Collaborative Estimation in Networks

A Thesis

Submitted to the Faculty

of

Drexel University

by

Sivagnanasundaram Ramanan

in partial fulfillment of the

requirements for the degree

of

Doctor of Philosophy in Electrical Engineering

July 2011

Acknowledgements

This dissertation would not be possible without the help of many people.

First of all, I would like to thank my advisor, Prof. John MacLaren Walsh, for his wonderful guidance last four years. His patience and encouragement throughout my graduate studies is much appreciated.

I also would like to thank my friend Ciira who was very helpful during many difficult periods.

I would not have been able to finish this work without continuous support and encouragement of my wife, parents, sisters and friends. Their love and support is much appreciated and will always be remembered.

I would not have been at Drexel without the help of Prof. Ratnajeevan Hoole and Prof. Nadarajah Vasanthan. Their help will always be remembered.

Dedications

To my parents.

Table of Contents

List of Figures

Abstract

Collaborative Estimation in Networks

Sivagnanasundaram Ramanan

Advisor: John MacLaren Walsh, PhD

Consider a network of nodes that are deployed to monitor a common phenomenon. In many cases, the network nodes need to estimate the common phenomenon from "noisy" observations they make. Although each node can independently obtain the estimate from its own observations, it can obtain a better estimate by communicating with the other nodes and by exploiting the interdependence between the observations at different nodes in estimation. In this dissertation, we study such a collaborative estimation problem in which a network of nodes, each indirectly observing an underlying source through "noisy" measurements, communicate with each other in order to form better estimates of the underlying source.

Two primary constraints, complexity and communication, should be taken into account in the design of collaborative estimation algorithms. Although it is preferrable to have both low complexity and low communication, the limits on the estimation error performance can be studied by relaxing either or both of these constraints. In particular, signal processing and machine learning based approaches tend to focus on developing low complexity collaborative estimation algorithms with less strict attention to communication, while information theory focusses on characterizing the fundamental tradeoff between communication rate and estimate performance, with less attention to complexity. Reconciling complexity with communications, modern practical coding theory can aid the design of collaborative estimation algorithms with low-complexity and low-communication.

In this dissertation, we apply tools from aforementioned areas of study to develop collaborative estimation algorithms with low-complexity, low-communication and low-complexity/low-communication, and evaluate the estimation error performances.

1. Introduction

In networks there often arises a scenario in which network nodes must estimate a common source from the noisy observations they made about the source. Each node can formulate the Bayesian estimation problem independently using its own observations and obtain the estimate of the source. However, because these obervations are about the same source, they are statistically dependent across different nodes and this dependence structure can be exploited to obtain better estimates. For example, Fig. 1.1 shows a network of sensor nodes deployed to monitor a forest fire by measuring carbon monoxide. The carbon monoxide measurements at different nodes are statistically dependent on each other and the source's location, because they are generated from the same source. Thus, the nodes can collaborate with each other to obtain better estimates of carbon monoxide level and source position, and take action accordingly.

In this dissertation, we study such a collaborative estimation problem in which

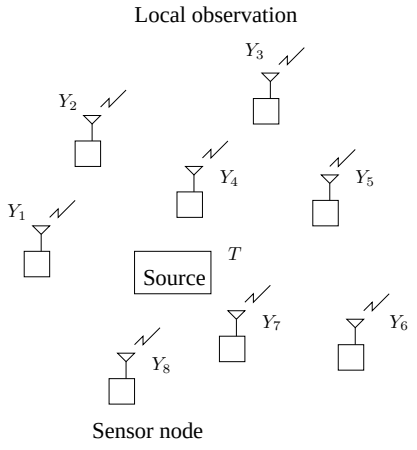


Figure 1.1: An example of collaborative estimation problem.

a network of nodes, each indirectly observing an underlying source through "noisy" measurements, communicate with each other in order to form better estimates of the underlying source. This type of scenario often arises in wireless sensor networks (WSN) [1] where a network of sensor nodes are deployed to monitor a common phenomenon. These sensor nodes are composed of 4 main units: sensor, power supply, processor and transceiver [1]. Power consumption is an important concern in wireless sensor networks, often because they are deployed in inaccessible terrains which forbids frequent replenishment of the power supply.

Keeping this in mind, two primary constraints, complexity and communication, should be taken into account in the design of collaborative estimation algorithms. This is because the power used for transmission and computation amounts to a significant portion of the power consumption at the nodes. Indeed, the number of computations and the number of messages exchanged between the nodes during the collaborative estimation directly influence the network lifetime. Also, the collaborative estimation algorithms are implemented on small processors at the nodes which are generally not very powerful. Thus, ideally a collaborative estimation algorithm that is applied in practice should have low implementation complexity and low information exchange between the nodes. However, we can study the limits on the estimation error performance by relaxing either or both of these constraints.

As shown in Table 1.1, the joint Bayesian estimation [2] provides a collaborative estimation algorithm which yields the best estimation error performance among all algorithms. Let T be a source and $Y_1, \ldots, Y_M$ be the observations of the source at M different nodes. Then, the joint Bayesian estimate $\hat{T}$ is given by

$$\hat{T} = \arg \min_{\tilde{T}(y_1,\ldots,y_M)} \int C[t, \tilde{T}(y_1, \ldots, y_M)] \; p(t|y_1, \ldots, y_M) \; dt \tag{1.1}$$

where $C[t, \tilde{T}]$ is Bayesian cost function which is discussed in more detail in Section

Table 1.1: Different areas of study from which tools can be applied to study collaborative estimation

		Communication	
		Low	High
Complexity	Low	Modern Practical Coding Theory	Signal Processing & Machine Learning
	High	Information Theory & Coding	Joint Bayesian Estimation

2.1.1. This algorithm does not take into account the complexity and communication. Indeed, it requires the pooling of the available data, followed by an intractable expectation and minimization. Signal processing and machine learning based approaches tend to focus on developing low complexity collaborative estimation algorithms with less strict attention to communication. Conversely, the discipline of information theory, as applied to collaborative estimation, focusses on characterizing the fundamental tradeoff between communication rate and estimate performance, with less attention to complexity. Reconciling complexity with communications, by borrowing tools from both information theory and machine learning, modern practical coding theory can aid the design of collaborative estimation algorithms with low-complexity and low-communication.

In this dissertation, we study collaborative estimation problem applying tools from aforementioned areas of study and develop algorithms with the following properties.

1. Low-complexity algorithms

2. Low-communication algorithms

3. Low-complexity and Low-communication algorithms

Low-Complexity Algorithms

The tools from signal processing and machine learning have been extensively used to develop low complexity collaborative estimation algorithms in the literature. In many cases, a closed form expression for the joint Bayesian estimate in (1.1) is not obtainable, because it requires expectation with respect to the joint distribution of source and observations which is in many cases analytically complex and intractable. The complexity of obtaining joint Bayesian estimates can be reduced by either restricting the estimate to be linear or applying approximate inference techniques from machine learning to approximate the intractable distributions with tractable distributions [3]. These techniques can provide low complexity algorithms that can be used to obtain an approximation to the joint Bayesian estimate. These algorithms often require exchange of messages only between the neighboring nodes unlike the joint Bayesian estimation which requires pooling of all data at one place.

Stochastic gradient algorithms from adaptive filter theory represent one class of such algorithms which can be applied to obtain low complexity linear minimum mean-square error (LMMSE) estimates. These algorithms can be adapted to perform distributed estimation in networks as LMS/RLS algorithms were adapted to derive Diffusion LMS/RLS based distributed estimation algorithms in [4, 5]. Consensus propagation based distributed estimation algorithms have also drawn much attention in recent years [6, 7, 8]. These algorithms aim to obtain global estimates of random/unknown parameters at each node from the observations by allowing communications only between the neighboring nodes. One problem with these algorithms is that under some formulations, the number of messages need to be exchanged between the nodes increases as the number of obervations increases.

We propose a low complexity collaborative estimation algorithm based on an approximate inference technique, Expectation Propagation (EP) [9, 3], for which the number of messages exchanged does not increase with the number of observations made. In particular, we derive a collaborative estimation algorithm based on EP which can be used for collaborative estimation of channel gains in a wireless sensor network. We show by simulations that our algorithm performs better than some other collaborative estimation algorithms including the diffusion LMS algorithm.

Low-Communication Algorithms

Theoretical bounds on the achievable tradeoffs between communication rates and estimation error performances can be studied by employing techniques from the rate-distortion theory [10]. In point-to-point rate-distortion theory, the encoder encodes the source as a rate constrained description and the decoder decodes the description such that expected distortion between the source and reconstruction is minimized. The distortion measure in the lossy source coding is analogous to the Bayesian cost in the estimation theory, as both measure the quality of the reconstruction or estimate. Rate distortion theory exhaustively characterizes the rates at the encoders and distortions (estimation error) achievable at the decoders. Lossy source coding is called direct source coding or remote source coding depending on whether the encoder observes the source directly or indirectly. As each encoder participating in our collaborative estimation algorithm only makes indirect observations of the source through "noisy" measurements, our collaborative estimation problem is a complex remote source coding problem.

Dobrushin and Tsybakov [11] first studied the remote source coding problem in the point-to-point context. A simple extension to the point-to-point remote source coding is to include a side information at the decoder [12, 13]. The real challenge

of the remote source coding problems lies in the network context. Many people have studied different remote source coding problems in the network context. One important class of problems is studied under the name of the CEO problem [14, 15, 16] where the central estimation officer (CEO) estimates an underlying source from the messages it received from his agents which encode their messages based on the indirect observations of the source. The complete rate distortion region for the CEO problem is known only for the quadratic Gaussian case which was independently proved by Oohama [17] and Prabhakaran et al. [18]. Another important class of problems is called the Multiple Descriptions problem [19, 20] and the successive refinement problem [21, 22, 23] where the encoder encodes multiple descriptions of the source to the decoders and the decoders reproduce the source with different fidelity depending on the subset of the descriptions they received.

Most of the literature studies distributed estimation problem when there is only one node either on encoder side or decoder side. However, there will be several encoders and several decoders in a collaborative estimation problem. Thus, we propose a suitable source code architecture for the collaborative estimation problem and study the rates and distortions achievable at the nodes by hybridizing the techniques from the CEO and multiple descriptions problems. We show that our achievable rate distortion region simplifies to the known bounds for some simpler problems in the literature.

Low-Complexity/Low-Communication Algorithms

The development of low-complexity and low-communication collaborative estimation algorithms via practical source codes emerged as a new area of study about a decade ago and is still arguably in its infancy. Modern practical coding theory borrows ideas from machine learning and coding theory to design such collaborative

estimation algorithms.

The design of practical codes started drawing interest of the researchers after the codes were designed for the binary Slepian-Wolf problem in the lossless coding context [24]. These codes were designed either by puncturing the Turbo codes or by sending syndromes of the LDPC codes [25, 26]. Following these techniques, codes were designed for the Wyner-Ziv problem in the lossy coding context for both discrete and continuous sources. The codes for continuous sources were designed by quantizing the source with sophisticated quantizers and then compressing the source with LDPC or Turbo codes [27, 28, 29]. Later, codes for multiterminal source coding problem and the CEO problem were designed by applying the Wyner-Ziv coding successively where at each stage previously decoded messages were used as side information [30, 31].

The practical multi-terminal code and decoder design literature for distributed estimation has thus far been focussed on non-interactive communication. In this dissertation, we design practical codes for a multiround collaborative estimation problem in which multiple successively refinable messages are broadcast at each round. We apply successively refinable trellis coded quantization (SR-TCQ) to quantize a continuous source and compress the quantized source using syndromes of the LDPC codes. The adaptation of the SR-TCQ and LDPC codes to this multiterminal multiround context allows for efficient low communication collaborative estimation, and belief propagation decoders allow for the effective use of side information while maintaining low complexity.

Notation

We will be using the following notations throughout the dissertation.

1. We denote the set $\{1, \ldots, M\}$ as $[M]$ for any natural number M. Also, $[M] \setminus i$ will denote the set $[M]$ with the element i removed.

2. For some set $\mathcal{A}$, $2^{\mathcal{A}}$ will be the power set of subsets from $\mathcal{A}$.

3. We use capital letters for random variables and small letters for realizations. We use superscripts for time indices and subscripts for node indices.

4. For a sequence of random variables $X^{(n)}$, $n \in [N]$, we define $\boldsymbol{X}$ as $\boldsymbol{X} := \left[X^{(1)}, \ldots, X^{(N)}\right]$, X^n as $X^n := \left[X^{(1)}, \ldots, X^{(n)}\right]$ and $X^{[m,n]}$ as $X^{[m,n]} := \left[X^{(m)}, \ldots, X^{(n)}\right]$.

5. For any set of subscript indices $K = \{k_1, \ldots, k_L\}$, the vector $(F_{k_1}, \ldots, F_{k_L})$ is denoted with F_K.

6. For a set of length N i.i.d. sequences $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_L$, the set of jointly strongly typical sequences [10] is denoted as $A^*_\epsilon(X_{[L]})$.

7. The notation $X \leftrightarrow Y \leftrightarrow Z$ means that X, Y, Z form a Markov chain.

The rest of the dissertation is organized as follows. Chapter 2 provides some background that is necessary to develop algorithms in this dissertation. This includes Bayesian inference and estimation theory, lossy source coding theory, and quantization and compression. In Chapter 3, we develop a low-complexity algorithm for joint channel estimation in a wireless sensor network and compare its performance with another network estimation algorithm, diffusion LMS. In Chapter 4, we propose a communication protocol for a general collaborative estimation algorithm and study rate-estimation error performance of low-communication algorithms. We prove inner and outer bounds to the rate distortion region and show our inner bound simplifies to the known bounds for some simpler problems. In Chapter 5, we propose a communication protocol for multiround collaborative estimation and design a low-complexity/low-communication algorithm for this problem. We simulate our low-complexity/low-communication algorithm and compare the performance with the theoretical bounds that we derive.

2. Background

In this chapter, we provide the background that is necessary to read this dissertation. As discussed in Chapter 1, we apply tools from signal processing & machine learning, information theory and modern practical coding theory to study collaborative estimation problem. In particular, we apply tools from statistical inference and estimation, source coding theory, and practical coding theory. We provide background in this chapter for each of these areas of study. We begin our discussion with inference and estimation theory.

2.1 Statistical inference and Estimation Theory

We apply inference and estimation theory to develop a low-complexity collaborative estimation algorithm in Chapter 3. In particular, we apply Bayesian estimation techniques to estimate parameters as we explain presently [3].

2.1.1 Parameter Estimation

Parameter estimation deals with estimating a parameter using some data that depends on the parameter. Depending on whether the parameter to be estimated is nonrandom (unknown) or random, the estimation is called nonrandom parameter estimation or random parameter estimation [2].

Nonrandom Parameter Estimation

Let θ be an unknown (nonrandom) parameter and $\boldsymbol{X}$ be a set of variables. Suppose that the conditional distribution of $\boldsymbol{X}$ given θ is

$$p_{\boldsymbol{X};\theta}(\boldsymbol{x};\theta)$$

The goal in nonrandom parameter estimation is to find the estimate $\hat{\Theta}$ of θ from the realizations $\boldsymbol{x}$ of $\boldsymbol{X}$. *Maximum Likelihood estimation* (MLE) is a widely used technique in unknown parameter estimation which finds the estimate $\hat{\Theta}(\boldsymbol{x})$ by finding the value θ that maximizes the likelihood of $\boldsymbol{x}$ [2].

$$\hat{\Theta}(\boldsymbol{x}) \in \arg \max_{\theta \in \Phi(\theta)} p_{\boldsymbol{X};\theta}(\boldsymbol{x};\theta) \tag{2.1}$$

where $\Phi(\theta)$ is the feasible set of parameters θ. MLE can also be used for the estimation of random parameters with unknown prior distribution. When the prior distribution of the random parameter to be estimated is known, random parameter estimation techniques can be used to estimate the parameter.

Random Parameter Estimation

Let Θ be a continuous random parameter and $\boldsymbol{X}$ be a set of variables. Suppose that $p_\Theta(\theta)$ is the prior distribution of Θ and $p_{\boldsymbol{X}|\Theta}(\boldsymbol{x}|\theta)$ be the conditional distribution of $\boldsymbol{X}$ given Θ.

The Bayesian estimation technique finds the estimate $\hat{\Theta}$ of Θ from the realizations $\boldsymbol{x}$ of $\boldsymbol{X}$ such that the expected value of a cost function $C(\Theta, \hat{\Theta}(\boldsymbol{x}))$ is minimized [2].

$$\hat{\Theta}(\boldsymbol{x}) = \arg \min_{\tilde{\Theta}(\boldsymbol{x})} \int C(\theta, \tilde{\Theta}(\boldsymbol{x}))\ p_{\Theta|\boldsymbol{X}}(\theta|\boldsymbol{x})\ d\theta \tag{2.2}$$

A commonly used cost function is the squared error function.

$$C(\Theta, \hat{\Theta}(\boldsymbol{x})) = (\Theta - \hat{\Theta}(\boldsymbol{x}))^2$$

We use this cost function is Chapter 3 and the resulting estimate is called Minimum

Mean Squared Error (MMSE) estimate.

$$\begin{aligned} \hat{\Theta}(\boldsymbol{x})_{\text{MMSE}} &= \arg \min_{\hat{\Theta}(\boldsymbol{x})} \mathbb{E}[(\Theta - \hat{\Theta}(\boldsymbol{x}))^2] \\ &= \int \theta \; p_{\Theta|\boldsymbol{X}}(\theta|\boldsymbol{x}) \; d\theta \end{aligned} \tag{2.3}$$

Thus, the computation of MMSE estimate requires the posterior distribution of the random parameter which can be inferred from the data $\boldsymbol{x}$.

2.1.2 Belief Propagation

In many cases, $\boldsymbol{\theta}$ is a vector, and only the marginal posterior distributions $p(\theta_i|\boldsymbol{x})$ need to be computed. This occurs in instances of cost functions which are separable, which include the MMSE case in (2.3). Belief propagation (BP) [3, 32] is an approximate Bayesian inference technique which can be applied to compute marginal posterior distributions from a multiplicative factoring of the joint distribution of the data and the parameters.

Belief propagation can be described using the sum-product algorithm on factor graphs [33].

Factor Graph

A factor graph is a bipartite graph which can be used to represent factorization of functions. A factor graph has two set of vertices: variable nodes and factor nodes. The variable nodes represent the variables of the function and the factor nodes represent the factors of the function. A variable node is connected to a factor node only if the factor is a function of that variable. For example, the factor graph in Fig. 2.1 represents the function $g(x_1, x_2, x_3, x_4, x_5) = f_1(x_1)f_2(x_2)f_3(x_1, x_2, x_3)f_4(x_3, x_4)f_5(x_3, x_5)$ where the variables $\{x_i\}_{i=1}^5$ are represented by the variable (circle) nodes and the fac-

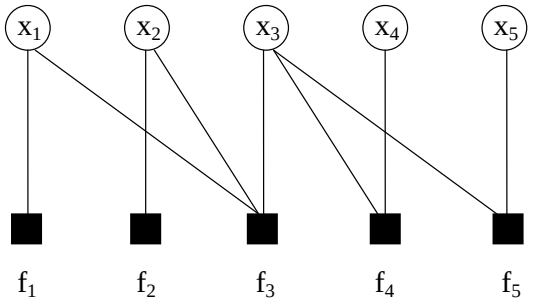


Figure 2.1: The factor graph represents the function $g(x_1, x_2, x_3, x_4, x_5)$.

tors $\{f_i\}_{i=1}^5$ are represented by the factor (square) nodes.

Now we return to the discussion of BP. Suppose that a joint distribution $p(x_1, \ldots, x_N)$ factorizes as follows.

$$p(x_1, \ldots, x_N) = \prod_k f_k(\boldsymbol{x}_k) \tag{2.4}$$

where $\boldsymbol{x}_k \subseteq \{x_1, \ldots, x_N\}$. BP iteratively passes messages between the variable and factor nodes to compute the marginals at the variable nodes. The message from variable node x_i to factor node f_j is computed as follows.

$$\lambda_{x_i \to f_j}(x_i) = \prod_{f_k \in \mathcal{N}(x_i) \setminus f_j} \mu_{f_k \to x_i}(x_i) \tag{2.5}$$

where $\mathcal{N}(x_i)$ is the set of neighbors of x_i in the factor graph. The message from factor node f_j to variable node x_i is computed as follows.

$$\mu_{f_j \to x_i}(x_i) = \sum_{\mathcal{N}(f_j) \setminus x_i} f_j(\boldsymbol{x}_j) \prod_{x_k \in \mathcal{N}(f_j) \setminus x_i} \lambda_{x_k \to f_j}(x_k) \tag{2.6}$$

where $\mathcal{N}(f_j)$ is the set of neighbors of f_j and $\boldsymbol{x}_j := \{x_n : n \in \mathcal{N}(f_j)\}$. After certain number of iterations, the marginal at variable node x_i is computed as follows.

$$\hat{p}(x_i) = \prod_{f_k \in \mathcal{N}(x_i)} \mu_{f_k \to x_i}(x_i) \tag{2.7}$$

These approximate marginals converge to the true marginals when the facor graph is cycle free. When the factor graph has cycles, BP gives an approximate solution to the true marginal.

2.1.3 Expectation Propagation

For many probabilistic models of interest, working with the true posterior distribution is intractable. In such situations, the true posterior distribution must be approximated with a tractable probability distribution such that the approximate distribution is as close as possible to the true distribution. Expectation propagation [34, 9, 3] is an approximate inference algorithm which approximates an intractable true posterior distribution having the form of product of factors with an exponential family distribution by minimizing the Kullback-Leibler divergence between the two distributions.

To mathematically describe expectation propagation, let $\mathcal{D}$ be data and $\boldsymbol{\theta}$ be latent variables. Suppose that the posterior distribution of the latent variables given the data is $p(\boldsymbol{\theta}|\mathcal{D})$ and that it is an intractable distribution. Now we want to approximate this distribution with an exponential family distribution $q(\boldsymbol{\theta})$ of the form

$$q(\boldsymbol{\theta}) = h(\boldsymbol{\theta}) g(\boldsymbol{\lambda}) \exp\{\boldsymbol{u}(\boldsymbol{\theta}) \boldsymbol{\lambda}^T\} \tag{2.8}$$

where $\boldsymbol{\lambda}$ are called the natural parameters. Once the exponential family (determined

by $\boldsymbol{u}(\boldsymbol{\theta})$) is decided, the goal is to determine $\boldsymbol{\lambda}$ that minimizes $KL(p\|q)$ [3].

$$\begin{aligned} KL(p\|q) &= \int p(\boldsymbol{\theta}|\mathcal{D}) \ln\left(\frac{p(\boldsymbol{\theta}|\mathcal{D})}{q(\boldsymbol{\theta})}\right) d\boldsymbol{\theta} \\ &= -\ln(g(\boldsymbol{\lambda})) - \boldsymbol{\lambda}^T \mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}[\boldsymbol{u}(\boldsymbol{\theta})] + v(\boldsymbol{\theta}) \end{aligned} \tag{2.9}$$

where $v(\boldsymbol{\theta})$ is some function of $\boldsymbol{\theta}$. Taking the partial derivatives with respect to $\boldsymbol{\lambda}$ and equating it to 0 we get

$$-\nabla_{\boldsymbol{\lambda}} \ln(g(\boldsymbol{\lambda})) = \mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}[\boldsymbol{u}(\boldsymbol{\theta})]$$

But for exponential family distributions, we have $-\nabla_{\boldsymbol{\lambda}} \ln(g(\boldsymbol{\lambda})) = \mathbb{E}_{q(\boldsymbol{\theta})}[\boldsymbol{u}(\boldsymbol{\theta})]$ [3]. Thus,

$$\mathbb{E}_{q(\boldsymbol{\theta})}[\boldsymbol{u}(\boldsymbol{\theta})] = \mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}[\boldsymbol{u}(\boldsymbol{\theta})] \tag{2.10}$$

This is equivalent to finding $q(\boldsymbol{\theta})$ which has the same expected sufficient statistics as $p(\boldsymbol{\theta}|\mathcal{D})$. However, since $p(\boldsymbol{\theta}|\mathcal{D})$ is intractable, it is not possible to find the approximate distribution directly.

Expectation propagation utilizes factors of the original distribution to approximate the distribution. Suppose that the posterior distribution of $\boldsymbol{\theta}$ given $\mathcal{D}$ factorizes as follows.

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{1}{p(\mathcal{D})} \prod_{i=0}^{n} f_i(\boldsymbol{\theta}_i) \tag{2.11}$$

where $\boldsymbol{\theta}_i \subseteq \boldsymbol{\theta}$. Here, $f_i(\boldsymbol{\theta}_i)$ can be a function of both $\boldsymbol{\theta}_i$ and data $\mathcal{D}$, however since we are interested in the parameters only we denote it as a function of $\boldsymbol{\theta}_i$ only. Suppose that this posterior distribution is intractable and let $q(\boldsymbol{\theta})$ be another distribution such that

$$q(\boldsymbol{\theta}) = \frac{1}{T} \prod_{i=0}^{n} \hat{f}_i(\boldsymbol{\theta}_i) \tag{2.12}$$

where T is the normalization constant. EP approximates the posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$ with distribution $q(\boldsymbol{\theta})$ by restricting the factors $\hat{f}_i(\boldsymbol{\theta})$ to be exponential family distributions and minimizing the KL divergence between the distributions in the reverse form, i.e. $KL(p\|q)$. Rather than trying to approximate it in one step, EP tries to approximate each factor in turn. In particular, EP first intializes each factor in $q(\boldsymbol{\theta})$ to 1. Then, it selects a factor to approximate, removes the factor from $q(\boldsymbol{\theta})$, includes the original factor and finds the approximate distribution in the selected family that minimizes KL divergence. To mathematically describe it, say we want to refine the factor $\hat{f}_j(\boldsymbol{\theta}_j)$. EP first removes the factor $\hat{f}_j(\boldsymbol{\theta}_j)$ from $q(\boldsymbol{\theta})$

$$q^{\setminus j}(\boldsymbol{\theta}) = \frac{q(\boldsymbol{\theta})}{\hat{f}_j(\boldsymbol{\theta}_j)}$$

It then includes the original factor $f_j(\boldsymbol{\theta})$ and computes the distribution by normalizing.

$$\frac{1}{T_j} f_j(\boldsymbol{\theta}_j) q^{\setminus j}(\boldsymbol{\theta})$$

where

$$T_j = \int f_j(\boldsymbol{\theta}_j) q^{\setminus j}(\boldsymbol{\theta}) d\boldsymbol{\theta}$$

It then finds $q_j^{new}(\boldsymbol{\theta}_j)$ that minimizes

$$KL\left(\frac{1}{T_j} f_j(\boldsymbol{\theta}_j) q^{\setminus j}(\boldsymbol{\theta}) \;\middle\|\; q_j^{new}(\boldsymbol{\theta}_j) q^{\setminus j}(\boldsymbol{\theta})\right) \tag{2.13}$$

In this fashion, each factor is refined in turn and the approximation is continued for several times. The approximate distribution is given by $q^{new}(\boldsymbol{\theta})$.

We will next see that when a fully factorized distribution is used for the approximate distribution, EP boils down to belief propagation (BP). We will use this

property of EP in Chapter 3. Now let

$$q(\boldsymbol{\theta}) = \frac{1}{T} \prod_{i=0}^{n} \prod_{k} \hat{f}_{ik}(\theta_k) \tag{2.14}$$

If we want to refine the factor $\hat{f}_{j\ell}(\theta_\ell)$, we remove the factor $\hat{f}_j(\boldsymbol{\theta}_j)$ from $q(\boldsymbol{\theta})$.

$$q^{\setminus j}(\boldsymbol{\theta}) = \frac{1}{T} \prod_{i \neq j} \prod_{k} \hat{f}_{ik}(\theta_k)$$

Then we multiply by the original factor

$$q^{\setminus j}(\boldsymbol{\theta}) f_j(\boldsymbol{\theta}_j)$$

and take the marginal. The refined factor will be

$$\hat{f}_{j\ell}(\theta_\ell) \propto \sum_{\theta_k \in \boldsymbol{\theta}_j,\ k \neq \ell} f_j(\boldsymbol{\theta}_j) \prod_{i \neq j} \prod_{k \neq \ell} \hat{f}_{ik}(\theta_k) \tag{2.15}$$

This is exactly the message from a factor node to a variable node in belief propagation.

2.2 Source Coding Theory

Information theory provides theoretical bounds to two important classes of problems in communications: source coding and channel coding [10]. Information theoretic bounds for source coding provide bounds on the number of bits required to represent a source. If the source is compressed such that it can be reconstructed with an error ocurring with arbitrarily low probability it is called lossless compression, otherwise it is called lossy compression. We apply tools from lossy source coding theory (rate distortion theory) to characterize the region of rates and estimation error performances in Chapter 4.

Let $X^{(1)}, \ldots, X^{(N)}$ be a sequence of random variables independent and identically distributed according to $P(x)$. The rate distortion theory studies the problem of representing the sequence such that it can be reconstructed with certain amount of error (distortion). In this section, we first define some of the terms that we use in Chapter 4 and discuss some of the important distributed source coding problems in the literature.

2.2.1 Definitions

Entropy

Entropy is a measure of uncertainty in a random variable. Let X be a discrete random variable which takes values x from $\mathcal{X}$ and $P(x)$ be its probability mass function. Then, the entropy $H(X)$ of X is defined as

$$H(X) = -\sum_{x \in \mathcal{X}} P(x) \ \log_2\{P(x)\} \tag{2.16}$$

When the base of the logarithm is 2, entropy is measured in "bits" and when natural logarithm is used it is measured in "nats".

Differential Entropy

When random variable X is continuous, the entropy is called differential entropy and it is defined as

$$h(X) = -\int p(x) \ \log_2\{p(x)\} \ dx \tag{2.17}$$

where $p(x)$ is the probability density function of x. Note that the differential entropy is denoted by $h(X)$ to differentiate it from the entropy $H(X)$ of a discrete random variable.

Joint Entropy

The joint entropy $H(X,Y)$ of two discrete random variables X and Y is defined as

$$H(X,Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\ \log_2\{P(x,y)\} \tag{2.18}$$

where $P(x,y)$ is the joint probability mass function of X and Y.

Conditional Entropy

Conditional entropy measures the uncertainty of a random variable given another random variable. The conditional entropy $H(X|Y)$ of X given Y is defined as

$$H(X|Y) = -\sum_{y\in\mathcal{Y}}\sum_{x\in\mathcal{X}} P(x,y)\ \log_2\{P(x|y)\} \tag{2.19}$$

where $P(x|y)$ is the conditional mass function of X given Y.

Mutual Information

Mutual information measures how much uncertainty of a random variable is removed when another random variable is given. The mutual information $I(X;Y)$ between X and Y is defined as

$$I(X;Y) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\ \log_2\left\{\frac{P(x,y)}{P(x)P(y)}\right\} \tag{2.20}$$

Distortion

Distortion measures the quality of reconstruction of the source. A *distortion measure* is defined as

$$d : \mathcal{X}\times\hat{\mathcal{X}} \rightarrow \mathcal{R}^{+} \tag{2.21}$$

where $\mathcal{X}$ is the source alphabet and $\hat{\mathcal{X}}$ is the reconstruction alphabet.

The distortion between two sequences $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ is defined as

$$d(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{1}{N} \sum_{n=1}^{N} d(x^{(n)}, \hat{x}^{(n)}) \tag{2.22}$$

Jointly Strongly Typical Sequences

The properties of jointly strongly typical sequences are widely exploited to prove achievable rate distortion regions. A pair of sequences $(\boldsymbol{x}, \boldsymbol{y}) \in \boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{Y}}$ is said to be ϵ-strongly typical with respect to a distribution if $p(x, y)$ on $\mathcal{X} \times \mathcal{Y}$ if

- For all $(a, b) \in \mathcal{X} \times \mathcal{Y}$ with $p(a, b) > 0$, we have

$$|\frac{1}{N}\text{Num}(a, b|\boldsymbol{x}, \boldsymbol{y}) - p(a, b)| < \frac{\epsilon}{|\mathcal{X}||\mathcal{Y}|}$$

- For all $(a, b) \in \mathcal{X} \times \mathcal{Y}$ with $p(a, b) = 0$, we have $\text{Num}(a, b|\boldsymbol{x}, \boldsymbol{y}) = 0$.

where $\text{Num}(a, b|\boldsymbol{x}, \boldsymbol{y})$ is the number of occurrences of the pair (a, b) in the pair of sequences $(\boldsymbol{x}, \boldsymbol{y})$.

We next discuss some of the important problems in the lossy source coding literature.

2.2.2 Point-to-Point Source Coding

Let $T^{(1)}, \ldots, T^{(N)}$ be a source sequence independent and identically distributed according to $P(t)$. In point-to-point lossy source coding, as shown in Fig. 2.2 an encoder observing the source sequence sends a message with index $m \in \{1, \ldots, 2^{NR}\}$ to the decoder which recontructs the source as $\hat{T}^{(1)}, \ldots, \hat{T}^{(N)}$ such that the expected distortion is D.

