } T(x) < \frac{h_a - h_b}{2} \end{cases} . \qquad (3.20)$$

Notice that the decoder only needs to know whether the value of $T(x)$ is larger or smaller than the dividing point $\frac{h_a - h_b}{2}$.

At node 1, we have that

$$\begin{aligned} f\big(T_1(x_1^{n/2})\big) &= \int_{-\infty}^{\infty} f\big(T_1(x_1^{n/2})\big)|\theta\big) f(\theta) d\theta \\ &= f\big(-T_1(x_1^{n/2})\big). \qquad (3.21)\end{aligned}$$

Therefore, $f(T_1(X_1^{n/2}))$ is symmetric around 0, and hence the optimal dividing point for a 1-bit compression scheme is 0. The encoder at node 1 chooses the encoding message as following

$$h_1(T_1(x_1^{n/2})) = \begin{cases} 1, \text{ if } T_1(x_1^{n/2}) \geq 0 \\ 0, \text{ if } T_1(x_1^{n/2}) < 0 \end{cases} . \tag{3.22}$$

Similarly, the encoder at node 2 chooses the encoding message as following

$$h_2(T_2(x_{(n/2)+1}^n)) = \begin{cases} 1, \text{ if } T_2(x_{(n/2)+1}^n) \geq 0 \\ 0, \text{ if } T_2(x_{(n/2)+1}^n) < 0 \end{cases} . \tag{3.23}$$

At the fusion center, we have that

$$\begin{aligned} f\big(\theta|(T_1(x_1^{n/2}) \geq 0, T_2(x_{(n/2)+1}^n) \geq 0)\big) &= \frac{\mathbf{Pr}\big(T_1(X_1^{n/2}) \geq 0|\theta\big)\mathbf{Pr}\big(T_2(x_{(n/2)+1}^n) \geq 0|\theta\big) f(\theta)}{\mathbf{Pr}(T_1(X_1^{n/2}) \geq 0, T_2(X_{(n/2)+1}^n) \geq 0)} \\ &= \frac{[Q(-\sqrt{\frac{n}{2}}\theta)]^2 f(\theta)}{\int_{-\infty}^{\infty}[Q(-\sqrt{\frac{n}{2}}\theta)]^2 f(\theta) d\theta}. \end{aligned} \tag{3.24}$$

Therefore,

$$\lim_{n\to\infty} f\big(\theta|(T_1(x_1^{n/2}) \geq 0, T_2(x_{(n/2)+1}^n) \geq 0)\big) = 2f(\theta) \quad , \theta \geq 0, \tag{3.25}$$

and

$$\lim_{n\to\infty} \mathbb{E}\big(\theta|(T_1(x_1^{n/2}) \geq 0, T_2(x_{(n/2)+1}^n) \geq 0)\big) = \sqrt{\frac{2}{\pi}}. \tag{3.26}$$

Similarly,

$$\lim_{n\to\infty} \mathbb{E}\big(\theta|(T_1(x_1^{n/2}) < 0, T_2(x_{(n/2)+1}^n) < 0)\big) = -\sqrt{\frac{2}{\pi}}. \tag{3.27}$$

We also have that

$$
\begin{aligned}
f\big(\theta|(T_1(x_1^{n/2}) \geq 0, T_2(x^n_{(n/2)+1}) < 0)\big) &= \frac{\text{Pr}\big(T_1(X_1^{n/2}) \geq 0|\theta\big)\text{Pr}\big(T_2(x^n_{(n/2)+1}) < 0|\theta\big) f(\theta)}{\text{Pr}(T_1(X_1^{n/2}) \geq 0, T_2(X^n_{(n/2)+1}) < 0)} \\
&= \frac{Q(-\sqrt{\frac{n}{2}}\theta)Q(\sqrt{\frac{n}{2}}\theta)f(\theta)}{\int_{-\infty}^{\infty} Q(-\sqrt{\frac{n}{2}}\theta)Q(\sqrt{\frac{n}{2}}\theta)f(\theta)d\theta}. \qquad (3.28)
\end{aligned}
$$

This shows that $f\big(\theta|(T_1(x_1^{n/2}) \geq 0, T_2(x^n_{(n/2)+1}) < 0)\big)$ is symmetric around 0, and hence

$$\mathbb{E}\big(\theta|(T_1(x_1^{n/2}) \geq 0, T_2(x^n_{(n/2)+1}) < 0)\big) = 0. \qquad (3.29)$$

Similarly,

$$\mathbb{E}\big(\theta|(T_1(x_1^{n/2}) < 0, T_2(x^n_{(n/2)+1}) \geq 0)\big) = 0. \qquad (3.30)$$

Therefore, the optimal estimation scheme is given in Fig. 3.2

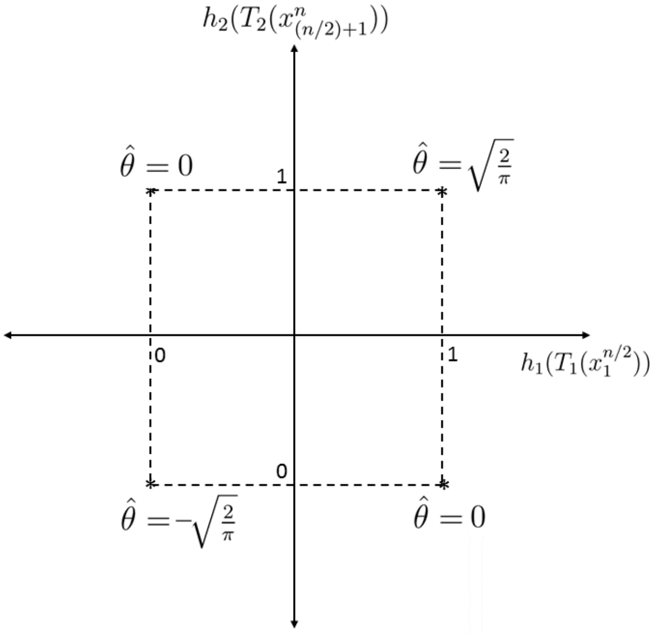
$h_2(T_2(x^n_{(n/2)+1}))$
$\hat{\theta} = 0$
1
$\hat{\theta} = \sqrt{\frac{2}{\pi}}$
0
1
$h_1(T_1(x_1^{n/2}))$
0
$\hat{\theta} = -\sqrt{\frac{2}{\pi}}$
$\hat{\theta} = 0$

Figure 3.2: The optimal estimation scheme for Gaussian distribution.

In this case, the AOB performance is given by

$$\begin{aligned} B_{AOB-1bit} &= \lim_{n\to\infty} \mathbb{E}[(\hat{\theta}-\theta)^2] \\ &= \lim_{n\to\infty} \int_{-\infty}^{\infty} \int_{\mathcal{X}_1^n} (\hat{\theta}-\theta)^2 f(x_1^n|\theta) f(\theta) dx_1^n d\theta \\ &= \lim_{n\to\infty} \int_{-\infty}^{\infty} [I_1 + I_2 + I_3] f(\theta) d\theta, \end{aligned} \tag{3.31}$$

where

$$I_1 = (\sqrt{\frac{2}{\pi}} - \theta)^2 [\Pr\big(T_1(X_1^{n/2}) \geq 0\big)|\theta\big)\Pr\big(T_2(X_{(n/2)+1}^n) \geq 0\big)|\theta\big)], \tag{3.32}$$

$$I_2 = (\sqrt{\frac{2}{\pi}} + \theta)^2 [\Pr\big(T_1(X_1^{n/2}) < 0\big)|\theta\big)\Pr\big(T_2(X_{(n/2)+1}^n) < 0\big)|\theta\big)], \tag{3.33}$$

$$\begin{aligned} I_3 &= \theta^2 [\Pr\big(T_1(X_1^{n/2}) \geq 0\big)|\theta\big)\Pr\big(T_2(X_{(n/2)+1}^n) < 0\big)|\theta\big) \\ &+ \Pr\big(T_1(X_1^{n/2}) < 0\big)|\theta\big)\Pr\big(T_2(X_{(n/2)+1}^n) \geq 0\big)|\theta\big)]. \end{aligned} \tag{3.34}$$

The AOB performance can be rewritten as

$$\begin{aligned} B_{AOB-1bit} &= \lim_{n\to\infty} \Big(\int_{-\infty}^{\infty} [(\sqrt{\frac{2}{\pi}} - \theta)^2 Q^2(-\sqrt{\frac{n}{2}}\theta) + (\sqrt{\frac{2}{\pi}} + \theta)^2 Q^2(\sqrt{\frac{n}{2}}\theta) \\ &+ 2\theta^2 Q(-\sqrt{\frac{n}{2}}\theta) Q(\sqrt{\frac{n}{2}}\theta)] f(\theta) d\theta \Big) \\ &= \int_0^{\infty} (\sqrt{\frac{2}{\pi}} - \theta)^2 f(\theta) d\theta + \int_{-\infty}^{0} (\sqrt{\frac{2}{\pi}} + \theta)^2 f(\theta) d\theta \\ &= 1 - \frac{2}{\pi}. \end{aligned} \tag{3.35}$$

□

This shows that the centralized estimation performance is not attainable at very low compression rates. However, the AOB performance is achievable through reducing the dimensionality of the observations prior to applying the compression scheme.

3.5 Simulation Results

In this section, we run two simulations to verify our theoretical results. In both simulations, the observations X_1^n are generated according to a Gaussian distribution $X \sim \mathcal{N}(\theta, 1)$, and the n observations are distributed evenly between two nodes such that $n_1 = n_2 = n/2$.

We use a 1-bit compression scheme to plot the simulated distortion against the number of observations, and we compare the results to the AOB performance. The simulated distortion is calculated as

$$D = |\hat{\theta} - \theta|^2. \tag{3.36}$$

In the first simulation, we assume that $\theta \in \mathbb{R}$ follows a Gaussian distribution $\theta \sim \mathcal{N}(0, 1)$, while in the second simulation, we assume that $\theta \in (-1, 1)$ follows a uniform distribution $\theta \sim U(-1, 1)$.

3.5.1 Gaussian Distribution: $\theta \sim \mathcal{N}(0, 1)$

Fig. 3.3 shows that the simulated distortion converges to the AOB performance given in (3.37) as the number of observations increase, which coincides with our results in Sections 3.2 and 3.4.

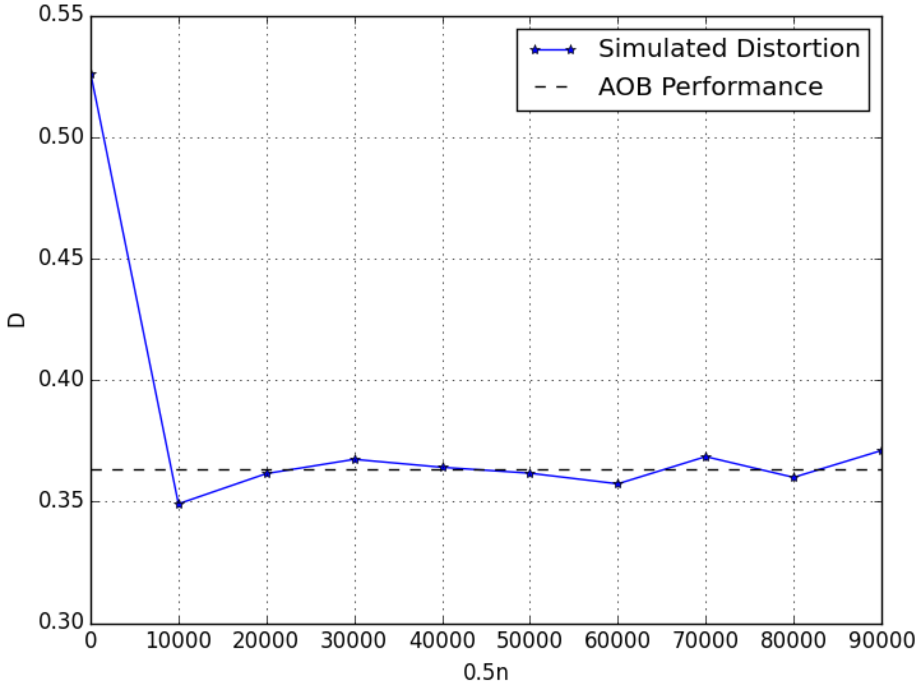

Figure 3.3: The simulated distortion for Gaussian distribution.

3.5.2 Uniform Distribution: $\theta \sim U(-1, 1)$

Our results in Section 3.4 can be easily extended to the case of uniform distribution. The encoders have the same design as in the Gaussian case, while the optimal estimation scheme is given in Fig. 3.4

In this case, the AOB performance is given by

$$\begin{aligned} B_{AOB-1bit} &= \int_0^1 (0.5-\theta)^2 f(\theta) d\theta + \int_{-1}^0 (0.5+\theta)^2 f(\theta) d\theta \\ &= \frac{1}{12}. \end{aligned} \tag{3.37}$$

Similar to the Gaussian case, Fig. 3.5 shows that the simulated distortion converges to the AOB performance as the number of observations increase.

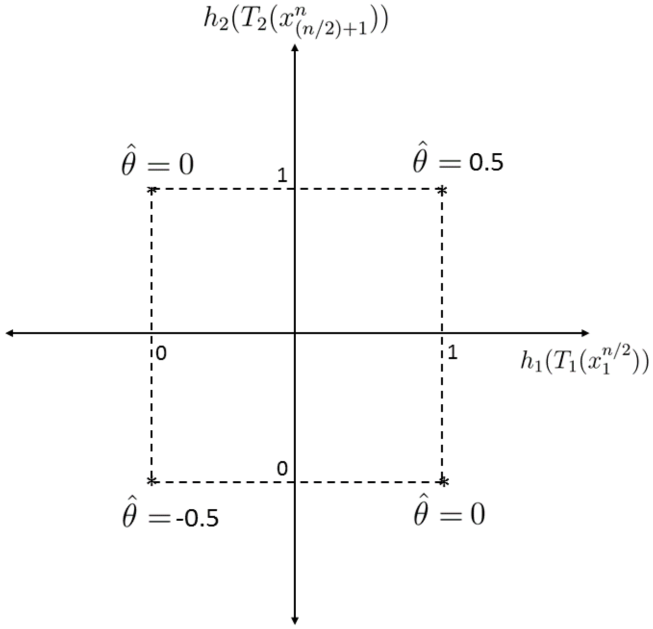

Figure 3.4: The optimal estimation scheme for uniform distribution.