The rate distortion pair (R, D) is said to be achievable if there exists an encoding

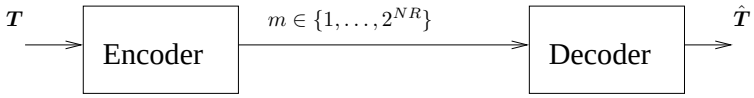


Figure 2.2: Point-to-point lossy source coding problem.

function f and a decoding function g

$$f : \boldsymbol{\mathcal{T}} \to \{1, \ldots, 2^{NR}\}, \quad g : \{1, \ldots, 2^{NR}\} \to \hat{\boldsymbol{\mathcal{T}}}$$

such that the expected distortion

$$\frac{1}{N} \sum_{n=1}^{N} E[d(T^{(n)}, \hat{T}^{(n)})] \leq D$$

The *rate distortion region* to this problem is defined as the convex hull of all achievable (R, D). The *rate distortion function* is defined as the minimum rate required to achieve distortion D, i.e.

$$R(D) := \min\{R : \ (R, D) \text{ is achievable}\} \tag{2.23}$$

The rate distortion function to this problem is given by [10]

$$R(D) = \min_{p(\hat{t}|t):\ E[d(T,\hat{T})] \leq D} I(T; \hat{T}) \tag{2.24}$$

2.2.3 Remote Source Coding

The encoder sometimes only get to observe a noisy version of the source, and this coding problem is called the remote source coding problem. Dobrushin and Tsybakov [11] first studied the remote source coding problem in the point-to-point context. In

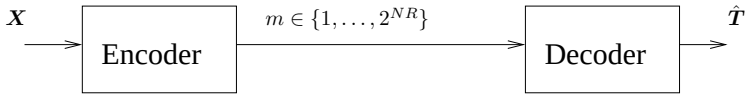


Figure 2.3: Point-to-point remote lossy source coding problem.

remote source coding problem, as shown in Fig. 2.3 the encoder observes a noisy version $\boldsymbol{X}$ of the source $\boldsymbol{T}$. The source and observation sequences are i.i.d. according to $p(t, x)$.

The rate distortion pair (R, D) is said to be achievable if there exists an encoding function f and a decoding function g

$$f : \boldsymbol{\mathcal{X}} \to \{1, \ldots, 2^{NR}\}, \quad g : \{1, \ldots, 2^{NR}\} \to \hat{\boldsymbol{\mathcal{T}}}$$

such that the expected distortion

$$\frac{1}{N} \sum_{n=1}^{N} E[d(T^{(n)}, \hat{T}^{(n)})] \leq D$$

The rate distortion region to this problem is defined as the convex hull of all achievable (R, D). The rate distortion function to this problem is given by [11]

$$R(D) = \min_{\hat{T} \in \Phi(\hat{T})} I(X; \hat{T}) \tag{2.25}$$

where $\Phi(\hat{T}) = \{\hat{T} : \ T \leftrightarrow X \leftrightarrow \hat{T}, \ E[d(T, \hat{T})] \leq D\}$.

2.2.4 Source Coding with Decoder Side Information

A simple extension to the point-to-point remote coding is to include a side information at the decoder. This problem is known as the indirect Wyner-Ziv problem

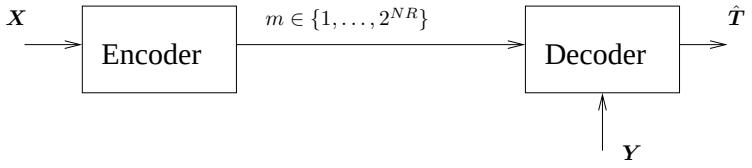


Figure 2.4: Remote Wyner-Ziv coding problem.

[13, 12, 35]. In the indirect Wyner-Ziv problem, both the encoder and decoder observe noisy versions of the source $\boldsymbol{T}$ as shown in Fig. 2.4. The source and observation sequences are i.i.d. according to $p(t, x, y)$.

The rate distortion pair (R, D) is said to be achievable if there exists an encoding function f and a decoding function g

$$f : \boldsymbol{\mathcal{X}} \to \{1, \ldots, 2^{NR}\}, \quad g : \boldsymbol{\mathcal{Y}} \times \{1, \ldots, 2^{NR}\} \to \hat{\boldsymbol{\mathcal{T}}}$$

such that the expected distortion

$$\frac{1}{N} \sum_{n=1}^{N} E[d(T^{(n)}, \hat{T}^{(n)})] \leq D$$

The rate distortion region to this problem is defined as the convex hull of all achievable (R, D). The rate distortion function to this problem is given by [13, 35]

$$R(D) = \min_{U \in \Phi(U)} I(X; U|Y) \tag{2.26}$$

where $\Phi(U) = \{U : \ T, Y \leftrightarrow X \leftrightarrow U, \ E[d(T, \hat{T}(Y, U))] \leq D\}$.

A special case of indirect Wyner-Ziv problem was studied in [36] where the decoder reproduces a deterministic function of the observation at the encoder and side information at the decoder.

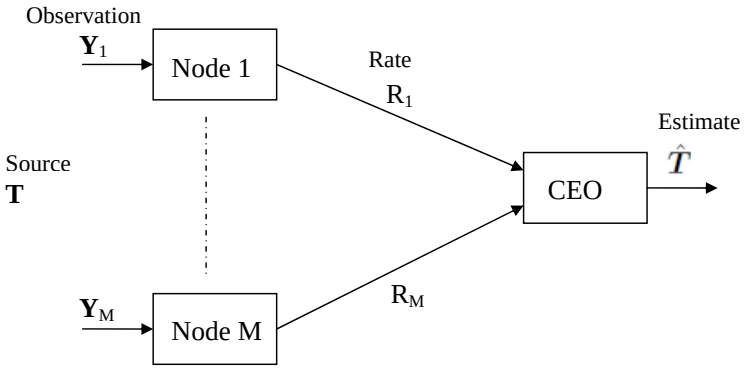


Figure 2.5: The CEO problem.

2.2.5 CEO Problem

The real challenge of the remote source coding problems lies in the network context. Many people have studied different remote source coding problems in the network context. One important class of problems is studied under the name of the CEO problem [14, 15, 16]. In the CEO problem, the central estimation officer (CEO) is interested in estimating a source and he employs some agents to report to him observing the source as shown in Fig. 2.5. The agents making "noisy" observations of the source independently encode their observations as rate constrained messages to the CEO. Receiving the messages, the CEO estimates the underlying source with some distortion. Let $p(t, y_1, \ldots, y_M)$ be the joint distribution of the source and observations.

The rate distortion vector $(R_1, \ldots, R_M, D)$ is said to be achievable if there exist encoding functions

$$f_i : \boldsymbol{\mathcal{Y}}_i \to \{1, \ldots, 2^{NR_i}\}, \quad \forall i \in \{1, \ldots, M\}$$

and a decoding function

$$g : \prod_{i=1}^{M} \{1, \ldots, 2^{NR_i}\} \rightarrow \hat{\boldsymbol{\mathcal{T}}}$$

such that the expected distortion

$$\frac{1}{N} \sum_{n=1}^{N} E[d(T^{(n)}, \hat{T}^{(n)})] \leq D$$

The rate distortion region for this problem is defined as the convex hull of all achievable $(R_1, \ldots, R_M, D)$.

For a general distribution of the source/observations and distortion metric, the complete rate distortion region is still unknown, but inner and outer bounds to the region are available [16].

Inner Bound:

Let $\Phi(U_{[M]})$ be the set of random vectors $(U_1, \ldots, U_M)$ that satisfy the following conditions.

1. $U_i \leftrightarrow Y_i \leftrightarrow T, Y_{[M]\setminus i}, U_{[M]\setminus i}$ for all i.
2. There exists a function $g : \mathcal{U}_1 \times \ldots \times \mathcal{U}_M \rightarrow \hat{\mathcal{T}}$ such that $E[d(T, \hat{T})] \leq D$ where $\hat{T} = g(U_1, \ldots, U_M)$.

Let

$$\begin{aligned} \mathcal{RD}(U_{[M]}) \;=\; & \{(R_1, \ldots, R_M, D) : \sum_{i \in \mathcal{A}} R_i \geq I(Y_{\mathcal{A}}; U_{\mathcal{A}} | U_{[M]\setminus\mathcal{A}}),\ \forall \mathcal{A} \subseteq [M] \\ & E[d(T, g(U_1, \ldots, U_M))] \leq D,\ (U_1, \ldots, U_M) \in \Phi(U_{[M]})\} \end{aligned} \quad (2.27)$$

and

$$\mathcal{RD}_{in} = \text{conv}\left(\bigcup_{U_{[M]} \in \Phi(U_{[M]})} \mathcal{RD}(U_{[M]}) \right)$$

where conv denotes convex hull. $\mathcal{RD}_{in}$ is an inner bound to the rate distortion region $\mathcal{RD}$, i.e. $\mathcal{RD}_{in} \subseteq \mathcal{RD}$. This inner bound is known as the Berger Tung inner bound [37, 38, 39, 15].

Outer Bound:

Let $\Phi(W_{[M]})$ be the set of random vectors $(W_1, \ldots, W_M)$ that satisfy the following conditions.

1. $W_i \leftrightarrow Y_i \leftrightarrow T, Y_{[M]\backslash i}$ for all i.
2. There exists a function $f : \mathcal{W}_1 \times \ldots \times \mathcal{W}_M \rightarrow \hat{\mathcal{T}}$ such that $E[d(T, \hat{T})] \leq D$ where $\hat{T} = f(W_1, \ldots, W_M)$.

Let

$$\begin{aligned} \mathcal{RD}(W_{[M]}) &= \{(R_1, \ldots, R_M, D) : \sum_{i \in \mathcal{A}} R_i \geq I(Y_{[M]}; W_{\mathcal{A}} | W_{[M]\backslash \mathcal{A}}),\ \forall \mathcal{A} \subseteq [M] \\ &\qquad E[d(T, f(W_1, \ldots, W_M))] \leq D,\ (W_1, \ldots, W_M) \in \Phi(W_{[M]})\} \end{aligned} \tag{2.28}$$

and

$$\mathcal{RD}_{out} = \bigcup_{W_{[M]} \in \Phi(W_{[M]})} \mathcal{RD}(W_{[M]})$$

$\mathcal{RD}_{out}$ is an outer bound to the rate distortion region $\mathcal{RD}$, i.e. $\mathcal{RD} \subseteq \mathcal{RD}_{out}$. This outer bound is known as the Berger Tung outer bound [37, 38, 15].

The complete rate distortion region for the CEO problem is known only for the quadratic Gaussian case which was independently proved by Oohama [17] and Prabhakaran et al. [18]. There, they have shown that the rate distortion region is equal to the Berger Tung inner bound. Chen [40] has proved that every point in the quadratic Gaussian rate distortion region can be achieved through successive Wyner-Ziv coding.

Wagner and Anantharam [16] have presented an outer bound to a general multi-

terminal source coding problem which provides a tighter outer bound than the Berger Tung outer bound for the CEO problem. A 2-terminal variant of the multiterminal source coding in [16] was studied in [41] where the decoder reproduces a function of the observations at the encoders with zero error probability. This paper presented an encoding technique based on graph color coding [42] that achieves entire rate region under a special condition "the zig-zag condition" [41].

2.2.6 Multiple Descriptions Problem

Another important class of problems is called the Multiple Descriptions problem [19, 20]. In the multiple descriptions problem, the encoder is required to send information about the source to the decoder over a unreliable channel. Rather than sending a single description, the encoder sends multiple descriptions of the source so that a subset of the descriptions will reach the decoder. The decoder is supposed to reconstruct the source with different distortions depending on the subset of descriptions it received. This problem can modeled with one encoder and multiple decoders each receiving a different subset of descriptions. For example, Fig. 2.6 shows an encoder sending two descriptions to the decoder.

The rate distortion vector $(\{R_i\}_{i=1}^{M}, \{D_{\mathcal{A}}\}_{\mathcal{A}\in\{1,\ldots,M\}\setminus\{0\}})$ is said to be achievable if there exist encoding functions

$$f_i : \boldsymbol{\mathcal{T}} \rightarrow \{1, \ldots, 2^{NR_i}\}, \quad \forall i \in \{1, \ldots, M\}$$

and decoding functions

$$g_{\mathcal{A}} : \prod_{i\in\mathcal{A}} \{1, \ldots, 2^{NR_i}\} \rightarrow \hat{\boldsymbol{\mathcal{T}}}_{\mathcal{A}}, \quad \forall \mathcal{A} \in 2^{\{1,\ldots,M\}} \setminus \{0\}$$

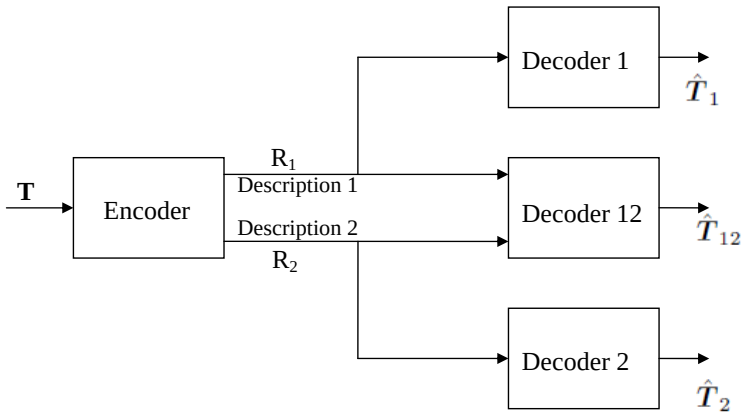


Figure 2.6: The 2-multiple descriptions problem.

such that

$$\frac{1}{N}\sum_{n=1}^{N} E[d(T^{(n)}, \hat{T}_{\mathcal{A}}^{(n)})] \leq D_{\mathcal{A}}, \quad \forall \mathcal{A} \in 2^{\{1,\ldots,M\}} \setminus \{0\}$$

The rate distortion region for this problem is defined as the convex hull of all achievable $(R_1, \ldots, R_M, D)$.

El-Gamal and Cover proved an achievable rate distortion region for the 2-descriptions ($M = 2$) problem which is known as the EGC region. This inner bound gives the entire rate distortion region in the quadratic Gaussian case [20] and in the so-called "no-excess rate" case [43] which is obtained when $R_1 + R_2 = R(D_{\{1,2\}})$, where $R(D_{\{1,2\}})$ is the rate distortion function in (2.24). However, the EGC region has been demonstrated not to be tight in other cases, including the binary Hamming case [44].

For general M-descriptions problem, an inner bound was proved in [45] by generalizing the EGC region and it was shown that their outer bound for the quadratic Gaussian case meets their inner bound if the rates of all descriptions are equal and the distortion constraints at all other decoders but the decoder receiving all descriptions are the same. The following is the inner bound presented in [45].

Inner Bound:

Let $\Phi(\hat{T}_{2^{[M]}\setminus\emptyset})$ be the set of random vectors $\{\hat{T}_{\mathcal{A}} : \mathcal{A} \in 2^{[M]}\setminus\emptyset\}$ that satisfy $E[d_{\mathcal{A}}(T,\hat{T}_{\mathcal{A}})] \leq D_{\mathcal{A}},\ \forall \mathcal{A} \in 2^{[M]} \setminus \emptyset$. Let

$$\begin{aligned}\mathcal{RD}(\hat{T}_{2^{[M]}\setminus\emptyset}) \;=\; \Big\{(R_{[M]}, D_{2^{[M]}\setminus\emptyset}) : \sum_{i\in\mathcal{A}} R_i \geq \sum_{\mathcal{B}\subseteq\mathcal{A}} H(\hat{T}_{\mathcal{B}}|\hat{T}_{2^{\mathcal{B}}\setminus\mathcal{B}}) - H(\hat{T}_{2^{\mathcal{A}}}|T) \\ E[d_{\mathcal{A}}(T,\hat{T}_{\mathcal{A}})] \leq D_{\mathcal{A}},\ \forall \mathcal{A} \in 2^{[M]} \setminus \emptyset\Big\} \qquad (2.29)\end{aligned}$$

and

$$\mathcal{RD}_{in} = \text{conv}\left(\bigcup_{\hat{T}_{2^{[M]}\setminus\emptyset}\in\Phi(\hat{T}_{2^{[M]}\setminus\emptyset})} \mathcal{RD}(\hat{T}_{2^{[M]}\setminus\emptyset})\right)$$

Then, $\mathcal{RD}_{in}$ is an inner bound to the rate distortion region [45].

A recent paper by Chen [46] provided an outer bound showing that the generalized EGC region also gives the rate distortion region for the quadratic Gaussian M descriptions case if there are only distortion constraints for the decoder receiving all of the messages, and distortion constraints on the individual messages. An achievable rate distortion region containing more points than the region in [45] was presented in [47] when rate of each description is the same and reconstruction distortion depends only on the number of decriptions received. Generalizing the problem in [47], an approximate rate region for the quadratic Gaussian problem was presented in [48] when the distortions depend only on the number of descriptions received (rates can be different).

An interesting extension to this problem is to consider side information at the decoder. For the quadratic Gaussian 2-descriptions case, the complete rate distortion region was presented in [49] when the side information is the same at all decoders, and later this region was extended in [50] to the case where the decoders receiving individual descriptions have different side information and the decoder receiving both

descriptions have both side information.

Some attempt has been made to solve the problem when a subset of the decoders losslessy reproduce deterministic function of the descriptions. It was shown for the 2 descriptions case, if a decoder receiving one of the descriptions reproduce a deterministic function of the description, then the rate distortion region is given by the EGC region [51]. Yeung and Zhang considered a more general problem in [52] with multiple sources, multiple encoders and multiple decoders where each encoder has access to a certain subset of the sources, each decoder has access to a certain subset of the encoders, and each decoder reconstructs a certain subset of the sources almost perfectly. They derived inner and outer bounds for this problem when the source variables are independent [52]. In the special "symmetric" lossless case where the decoder receiving the subset of descriptions $\mathcal{A}$ must reproduce a collection of sources $U_1, \ldots, U_{|\mathcal{A}|}$, work by Roche [53] ($M = 3$ and independent case) and Yeung and Zhang [54] has determined the lossless rate region.

2.2.7 Successive Refinement Problem

Successive refinement problem is a special case of the multiple descriptions problem where a coarse description of the source is sent in the first stage, and if it is necessary more information is sent in the subsequent stages to refine the coarse description [21, 55]. In the successive refinement problem only the distortions of all descriptions (not any subset) available at each stage need to be considered, which makes this one a special case of the multiple descriptions problem. This problem can be modeled with one encoder and with the number of decoders equal to the number of refinement stages, where the decoder corresponding to a particular stage receives all descriptions up to that stage. Fig. 2.7 shows an encoder sending 2 successively refinable descriptions.

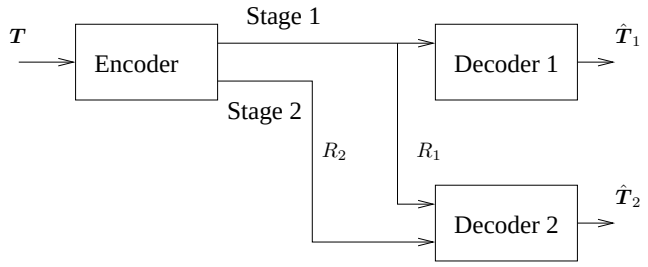


Figure 2.7: The 2-stage successive refinement problem.

The rate distortion vector $\{R_i, D_i\}_{i=1}^{M}$ is said to be achievable if there exist encoding functions

$$f_i : \boldsymbol{\mathcal{T}} \rightarrow \{1, \ldots, 2^{NR_i}\}, \quad \forall\ i \in \{1, \ldots, M\}$$

and decoding functions

$$g_i : \prod_{k=1}^{i} \{1, \ldots, 2^{NR_k}\} \rightarrow \hat{\boldsymbol{\mathcal{T}}}_i, \quad \forall\ i \in \{1, \ldots, M\}$$

such that the expected distortion

$$\frac{1}{N} \sum_{n=1}^{N} E[d(T^{(n)}, \hat{T}_i^{(n)})] \leq D_i, \quad \forall\ i \in \{1, \ldots, M\}$$

The rate distortion region for this problem is defined as the convex hull of all achievable $\{R_i, D_i\}_{i=1}^{M}$.

Equitz and Cover [21] derived the rate distortion region from the EGC region for a special case of the 2-stage successive refinement problem when $R_1 + R_2 = R(D_2)$, where $R(D_2)$ is the rate distortion function in (2.24). Note that this the "no-excess rate" case in the multiple descriptions problem and the EGC region provides the entire rate distortion region in this case. The region in [21] was later generalized to

the general 2-stage problem in [55] as

$$\begin{aligned}\mathcal{RD} \;=\; \text{conv}\Big\{(R_1,R_2,D_1,D_2):\; R_1 \geq I(T;\hat{T}_1),\; R_1 \geq I(T;\hat{T}_1,\hat{T}_2) \\ E[d(T;\hat{T}_1)] \leq D_1,\; E[d(T;\hat{T}_2)] \leq D_2\Big\}\end{aligned}$$

The rate distortion region for the 2-stage problem with decoder side information was proved in [22] and it was shown that the side information may be absent problem considered in [23] can be derived as a special case of this problem.

A source is said to be *successively refinable* if the descriptions at each stage can be refined to a distortion equals to the distortion rate function for the problem [21]. It has been shown that not every source (distribution) is successively refinable, and the necessary and sufficient conditions for successive refinability have been provided for different problems [21, 55, 22].

2.3 Practical Source Coding Theory

Practical source coding theory borrows ideas from machine learning and coding theory to design source codes that approach information theoretic bounds. The source code design for continuous sources involves two main steps: quantization, then compression.

The modern practical coding theory employs sophisticated scalar or vector quantizers to quantize the source and applies syndromes of powerful channel codes together with belief propagation decoders as a means of compressing the quantized source when side information is available at the decoder. In this section, we discuss the quantization and channel coding techniques that we apply in Chapter 5 for our practical code design. Such a technique is frequently referred to as entropy coded quantization [56].

2.3.1 Quantization

Continuous sources require an infinite number of bits to represent them losslessly. Since this is not possible in practice, continuous sources are compressed with some loss. One commonly used approach in lossy compression of continuous sources is to quantize the source up to a desired distortion, and then to losslessly compress the quantized source.

Scalar Quantizers

The simplest way to quantize a continuous source is to apply a scalar quantizer. In scalar quantization, we have a set of quatization points $\{q_1, \ldots, q_M\}$. A sequence $x^{(1)}, \ldots, x^{(N)}$ is quantized such that

$$\hat{x}^{(n)} = \arg \min_{q_i,\ i=1,\ldots,M} (x^{(n)} - q_i)^2, \quad n = 1, \ldots, N \tag{2.30}$$

where q_i is the ith quantization point. These quantization points are selected to minimize the mean squared error.

Vector Quantizers

Scalar quantization (SQ) has an important deficiency as it is applied to a source sequence, namely the Voronoi region induced by SQ is restricted to be cubic. Vector quantization (VQ) allows for Voronoi cells with any shape and some of the shapes, including spherical cells, are shown to give better distortion performance than that of the cubic Voronoi cells [57].

In vector quantization [57, 58], a sequence $\boldsymbol{x} = [x^{(1)}, \ldots, x^{(N)}]$ is quantized such that

$$\hat{\boldsymbol{x}} = \arg \min_{\boldsymbol{q}_\ell,\ \ell=1,\ldots,L} \frac{1}{N} \sum_{n=1}^{N} (x^{(n)} - q_\ell^{(n)})^2 \tag{2.31}$$

where L is the number of quantized sequences (codewords) and $q_\ell^{(n)}$ is nth element of ℓth quantized sequence $\boldsymbol{q}_\ell$. The set of quantized sequences for a vector quantizer can be selected through training by applying the generalized Lloyd algorithm [57]. This algorithm can be described as follows.

1. Select a set of training vectors $\boldsymbol{\mathcal{X}} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\}$ and an initial codebook with codewords $\mathcal{C} = \{\boldsymbol{q}_1, \ldots, \boldsymbol{q}_L\}$. Set the iteration number $j = 1$.

2. Quantize the training vectors with the codewords in $\mathcal{C}$. Let $\mathcal{B}_i = \{\boldsymbol{x} \in \boldsymbol{\mathcal{X}} : Q(\boldsymbol{x}) = \boldsymbol{q}_i\}$, where $Q(\boldsymbol{x})$ is the quantized sequence of $\boldsymbol{x}$.

3. Find the new codewords by

$$\boldsymbol{q}_i = \frac{1}{|\mathcal{B}_i|} \sum_{\boldsymbol{x} \in \mathcal{B}_i} \boldsymbol{x}, \quad \forall\ i = 1, \ldots, L$$

   where $|\mathcal{B}_i|$ is the cardinality of the set.

4. Compute the average distortion d_j. If $(d_{j-1} - d_j)/d_j < \epsilon$ stop; else set $j = j + 1$ and go to step 2.

Trellis Coded Quantization (TCQ)

Trellis Coded Quantization (TCQ) [59], the source coding counterpart of Trellis Coded Modulation (TCM) [60], is a type of vector quantization. The trellis structure determines the quantization sequences that are allowed in TCQ.

A r-bit TCQ uses 2^{r+1} quantization points. In general, these quantization points are partitioned into m cosets $(C^0, C^1, \ldots, C^m)$ by assigning the first quantization point from the left to C^0, the second quantization point to C^1 and so on. In this work, we partition the quantization points into 4 ($m = 4$) cosets and thus we explain TCQ for this special case.

The trellis for TCQ is generated using a rate 1/2 convolutional code. The amount of memory used in the convolutional code determines the number of states in the trellis. If M memory elements are used, there will be 2^M states in the trellis. The output of the convolutional code determines the coset selected. Thus, from a particular state only two of the four cosets can be selected depending on the input (0 or 1). This means that given the current state we need only 1 bit to represent the coset and $r-1$ bits to represent the quantization points within the coset, although we have 2^{r+1} quantization points.

In TCQ, a source sequence is quantized by applying Viterbi's algorithm on the trellis to obtain the quantized sequence with minimal distortion to the observed sequence [59]. Indeed, TCQ focuses on minimizing MSE for a given number of quantization points. A variant of TCQ, Entropy coded TCQ (ECTCQ), focuses on minimizing MSE for a given entropy, which allows encoding of trellis coded quantized sequences to their entropy [56, 57].

ECTCQ achieves distortions 0.2dB away from the distortion rate function for Gaussian sources and squared error distortion, which improves the performance of entropy coded SQ (ECSQ) by 1.33dB [57]. ECTCQ has also proved its potential in the Wyner-Ziv coding [29].

Successively Refinable TCQ (SR-TCQ)

We will need to generate successively refinable descriptions in Chapter 5. Although TCQ cannot be directly applied to generate successively refinable descriptions, an extension of TCQ called Successively Refinable TCQ (SR-TCQ) can be applied for this purpose [61]. We apply SR-TCQ to generate successively refinable descriptions in Chapter 5.

SR-TCQ [61] is described here for only 2 refinement stages since that is enough for

the reader to understand the work in this dissertation. Suppose that in the first stage the description is sent at rate r_1 and in the second stage the description is sent at rate r_2. Then, we need to have two sets of quantization points Q_1 and Q_2, one for each stage. The set Q_1 consists of 2^{r_1+1} quantization points, $Q_1 = \{q_i : i \in \{1, \ldots, 2^{r_1+1}\}\}$ and the set Q_2 consists of 2^{r_2+1} quantization points for each one of the quantization points $i \in Q_1$, i.e. $Q_2 = \{q_{i,j} : j \in \{1, \ldots, 2^{r_2+1}\},\ i \in \{1, \ldots, 2^{r_1+1}\}\}$. At both stages, the quantization points are partitioned into 4 cosets (C^0, C^1, C^2, C^3) in the same way it is done in the TCQ. Next, consider the construction of the trellis used in SR-TCQ.

The trellis for SR-TCQ is constructed by taking the tensor product of the trellises of the two refinement stages. In particular, the trellis for our problem is constructed by taking the tensor product $T_1 \otimes T_2$ of the first stage trellis T_1 and the second stage trellis T_2. Suppose that T_1 and T_2 have states $v_1, v_2, \ldots, v_{2^{n_1}}$ and $w_1, w_2, \ldots, w_{2^{n_2}}$, respectively. Then $T_1 \otimes T_2$ consists of $2^{n_1+n_2}$ states $v_i \otimes w_j, (i, j) \in \{1, 2, \ldots, 2^{n_1}\} \times \{1, 2, \ldots, 2^{n_2}\}$. There is a transition between states $v_i \otimes w_j$ and $v_k \otimes w_\ell$ in $T_1 \otimes T_2$ if and only if there is a transition between v_i and v_k in T_1, and there is a transition between w_j and w_ℓ in T_2. Denote the trellis constructed by the tensor product by $T_{1,2} \triangleq T_1 \otimes T_2$.

We apply a 2-stage SR-TCQ to generate two successively refinable quantized sequences, one of which takes values from Q_1 and the other one takes values from Q_2. In SR-TCQ, a source sequence is quantized by applying Viterbi's algorithm on the trellis $T_{1,2}$. When the Viterbi's algorithm is applied, the distortions of both of the sequences should be considered and this can be done by minimizing a weighted distortion D of the distortions of both sequences.

$$D = \alpha D_1 + (1 - \alpha) D_2 \tag{2.32}$$

where $\alpha,\ 0 \leq \alpha \leq 1$ is the distortion weighting factor, and D_1 and D_2 are the

distortions of the first and second stage quantized sequences, respectively.

Having discussed the techniques that can be applied to generate descriptions, we next discuss the techniques that can be used to compress those descriptions.

2.3.2 Channel Codes

Channel codes are originally intended for reliably transmitting the bits over channels by increasing redundancy. However, much of modern practical source coding theory transmits syndromes of the channel codes, treating the compressed sequence as the received sequence for a linear block code, to compress the sources when side information is available at the decoder [29, 27]. The side information provides a prior distribution for the source which is used, together with the syndrome, to determine the source sequence in a belief propagation decoder. We presently discuss this structure in more detail. We will discuss block codes since we are using syndromes of a block code for compression in Chapter 5.

Linear Block Codes

A (n, k) linear block code maps k information bits into $n(\geq k)$ coded bits. The rate of a (n, k) block code is k/n. The linear map can be described by a matrix called *Generator Matrix* ($\mathbf{G}_{k\times n}$) with rank k. A $1 \times k$ information bit vector $\mathbf{u}$ is mapped into the coded bits as follows.

$$\mathbf{c} = \mathbf{uG} \tag{2.33}$$

where $1 \times n$ vector $\mathbf{c}$ called a codeword. Any generator matrix can be written in systematic form.

$$\mathbf{G} = [\mathbf{I}_k|\ \mathbf{P}] \tag{2.34}$$

where $\mathbf{I}_k$ is a $k \times k$ identity matrix and $\mathbf{P}$ is a $k \times (n-k)$ matrix that generates parity bits.

For a generator matrix $\mathbf{G}$ in (2.34), the *parity check matrix* $\mathbf{H}$ is defined as

$$\mathbf{H} = [\mathbf{P}^T | \ \mathbf{I}_{n-k}] \tag{2.35}$$

The parity check matrix can be used to decode linear block codes. Let $\mathbf{r}$ be the received bits at the receiver.

$$\mathbf{r} = \mathbf{c} + \mathbf{e} \tag{2.36}$$

where $\mathbf{e}$ is $1 \times n$ error vector introduced by the channel. Then the *syndrome* of $\mathbf{r}$ is defined as

$$\mathbf{s} = \mathbf{H}\mathbf{r}^T \tag{2.37}$$

Note that when $\mathbf{e} = \mathbf{0}$

$$\mathbf{s} = \mathbf{H}\mathbf{c}^T = \mathbf{H}\mathbf{G}^T\mathbf{u}^T = \mathbf{0} \tag{2.38}$$

LDPC Codes

Low-density parity-check (LDPC) code is a linear block code with relatively low number of 1s compared to the number of 0s in its parity check matrix [62, 63, 64]. LDPC codes have been shown to perform very close to the Shannon limit when the length of the code is long.

Our interest in LDPC codes lies in using the syndromes of the LDPC codes to compress the bits and decoding the syndromes with the help of the side information at the decoder. Thus, we will demonstrate here how the syndromes are used to compress bits and how these syndromes are decoded with the help of side information in the belief propagation decoder.

Let $\mathbf{H}$ be a low-density parity-check matrix. The parity-check matrix $\mathbf{H}$ can be

represented by a factor graph, by representing the columns with the variable nodes and the rows with the factor (check) nodes, and then connecting the rows and columns of 1s in $\mathbf{H}$ by edges. In this representation, the parity-check matrix of a LDPC code can be specified using two *degree distribution*s, variable node and check node degree distributions, where a *degree distribution* is defined as a polynomial $\gamma(x)$ [64]

$$\gamma(x) := \sum_{i \geq 2} \gamma_i x^{i-1} \tag{2.39}$$

with nonnegative coefficients and $\gamma(1) = 1$. The variable node degree distribution $\lambda(x)$ is defined as

$$\lambda(x) = \sum_{i=2}^{d_v} \lambda_i x^{i-1} \tag{2.40}$$

where d_v is the maximum number of edges connected to a variable node and λ_i is

$$\lambda_i = \frac{n_i \; i}{N} \tag{2.41}$$

where n_i is the number of variable nodes with i edges and N is the total number of edges. Similarly, the check node degree distribution $\rho(x)$ is defined as

$$\rho(x) = \sum_{i=2}^{d_c} \rho_i x^{i-1} \tag{2.42}$$

where d_c is the maximum number of edges connected to a check node and ρ_i is defined in the same way λ_i was defined. For a given code length n and code rate k/n, the degree distributions can be optimized to obtain a good parity check matrix [64].

We presently discuss compression of bits at the encoders. Let $\mathbf{r}$ be a $n \times 1$ binary vector that need to be compressed. Then, the encoder sends the syndromes $\mathbf{s}$ to the decoder.

$$\mathbf{s} = \mathbf{H}\mathbf{r} \tag{2.43}$$

Note that $\mathbf{s}$ can be non-zero, because all length n binary sequences are allowed for $\mathbf{r}$.

Given the side information $\mathbf{y}$ (can be non-binary) and syndromes $\mathbf{s}$, the goal of the decoder is to decode the compressed bits such that

$$\hat{\mathbf{r}} = \arg \max_{\tilde{\mathbf{r}}} \ P(\tilde{\mathbf{r}}|\ \mathbf{y}, \mathbf{s}) \tag{2.44}$$

This decoding can be effectively done using message-passing algorithms such as belief propagation (BP). We describe belief propagation decoding of the syndromes here as we will use this decoding algorithm in Chapter 5.

When $(\mathbf{r}, \mathbf{y})$ are i.i.d. according to $p(r, y)$, the joint conditional distribution $P(\mathbf{r}, \mathbf{s}|\mathbf{y})$ can be written as

$$\begin{aligned} P(\mathbf{r}, \mathbf{s}|\mathbf{y}) &= P(\mathbf{r}|\mathbf{y})\ P(\mathbf{s}|\mathbf{r}) \\ &= P(\mathbf{r}|\mathbf{y})\ \mathbf{1}[\mathbf{Hr} = \mathbf{s}] \\ &= \prod_n P(r_n|y_n) \prod_m \mathbf{1}[\sum_{n \in \mathcal{N}(m)} r_n = s_m] \end{aligned} \tag{2.45}$$

where $\mathbf{1}$ is indicator function and $\mathcal{N}(m)$ is the set of neighbors of s_m.

We can use a factor graph to represent this function by representing r_n with variable nodes and the factorized functions with factor nodes. Then, BP algorithm can be applied to compute the marginal probabilities $P(r_n|\mathbf{y}, \mathbf{s})$ as described in 2.1.2. The bit r_n is then decoded from the approximate marginals as

$$\hat{r}_n = \arg \max_{i=0,1} \ P(r_n = i|\mathbf{y}, \mathbf{s}) \tag{2.46}$$

Having provided the background necessary to develop collaborative estimation algorithms, we begin developing the algorithms in the next chapter.

3. Low-Complexity Collaborative Estimation Algorithms

In this chapter we study low-complexity collaborative estimation [65] algorithms for networks. In particular, we develop an algorithm for collaborative estimation of channel gains in wireless sensor networks [66, 67]. As it was discussed in Chapter 1, energy consumption is a key issue in wireless sensor networks [1, 68]. While part of the energy in the sensors is spent on processing data, a sizable portion of their energy is expended on communication because of the necessary power amplification of the communications signals. This energy consumption for communications purposes can be minimized, maximizing the communications energy efficiency of the network, through distributed power control [69] if the network nodes are aware of the link gains on the network's wireless channels.

However, in many cases the sensors are deployed randomly, for instance by dropping them out of the back of a plane, and, thus, they do not initially know their positions, neighbors, or channel gains. Thus, they must first estimate the channel gains in order to determine their neighbors and to minimize transmission powers. During this initial channel gain estimation phase, power consumption may be further reduced by duty cycling [70, 71], i.e. keeping only a small subset of the sensors in a high power "awake" state at each time instant.

Following these practical constraints, we derive a low-complexity collaborative estimation algorithm from the Expectation Propagation (EP) [9] principle for a wireless sensor network in which each sensor estimates the channel gains by collaborating with a few other network nodes. While performing this channel estimation we maintain a low average network energy consumption by employing a random sleep strategy. We apply our algorithm and the diffusion Least-Mean Squares (LMS)[4] algorithm to the same channel estimation problem and compare their performance in terms of

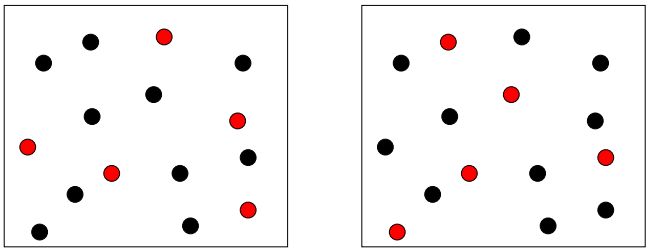
estimation error.

3.1 Problem Formulation

Consider a network of N sensor nodes $1, \ldots, N$ which are randomly placed on a flat terrain to monitor a common phenomenon. Assuming symmetry of the link between two nodes, let $\mathbf{h} := [h_{i,j} \mid i, j \in \{1, 2, \ldots, N\}, i < j]$ be the set of channel gains in the network, where $h_{i,j}$ is the gain of the link between nodes i and j. The goal of this work is to estimate at each node the length $N(N-1)/2$ channel gain vector $\mathbf{h}$ by collaborating with a few other nodes and applying distributed Bayesian estimation techniques.

To reduce the power consumption during the channel estimation phase, we apply to the network a regular cyclic random sleep strategy [72], in which at each discrete time instant a randomly selected collection of d nodes are awake and each sensor maintains the same average power consumption. Each sleep cycle consists of K such discrete time instants after which the cycle repeats. Thus, if we denote the set of nodes awake at time instant k with $\mathcal{S}(k), k \in \{1, \ldots, K\}$, then $\mathcal{S}(K+k) = \mathcal{S}(k)$. Fig. 3.1 shows two different random set of nodes which are awake during two different sleep cycle instants $k_1, k_2 \in \{1, \ldots, K\},\ k_1 \neq k_2$. In Fig. 3.1, the awake nodes are shown in red. Next denote the number of times one node is awake during a sleep cycle with c, which we require to be the same for all nodes in order to maintain equal power consumption throughout the network and thus equal node lifetime. Then, the total number of time instants in a sleep cycle is $K = \frac{c}{d}N$.

To the network model which we described above, we employ a typical wireless communication channel estimation technique, *channel training*, to estimate the channel gains of the links in the network. For communications between the nodes during the training phase and the channel estimation phase which follows the training phase,



(a) Sleep cycle instant k_1 (b) Sleep cycle instant k_2

Figure 3.1: Random set of (red) sensor nodes are awake during two different sleep cycle instants.

we use TDMA based medium access control [73] which increases the energy savings by avoiding collisions and retransmissions. These energy savings result at the cost of synchronization which could be achieved using the schemes proposed in the literature [74] [75] [76] [77].

To implement the TDMA based medium access control, we further divide each sleep cycle time instant k into more time slots. During each of the first c of these slots, each awake node takes turns transmitting its training sequence while all other awake nodes record their observations. For example, Fig. 3.2 depicts a node transmitting its training sequence at the first time slot while the other nodes which are awake during that sleep cycle instant are listening to it. The remaining slots of a sleep cycle time instant are used for the nodes to exchange estimate information in a manner to be described momentarily.

Now consider the first sleep cycle. Suppose that node i is awake at sleep cycle instant k and it transmits its training sequence $\mathbf{u}_i = [u_{i,1}, \ldots, u_{i,M}]$ during its turn, where M is the length of the training sequence. Then, each node $j \in \mathcal{S}(k) \setminus \{i\}$ records its observation. We model the observation $r_{k,j,i,m}$ made for the symbol $u_{i,m}$ at the node j as a function of the channel gain $h_{i,j}$ of the link between nodes i and j

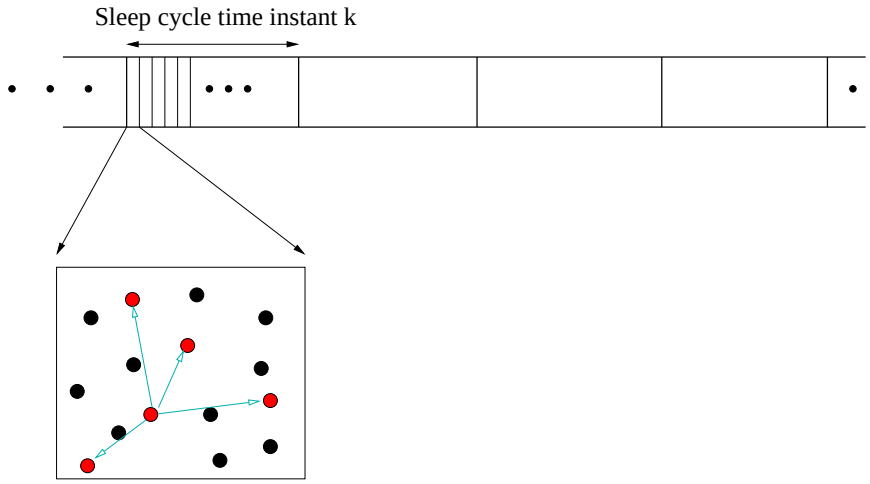


Figure 3.2: One of the awake nodes during sleep cycle instant k transmits its training sequence in the first time slot.

as

$$r_{k,j,i,m} = h_{i,j}u_{i,m} + v_{k,j,i,m} \tag{3.1}$$

where $m \in \{1, \ldots, M\}$ and $v_{k,j,i,m}$ is noise which is assumed to be spatially and temporally independent and to be Gaussian distributed with mean zero and variance σ_N^2. Collect all observations made by node j for symbols $\mathbf{u}_i$ during sleep cycle instant k into a vector $\mathbf{r}_{k,j,i} := [r_{k,j,i,m} |\ m \in \{1, \ldots, M\}]$ and define the following.

$$\mathbf{r}_{k,j} := [\mathbf{r}_{k,j,i} \mid i \in \mathcal{S}(k) \setminus j] \tag{3.2}$$

$$\mathbf{r}_{k} := [\mathbf{r}_{k,i} \mid i \in \mathcal{S}(k)] \tag{3.3}$$

Note that because of the random sleep strategy we use, a node which is awake during a sleep cycle instant k can gather information about only the links with the other nodes that are awake at that particular time instant.

Next each node processes the information gathered and disseminates processed information in the following sleep cycles in hopes of helping other nodes to obtain

better estimates of the channel gains. How the information is processed or what information is disseminated at each sleep cycle depends on the collaborative estimation algorithm used. When the processed information is disseminated, due to the limited computational abilities we assume that the nodes will take time on the order of the amount of an entire sleep cycle to decode the information (messages are encoded because they are to be sent over noisy channels) and to use it to encode any outgoing information. Therefore, after a few complete sleep cycles a sensor node will only have an opportunity to obtain information from only those nodes that can be communicated with directly or indirectly(through other nodes) within that number of sleep cycles. To derive a collaborative estimation algorithm for this problem, we first consider the raw information that would be available at each node after certain number of sleep cycles if the information were disseminated without processing. Denote the raw information (the observations) that would be available at node i after ℓ sleep cycles with $\mathbf{r}(\mathcal{T}(i,\ell))$, i.e.

$$\mathbf{r}(\mathcal{T}(i,\ell)) := \{\mathbf{r}_k | k \in \mathcal{T}(i,\ell)\} \tag{3.4}$$

where we define $\mathcal{P}(i,\ell)$ as the set of nodes j with which node i can communicate directly or indirectly after ℓ complete sleep cycles and $\mathcal{T}(i,\ell)$ as

$$\mathcal{T}(i,\ell) := \{k|\ j \in \mathcal{S}(k) \text{ and } j \in \mathcal{P}(i,\ell)\} \tag{3.5}$$

Then, after ℓ complete sleep cycles, each node i estimates its channel gains by applying Bayesian estimation techniques, which effectively use the prior information of the channel gains together with the information $\mathbf{r}(\mathcal{T}(i,\ell))$ received from the other nodes. In particular, each node i computes its MMSE estimates of the channel gains

as

$$\hat{\mathbf{h}}_i = \int \mathbf{h}\ p_{\mathbf{h}|\mathbf{r}(\mathcal{T}(i,\ell))}\ d\mathbf{h} \tag{3.6}$$

As we see from (3.6), each node needs the posterior distribution $p_{\mathbf{h}|\mathbf{r}(\mathcal{T}(i,\ell))}$ to compute the MMSE estimates of the channel gains. We derive an algorithm from expectation propagation (EP) [34, 9, 72] principle in Section 3.3 which provides efficient means for the nodes to disseminate the information that is needed to compute the posterior distribution $p_{\mathbf{h}|\mathbf{r}(\mathcal{T}(i,\ell))}$ and in turn the channel gain estimates at each node. The prior information used by this MMSE estimator is due to the path loss effect and we first show in the next section how one can obtain the prior information on the channel gains which is due to the path loss effect.

3.2 Prior Information on the Channel Gains

Statistical channel modeling studies have consistently shown that the channel gain on a link depends on 3 components: path loss, large-scale shadowing and small-scale fading [78] [79] [80] [81] [82]. The small scale fading phenomenon refers to fast variations of the received power around a nominal average power which are caused primarily by the constructive and deconstructive interference of different multipath components arriving at mobile receiver [83] [84] [82]. This fast fading, which is less important in immobile scenarios such as the one considered here, can be compensated for using channel coding if the average link gain dictated by the path loss and large scale shadowing effects can be determined. The average link gain, henceforth referred to as the channel gain in this chapter, can in turn be estimated using periodic channel training on a link by link basis as described in the previous section. Since this average link gain is primarily determined by the path loss and large scale shadowing effects, distributions on these quantities obtained by numerous channel measurement campaigns provide significant prior knowledge about the channel gains to be estimated,

as we point out presently for the path loss component.

Path loss models capture the dependence of the channel gains on the distance between transmitter and receiver. In particular, in path loss models the channel gain between two nodes separated by a distance of R is deemed proportional to R^{-n}, where n is known as the path loss exponent. Many measurement campaigns have shown that depending on the nature of the ground on which the network lies, the path loss exponent varies between 2 and 6 [78] [79] [80] [81] [82]. We have chosen a path loss exponent of 4 for our work, although our analysis is amenable to other exponents and unknown exponents as well, as simulation results will later show. For the purposes of prior information for our estimation algorithm, we then model the channel gain $h_{i,j}$ between two nodes i and j as

$$h_{i,j} \propto \|\boldsymbol{\chi}_i - \boldsymbol{\chi}_j\|_2^{-2} \tag{3.7}$$

where $\boldsymbol{\chi}_i$ and $\boldsymbol{\chi}_j$ are the positions of the nodes i and j, respectively.

The influence of this path loss effect has significant implications for channel estimation when viewed from a network standpoint which are far less important when channel estimation is viewed from a single link perspective (as is traditionally the case). To see this, observe that if the problem of channel estimation is viewed as a single node problem, in which each node in the network estimates only those channel gains incident on it, and then disseminates this knowledge throughout the network, each network node would be estimating $\leq N-1$ channel gains. The path loss phenomenon dictates that these gains are heavily influenced (together with large scale shadowing effects) by the positions of the sensor nodes involved, which, if the nodes are assumed to lie on a flat plain, can be specified using $2N$ real numbers (e.g. Cartesian coordinates). The number of parameters dictating these positions is larger than the number of channel gains that any one node will estimate in an uncoordinated

single node approach. Thus, it is unlikely that a path loss model will provide any useful prior information for channel gain estimation carried out at a single network node, since this means the number of unknown parameters in the prior (the positions) is far larger than the number of gains to estimate.

However, when the channel estimation problem is viewed from a global network coordinated perspective, the situation changes significantly. There are a total of $N(N-1)$ (or $\frac{N(N-1)}{2}$ depending on whether symmetry is assumed) channel gains throughout the network, while all of the node positions are specified with only $2N$ real numbers (Cartesian coordinates). In this instance, the prior information offered by the path loss phenomenon is significant. Namely, the prevalence of path loss models dictates that the $N(N-1)$ channel gains are heavily biased (albeit not entirely determined by) by a model dependent on just $2N$ parameters (the node positions). Even for moderate N, that a $\frac{N(N-1)}{2}$ variate model is largely determined by $2N$ parameters is significant. In particular, the path loss phenomenon dictates that the $N(N-1)$ dimensional vector of all channel gains in the network will live in a set that is highly concentrated around a $2N$ dimensional manifold in $\mathbb{R}^{N(N-1)}$.

In fact, even if the positions of the nodes are not known, the path loss phenomenon provides significant prior information for the network channel estimation problem. To see this, suppose that the nodes are placed randomly and independently of one another, and these positions are unknown. Then consider two links which are incident on a common node i. The gains of these two links, $h_{i,j}$ and $h_{i,m}$, are functions of node positions $(\boldsymbol{\chi}_i, \boldsymbol{\chi}_j)$ and $(\boldsymbol{\chi}_i, \boldsymbol{\chi}_m)$, respectively. Clearly the random variables $h_{i,j}$ and $h_{i,m}$ are dependent because they are functions of a common random variable $\boldsymbol{\chi}_i$. Now, consider two links that are not incident on a common node. The gains of these links, $h_{i,j}$ and $h_{m,n}$, are functions of node positions $(\boldsymbol{\chi}_i, \boldsymbol{\chi}_j)$ and $(\boldsymbol{\chi}_m, \boldsymbol{\chi}_n)$, respectively. Since $h_{i,j}$ and $h_{m,n}$ are functions of independent variables and these functions do not

have a random variable in common, they are independent of one another. Thus, in a path loss dominated regime if the nodes are placed randomly and independently of one another, and these positions are unknown, any two channel gains incident on a common network node are statistically dependent. Conversely, any two channel gains which do not share any common network nodes are statistically independent. This knowledge may be expressed in terms of a prior distribution for the network channel gains.

The prior distribution of the channel gains depends on the distribution of the node positions which governs the random placement of the nodes. For the purpose of selecting a distribution for the node positions, we specify the random node positions by Cartesian coordinates in $\mathbb{R}^2$ space, the origin of which is taken to be the position of the common phenomenon. For example, the position of node i is specified by $\boldsymbol{\chi}_i \triangleq (x_i, y_i)$. We need to keep in mind few things when selecting a suitable distribution for the coordinates. The random coordinates can take continuous values; however, practically there must be a minimum separation between any two nodes. Also ideally, we would want more sensors to be placed near the phenomenon to be monitored and fewer sensors to be placed far from the phenomenon to be monitored. Considering these facts, we choose the sensor positions $\{\boldsymbol{\chi}_1, \ldots, \boldsymbol{\chi}_N\}$ to be i.i.d. according to a Gaussian distribution satisfying a minimum separation between any two nodes, although other position distributions may be equally viable and will also be amenable to our analysis. This prior distribution is both analytically complex and intractable due to the inverse nonlinear dependence on the random node positions. For that reason, we employ an approximate inference algorithm, EP, to approximate this distribution with a tractable distribution.

The ultimate aim of this work is to demonstrate that the network channel gains can be estimated by exploiting this prior information along with the information received

from standard channel training techniques discussed in the previous section. While the phenomenon of large scale shadowing also provides significant prior information which could be exploited in estimating the average link gains throughout the network, we start for the sake of simplicity with only the prior information afforded by path loss. As will be evidenced in the simulations, which include both shadowing and path loss exponent mismatch, significant estimation performance improvement can be obtained by incorporating prior information into the estimator due to path loss effects alone.

In the next section, we derive an algorithm from EP which effectively uses this prior information together with the information obtained from the channel training to estimate the channel gains (average link gains) in the network.

3.3 Distributed Estimation with Expectation Propagation

The prior information used for this collaborative channel estimation is both analytically complex and intractable because of the inverse nonlinear dependence on the node positions as in (3.7). EP is applied in this inference problem after approximating the complex nonlinear joint prior distribution for the channel gains with a Gaussian distribution. Under this Gaussian approximation, exact statistical inference with belief propagation [32] or expectation propagation can be performed, provided the associated approximated factor graph, which is to be defined momentarily, is without loops.

3.3.1 Gaussian Approximation of the Prior Distribution

Presently we provide the specific information about this Gaussian approximation. We first approximate the distribution of the channel gains in dB with a Gaussian distribution with the same mean and covariance for tractability. Then we show that

the distribution of the channel gains in the linear scale can also be approximated to a Gaussian distribution.

To see this, denote the channel gains in dB with $\mathbf{h}_{dB}$. Suppose that the mean and covariance of the channel gains in dB are $\mathbf{m}$ and Σ, respectively. Then, we can write the approximate distribution of $\mathbf{h}_{dB}$ as

$$\mathbf{h}_{dB} \sim \mathcal{N}(\mathbf{m}, \mathbf{\Sigma})$$

Furthermore, we can write $\mathbf{h}_{dB}$ as

$$\mathbf{h}_{dB} = \mathbf{m} + \mathbf{w} \tag{3.8}$$

where $\mathbf{w}$ is a random vector with distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$. We can now write the channel gains $\mathbf{h}$ as

$$\begin{aligned} \mathbf{h} &= 10^{(\mathbf{m}/10 + \mathbf{w}/10)} \\ &= 10^{\mathbf{m}/10}\, e^{(\ln(10)/10)\mathbf{w}} \end{aligned} \tag{3.9}$$

where element-wise operation exp and element-wise multiplication is implied. Approximating the term $e^{(\ln(10)/10)\mathbf{w}}$, the channel gains linear scale $\mathbf{h}$ can be written as

$$\mathbf{h} \approx 10^{\mathbf{m}/10} \left(1 + \frac{\ln(10)}{10}\mathbf{w}\right) \tag{3.10}$$

Thus the prior distribution of the channel gains can be approximated as

$$\mathbf{h} \sim \mathcal{N}\Big(10^{\mathbf{m}/10}, \Big(\frac{\ln(10)}{10}\Big)^2 \mathrm{diag}(10^{\mathbf{m}/10})\, \mathbf{\Sigma}\, \mathrm{diag}(10^{\mathbf{m}/10})\Big) \tag{3.11}$$

Even though both the initial log-normal approximation and then the normal approxi-

mation are very coarse, they make the ultimate inference algorithm tractable. We will observe in Section 3.5 via simulation that these coarse approximations still provide sufficient prior information to greatly enhance channel estimate performance over an algorithm not employing any use of prior information.

3.3.2 Factor Graph and Expectation Propagation

We presently illustrate how EP can be applied to this inference problem under the Gaussian approximation by associating the inference problem to a probabilistic graphical model. Suppose that each node $i \in \{1, 2, \ldots, N\}$ in the network has an estimate $\mathbf{h}_i$ of $\mathbf{h}$ and all nodes have the same prior information, i.e. $p_{\mathbf{h}}(\mathbf{h}_i) = p_{\mathbf{h}}(\mathbf{h}_j)$ for all i, j.

We write a joint distribution indicating the information available to node i after ℓ complete sleep cycles as

$$p_{\mathbf{r}(\mathcal{T}(i,\ell)),\mathbf{h},\mathbf{h}(\mathcal{P}(i,\ell))} = \prod_{k\in\mathcal{T}(i,\ell)} p_{\mathbf{r}_k|\mathbf{h}} \prod_{j\in\mathcal{P}(i,\ell)} \delta(\mathbf{h}-\mathbf{h}_j)(p_{\mathbf{h}}(\mathbf{h}_j))^{\frac{1}{g(\ell)}} \tag{3.12}$$

where δ is the point mass distribution at zero and $g(\ell) = c(c-1)^{\ell}(d-1)^{\ell}$ is the number of sensor nodes with which node i can communicate directly or indirectly after ℓ complete sleep cycles. Also, we define $\mathbf{h}(\mathcal{P}(i,\ell))$ as

$$\mathbf{h}(\mathcal{P}(i,\ell)) := \{\mathbf{h}_j | j \in \mathcal{P}(i,\ell)\} \tag{3.13}$$

and $\mathbf{r}(\mathcal{T}(i,\ell))$ as in (3.4). Note that in (3.12), we have used the fact that the observations $\mathbf{r}_k$ collected at different time instants are independent given the channel gains, because all of the training sequences in the network are known ahead of time at each node.

If the posterior distribution $p(\mathbf{h}(\mathcal{P}(i,\ell))|\ \mathbf{r}(\mathcal{T}(i,\ell)))$ is approximated by applying

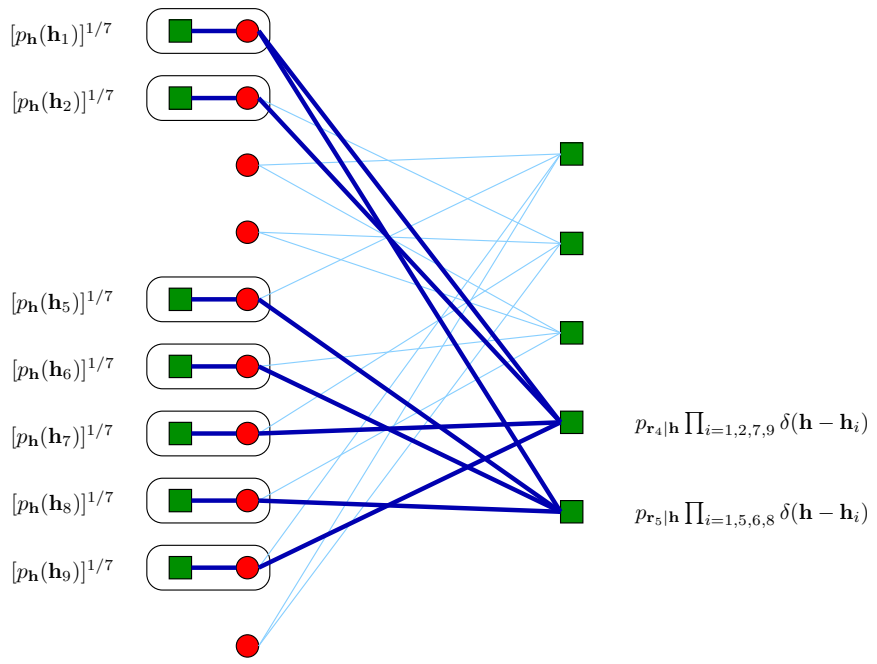


Figure 3.3: An example factor graph used for EP based channel estimation with only one sleep cycle ($l = 1$)

EP using the set of approximating distributions $q(\mathbf{h}(\mathcal{P}(i,\ell))$ taking the form

$$q(\mathbf{h}(\mathcal{P}(i,\ell)) \propto \prod_{j \in \mathcal{P}(i,\ell)} q(\mathbf{h}_j) \tag{3.14}$$

with a Gaussian initialization to all $q(\mathbf{h}_j)$, then the optimal solution for each factor $q(\mathbf{h}_j)$ is given by the corresponding marginal of $p(\mathbf{h}(\mathcal{P}(i,\ell))|\ \mathbf{r}(\mathcal{T}(i,\ell)))$ [3]. In such a case, the expectation propagation reduces to the loopy belief propagation as explained in Section 2.1.3. These optimal solutions, the marginals of $p(\mathbf{h}(\mathcal{P}(i,\ell))|\ \mathbf{r}(\mathcal{T}(i,\ell)))$, can be computed by associating the corresponding joint distribution with a factor graph [33].

For that reason, we now associate the model in (3.12) with an approximate factor graph as shown in Fig. 3.3. We will call it an approximate factor graph since

it represents an approximate joint distribution. Let us represent the sensor nodes $1, \ldots, N$ with the left side nodes (variable nodes) of the factor graph and the time instants $1, 2, \ldots, K$ of the random sleep cycle with the right side nodes (factor nodes) of the factor graph. We use an edge to connect the right node $i \in \mathcal{S}(k)$, which is awake during the sleep cycle instant k, with k^{th} left node in the factor graph which corresponds to the k^{th} sleep cycle instant.

Each node in the network, knowing the prior statistics on the channel gains, has an initial estimate of channel gains $\mathbf{h}_i = \mathbf{m_h}$, where $\mathbf{m_h} = 10^{\mathbf{m}/10}$. They can update their estimates by updating the statistics (mean and covariance) of the gains using the observations they made during the training phase. Once we have associated the joint distribution on the channel gains $\mathbf{h}$ and the observations $\mathbf{r}$ with a factor graph as shown in Fig. 3.3, we can apply EP [9] [72] to compute the posterior distribution of $\mathbf{h}$. When the conditions for belief propagation (BP) [32] are satisfied, exact statistical inference with expectation propagation (EP) can be performed, provided the associated factor graph is without loops.

The computation of the posterior distribution can be described as message passing on the factor graph as in Section 3.3.3. Equivalently, this computation can be explained in terms of real information exchange between the sensor nodes as in Section 3.3.4.

3.3.3 Message Passing on Factor Graphs

The computation of these marginals could be described using the sum-product algorithm [33] which passes messages along the edges of the factor graph to calculate beliefs as described in Section 2.1.2. By a "message" we mean an appropriate description of the corresponding function and by a "product of messages" we mean an appropriate description of the product of corresponding functions.

We next determine an appropriate description for the functions connected to our factor graph. When the functions connected to the factor graph are from exponential families (even if from different families), EP enables us to choose the "message" family from a common exponential family as illustrated in [72]. The functions represented by the factor graph are the approximated prior joint distribution $p_{\mathbf{h}}$ of the channel gains and the conditional joint distributions $p_{\mathbf{r}_k|\mathbf{h}}$ of the observations. Consider first the approximated prior joint distribution $p_{\mathbf{h}}$ which is given by

$$p_{\mathbf{h}}(\mathbf{h}) \propto \exp\{-\frac{1}{2}[(\mathbf{h}-\mathbf{m}_{\mathbf{h}})^T\mathbf{\Sigma}_{\mathbf{h}}^{-1}(\mathbf{h}-\mathbf{m}_{\mathbf{h}})]\} \tag{3.15}$$

where $\mathbf{m}_{\mathbf{h}}$ is the mean and $\mathbf{\Sigma}_{\mathbf{h}}$ is the covariance. Consider next the conditional joint distribution on the observations $\mathbf{r}_{k,i}$ collected during sleep cycle instant k at node $i \in \mathcal{S}(k)$ which is given by

$$p_{\mathbf{r}_{k,i}|\mathbf{h}}(\mathbf{r}_{k,i}|\mathbf{h}) \propto \exp\{-\frac{1}{2}[(\mathbf{r}_{k,i}-\mathbf{m}_{\mathbf{r}_{k,i}|\mathbf{h}})^T\mathbf{\Sigma}_{\mathbf{r}_{k,i}|\mathbf{h}}^{-1}(\mathbf{r}_{k,i}-\mathbf{m}_{\mathbf{r}_{k,i}|\mathbf{h}})]\} \tag{3.16}$$

where

$$\begin{aligned}\mathbf{m}_{\mathbf{r}_{k,i}|\mathbf{h}} &:= [h_{i,j}\mathbf{u}_j | j \in \mathcal{S}(k) \setminus \{i\}] \\ \mathbf{\Sigma}_{\mathbf{r}_{k,i}|\mathbf{h}} &:= \sigma_N^2\mathbf{I}_{(d-1)M\times(d-1)M}\end{aligned}$$

where σ_N^2 is noise variance. Since both distributions are exponential family distributions (Gaussian), they can be easily parameterized. This enables us to select the sufficient statistics of the exponential family distributions to be

$$\mathbf{v}(\mathbf{h}) = \begin{pmatrix}\mathbf{h}_y & \mathbf{h}_z & \mathbf{h}\end{pmatrix}^T \tag{3.17}$$

where

$$\mathbf{h}_y := [h_{i,j}^2 | i,j \in \{1,\ldots,N\}, i<j] \tag{3.18}$$

$$\mathbf{h}_z := [h_{i,j}h_{m,n} | i,j,m,n \in \{1,\ldots,N\}, i<j, m<n, m>i] \tag{3.19}$$

We can rewrite the prior distribution in terms of parameterization of the message exponential family as

$$p_{\mathbf{h}}(\mathbf{h}) \propto \exp\{-\frac{1}{2}(\mathbf{v}(\mathbf{h}).\boldsymbol{\tau} + {\mathbf{m}_\mathbf{h}}^T \boldsymbol{\Sigma}_\mathbf{h}^{-1}\mathbf{m}_\mathbf{h})\} \tag{3.20}$$

where the parameter vector $\boldsymbol{\tau}$ is

$$\boldsymbol{\tau} = \begin{pmatrix} \mathbf{a}_y & 2\mathbf{a}_z & -2\boldsymbol{\Sigma}_\mathbf{h}^{-1}\mathbf{m}_\mathbf{h} \end{pmatrix}^T \tag{3.21}$$

where $\boldsymbol{\Sigma}_\mathbf{h}^{-1} = [a_{i,j}]_{\frac{1}{2}N(N-1)\times\frac{1}{2}N(N-1)}$ and

$$\mathbf{a}_y := [a_{i,i} | i \in \{1,\ldots,\frac{1}{2}N(N-1)\}]$$
$$\mathbf{a}_z := [a_{m,n} | m,n \in \{1,\ldots,\frac{1}{2}N(N-1)\}, n>m]$$

We can also rewrite the conditional joint distribution on the observations as

$$p_{\mathbf{r}_{k,i}|\mathbf{h}}(\mathbf{r}_{k,i}|\mathbf{h}) \propto \exp\{-\frac{1}{2\sigma_N^2}(\mathbf{v}(\mathbf{h}).\mathbf{t}_{k,i} + \mathbf{r}_{k,i}^T\mathbf{r}_{k,i})\} \tag{3.22}$$

where

$$\mathbf{t}_{k,i} = \begin{pmatrix} \boldsymbol{\nu}_{k,i} & \mathbf{0} & \boldsymbol{\mu}_{k,i} \end{pmatrix}^T \tag{3.23}$$

where

$$\begin{aligned}
\boldsymbol{\nu}_{k,i} &:= [\mathbf{u}_n^T \mathbf{u}_n \delta(i-m)\delta(j-n) | m,n \in \{1,\ldots,N\}, \\
&\quad m < n \,\text{if}\, i < j, m > n \,\text{if}\, i > j,\ j \in \mathcal{S}(k) \setminus \{i\}] \\
\boldsymbol{\mu}_{k,i} &:= [-2\mathbf{u}_n^T \mathbf{r}_{k,m,n} \delta(i-m)\delta(j-n) | m,n \in \{1,\ldots,N\}, \\
&\quad m < n \,\text{if}\, i < j, m > n \,\text{if}\, i > j,\ j \in \mathcal{S}(k) \setminus \{i\}]
\end{aligned}$$

Here note that each vector in $\mathbf{t}_{k,i}$ is of the same length as the corresponding vectors in $\mathbf{v}(\mathbf{h})$.