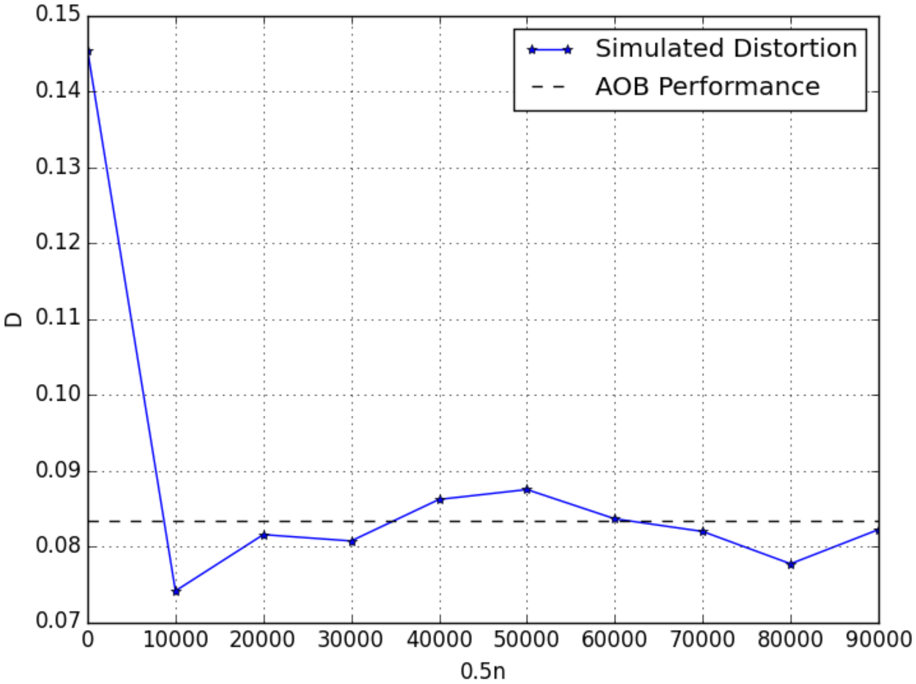

Figure 3.5: The simulated distortion for uniform distribution.

Chapter 4

Distributed Optimization with Quantized Updates

In this chapter, we analyze the convergence rate for distributed optimization algorithms in the presence of communication constraints. The chapter is organized as follows. We give a formal statement of the problem in Section 4.1. In Section 4.2 we introduce our algorithm. We analyze the convergence rate of our algorithm, and we derive sufficient conditions on the quantization error in Section 4.3. We extend our results to the general case of block coordinate case in Section 4.4. In Section 4.5, we analyze the convergence rate for the parallel setting, and we compare the two case of synchronous and asynchronous processing. We verify our results by running an experiment in Section 4.6.

4.1 Problem Formulation

We consider an unconstrained convex minimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}), \tag{4.1}$$

where $\mathbf{x} = \{x_1, x_2, ..., x_d\}$, and $f(\mathbf{x})$ is an L-smooth and m-strongly convex function, such that for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, we have that

$$\begin{aligned} ||\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})|| &\leq L||\mathbf{x} - \mathbf{y}||, & (4.2) \\ \langle \nabla f(\mathbf{x}) - \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle &\geq m||\mathbf{x} - \mathbf{y}||^2, & (4.3) \end{aligned}$$

where L is the Lipschitz constant and m is the strong convexity parameter. The condition number of f is defined as $g = L/m$. As a result of the strong convexity, the function $f(\mathbf{x})$ has a unique minimum at $\mathbf{x}^*$.

In the distributed coordinate descent algorithm, the data examples related to the problem are distributed over d nodes such that each node can calculate one coordinate of the gradient $\nabla f(\mathbf{x})$ as explained in Section 6 of [54]. The algorithm we study in this chapter is the randomized coordinate descent, in which at each iteration a coordinate is randomly selected to be updated. There are different ways to randomly select the coordinate. We focus on the case in which the coordinates are selected with a uniform distribution. The channels connecting machines are capacity limited with a quantization resolution of Δ, which means that machine i can only send a quantized version $Q\left(\frac{\partial f(\mathbf{x})}{\partial x_i}\right)$ of its update $\frac{\partial f(\mathbf{x})}{\partial x_i}$, such that

$$Q\left(\frac{\partial f(\mathbf{x})}{\partial x_i}\right) = y\Delta, \text{ if } (y - \frac{1}{2})\Delta \leq \frac{\partial f(\mathbf{x})}{\partial x_i} < (y + \frac{1}{2})\Delta, \tag{4.4}$$

in which $Q(\cdot)$ is the quantization operator. Let $[\nabla f(\mathbf{x})]_i \in \mathbb{R}^d$ denote a vector that has only one nonzero element at position i that is equal to $\frac{\partial f(\mathbf{x})}{\partial x_i}$. By applying the quantization operator to the nonzero element of the vector $[\nabla f(\mathbf{x})]_i \in \mathbb{R}^d$, we can rewrite (4.4) as

$$Q([\nabla f(\mathbf{x})]_i) = [\nabla f(\mathbf{x})]_i - \mathbf{n}, \tag{4.5}$$

where $\mathbf{n} \in \mathbb{R}^d$ is the quantization noise vector. The noise vector $\mathbf{n}$ has only one nonzero element n_i that is bounded as $|n_i| \leq \Delta/2$. Hence,

$$||\mathbf{n}|| \leq \frac{\Delta}{2}. \tag{4.6}$$

Throughout the chapter, we use $\mathbf{x}_k$ and $\mathbf{x}_k^q$ to denote the kth update of $\mathbf{x}$ before and after adding the quantization noise, respectively. An upper case letter S is used for a random variable, while a lower case letter s is used for a realization of S. We also use $||\mathbf{x}||$ to denote the Euclidean norm of the vector $\mathbf{x}$, and we use $Q(\cdot)$ to denote the quantization operator.

4.2 Quantized Randomized Coordinate Descent

Here, we describe the randomized coordinate descent algorithm with quantized update. The algorithm starts from an initial point $\mathbf{x}_0$, and stops after a predetermined number of iterations T. Set $\mathbf{x}_0^q = \mathbf{x}_0$. At iteration $(j+1)$, a machine $s_{j+1} \in \{1, 2, ..., d\}$ is randomly (with a uniform distribution) selected, who calculates $[\triangledown f(\mathbf{x}_j^q)]_{s_{j+1}}$ and then sends the quantized update $Q\left([\triangledown f(\mathbf{x}_j^q)]_{s_{j+1}}\right)$, all machines update

$$\mathbf{x}_{j+1}^q = \mathbf{x}_j^q - tdQ([\triangledown f(\mathbf{x}_j^q)]_{s_{j+1}}), \tag{4.7}$$

where t is the step size.

To facilitate the analysis, we also record the sequence

$$\mathbf{x}_{j+1} = \mathbf{x}_j^q - td[\triangledown f(\mathbf{x}_j^q)]_{s_{j+1}}. \tag{4.8}$$

Algorithm 1: Quantized Randomized Coordinate Descent

```
$\mathbf{x}_0^q = \mathbf{x}_0$
for $j = 0, 1, ..., (T-1)$ do
    a machine is randomly selected to send its update
    selected machine $s_{j+1}$ computes $[\nabla f(\mathbf{x}_j^q)]_{s_{j+1}}$
    machine $s_{j+1}$ communicates $Q([\nabla f(\mathbf{x}_j^q)]_{s_{j+1}})$
    all machines update $\mathbf{x}_{j+1}^q = \mathbf{x}_j^q - tdQ([\nabla f(\mathbf{x}_j^q)]_{s_{j+1}})$
end for
```

Using (4.23), we can show that

$$\mathbf{x}_j^q = \mathbf{x}_j + td\mathbf{n}_j,\ j = \{1, 2, ..., T\}. \tag{4.9}$$

It is desirable that the algorithm converges within k iterations to an accuracy level of ϵ and a confidence level of $\rho \in (0, 1)$, such that

$$\text{Pr}(||\mathbf{x}_k - \mathbf{x}^*||^2 \leq \epsilon) \geq 1 - \rho. \tag{4.10}$$

By applying Markov inequality, the convergence condition in (4.10) is achieved if

$$\mathbb{E}||\mathbf{x}_k - \mathbf{x}^*||^2 \leq \epsilon\rho. \tag{4.11}$$

4.3 Convergence Analysis

In this section, we analyze the convergence rate of the quantized randomized coordinate descent algorithm.

Theorem 7. *Given that the quantization error Δ is bounded as following*

$$\Delta \leq \frac{\epsilon\rho L^2}{2m}(\frac{1}{C_{min}} - 1), \tag{4.12}$$

the number of iterations required for the quantized randomized coordinate descent algorithm to converge to the optimal solution $\mathbf{x}^*$ *is at most*

$$\begin{aligned} k^q &= \frac{\log(2||\mathbf{x}_0 - \mathbf{x}^*||^2/\epsilon\rho)}{\log(1/C_{min})} \\ &+ \frac{\log(2||\mathbf{x}_0 - \mathbf{x}^*||^2)}{\log(1/(C_{min} + \frac{\epsilon\rho}{2}(1 - C_{min}))}, \end{aligned} \tag{4.13}$$

where

$$C_{min} = 1 - \frac{1}{g^2 d}.$$

Proof. We have that

$$\begin{aligned} ||\mathbf{x}_{j+1} - \mathbf{x}^*||^2 &= ||\mathbf{x}_j^q - \mathbf{x}^* - td[\nabla f(\mathbf{x}_j^q)]_{s_{j+1}}||^2 \\ &= ||\mathbf{x}_j^q - \mathbf{x}^*||^2 + t^2 d^2 ||[\nabla f(\mathbf{x}_j^q)]_{s_{j+1}}||^2 \\ &- 2td\langle [\nabla f(\mathbf{x}_j^q)]_{s_{j+1}}, \mathbf{x}_j^q - \mathbf{x}^* \rangle. \end{aligned} \tag{4.14}$$

Taking the expectation of both sides with respect to the independent and identically distributed (i.i.d.) random variables $S_1, S_2, ...S_{j+1}$

$$\begin{aligned} \mathbb{E}||\mathbf{x}_{j+1} - \mathbf{x}^*||^2 &= \mathbb{E}||\mathbf{x}_j^q - \mathbf{x}^*||^2 \\ &+ t^2 d^2 \mathbb{E}||[\nabla f(\mathbf{x}_j^q)]_{s_{j+1}}||^2 \\ &- 2td\mathbb{E}\langle [\nabla f(\mathbf{x}_j^q)]_{s_{j+1}}, \mathbf{x}_j^q - \mathbf{x}^* \rangle. \end{aligned} \tag{4.15}$$

Since $\mathbb{E}_{s_{j+1}}[\triangledown f(\mathbf{x}_j^q)]_{s_{j+1}} = \frac{1}{d}(\triangledown f(\mathbf{x}_j^q))$, then

$$\begin{aligned}\mathbb{E}||\mathbf{x}_{j+1} - \mathbf{x}^*||^2 &= \mathbb{E}||\mathbf{x}_j^q - \mathbf{x}^*||^2 + t^2 d\mathbb{E}||\triangledown f(\mathbf{x}_j^q)||^2 \\ &- 2t\mathbb{E}\langle \triangledown f(\mathbf{x}_j^q), \mathbf{x}_j^q - \mathbf{x}^*\rangle. \end{aligned} \tag{4.16}$$

By applying inequalities (4.2) and (4.3), and using the fact that $\triangledown f(\mathbf{x}^*) = 0$, we have that

$$||\triangledown f(\mathbf{x}_j^q)|| \leq L||\mathbf{x}_j^q - \mathbf{x}^*||, \tag{4.17}$$

and

$$\langle \triangledown f(\mathbf{x}_j^q), \mathbf{x}_j^q - \mathbf{x}^*\rangle \geq m||\mathbf{x}_j^q - \mathbf{x}^*||^2. \tag{4.18}$$

Substituting (4.17) and (4.18) in (4.16), we get that

$$\mathbb{E}||\mathbf{x}_{j+1} - \mathbf{x}^*||^2 \leq C\mathbb{E}||\mathbf{x}_j^q - \mathbf{x}^*||^2, \tag{4.19}$$

where $C = t^2L^2d - 2tm + 1$. The remaining of the proof can be found in Appendix E. □

This shows that the quantization error does not propagate, and hence the algorithm with quantized updates still converges to the optimal solution given that the error is bounded.

Note that By setting $\Delta = 0$ and hence $x_j^q = x_j$ in (4.19), the quantization-free scenario can be recovered. It follows that the number of iterations required for the quantization-free algorithm to converge is at most

$$k = \frac{\log(||\mathbf{x}_0 - \mathbf{x}^*||^2/\epsilon\rho)}{\log(1/C_{min})}, \tag{4.20}$$

which coincides with the result obtained in [55].

The main difference between (4.13) and (4.20) is the extra number of iterations required in the quantization scenario to get $||\mathbf{x}_{k^q} - \mathbf{x}^*||^2$ below 1 as shown in Fig. 4.7. For $\epsilon = 10^{-15}$, $\rho = 0.01$, $C_{min} = 0.5$, this difference is approximately fixed at around 17 iterations for $||\mathbf{x}_0 - \mathbf{x}^*||^2$ ranging from $10,000$ to $100,000$.

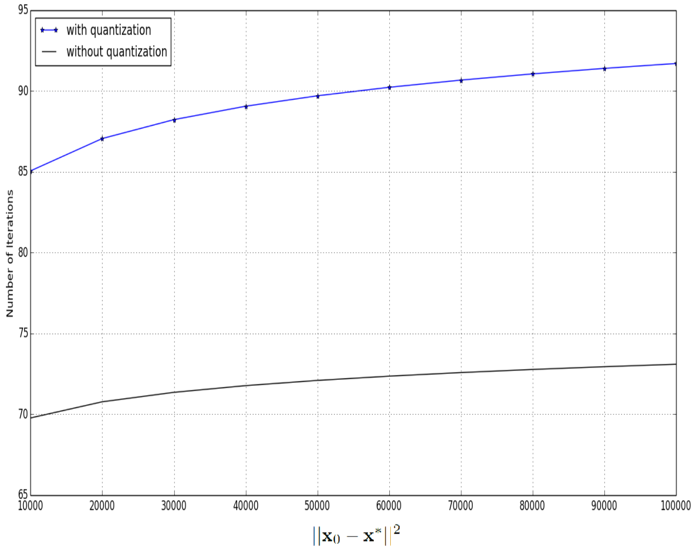

Figure 4.1: Comparison between k^q and k as a function of $||\mathbf{x}_0 - \mathbf{x}^*||^2$.

4.4 Quantized Block Coordinate Descent

In this section, we extend our results to the general case of randomized block coordinate descent, where the number of nodes M can take any value between 2 and d. In this case, node s can update a block $I_s \subset D = \{1, 2, ..., d\}$ of the coordinates. We have that

$$\begin{aligned} \bigcup_{i=1}^{M} I_i &= D, \\ I_i \bigcap I_j &= \phi, \;\; i \neq j. \end{aligned} \tag{4.21}$$

Our algorithm can handle this general problem in two ways:

- Algorithm 2: The selected node by the fusion center sends a full update for all its coordinates.

- Algorithm 3: The selected node samples a smaller set of its coordinates randomly and sends the update for the sampled set only.

The advantage of algorithm 3 is to reduce the computational cost at the selected node especially if it has a large block of coordinates. On the other hand, algorithm 2 requires less number of iterations for the algorithm to converge to the optimal solution. We start by analyzing algorithm 2.

Algorithm 2: Quantized Block Coordinate Descent - without Sampling

```
$\mathbf{x}_0^q = \mathbf{x}_0$
for $j = 0, 1, ..., (T-1)$ do
    a machine is randomly selected to send its update
    selected machine $s_{j+1}$ computes $[\nabla f(\mathbf{x}_j^q)]_{I_{s_{j+1}}}$
    machine $s_{j+1}$ communicates $Q([\nabla f(\mathbf{x}_j^q)]_{I_{s_{j+1}}})$
    all machines update $\mathbf{x}_{j+1}^q = \mathbf{x}_j^q - tMQ([\nabla f(\mathbf{x}_j^q)]_{I_{s_{j+1}}})$
end for
```

At iteration $(j+1)$, the fusion center randomly chooses node $s_{j+1} \in D$ with a uniform distribution to send its quantized update $Q([\nabla f(\mathbf{x}_j^q)]_{I_{s_{j+1}}})$. We have that

$$\mathbf{x}_{j+1} = \mathbf{x}_j^q - tM[\nabla f(\mathbf{x}_j^q)]_{I_{s_{j+1}}}, \tag{4.22}$$

where the vector of partial derivatives $[\nabla f(\mathbf{x}_j^q)]_{I_{s_{j+1}}} \in \mathbb{R}^d$ has only nonzero elements at positions $I_{s_{j+1}}$. In this case, the norm of the quantization noise vector $\mathbf{n}$ is bounded as

$$||\mathbf{n}|| \leq \frac{\Delta\sqrt{l}}{2}, \tag{4.23}$$

where l is the number of nonzero elements in $\mathbf{n}$. Define l_m as the maximum block length $l_m = \max_{i=1}^{M} l_i$, where $l_i = |I_i|$.

Corollary 1. *Given that the quantization error Δ is bounded as following*

$$\Delta \leq \frac{\epsilon \rho L^2}{2m\sqrt{l_m}}(\frac{1}{C_{1_{min}}} - 1), \tag{4.24}$$

the number of iterations required for algorithm 2 to converge to the optimal solution $\mathbf{x}^*$ is at most

$$\begin{aligned} k^q &= \frac{\log(2||\mathbf{x}_0 - \mathbf{x}^*||^2/\epsilon\rho)}{\log(1/C_{1_{min}})} \\ &+ \frac{\log(2||\mathbf{x}_0 - \mathbf{x}^*||^2)}{\log(1/(C_{1_{min}} + \frac{\epsilon\rho}{2}(1 - C_{1_{min}}))}. \end{aligned} \tag{4.25}$$

where

$$C_{1_{min}} = 1 - (1/(g^2 M)).$$

Proof. The proof of this corollary can be found in Appendix F. □

Notice that a small number of nodes M leads to a faster convergence rate at the expense of a higher computational cost per node. Next, we analyze algorithm 3.

Algorithm 3: Quantized Block Coordinate Descent - with Sampling

```
$\mathbf{x}_0^q = \mathbf{x}_0$
for $j = 0, 1, ..., (T-1)$ do
    a machine is randomly selected to send its update
    selected machine $s_{j+1}$ randomly samples $Q_{s_{j+1}} \subset I_{s_{j+1}}$ to update
    machine $s_{j+1}$ computes $[\nabla f(\mathbf{x}_j^q)]_{Q_{s_{j+1}}}$
    machine $s_{j+1}$ communicates $Q([\nabla f(\mathbf{x}_j^q)]_{Q_{s_{j+1}}})$
    all machines update $\mathbf{x}_{j+1}^q = \mathbf{x}_j^q - tMQ([\nabla f(\mathbf{x}_j^q)]_{Q_{s_{j+1}}})$
end for
```

At iteration $(j+1)$, the fusion center randomly chooses node $s_{j+1} \in D$ with a uniform distribution to send its update. The selected node s_{j+1} randomly samples a set of coordinates $Q_{s_{j+1}} \subset I_{s_{j+1}}$ to update.

Let q_i denotes the size of the set Q_i. For simplicity, we assume that q_i takes the following values

$$q_i = \begin{cases} q, \text{ if } q < l_i \\ l_i, \text{ otherwise} \end{cases}, \tag{4.26}$$

where q is a constant that can be adjusted as one of the algorithm parameters. This simply means that node i updates all its coordinates if $l_i \leq q$. Otherwise, the node samples and updates a subset of its coordinates $Q_i \subset I_i$, such that $|Q_i| = q$. In this case, we have that

$$\mathbf{x}_{j+1} = \mathbf{x}_j^q - tM[\nabla f(\mathbf{x}_j^q)]_{Q_{s_{j+1}}}, \tag{4.27}$$

where the vector of partial derivatives $[\nabla f(\mathbf{x}_j^q)]_{Q_{s_{j+1}}} \in \mathbb{R}^d$ has only nonzero elements at positions $Q_{s_{j+1}}$.

Corollary 2. *Given that the quantization error Δ is bounded as following*

$$\Delta \leq \frac{\epsilon \rho L^2 l_m}{2mq\sqrt{q}}(\frac{1}{C_{2_{min}}} - 1), \tag{4.28}$$

the number of iterations required for algorithm 3 to converge to the optimal solution $\mathbf{x}^*$ is at most

$$\begin{aligned} k^q &= \frac{\log(2||\mathbf{x}_0 - \mathbf{x}^*||^2/\epsilon\rho)}{\log(1/C_{2_{min}})} \\ &+ \frac{\log(2||\mathbf{x}_0 - \mathbf{x}^*||^2)}{\log(1/(C_{2_{min}} + \frac{\epsilon\rho}{2}(1 - C_{2_{min}}))}. \end{aligned} \tag{4.29}$$

where

$$C_2 = t^2 L^2 M - \frac{2tmq}{l_m} + 1.$$

Proof. The proof of this corollary can be found in Appendix G. □

Notice that a large value of the ratio q/l_m leads to a faster convergence rate at the expense of a higher computational cost per node. We also observe that reducing the value of the parameter q results in a tighter upper bound on the quantization error. To explain this, we notice that a decrease in q means less quantization noise per iteration, but it also leads to a slower convergence rate, and hence increases the overall quantization noise, which requires a tighter bound on the quantization error to guarantee the convergence of the algorithm. This observation is illustrated in Fig. 4.2 for ($\epsilon = 10^{-15}$, $\rho = 0.01$, $g^2 = 2$, $m = 1$, $M = 20$, $l_m = 10$).

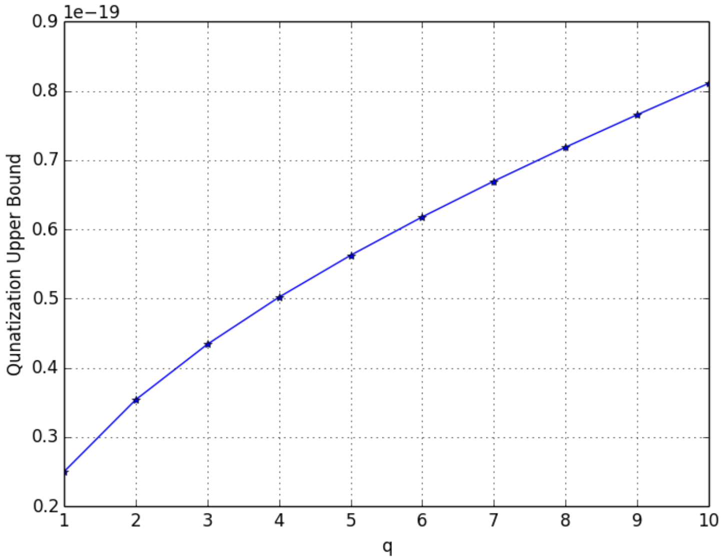

Figure 4.2: The effect of the parameter q on the quantization condition.

4.5 Parallel Processing

In this section, we analyze a parallel version of our algorithm in both scenarios, where the nodes are synchronized or not. In the synchronous scenario, all nodes send their updates as they finish computing them. The fusion center combines all the updates in a single message and send it to all nodes in order to compute their next updates. The main disadvantage of this scenario is that if there is one slow node, then all other nodes have to wait for that node before they start computing the next update. In other words, the processing speed is bounded by the slowest node. On the other hand, in th asynchronous scenario, the fusion center forwards the individual updates as it receives them. This potentially leads to different nodes processing different updates, and hence the convergence analysis is much difficult than the synchronous case. We start by analyzing the convergence rate for the synchronous scenario.

4.5.1 Synchronous Parallel Processing

Similar to algorithm 3 in Section 4.4, node i updates all its coordinates if $l_i \leq q$. Otherwise, it samples and updates a subset of the coordinates $Q_i \subset I_i$, such that $|Q_i| = q$.

Algorithm 4: Quantized Synchronous Parallel Coordinate Descent

```
1: $\mathbf{x}_0^q = \mathbf{x}_0$
2: for $j = 0, 1, ..., (T-1)$ do
3: machine $i$ computes $[\triangledown f(\mathbf{x}_j^q)]_{Q_{j,i}}$, $i = 1, 2, ..., M$
4: machine $i$ sends its quantized update $Q([\triangledown f(\mathbf{x}_j^q)]_{Q_{j,i}})$, $i = 1, 2, ..., M$
5: all machines update $\mathbf{x}_{j+1}^q = \mathbf{x}_j^q - t\sum_{i=1}^{M} Q([\triangledown f(\mathbf{x}_j^q)]_{Q_{j,i}})$
6: end for
```

Corollary 3. *Given that the quantization error Δ is bounded as following*

$$\Delta \leq \frac{\epsilon \rho L^2 l_m}{2mq\sqrt{r}}(\frac{1}{C}_{s_{min}} - 1), \tag{4.30}$$

the number of iterations required for algorithm 4 to converge to the optimal solution $\mathbf{x}^*$ *is at most*

$$\begin{aligned} k^q &= \frac{\log(2||\mathbf{x}_0 - \mathbf{x}^*||^2/\epsilon\rho)}{\log(1/C_{s_{min}})} \\ &+ \frac{\log(2||\mathbf{x}_0 - \mathbf{x}^*||^2)}{\log(1/(C_{s_{min}} + \frac{\epsilon\rho}{2}(1 - C_{s_{min}}))}. \end{aligned} \tag{4.31}$$

where

$$C_s = t^2L^2 - \frac{2tmq}{l_m} + 1.$$

and r *is defined as* $r = \min\{d, qM\}$*, which is the maximum number of coordinates that can be updated in a single iteration.*

Proof. The proof of this corollary can be found in Appendix H. □

Notice that the number of nodes M does not affect the convergence rate. Similar to algorithm 3 in the previous section, a large value of the ratio q/l_m leads to a faster convergence rate at the expense of a higher computational cost per node.

Similar to algorithm 3, decreasing the value of the parameter q results in a tighter upper bound on the quantization error. For ($\epsilon = 10^{-15}$, $\rho = 0.01$, $g^2 = 2$, $m = 1$, $l_m = 10$, $d = 100$, $M = 20$), the rate of change in the upper bound decreases as q falls below 5 ($qM < d$) as shown in Fig. 4.4. Next, we analyze the case of asynchronous parallel processing.

4.5.2 Asynchronous Parallel Processing

The analysis of algorithm 5 can be difficult especially in the presence of quantization error. Therefore, we analyze the convergence rate for the special case of two nodes,

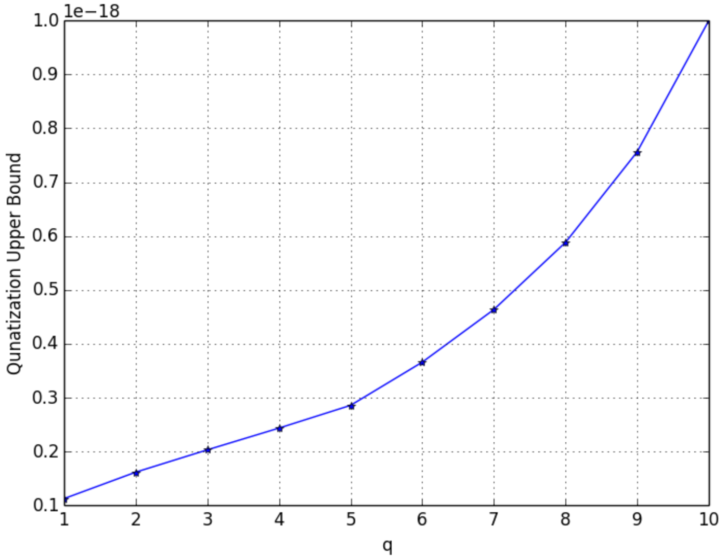

Figure 4.3: Synchronous Parallel processing: the effect of the parameter q on the quantization condition.

Algorithm 5: Quantized Asynchronous Parallel Coordinate Descent

```
1: $\mathbf{x}_0^q = \mathbf{x}_0$
2: for $j = 0, 1, ..., (T-1)$ do
3: machine $i$ computes $[\nabla f(\mathbf{x}_j^q)]_{Q_{j,i}}$
4: machine $i$ sends its quantized update $Q([\nabla f(\mathbf{x}_j^q)]_{Q_{j,i}})$
5: the fusion center updates $\mathbf{x}_{j+1}^q = \mathbf{x}_j^q - tQ([\nabla f(\mathbf{x}_j^q)]_{Q_{j,i}})$
6: machine $i$ requests the updated value of $\mathbf{x}$ from the fusion center
7: end for
```

where node s_1 is twice as fast as node s_2, and each node updates a single coordinate. We also assume stronger conditions on the function smoothness and convexity, such that

$$||[\nabla f(\mathbf{x})]_{s_i}||^2 \leq \frac{L^2}{4}||\mathbf{x} - \mathbf{x}^*||^2, \quad i = 1, 2, \tag{4.32}$$

and

$$\langle \mathbf{x} - \mathbf{x}^*, [\nabla f(\mathbf{x})]_{s_i} \rangle \geq \frac{m}{2}||\mathbf{x} - \mathbf{x}^*||^2, \quad i = 1, 2. \tag{4.33}$$

For $j = \{0, 2, ...\}$, both nodes s_1 and s_2 are processing the same value of the optimization variable $\mathbf{x}_j^q$. We have that

$$\mathbf{x}_{j+1}^q = \mathbf{x}_j^q - tQ([\nabla f(\mathbf{x}_j^q)]_{s_1}) \tag{4.34}$$

and

$$\begin{aligned} \mathbf{x}_{j+2}^q &= \mathbf{x}_{j+1}^q - tQ([\nabla f(\mathbf{x}_{j+1}^q)]_{s_1}) - tQ([\nabla f(\mathbf{x}_j^q)]_{s_2}) \\ &= \mathbf{x}_j^q - tQ(\nabla f(\mathbf{x}_j^q)) - tQ([\nabla f(\mathbf{x}_{j+1}^q)]_{s_1}) \end{aligned} \tag{4.35}$$

Let

$$\mathbf{x}_{j+1} = \mathbf{x}_j^q - t([\nabla f(\mathbf{x}_j^q)]_{s_1}) \tag{4.36}$$

and

$$\mathbf{x}_{j+2} = \mathbf{x}_j^q - t(\nabla f(\mathbf{x}_j^q)) - t([\nabla f(\mathbf{x}_{j+1})]_{s_1}) \tag{4.37}$$