It is useful to note an important property of exponential family distributions before we continue. Consider a set of distributions $\{p_{\boldsymbol{\theta}_i|\mathbf{h}} | \, i \in \{1,\ldots,L\}\}$, in which each distribution takes the form

$$p_{\boldsymbol{\theta}_i|\mathbf{h}} \propto \exp\{-\frac{1}{2}[\mathbf{v}(\mathbf{h}).\mathbf{f}_i(\boldsymbol{\theta}_i) - \mathbf{w}_i(\boldsymbol{\theta}_i)]\} \quad \forall i \in \{1,\ldots,L\} \tag{3.24}$$

Then, the product of the distributions can be written as

$$\prod_{i=1}^{L} p_{\boldsymbol{\theta}_i|\mathbf{h}} \propto \exp\{-\frac{1}{2}[\mathbf{v}(\mathbf{h}).\sum_{i=1}^{L} \mathbf{f}_i(\boldsymbol{\theta}_i)]\} \tag{3.25}$$

This special property of the exponential family distribution makes the calculation of the messages easy. In particular, the appropriate description of the product of messages (functions) is the summation of the parameters of corresponding function. Thus, variable node i computes the message $\phi_{i\to k}^{(p)}$ to factor node k during sleep cycle p in terms of the messages $\varphi_{k'\to i}^{(p-1)}$ received from the factor nodes during sleep cycle $p-1$ as

$$\phi_{i\to k}^{(p)} = \sum_{k' \in \mathcal{N}(i)\setminus\{k\}} \varphi_{k'\to i}^{(p-1)} \tag{3.26}$$

where $\mathcal{N}(i) := \{k | i \in \mathcal{S}(k)\}$. Factor node k computes the message $\varphi_{k\to i}^{(p)}$ to variable node i during sleep cycle p in terms of the messages $\phi_{i'\to k}^{(p-1)}$ received from the variables nodes during sleep cycle $p-1$ and the corresponding factor as

$$\varphi_{k\to i}^{(p)} = \mathbf{t}_k + \sum_{i' \in \mathcal{S}(k)\setminus\{i\}} \phi_{i'\to k}^{(p-1)} \tag{3.27}$$

where $\mathbf{t}_k := \sum_{i\in\mathcal{S}(k)} \mathbf{t}_{k,i}$ and $\mathbf{t}_{k,i}$ is defined in (3.23). The number of iterations ℓ that EP is to be run is decided ahead of time and the message passing is initialized by setting $\phi_{i\to k}^{(0)} = \boldsymbol{\tau}/g(\ell)$. At the final iteration, variable node i sums its incoming messages to get the parameters corresponding to the approximated posterior distribution. We explained the computation of the posterior distribution on the factor graph treating the factor nodes as if they were processors. Although this is mathematically correct, there are no such processors in reality. Since all the processors are located at the sensor nodes in reality, it is interesting to see the exchange of information between the sensor nodes during the computation of the posterior distribution.

3.3.4 Information Exchange between the Sensor Nodes

We presently describe the information exchange that occurs between the sensor nodes during the computation of the posterior distribution. Each node initializes the parameter vector $\boldsymbol{\tau}$ of the prior distribution $p_{\mathbf{h}}(\mathbf{h})$ to the values in (3.21). Also, each node i initializes the $\frac{N(N-1)}{2}$ vector $\boldsymbol{\mu}_{k,i}$ in (3.23) to all zeros. The remaining vectors in (3.23) need not be initialized or passed during message passing, because the parameters corresponding to the vector $\mathbf{h}_y$ in $\mathbf{v}(\mathbf{h})$ involve only the training sequences which are already available at each node. Thus, each node can calculate the parameters corresponding to the vectors $\mathbf{h}_y$ and $\mathbf{h}_z$ using the information available at the node.

During the *first* sleep cycle $p = 1$, during each sleep cycle instant k each node

$i \in \mathcal{S}(k)$ calculates $-2\mathbf{u}_j^T \mathbf{r}_{k,i,j}$ for each other awake node $j \in \mathcal{S}(k) \setminus \{i\}$, and adds it to the appropriate element of the vector $\boldsymbol{\mu}_{k,i}$.

$$[\boldsymbol{\mu}_{k,i}]_{i,j} = [\boldsymbol{\mu}_{k,i}]_{i,j} - 2\mathbf{u}_j^T \mathbf{r}_{k,i,j} \tag{3.28}$$

Here, by $[\boldsymbol{\mu}_{k,i}]_{i,j}$ we mean the element corresponding to node j (corresponding to the channel gain $h_{i,j}$) in the vector $\boldsymbol{\mu}_{k,i}$. At every iteration p and every sleep cycle instant k, the awake nodes $i \in \mathcal{S}(k)$ multiply $p_{\mathbf{r}_{k,i}|\mathbf{h}}$ with the functions corresponding to the messages obtained in all of the *other* $c-1$ sleep cycle time instants ($\mathcal{N}(i) \setminus \{k\}$) it was awake during the previous $(p-1)$th sleep cycle to obtain the outgoing message. Since all the functions are from the same exponential family with sufficient statistics $\mathbf{v}(\mathbf{h})$, when the messages are multiplied the parameters of the messages sum up as explained above. Each node then passes the parameters corresponding to the product of the functions. Furthermore, nodes need to pass only the vectors $\boldsymbol{\mu}_{k,i}$ because each node can calculate the other vectors based on the information available at the node. Thus, the nodes $i \in \mathcal{S}(k)$ sum $\boldsymbol{\mu}_{k,i}$ with the vectors $\boldsymbol{\lambda}_{k' \to i}^{(p-1)}$ to obtain $\rho_{i \to k}^{(p)}$.

$$\rho_{i \to k}^{(p)} = \boldsymbol{\mu}_{k,i} + \sum_{k' \in \mathcal{N}(i) \setminus \{k\}} \boldsymbol{\lambda}_{k' \to i}^{(p-1)} \tag{3.29}$$

The $\frac{N(N-1)}{2}$ dimensional vector $\rho_{i \to k}^{(p)}$ is then broadcast to all other awake nodes.

Each node $i \in \mathcal{S}(k)$ then sums the $d-1$ messages $\rho_{j \to k}^{(p)}$ it heard from the other awake nodes $j \in \mathcal{S}(k) \setminus \{i\}$ with $\boldsymbol{\mu}_{k,i}$, and stores the result in $\boldsymbol{\lambda}_{k \to i}^{(p)}$.

$$\boldsymbol{\lambda}_{k \to i}^{p} = \boldsymbol{\mu}_{k,i} + \sum_{i' \in \mathcal{S}(k) \setminus \{i\}} \boldsymbol{\rho}_{i' \to k}^{p} \tag{3.30}$$

At the final iteration, node i sums $\boldsymbol{\lambda}_{k \to i}^{(p)}$ from k in all c sleep cycle instants it was awake, adds it to $\boldsymbol{\tau}$, and multiplies the result by the (offline computed $\frac{N(N-1)}{2} \times \frac{N(N-1)}{2}$

dimensional) new covariance matrix formed from the training data to get its estimate. We summarize this algorithm below.

- Initialize $\boldsymbol{\mu}_{k,i}$ to all zeros and $\boldsymbol{\tau}$ as in (3.21)
- During the 1st sleep cycle $p = 1$ and each sleep cycle instant k, at each node $i \in \mathcal{S}(k)$ calculate $[\boldsymbol{\mu}_{k,i}]_{i,j}$ as in (3.28).
- During sleep cycle $1 \leq p \leq \ell - 1$ and each sleep cycle instant k, at each node $i \in \mathcal{S}(k)$ repeat:
  - Calculate the message $\rho_{i \to k}^{(p)}$ as in (3.29) and broadcast it.
  - Sum the messages $\rho_{j \to k}^{(p)}$ received from nodes $j \in \mathcal{S}(k) \setminus \{i\}$ with $\boldsymbol{\mu}_{k,i}$ as in (3.30) to get $\boldsymbol{\lambda}_{k \to i}^{p}$ and go to sleep.
- At final sleep cycle $p = \ell$, at node i:
  Sum $\boldsymbol{\lambda}_{k \to i}^{(p)}$ from $k \in \mathcal{N}(i)$, add to $\boldsymbol{\tau}$ and multiply with the new covariance matrix to get the estimate.

3.3.5 Convergence and Sensitivity of the Algorithm

Having described the EP based algorithm for distributed estimation of channel gains, we next discuss the convergence properties and its sensitivity to node failures. As it was discussed in Section 3.1, after ℓ complete sleep cycles each (variable) node i will have communicated with nodes up to 2ℓ edges away from it in the factor graph. For any finite number of iterations ℓ, as the number of nodes $N \to \infty$ the subgraph which has root at i and contains the nodes no more than 2ℓ edges away from i becomes a tree with probability $\to 1$ [85] [72]. When applied for an appropriate finite number of iterations in such a case, our algorithm converges to the approximate posterior distribution of the channel gains given the observations at nodes no more than 2ℓ edges away from node i, because after approximating the prior distribution

this equivalent to applying BP and it is well known that BP converges on trees [32] [33].

Under this convergence assumption (which is the case for larger networks), we now examine the robustness of our algorithm to node failures. Consider the subgraph (tree) which has root at node i and contains the nodes no more than 2ℓ edges away from i. Suppose that one of the internal nodes in the tree has failed. This node failure results in a situation in which node i cannot exploit the observations of those nodes which should be communicated with through the failed node. However, as it can be seen from the discussion in Section 3.3.4, the information exchange between the sensor nodes will continue until the algorithm converges to a solution which is less accurate than it would have been otherwise.

3.4 Distributed Estimation with Diffusion LMS

Since the ultimate aim of this work is to demonstrate that the prior information can be effectively used to estimate network channel gains, we compare the performance of our EP based estimation algorithm with an another algorithm, the diffusion LMS [4], which does not make use of the prior distribution. The diffusion LMS uses only the observations made during the training phase to estimate the channel gains.

Suppose that each node has a copy of the channel gain vector $\mathbf{h}$ and it takes an initial value of $\mathbf{m}_{\mathbf{h}}$. If node i transmits its training sequence during a sleep cycle instant k, then all other nodes which are awake during the sleep cycle instant k have access to $\{u_{i,m}, r_{k,i',i,m}\}$ where $i' \in \mathcal{S}(k) \setminus \{i\}$ and $u_{i,m}$ is the input regression signal and $r_{k,i',i,m}$ is the desired signal. Note that $u_{i,m}$ and $r_{k,i',i,m}$ obey the equation

$$r_{k,i',i,m} = h_{i,i'} u_{i,m} + v_{k,i',i,m}$$

The network nodes $i' \in \mathcal{S}(k) \setminus \{i\}$ can use the diffusion LMS algorithm [4] to estimate $h_{i,i'}$. Denote the estimate of $h_{i,i'}$ at time instant m of sleep cycle instant k by $\hat{h}_{i,i'}^{k,m}$. Then,

$$\hat{h}_{i,i'}^{k,m} = \hat{h}_{i,i'}^{k,m-1} + \mu u_{i,m}(r_{k,i',i,m} - \hat{h}_{i,i'}^{k,m-1} u_{i,m}) \tag{3.31}$$

where μ is the step size.

At the end of each sleep cycle instant k, the nodes that are awake diffuse their estimates to get the combined estimate $\tilde{\mathbf{h}}^k$ as

$$\tilde{\mathbf{h}}^k = \sum_{i \in \mathcal{S}(k)} a(k,i) \hat{\mathbf{h}}_i^k \tag{3.32}$$

where $\hat{\mathbf{h}}_i^k$ is the estimate of $\mathbf{h}$ at node i at the end of the sleep cycle instant k and $a(k,i)$ satisfy $\sum_{i \in \mathcal{S}(k)} a(k,i) = 1$. The nodes $i \in \mathcal{S}(k)$ use the combined estimate $\tilde{\mathbf{h}}^k$ for estimation during the later sleep cycle instants.

We summarize this algorithm below.

- At each node i, initialize $\hat{\mathbf{h}}_i$ to $\mathbf{m_h}$
- During each sleep cycle and sleep cycle instant k, at each node $i \in \mathcal{S}(k)$ repeat:
  - For all $i' \in \mathcal{S}(k) \setminus \{i\}$ calculate estimate $\hat{h}_{i,i'}^{k,m}$ as in (3.31).
  - At the end of sleep cycle instant k, diffuse the estimate to get $\tilde{\mathbf{h}}^k$ as in (3.32).

3.5 Simulation Results

We have simulated the algorithms described in the previous sections to estimate the channel gains in a network and have plotted the estimation errors for both algorithms. We describe the experiment and present the results in this section. In

our experiment, we estimate the channel gain vector $\mathbf{h}$ for a network with 20 sensors applying EP and LMS. We selected a moderate size (20 nodes) network for our simulations, because testbeds on which initial studies can be done consist of nodes on the order of 10 [86]. The network is formed as described below. Candidate sensor positions are generated on the plane $\mathbb{R}^2$ such that they are i.i.d. and Gaussian distributed with mean zero and variance 1. These candidate sensor positions are then refined to actual sensor positions by keeping only those positions that are 0.08 apart from one another, because when the separation is less than 0.08 the channel gains become unrealistically large. A random sleep strategy with $K = 10$ and $d = 4$ is applied to this network, where the value of d is selected by simulating the algorithm for different values of d for fixed N, K and by choosing the one which gives better estimation error performance.

Although the diffusion LMS does not require the statistics of these channel gains for operation, EP requires the statistics for the estimation of these channel gains. As discussed in Section 3.2, the joint prior distribution of the channel gains is analytically complex and intractable. Thus, we empirically calculate the statistics (mean and covariance matrix) of the channel gains using many channel gains generated by plugging in sensor positions in the equation obtained by applying proportionality constant 1 to (3.7). Then, we generate a new set of channel gains as described in Section 3.5.1 to test the algorithms with. Next, we generate the BPSK training sequences of length 1000 randomly and uniformly. We run 1000 Monte Carlo simulations for each algorithm using a noise variance of $\sigma_N^2 = 1$ for this experiment.

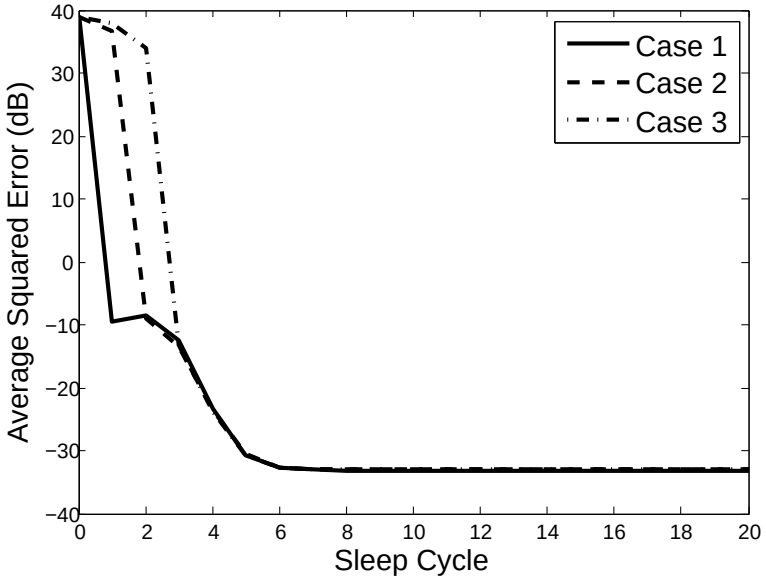
Due to the random sleep strategy and to the local nature of both algorithms, after ℓ iterations the observations made at a particular network node can propagate to the nodes only up to 2ℓ edges away from that node in the factor graph. Thus, after ℓ iterations each node will have observed information (either directly or indirectly)

about only a subset of the network links and this subset differs for each node. To plot the average estimation error we consider only these subsets of links, because including the estimation errors of those channels unobserved gives large average estimation errors since the nodes cannot make good estimates of the unobserved channels unless the correlations between the channels incident on the same node is very (physically unrealistically) large. Say node i has information about subset $\mathbf{h}_{i,\ell}$ of links after ℓ iterations. Then, we calculate the average squared estimation error first by averaging the squared estimation errors of $\mathbf{h}_{i,\ell}$ at each node i and then averaging over all nodes. Note that, here ℓ can be chosen independently from the number of iterations that we run the algorithm and $\mathbf{h}_{i,\ell_1} \subseteq \mathbf{h}_{i,\ell_2}$ for $\ell_1 \leq \ell_2$. Thus, one may consider different such $\mathbf{h}_{i,\ell}$ (each corresponds to different ℓ) at each node i and plot the average estimation error. For simplicity, we consider the subsets $\mathbf{h}_{i,\ell}$ only for $\ell = 1, 2, 3$ and plot the average estimation errors in Section 3.5.1. We call each of these cases 'Case 1', 'Case 2' and 'Case 3', respectively.

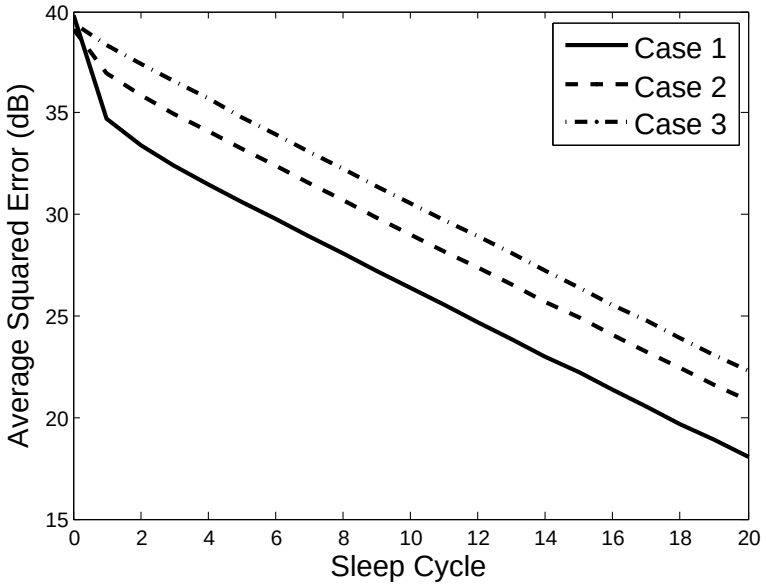
3.5.1 Comparison of EP and Diffusion LMS

We now simulate EP and the diffusion LMS and compare the performance of the two algorithms. Using these simulations, we show that although EP utilizes a path loss model it is also robust to the shadowing effects in the channel gains.

We do not consider the fading effects here, because fading is less important in immobile scenarios such as the one considered here and any multipath effects caused by the fixed reflectors can be included in the log-normal shadowing. Thus, we generate the channel gains to be estimated with path loss and log-normal shadowing effects. The shadowing effect is included in the channel gains by first generating gains as described above, and, then adding a Gaussian variable distributed $\mathcal{N}(0, 18.5)$ to the associated power in dB.



(a) EP



(b) LMS

Figure 3.4: Average squared estimation error of only those channel gains observed directly or indirectly by the nodes after 1st, 2nd and 3rd sleep cycles

Fig. 3.4(a) shows the average squared estimation error of the observed channel gains in dB when EP is applied. Interestingly, even with the Gaussian approximation

of the joint prior distribution and inclusion of shadowing effects, EP yields impressive estimation error performance for estimation of the true channel gains. Note that there is a drastic change in the average estimation error after first sleep cycle for 'Case 1' since these links were observed by the nodes directly. Also note that the drastic change shifts to the second sleep cycle and third sleep cycle for 'Case 2' and 'Case 3', respectively, as is to be expected. The estimation errors continue to decrease even after this initial drastic change because the nodes make use of the correlation with the other channel gains observed in subsequent iterations to refine estimates of these particular channel gains.

Fig. 3.4(b) shows the average squared error of the estimated channel gains when diffusion LMS is used with step size $\mu = 1$. This step size was chosen to be the maximum step size for which the diffusion LMS does not diverge in order to give the algorithm the chance to converge as quickly as possible, since EP converges faster. Note that EP gives better performance than the diffusion LMS when the network is required to estimate the channel coefficients within a small number of sleep cycles.

3.5.2 Mismatch of Path loss Exponent

The presented EP channel estimation algorithm can still be used to estimate the channel gains with small errors even if the actual path loss exponent differs by a small range of values from the path loss exponent used in the prior distribution. To show this, we generate the prior statistics with path loss exponent 4 and the channel gains to be estimated as described in Section 3.5.1 but with path loss exponent 6. Fig. 3.5 shows the average estimation error when EP is applied for this experiment and proves that our algorithm is robust to path loss exponent mismatch between 4 and 6 when the nodes are placed i.i.d. according to a Gaussian distribution.

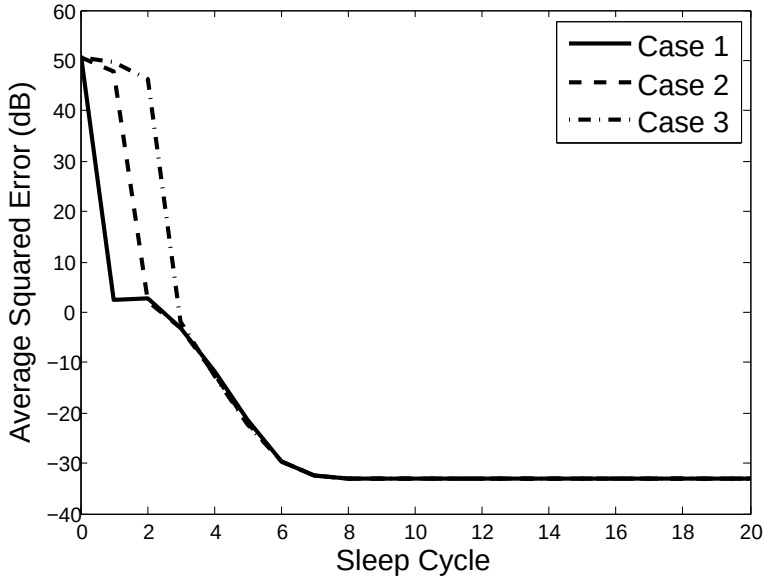


Figure 3.5: Average squared estimation error when EP (uses path loss exponent 4) is applied to estimate channel gains having path loss exponent 6

3.6 Computational Complexity, Message Passing Overhead and Memory Requirement

As we developed a low-complexity collaborative estimation algorithm with less attention to communication, it is important to provide the computational computational complexity of the algorithm. In this section in addition to providing computational complexity, we provide the number of messages that are to be passed between the nodes and the number of memory required for both algorithms during distributed estimation of channel gains.

3.6.1 Computational Complexity of EP and diffusion LMS

We calculate the number of computations required for each algorithm based on the expressions given in Section 3.3.4 and Section 3.4. We first consider the computational complexity of EP.

Table 3.1: Computational complexity, message passing overhead and memory requirements

	EP	Diffusion LMS
Computational complexity (per P cycles)		
Additions	$Kd(d-1)(M+1)$ $+PK(dc+d^2)\frac{N(N-1)}{2}$ $+\frac{N^2(N-1)}{2}\left(c+\frac{N(N-1)}{2}\right)$	$PKd\left(2(d-1)M+d\frac{N(N-1)}{2}\right)$
Multiplications	$Kd(d-1)M+N\left(\frac{N(N-1)}{2}\right)^2$	$PKd(d-1)\left(2M+\frac{N(N-1)}{2}\right)$
Message passing overhead (per sleep cycle)	$Kd\frac{N(N-1)}{2}$	$Kd\frac{N(N-1)}{2}$
Memory requirement	$\frac{N(N-1)(N^2-N+2c+5)}{4}+MN$	$\frac{N(N-1)}{2}+MN$

The computations related to the parameters of the prior distribution $p_{\mathbf{h}}(\mathbf{h})$ are done offline. Each instance of (3.28) requires M multiplications and $M+1$ additions, and is run in each of d nodes for $d-1$ other nodes in each sleep cycle instant in the first iteration, giving a total of $d(d-1)M$ multiplications and $d(d-1)(M+1)$ additions spent in (3.28) per sleep cycle instant in the first iteration (over the whole network). Each instance of (3.29) requires $c\frac{N(N-1)}{2}$ additions, and is run in each of d nodes in each sleep cycle instant, giving a total of $dc\frac{N(N-1)}{2}$ additions spent in (3.29) in each sleep cycle instant (over the whole network). Each instance of (3.30) requires $d\frac{N(N-1)}{2}$ additions, and is run in each of d nodes in each sleep cycle instant, giving a total of $d^2\frac{N(N-1)}{2}$ additions spent in (3.29) in each sleep cycle instant (over the whole network).

Finally, the final iteration requires each of N nodes to perform $(c+1)\frac{N(N-1)}{2}$

additions, then multiplication of a matrix times a vector requiring $\left[\frac{N(N-1)}{2}\right]^2$ multiplications and $\frac{N(N-1)}{2}\left[\frac{N(N-1)}{2}-1\right]$ additions. This gives a total of $N\left[(c+1)\frac{N(N-1)}{2}+\frac{N(N-1)}{2}\left(\frac{N(N-1)}{2}-1\right)\right]$ additions and $N\left[\frac{N(N-1)}{2}\right]^2$ multiplications over the entire network during the last iteration.

For the diffusion LMS, (3.31) is run during each sleep cycle instant k at each awake node $i \in \mathcal{S}(k)$ for every other awake node $i' \in \mathcal{S}(k) \setminus \{i\}$ for each training instant $m \in \{1, \ldots, M\}$. One calculation of (3.31) consists of two multiplications and two additions. Thus, $2d(d-1)M$ multiplications and $2d(d-1)M$ additions are spent in each sleep cycle time instant on calculations of the form (3.31). Then the nodes diffuse the estimates and calculate (3.32). This involves $d\frac{N(N-1)}{2}$ multiplications and $(d-1)\frac{N(N-1)}{2}$ additions in each of d sensor nodes, giving a total of $d^2\frac{N(N-1)}{2}$ multiplications and $d(d-1)\frac{N(N-1)}{2}$ additions spent on calculations for (3.32) per sleep cycle time instant.

If a total of P sleep cycles are performed, required number of additions and multiplications over the entire network for EP and diffusion LMS are given in Table 3.1.

3.6.2 Message Passing Overhead for EP and diffusion LMS

A significant portion of the power in the nodes is expended on internode communication and this heavily depends on the message passing overhead. Thus we compute the message passing overhead of algorithms in this subsection.

The message passing overhead for EP can be calculated by analyzing the message exponential family. Although the sufficient statistics vector $\mathbf{v}(\mathbf{h})$ of the message exponential family has a dimension of $2\,\frac{N(N-1)}{2}+\frac{[N(N-1)-1][N(N-1)]}{8}$, each node i needs to pass only $\frac{N(N-1)}{2}$ parameters $\boldsymbol{\mu}_{k,i}$ which are corresponding to the vector $\mathbf{h}$ in $\mathbf{v}(\mathbf{h})$ in (3.17). The parameters corresponding to the vector $\mathbf{h}_y$ in $\mathbf{v}(\mathbf{h})$ in (3.17) involve

only the training sequences that are already available at each node. Similarly, the parameters corresponding to $\mathbf{h}_z$ in $\mathbf{v}(\mathbf{h})$ in (3.17) involve only the elements of the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{h}}$ of the prior distribution. Thus, each node can calculate the parameters corresponding to the vectors $\mathbf{h}_y$ and $\mathbf{h}_z$ using the internal information. Thus, the number of messages required to be passed for EP during a complete sleep cycle is $Kd\frac{N(N-1)}{2}$.

When the diffusion LMS is applied to the channel gain estimation, each node calculates its own estimates and, then, diffuses its estimates at the end of the sleep cycle instant. Thus, the message passing overhead for the diffusion LMS is simply the number of channel gains in the network. The total message passing overhead for the diffusion LMS is $Kd\frac{N(N-1)}{2}$ per sleep cycle. Thus, the total message passing overhead is exactly the same for both algorithms.

One might argue that not all $\frac{N(N-1)}{2}$ parameters in the vectors need to be passed, because when EP is applied many elements in $\boldsymbol{\mu}_{k,i}$ are zero and when diffusion LMS is applied many elements in $\hat{\mathbf{h}}_i^k$ remain unchanged. However, if one wants to take the zero and unchanged elements into account to reduce the message passing overhead, one must keep track of the elements that change and perform some indexing when he passes messages. Thus, clearly there is a trade off between reduction in the number of parameters that are passed and increment in the number of computations. It is unclear whether there is any saving on the energy consumption at the nodes from such a practice. For that reason, we pass all $\frac{N(N-1)}{2}$ parameters in the vectors.

3.6.3 Memory Requirement

It is also important to analyze memory requirement of the algorithms, because in some applications memory is limited. We first calculate the memory requirement for EP. When EP is employed for channel gain estimation, some parameters are

calculated offline and stored in the sensor nodes while the rest of the parameters are calculated and stored online. First, consider the offline computed parameters. Each node requires storing information on the prior distribution and the training sequences of all nodes. The storage of the parameter vector $\boldsymbol{\tau}$ and the training sequences $\mathbf{u} = \{\mathbf{u}_i | i \in \{1, \ldots, N\}\}$ requires $\frac{N(N-1)(N^2-N+6)}{8}$ and MN memory locations, respectively. Also, each node initializes the vector $\boldsymbol{\mu}_{k,i}$ to all zeros which requires $\frac{N(N-1)}{2}$ memory locations. Furthermore, at the final iteration each node utilizes a offline computed $\frac{N(N-1)}{2} \times \frac{N(N-1)}{2}$ matrix to calculate its estimate. Since this is a symmetric matrix, this requires $\frac{N^2(N-1)^2}{8}$ memory locations at each node. When EP is run, each node will allocate memory to store messages that are to be received during different sleep cycle instants. This will require $c\frac{N(N-1)}{2}$ memory locations at each node. Thus, when EP is employed each node requires a total of $\frac{N(N-1)(N^2-N+2c+5)}{4} + MN$ memory locations.

When diffusion LMS is employed, each node initializes the estimates and stores the training sequence offline. The initialization of the estimates require $\frac{N(N-1)}{2}$ memory locations while the storage of the training sequence requires MN memory locations. The diffusion LMS does not require any additional memory when the algorithm is in progress. Thus, the diffusion LMS requires a total of $\frac{N(N-1)}{2} + MN$ memory locations at each node.

Table 3.1 summarizes the results derived in Section 3.6.

3.7 Conclusions

We considered a distributed channel estimation problem in a sensor network which employs a random sleep strategy to conserve energy. We modeled the channel gains in the network using a path loss model and gathered information about these channels using channel sounding. We derived a low-complexity collaborative estimation algorithm for this problem from expectation propagation (EP) principle which provides

efficient means for the nodes to disseminate the information required to compute MMSE estimates at each node.

We compared the performance of EP based algorithm with the diffusion LMS and showed EP gives a better estimation error performance using the simulation results. We also proved that although our algorithm utilizes a path loss model with an a priori fixed path loss exponent, it is robust to shadowing effects and variation of path loss exponents. Finally, we compared the message passing overhead, computational complexity, and memory requirements of both algorithms.

4. Low-Communication Collaborative Estimation Algorithms

In the previous chapter we discussed about reducing transmission power in wireless sensor networks through efficient communication strategies. The power spent on communication can be further reduced by efficiently representing the information exchanged between the nodes while achieving the desired estimation error performances at the nodes. The tools from distributed source coding theory can be applied to develop algorithms that efficiently represent information exchanged while paying less attention to complexity. We study such low-communication collaborative estimation algorithms in this chapter.

To study such an algorithm, consider a network of nodes deployed to monitor a common phenomenon. Each node in the network making indirect observations of the common phenomenon communicates with the other nodes and estimates the common phenomenon with a certain estimation error performance (Bayesian cost). We study collaborative estimation schemes that can efficiently encode/decode information exchanged between the nodes without any constraints on the complexity [87]. In particular, we study such collaborative estimation schemes which accomplish this with separated network/channel and source coding.

Our first insight, made in Section 4.2, is that under this decomposition the proper source coding model reflecting the capabilities of the network code is one in which each node multicasts a different message to every possible subset of other nodes in the network. For example, in the lowest dimensional $M = 3$ nontrivial case, each node will create multiple descriptions of its observations, one for each of the other two nodes in the network individually, and one for both of them as shown in Fig. 4.1.

We employ a classic technique from multiterminal information theory [39] [16] to study the relationship between the rates of the source code used, and the estimation

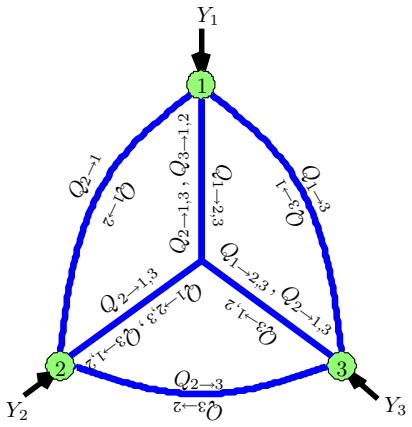


Figure 4.1: The "peace" network, which depicts the lowest dimensional $M = 3$ non-trivial case of the problem of collaborative distributed estimation. To determine the direction of travel of a message, note that the messages flow in the direction in which they are read.

errors that each of the nodes can obtain in estimating the common phenomenon. The rate distortion region explains the relationship between the length in bits of the different messages multicast between the nodes and the estimation errors (measured in terms of average costs for Bayesian estimation) that decoder/estimators at these nodes can obtain. We derive inner and outer bounds to the rate distortion region and show that our inner bound simplifies to the known bounds for some simpler problems.

4.1 Problem Formulation

Consider a network of M nodes deployed to monitor a common phenomenon embodied by a sequence of random variables $T^{(n)}$. Each node $j \in [M]$ in the network makes indirect observations of this phenomenon, embodied as another sequence of random variables $Y_j^{(n)}$ statistically related to $T^{(n)}$. Let the sequence $\left(T^{(n)}, Y_1^{(n)}, \ldots, Y_M^{(n)}\right)$ be i.i.d. according to joint probability distribution $p_{T,Y_1,\ldots,Y_M}$.