Similar to our analysis in Section 4.3, we get that

$$||\mathbf{x}_{j+1} - \mathbf{x}^*||^2 \leq (\frac{t^2L^2}{4} - tm + 1)||\mathbf{x}_j^q - \mathbf{x}^*||^2, \tag{4.38}$$

and

$$\begin{aligned} ||\mathbf{x}_{j+2} - \mathbf{x}^*||^2 &\leq (t^2L^2 - 2tm + 1)||\mathbf{x}_j^q - \mathbf{x}^*||^2 \\ &+ \frac{t^2L^2}{4}||\mathbf{x}_{j+1} - \mathbf{x}^*||^2 \qquad (4.39) \\ &- 2t\langle \mathbf{x}_{j+1} - \mathbf{x}^*, [\triangledown f(\mathbf{x}_{j+1})]_{s_1}\rangle \\ &\leq C_a^2||\mathbf{x}_j^q - \mathbf{x}^*||^2, \qquad (4.40) \end{aligned}$$

where

$$C_a^2 = (t^2L^2 - 2tm + 1) + (\frac{t^2L^2}{4} - tm + 1)(\frac{t^2L^2}{4} - tm). \qquad (4.41)$$

Since

$$\mathbf{x}_j^q = \mathbf{x}_j + t\mathbf{n}_j, \qquad (4.42)$$

where $|n_{j_1}| \leq \Delta$ and $|n_{j_2}| \leq \frac{\Delta}{2}$, then

$$\begin{aligned} ||\mathbf{x}_j^q - \mathbf{x}^*||^2 &\leq ||\mathbf{x}_j - \mathbf{x}^*||^2 + \sqrt{5}t\Delta||\mathbf{x}_j - \mathbf{x}^*|| \\ &+ \frac{5t^2\Delta^2}{4}. \qquad (4.43) \end{aligned}$$

Following similar steps to the analysis in Section 4.3, the sufficient condition on the quantization error is given by

$$\Delta \leq \frac{\epsilon\rho}{2\sqrt{5}t}(\frac{1}{C_a} - 1), \qquad (4.44)$$

and the number of iterations required for convergence is at most

$$\begin{aligned} k^q &= \frac{\log(2||\mathbf{x}_0 - \mathbf{x}^*||^2/\epsilon\rho)}{\log(1/C_a)} \\ &+ \frac{\log(2||\mathbf{x}_0 - \mathbf{x}^*||^2)}{\log(1/(C_a + \frac{\epsilon\rho}{2}(1 - C_a))}. \end{aligned} \tag{4.45}$$

To compare this result to the synchronous scenario, we use the same step size $t = 1/(gL)$, and we define a time unit (I) as the time needed for node s_1 to compute a single update.

For the synchronous scenario, the upper bound on the quantization error is given by

$$\Delta_s \leq \frac{\epsilon\rho L^2}{2m}(\frac{1}{C}_s - 1), \tag{4.46}$$

where $C_s = 1 - \frac{1}{g^2}$, and the number of time units required for convergence is at most

$$\begin{aligned} I_s &= \frac{2\log(2||\mathbf{x}_0 - \mathbf{x}^*||^2/\epsilon\rho)}{\log(1/C_s)} \\ &+ \frac{2\log(2||\mathbf{x}_0 - \mathbf{x}^*||^2)}{\log(1/(C_s + \frac{\epsilon\rho}{2}(1 - C_s))}, \end{aligned} \tag{4.47}$$

while for the asynchronous scenario, the upper bound on the quantization error is given by

$$\Delta_a \leq \frac{\epsilon\rho L^2}{2\sqrt{5}m}(\frac{1}{C}_a - 1), \tag{4.48}$$

where $C_a = \sqrt{(1 + \frac{9}{16g^4} - \frac{3}{4g^2})}$, and the number of time units required for convergence is at most

$$I_a = \frac{\log(2||\mathbf{x}_0 - \mathbf{x}^*||^2/\epsilon\rho)}{\log(1/C_a)} + \frac{\log(2||\mathbf{x}_0 - \mathbf{x}^*||^2)}{\log(1/(C_a + \frac{\epsilon\rho}{2}(1 - C_a))}. \tag{4.49}$$

In figures 4.4 and 4.5, we plot (Δ_s, Δ_a) against $1/g$, and (I_s, I_a) against $1/g$, respectively. We assume that ($\epsilon = 10^{-15}$, $\rho = 0.01$, $m = 1$, $||\mathbf{x}_0 - \mathbf{x}^*||^2 = 50,000$).

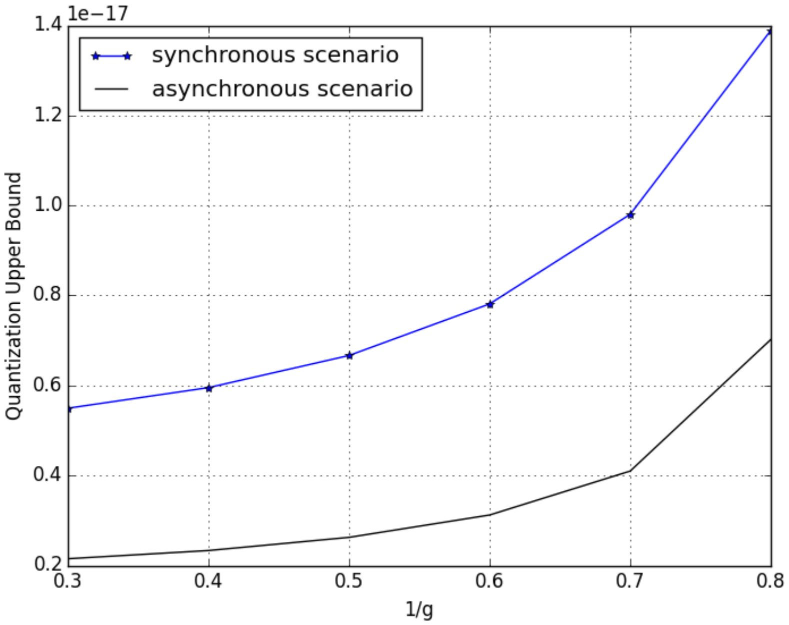

Figure 4.4: The quantization upper bound in the synchronous and asynchronous scenarios.

Although the quantization upper bound is tighter in the asynchronous scenario compared to the synchronous one as shown in Fig. 4.4, but the convergence speed is faster in the asynchronous case as shown in Fig. 4.5. This results is intuitive as in the synchronous scenario, node s_1 has to wait for node s_2 to finish processing before it works on the next update, while in the asynchronous scenario, the two nodes are totally independent. Next, we run an experiment to verify our theoretical results.

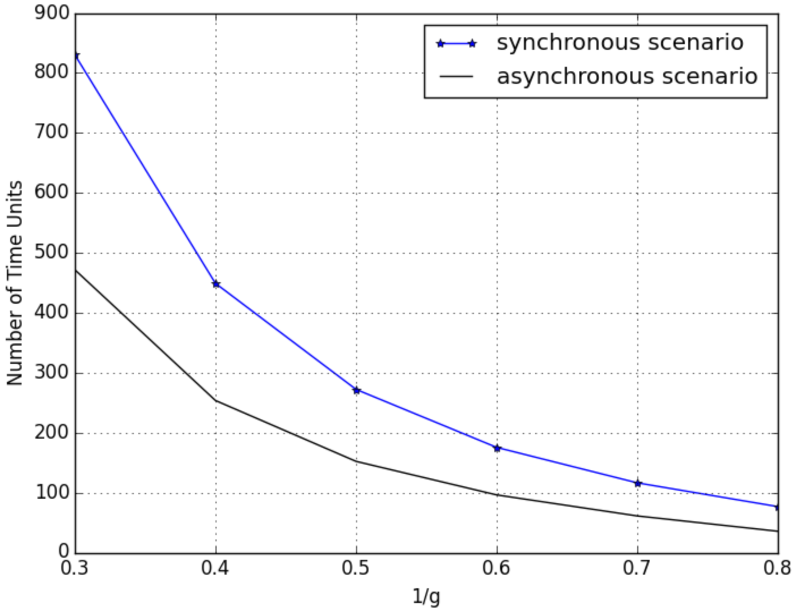

Figure 4.5: The convergence speed in the synchronous and asynchronous scenarios.

4.6 Simulation Results

In this section, we run an experiment to verify that the quantization error does not propagate and hence the convergence is possible. For that purpose, we apply the quantized randomized coordinate descent algorithm to solve a linear regression problem. The data set we use is collected from a power plant [59]. It has four predictors (Temperature, Pressure, Humidity, and Exhaust Vacuum) and one output (Electrical Energy). All data is normalized to have zero mean and a standard deviation of one. The number of observations is $n = 9568$.

To solve this problem, it is required to minimize the square loss function

$$f(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{n}(y_i - A_{i:}\mathbf{x})^2, \tag{4.50}$$

where A is the data matrix, $A_{i:}$ is the ith row of A, and $\mathbf{y}$ is the output vector. Notice that the first column of A is a vector of ones, which is added to evaluate for the intercept. The network consists of five nodes in addition to the fusion center; the first node calculates the derivative in the direction of the intercept coefficient, while each of the remaining nodes calculates the derivative in the direction of one predictor coefficient. The algorithm starts from $\mathbf{x}_0 = [1, 1, 1, 1, 1]^T$ and iterates to reduce the coefficients residual $||\mathbf{x}_j - \mathbf{x}^*||^2$.

Experiment 1: $t = 10^{-4}$, $\Delta = 10^5$.

First, we plot the coefficients residual against the number of iterations as shown in Fig. 4.6.

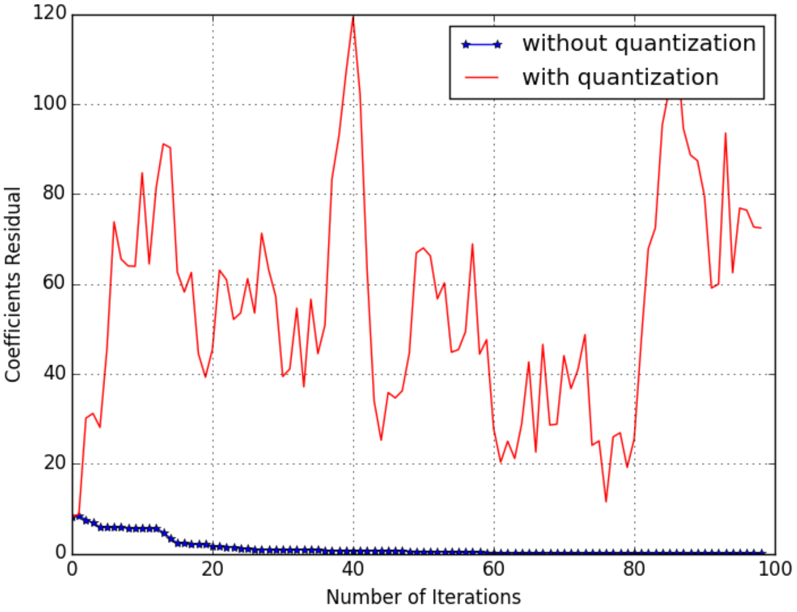

Figure 4.6: Effect of the quantization error ($\Delta = 10^5$) on the coefficients residual.

Then, we plot the predicted value for an input of all ones $A_1 = [1, 1, 1, 1, 1]$ against the number of iterations as shown in Fig. 2.2.

Figures 4.6 and 4.7 show that the quantized randomized coordinate descent algorithm diverges if the quantization error $\Delta = 10^5$. This result is intuitive since a large quantization error is expected to prevent the algorithm from converging to the optimal solution.

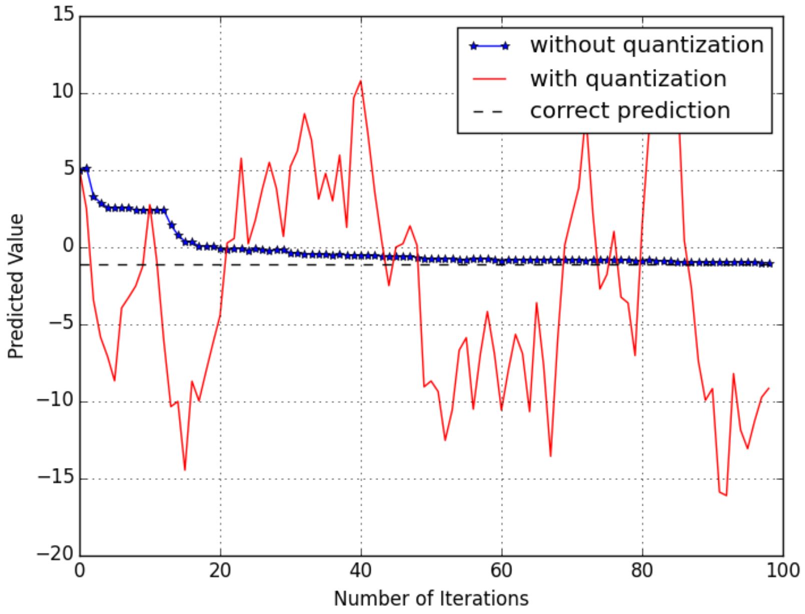

Figure 4.7: Effect of the quantization error ($\Delta = 10^5$) on the predicted value.

Experiment 2: $t = 10^{-4}$, $\Delta = 10^3$.

First, we plot the coefficients residual against the number of iterations as shown in Fig. 4.8.

Then, we plot the predicted value for an input of all ones $A_1 = [1, 1, 1, 1, 1]$ against the number of iterations as shown in Fig. 4.9.

Figures 4.8 and 4.9 show that the quantized randomized coordinate descent algorithm converges for a smaller value of the quantization error $\Delta = 10^3$. This verifies that the quantization error does not propagate and hence the convergence is possible if the quantization error is bounded, which coincides with the result we obtained in Theorem 1.

Notice that the exact bound on the quantization error cannot be computed in this case as the Lipschitz constant L and the strong convexity parameter m are unknown for the square loss function.

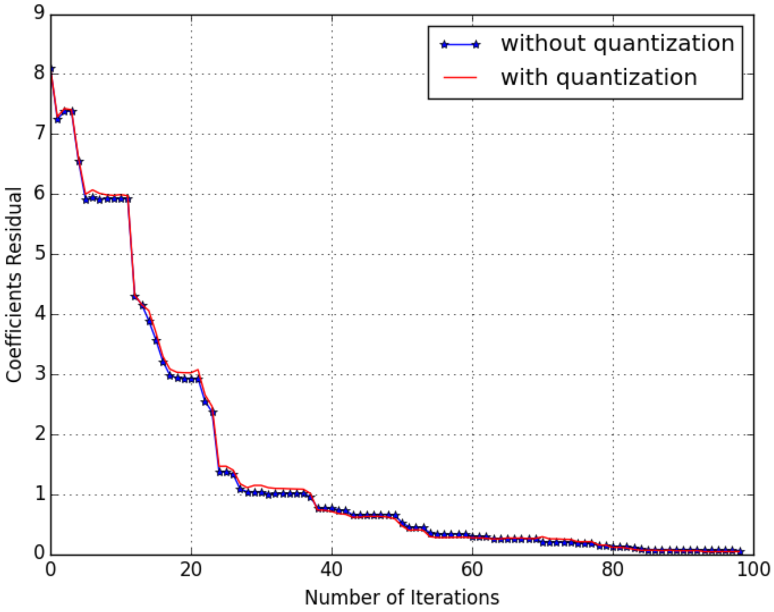

Figure 4.8: Effect of the quantization error ($\Delta = 10^3$) on the coefficients residual.