The source encoder at each node j encodes its observations $\boldsymbol{Y}_j$ into a common

message $Q_{j\to\mathcal{A}} \in \{1, 2, \ldots, 2^{NR_{j\to\mathcal{A}}}\}$ to each of the nodes with indices in some subset $\mathcal{A}$ of the other nodes using an average of $R_{j\to\mathcal{A}}$ bits per observation symbol. A different such message is encoded at each node j for each such subset $\mathcal{A} \subseteq [M] \setminus j$, and then reliably multicasted (e.g. with the aid of some channel and network codes) to the nodes in $\mathcal{A}$.

We employ a classic technique from multiterminal information theory [39] [16] to study the relationship between the rates $\{R_{j\to\mathcal{A}} \mid j \in [M],\ \mathcal{A} \in 2^{[M]\setminus j}\}$ of the source code used, and the estimation errors D_j that each of the nodes can obtain in estimating the sequence $T^{(n)}, n \in [N]$ from their own observations $\boldsymbol{Y}_j$ and the messages $Q_{\mathcal{D}_j} := \left[Q_{i\to\mathcal{A}} \,\middle|\, j \in \mathcal{A},\ \mathcal{A} \in 2^{[M]\setminus i}\right]$ they have received.

The vector $(\mathbf{r}, \mathbf{d})$ of multicast rates $\mathbf{r} := \left[R_{j\to\mathcal{A}} \,\middle|\, j \in [M], \mathcal{A} \in 2^{[M]\setminus j}\right]$ and average estimation errors $\mathbf{d} := [D_j \,|j \in [M]]$ is said to be achievable if there exist block length N, encoders

$$f_{j\to\mathcal{A}} : \boldsymbol{\mathcal{Y}}_j \to [L_{j\to\mathcal{A}}], \quad \forall\ \mathcal{A} \subseteq [M] \setminus j,\ j \in [M] \tag{4.1}$$

and decoders

$$g_i : \boldsymbol{\mathcal{Y}}_i \times \prod_{(j\to\mathcal{A})\in\mathcal{D}_i} [L_{j\to\mathcal{A}}] \to \hat{\boldsymbol{T}}_i, \quad \forall\ i \in [M] \tag{4.2}$$

with $\hat{\boldsymbol{T}}_i = g_i(\boldsymbol{Y}_i, Q_{\mathcal{D}_i})$ such that

$$R_{j\to\mathcal{A}} \geq \frac{1}{N} \log_2 L_{j\to\mathcal{A}}, \quad \forall\ \mathcal{A} \subseteq [M] \setminus j,\ j \in [M] \tag{4.3}$$

and

$$E\left[\frac{1}{N}\sum_{n=1}^{N} d_i(T^{(n)}, \hat{T}_i^{(n)})\right] \leq D_i, \quad \forall\ i \in [M] \tag{4.4}$$

The rate distortion region $\mathcal{RD}$ for this problem is defined as the closure of the region of achievable vectors $(\mathbf{r}, \mathbf{d})$.

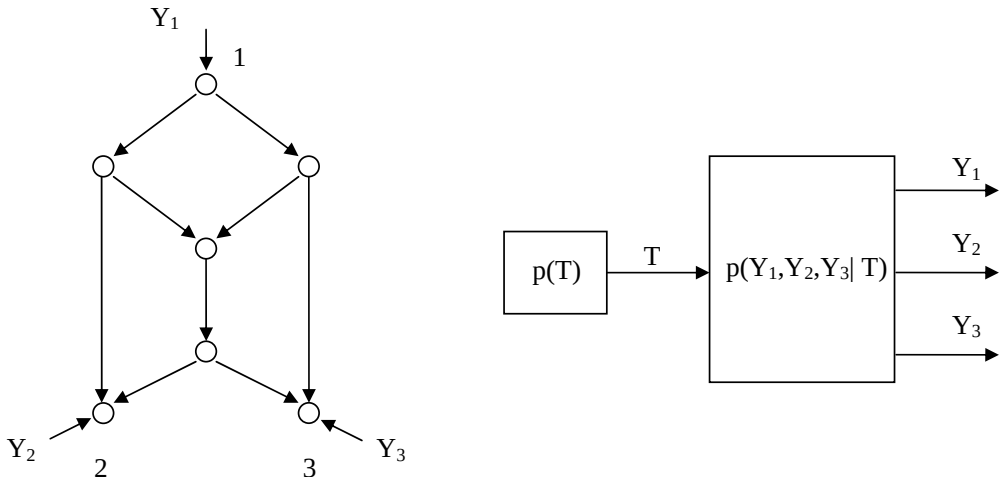


Figure 4.2: This network demonstrates that considering a source code at node 1 which only encodes a dedicated message to node 2 and a dedicated message to node 3 is not general enough. Instead, the source encoder at node 1 should encode a separate message for each possible subset of other nodes in the network.

4.2 Distributed Estimation and Multiterminal Source Coding

As outlined in the introduction, suppose we aim to separate the source coding part of the distributed estimation problem from the network/channel coding part, despite the fact that such a separation may be suboptimal. Here we argue that the best model for the distributed source code is one in which each encoder multicasts a message to each subset of other nodes in the network, rather than sending an individual message to each other node in the network.

To see that such a model is the appropriate one, consider a simple wired network depicted in Fig. 4.2 in which three nodes $(1, 2, 3)$ making local observations $Y_1^{(n)}, Y_2^{(n)}, Y_3^{(n)}$ statistically related to a common underlying sequence $T^{(n)}$ would like to communicate over the butterfly network in order to form local estimates $\hat{T}_1^{(n)}, \hat{T}_2^{(n)}, \hat{T}_3^{(n)}$ of $T^{(n)}$. Because of the unidirectionality of the links, only node 1 may transmit information. Suppose further that the observations at node 2 and 3

are statistically identical and the distortion metrics are the same, and we wish to obtain the same target average estimation error $D_2 = D_3$ at the two nodes. If node 1 encodes a separate message for node 2 and node 3, then it would suffice to take these two messages to be the same in this symmetric case. However, the network code can not know this, because we have forced the source coding construction to have a separate message for each of nodes 2 and 3. Thus, the network code is forced to attempt to transmit two unicasts, one between 1 and 2 with rate $R_{1\to 2}$, and one in between 1 and 3 with rate $R_{1\to 3}$. If each link in the network is unit capacity, and the network code is forced to treat the information flowing in between nodes 1 and 2 as independently unicast from the unicast between 1 and 3, then the highest symmetric rate $R = R_{1\to 2} = R_{1\to 3}$ which can be obtained is 3/2. However, had we chosen our source code as outputting three messages $Q_{1\to 2}, Q_{1\to 3}, Q_{1\to 2,3}$, so that we included one which was *multicast* from 1 to both 2 and 3, then the network code could support a symmetric rate of $R_{1\to 2,3} = 2$ [88]. This would not send any unicast information at all $R_{1\to 2} = R_{1\to 3} = 0$. This way 33% more useful information flows from 1 to 2 and 3 as would have had we required only unicasts, and the distortion obtained at nodes 2 and 3 will thus be lower.

A similar conclusion concerning the insufficiency of a distributed source code creating unicasts in this context of networked estimation can be drawn in a wireless context as well. In particular, consider the network depicted in Fig. 4.3 in which node 1 transmits a signal which is overheard by both 2 and 3, through independent Gaussian noise of differing powers, so that the received signal to noise power ratio at node 2 is lower than that at node 3. The capacity region of such a degraded broadcast channel is known, and a channel code can be constructed which reliably transmits the multicast messages $Q_{1\to 2}, Q_{1\to 3}, Q_{1\to 2,3}$ of rates $R_{1\to 2}, R_{1\to 3}, R_{1\to 2,3}$ respectively if

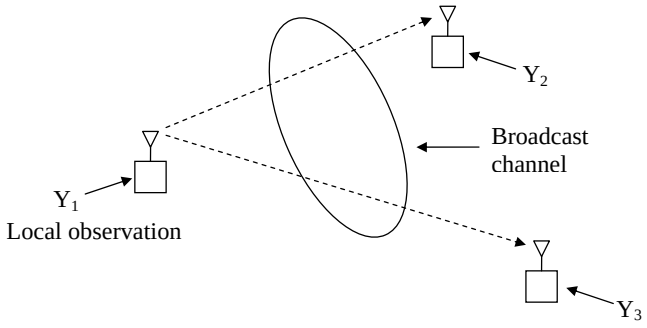


Figure 4.3: Due to the broadcast nature of the wireless medium, an appropriate source coding model for collaborative inference over a wireless channel should involve communication with subsets of other users rather than only point to point communication.

there is some $\alpha \in [0,1]$ such that

$$R_{1\to 2} + R_{1\to 2,3} < \frac{1}{2}\log_2\left(1 + \frac{(1-\alpha)P_T}{\alpha P_T + N_2}\right) \tag{4.5}$$

$$R_{1\to 3} < \frac{1}{2}\log_2\left(1 + \alpha\frac{P_T}{N_3}\right) \tag{4.6}$$

If we again consider the application goal of the network for 1 to transmit information in order to help 2 and 3 with their estimation problems, and consider the special case that the local observations at 2 and 3 are statistically identical, then clearly the use of multicast messages (as opposed to unicast messages alone) is desirable. This is because any information sent to node 2 can also be heard at node 3 with no extra cost in information, and thus our source code ought to exploit this capability of the channel code.

From these two simple examples we can easily infer that a proper separated source and network/channel coding approach treats the source code within network node i as producing an array of 2^{M-1} multicast messages, with one message $Q_{i\to\mathcal{A}}$ for each subset $\mathcal{A} \subseteq [M] \setminus i$. The capabilities of the possible network/channel codes are then

summarized by a region $\mathcal{C}$ of vectors of such multicast rates

$$\mathbf{r} := [R_{j\to\mathcal{A}} | j \in [M],\ \mathcal{A} \subseteq [M] \setminus j] \tag{4.7}$$

which are simultaneously supportable by the network infrastructure. The capabilities of the possible source codes are summarized by a *rate distortion region* $\mathcal{RD}$ describing the set of simultaneously achievable multicast rates $\mathbf{r}$ and average estimation errors

$$\mathbf{d} := [D_i | i \in [M]], \quad D_i := \frac{1}{N}\sum_{n=1}^{N} E\left[d_i\left(T^{(n)}, \hat{T}_i^{(n)}\right)\right] \tag{4.8}$$

An overall source channel code achieving average estimation errors lower than $\mathbf{d}$ is selected by choosing a rate vector $\mathbf{r}$ that is in both $\mathcal{C}$ and also in $\mathcal{RD}$, i.e. with $(\mathbf{r}, \mathbf{d}) \in \mathcal{RD}$. We now focus our efforts on characterizing the rate distortion region for the associated family of source codes we have selected.

4.3 Inner and Outer Bounds to the Rate Distortion Region

We derive inner and outer bounds to the rate distortion region in this section. To do this, denote the set of message indices leaving node i by $\mathcal{S}_i := \left\{(i \to \mathcal{A}) \,|\, \mathcal{A} \in 2^{[M]\setminus i}\right\}$ and the set $\{U_{i\to\mathcal{A}} \mid \mathcal{A} \in 2^{[M]\setminus i}\}$ as $U_{\mathcal{S}_i}$. If we define $\mathcal{S} := \bigcup_{i\in[M]} \mathcal{S}_i$, then we have the following theorem.

4.3.1 Inner Bound

Theorem 1:
Given a joint distribution $p_{T,Y_{[M]}}(t, y_{[M]})$, let $\Xi(\mathbf{d})$ be the collection of random vectors $\boldsymbol{\xi} = U_{\mathcal{S}}$ which are jointly distributed with T and $Y_{[M]}$ such that the following conditions are satisfied

1. $T, Y_{[M]\setminus i}, U_{\mathcal{S}\setminus\mathcal{S}_i} \leftrightarrow Y_i \leftrightarrow U_{\mathcal{S}_i}$ for all $i \in [M]$

2. There exists a decoding function $g_i : \mathcal{U}_{\mathcal{D}_i} \times \mathcal{Y}_i \to \hat{\mathcal{T}}_i$ such that $E\left[d_i(T, g_i(U_{\mathcal{D}_i}, Y_i))\right] \leq D_i$ for all $i \in [M]$

For each $\boldsymbol{\xi} \in \Xi(\mathbf{d})$, define $\boldsymbol{\Phi}(\boldsymbol{\xi})$ as

$$\boldsymbol{\Phi}(\boldsymbol{\xi}) = \left\{ \tilde{R}_{\mathcal{S}} \ : \ \sum_{(j \to \mathcal{A}) \in \mathcal{P}_j} \tilde{R}_{j \to \mathcal{A}} > \sum_{(j \to \mathcal{A}) \in \mathcal{P}_j} H(U_{j \to \mathcal{A}}) - H(U_{\mathcal{P}_j} | Y_j), \ \forall \, \mathcal{P}_j \subseteq \mathcal{S}_j, \ j \in [M] \right\} \tag{4.9}$$

Also, for each $\boldsymbol{\xi} \in \Xi(\mathbf{d})$ and for each $\boldsymbol{\phi} \in \boldsymbol{\Phi}(\boldsymbol{\xi})$, define $\mathcal{RD}_{in}(\boldsymbol{\xi}, \boldsymbol{\phi})$ as

$$\begin{aligned} \mathcal{RD}_{in}(\boldsymbol{\xi}, \boldsymbol{\phi}) \ = \ \Big\{ R_{\mathcal{S}} \ : \ & \sum_{(j \to \mathcal{A}) \in \mathcal{C}_i} R_{j \to \mathcal{A}} \geq \sum_{(j \to \mathcal{A}) \in \mathcal{C}_i} \left(\tilde{R}_{j \to \mathcal{A}} - H(U_{j \to \mathcal{A}}) \right) \\ & + H(U_{\mathcal{C}_i} | U_{\mathcal{D}_i \setminus \mathcal{C}_i}, Y_i), \ \forall \, \mathcal{C}_i \subseteq \mathcal{D}_i, \ i \in [M] \Big\} \end{aligned} \tag{4.10}$$

Let

$$\mathcal{RD}_{in} := \bigcup_{\boldsymbol{\xi} \in \Xi(\mathbf{d})} \bigcup_{\boldsymbol{\phi} \in \boldsymbol{\Phi}(\boldsymbol{\xi})} \mathcal{RD}_{in}(\boldsymbol{\xi}, \boldsymbol{\phi}) \tag{4.11}$$

Then, the convex hull $\mathbf{conv}\left(\mathcal{RD}_{in}\right)$ of $\mathcal{RD}_{in}$ is an inner bound to the rate distortion region, i.e. $\mathbf{conv}\left(\mathcal{RD}_{in}\right) \subseteq \mathcal{RD}$.

Sketch of the Proof

This result is an adaptation of a well known inner bound in the multiterminal source coding community known as the Berger-Tung inner bound, as clarified by Han and Kobayashi [39], with the twist that the multiple (dependent) descriptions at each encoder require an additional set of encoder inequalities. A sketch of the proof is provided here and a detailed proof is provided in Appendix A.

Distribution of Auxiliary Variables:

Select a joint conditional distribution $p\left(u_{\mathcal{S}} \,\middle|\, t, y_{[M]}\right)$, a set of encoding functions

$\{f_{j\to\mathcal{A}} \,|\, (j\to\mathcal{A})\in\mathcal{S}\}$ and a set of decoding functions $\{g_i \,|\, i\in[M]\}$ such that the rates $R_{\mathcal{S}}$ are in $\mathcal{RD}_{in}$. Calculate the marginal distributions $p(u_{j\to\mathcal{A}})$.

Codebook Generation:

At each node $j\in[M]$, for each subset of nodes $\mathcal{A}\subseteq 2^{[M]\backslash j}$, generate a codebook $C_{j\to\mathcal{A}}$ with $2^{N\tilde{R}_{j\to\mathcal{A}}}$ length-N codewords by randomly drawing the elements such that they are i.i.d. according to the distribution $p(u_{j\to\mathcal{A}})$ and rates $\tilde{R}_{j\to\mathcal{A}}$ satisfy

$$\sum_{(j\to\mathcal{A})\in\mathcal{P}_j}\tilde{R}_{j\to\mathcal{A}} > \sum_{(j\to\mathcal{A})\in\mathcal{P}_j} H(U_{j\to\mathcal{A}}) - H(U_{\mathcal{P}_j}|Y_j), \quad \forall\ \mathcal{P}_j\subseteq\mathcal{S}_j \tag{4.12}$$

Index the codewords by $m_{j\to\mathcal{A}}\in\{1,\dots,2^{N\tilde{R}_{j\to\mathcal{A}}}\}$. Partition the codewords into $2^{NR_{j\to\mathcal{A}}}$ bins by randomly and uniformly assigning the indices to the bins. Index the bins by $b_{j\to\mathcal{A}}\in\{1,\dots,2^{NR_{j\to\mathcal{A}}}\}$ and denote the set of codewords in bin $b_{j\to\mathcal{A}}$ by $\mathcal{B}_{j\to\mathcal{A}}(b_{j\to\mathcal{A}})$.

Encoding:

At each node $j\in[M]$, encode the observation sequence $\boldsymbol{Y}_j$ by selecting one codeword $\boldsymbol{U}_{j\to\mathcal{A}}(m_{j\to\mathcal{A}})$ from each codebook $C_{j\to\mathcal{A}}$, for each $(j\to\mathcal{A})\in\mathcal{S}_j\}$, such that $\big(\boldsymbol{U}_{\mathcal{S}_j}(\boldsymbol{m}_{\mathcal{S}_j}),\,\boldsymbol{Y}_j\big)\in A_\epsilon^*(U_{\mathcal{S}_j},Y_j)$, where A_ϵ^* is the set of strongly typical sequences. If there are more than one such $\boldsymbol{U}_{\mathcal{S}_j}(\boldsymbol{m}_{\mathcal{S}_j})$, select the codewords with the smallest indices under lexicographic ordering. If there is no such $\boldsymbol{U}_{\mathcal{S}_j}(\boldsymbol{m}_{\mathcal{S}_j})$, select an arbitrary set of codewords. For each subset of nodes $\mathcal{A}\subseteq 2^{[M]\backslash j}$, send the index $b_{j\to\mathcal{A}}$ of the bin that contains $\boldsymbol{U}_{j\to\mathcal{A}}(m_{j\to\mathcal{A}})$ to the nodes in $\mathcal{A}$, i.e. $\boldsymbol{U}_{j\to\mathcal{A}}(m_{j\to\mathcal{A}})\in\mathcal{B}_{j\to\mathcal{A}}(b_{j\to\mathcal{A}})$. This requires $R_{j\to\mathcal{A}}$ bits to multicast a message to a subset of nodes $\mathcal{A}\subseteq 2^{[M]\backslash j}$.

Decoding:

At each node $i \in [M]$, decode the messages received at the node by selecting the codeword $\boldsymbol{U}_{j\to\mathcal{A}}(\ell_{j\to\mathcal{A}})$ in bin $\mathcal{B}_{j\to\mathcal{A}}(b_{j\to\mathcal{A}})$ for each $(j \to \mathcal{A}) \in \mathcal{D}_i$ such that $(\boldsymbol{U}_{\mathcal{D}_i}(\boldsymbol{\ell}_{\mathcal{D}_i}), \boldsymbol{Y}_i) \in A_\epsilon^* (U_{\mathcal{D}_i}, Y_i)$, where $U_{\mathcal{D}_i} \triangleq (U_{j\to\mathcal{A}})_{(j\to\mathcal{A})\in\mathcal{D}_i}$. If there is no such a set of codewords, select an arbitrary set of codewords. Reproduce the underlying sequence $\boldsymbol{T}$ by $\hat{\boldsymbol{T}}_i = g_i (\boldsymbol{Y}_i, \boldsymbol{U}_{\mathcal{D}_i}(\boldsymbol{\ell}_{\mathcal{D}_i}))$. $\square$

4.3.2 Outer Bound

We next present our outer bound.

Theorem 2:

Given a joint distribution $p_{T,Y_{[M]}}(t, y_{[M]})$, let Ψ be the collection of random vectors $\psi = \left(W_{[M]}, Z_{\mathcal{S}}\right)$ which are jointly distributed with T and $Y_{[M]}$ such that the following conditions are satisfied

1. $T, Y_{[M]\setminus j} \leftrightarrow Y_j \leftrightarrow Z_{j\to\mathcal{A}}$ for all $j \in [M]$ and $(j \to \mathcal{A}) \in \mathcal{S}$
2. $p\left(t, y_{[M]}, w_{[M]}\right) = p\left(t, y_{[M]}\right)\ p\left(w_{[M]}\right)$
3. There exists a decoding function $g_i : \mathcal{Y}_i \times \mathcal{W}_i \times \mathcal{Z}_{\mathcal{D}_i} \to \hat{\mathcal{T}}_i$ such that $E[d_i(T, g_i(Y_i, W_i, Z_{\mathcal{D}_i}))] \le D_i$ for all $i \in [M]$

Let

$$\mathcal{RD}_{out}(\psi) \triangleq \left\{ R_{\mathcal{S}} \;\middle|\; \sum_{(j\to\mathcal{A})\in\mathcal{C}_i} R_{j\to\mathcal{A}} \ge I\left(Y_{[M]\setminus i}; Z_{\mathcal{C}_i} \;\middle|\; Y_i, W_i, Z_{\mathcal{D}_i\setminus\mathcal{C}_i}\right), \quad \forall \mathcal{C}_i \subseteq \mathcal{D}_i,\ i \in [M] \right\} \tag{4.13}$$

Define

$$\mathcal{RD}_{out} := \bigcup_{\psi\in\Psi} \mathcal{RD}_{out}(\psi) \tag{4.14}$$

Then, the convex hull $\mathbf{conv}\,(\mathcal{RD}_{out})$ of $\mathcal{RD}_{out}$ is an outer bound to the rate distortion region, i.e. $\mathbf{conv}\,(\mathcal{RD}_{out}) \supseteq \mathcal{RD}$. $\square$

The proof of Theorem 2 is given in Appendix B.

4.3.3 Structural Properties of the Inner Bound

We next analyze the structure of the achievable rate region (inner bound), because knowing the structure of the rate region may be helpful when we optimize some function of rates over the rate region. We indeed use some structural properties of the inner bound to simplify our bound to simpler problems in Section 4.4, and, thus present those structural properties below.

Definition: Let $f : 2^{[M]} \to \mathcal{R}^+$ be a set function. The polyhedron

$$\mathcal{G}(f) \triangleq \{(x_1, \ldots, x_M) : \sum_{i \in \mathcal{A}} x_i \geq f(\mathcal{A}),\ \forall \mathcal{A} \subset [M]\} \tag{4.15}$$

is a *contra-polymatroid* if f satisfies [89, 90]

1. $f(\emptyset) = 0$ (normalized)
2. $f(\mathcal{A}) \leq f(\mathcal{B})$ if $\mathcal{A} \subset \mathcal{B}$ (nondecreasing)
3. $f(\mathcal{A}) + f(\mathcal{B}) \leq f(\mathcal{A} \cup \mathcal{B}) + f(\mathcal{A} \cap \mathcal{B})$ (supermodular)

If f satisfies the three properties, f is called a *rank* function.

Proposition 1: For each $\boldsymbol{\xi} \in \Xi(\mathbf{d})$, $\Phi(\boldsymbol{\xi})$ is a contra-polymatroid.

Proof: The set $\mathcal{S}$ is implied to be the ground set, and the rank function $\rho : 2^{\mathcal{S}} \to \mathbb{R}$ is defined as

$$\rho(\mathcal{P}) \triangleq \sum_{j \in [M]} \sum_{(j \to \mathcal{A}) \in \mathcal{P} \cap \mathcal{S}_j} H(U_{j \to \mathcal{A}}) - H(U_{\mathcal{P} \cap \mathcal{S}_j} | Y_j) \tag{4.16}$$

We must show that ρ is indeed a rank function. Consider two sets $\mathcal{Q}$ and $\mathcal{P}$ such that $\mathcal{Q} \subseteq \mathcal{P} \subseteq \mathcal{S}$, then

$$\begin{aligned}\rho(\mathcal{P}) - \rho(\mathcal{Q}) &= \sum_{j\in[M]} \Big(\sum_{(j\to\mathcal{A})\in\mathcal{L}_j} H(U_{j\to\mathcal{A}}) - H(U_{\mathcal{P}\cap\mathcal{S}_j}|Y_j) + H(U_{\mathcal{Q}\cap\mathcal{S}_j}|Y_j)\Big) \\ &= \sum_{j\in[M]} \Big(\sum_{(j\to\mathcal{A})\in\mathcal{L}_j} H(U_{j\to\mathcal{A}}) - H(U_{\mathcal{L}_j}|U_{\mathcal{Q}\cap\mathcal{S}_j}, Y_j)\Big) \\ &\geq 0\end{aligned}$$

where $\mathcal{L}_j := (\mathcal{P}\cap\mathcal{S}_j) \setminus (\mathcal{Q}\cap\mathcal{S}_j)$. This establishes that ρ is non-decreasing. Next consider any two sets $\mathcal{P} \subseteq \mathcal{S}$ and $\mathcal{Q} \subseteq \mathcal{S}$. We have

$$\begin{aligned}&\rho(\mathcal{P}) + \rho(\mathcal{Q}) - \rho(\mathcal{P}\cap\mathcal{Q}) - \rho(\mathcal{P}\cup\mathcal{Q}) \\ = &\sum_{j\in[M]} (H(U_{\mathcal{P}\cap\mathcal{Q}\cap\mathcal{S}_j}|Y_j) + H(U_{(\mathcal{P}\cup\mathcal{Q})\cap\mathcal{S}_j}|Y_j) - H(U_{\mathcal{P}\cap\mathcal{S}_j}|Y_j) - H(U_{\mathcal{Q}\cap\mathcal{S}_j}|Y_j)) \\ = &\sum_{j\in[M]} (H(U_{\mathcal{P}\cap\mathcal{Q}^c\cap\mathcal{S}_j}|U_{\mathcal{Q}\cap\mathcal{S}_j}, Y_j) - H(U_{\mathcal{P}\cap\mathcal{Q}^c\cap\mathcal{S}_j}|U_{\mathcal{P}\cap\mathcal{Q}\cap\mathcal{S}_j}, Y_j)) \\ \leq &\ 0\end{aligned}$$

which implies that ρ is a rank function of a contra-polymatroid. To see that this contra-polymatroid is equal to $\Phi(\boldsymbol{\xi})$, simply note that evaluating the rank function ρ and writing the corresponding inequality for every subset of $\mathcal{S}_j$ gives the list of inequalities for node j. The collection of these inequalities over $j \in [M]$ then yields $\Phi(\boldsymbol{\xi})$. Finally, note that evaluating the rank function at any collection of indices corresponding to message sent from different encoders simply sums the corresponding individual inequalities for the different encoders. $\square$

Corollary 1: For each $\boldsymbol{\xi} \in \Xi(\mathbf{d})$, the generating vertices of the polyhedron $\Phi(\boldsymbol{\xi})$ are exactly $\{\boldsymbol{\phi}(\boldsymbol{\pi})|\boldsymbol{\pi} \in \Pi(\mathcal{S})\}$ where $\Pi(\mathcal{S})$ is the set of permutations of the indices in

$\mathcal{S}$, and $\boldsymbol{\phi}(\boldsymbol{\pi})$ is the vector given by

$$\phi_{\pi(1)}(\boldsymbol{\pi}) \triangleq \rho(\pi(1)) = I(U_{\pi(1)}; Y_{[M]})$$

and for every $i \in \{2, \ldots, |\mathcal{S}|\}$

$$\begin{aligned} \phi_{\pi(i)}(\boldsymbol{\pi}) &\triangleq \rho(\{\pi(1), \ldots, \pi(i)\}) - \rho(\{\pi(1), \ldots, \pi(i-1)\}) \\ &= I(U_{\pi(i)}; U_{\{\pi(1),\ldots,\pi(i-1)\}}, Y_{[M]}) \end{aligned} \tag{4.17}$$

and where ρ is the rank function defined in (4.16). Additionally, for any $\boldsymbol{\lambda} \in \mathbb{R}_+^{|\mathcal{S}|}$, then the solution to the linear program $\min_{\boldsymbol{\phi} \in \Phi(\boldsymbol{\xi})} \boldsymbol{\lambda} \cdot \boldsymbol{\phi}$ is attained by $\boldsymbol{\phi}(\boldsymbol{\pi})$ for $\boldsymbol{\pi}$ any permutation of the elements of $\mathcal{S}$ such that $\lambda_{\pi(1)} \geq \cdots \geq \lambda_{\pi(|\mathcal{S}|)}$.

Proof: These are standard properties of contra-polymatroids [89]. See, for instance, Lemma 3.3 of [90]. □

We next use these structural properties of the achievable rate distortion region to simplify this bound to some simpler problems.

4.4 Simplification of Inner Bound to Simpler Problems

Because we have argued that the collaborative distributed estimation problem is essentially a hybrid between a collection of CEO problems and a multiple descriptions problem, it is important to show that the inner bound we have given specializes to known inner bounds for these problems in special cases.

4.4.1 Simplification to Multiple Descriptions Problem

The multiple descriptions problem for two descriptions can be obtained as a special case of our collaborative estimation problem for $M = 4$ nodes. Only one node,

say node 1, gets to make observations which it would like to inform the other 3 network nodes about, so that $Y_1^{(n)} = T^{(n)}$ and $Y_i^{(n)} = 0$ for all $i \neq 1$. Additionally, node 1 structures its encodings so that nodes 2 and 3 receive different encodings, while node 4 receives everything that is available to node 2 and 3. The coding strategy introduced in [91] to this problem can be accomplished by dividing $Q_{1\to\{4\}}$ up into two parts $Q_{1\to\{4\}} = (Q^1_{1\to\{4\}}, Q^2_{1\to\{4\}})$ containing $\Delta_1 \geq 0$ and $\Delta_2 \geq 0$ bits per symbol with $\Delta_1 + \Delta_2 = R_{1\to\{4\}}$ and forming two descriptions $X_1 \triangleq (Q_{1\to\{2,4\}}, Q^1_{1\to\{4\}})$ and $X_2 \triangleq (Q_{1\to\{3,4\}}, Q^2_{1\to\{4\}})$. When only one of the two descriptions X_1 or X_2 is available, the achievability coding strategy introduced in [91] simply discards the part of the description associated with $Q_{1\to\{4\}}$ and utilizes only $U_{1\to\{3,4\}}$ or $U_{1\to\{2,4\}}$, respectively. When both descriptions are available, the achievability coding strategy introduced in [91] uses all of the encodings $(Q_{1\to\{2,4\}}, Q_{1\to\{3,4\}}, Q_{1\to\{4\}})$. Additionally, since $R_1 = R_{2,4} + \Delta_1$ and $R_2 = R_{3,4} + \Delta_2$, we can remove the redundant variables Δ_1 and Δ_2, and rewrite the constraint for R_4 as $R_4 = R_1 - R_{2,4} + R_2 - R_{3,4}$. These identifications may be summarized with the following notation

$$U_{1\to\{2,3\}} \triangleq U_{2,3},\ U_{1\to\{3,4\}} \triangleq U_{3,4},\ U_{1\to\{4,\}} \triangleq U_4 \tag{4.18}$$

$$R_{1\to\{2,3\}} \triangleq R_{2,3},\ R_{1\to\{3,4\}} \triangleq R_{3,4},\ R_{1\to\{4,\}} \triangleq R_4 \tag{4.19}$$

$$\tilde{R}_{1\to\{2,3\}} \triangleq \tilde{R}_{2,3},\ \tilde{R}_{1\to\{3,4\}} \triangleq \tilde{R}_{3,4},\ \tilde{R}_{1\to\{4\}} \triangleq \tilde{R}_4 \tag{4.20}$$

$$U_{j\to\mathcal{A}} = \emptyset,\ R_{j\to\mathcal{A}} = \emptyset,\ \tilde{R}_{j\to\mathcal{A}} = \emptyset \text{ all other } \mathcal{A} \tag{4.21}$$

Where the auxiliary random variables $U_4, U_{2,4}, U_{3,4}$ are selected such that

$$p(U_4, U_{2,4}, U_{3,4}, T) = p(T)p(U_4, U_{2,4}, U_{3,4}|T) \tag{4.22}$$

$$D_1 \geq \mathbb{E}\left[d(T, \hat{T}_2)\right],\ D_2 \geq \mathbb{E}\left[d(T, \hat{T}_3)\right],\ D_0 \geq \mathbb{E}\left[d(T, \hat{T}_4)\right]$$

Under these identifications, the inner bound becomes

$$\begin{aligned} R_4 &\geq \tilde{R}_4 - H(U_4) + H(U_4|U_{3,4}, U_{2,4}) && (4.23)\\ R_{2,4} &\geq \tilde{R}_{2,4} && (4.24)\\ R_{3,4} &\geq \tilde{R}_{3,4} && (4.25) \end{aligned}$$

Having the inequalities $R_1 \geq R_{2,4},\ R_2 \geq R_{3,4}$ (because $\Delta_1, \Delta_2 \geq 0$) in hand, we replace R_4 with $R_1 - R_{2,4} + R_2 - R_{3,4}$ in (4.23) and use the inequalities (4.23)-(4.25) to obtain a bound on the rate region (R_1, R_2) which is given by

$$\begin{aligned} R_1 &\geq \tilde{R}_{2,4} && (4.26)\\ R_2 &\geq \tilde{R}_{3,4} && (4.27)\\ R_1 + R_2 &\geq \tilde{R}_{2,4} + \tilde{R}_{3,4} + \tilde{R}_4 - H(U_4) + H(U_4|U_{3,4}, U_{2,4}) && (4.28) \end{aligned}$$

We note that the minimum of $\tilde{R}_{2,4} + \tilde{R}_{3,4} + \tilde{R}_4$ from the encoder inequalities to be

$$H(U_4) + H(U_{2,4}) + H(U_{3,4}) - H(U_4, U_{2,4}, U_{3,4}|T)$$

Thus right hand side of (4.28) becomes

$$\begin{aligned} & H(U_{2,4}) + H(U_{3,4}) - H(U_4, U_{2,4}, U_{3,4}|T) + H(U_4|U_{2,4}, U_{3,4})\\ = \ & H(U_{2,4}) + H(U_{3,4}) - H(U_4, U_{2,4}, U_{3,4}|T) + H(U_4, U_{2,4}, U_{3,4}) - H(U_{2,4}, U_{3,4})\\ = \ & I(U_{2,4}; U_{3,4}) + I(T; U_4, U_{2,4}, U_{3,4}) \end{aligned}$$

We next point out that by the contra-polymatroid property of the source encoder region describing the collection of variables $\tilde{R}_{2,4}, \tilde{R}_{3,4}, \tilde{R}_4$ by Corollary 1, this minimum is attained for 6 (permutations of $\lambda_1 = 1, \lambda_2 = 1, \lambda_3 = 1$) possible solutions of

$\tilde{R}_{2,4}, \tilde{R}_{3,4}, \tilde{R}_4$. However, we are interested in only two of the 6 solutions which are useful in finding the region of (R_1, R_2) and present the values of $\tilde{R}_{2,4}, \tilde{R}_{3,4}$ below.

1. $\tilde{R}_{2,4} = I(U_{2,4}; T), \quad \tilde{R}_{3,4} = I(U_{3,4}; U_{2,4}, T)$

2. $\tilde{R}_{2,4} = I(U_{2,4}; U_{3,4}, T), \quad \tilde{R}_{3,4} = I(U_{3,4}; T)$

Using time sharing argument of these two solutions we write the region of rates (R_1, R_2) as

$$\begin{aligned} R_1 &\geq I(U_{2,4}; T) + \alpha\ I(U_{2,4}; U_{3,4}|T) && (4.29) \\ R_2 &\geq I(U_{3,4}; T) + (1-\alpha)\ I(U_{2,4}; U_{3,4}|T) && (4.30) \\ R_1 + R_2 &\geq I(U_{2,4}; U_{3,4}) + I(T; U_4, U_{2,4}, U_{3,4}) && (4.31) \end{aligned}$$

where $0 \leq \alpha \leq 1$. We next show that any point in the achievable rate region (ECG region) proved in [91] also lies in the region we proved above. To prove this, we rewrite the EGC region in the following form

$$\begin{aligned} r_1 &\geq I(U_{2,4}; T) \\ r_2 &\geq \max\{I(U_{3,4}; T), I(U_{2,4}; U_{3,4}) + I(T; U_4, U_{2,4}, U_{3,4}) - r_1\} \end{aligned}$$

and let

$$\alpha = \min\left\{\frac{r_1 - I(U_{2,4}; T)}{I(U_{2,4}; U_{3,4}|T)}, 1\right\} \tag{4.32}$$

Then

$$R_1 \geq \min\left\{r_1, I(U_{2,4}; T) + I(U_{2,4}; U_{3,4}|T)\right\} \leq r_1 \tag{4.33}$$

and

$$\begin{aligned} R_2 &\geq I(U_{3,4};T) + I(U_{2,4};U_{3,4}|T) - \min\{r_1 - I(U_{2,4};T),\ I(U_{2,4};U_{3,4}|T)\} \\ &= \max\{I(U_{3,4};T), I(U_{2,4};T) + I(U_{3,4};T) + I(U_{2,4};U_{3,4}|T) - r_1\} \\ &\leq r_2 \end{aligned}$$

In the above proof we used the following inequality which can be easily proved.

$$\begin{aligned} & I(U_{2,4};U_{3,4}) + I(T;U_4,U_{2,4},U_{3,4}) \\ \geq\ & I(U_{2,4};T) + I(U_{3,4};T) + I(U_{2,4};U_{3,4}|T) \end{aligned}$$

This completes the proof that our inner bound contains every point in the EGC region.

4.4.2 Simplification to CEO problem

We next show that CEO problem can be obtained as a simplification of our model and that our inner bound simplifies the Berger-Tung inner bound for this case. To see this, suppose that the nodes $i \in [M] \setminus M$ observe the common phenomenon embodied by the sequence $T^{(n)}$ and send one message each to the CEO node M. Using these messages received from the nodes $i \in [M-1]$, the CEO node produces an estimate $\hat{T}$ ($\hat{T}_M = \hat{T}$) of T such that the expected distortion $E[d(T,\hat{T})] < D$.

Since the nodes $i \in [M-1]$ send messsages only to node M, we set the rates corresponding to the other messages to 0 and redefine the rates and variables relevent

to this problem as follows.

$$R_{j\to M} \triangleq R_j,\ U_{j\to M} \triangleq U_j\ \forall\ j \in [M-1]$$
$$R_{j\to\mathcal{A}} = 0,\ \tilde{R}_{j\to\mathcal{A}} = 0, U_{j\to\mathcal{A}} = \emptyset \text{ all other } \mathcal{A}, j \in [M-1]$$
$$D_M \triangleq D,\ R_{M\to\mathcal{A}} = 0,\ \tilde{R}_{M\to\mathcal{A}} = 0, U_{M\to\mathcal{A}} = \emptyset$$

Note that the random vectors $\boldsymbol{\xi} = (U_{[M-1]})$ satisfy the following constraints.

- $T, Y_{[M-1]\setminus i}, U_{[M-1]\setminus j} \leftrightarrow Y_j \leftrightarrow U_j$ for all $j \in [M-1]$
- There exists a decoding function $g : \mathcal{U}_{[M-1]} \to \hat{\mathcal{T}}$ such that $D > E\left[d(T,\hat{T})\right]$

If we denote the set $[M-1]$ as $\mathcal{D} := [M-1]$ as, then $\Phi(\boldsymbol{\xi})$ becomes

$$\Phi(\boldsymbol{\xi}) = \{\tilde{R}_{\mathcal{D}} | \tilde{R}_j > H(U_j - H(U_j|Y_j), \forall j \in [M-1]\}$$

Here, $\tilde{R}_j$ can be selected such that $\tilde{R}_j = I(U_j;Y_j) + \epsilon_j$ for all $j \in [M-1]$ where ϵ_j can be made arbitrarily small. Note that selecting the rates so will not change the rate region. If we select $\tilde{R}_j = I(U_j;Y_j) + \epsilon_j$, there will be only 1 rate vector $\tilde{R}_{\mathcal{D}}$ in the set $\Phi(\boldsymbol{\xi})$. Thus, Ψ is only a function of $\boldsymbol{\xi}$, i.e. $\Psi(\boldsymbol{\xi}, \phi) = \Psi(\boldsymbol{\xi})$. Hence, $\Psi(\boldsymbol{\xi})$ is the collection of rate vectors $R_{\mathcal{D}} \geq 0$ obeying

$$\begin{aligned}
\sum_{j\in\mathcal{C}} R_j &> \sum_{j\in\mathcal{C}} (\tilde{R}_j - H(U_j)) + H(U_{\mathcal{C}}|U_{\mathcal{D}\setminus\mathcal{C}}) \\
&= H(U_{\mathcal{C}}|U_{\mathcal{D}\setminus\mathcal{C}}) - \sum_{j\in\mathcal{C}} H(U_j|Y_j) \\
&= H(U_{\mathcal{C}}|U_{\mathcal{D}\setminus\mathcal{C}}) - H(U_{\mathcal{C}}|Y_{\mathcal{C}}) \\
&= H(U_{\mathcal{C}}|U_{\mathcal{D}\setminus\mathcal{C}}) - H(U_{\mathcal{C}}|Y_{\mathcal{C}}, U_{\mathcal{D}\setminus\mathcal{C}}) \\
&= I(U_{\mathcal{C}}; Y_{\mathcal{C}}|U_{\mathcal{D}\setminus\mathcal{C}})
\end{aligned}$$

for all $\mathcal{C} \subseteq \mathcal{D}$. Here, we have used the facts that node M (CEO) does not have any

side information ($Y_M = 0$) and $U_{\mathcal{C}} \leftrightarrow Y_{\mathcal{C}} \leftrightarrow U_{\mathcal{D}\setminus\mathcal{C}}$. Thus the inner bound for the rate-distortion region for the CEO problem becomes

$$\mathcal{RD}_{in} = \left\{ (R_{[M-1]}, D) \middle| R_{[M-1]} \in \bigcup_{\boldsymbol{\xi}\in\boldsymbol{\Xi}(D)} \Psi(\boldsymbol{\xi}) \right\}$$

where $\boldsymbol{\Xi}(D)$ is the collection of random vectors $\boldsymbol{\xi}$. This is exactly the Berger Tung inner bound for the CEO problem given in [15].

4.4.3 Simplification to Side Information May Be Absent at the Decoder

The "side information may be absent at the decoder" problem studied by Heegard and Berger in [23] can also be obtained as a simplification of our model. To see this, let the number of nodes $M = 3$ and, suppose that node 3 directly observes the source, i.e. $Y_3 = T$, and node 1 has side information about the source $Y_1 = Y$ while node 2 has no side information. Also, suppose that node 3 sends a common description to both $1, 2$ and an individual description to only node 1 as it is implicitly done in [23]. We show that sum of the rates of these two descriptions derived from our inner bound is equal to the rate-distortion function proved for the sum-rate in [23].

To prove this, set the rates and variables which are not involved in the problem zero and redefine the necessary rates and variables as follows.

$$Y_1 \triangleq Y,\ Y_3 \triangleq T, \quad \text{all other } Y_i = \emptyset \tag{4.34}$$

$$U_{3\to 1} \triangleq U,\ U_{3\to\{1,2\}} \triangleq W, \quad \text{all other } U_{j\to\mathcal{A}} = \emptyset \tag{4.35}$$

$$\tilde{R}_{3\to 1} \triangleq \tilde{R}_1,\ \tilde{R}_{3\to\{1,2\}} \triangleq \tilde{R}_{1,2}, \quad \text{all other } \tilde{R}_{j\to\mathcal{A}} = \emptyset \tag{4.36}$$

$$R_{3\to 1} \triangleq R_1,\ R_{3\to\{1,2\}} \triangleq R_{1,2}, \quad \text{all other } R_{j\to\mathcal{A}} = \emptyset \tag{4.37}$$

Note that the variables T, Y, U, W, V satisfy the following conditions.

1. $Y \leftrightarrow T, V \leftrightarrow U, W$.

2. There exist functions $\hat{T}_1(U, W, Y)$ and $\hat{T}_2(W)$ such that $E[d_1(T, \hat{T}_1)] \leq D_1$ and $E[d_2(T, \hat{T}_2)] \leq D_2$.

With the redefined variables, the constraints on $\tilde{R}_1, \tilde{R}_{1,2}$ become

$$\begin{aligned} \tilde{R}_1 &\geq H(U|V) - H(U|T,V) && (4.38) \\ \tilde{R}_{1,2} &\geq H(W|V) - H(W|T,V) && (4.39) \\ \tilde{R}_1 + \tilde{R}_{1,2} &\geq H(U|V) + H(W|V) - H(U,W|T,V) && (4.40) \end{aligned}$$

and the constraints on $R_1, R_{1,2}$ become

$$\begin{aligned} R_1 &\geq \tilde{R}_1 - H(U|V) + H(U|W,Y,V) && (4.41) \\ R_{1,2} &\geq \tilde{R}_{1,2} - H(W|V) + H(W|U,Y,V) && (4.42) \\ R_{1,2} &\geq \tilde{R}_{1,2} && (4.43) \\ R_1 + R_{1,2} &\geq \tilde{R}_1 - H(U|V) + \tilde{R}_{1,2} - H(W|V) + H(U,W|Y,V) && (4.44) \end{aligned}$$

Observing that the tight bound on $R_1 + R_{1,2}$ comes from adding (4.41) and (4.43), we write

$$R_1 + R_{1,2} \geq \tilde{R}_1 + \tilde{R}_{1,2} - H(U|V) + H(U|W,Y,V) \quad (4.45)$$

Using the bound on $\tilde{R}_1 + \tilde{R}_{1,2}$ in (4.40), we obtain

$$\begin{aligned} R_1 + R_{1,2} &\geq \tilde{R}_1 + \tilde{R}_{1,2} - H(U|V) + H(U|W,Y,V) && (4.46) \\ &\geq H(W|V) - H(U,W|T,V) + H(U|W,Y,V) && (4.47) \\ &= I(T;W|V) + (T;U|W,Y,V) && (4.48) \end{aligned}$$

This is exactly the rate-distortion function for the sum-rate which Heegard and Berger

proved in [23].

4.5 Conclusions

We analyzed low-communication algorithms for collaborative distributed estimation via multiterminal information theory. We argued that the proper model for a distributed source code for collaborative distributed estimation involves multiple multicast messages from each encoder rather than unicast messages, yielding a hybrid coding problem between multiple descriptions and the CEO problem. An achievable rate region which hybridized the Berger Tung inner bound and multiple descriptions proof techniques were presented. The inner bound was shown to be equal to the known bounds for some simpler problems by exploiting the structural properties of the rate region. An outer bound to the rate distortion region was also derived from basic information theory principles.

5. Low-Complexity/Low-communication Collaborative Estimation Algorithms

In the previous two chapters we developed collaborative estimation algorithms that require either low complexity or low communication. As it was discussed in Chapter 1, a collaborative estimation algorithm that is implemented in practice should preferably have both low complexity and low communication. In this chapter we study such low-complexity/low-communication collaborative estimation algorithm.

We develop such an algorithm for an estimation problem in which the nodes interactively communicate with each other over several rounds. In particular, we consider the following multiround protocol for collaborative estimation among M nodes. At each round in the protocol, exactly one node broadcasts a message to all other nodes in the network. The nodes are selected to transmit in a round-robin fashion during subsequent rounds. The broadcasted message consists of $M-1$ "descriptions", and the mth description is only utilized by the $M-m$ best decoders, $m \in \{1, \ldots, M-1\}$. For instance, a single round of communication among $M = 3$ nodes is depicted in Fig. 5.1. The observations and previously received messages at the two receiving nodes act as side information at the two decoders. The message broadcasted to the two decoders consists of two descriptions: a common description is encoded for both decoders, and an individual refinement description is encoded for the decoder with better side information [23].

We first derive relevant theoretical limits from multiterminal source coding in order to characterize efficient tradeoffs between communication and estimation performance and inspire the architecture of the collaborative estimation code. Although the theoretical limits characterize a region of rates and distortions as achievable, in practice only a subset of this region is achievable with low complexity encoders and

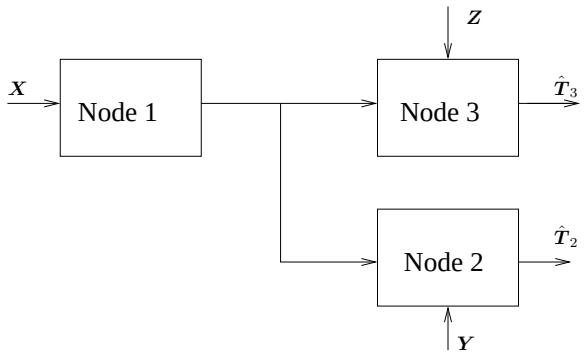


Figure 5.1: A network of 3 nodes make indirect observations of the source, communicate with each other and estimate the underlying source.

decoders mimicking the same construction. The main goal of this work is to develop a practical collaborative estimation scheme that approaches the theoretical bound when the source and observations are jointly Gaussian distributed and the distortion measure is squared error.

Towards this goal, we develop such an algorithm approaching these performance limits for the single round case by utilizing low complexity machine learning and signal processing tools from modern practical coding theory. The developed algorithm utilizes successively refined trellis coded quantization (SR-TCQ) to provide necessary diverse descriptions of the source, and low-density parity-check (LDPC) codes to provide an efficient means of compressing these descriptions for low complexity belief propagation decoding with side information. For our multiround collaborative estimation, at each round we employ the code we developed for the single round case where we use the estimates from the previous round as the observations/side information.

For notational convenience and ease of exposition, we focus on the 3-node network in Fig. 5.1 which is the lowest dimensional non-trivial network for this source coding architecture. However, the same techniques can be easily extended to an arbitrary

M-node network and these extensions are provided in the end of this chapter in Section 5.6. Comparing the communication vs. estimate quality tradeoff performance attained by the developed low complexity scheme with that obtainable with a theoretical argument, an average distortion gap of only 1.0dB and 1.6dB at average rates of 3.32 b/s and 2.33 b/s, respectively, is observed.

5.1 Problem Formulation

Consider a network of 3 nodes where the nodes make observations of an underlying source sequence $\{T^{(n)}\}_{n=1}^{N}$ as $\{Y_i^{(n)}\}_{n=1}^{N}$, $i = 1, 2, 3$. The source and observation sequences are i.i.d. (temporally) according to $p(t, y_1, y_2, y_3)$. The nodes communicate with each other over several rounds and at the end of the collaboration they estimate the underlying sequence as $\{\hat{T}_1^{(n)}\}_{n=1}^{N}, \{\hat{T}_2^{(n)}\}_{n=1}^{N}, \{\hat{T}_3^{(n)}\}_{n=1}^{N}$. During the collaboration, they follow the following communication scheme.

At each round only one node is allowed to transmit and the nodes take turns encoding the messages, for instance, $1, 2, 3, 1, 2, \ldots$. The node that transmits at a certain round encodes its message based on its observations and the messages heard in the previous rounds. In particular, it encodes a common description to both decoders and an individual description only to the best decoder which is determined by (5.5). The decoders use their observations and the messages heard in the previous rounds as the side information. The nodes communicate in this fashion for K rounds before they make estimates of the source sequence based on their observations and the messages received.

Let $\mu(k)$, $\lambda(k)$ and $\phi(k)$ denote the encoder, the best decoder and the worst

decoder at round k, respectively, where

$$\mu(k) = \begin{cases} k \mod 3 & , (k \mod 3) \neq 0 \\ 3 & , (k \mod 3) = 0 \end{cases}$$

Also, let $\mathbf{1}[.]$ denote the indicator function.

The rate distortion vector $(\{R^k\}_{k=1}^K, D_1, D_2, D_3)$ is said to be achievable if there exist the encoding functions $\{f^k_{\mu(k),1}, f^k_{\mu(k),2}\}_{k=1}^K$

$$f^k_{\mu(k),1} : \boldsymbol{\mathcal{Y}}_{\mu(k)} \times [L_1^1] \times \ldots \times [L_1^{k-1}] \times \{[L_2^\ell]\mathbf{1}[\lambda(\ell) = \mu(k)]\}_{\ell=1}^{k-1} \to [L_1^k]$$
$$f^k_{\mu(k),2} : \boldsymbol{\mathcal{Y}}_{\mu(k)} \times [L_1^1] \times \ldots \times [L_1^{k-1}] \times \{[L_2^\ell]\mathbf{1}[\lambda(\ell) = \mu(k)]\}_{\ell=1}^{k-1} \times [L_1^k] \to [L_2^k]$$

and the decoding functions $\{g_i\}_{i=1}^3$

$$g_i : \boldsymbol{\mathcal{Y}}_i \times [L_1^1] \times \ldots \times [L_1^K] \times \{[L_2^k]\mathbf{1}[\lambda(k) = i]\}_{k=1}^K \to \hat{T}_i^N$$

such that

$$\frac{1}{N}(\log_2(L_1^k) + \log_2(L_2^k)) \leq R^k$$

and

$$\frac{1}{N}\sum_{n=1}^{N} E[d_i(T^{(n)}, \hat{T}_i^{(n)})] \leq D_i \quad , i = 1, 2, 3$$

Let $\bar{\mathcal{RD}}$ be the set of all achievable $(\{R^k\}_{k=1}^K, D_1, D_2, D_3)$. Then, the convex hull $\mathcal{RD} = \text{conv}(\bar{\mathcal{RD}})$ of $\bar{\mathcal{RD}}$ gives the rate distortion region for this problem. We derive an inner bound to $\mathcal{RD}$ in Section 5.2.

Our main goal is to design a practical source coding scheme for this collaborative estimation problem. In particular, we aim to design the practical coding scheme when the source and the observations are jointly distributed according to Gaussian

distribution and the distortion measure is squared error. We define

$$Y_1 \triangleq X, \quad Y_2 \triangleq Y, \quad Y_3 \triangleq Z$$

and model the source and observations as follows.

$$X = T + N_1 \tag{5.1}$$

$$Y = T + N_2 \tag{5.2}$$

$$Z = T + N_3 \tag{5.3}$$

where $T \sim \mathcal{N}(0, \sigma_t^2)$ and $N_i \sim \mathcal{N}(0, \sigma_{n_i}^2),\ i = 1, 2, 3$ such that $\sigma_{n_1}^2 \leq \sigma_{n_2}^2 \leq \sigma_{n_3}^2$.

The nodes take turns transmitting messages to the other nodes for K rounds. At each round k, the encoder encodes its observations and the messages heard in the previous rounds into a common description to both "worst decoder", "best decoder" and an individual description only to the "best decoder". At the end of K rounds, the nodes are expected to reproduce the underlying source T such that

$$E[d_i(T, \hat{T}_i)] \leq D_i, \quad i = 1, 2, 3$$

We begin our study on this collaborative estimation problem with the theoretical bounds for the problem.

5.2 Theoretical Bound

We derive a theoretical bound on the achievable rate distortion region for our problem by closely following the techniques in [23]. The rate distortion region studied in [23] is equivalent to the rate distortion region in the first round of our collaborative estimation problem, and can provide an achievability construction for the multiround

problem by sewing together single round bounds from [23] in a manner similar to the way Selpian-Wolf/Wyner-Ziv regions are sewn together in [13, 92, 93, 94, 95, 96].

We present an inner bound to the rate distortion region and a sketch of the proof now. A detailed proof of the theorem with the probability of error analysis is given in the Appendix.

Theorem 3:

Let $\Phi(\{R^k\}_{k=1}^K, D_1, D_2, D_3)$ be the set of random vectors $\{W_k, U_k\}_{k=1}^K$ that are jointly distributed with T, Y_1, Y_2, Y_3 such that the following conditions are satisfied.

1. $W_k, U_k \leftrightarrow Y_{\mu(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell) = \mu(k)]\}_{\ell=1}^{k-1} \leftrightarrow$
   $Y_{\lambda(k)}, Y_{\phi(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell) = \lambda(k)]\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell) = \phi(k)]\}_{\ell=1}^{k-1}$

2. There exist decoding functions $\bar{g}_i$ such that $E[d_i(T, \bar{g}_i(Y_i, \{W_k\}_{k=1}^K, \{U_k \mathbf{1}[\lambda(k) = i]\}_{k=1}^K))] \leq D_i$ for $i = 1, 2, 3$.

An inner bound to $\mathcal{RD}$ is given by

$$
\begin{aligned}
\mathcal{RD}_{in} \;=\; \Big\{ & (\{R^k\}_{k=1}^K, D_1, D_2, D_3) : \\
& R^k \geq I(W_k; Y_{\mu(k)}, \{U_\ell \mathbf{1}[\lambda(\ell) = \mu(k)]\}_{\ell=1}^{k-1} \mid \\
& \qquad Y_{\phi(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell) = \phi(k)]\}_{\ell=1}^{k-1}) + \\
& \quad I(U_k; Y_{\mu(k)}, \{U_\ell \mathbf{1}[\lambda(\ell) = \mu(k)]\}_{\ell=1}^{k-1} \mid \\
& \qquad W_k, Y_{\lambda(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell) = \lambda(k)]\}_{\ell=1}^{k-1}) \\
& E[d_i(T, \hat{T}_i)] \leq D_i, \;\; i = 1, 2, 3, \\
& \{W_k, U_k\}_{k=1}^K \in \Phi(\{R^k\}_{k=1}^K, D_1, D_2, D_3) \Big\}
\end{aligned} \tag{5.4}
$$

□

Note:

The best decoder at round k is determined by

$$\lambda(k) = \arg \min_{i \in \{1,2,3\} \setminus \mu(k)} h\Big(W_k |\ Y_i, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell) = i]\}_{\ell=1}^{k-1}\Big) \tag{5.5}$$

Proof:

Distribution of Auxiliary Variables:

Select the joint distribution $p(w_1, u_1, \ldots, w_K, u_K | t, y_1, y_2, y_3)$ such that the conditions in the theorem are satisfied.

Codebook Generation:

We generate 2 codebooks for each round of communication. We generate the first codebook for round k with $\tilde{L}_1^k$ length-N codewords $\boldsymbol{W}_k$ by drawing the elements i.i.d. according to $p(w_k)$. Index these codewords by $i_k \in \{1, \ldots, \tilde{L}_1^k\}$. Then, we build the second codebook by generating $\tilde{L}_2^k$ codewords $\boldsymbol{U}_k$ for each $\boldsymbol{W}_k(i_k),\ i_k \in \{1, \ldots, \tilde{L}_1^k\}$ i.i.d. according to $p(u_k|w_k)$. Index the codewords $\boldsymbol{U}_k$ by $(i_k, j_k) \in \{1, \ldots, \tilde{L}_1^k\} \times \{1, \ldots, \tilde{L}_2^k\}$. Then, partion the codewords $\boldsymbol{W}_k$ into L_1^k bins, and for each $\boldsymbol{W}_k(i_k)$ partition the codewords $\boldsymbol{U}_k$ into L_2^k bins by randomly and uniformly assigning the codewords to one of the bins. Index these bins by $b_1^k \in \{1, \ldots, L_1^k\}$ and $b_{2,i_k}^k \in \{1, \ldots, L_2^k\}$, respectively. Also, denote the set of codewords in the bins with indices b_1^k and b_{2,i_k}^k by $\mathcal{B}_1^k(b_1^k)$ and $\mathcal{B}_{2,i_k}^k(b_{2,i_k}^k)$, respectively.

Encoding:

Assuming that encoding and decoding are done without any error upto $k-1$ rounds, we describe the encoding process at round k. At round k, node $\mu(k)$ selects the first codeword $\boldsymbol{W}_k(i_k)$ such that

$$(\boldsymbol{Y}_{\mu(k)}, \{\boldsymbol{W}_\ell(i_\ell)\}_{\ell=1}^{k-1}, \{\boldsymbol{U}_\ell(i_\ell, j_\ell)\mathbf{1}[\lambda(\ell) = \mu(k)]\}_{\ell=1}^{k-1}, \boldsymbol{W}_k(i_k)) \in \mathcal{A}_\epsilon^N$$

If there is no such a codeword, node $\mu(k)$ selects the first codeword in the codebook. Then, node $\mu(k)$ selects the first codeword $\boldsymbol{U}_k(i_k, j_k)$ such that

$$(\boldsymbol{Y}_{\mu(k)}, \{\boldsymbol{W}_\ell(i_\ell)\}_{\ell=1}^{k}, \{\boldsymbol{U}_\ell(i_\ell, j_\ell)\mathbf{1}[\lambda(\ell) = \mu(k)]\}_{\ell=1}^{k-1}, \boldsymbol{U}_k(i_k, j_k)) \in \mathcal{A}_\epsilon^N$$

If there is no such a codeword, node $\mu(k)$ selects the first codeword in the codebook. Node $\mu(k)$ sends the bin index b_1^k to both decoders and b_{2,i_k}^k to only the best decoder, where $W_k(i_k) \in \mathcal{B}_1^k(b_1^k)$ and $U_k(i_k, j_k) \in \mathcal{B}_{2,i_k}^k(b_{2,i_k}^k)$.

Decoding:

Assuming that encoding and decoding are done without any error upto $k-1$ rounds, we describe the decoding process at round k. At round k, node $\lambda(k)$ decodes the codewords by selecting $\boldsymbol{W}_k(i'_k) \in B_1^k(b_1^k)$ and $\boldsymbol{U}_k(i'_k, j'_k) \in B_{2,i'_k}^k(b_{2,i'_k}^k)$ such that

$$(\boldsymbol{Y}_{\lambda(k)}, \{\boldsymbol{W}_\ell(i_\ell)\}_{\ell=1}^{k-1}, \{\boldsymbol{U}_\ell(i_\ell, j_\ell)\mathbf{1}[\lambda(\ell) = \lambda(k)]\}_{\ell=1}^{k-1}, \boldsymbol{W}_k(i'_k), \boldsymbol{U}_k(i'_k, j'_k)) \in \mathcal{A}_\epsilon^N$$

Node $\phi(k)$ decodes the codeword by selecting $\boldsymbol{W}_k(\tilde{i}_k) \in B_1^k(b_1^k)$ such that

$$(\boldsymbol{Y}_{\phi(k)}, \{\boldsymbol{W}_\ell(i_\ell)\}_{\ell=1}^{k-1}, \{\boldsymbol{U}_\ell(i_\ell, j_\ell)\mathbf{1}[\lambda(\ell) = \phi(k)]\}_{\ell=1}^{k-1}, \boldsymbol{W}_k(\tilde{i}_k)) \in \mathcal{A}_\epsilon^N$$

Reconstruction:

When there is no error in the encoding and decoding processes, the nodes reconstruct the source sequence as

$$\hat{T}_i^{(n)} = \bar{g}_i(Y_i^{(n)}, \{\boldsymbol{W}_k^{(n)}(i_k)\}_{k=1}^{K}, \{\boldsymbol{U}_k^{(n)}(i_k, j_k)\mathbf{1}[\lambda(k) = i]\}_{k=1}^{K})$$

$i = 1, 2, 3$. Since there exist functions $\bar{g}_i,\ i = 1, 2, 3$ that satisfy the expected distortion

conditions, when there is no error in encoding and decoding

$$E[d_i(T, \hat{T}_i)] \leq D_i \quad , \; i = 1, 2, 3$$

In Appendix C, we show that the probability of error occuring in encoding and decoding is almost 0. Having provided a theoretical bound, we next discuss a practical source coding scheme for this collaborative estimation problem.

5.3 Practical Source Coding Scheme for Single Round

In this section, we develop a practical source coding scheme for single round case for the network in Fig. 5.1 where node 1 encodes its observation X into a common description to both nodes 2,3 and an individual description only to node 2. The decoders at node 2 and 3 are expected to reproduce T such that $E[d_3(T, \hat{T}_3(W, Z))] \leq D_3$ and $E[d_2(T, \hat{T}_2(U, W, Y))] \leq D_2$ respectively, where W is the common description and U is the description intended only for node 2.

Although the achievability proof given in [23] cannot be directly applied in practice, the theoretical construction provides some insight into the practical code design. In the theoretical construction, a common description is sent to both decoders and an individual description is sent only to the best decoder. The individual description cannot be decoded without the common description, and thus the code construction to this problem is closely related to the successive refinement problem [21, 22].

Since we are encoding a continuous source, the source should be quantized first which could be also used to generate successively refinable descriptions. One easy way to generate successively refinable descriptions is to apply a uniform nested scalar quantizer (NSQ) as applied in [97, 98] for successive refinement for the Wyner-Ziv problem [12]. It was shown in [97] that uniform NSQ followed by LDPC based Slepian-

Wolf coding achieves distortions $1.29 - 3.45$dB away from the theoretical limits for the Gaussian successive refinement Wyner-Ziv problem.

In [27], trellis coded quantization (TCQ) [59] was shown to achieve better performance than the scalar quantizer for the Wyner-Ziv problem without successive refinement. This insight was further supported by Yang et al. [29] development of codes based on TCQ and LDPC codes [64] which are very close the the Wyner-Ziv theoretical limit. This suggests that TCQ may better suited to successive refinement with side information than NSQ. For this, TCQ must be adapted to successive refinability.

In this vein, Jafarkhani and Tarokh introduced a rate-scalable trellis quantization called successively refinable TCQ (SR-TCQ) in [61]. Here, we apply SR-TCQ with 2 refinement stages to get 2 descriptions of the source for our problem [99]. We apply universal source coding techniques to losslessly compress the bit-planes that represent the branches of the trellis. To compress the other bit-planes of the descriptions, we send the syndrome of a powerful LDPC channel code. This syndrome is then decoded to regain the bit-planes at the receivers by exploiting the correlation between the quantized source and the side information in a belief propagation decoder as in [29]. The use of the LDPC code and belief propagation allows for the effective exploitation of the side information with a low complexity decoder.

We begin our discussion on practical collaborative estimation algorithm construction with generation of descriptions.

5.3.1 Generating Descriptions Using SR-TCQ

We apply 2-refinement stage SR-TCQ to generate the common description W and individual description U from the source X. Suppose that in the first stage the description (W) is sent at rate r_1 and in the second stage the description (U) is sent

at rate r_2. Then, we have two sets of quantization points Q_1 and Q_2, one for each stage. The set Q_1 consists of 2^{r_1+1} quantization points, $Q_1 = \{q_i : i \in \{1, \ldots, 2^{r_1+1}\}\}$ and the set Q_2 consists of 2^{r_2+1} quantization points for each one of the quantization points $i \in Q_1$, i.e. $Q_2 = \{q_{i,j} : j \in \{1, \ldots, 2^{r_2+1}\},\ i \in \{1, \ldots, 2^{r_1+1}\}\}$.

To quantize the source sequence, we break it into blocks of N symbols and apply SR-TCQ for each block. In SR-TCQ, the Viterbi algorithm is used to find the quantization sequence which minimizes the error between quantized value and the expected value of the source given the observation at the encoder. Unlike in the single description case, here the quantization errors (distortions) at both stages should be taken into account when the distortion is minimized. Depending on the application, different weights can be given to the distortions D_2 and D_3 when the distortion to be minimized D is selected. Thus, for a block $x^{(1)}, \ldots, x^{(N)}$, D can be defined as [61]

$$D = \frac{1}{N}\{\alpha \sum_{n=1}^{N}[(E[T|x^{(n)}] - Q_1(x^{(n)})]^2 + (1-\alpha)\sum_{n=1}^{N}[(E[T|x^{(n)}] - Q_2(x^{(n)})]^2\} \quad (5.6)$$

where α is the weighting factor.