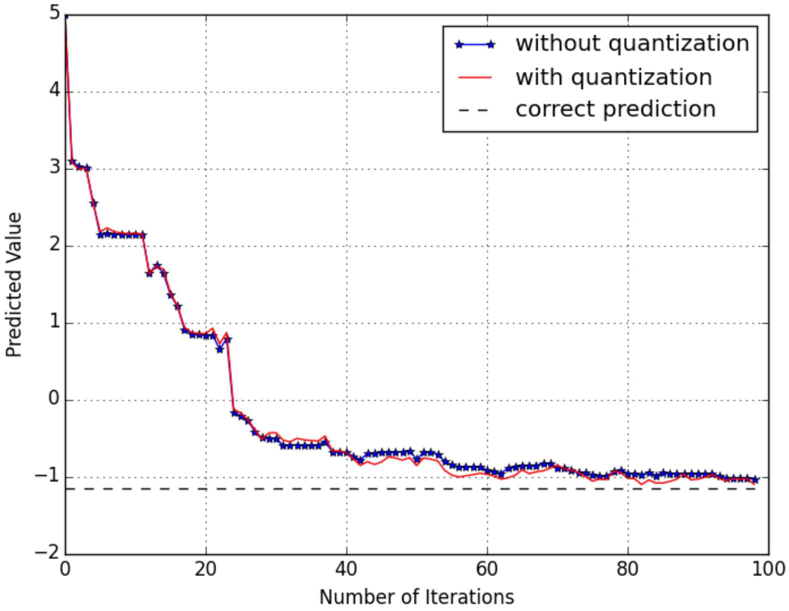

Figure 4.9: Effect of the quantization error ($\Delta = 10^3$) on the predicted value.

Chapter 5

Summary and Future Work

5.1 Summary

We have studied distributed statistical learning in the presence of communication constraints. In particular, we have considered the problems of distributed parameter estimation and distributed optimization. For distributed parameter estimation, we have analyzed the rate requirements for the distributed setup to achieve the optimal centralized performance. Then, we have examined the optimality of reducing the dimensionality of the observation prior to applying the compression scheme for the case when the rate is not high enough to achieve the centralized performance. For distributed optimization, we have analyzed the convergence rate for different distributed optimization algorithms in the presence of quantization error.

In particular, we have first answered the question: Are Slepian-Wolf rates necessary to achieve the same estimation performance as that of the centralized case? We have showed that the answer to this question is negative by constructing an asymptotically MVUE for binary symmetric sources using rates less than the conditions required in the Slepian-Wolf rate region. We have showed that our estimation algorithm can work for general binary sources to achieve the centralized estimation performance. We have also extended our work to non-binary information sources and multiple source networks by modifying our estimation algorithm. We have further proposed a practical design of our estimation algorithm and compared our results to the best known estimator by Han and Amari to show the superiority of our estimator.

We have then examined the optimality of reducing the dimensionality of the observations before applying the compression scheme. We have showed that reducing the dimensionality by extracting sufficient statistics of the parameter to be estimated does not degrade the overall estimation performance in the presence of communication constraints. We have established this result by comparing two system models, one applies the compression scheme to raw observations, and the other applies the compression scheme to the extracted sufficient statistics. We have proved that both system models have the same performance measured by the Bayesian risk. We have further analyzed the asymptotic optimal Bayesian performance in the presence of communication constraints, and we have verified our results through simulations.

We have finally studied the problem of distributed optimization under communication constraints. We have modified the randomized coordinate descent algorithm to solve an unconstrained convex minimization problem in the presence of quantization error. We have analyzed the convergence rate of our algorithm, and we have derived sufficient conditions on the quantization error to guarantee that the algorithm converges to the optimal solution. We have extended our results to the general case of block coordinate descent. We have analyzed the convergence rate for the parallel setting, and we have compared the two cases of synchronous and asynchronous parallel processing. We have further verified that the convergence is possible in the presence of quantization error by running an experiment that solves a linear regression problem.

5.2 Future Work

The future work of our research can be directed to a number of interesting problems, such as:

- Obtaining necessary conditions on the required rates for distributed parameter estimation. These conditions along with our results will fully identify the optimal rate region required to achieve the centralized estimation performance. For binary symmetric sources, the MVUE we have established is unique since it is a function of a complete statistic

  $$\hat{\theta} = \frac{n(0|\hat{Z}^n)}{n}. \tag{5.1}$$

  This fact can be used to simplify the problem to finding the optimal rates for computing a specific function, which is the MVUE in our case. It is also possible to apply some of the information-theoretic tools that are used to obtain the converse result for the source coding problem.

- Studying the optimality of sufficiency based data reduction for the non-Bayesian case using the minimax risk

  $$\begin{aligned} M_a &= \inf_{g_1,g_2,\phi_a} \sup_{\Theta} \mathbb{E}_X[(\theta - \hat{\theta}_a)^2], && (5.2) \\ M_b &= \inf_{h_1,h_2,\phi_b} \sup_{\Theta} \mathbb{E}_X[(\theta - \hat{\theta}_b)^2]. && (5.3) \end{aligned}$$

  In this case, the unknown parameter is deterministic, and hence it is required to consider the worst case scenario. This complicates the problem as the encoder has no knowledge of the value of θ that results in the worst performance outcome. It is possible to solve the problem through establishing the equivalence of the two system models (a) and (b) at each value of θ. One can argue then that the two models have the same worst case performance.

- Considering more practical models for asynchronous parallel processing in solving the problem of distributed optimization with quantized updates. This problem is very challenging as there are two sources of noise in this case: 1) the quantization noise ($\mathbf{n}_Q$); 2) the noise introduced from processing inconsistent updates ($\mathbf{n}_A$)

  $$\mathbf{x}^q = \mathbf{x} + td(\mathbf{n}_Q + \mathbf{n}_A). \tag{5.4}$$

  It is possible to solve the problem by combining the techniques we used in this study with the ones applied in the convergence analysis of asynchronous randomized coordinate descent [55].

Appendix A

Error Probability Analysis in the Proof of Theorem 1

To analyze the probability of the decoding error, let $\tilde{z}^n \in \{0,1\}^n$ denote a sequence such that

$$\tilde{z}^n \neq z^n, \quad f(\tilde{z}^n) = f(z^n). \tag{A.1}$$

Let $\tilde{Z}^{(n)}$ be a dummy random variable whose PMF $P_{\tilde{Z}^{(n)}}$ is the same as the type of $\tilde{z}^n$. Define $\mathcal{P}^{(n)}_{Z\tilde{Z}}$ as the set of all joint types between any two sequences z^n and $\tilde{z}^n$. For any given f (equivalently for a given encoding matrix A), define $N^n_f(Z\tilde{Z})$ as the number of sequences z^n such that there exists another sequence $\tilde{z}^n$ having the joint type $P_{Z^{(n)}\tilde{Z}^{(n)}} \in \mathcal{P}^{(n)}_{Z\tilde{Z}}$ and (A.1) holds.

Since each entry in A is uniformly distributed, then each element in $f(z^n)$ is uniformly distributed if z^n is a nonzero sequence. Therefore,

$$\text{Pr}(f(z^n) = 0) = (0.5)^{nR} = \frac{1}{||f||}, \tag{A.2}$$

in which the probability is computed over all codebooks. This implies that

$$\Pr(f(\tilde{z}^n) = f(z^n)) = \Pr(f(\tilde{z}^n - z^n) = 0) = \frac{1}{||f||}. \tag{A.3}$$

Define $T_{P_{Z^{(n)}\tilde{Z}^{(n)}}}$ as the set of all sequence pairs $(z^n, \tilde{z}^n)$ that have the joint type $P_{Z^{(n)}\tilde{Z}^{(n)}}$, $T_{P_{Z^{(n)}}}$ as the set of all sequences z^n that have the marginal type $P_{Z^{(n)}}$, and $T_{P_{\tilde{Z}^{(n)}|Z^{(n)}}}(z^n)$ as the set of all sequences $\tilde{z}^n$ that have the joint type $P_{Z^{(n)}\tilde{Z}^{(n)}}$ with z^n. The sizes of the sets $T_{P_{Z^{(n)}}}$ and $T_{P_{\tilde{Z}^{(n)}|Z^{(n)}}}(z^n)$ are bounded as [60]

$$\begin{aligned} ||T_{P_{Z^{(n)}}}|| &\leq 2^{nH(Z^{(n)})}, \\ ||T_{P_{\tilde{Z}^{(n)}|Z^{(n)}}}(z^n)|| &\leq 2^{nH(\tilde{Z}^{(n)}|Z^{(n)})+\epsilon}, \end{aligned} \tag{A.4}$$

where ϵ is an arbitrary small number. Notice that, for any given $P_{Z^{(n)}\tilde{Z}^{(n)}}$, $N_f^n(Z\tilde{Z})$ is a random variable (random over f) that can be expressed as

$$\begin{aligned} N_f^n(Z\tilde{Z}) &= \sum_{z^n \in T_{P_{Z^{(n)}}}} \mathbf{1}\big(\exists \tilde{z}^n \neq z^n : f(\tilde{z}^n) = f(z^n), \\ &\quad \text{and} \quad (z^n, \tilde{z}^n) \in T_{P_{Z^{(n)}\tilde{Z}^{(n)}}}\big) \\ &= \sum_{z^n \in T_{P_{Z^{(n)}}}} \mathbf{1}\big(\exists \tilde{z}^n \neq z^n : f(\tilde{z}^n) = f(z^n), \\ &\quad \text{and} \quad \tilde{z}^n \in T_{P_{\tilde{Z}^{(n)}|Z^{(n)}}}(z^n)\big), \end{aligned} \tag{A.5}$$

where $\mathbf{1}(\cdot)$ is the indication function. The expectation of $N_f^n(Z\tilde{Z})$ over all possible codebooks f is

$$\begin{aligned}\mathbb{E}[N_f^n(Z\tilde{Z})] &= \sum_{z^n \in T_{P_{Z^{(n)}}}} \mathbb{E}\big[\mathbf{1}\big(\exists\, \tilde{z}^n \neq z^n \;:\; f(\tilde{z}^n) = f(z^n),\\ &\text{and} \quad \tilde{z}^n \in T_{P_{\tilde{Z}^{(n)}|Z^{(n)}}}(z^n)\big)\big]\\ &\leq \sum_{z^n \in T_{P_{Z^{(n)}}}} \sum_{\tilde{z}^n \in T_{P_{\tilde{Z}^{(n)}|Z^{(n)}}}(z^n)} \text{Pr}(f(\tilde{z}^n) = f(z^n)). \qquad \text{(A.6)}\end{aligned}$$

(A.3), (A.4), and (A.6) imply that

$$\mathbb{E}[N_f^n(Z\tilde{Z})] \leq \frac{2^{n(H(Z^{(n)})+H(\tilde{Z}^{(n)}|Z^{(n)})+\epsilon)}}{||f||}. \qquad \text{(A.7)}$$

Applying the Markov's inequality, we have

$$\text{Pr}\left(N_f^n(Z\tilde{Z}) \geq \frac{2^{n(H(Z^{(n)})+H(\tilde{Z}^{(n)}|Z^{(n)})+\epsilon)}(||\mathcal{P}_{Z\tilde{Z}}^{(n)}||+\delta)}{||f||}\right) \leq \frac{1}{||\mathcal{P}_{Z\tilde{Z}}^{(n)}||+\delta}, \qquad \text{(A.8)}$$

where $||\mathcal{P}_{Z\tilde{Z}}^{(n)}||$ is the total number of possible joint types and δ is an arbitrary small number. To simplify the notation, let

$$B^n(Z\tilde{Z}) \triangleq \frac{2^{n(H(Z^{(n)})+H(\tilde{Z}^{(n)}|Z^{(n)})+\epsilon)}(||\mathcal{P}_{Z\tilde{Z}}^{(n)}||+\delta)}{||f||}. \qquad \text{(A.9)}$$

Considering all joint types $P_{Z^{(n)}\tilde{Z}^{(n)}}$ simultaneously, the union bound and (A.8) imply that

$$\begin{aligned}\text{Pr}\left(N_f^n(Z\tilde{Z}) \leq B^n(Z\tilde{Z}),\, \forall P_{Z^{(n)}\tilde{Z}^{(n)}} \in \mathcal{P}_{Z\tilde{Z}}^{(n)}\right) &\geq 1 - \sum_{1}^{||\mathcal{P}_{Z\tilde{Z}}^{(n)}||} \frac{1}{||\mathcal{P}_{Z\tilde{Z}}^{(n)}||+\delta}\\ &> 0. \qquad \text{(A.10)}\end{aligned}$$

Since the probability in (A.10) is positive, then there exists a codebook f^* that the following equation holds for all joint types $P_{Z\tilde{Z}}$ simultaneously

$$N^n_{f^*}(Z\tilde{Z}) \leq \frac{2^{n(H(Z^{(n)})+H(\tilde{Z}^{(n)}|Z^{(n)})+\epsilon)}(||\mathcal{P}^{(n)}_{Z\tilde{Z}}|| + \delta)}{||f^*||}. \tag{A.11}$$

As $||f^*|| = 2^{nR}$ and $||\mathcal{P}^{(n)}_{Z\tilde{Z}}|| \leq (n+1)^4$, we further have

$$\begin{aligned} & N^n_{f^*}(Z\tilde{Z}) \\ & \leq ((n+1)^4 + \delta)\ \ 2^{n(H(Z^{(n)})+H(\tilde{Z}^{(n)}|Z^{(n)})+\epsilon-R)}. \end{aligned} \tag{A.12}$$

In the following, we will focus on f^*.