For each block, SR-TCQ outputs a sequence of quantized values $\{q_{i,j}^{(n)}\}_{n=1}^{N}$. Given $q_{i,j}^{(n)} \in Q_2$, one can easily find the quantization point $q_i^{(n)} \in Q_1$ selected for the first stage and this is the common description $W^{(n)}$ of the source $X^{(n)}$ which is sent to both decoders. To describe (before compression) W we need only r_1 bits, because only 2 cosets leave each state in trellis T_1 [59, 61]. The refinement description $U^{(n)}$ describes which of the quantization points in Q_2 nested within $q_i^{(n)}$ is selected. Again, because of the same argument made for the first stage, only r_2 bits (before compression) are required to represent the decription U.

For each symbol $n \in \{1, \ldots, N\}$, denote the bits that represent the description $W^{(n)}$ by $b_1^{1,(n)}, \ldots, b_1^{r_1,(n)}$ and denote the bits that represent the description $U^{(n)}$ by $b_2^{1,(n)}, \ldots, b_2^{r_2,(n)}$. Provided the current state in the trellis T_1, the bit $b_1^{1,(n)}$ determines

which one of the two branches (which in turn determines the coset) is selected, while the remaining bits $b_1^{2,(n)}, \ldots, b_1^{r_1,(n)}$ determine which one of the quantization points within that coset is selected. Similarly, given the common quantization level, $b_2^{1,(n)}$ determines which one of the two branches in T_2 is selected while $b_2^{2,(n)}, \ldots, b_2^{r_2,(n)}$ determine which one of the quantization points within that coset is selected. For each $m \in \{1, 2\}$ and $k \in \{1, \ldots, r_m\}$, we call the vector $\boldsymbol{b}_m^k = [b_m^{k,(1)}, \ldots, b_m^{k,(N)}]$ a bit-plane. Also, denote the collection of the bit-planes by $\boldsymbol{B}_m = [\boldsymbol{b}_m^k : \ k \in \{1, \ldots, r_m\}], \ m \in \{1, 2\}$.

We next turn our attention to lossless compression of the bit-planes discussed above.

5.3.2 Lossless Compression of the Bit-Planes

There exists significant redundancy in the SR-TCQ bit-planes in $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ that can be exploited to compress them losslessly as shown in Fig. 5.2. This lossless compression allows one to achieve the same distortions with a lower rate.

First consider the bit-planes $\boldsymbol{b}_1^1$ and $\boldsymbol{b}_2^1$. These two bit-planes have memory in them and should be compressed with an universal source coding technique (e.g. Lempel-Ziv). Also, these two bit planes can be decompressed at the decoders without the knowledge of the rest of the bit planes and can be used as side information when other bit planes are decoded. The bit-planes $\boldsymbol{B}_1 \setminus \{\boldsymbol{b}_1^1\}$ should be compressed such that they can be losslessly decoded at both decoders. Since the description compressed for node 3 can also be decoded by node 2 (because $h(X|Y) \leq h(X|Z)$), we compress $\boldsymbol{B}_1 \setminus \{\boldsymbol{b}_1^1\}$ such that it can be losslessly decompressed with side information $\boldsymbol{b}_1^1$, $\boldsymbol{b}_2^1$ and $\boldsymbol{Z}$. Given $\boldsymbol{b}_1^1$ and $\boldsymbol{b}_2^1$ the bits in the other bit-planes are independent of one another

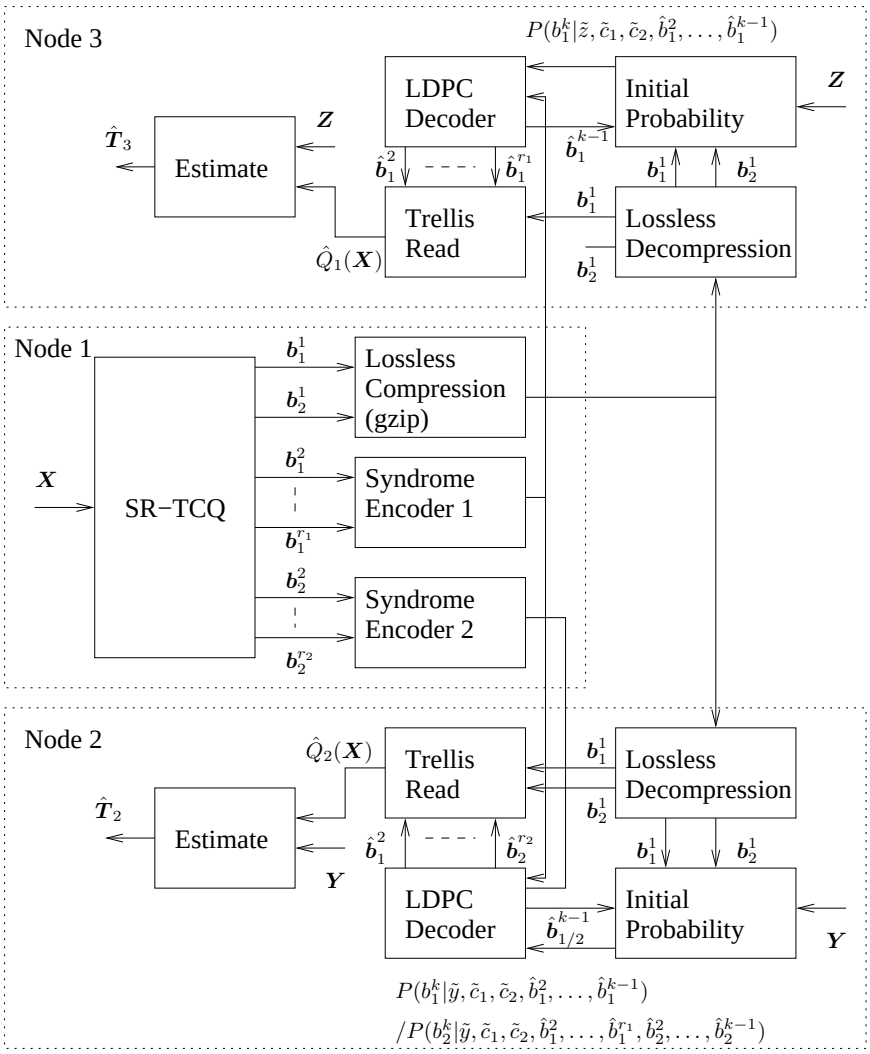


Figure 5.2: The system architecture of the practical code design we propose.

and as a result we have the following.

$$
\begin{aligned}
& H(\boldsymbol{b}_1^2,\ldots,\boldsymbol{b}_1^{r_1}|\boldsymbol{Z},\boldsymbol{b}_1^1,\boldsymbol{b}_2^1) \\
= & \sum_{n=1}^{N}\Big(H(b_1^{2,(n)}|z^{(n)},\boldsymbol{b}_1^1,\boldsymbol{b}_2^1)+\ldots+H(b_1^{r_1,(n)}|z^{(n)},\boldsymbol{b}_1^1,\boldsymbol{b}_2^1,b_1^{2,(n)},\ldots,b_1^{r_1-1,(n)})\Big) \\
= & \sum_{n=1}^{N}\Big(H(b_1^{2,(n)}|z^{(n)},c_1^{(n)},c_2^{(n)})+\ldots+ \\
& \qquad H(b_1^{r_1,(n)}|z^{(n)},c_1^{(n)},c_2^{(n)},b_1^{2,(n)},\ldots,b_1^{r_1-1,(n)})\Big) \qquad (5.7)
\end{aligned}
$$

where $c_1^{(n)}$ and $c_2^{(n)}$ are the cosets selected for symbol $x^{(n)}$ in the first and second levels, respectively. The decoder at node 2 can exploit the side information $\boldsymbol{Y}$ and $\{\boldsymbol{B}_1, \boldsymbol{b}_2^1\}$ when it decodes the bit-planes in $\boldsymbol{B}_2 \setminus \{\boldsymbol{b}_2^1\}$, and thus $\boldsymbol{B}_2 \setminus \{\boldsymbol{b}_2^1\}$ can be compressed to the following conditional entropy.

$$\begin{aligned} & H(\boldsymbol{b}_2^2, \ldots, \boldsymbol{b}_2^{r_2} | \boldsymbol{Y}, \boldsymbol{B}_1, \boldsymbol{b}_2^1) \\ = & \sum_{n=1}^{N} \Big(H(b_2^{2,(n)} | y^{(n)}, c_1^{(n)}, c_2^{(n)}, b_1^{2,(n)}, \ldots, b_1^{r_1,(n)}) + \ldots \\ & + H(b_2^{r_2,(n)} | y^{(n)}, c_1^{(n)}, c_2^{(n)}, b_1^{2,(n)}, \ldots, b_1^{r_1,(n)}, b_2^{2,(n)}, \ldots, b_2^{r_2-1,(n)}) \Big) \end{aligned} \tag{5.8}$$

Thus, the goal now is to compress these bit-planes successively to their conditional entropies. We defer our discussion on the computation of the conditional entropies $H(\boldsymbol{b}_1^2, \ldots, \boldsymbol{b}_1^{r_1} | \boldsymbol{Z}, \boldsymbol{b}_1^1, \boldsymbol{b}_2^1)$ and $H(\boldsymbol{b}_2^2, \ldots, \boldsymbol{b}_2^{r_2} | \boldsymbol{Y}, \boldsymbol{B}_1, \boldsymbol{b}_2^1)$ until Section 5.3.3, and now discuss how the bit-planes can be successively compressed.

The *Lossless Compression* block in Fig. 5.2, takes the bit-planes $\{\boldsymbol{b}_1^1, \boldsymbol{b}_2^1\}$ and applies an universal source coding technique to $\{\boldsymbol{b}_1^1, \boldsymbol{b}_2^1\}$. For maximal simplicity, we applied *gzip* program to compress the bit-planes $\{\boldsymbol{b}_1^1, \boldsymbol{b}_2^1\}$ for our simulations. These compressed sequences are then sent over a noiseless channel to both decoders.

In recent years, many people have successfully applied powerful channel codes (Turbo [100] and LDPC codes [64]) to compress the sources when highly correlated side information is available at the decoder [29]. Following these constructions, the *Syndrome Encoder 1* applies $r_1 - 1$ LDPC codes, one for each bit-plane, to compress $\boldsymbol{B}_1 \setminus \{\boldsymbol{b}_1^1\}$ and generates syndrome sequences $\boldsymbol{S}_1 = [\boldsymbol{s}_1^k : k \in \{2, \ldots, r_1\}]$ [29] which are then sent to both nodes 2,3. Similarly, the *Syndrome Encoder 2* applies $r_2 - 1$ LDPC codes to compress $\boldsymbol{B}_2 \setminus \{\boldsymbol{b}_2^1\}$ and generates syndrome sequences $\boldsymbol{S}_2 = [\boldsymbol{s}_2^k : k \in \{2, \ldots, r_2\}]$ which are then sent only to node 2. The rates (compression ratio) of the codes used for different bit-planes differ depending on the conditional entropies of the

bit planes. Normally, the compression increases as the bit-plane number increases.

At the decoders, first the *Lossless Decompresssion* blocks losslessly decode the compressed sequences of $\{\boldsymbol{b}_1^1, \boldsymbol{b}_2^1\}$. Then, the decoders decode the syndrome sequences sent by *Syndrome Encoder 1*. To do this, the *Initial Probability* blocks take $\{\boldsymbol{b}_1^1, \boldsymbol{b}_2^1\}$, the side information $\boldsymbol{Y(Z)}$ and previously decoded bit-planes $\hat{\boldsymbol{b}}_1^2, \ldots, \hat{\boldsymbol{b}}_1^{k-1}$ as input and calculate the probability $P(b_1^k|\tilde{y}(\tilde{z}), \tilde{c}_1, \tilde{c}_2, \hat{b}_1^2, \ldots, \hat{b}_1^{k-1})$ required to decode $\boldsymbol{s}_1^k$ at the *LDPC Decoder*, where $\tilde{x}$ denotes the realization of x. The *LDPC Decoder* successively decodes the syndrome sequences $\boldsymbol{S}_1$ by running a message-passing algorithm which uses the probability $P(b_1^k|\tilde{y}(\tilde{z}), \tilde{c}_1, \tilde{c}_2, \hat{b}_1^2, \ldots, \hat{b}_1^{k-1})$. The *LDPC Decoder* and the *Initial Probability* block communicate with each other every time a bit-plane is decoded. After decoding $\boldsymbol{S}_1$, the decoder at node 2 decodes $\boldsymbol{S}_2$ with the help of the *Initial Probability* block which provides $P(b_2^k|\tilde{y}, \tilde{c}_1, \tilde{c}_2, \hat{b}_1^2, \ldots, \hat{b}_1^{r_1}, \hat{b}_2^2, \ldots, \hat{b}_2^{k-1})$.

Once all the bit-planes are decoded, the *Trellis Read* modules at node 3 and 2 use the decoded bit-planes to reconstruct the quantized value in the first stage $\hat{Q}_1(\boldsymbol{x})$ and the quantized value in the refinement stage $\hat{Q}_2(\boldsymbol{x})$, respectively. Note that $\boldsymbol{b}_2^1$ is not used during the reconstruction of quantized value at node 3, because knowing the first bit of U alone is not enough to predict the best quantization point in Q_2. However, $\boldsymbol{b}_2^1$ can be used for compression purposes as in (5.7).

Finally, the *Estimate* blocks compute the MMSE estimates at node 3 and node 2 by using the conditional probabilities $p(t|\hat{Q}_1(\tilde{\boldsymbol{x}}), \tilde{z})$ and $p(t|\hat{Q}_2(\tilde{\boldsymbol{x}}), \tilde{y})$, respectively.

5.3.3 Computation of the Entropies

We now explain how the entropies and conditional distributions used in the code construction are computed. The entropy $H(\boldsymbol{b}_1^1, \boldsymbol{b}_2^1)$ is calculated by generating long sequences of the bit-planes $\{\boldsymbol{b}_1^1, \boldsymbol{b}_2^1\}$, compressing them using *gzip* and calculating the compression ratio achieved by *gzip* under the assumption that it achieves a com-

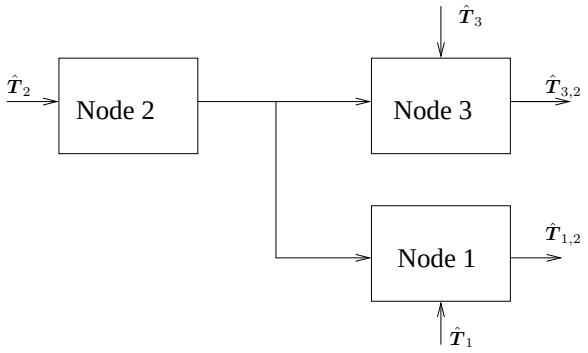


Figure 5.3: In the second round, node 2 encodes its estimate from round 1 as one common and one individual descriptions while nodes $1, 3$ use their estimates from round 1 as the side information.

pression ratio close to the entropy. For the bit-planes $\boldsymbol{b}_1^k,\ k \in \{2, \ldots, r_1\}$ and $\boldsymbol{b}_2^k,\ k \in \{2, \ldots, r_2\}$, the rates for the LDPC codes are selected by calculating the entropies $H(b_1^k|z, c_1, c_2, b_1^2, \ldots, b_1^{k-1})$ and $H(b_2^k|y, c_1, c_2, b_1^2, \ldots, b_1^{r_1}, b_2^2, \ldots, b_2^{k-1})$, respectively, where we assumed that the entropy is equal for all $n \in \{1, \ldots, N\}$ in (5.7) and (5.8).

It remains to explain, extending [29], how to compute the conditional entropies $H(b_2^k|y, c_1, c_2, b_1^2, \ldots, b_1^{r_1}, b_2^2, \ldots, b_2^{k-1})$ and the probabilities $P(b_2^k|\tilde{y}, \tilde{c}_1, \tilde{c}_2, \hat{b}_1^2, \ldots, \hat{b}_1^{r_1}, \hat{b}_2^2, \ldots, \hat{b}_2^{k-1})$. We use the same technique used in ([29]-equation (11)) to calculate the entropies, but we need to use the probabilty, $P(b_2^k|y, c_1, c_2, b_1^2, \ldots, b_1^{r_1}, b_2^2, \ldots, b_2^{k-1})$, in the calculation. The probabilities required at the LDPC decoder are also computed by using the technique used in ([29]-equation (12)) where we use the probability $P(b_2^k|\tilde{y}, \tilde{c}_1, \tilde{c}_2, \hat{b}_1^2, \ldots, \hat{b}_1^{r_1}, \hat{b}_2^2, \ldots, \hat{b}_2^{k-1})$.

5.4 Practical Source Coding Scheme for Multi Round

In this section, we present a practical coding scheme for our multiround collaborative estimation problem extending our code design for the single round collaborative

estimation in the previous section.

5.4.1 Practical Code Construction

In the first round ($k = 1$), we use the same code that we designed for the single round collaborative estimation. In the subsequent rounds ($k \geq 2$), the first question that arises is that what information should be used to encode and what information should be used as the side information. For the simplicity of the code design, we take the estimates from the previous round ($k-1$) as the observation/side information in the current round k as shown in Fig. 5.3.

$$X_k = \hat{T}_{1,k-1}, \quad Y_k = \hat{T}_{2,k-1}, \quad Z_k = \hat{T}_{3,k-1} \tag{5.9}$$

where $\hat{T}_{i,k-1}$ is the estimate of node i at round $k-1$. We take $\hat{T}_{i,k-1} = \hat{T}_{i,k-2}$ when $\mu(k-1) = i$ and $\hat{T}_{1,0} = X, \hat{T}_{2,0} = Y, \hat{T}_{3,0} = Z$.

We model the observations/side informations at round $k \geq 2$ as follows.

$$\hat{T}_{\lambda(k-1),k-1} = a_1\hat{T}_{\lambda(k-1),k-2} + a_2W_{k-1} + a_3U_{k-1} \tag{5.10}$$

$$\hat{T}_{\phi(k-1),k-1} = b_1\hat{T}_{\phi(k-1),k-2} + b_2W_{k-1} \tag{5.11}$$

where a_1, a_2, a_3, b_1, b_2 are scalars and W_{k-1}, U_{k-1} are Gaussian distributed according to

$$U_{k-1} = \alpha_{k-1}\hat{T}_{\mu(k-1),k-2} + Z_{1,k-1} \tag{5.12}$$

$$W_{k-1} = \beta_{k-1}\hat{T}_{\mu(k-1),k-2} + Z_{2,k-1} \tag{5.13}$$

where $\alpha_{k-1}, \beta_{k-1}$ are two scalars and $Z_{1,k-1} \sim \mathcal{N}(0, \sigma^2_{z_{1,k-1}}), Z_{2,k-1} \sim \mathcal{N}(0, \sigma^2_{z_{2,k-1}})$ are two Gaussian variables independent of $\hat{T}_{\mu(k-1),k-2}$ and of each other.

We use this model to do offline computations which include building the histogram of the quantization levels given the source and conditional entropies of the bit planes. We now explain how these offline computations are done. Suppose that at round k node 1 is supposed to transmit its observation. We generate the training sequences using the observation model in (5.9). We build a histogram of the quantization levels given the source using the training sequence. We then use this histogram to obtain an estimate of how much the common and individual descriptions should be compressed when the practical code is used online. To do this, we first determine which one is the best decoder and which one is the worst decoder. We call decoder 2 the "best decoder" if $h(X_k|Y_k) \leq h(X_k|Z_k)$, and the "worst decoder" otherwise.

$$\begin{aligned} h(X_k|Y_k) &= \frac{1}{2} log_2[2\pi e(1-\rho^2_{x_k y_k})\sigma^2_{x_k}] \\ h(X_k|Z_k) &= \frac{1}{2} log_2[2\pi e(1-\rho^2_{x_k z_k})\sigma^2_{x_k}] \end{aligned}$$

where σ^2_u is the variance of U and ρ_{uv} is the correlation coefficient of U, V. Hence,

$$h(X_k|Y_k) \leq h(X_k|Z_k) \text{ if } \rho^2_{x_k y_k} \geq \rho^2_{x_k z_k}$$

Suppose that decoder 2 is the "best decoder" and decoder 3 is the "worst decoder". Then, the bit planes of the common description is compressed to the conditional entropy

$$H(\boldsymbol{b}_1^2, \ldots, \boldsymbol{b}_1^{r_1} | \boldsymbol{Z}_k, \boldsymbol{b}_1^1, \boldsymbol{b}_2^1)$$

and the bit planes of the individual description is compressed to the conditional entropy

$$H(\boldsymbol{b}_2^2, \ldots, \boldsymbol{b}_2^{r_2} | \boldsymbol{Y}_k, \boldsymbol{B}_1, \boldsymbol{b}_2^1)$$

These offline computations are done for all K rounds before the practical code is

applied online.

When this practical code is applied online, the estimates from the previous round are used as the observation/side information in the current round without any assumption on the distribution of the estimates. At round k, the observations at the encoder are quantized using the codebooks used for offline compuations for round k and the descriptions are compressed to the conditional entropies computed offline.

5.4.2 Comparison of Practical Codes with the Theoretical Bounds

In this section, we simplify the theoretical bound to the quadratic Gaussian case so that we can compare the performance of our proposed practical coding scheme with the theoretical bounds. When we simplify the bound in (5.4), the number of variables that the pair (W_k, U_k) depends on increases quickly as k increases, which makes the simplification difficult. To avoid this difficulty, we use a technique where (W_k, U_k) depends only on the estimate $\hat{T}_{\mu(k),k-1}$ from the previous round, which is a function of the variables that (W_k, U_k) depends on in (5.4). We let the decoders also use the estimates $\hat{T}_{\lambda(k),k-1}, \hat{T}_{\phi(k),k-1}$ from the previous round as the side information. Since we will be using only a subset of the distributions in (5.4) by doing so, we will obtain a subset of the rate distortion region in (5.4).

When we apply this technique, the rate expression for round k becomes

$$R^k \geq I(\hat{T}_{\mu(k),k-1}; W_k|\hat{T}_{\phi(k),k-1}) + I(\hat{T}_{\mu(k),k-1}; U_k|W_k, \hat{T}_{\lambda(k),k-1}) \tag{5.14}$$

where $(W_k, U_k) \in \Phi'(\{R^k\}_{k=1}^K, D_1, D_2, D_3)$ and $\Phi'(\{R^k\}_{k=1}^K, D_1, D_2, D_3)$ is the set of random vectors $\{W_k, U_k\}_{k=1}^K$ that are jointly distributed with T, Y_1, Y_2, Y_3 such that the following conditions are satisfied.

1. $W_k, U_k \leftrightarrow \hat{T}_{\mu(k),k-1} \leftrightarrow T, \hat{T}_{\lambda(k),k-1}, \hat{T}_{\phi(k),k-1}$

2. There exist decoding functions $\bar{g}_i$ such that $E[d_i(T, \hat{T}_{i,K})] \leq D_i$ for $i = 1, 2, 3$.

We now evaluate the rate and distortion expressions for the quadratic Gaussian case. Note that to evaluate these expressions we should consider all possible distributions of (W_k, U_k) that satisfy the above conditions. Since this is almost impossible to do, we consider only Gaussian distributions for (W_k, U_k) which will give an upper bound to (5.14). To do this, let

$$U_k = \alpha_k \hat{T}_{\mu(k),k-1} + Z_{1,k} \tag{5.15}$$

$$W_k = \beta_k \hat{T}_{\mu(k),k-1} + Z_{2,k} \tag{5.16}$$

where α_k, β_k are two scalars and $Z_{1,k} \sim \mathcal{N}(0, \sigma^2_{z_{1,k}}), Z_{2,k} \sim \mathcal{N}(0, \sigma^2_{z_{2,k}})$ are two Gaussian variables independent of $\hat{T}_{\mu(k),k-1}$ and of each other. In the following evaluation, we model the correlation between the estimates using (5.10) and (5.11). We first

evaluate the rate used by the encoder at round k.

$$
\begin{aligned}
R^k &\geq I(\hat{T}_{\mu(k),k-1}; W_k|\hat{T}_{\phi(k),k-1}) + I(\hat{T}_{\mu(k),k-1}; U_k|W_k, \hat{T}_{\lambda(k),k-1}) \\
&= I(\hat{T}_{\mu(k),k-1}; W_k|\hat{T}_{\phi(k),k-1}) + I(\hat{T}_{\mu(k),k-1}; U_k, W_k, \hat{T}_{\lambda(k),k-1}) \\
&\quad -I(\hat{T}_{\mu(k),k-1}; \hat{T}_{\lambda(k),k-1}) - I(\hat{T}_{\mu(k),k-1}; W_k|\hat{T}_{\lambda(k),k-1}) \\
&= h(W_k|\hat{T}_{\phi(k),k-1}) - h(W_k|\hat{T}_{\mu(k),k-1}) + h(U_k, W_k, \hat{T}_{\lambda(k),k-1}) \\
&\quad -h(U_k, W_k, \hat{T}_{\lambda(k),k-1}|\hat{T}_{\mu(k),k-1}) - h(\hat{T}_{\mu(k),k-1}) + h(\hat{T}_{\mu(k),k-1}|\hat{T}_{\lambda(k),k-1}) \\
&\quad -h(W_k|\hat{T}_{\lambda(k),k-1}) + h(W_k|\hat{T}_{\mu(k),k-1}) \\
&= h(W_k|\hat{T}_{\phi(k),k-1}) + h(U_k, W_k, \hat{T}_{\lambda(k),k-1}) - h(U_k, W_k, \hat{T}_{\lambda(k),k-1}|\hat{T}_{\mu(k),k-1}) \\
&\quad -h(\hat{T}_{\mu(k),k-1}) + h(\hat{T}_{\mu(k),k-1}|\hat{T}_{\lambda(k),k-1}) - h(W_k|\hat{T}_{\lambda(k),k-1}) \\
&= \frac{1}{2}\log_2 \left\{ \frac{\left[1 + \frac{\beta_k^2}{\sigma_{z_{2,k}}^2}(1 - \rho^2_{\hat{t}_{\mu(k),k-1}\hat{t}_{\phi(k),k-1}})\sigma^2_{\hat{t}_{\mu(k),k-1}}\right]}{\left[1 + \frac{\beta_k^2}{\sigma_{z_{2,k}}^2}(1 - \rho^2_{\hat{t}_{\mu(k),k-1}\hat{t}_{\lambda(k),k-1}})\sigma^2_{\hat{t}_{\mu(k),k-1}}\right]} \right. \\
&\qquad \left. \left[1 + (\frac{\alpha_k^2}{\sigma_{z_{1,k}}^2} + \frac{\beta_k^2}{\sigma_{z_{2,k}}^2})(1 - \rho^2_{\hat{t}_{\mu(k),k-1}\hat{t}_{\lambda(k),k-1}})\sigma^2_{\hat{t}_{\mu(k),k-1}}\right] \right\} \qquad (5.17)
\end{aligned}
$$

We next evaluate the distortions $D^k_{\phi(k)}$ and $D^k_{\lambda(k)}$ at round k at nodes $\phi(k)$ and $\lambda(k)$, respectively.

$$
\begin{aligned}
& D^k_{\phi(k)} \\
&= \sigma^2_{t|w_k,\hat{t}_{\phi(k),k-1}} \\
&= \sigma_t^2 - \sigma_t^2 \frac{\rho^2_{t\hat{t}_{\phi(k),k-1}} + \frac{\beta_k^2}{\sigma_{z_{2,k}}^2}\sigma^2_{\hat{t}_{\mu(k),k-1}}(\rho^2_{t\hat{t}_{\mu(k),k-1}} + \rho^2_{t\hat{t}_{\phi(k),k-1}})}{1 + \frac{\beta_k^2}{\sigma_{z_{2,k}}^2}(1 - \rho^2_{\hat{t}_{\mu(k),k-1}\hat{t}_{\phi(k),k-1}})\sigma^2_{\hat{t}_{\mu(k),k-1}}} \\
&\quad + \sigma_t^2 \frac{2\frac{\beta_k^2}{\sigma_{z_{2,k}}^2}\sigma^2_{\hat{t}_{\mu(k),k-1}}\rho_{t\hat{t}_{\mu(k),k-1}}\rho_{\hat{t}_{\mu(k),k-1}\hat{t}_{\phi(k),k-1}}\rho_{t\hat{t}_{\phi(k),k-1}}}{1 + \frac{\beta_k^2}{\sigma_{z_{2,k}}^2}(1 - \rho^2_{\hat{t}_{\mu(k),k-1}\hat{t}_{\phi(k),k-1}})\sigma^2_{\hat{t}_{\mu(k),k-1}}} \qquad (5.18)
\end{aligned}
$$

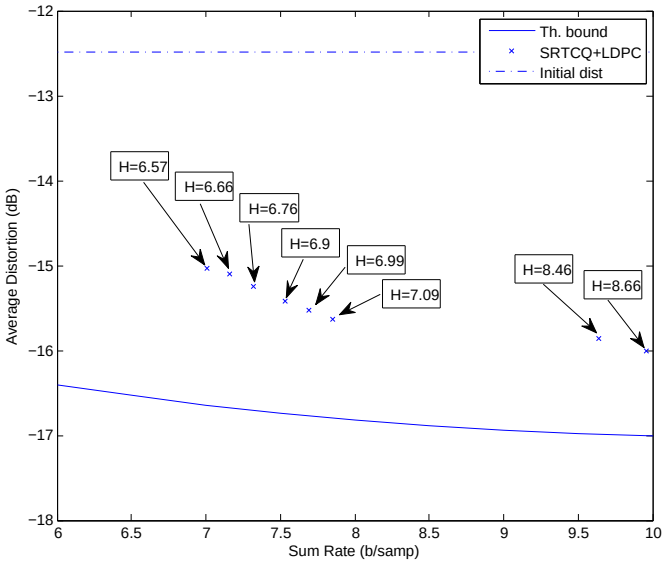
$$
\begin{aligned}
& D^k_{\lambda(k)} \\
= \;& \sigma^2_{t|u_k,w_k,\hat{t}_{\lambda(k),k-1}} \\
= \;& \sigma_t^2 - \sigma_t^2 \frac{\rho^2_{t\hat{t}_{\lambda(k),k-1}} + (\frac{\alpha_k^2}{\sigma^2_{z_{1,k}}} + \frac{\beta_k^2}{\sigma^2_{z_{2,k}}})\sigma^2_{\hat{t}_{\mu(k),k-1}}(\rho^2_{t\hat{t}_{\mu(k),k-1}} + \rho^2_{t\hat{t}_{\lambda(k),k-1}})}{1 + (\frac{\alpha_k^2}{\sigma^2_{z_{1,k}}} + \frac{\beta_k^2}{\sigma^2_{z_{2,k}}})(1 - \rho^2_{\hat{t}_{\mu(k),k-1}\hat{t}_{\lambda(k),k-1}})\sigma^2_{\hat{t}_{\mu(k),k-1}}} \\
& + \sigma_t^2 \frac{2(\frac{\alpha_k^2}{\sigma^2_{z_{1,k}}} + \frac{\beta_k^2}{\sigma^2_{z_{2,k}}})\sigma^2_{\hat{t}_{\mu(k),k-1}}\rho_{t\hat{t}_{\mu(k),k-1}}\rho_{\hat{t}_{\mu(k),k-1}\hat{t}_{\lambda(k),k-1}}\rho_{t\hat{t}_{\lambda(k),k-1}}}{1 + (\frac{\alpha_k^2}{\sigma^2_{z_{1,k}}} + \frac{\beta_k^2}{\sigma^2_{z_{2,k}}})(1 - \rho^2_{\hat{t}_{\mu(k),k-1}\hat{t}_{\lambda(k),k-1}})\sigma^2_{\hat{t}_{\mu(k),k-1}}}
\end{aligned}
\tag{5.19}
$$

In the next section, we use these expressions to plot the theoretical bounds.

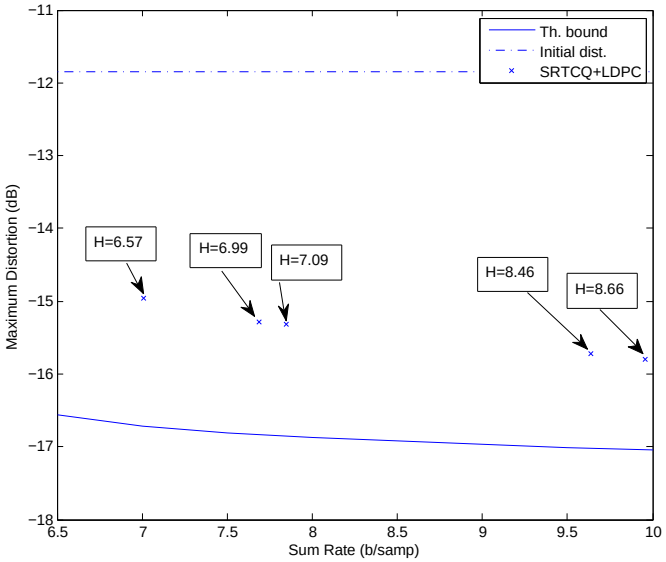
5.5 Experimental Results

We implemented our proposed practical source coding scheme for multi round collaborative estimation and tested it via simulation. We test our practical coding scheme for two different set of side information qualities. We let the nodes communicate with each other for $K = 3$ rounds and estimate the underlying source. The observation sequences are broken into blocks of length $N = 1000$ before SR-TCQ is applied to each block. In SR-TCQ, Ungerboeck's [60] 16-state trellis is used for both T_1 and T_2. We use uniform codebooks and distortion weighting factor $\alpha = 0.5$ in (5.6) to quantize the observation sequence at the encoder.