Let $P^{(n)}_{e,f^*}(Z\tilde{Z})$ denote the portion of error probability associated with a fixed joint type $P_{Z^{(n)}\tilde{Z}^{(n)}}$

$$\begin{aligned} P^{(n)}_{e,f^*}(Z\tilde{Z}) \quad \triangleq & \sum_{z^n \in T_{P_{Z^{(n)}}}} P_\theta(z^n)\mathbf{1}\big(\exists \tilde{z}^n \neq z^n : f^*(\tilde{z}^n) = f^*(z^n), \\ & \text{and} \quad (z^n, \tilde{z}^n) \in T_{P_{Z^{(n)}\tilde{Z}^{(n)}}}\big). \end{aligned}$$

The total decoding error probability $P^{(n)}_{e,f^*}$, when using f^*, can be expressed as

$$P^{(n)}_{e,f^*} = \sum_{P_{Z^{(n)}\tilde{Z}^{(n)}}} P^{(n)}_{e,f^*}(Z\tilde{Z}). \tag{A.13}$$

Let $A^{(n)}_{\epsilon_1}$ denote the set of marginal types $P_{Z^{(n)}}$ such that $|P_{Z^{(n)}}(z=i) - P_\theta(z=i)| < \frac{\epsilon_1}{2}$ for $i \in \{0,1\}$, where ϵ_1 is an arbitrarily small number. Using the definition of $A^{(n)}_{\epsilon_1}$, (A.13) can be rewritten as

$$\begin{aligned} P^{(n)}_{e,f^*} \quad & = \sum_{P_{Z^{(n)}\tilde{Z}^{(n)}}, P_{Z^{(n)}} \in A^{(n)}_{\epsilon_1}} P^{(n)}_{e,f^*}(Z\tilde{Z}) + \sum_{P_{Z^{(n)}\tilde{Z}^{(n)}}, P_{Z^{(n)}} \in \bar{A}^{(n)}_{\epsilon_1}} P^{(n)}_{e,f^*}(Z\tilde{Z}) \\ & \triangleq S_1 + S_2, \end{aligned} \tag{A.14}$$

where $\bar{A}_{\epsilon_1}^{(n)}$ denotes the complimentary set of $A_{\epsilon_1}^{(n)}$. For S_2, we have that

$$P_{e,f^*}^{(n)}(Z\tilde{Z}) \leq 2^{-n(D(P_{Z^{(n)}}||P_\theta(Z)))}, \tag{A.15}$$

where $D(P_{Z^{(n)}}||P_\theta(Z))$ is the KullbackLeibler divergence [57] between the marginal type $P_{Z^{(n)}}$ and the true PMF $P_\theta(Z)$ of $Z = X \oplus Y$. Using Pinsker's inequality [61], for $P_{Z^{(n)}} \in \bar{A}_{\epsilon_1}^{(n)}$, we have

$$D(P_{Z^{(n)}}||P_\theta(Z)) \geq 2\epsilon_1^2. \tag{A.16}$$

Therefore,

$$\begin{aligned} S_2 &\leq \sum_{P_{Z^{(n)}\tilde{Z}^{(n)}}} 2^{-2n\epsilon_1^2} \\ &\leq (n+1)^4 \ 2^{-2n\epsilon_1^2}. \end{aligned} \tag{A.17}$$

(A.17) implies that $S_2 \to 0$ exponentially as $n \to \infty$.

For S_1, we have that

$$P_{e,f^*}^{(n)}(Z\tilde{Z}) \leq N_{f^*}^n(Z\tilde{Z}) \ 2^{-n(H(Z^{(n)})+D(P_{Z^{(n)}}||P_\theta(Z))}. \tag{A.18}$$

Using (A.12), we further have

$$\begin{aligned} &P_{e,f^*}^{(n)}(Z\tilde{Z}) \leq \\ &((n+1)^4+\delta)\, 2^{-n\left(D(P_{Z^{(n)}}||P_\theta(Z))+R-H(\tilde{Z}^{(n)}|Z^{(n)})-\epsilon\right)}. \end{aligned} \tag{A.19}$$

As we use the minimum entropy decoder, we have $H(\tilde{Z}^{(n)}) \leq H(Z^{(n)})$, which implies

$H(\tilde{Z}^{(n)}|Z^{(n)}) \leq H(\tilde{Z}^{(n)}) \leq H(Z^{(n)})$. Therefore,

$$\begin{aligned} & P_{e,f^*}^{(n)}(Z\tilde{Z}) \\ & \leq ((n+1)^4 + \delta)\, 2^{-n\left(D(P_{Z^{(n)}}||P_\theta(Z)) + R - H(Z^{(n)}) - \epsilon\right)}. \end{aligned} \tag{A.20}$$

Since $P_{Z^{(n)}} \in A_{\epsilon_1}^{(n)}$, it is easy to check that

$$|H(Z^{(n)}) - H_\theta(Z)| \leq D(P_{Z^{(n)}}||P_\theta(Z)) + \epsilon_2. \tag{A.21}$$

Here

$$\epsilon_2 = -\frac{\epsilon_1}{2} \sum_i \log P_\theta(z = i), \tag{A.22}$$

which can be made arbitrarily small as $\epsilon_1 \downarrow 0$ for $\theta \in (0, 1)$.

Therefore,

$$P_{e,f^*}^{(n)}(Z\tilde{Z}) \leq ((n+1)^4 + \delta)\, 2^{-n\left(R - H_\theta(Z) - \epsilon_3\right)}, \tag{A.23}$$

in which $\epsilon_3 = \epsilon + \epsilon_2$.

This implies that $S_1 \to 0$ exponentially as $n \to \infty$ if

$$R > H_\theta(Z). \tag{A.24}$$

Therefore, (A.24) is sufficient to guarantee that $P_{e,f^*}^{(n)} \to 0$ exponentially as $n \to \infty$. It is easy to check that $H_\theta(Z) = H_\theta(X|Y) = H_\theta(Y|X)$. The proof is complete.

Appendix B

Detailed Analysis in the Proof of Theorem 2

Optimal Centralized Estimator: First consider the centralized case in which X^n and Y^n are both known perfectly. Let $\left(\frac{n_1}{n}, \frac{n_2}{n}, \frac{n_3}{n}, \frac{n_4}{n}\right)$ denote the joint type of the sequences x^n and y^n, where (n_1, n_2, n_3, n_4) are the frequencies of occurrence of the pairs $(0,0), (1,1), (0,1), (1,0)$, respectively. The joint PMF of (x^n, y^n) is

$$P_\theta(x^n, y^n) \quad = \quad \left(\frac{\theta}{2}\right)^{(n_1+n_2)} \left(\frac{1-\theta}{2}\right)^{(n_3+n_4)}. \tag{B.1}$$

Consider the centralized estimator

$$\hat{\theta}_c = \frac{(n_1 + n_2)}{n}. \tag{B.2}$$

This estimator is unbiased since

$$\mathbb{E}_\theta[\hat{\theta}_c] = \theta. \tag{B.3}$$

The variance of the estimator is calculated as

$$\begin{aligned}\mathrm{Var}_\theta[\hat{\theta}_c] &= \frac{1}{n^2}\mathbb{E}_\theta[(n_1+n_2)^2]-\theta^2 \\ &= \frac{\theta(1-\theta)}{n}.\end{aligned} \tag{B.4}$$

The variance index is given by

$$V_\theta[\hat{\theta}_c] = \lim_{n\to\infty} n\mathrm{Var}_\theta[\hat{\theta}_c] = \theta(1-\theta). \tag{B.5}$$

The Cramer-Rao lower bound (CRLB) of the centralized case is

$$\begin{aligned}\mathrm{CRLB} &= -1/\mathbb{E}_\theta\left[\frac{\partial^2 \ln[P_\theta(x^n,y^n)]}{\partial^2\theta}\right] \\ &= \frac{\theta(1-\theta)}{n} = \mathrm{Var}_\theta[\hat{\theta}_c].\end{aligned} \tag{B.6}$$

This implies that $\hat{\theta}_c$ is an MVUE for the centralized case.

Comparison: Now, come back to our decentralized case, for which our estimator is

$$\hat{\theta} = \frac{n(0|\hat{Z}^n)}{n}. \tag{B.7}$$

We will compare the performance of $\hat{\theta}$ with that of the optimal centralized estimator $\hat{\theta}_c$.

For the codebook f^*, define $T_e^{(n)}$ as the set of sequences z^ns that are incorrectly decoded. Therefore,

$$P_{e,f^*}^{(n)} = \sum_{z^n\in T_e^{(n)}} P_\theta(z^n). \tag{B.8}$$

The expected value of our estimator is given by

$$\mathbb{E}_\theta[\hat{\theta}] = \sum_{z^n \in \{0,1\}^n} \Pr(\hat{Z}^n = z^n) \frac{n(0|z^n)}{n}. \tag{B.9}$$

Note that $\Pr(\hat{Z}^n = z^n)$ is not necessarily equal to $P_\theta(z^n)$, and the sum of the probability difference can be bounded as

$$\sum_{z^n \in \{0,1\}^n} |\Pr(\hat{Z}^n = z^n) - P_\theta(Z^n = z^n)| \leq 2 \sum_{z^n \in T_e^{(n)}} P_\theta(z^n) = 2P_{e,f^*}^{(n)}. \tag{B.10}$$

We have that

$$|\mathbb{E}_\theta[\hat{\theta}] - \mathbb{E}_\theta[\hat{\theta}_c]| \leq \sum_{z^n \in \{0,1\}^n} |\Pr(\hat{Z}^n = z^n) - P_\theta(Z^n = z^n)| \frac{n(0|z^n)}{n}.$$

Since

$$0 \leq \frac{n(0|z^n)}{n} \leq 1, \tag{B.11}$$

then

$$\begin{aligned} |\mathbb{E}_\theta[\hat{\theta}] - \mathbb{E}_\theta[\hat{\theta}_c]| &\leq \sum_{z^n \in \{0,1\}^n} |\Pr(\hat{Z}^n = z^n) - P_\theta(Z^n = z^n)| \\ &\leq 2P_{e,f^*}^{(n)}, \end{aligned} \tag{B.12}$$

in which the last inequality is due to (B.10).

As $P_{e,f^*}^{(n)}$ is shown to converge to zero exponentially fast in Section 2.2.1, we have

$$\lim_{n\to\infty} \mathbb{E}_\theta[\hat{\theta}] = \mathbb{E}_\theta[\hat{\theta}_c] = \theta. \tag{B.13}$$

This shows that our estimator is asymptotically unbiased. Similarly, we have

$$\begin{aligned} |\text{Var}_\theta[\hat{\theta}] - \text{Var}_\theta[\hat{\theta}_c]| &\leq |\mathbb{E}_\theta[\hat{\theta}^2] - \mathbb{E}_\theta[\hat{\theta}_c^2]| + |\mathbb{E}_\theta[\hat{\theta}] - \mathbb{E}_\theta[\hat{\theta}_c]| \\ &\leq 4P_{e,f^*}^{(n)}. \end{aligned} \tag{B.14}$$

Hence,

$$|V_\theta[\hat{\theta}] - V_\theta[\hat{\theta}_c]| \leq \lim_{n\to\infty} 4nP_{e,f^*}^{(n)}. \tag{B.15}$$

As $n \to \infty$, $P_{e,f^*}^{(n)} \to 0$ exponentially, we have $4nP_{e,f^*}^{(n)} \to 0$. Therefore,

$$V_\theta[\hat{\theta}] = V_\theta[\hat{\theta}_c] = \theta(1-\theta). \tag{B.16}$$

This proves that our estimator is asymptotically unbiased and achieves the same minimum variance that can be achieved even in the centralized case. Hence, our estimator is optimal. The proof is complete.

Appendix C

Detailed Analysis in the Proof of Theorem 3

Step 1: Computing a Sufficient Statistic

Different from the binary symmetric case considered in Section 2.2, Z^n is not a sufficient statistic for the general binary case anymore. Now, we show the joint type $P_{X^{(n)}Y^{(n)}} = \left(\frac{n_1}{n}, \frac{n_2}{n}, \frac{n_3}{n}, \frac{n_4}{n}\right)$ of the observation sequences (x^n, y^n) is a sufficient statistic and show how to compute this statistics at the fusion center using rates (2.18).

Let $T_{P_{X^{(n)}Y^{(n)}}}$ be the set of all sequence pairs (x^n, y^n) that have the joint type $P_{X^{(n)}Y^{(n)}}$. The conditional PMF of (X^n, Y^n) given the joint type $P_{X^{(n)}Y^{(n)}}$ is

$$P_\theta(x^n, y^n | P_{X^{(n)}Y^{(n)}}) = \begin{cases} 0, \text{ if } (x^n, y^n) \notin T_{P_{X^{(n)}Y^{(n)}}} \\ \frac{1}{||T_{P_{X^{(n)}Y^{(n)}}}||}, \quad \text{otherwise} \end{cases}, \tag{C.1}$$

which is not a function of θ. Therefore, the joint type $P_{X^{(n)}Y^{(n)}}$ is a sufficient statistic of θ.

Now we show how to compute this statistic at the fusion center with rates in (2.18).

Encoding: At terminals A and B, we first encode X^n and Y^n using the same scheme presented in Section 2.2.1. This will enable terminal C to compute Z^n. In addition, each terminal will send the marginal types $P_{X^{(n)}} \in \mathcal{P}_X^{(n)}$ and $P_{Y^{(n)}} \in \mathcal{P}_Y^{(n)}$ of the sequences x^n and y^n, respectively. The number of marginal types can be bounded as [57]

$$||\mathcal{P}_X^{(n)}|| = ||\mathcal{P}_Y^{(n)}|| \leq (n+1)^2. \tag{C.2}$$

Therefore each of the marginal types can be encoded using the rate $\frac{2\log(n+1)}{n}$, which goes to zero as n increases. Hence, sending these additional information requires diminishing additional rates.

Decoding: At terminal C, we first decode $\hat{Z}^n$ using the same scheme as discussed in Section 2.2.1. Once $\hat{Z}^n$ is decoded, terminal C will compute the joint type $\hat{P}_{X^{(n)}Y^{(n)}} = \left(\frac{\hat{n}_1}{n}, \frac{\hat{n}_2}{n}, \frac{\hat{n}_3}{n}, \frac{\hat{n}_4}{n}\right)$ by combining $\hat{Z}^n$ along with the additional information $P_{X^{(n)}}$, $P_{Y^{(n)}}$ sent from terminals A and B respectively. In particular, from these information, we have the following relationship

$$\begin{aligned} \hat{n}_1 + \hat{n}_2 &= n(0|\hat{Z}^n), && \text{(C.3)}\\ \frac{\hat{n}_1 + \hat{n}_3}{n} &= P_{X^{(n)}}(x=0), && \text{(C.4)}\\ \frac{\hat{n}_1 + \hat{n}_4}{n} &= P_{Y^{(n)}}(y=0), && \text{(C.5)}\\ \sum_{i=1}^{4} \hat{n}_i &= n. && \text{(C.6)} \end{aligned}$$

From these four equations, we can easily obtain $\hat{P}_{X^{(n)}Y^{(n)}}$.

Error Probability: Define $P_e^{(n)}$ as

$$P_e^{(n)} = \Pr(\hat{P}_{X^{(n)}Y^{(n)}} \neq P_{X^{(n)}Y^{(n)}}). \tag{C.7}$$

As shown in Section 2.2.1, Z^n can be decoded at the rates given in (2.18) with an exponentially decreasing probability of error. Furthermore, the marginal types can be perfectly recovered at asymptotically zero rates, then the joint type $P_{X^{(n)}Y^{(n)}}$ can be computed with an exponentially decreasing error probability $P_e^{(n)}$.

Step 2: Estimation

In the binary symmetric case considered in Section 2.2, we have MVUE for the centralized case and hence we can compare our distributed estimator with this centralized MVUE. In the general binary model, this approach will not work as we don't know whether or not an MVUE exists. Furthermore, even if it exists, the form of MVUE is model specific. In the following, we show a stronger result that, for any given centralized estimator, we can construct an estimator that achieves the same variance index. This implies that, if the minimum variance unbiased estimator (MVUE) exists in the centralized case, we can construct a distributed estimator that achieves the same variance index. Furthermore, even if MVUE does not exist in the centralized case, we can still construct a distributed estimator that has the same performance as that of the best estimator in the centralized case.

First, as $P_{X^{(n)}Y^{(n)}}$ is a sufficient statistic for the centralized case, by Rao-Blackwell theorem [56], if we want to minimize the variance of unbiased estimators, we can focus on estimators that are functions of $P_{X^{(n)}Y^{(n)}}$, namely $F_c = F(P_{X^{(n)}Y^{(n)}})$, for the centralized case. For any unbiased F_c, we design the following simple plugin estimator

$$\hat{F} \quad = \quad F(\hat{P}_{X^{(n)}Y^{(n)}}). \tag{C.8}$$

In the following, we compare the performance of F_c and $\hat{F}$. We have that

$$\mathbb{E}_\theta[\hat{F}] = \sum_{P_{x^{(n)}y^{(n)}}} \Pr(\hat{P}_{X^{(n)}Y^{(n)}} = P_{x^{(n)}y^{(n)}})F(P_{x^{(n)}y^{(n)}}), \tag{C.9}$$

and

$$|\mathbb{E}_\theta[\hat{F}] - \mathbb{E}_\theta[F_c]| \leq \sum_{P_{x^{(n)}y^{(n)}}} |\text{Pr}(\hat{P}_{X^{(n)}Y^{(n)}} = P_{x^{(n)}y^{(n)}}) - P_\theta(P_{x^{(n)}y^{(n)}})| \cdot |F(P_{x^{(n)}y^{(n)}})|. \tag{C.10}$$

Since $F(P_{x^{(n)}y^{(n)}}) \in \Theta$ and Θ is bounded, we have $|F(P_{x^{(n)}y^{(n)}})| \leq \theta_u$. Furthermore, following similar steps as that of (B.10), we have

$$\sum_{P_{x^{(n)}y^{(n)}}} |\text{Pr}(\hat{P}_{X^{(n)}Y^{(n)}} = P_{x^{(n)}y^{(n)}}) - P_\theta(P_{x^{(n)}y^{(n)}})| \leq 2P_e^{(n)}.$$

As the result, we have

$$|\mathbb{E}_\theta[\hat{F}] - \mathbb{E}_\theta[F_c]| \leq 2P_e^{(n)}\theta_u, \tag{C.11}$$

hence

$$\lim_{n\to\infty} \mathbb{E}_\theta[\hat{F}] = \mathbb{E}_\theta[F_c] = \theta, \tag{C.12}$$

as $P_e^{(n)}$ goes to zero exponentially. Similarly,

$$|\text{Var}_\theta[\hat{F}] - \text{Var}_\theta[F_c]| \leq 2P_e^{(n)}(\theta_u^2 + \theta_u). \tag{C.13}$$

Therefore,

$$V_\theta[\hat{F}] = V_\theta[F_c]. \tag{C.14}$$

This implies that the plugin distributed estimator $\hat{F}$ achieves the same performance as that of the centralized estimator F_c if the rate condition (2.18) is satisfied. The proof is complete.

Appendix D

The Proof of Theorem 4

Case 1: $R \geq H(t)$

$$\begin{aligned} R &\geq H(t) \\ &> H_\theta(Z) = H_\theta(X|Y) = H_\theta(Y|X). \end{aligned} \tag{D.1}$$

In this case, $p = n$, and our estimator is given by

$$\hat{\theta} = \frac{n(0|\hat{Z}^n)}{n}. \tag{D.2}$$

As we proved in the previous sections, this estimator is an asymptotically MVUE if $R > H_\theta(Z)$.

Case 2: $R < H(t)$

In the centralized case, consider the estimator

$$\hat{\theta}_c = \frac{(n_1 + n_2)}{p}, \tag{D.3}$$

where n_1 and n_2 are the frequency of occurrence of the pairs $(0, 0)$ and $(1, 1)$ in the observations (x^p, y^p), respectively. We have that

$$\mathbb{E}_\theta[\hat{\theta}_c] = \frac{p\theta}{p} = \theta, \tag{D.4}$$

and

$$\text{Var}_\theta[\hat{\theta}_c] = \frac{\theta(1-\theta)}{p}. \tag{D.5}$$

In the decentralized case, the effective rate per observation is given by

$$R_{eff} = \frac{nR}{p}. \tag{D.6}$$

Since

$$p \leq \frac{nR}{H(t)}, \tag{D.7}$$

then

$$R_{eff} \geq H(t) > H_\theta(Z). \tag{D.8}$$

For this range of rates, we showed that

$$\lim_{n\to\infty} \mathbb{E}_\theta[\hat{\theta}] = \mathbb{E}_\theta[\hat{\theta}_c] = \theta. \tag{D.9}$$

Therefore, our estimator is asymptotically unbiased. We also have that

$$\begin{aligned} V_\theta[\hat{\theta}] &= \lim_{n\to\infty} n\text{Var}_\theta[\hat{\theta}_c] \\ &= \lim_{n\to\infty} \frac{n\theta(1-\theta)}{p}. \end{aligned} \tag{D.10}$$

It is obvious that

$$p \geq \frac{nR}{H(t)} - 1. \tag{D.11}$$

Hence,

$$V_\theta[\hat{\theta}] \leq \frac{H(t)\theta(1-\theta)}{R}. \tag{D.12}$$

The proof is complete.

Appendix E

The Remaining of the Proof of Theorem 7

We also have that

$$\begin{aligned}
||\mathbf{x}_j^q - \mathbf{x}^*||^2 &= ||\mathbf{x}_j - \mathbf{x}^* + td\mathbf{n}_j||^2 \\
&= ||\mathbf{x}_j - \mathbf{x}^*||^2 + t^2d^2||\mathbf{n}_j||^2 \\
&+ 2td\langle \mathbf{x}_j - \mathbf{x}^*, \mathbf{n}_j\rangle \\
&\leq ||\mathbf{x}_j - \mathbf{x}^*||^2 + t^2d^2||\mathbf{n}_j||^2 \\
&+ 2td||\mathbf{x}_j - \mathbf{x}^*||||\mathbf{n}_j|| \\
&\leq ||\mathbf{x}_j - \mathbf{x}^*||^2 + td\Delta||\mathbf{x}_j - \mathbf{x}^*|| \\
&+ \frac{t^2d^2\Delta^2}{4}, \qquad \text{(E.1)}
\end{aligned}$$

where the first inequality follows from Cauchy-Schwarz inequality, and the second inequality follows from (4.6).

To proceed with the convergence analysis, we have two different cases.

Case 1 ($||\mathbf{x}_0 - \mathbf{x}^*|| \leq 1$):

In this case, $\mathbb{E}||\mathbf{x}_j - \mathbf{x}^*|| \leq 1$. Therefore,

$$\mathbb{E}||\mathbf{x}_{j+1} - \mathbf{x}^*||^2 \leq C\mathbb{E}||\mathbf{x}_j - \mathbf{x}^*||^2 + Ctd\Delta(1 + \frac{td\Delta}{4}). \tag{E.2}$$

Let k_1 denotes the minimum number of iterations required to achieve the convergence condition. Hence,

$$\begin{aligned} \mathbb{E}||\mathbf{x}_{k_1} - \mathbf{x}^*||^2 &\leq C^{k_1}||\mathbf{x}_0 - \mathbf{x}^*||^2 \\ &+ Ctd\Delta(1 + \frac{td\Delta}{4})(1 + C + .. + C^{k_1 - 1}). \end{aligned} \tag{E.3}$$

Since $C < 1$, then

$$\begin{aligned} \mathbb{E}||\mathbf{x}_{k_1} - \mathbf{x}^*||^2 &\leq C^{k_1}||\mathbf{x}_0 - \mathbf{x}^*||^2 \\ &+ \frac{C}{1 - C}td\Delta(1 + \frac{td\Delta}{4}). \end{aligned} \tag{E.4}$$

For the algorithm to converge, let

$$C^{k_1}||\mathbf{x}_0 - \mathbf{x}^*||^2 \leq \frac{\epsilon\rho}{2}, \tag{E.5}$$

and

$$\frac{C}{1 - C}td\Delta(1 + \frac{td\Delta}{4}) \leq \frac{\epsilon\rho}{2}, \tag{E.6}$$

Case 2 ($||\mathbf{x}_0 - \mathbf{x}^*|| > 1$):

Let k_2 denotes the minimum number of iterations required such that $\mathbb{E}||\mathbf{x}_{k_2} - \mathbf{x}^*|| \leq 1$.

For all $j \leq k_2$, we have that $\mathbb{E}||\mathbf{x}_j - \mathbf{x}^*|| \leq \mathbb{E}||\mathbf{x}_j - \mathbf{x}^*||^2$. Therefore,

$$\begin{aligned} \mathbb{E}||\mathbf{x}_{j+1} - \mathbf{x}^*||^2 &\leq C(1 + td\Delta)\mathbb{E}||\mathbf{x}_j - \mathbf{x}^*||^2 \\ &+ \frac{Ct^2d^2\Delta^2}{4}. \end{aligned} \tag{E.7}$$

After k_2 iterations, we have that

$$\begin{aligned} \mathbb{E}||\mathbf{x}_{k_2} - \mathbf{x}^*||^2 &\leq (C(1 + td\Delta))^{k_2}||\mathbf{x}_0 - \mathbf{x}^*||^2 \\ &+ \frac{Ct^2d^2\Delta^2}{4(1-C)}. \end{aligned} \tag{E.8}$$

For the algorithm to converge, let

$$(C(1 + td\Delta))^{k_2}||\mathbf{x}_0 - \mathbf{x}^*||^2 \leq \frac{1}{2}, \tag{E.9}$$

and

$$\frac{Ct^2d^2\Delta^2}{4(1-C)} \leq \frac{1}{2}. \tag{E.10}$$

Finally, the total number of iterations required for convergence is given by

$$k^q = k_1 + k_2. \tag{E.11}$$

To achieve the fastest convergence rate, the step size t is chosen to minimize C. Hence,

$$t_{opt} = \frac{1}{gLd} \text{ ,and } C_{min} = 1 - \frac{1}{g^2d} \tag{E.12}$$

From (E.6) and (E.10), a sufficient condition on the quantization error is given by

$$\Delta \leq \frac{\epsilon \rho L^2}{2m} (\frac{1}{C_{min}} - 1). \tag{E.13}$$

From (E.5), (E.9), and (4.12), the number of iterations required for the algorithm to converge is at most

$$\begin{aligned} k^q &= \frac{\log(2||\mathbf{x}_0 - \mathbf{x}^*||^2/\epsilon\rho)}{\log(1/C_{min})} \\ &+ \frac{\log(2||\mathbf{x}_0 - \mathbf{x}^*||^2)}{\log(1/(C_{min} + \frac{\epsilon\rho}{2}(1 - C_{min}))}. \end{aligned} \tag{E.14}$$

The proof is complete.

Appendix F

The Proof of Corollary 1

Similar to our analysis in Section 4.3, we have that $\mathbb{E}_{s_{j+1}}[\triangledown f(\mathbf{x}_j^q)]_{I_{s_{j+1}}} = \frac{1}{M}(\triangledown f(\mathbf{x}_j^q))$. Therefore,

$$\mathbb{E}||\mathbf{x}_{j+1} - \mathbf{x}^*||^2 \quad \leq \quad C_1\mathbb{E}||\mathbf{x}_j^q - \mathbf{x}^*||^2. \tag{F.1}$$

where $C_1 = t^2L^2M - 2tm + 1$. We also have that

$$\begin{aligned} ||\mathbf{x}_j^q - \mathbf{x}^*||^2 &\leq ||\mathbf{x}_j - \mathbf{x}^*||^2 + t^2M^2||\mathbf{n}_j||^2 \\ &+ 2tm||\mathbf{x}_j - \mathbf{x}^*||||\mathbf{n}_j|| \\ &\leq ||\mathbf{x}_j - \mathbf{x}^*||^2 + tm\Delta\sqrt{l_m}||\mathbf{x}_j - \mathbf{x}^*|| \\ &+ \frac{t^2M^2\Delta^2 l_m}{4}, \end{aligned} \tag{F.2}$$

where the second inequality follows from (4.23).

Following the same steps of our analysis in Section 4.3, we get a sufficient condition on the quantization error to guarantee the convergence of the algorithm that is given by

$$\Delta \leq \frac{\epsilon\rho L^2}{2m\sqrt{l_m}}(\frac{1}{C_{1_{min}}} - 1), \tag{F.3}$$

where $C_{1_{min}} = 1 - (1/(g^2 M))$ for a step size $t_{opt} = 1/(gLM)$. This condition shows that for a fixed number of nodes M, the upper bound on the quantization error is tighter for a larger value of l_m. We also get that the number of iterations required for convergence is at most

$$\begin{aligned} k^q &= \frac{\log(2||\mathbf{x}_0 - \mathbf{x}^*||^2/\epsilon\rho)}{\log(1/C_{1_{min}})} \\ &+ \frac{\log(2||\mathbf{x}_0 - \mathbf{x}^*||^2)}{\log(1/(C_{1_{min}} + \frac{\epsilon\rho}{2}(1 - C_{1_{min}}))}. \end{aligned} \tag{F.4}$$

The proof is complete.

Appendix G

The Proof of Corollary 2

We have that

$$E_{s_{j+1}}[\nabla f(\mathbf{x}_j^q)]_{Q_{s_{j+1}}} = \begin{bmatrix} w_1 \frac{\partial f(\mathbf{x}_j^q)}{\partial x_1} \\ w_2 \frac{\partial f(\mathbf{x}_j^q)}{\partial x_2} \\ . \\ . \\ w_d \frac{\partial f(\mathbf{x}_j^q)}{\partial x_1}, \end{bmatrix} \tag{G.1}$$

where $w_j = q_i/Ml_i$ if coordinate j is updated by node i. Since

$$\frac{q}{Ml_m} \le w_j \le \frac{1}{M}, \;\; j \in \{1, 2, ..., d\}, \tag{G.2}$$

then

$$\begin{aligned} \frac{q}{Ml_m} ||(\nabla f(\mathbf{x}_j^q))|| &\le \mathbb{E}_{s_{j+1}} ||[\nabla f(\mathbf{x}_j^q)]_{Q_{s_{j+1}}}|| \\ &\le \frac{1}{M} ||(\nabla f(\mathbf{x}_j^q))||. \end{aligned} \tag{G.3}$$

Similar to our analysis in Section 4.3, we get that

$$\mathbb{E}||\mathbf{x}_{j+1} - \mathbf{x}^*||^2 \;\; \leq \;\; C_2\mathbb{E}||\mathbf{x}_j^q - \mathbf{x}^*||^2. \tag{G.4}$$

where $C_2 = t^2L^2M - \frac{2tmq}{l_m} + 1$. We also get that

$$\begin{aligned} ||\mathbf{x}_j^q - \mathbf{x}^*||^2 \;\; & \leq \;\; ||\mathbf{x}_j - \mathbf{x}^*||^2 + t^2M^2||\mathbf{n}_j||^2 \\ & + \;\; 2tM||\mathbf{x}_j - \mathbf{x}^*||||\mathbf{n}_j|| \\ & \leq \;\; ||\mathbf{x}_j - \mathbf{x}^*||^2 + tM\Delta\sqrt{q}||\mathbf{x}_j - \mathbf{x}^*|| \\ & + \;\; \frac{t^2M^2\Delta^2q}{4}, \end{aligned} \tag{G.5}$$

where the second inequality follows from (4.23) and that the maximum number of updated coordinates in a single iteration is q.

Following the same steps of our analysis in Section 4.3, the sufficient condition on the quantization error is given by

$$\Delta \leq \frac{\epsilon\rho L^2 l_m}{2mq\sqrt{q}}(\frac{1}{C_{2_{min}}} - 1), \tag{G.6}$$

where $C_{2_{min}} = 1 - (q^2/({l_m}^2g^2M))$ for a step size $t_{opt} = q/(l_mgLM)$. The effect of the parameter q on the quantization condition is discussed in Section 2.7. We also get that the number of iterations required for convergence is at most

$$\begin{aligned} k^q \;\; & = \;\; \frac{\log(2||\mathbf{x}_0 - \mathbf{x}^*||^2/\epsilon\rho)}{\log(1/C_{2_{min}})} \\ & + \;\; \frac{\log(2||\mathbf{x}_0 - \mathbf{x}^*||^2)}{\log(1/(C_{2_{min}} + \frac{\epsilon\rho}{2}(1 - C_{2_{min}}))}. \end{aligned} \tag{G.7}$$

The proof is complete.

Appendix H

The Proof of Corollary 3

Let R_{j+1} denotes the set of updated coordinates at iteration $j+1$. Hence

$$R_{j+1} = \bigcup_{i=1}^{M} Q_{j,i} \tag{H.1}$$

We have that

$$\mathbf{x}_{j+1} = \mathbf{x}_j^q - t[\triangledown f(\mathbf{x}_j^q)]_{R_{j+1}}. \tag{H.2}$$

The expected value $E_{j+1}[\triangledown f(\mathbf{x}_j^q)]_{R_{j+1}}$ can be expressed as in G.1 with

$$\frac{q}{l_m} \leq w_j \leq 1, \;\; j \in \{1, 2, ..., d\}. \tag{H.3}$$

Hence,

$$\begin{aligned} \frac{q}{l_m} ||(\triangledown f(\mathbf{x}_j^q))|| \;\; &\leq \;\; \mathbb{E}_{j+1} ||[\triangledown f(\mathbf{x}_j^q)]_{R_{j+1}}|| \\ &\leq \;\; ||(\triangledown f(\mathbf{x}_j^q))||. \end{aligned} \tag{H.4}$$

Similar to our analysis in the previous sections, we get that

$$\mathbb{E}||\mathbf{x}_{j+1} - \mathbf{x}^*||^2 \;\; \leq \;\; C_s \mathbb{E}||\mathbf{x}_j^q - \mathbf{x}^*||^2. \tag{H.5}$$

where $C_s = t^2L^2 - \frac{2tmq}{l_m} + 1$. We also get that

$$\begin{aligned} ||\mathbf{x}_j^q - \mathbf{x}^*||^2 &\leq ||\mathbf{x}_j - \mathbf{x}^*||^2 + t^2||\mathbf{n}_j||^2 \\ &+ 2t||\mathbf{x}_j - \mathbf{x}^*||||\mathbf{n}_j|| \\ &\leq ||\mathbf{x}_j - \mathbf{x}^*||^2 + t\Delta\sqrt{r}||\mathbf{x}_j - \mathbf{x}^*|| + \frac{t^2\Delta^2 r}{4}, \end{aligned} \tag{H.6}$$

The sufficient condition on the quantization error is given by

$$\Delta \leq \frac{\epsilon\rho L^2 l_m}{2mq\sqrt{r}}(\frac{1}{C_{s_{min}}} - 1), \tag{H.7}$$

where $C_{s_{min}} = 1 - (q^2/({l_m}^2 g^2))$ for a step size $t_{opt} = q/(l_m g L)$. The number of iterations required for convergence is at most

$$\begin{aligned} k^q &= \frac{\log(2||\mathbf{x}_0 - \mathbf{x}^*||^2/\epsilon\rho)}{\log(1/C_{s_{min}})} \\ &+ \frac{\log(2||\mathbf{x}_0 - \mathbf{x}^*||^2)}{\log(1/(C_{s_{min}} + \frac{\epsilon\rho}{2}(1 - C_{s_{min}}))}. \end{aligned} \tag{H.8}$$

The proof is complete.

Bibliography

[1] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Trans. Automatic Control*, vol. 57, pp. 592–606, Mar. 2012.

[2] Y. Zhang, J. Duchi, and M. Wainwright, "Communication-efficient algorithms for statistical optimization," *Journal of Machine Learning Research*, vol. 14, pp. 3321–3363, Nov. 2013.

[3] R. Kannan, S. Vempala, and D. Woodruff, "Principal component analysis and higher correlations for distributed data," *In Proceedings of the 27th Conference on Learning Theory, (Barcelona, Spain)*, pp. 1040–1057, Jun. 2014.

[4] A. Kleiner, A. Talwalkar, P. Sarkar, and M. I. Jordan, "A scalable bootstrap for massive data," *Journal of the Royal Statistical Society*, vol. 76, pp. 795–816, Mar. 2014.

[5] O. Shamir, N. Srebro, and T. Zhang, "Communication-efficient distributed optimization using an approximate newton-type method," *In Proceedings of the International Conference on Machine Learning, (Beijing, China)*, pp. 1000–1008, Jun. 2014.

[6] L. Mackey, A. Talwalkar, and M. I. Jordan, "Distributed matrix completion and robust factorization," *Journal of Machine Learning Research*, vol. 16, pp. 913–960, Jan. 2015.

[7] Y. Zhang and X. Lin, “Communication efficient distributed optimization of self-concordant empirical loss,” *arXiv:1501.00263*, 2015.

[8] J. Lee, Y. Sun, Q. Liu, and J. Taylor, “Communication efficient sparse regression: a one-shot approach,” *arXiv:1503.04337*, 2015.

[9] M. Suchard, Q. Wang, C. Chan, J. Frelinger, M. Cron, and M. West, “Understanding GPU programming for statistical computation: Studies in massively parallel massive mixtures,” *Journal of Computational and Graphical Statistics*, vol. 19, pp. 419–438, Feb. 2010.

[10] W. Cleveland and R. Hafen, “Divide and recombine: Data science for large complex data,” *Statistical Analysis and Data Mining*, vol. 7, pp. 425–433, Nov. 2014.

[11] D. Maclaurin and R. Adams, “Firey Monte Carlo: Exact MCMC with subsets of data,” *arXiv:1403.5693*, 2014.

[12] X. Wang and D. Dunson, “Parallelizing MCMC via weierstrass sampler,” *arXiv:1312.4605*, 2015.

[13] W. Neiswanger, C. Wang, and E. Xing, “Asymptotically exact, embarrassingly parallel MCMC,” *arXiv:1311.4780*, 2015.

[14] M. Rabinovich, E. Angelino, and M. Jordan, “Variational consensus Monte Carlo,” in *Advances in Neural Information Processing Systems*, (Montreal, Canada), pp. 1207–1215, Dec. 2015.

[15] S. Scott, A. Blocker, F. Bonassi, H. Chipman, E. George, and R. McCulloch, “Bayes and big data: the consensus Monte Carlo algorithm,” *International Journal of Management Science and Engineering Management*, vol. 11, pp. 78–88, Feb. 2016.

[16] A. Terenin, D. Simpson, and D. Draper, “Asynchronous Gibbs sampling,” *arXiv:1509.08999*, 2016.

[17] G. Xu, S. Zhu, and B. Chen, “Decentralized data reduction with quantization constraints,” *IEEE Trans. Signal Processing*, vol. 62, pp. 1775–1784, Apr. 2014.

[18] A. Vempaty, H. He, B. Chen, and P. K. Varshney, “On quantizer design for distributed Bayesian estimation in sensor networks,” *IEEE Trans. Signal Processing*, vol. 62, pp. 5359–5369, Oct. 2014.

[19] J. Zhu, X. Lin, R. S. Blum, and Y. Gu, “Parameter estimation from quantized observations in multiplicative noise environments,” *IEEE Trans. Signal Processing*, vol. 63, pp. 4037–4050, Aug. 2015.

[20] J. Zhang, R. S. Blum, X. Lu, and D. Conus, “Asymptotically optimum distributed estimation in the presence of attacks,” *IEEE Trans. Signal Processing*, vol. 63, pp. 1086–1101, Mar. 2015.

[21] S. Kar, H. Chen, and P. K. Varshney, “Optimal identical binary quantizer design for distributed estimation,” *IEEE Trans. Signal Processing*, vol. 60, pp. 3896–3901, Jul. 2012.

[22] J. Fang and H. Li, “Optimal/near-optimal dimensionality reduction for distributed estimation in homogeneous and certain inhomogeneous scenarios,” *IEEE Trans. Signal Processing*, vol. 58, pp. 4339–4353, Aug. 2010.

[23] F. S. Cattivelli and A. H. Sayed, “Diffusion LMS strategies for distributed estimation,” *IEEE Trans. Signal Processing*, vol. 58, pp. 1035–1048, Oct. 2009.

[24] S. Cui, J.-J. Xiao, A. J. Goldsmith, Z.-Q. Luo, and H. V. Poor, "Estimation diversity and energy efficiency in distributed sensing," *IEEE Trans. Signal Processing*, vol. 55, pp. 4683–4695, Sep. 2007.

[25] M. Raginsky, "Learning from compressed observations," in *Proc. IEEE Inform. Theory Workshop*, (Tahoe City, CA), pp. 420–425, Sep. 2007.

[26] A. Xu and M. Raginsky, "Converses for distributed estimation via strong data processing inequalities," in *Proc. IEEE Intl. Symposium on Inform. Theory*, (Hong Kong, China), Jun. 2015.

[27] Y. Zhang, J. Duchi, M. I. Jordan, and M. J. Wainwright, "Information-theoretic lower bounds for distributed statistical estimation with communication constraints," in *Advances in Neural Information Processing Systems*, (Stateline, NV), pp. 2328–2336, Dec. 2013.

[28] O. Shamir and N. Srebro, "Distributed stochastic optimization and learning," in *Proc. Allerton Conf. on Communication, Control, and Computing*, (Monticello, IL), pp. 850–857, Oct. 2014.

[29] J.-J. Xiao, S. Cui, Z.-Q. Luo, and A. J. Goldsmith, "Power scheduling of universal decentralized estimation in sensor networks," *IEEE Trans. Signal Processing*, vol. 54, pp. 413–422, Feb. 2006.

[30] A. Ribeiro and G. B. Giannakis, "Bandwidth-constrained distributed estimation for wireless sensor networks, part I: Gaussian case," *IEEE Trans. Signal Processing*, vol. 54, pp. 1131–1143, Mar. 2006.

[31] A. Ribeiro and G. B. Giannakis, "Bandwidth-constrained distributed estimation for wireless sensor networks, part II: Unknown probability density function," *IEEE Trans. Signal Processing*, vol. 54, pp. 2784–2796, Jul. 2006.

[32] J. Li and G. AlRegib, “Rate-constrained distributed estimation in wireless sensor networks,” *IEEE Trans. Signal Processing*, vol. 55, pp. 1634–1643, May. 2007.

[33] D. Slepian and J. K. Wolf, “Noiseless coding of correlated information sources,” *IEEE Trans. Inform. Theory*, vol. 19, pp. 471–480, Nov. 1973.

[34] I. Csiszár, “Linear codes for sources and source networks: Error exponents, universal coding,” *IEEE Trans. Inform. Theory*, vol. 28, pp. 585–592, Jul. 1982.

[35] A. Zia, J. P. Reilly, and S. Shirani, “Distributed estimation; three theorems,” in *Proc. IEEE Inform. Theory Workshop*, (Tahoe City, CA), pp. 517–522, Sep. 2007.

[36] T. S. Han and S.-I. Amari, “Parameter estimation with multiterminal data compression,” *IEEE Trans. Inform. Theory*, vol. 41, pp. 1802–1833, Nov. 1995.

[37] R. Ahlswede and I. Csiszár, “To get a bit of information may be as hard as to get full information,” *IEEE Trans. Inform. Theory*, vol. 27, pp. 398–408, Jul. 1981.

[38] Z. Zhang and T. Berger, “Estimation via compressed information,” *IEEE Trans. Inform. Theory*, vol. 34, pp. 198–211, Mar. 1988.

[39] W. M. Lam and A. R. Reibman, “Design of quantizers for decentralized estimation systems,” *IEEE Trans. Communications*, vol. 41, pp. 1602–1605, Apr. 1993.

[40] J. A. Gubner, “Distributed estimation and quantization,” *IEEE Trans. Inform. Theory*, vol. 39, pp. 1456–1459, Jul. 1993.

[41] V. N. Vapnik, *Statistical learning theory.* New York: Wiley, 1998.

[42] R. A. Fisher, “On the mathematical foundations of theoretical statistics,” *Philosophical Transactions of The Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 222, no. 5, pp. 309–368, Jan. 1922.

[43] G. Casella and R. L. Berger, *Statistical Inference*. Belmont, CA: Duxbury, 1990.

[44] E. L. Lehmann and G. Casella, *Thoery of Point Estimation*. New York: Springer, 2nd ed., 1998.

[45] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*. New York: Wiley, 1965.

[46] M. Jaggi, V. Smith, M. Takáč, J. Terhorst, S. Krishnan, T. Hofmann, and M. I. Jordan, *Coordinate descent algorithms*. In Advances in Neural Information Processing Systems, (Montreal, Canada), pp. 3068–3076, Dec. 2014.

[47] Y. Zhang and L. Xiao, *DiSCO: Distributed optimization for self-concordant empirical loss*. In International Conference on Machine Learning, (Lille, France), pp. 362–370, Jul. 2015.

[48] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Foundations and Trends in Machine Learning, vol. 3, pp. 1–122, Jan. 2011.

[49] W. Deng and W. Yin, *On the global and linear convergence of the generalized alternating direction method of multipliers*. Journal of Scientific Computing, vol. 66,pp. 1–28, Mar. 2016.

[50] S. Zhu, M. Hong, and B. Chen, "Quantized consensus ADMM for multi-agent distributed optimization," in *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, (Shanghai, China), pp. 4134–4138, Mar. 2016.

[51] S. J. Wright, *Coordinate descent algorithms*. Mathematical Programming, vol. 151, no. 1, pp. 3–34, Mar. 2015.

[52] P. Richtárik and M. Takáč, *Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function*. Mathematical Programming, pages 1–38, Dec. 2012.

[53] R. Tappenden, P. Richtárik, and J. Gondzio, “Inexact coordinate descent: complexity and preconditioning,” *Journal of Optimization Theory and Applications*, vol. 170, pp. 144–176, Jul. 2016.

[54] P. Richtárik and M. Takáč, *Distributed coordinate descent method for learning with big data*. arXiv:1310.2059, 2013.

[55] H. Mania, X. Pan, D. Papailiopoulos, B. Recht, K. Ramchandran, and M. I. Jordan, *Perturbed iterate analysis for asynchronous stochastic optimization*. arXiv:1507.06970v2, 2016.

[56] H. V. Poor, *An introduction to signal detection and estimation*. New York: Springer Science & Business Media, 2013.

[57] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley Series in Telecommunications and Signal Processing, Wiley-Interscience, 2006.

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

[59] P. Tufekci, *Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods*. International Journal of Electrical Power and Energy Systems, vol. 60, pp. 126–140, Feb. 2014.

[60] I. Csiszár, “The method of types,” *IEEE Trans. Inform. Theory*, vol. 44, pp. 2505–2523, Oct. 1998.

[61] I. Csiszár and J. Körner, *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Cambridge University Press, 2011.