We use *gzip* program to losslessly compress the bit-planes $\{\boldsymbol{b}_1^1, \boldsymbol{b}_2^1\}$ and to compute the entropy $H(\boldsymbol{b}_1^1, \boldsymbol{b}_2^1)$ that is reported in this section. For the compression of the rest of the bit-planes, we apply LDPC codes of length 10^5 which means that we group together the SR-TCQ output of 100 blocks before we apply LDPC. The rates of the codes are selected according to the entropies calculated. In particular, for a bit-plane with entropy H_k we select the code rate such that the compression achieved by the



(a) Average distortion

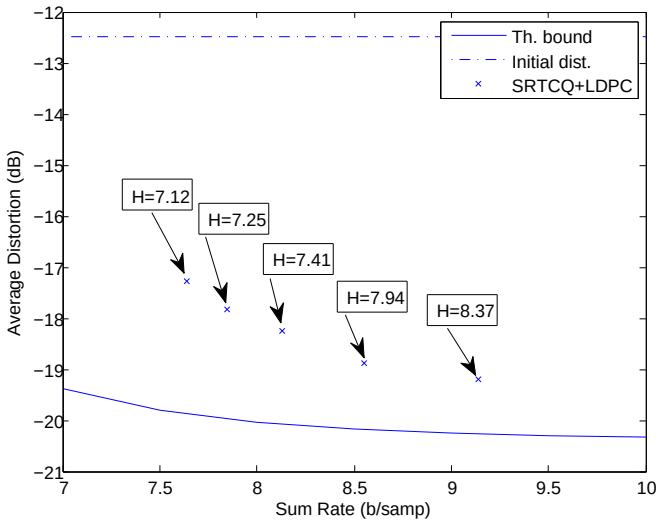


(b) Maximum distortion

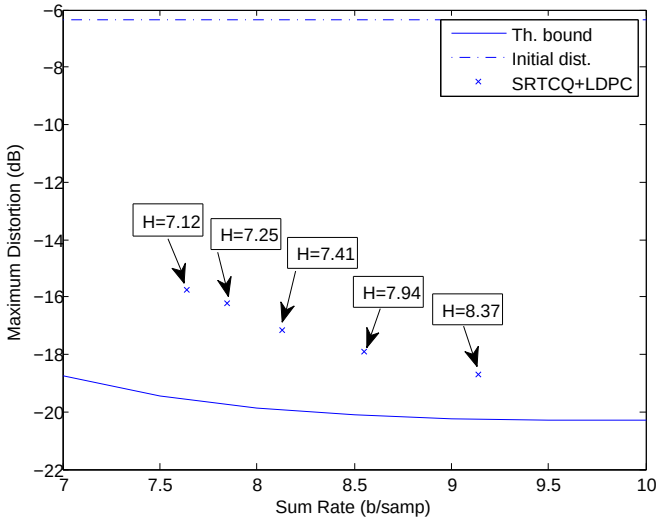
Figure 5.4: Comparison of the rate distortion points obtained using our practical code design with the theoretical bounds. The distribution of the source and observations are selected such that $\sigma_t^2 = 1$ and $\sigma_{n_1}^2 = 0.05, \sigma_{n_2}^2 = 0.06, \sigma_{n_3}^2 = 0.07$ in (5.1)-(5.3).

code is

$$r = \frac{H_k + \delta(1 + H_k)}{1 - \delta(1 + H_k)} \tag{5.20}$$



(a) Average distortion



(b) Maximum distortion

Figure 5.5: Comparison of the rate distortion points obtained using our practical code design with the theoretical bounds. The distribution of the source and observations are selected such that $\sigma_t^2 = 1$ and $\sigma_{n_1}^2 = 0.01, \sigma_{n_2}^2 = 0.15, \sigma_{n_3}^2 = 0.3$ in (5.1)-(5.3).

where δ is selected such that the bit error rate of the code is almost 0. We use a degree distribution of $\lambda(x) = x^2$ and $\rho(x) = (1 - \rho_m)x^{m-2} + \rho_m x^{m-1}$ for our LDPC codes [26], where $m = \lceil 3/H_k \rceil$.

We make two kinds of plots: 'Average distortion Vs Sum rate' and 'Maximum distortion Vs Sum rate'. To plot the theoretical bound for 'Average distortion Vs Sum rate', we formulate the following optimization and solve it numerically.

$$\min_{\left\{\alpha_k,\beta_k,\sigma^2_{z_{1,k}},\sigma^2_{z_{2,k}}\right\}_{k=1}^{K}} (D_1^K + D_2^K + D_3^K)/3$$
$$\text{subject to } \sum_{k=1}^{K} R^k \leq R$$

where $D_i^K = D_i,\ i = 1, 2, 3$ in (5.4). To plot the theoretical bound for 'Maximum distortion Vs Sum rate', we solve the following optimization problem.

$$\min_{\left\{\alpha_k,\beta_k,\sigma^2_{z_{1,k}},\sigma^2_{z_{2,k}}\right\}_{k=1}^{K}} \max(D_1^K, D_2^K, D_3^K)$$
$$\text{subject to } \sum_{k=1}^{K} R^k \leq R$$

where $D_i^K = D_i,\ i = 1, 2, 3$ in (5.4).

Fig. 5.4 compares our practical coding scheme with the theoretical bound when the side information qualities are roughly equal, where we selected the noise of the side information such that $\sigma^2_{n_1} = 0.05, \sigma^2_{n_2} = 0.06, \sigma^2_{n_3} = 0.07$ in (5.1)-(5.3). Fig. 5.4(a) shows the average final distortions of the nodes and Fig. 5.4(b) shows the maximum distortions after 3 rounds for different sum of the rates used at each round. In the plots, H denotes the entropy of all bit-planes of each sample ($b/sample$).

If we assume ideal compression/decompression of the bit-planes, the gap between our scheme and the theoretical bound is 0.89 dB, 1.2 dB, 1.51 dB in the average distortion case and 1.13 dB, 1.41 dB, 1.62 dB in the maximum distortion case for the average rates per round of 2.89 b/s, 2.36 b/s, 2.19 b/s, respectively. When we apply LDPC codes for compression of the bit-planes, the gap is 1.0 dB, 1.16 dB, 1.6 dB in

the average distortion case and 1.23 dB, 1.5 dB, 1.76 dB in the maximum distortion case for the average rates per round of 3.32 b/s, 2.62 b/s, 2.33 b/s, respectively. As one can easily see the gap between our practical coding scheme and the theoretical bound decreases as the rate increases.

Fig. 5.5 compares our practical coding scheme with the theoretical bound when the side informations have different qualities where we selected the noise of the side information such that $\sigma^2_{n_1} = 0.01, \sigma^2_{n_2} = 0.15, \sigma^2_{n_3} = 0.3$ in (5.1)-(5.3). If we assume ideal compression/decompression of the bit-planes, the gap between our scheme and the theoretical bound is 2.2 dB, 1.48 dB, 0.96 dB in the average distortion case and 3.16 dB, 2.18 dB, 1.8 dB in the maximum distortion case for the average rates per round of 2.37 b/s, 2.47 b/s, 2.79 b/s, respectively.

Note that when the side informations are of roughly equal quality, then it is beneficial use equal rate at each round. However when the side informations are of different qualities, it is beneficial to allocate more rate to the node with best quality.

5.6 Extension of Practical Coding Scheme to M-node Network

In this section, we discuss how the practical coding scheme that we described for a 3-node network can be extended to a M-node network. To do this, let $\mu(k)$ be the encoder and $\lambda_i(k)$ be the ith ($i = 1, \ldots, M-1$) best decoder at round k. Then, any information sent to $\lambda_j(k)$ can be decoded by $\lambda_i(k)$ if $i \leq j$. At round k, node $\mu(k)$ encodes $M-1$ descriptions, one for each set of decoders $\{\lambda_1(k), \ldots, \lambda_i(k)\}$, $i = 1, \ldots, M-1$.

To generate $M-1$ successively refinable descriptions, we need to use $M-1$ refinement stage SR-TCQ. Suppose that we use the rates $r_1, \ldots, r_{M-1}$ at each stage. Then, we will have $M-1$ set of quantization points $Q_1, \ldots, Q_{M-1}$ where Q_ℓ has $2^{r_\ell+1}$ quantization points for each quantization point in $Q_{\ell-1}$, $\ell = 2, \ldots, M-1$. The trellis

for $(M-1)$-stage SR-TCQ is constructed by taking the tensor product of the trellis at each stage $T_1, \ldots, T_{M-1}$ [61]. Then, the observation sequence is quantized using the Viterbi algorithm such that the weighted distortion D of all stages is minimized.

$$D = \sum_{i=1}^{M-1} \alpha_i D_i, \quad \sum_{i=1}^{M-1} \alpha_i = 1$$

After generating $M-1$ descriptions, the bit-planes $\boldsymbol{b}_1^1, \ldots, \boldsymbol{b}_{M-1}^1$ that represent the branches of the trellis are compressed using a universal lossless source coding technique. The rest of the bit-planes are compressed by sending the syndromes of LDPC codes. For example, the $j(2 \leq j \leq r_i)$th bit-plane of the $i(1 \leq i \leq M-1)$th description $\boldsymbol{b}_i^j$ is compressed to the following conditional entropy using the LDPC codes.

$$H(\boldsymbol{b}_i^j |\ \boldsymbol{Y}_{\lambda_{M-i}(k)}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_{i-1}, \boldsymbol{b}_i^1, \ldots, \boldsymbol{b}_i^{j-1})$$

The decoding of the syndromes is done as described in Section 5.3.2.

5.7 Conclusions

We proposed a low-complexity/low-communication algorithm design based on SR-TCQ and LDPC codes for multiround collaborative estimation when the source and observations are Gaussian distributed and the distortion measure is squared error. In this code design at each round we generated the descriptions using $M-1$ refinement level SR-TCQ, compressed the bit-planes that represent the branches of the trellis using standard universal lossless compression, while compressing the rest of the bit-planes using LDPC codes for decoding with side information. We also derived a theoretical bound for this collaborative estimation problem. We showed the gap between our practical coding scheme and the theoretical bound for a 3-node network is only 1.0 dB (average final distortion) for average rate per round of 3.32 b/s.

The simulation shows that the gap between our scheme and the theoretical bound decreases with increasing rate.

6. Conclusions

In this dissertation we studied a collaborative estimation problem in which a network of nodes, each indirectly observing an underlying source through "noisy" measurements, communicate with each other in order to form better estimates of the underlying source. We argued that any collaborative estimation algorithm applied in practice should consider two primary constraints: complexity and communication. Furthermore, we demonstrated that signal processing and machine learning tools can be used to develop low complexity algorithms with less attention to communication, while information theory can be used to develop algorithms that can efficiently trade communication for performance with less attention to complexity. It was also made clear that the modern practical coding theory provides tools to design low-complexity/low-communication algorithms by melding ideas from machine learning, signal processing, and information theory. In this dissertation, we developed collaborative estimation algorithms for each of these classes by applying tools from the respective area of study and evaluated the estimation error performances of those algorithms.

In Chapter 3, we derived a low-complexity algorithm for collaborative estimation of channel gains in wireless sensor networks via expectation propagation based approximate Bayesian inference. This algorithm makes effective use of the prior distribution of the channel gains along with the information obtained from channel training and iteratively passes messages between the nodes to obtain MMSE estimates at each node. The estimation error performance of our algorithm was evaluated by simulations and was shown to perform better than another distributed estimation algorithm, diffusion LMS, as applied to the same estimation problem. The computational complexity, message passing overhead and memory requirements were also presented for

both algorithms.

In Chapter 4, we applied tools from multiterminal information theory to study theoretical limits on communication and estimation error performances. We proposed a source code architecture for collaborative estimation problem and argued that it is the proper model that exploits the capabilities of network/channel codes. We derived an achievable rate distortion region by hybridizing techniques from the CEO and multiple descriptions problems. We showed that this achievable rate distortion region simplifies to some known bounds for some simpler problems. We also derived an outer bound from the basic principles of information theory.

In Chapter 5, we designed a low-complexity/low-communication collaborative estimation algorithm by utilizing the tools from modern practical coding theory. This algorithm was designed for a particular communication protocol in which nodes communicate over several rounds and exactly one node is allowed to multicast messages each round which contain successively refinable descriptions. This algorithm was designed by generating the successively refinable descriptions using SR-TCQ and compressing them with syndromes of the LDPC codes. This algorithm was tested via simulations and it was shown that it attains average estimation error performances within 1dB away from the theoretical bounds that we derived for this problem.

Appendix A. Proof of Theorem 1

We begin the proof by listing the possible errors.

1. E_0: $\left(\boldsymbol{T}, \boldsymbol{Y}_{[M]}\right) \notin A_\epsilon^*(T, Y_{[M]})$
2. E_1: At least at one (encoder) node $j \in [M]$, $\left(\boldsymbol{U}_{\mathcal{S}_j}(\boldsymbol{m}_{\mathcal{S}_j}), \boldsymbol{Y}_j\right) \notin A_\epsilon^*\left(U_{\mathcal{S}_j}, Y_j\right)$ for all $\boldsymbol{m}_{\mathcal{S}_j} \in \prod_{(j\to\mathcal{A})\in\mathcal{S}_j}\left[2^{N\tilde{R}_{j\to\mathcal{A}}}\right]$.
3. E_2: At least at one (decoder) node $i \in [M]$, $\left(\boldsymbol{U}_{\mathcal{D}_i}(\boldsymbol{m}_{\mathcal{D}_i}), \boldsymbol{Y}_{\mathcal{E}(\mathcal{D}_i)}, \boldsymbol{Y}_i, \boldsymbol{T}\right) \notin A_\epsilon^*\left(U_{\mathcal{D}_i}, Y_{\mathcal{E}(\mathcal{D}_i)}, Y_i, T\right)$ where $\boldsymbol{m}_{\mathcal{D}_i}$ is defined as follows

$$\boldsymbol{m}_{\mathcal{D}_i} := (m_{j\to\mathcal{A}})_{(j\to\mathcal{A})\in\mathcal{D}_i}$$

where $m_{j\to\mathcal{A}}$ is the index of the codeword selected by the encoding function $\boldsymbol{f}_{j\to\mathcal{A}}$.

4. E_3: At least at one (decoder) node $i \in [M]$, $\exists\, \boldsymbol{U}_{\mathcal{D}_i}(\boldsymbol{m}_{\mathcal{D}_i})$ such that $\left(\boldsymbol{U}_{\mathcal{D}_i}(\boldsymbol{m}_{\mathcal{D}_i}), \boldsymbol{Y}_{\mathcal{E}(\mathcal{D}_i)}, \boldsymbol{Y}_i, \boldsymbol{T}\right) \in A_\epsilon^*(U_{\mathcal{D}_i}, Y_{\mathcal{E}(\mathcal{D}_i)}, Y_i, T)$, but $\boldsymbol{U}_{\mathcal{D}_i}(\boldsymbol{\ell}_{j\to\mathcal{A}}) \neq \boldsymbol{U}_{\mathcal{D}_i}(\boldsymbol{m}_{j\to\mathcal{A}})$ where $\boldsymbol{\ell}_{\mathcal{D}_i}$ is defined as follows

$$\boldsymbol{\ell}_{\mathcal{D}_i} := (\ell_{j\to\mathcal{A}})_{(j\to\mathcal{A})\in\mathcal{D}_i}$$

where $\ell_{j\to\mathcal{A}}$ is the index of the codeword selected at the decoder i such that $\boldsymbol{U}_{j\to\mathcal{A}}(\ell_{j\to\mathcal{A}}) \in B_{j\to\mathcal{A}}(b_{j\to\mathcal{A}})$.

Define the coding error E as $E := \cup_{i=0}^{3} E_i$. Then the probability of error $Pr(E)$ is bounded above by

$$Pr(E) \leq Pr(E_0) + \sum_{i=1}^{3} Pr(E_i \cap E_0^c)$$

We now show that the probabilities of these errors are small.

1. Clearly $Pr(E_0) \to 0$ as $N \to \infty$.

2. To prove $Pr(E_1 \cap E_0^c)$ is small, for each $\boldsymbol{y}_j \in A_\epsilon^*(Y_j)$, $j \in [M]$, define the random set $G_{\mathcal{S}_j}(\boldsymbol{y}_j)$ as

$$G_{\mathcal{S}_j}(\boldsymbol{y}_j) := \left\{ \boldsymbol{m}_{\mathcal{S}_j} \;\middle|\; \boldsymbol{U}_{\mathcal{S}_j}(\boldsymbol{m}_{\mathcal{S}_j}) \in A_\epsilon^*(U_{\mathcal{S}_j}|\boldsymbol{y}_j) \right\}$$

For each set $\mathcal{S}_j$, define the event $E_{1,\mathcal{S}_j}$ as

$$E_{1,\mathcal{S}_j} := \{(\boldsymbol{U}_{\mathcal{S}_j}(\boldsymbol{m}_{\mathcal{S}_j}), \boldsymbol{Y}_j) \notin A_\epsilon^*(U_{\mathcal{S}_j}, Y_j) \quad \text{for all} \quad \boldsymbol{m}_{\mathcal{S}_j} \in \prod_{(j \to \mathcal{A}) \in \mathcal{S}_j} [2^{N\tilde{R}_{j\to\mathcal{A}}}]\}$$

Then, we have

$$\begin{aligned} Pr\left[E_{1,\mathcal{S}_j} \text{ and } \boldsymbol{Y}_j \in A_\epsilon^*(Y_j)\right] &= Pr\left[|G_{\mathcal{S}_j}(\boldsymbol{Y}_j)| = 0 \text{ and } \boldsymbol{Y}_j \in A_\epsilon^*(Y_j)\right] \\ &\leq \max_{\boldsymbol{y}_j \in A_\epsilon^*(Y_j)} Pr\left[|G_{\mathcal{S}_j}(\boldsymbol{y}_j)| = 0\right] \end{aligned}$$

Using Chebyshev's inequality [101] [91], for all $y_j^N \in A_\epsilon^*(Y_j)$ and $0 < \alpha < 1$, we write

$$\begin{aligned} Pr\left[|G_{\mathcal{S}_j}(\boldsymbol{y}_j)| = 0\right] &\leq Pr\left[\left||G_{\mathcal{S}_j}(\boldsymbol{y}_j)| - E\left[|G_{\mathcal{S}_j}(\boldsymbol{y}_j)|\right]\right| \geq \alpha E\left[|G_{\mathcal{S}_j}(\boldsymbol{y}_j)|\right]\right] \\ &\leq \frac{\text{Var}\left[|G_{\mathcal{S}_j}(\boldsymbol{y}_j)|\right]}{\alpha^2 \left(E\left[|G_{\mathcal{S}_j}(\boldsymbol{y}_j)|\right]\right)^2} \end{aligned}$$

We now bound $E\left[|G_{\mathcal{S}_j}(\boldsymbol{y}_j)|\right]$. Define the indicator function

$$1\left(\boldsymbol{U}_{\mathcal{S}_j}(\boldsymbol{m}_{\mathcal{S}_j}) \in G_{\mathcal{S}_j}(\boldsymbol{y}_j)\right) = \begin{cases} 1 & \text{if } \boldsymbol{U}_{\mathcal{S}_j}(\boldsymbol{m}_{\mathcal{S}_j}) \in G_{\mathcal{S}_j}(\boldsymbol{y}_j) \\ 0 & \text{otherwise} \end{cases}$$

Then the cardinality of the set $G_{\mathcal{S}_j}(\boldsymbol{y}_j)$ is given by

$$\left|G_{\mathcal{S}_j}(\boldsymbol{y}_j)\right| = \sum_{\boldsymbol{m}_{\mathcal{S}_j} \in [2^{N\tilde{R}_{\mathcal{S}_j}}]} 1\left(\boldsymbol{U}_{\mathcal{S}_j}(\boldsymbol{m}_{\mathcal{S}_j}) \in G_{\mathcal{S}_j}(\boldsymbol{y}_j)\right)$$

where $\left[2^{N\tilde{R}_{\mathcal{S}_j}}\right]$ denotes the Cartesian product of the sets $\left[2^{N\tilde{R}_{j\to\mathcal{A}}}\right], (j \to \mathcal{A}) \in \mathcal{S}_j$. Since $E\left[1\left(\boldsymbol{U}_{\mathcal{S}_j} \in G_{\mathcal{S}_j}(\boldsymbol{y}_j)\right)\right] \geq 2^{-N\left[\sum_{(j\to\mathcal{A})\in\mathcal{S}_j} H(U_{j\to\mathcal{A}}) - H(U_{\mathcal{S}_j}|Y_j) + \epsilon_1\right]}$ where $\epsilon_1 \to 0$ as $\epsilon \to 0$, we have

$$\begin{aligned} E\left[|G_{\mathcal{S}_j}(\boldsymbol{y}_j)|\right] &\geq \left|\left[2^{N\tilde{R}_{\mathcal{S}_j}}\right]\right| E\left[1\left(\boldsymbol{U}_{\mathcal{S}_j}(\boldsymbol{m}_{\mathcal{S}_j}) \in G_{\mathcal{S}_j}(\boldsymbol{y}_j)\right)\right] \\ &\geq 2^{N\left[\sum_{(j\to\mathcal{A})\in\mathcal{S}_j}\left(\tilde{R}_{j\to\mathcal{A}} - H(U_{j\to\mathcal{A}})\right) + H(U_{\mathcal{S}_j}|Y_j) - \epsilon_1\right]} \end{aligned}$$

We next bound $\mathrm{Var}\left[|G_{\mathcal{S}_j}(\boldsymbol{y}_j)|\right]$. Consider

$$\begin{aligned} & \mathrm{Var}\left[|G_{\mathcal{S}_j}(\boldsymbol{y}_j)|\right] \\ \leq\ & E\left[\left|G_{\mathcal{S}_j}(\boldsymbol{y}_j)\right|^2\right] \\ =\ & E\Bigg[\Bigg(\sum_{\boldsymbol{m}_{\mathcal{S}_j} \in [2^{N\tilde{R}_{\mathcal{S}_j}}]} \sum_{\boldsymbol{m}'_{\mathcal{S}_j} \in [2^{N\tilde{R}_{\mathcal{S}_j}}]} 1\left(\boldsymbol{U}_{\mathcal{S}_j}(\boldsymbol{m}_{\mathcal{S}_j}) \in G_{\mathcal{S}_j}(\boldsymbol{y}_j),\right. \\ & \qquad\qquad \left.\boldsymbol{U}_{\mathcal{S}_j}(\boldsymbol{m}'_{\mathcal{S}_j}) \in G_{\mathcal{S}_j}(\boldsymbol{y}_j)\right)\Bigg)\Bigg] \\ =\ & \sum_{\boldsymbol{m}_{\mathcal{S}_j}} \sum_{\boldsymbol{m}'_{\mathcal{S}_j}} Pr\left[\boldsymbol{U}_{\mathcal{S}_j}(\boldsymbol{m}'_{\mathcal{S}_j}) \in A^*_\epsilon(U_{\mathcal{S}_j}|\boldsymbol{y}_j) \,\middle|\, \boldsymbol{U}_{\mathcal{S}_j}(\boldsymbol{m}_{\mathcal{S}_j}) \in A^*_\epsilon(U_{\mathcal{S}_j}|\boldsymbol{y}_j)\right] \\ & \qquad\qquad Pr\left[\boldsymbol{U}_{\mathcal{S}_j}(\boldsymbol{m}_{\mathcal{S}_j}) \in A^*_\epsilon(U_{\mathcal{S}_j}|\boldsymbol{y}_j)\right] \\ =\ & \sum_{\boldsymbol{m}_{\mathcal{S}_j}} \left(1 + \sum_{\boldsymbol{m}_{\mathcal{S}_j} \neq \boldsymbol{m}'_{\mathcal{S}_j}} Pr\left[\boldsymbol{U}_{\mathcal{S}_j}(\boldsymbol{m}'_{\mathcal{S}_j}) \in A^*_\epsilon(U_{\mathcal{S}_j}|\boldsymbol{y}_j) \,\middle|\, \boldsymbol{U}_{\mathcal{S}_j}(\boldsymbol{m}_{\mathcal{S}_j}) \in A^*_\epsilon(U_{\mathcal{S}_j}|\boldsymbol{y}_j)\right]\right) \\ & \qquad\qquad Pr\left[\boldsymbol{U}_{\mathcal{S}_j}(\boldsymbol{m}_{\mathcal{S}_j}) \in A^*_\epsilon(U_{\mathcal{S}_j}|\boldsymbol{y}_j)\right] \end{aligned}$$

$$
\begin{aligned}
&= \sum_{\boldsymbol{m}_{\mathcal{S}_j}} \Bigg(1 + \sum_{\mathcal{P}_j \subset \mathcal{S}_j} \sum_{\boldsymbol{m}'_{\bar{\mathcal{P}}_j} | m_{j\to\mathcal{A}} \neq m'_{j\to\mathcal{A}}} \sum_{\forall (j\to\mathcal{A}) \in \bar{\mathcal{P}}_j} Pr\Big[\Big(\boldsymbol{U}_{\mathcal{P}_j}(\boldsymbol{m}'_{\mathcal{P}_j}), \boldsymbol{U}_{\bar{\mathcal{P}}_j}(\boldsymbol{m}'_{\bar{\mathcal{P}}_j}) \Big) \\
&\qquad \in A^*_\epsilon(U_{\mathcal{S}_j}|\boldsymbol{y}_j) \,\Big|\, \big(\boldsymbol{U}_{\mathcal{P}_j}(\boldsymbol{m}_{\mathcal{P}_j}), \boldsymbol{U}_{\bar{\mathcal{P}}_j}(\boldsymbol{m}_{\bar{\mathcal{P}}_j}) \big) \in A^*_\epsilon(U_{\mathcal{S}_j}|\boldsymbol{y}_j) \Big] \Bigg) \\
&\qquad Pr\left[\boldsymbol{U}_{\mathcal{S}_j}(\boldsymbol{m}_{\mathcal{S}_j}) \in A^*_\epsilon(U_{\mathcal{S}_j}|\boldsymbol{y}_j) \right] \\
&\leq \sum_{\boldsymbol{m}_{\mathcal{S}_j}} \left(1 + \sum_{\mathcal{P}_j \subset \mathcal{S}_j} 2^{N\left[\sum_{(j\to\mathcal{A})\in\bar{\mathcal{P}}_j} (\tilde{R}_{j\to\mathcal{A}} - H(U_{j\to\mathcal{A}})) + H(U_{\bar{\mathcal{P}}_j}|U_{\mathcal{P}_j}, Y_j) + \epsilon_2 \right]} \right) \\
&\qquad Pr\left[\boldsymbol{U}_{\mathcal{S}_j}(\boldsymbol{m}_{\mathcal{S}_j}) \in A^*_\epsilon(U_{\mathcal{S}_j}|\boldsymbol{y}_j) \right] \\
&= \left(1 + \sum_{\mathcal{P}_j \subset \mathcal{S}_j} 2^{N\left[\sum_{(j\to\mathcal{A})\in\bar{\mathcal{P}}_j} (\tilde{R}_{j\to\mathcal{A}} - H(U_{j\to\mathcal{A}})) + H(U_{\bar{\mathcal{P}}_j}|U_{\mathcal{P}_j}, Y_j) + \epsilon_2 \right]} \right) E\left[\left| G_{\mathcal{S}_j}(\boldsymbol{y}_j) \right| \right]
\end{aligned}
$$

Here $\bar{\mathcal{P}}_j = \mathcal{S}_j \setminus \mathcal{P}_j$ and $\epsilon_2 \to 0$ as $\epsilon \to 0$. Hence, we have

$$
\begin{aligned}
Pr\left[\left| G_{\mathcal{S}_j}(\boldsymbol{y}_j) \right| = 0 \right] \;\leq\; & 2^{-N\left[\sum_{(j\to\mathcal{A})\in\mathcal{S}_j} (\tilde{R}_{j\to\mathcal{A}} - H(U_{j\to\mathcal{A}})) + H(U_{\mathcal{S}_j}|Y_j) - \epsilon_1 \right]} \\
& + \sum_{\mathcal{P}_j \subset \mathcal{S}_j} 2^{-N\left[\sum_{(j\to\mathcal{A})\in\mathcal{P}_j} (\tilde{R}_{j\to\mathcal{A}} - H(U_{j\to\mathcal{A}})) + H(U_{\mathcal{P}_j}|Y_j) - \epsilon_1 - \epsilon_2 \right]}
\end{aligned}
$$

Note that $Pr(E_1 \cap E_0^c) \leq \sum_{j\in[M]} Pr(E_{1,\mathcal{S}_j} \cap E_0^c)$. Thus, $Pr(E_1 \cap E_0^c)$ can be made arbitrary small by selecting the rates $\tilde{R}_{j\to\mathcal{A}}$, $(j \to \mathcal{A}) \in \mathcal{S}_j$ such that for each subset $\mathcal{P}_j \subseteq \mathcal{S}_j$ the following rate condition is satisfied

$$
\sum_{(j\to\mathcal{A})\in\mathcal{P}_j} \tilde{R}_{j\to\mathcal{A}} > \sum_{(j\to\mathcal{A})\in\mathcal{P}_j} H(U_{j\to\mathcal{A}}) - H(U_{\mathcal{P}_j}|Y_j) + \epsilon_1 + \epsilon_2 \tag{A.1}
$$

3. When the rate conditions in (A.1) are satisfied at each encoder $j \in [M]$, by Lemma 1 (see Appendix) $Pr(E_2 \cap E_0^c) \to 0$ as $N \to \infty$.

4. To prove $Pr(E_3 \cap E_0^c)$ is small, for each $i \in [M]$ define $E_{3,\mathcal{D}_i}$ as

$$\begin{aligned} E_{3,\mathcal{D}_i} \;=\; & \Big\{ \left(\boldsymbol{U}_{\mathcal{D}_i}(\boldsymbol{m}_{\mathcal{D}_i}), \boldsymbol{Y}_{\mathcal{E}(\mathcal{D}_i)}, \boldsymbol{Y}_i, \boldsymbol{T}\right) \in A_\epsilon^* \left(U_{\mathcal{D}_i}, Y_{\mathcal{E}(\mathcal{D}_i)}, Y_i, T\right) \quad \text{and} \\ & (\boldsymbol{U}_{\mathcal{D}_i}(\boldsymbol{\ell}_{\mathcal{D}_i}), \boldsymbol{Y}_i) \in A_\epsilon^* \left(U_{\mathcal{D}_i}, Y_i\right) \text{ for some } \quad \boldsymbol{U}_{\mathcal{D}_i}(\boldsymbol{\ell}_{\mathcal{D}_i}) \in \mathcal{B}_{\mathcal{D}_i}(b_{\mathcal{D}_i}) \\ & \text{and} \quad \boldsymbol{U}_{\mathcal{D}_i}(\boldsymbol{m}_{\mathcal{D}_i}) \neq \boldsymbol{U}_{\mathcal{D}_i}(\boldsymbol{\ell}_{\mathcal{D}_i}) \Big\} \end{aligned}$$

Note that $\boldsymbol{\ell}_{\mathcal{D}_i} \neq \boldsymbol{m}_{\mathcal{D}_i}$ if $\ell_{j\to\mathcal{A}} \neq m_{j\to\mathcal{A}}$ for at least one $(j \to \mathcal{A}) \in \mathcal{D}_i$. Define the event $E'_{3,\mathcal{D}_i}$ as

$$\begin{aligned} E'_{3,\mathcal{D}_i} \;:=\; & \Big\{ (\boldsymbol{U}_{\mathcal{D}_i}(\boldsymbol{\ell}_{\mathcal{D}_i}), \boldsymbol{Y}_i) \in A_\epsilon^* \left(U_{\mathcal{D}_i}, Y_i\right) \quad \text{for some} \quad \boldsymbol{U}_{\mathcal{D}_i}(\boldsymbol{\ell}_{\mathcal{D}_i}) \in \mathcal{B}_{\mathcal{D}_i}(b_{\mathcal{D}_i}) \\ & \text{and } \boldsymbol{U}_{\mathcal{D}_i}(\boldsymbol{m}_{\mathcal{D}_i}) \neq \boldsymbol{U}_{\mathcal{D}_i}(\boldsymbol{\ell}_{\mathcal{D}_i}) \Big\} \end{aligned}$$

Then

$$Pr(E_{3,\mathcal{D}_i}) \leq Pr\left[E'_{3,\mathcal{D}_i} \mid \left(\boldsymbol{U}_{\mathcal{D}_i}(\boldsymbol{m}_{\mathcal{D}_i}), \boldsymbol{Y}_i\right) \in A_\epsilon^*(U_{\mathcal{D}_i}, Y_i)\right] = P_e$$

Let the set $\mathcal{C}_i$ be

$$\mathcal{C}_i := \{(j \to \mathcal{A}) \in \mathcal{D}_i | \ell_{j\to\mathcal{A}} \neq m_{j\to\mathcal{A}}\}$$

For a particular $\boldsymbol{\ell}_{\mathcal{D}_i} = \boldsymbol{\ell}'_{\mathcal{D}_i} (\neq \boldsymbol{m}_{\mathcal{D}_i})$, define $E_{3,\mathcal{C}_i}(\boldsymbol{\ell}'_{\mathcal{C}_i})$ as

$$\begin{aligned} E_{3,\mathcal{C}_i}(\boldsymbol{\ell}'_{\mathcal{C}_i}) \;:=\; & \Big\{ \left(\boldsymbol{U}_{\mathcal{D}_i}(\boldsymbol{\ell}'_{\mathcal{D}_i}), \boldsymbol{Y}_i\right) \in A_\epsilon^* \left(U_{\mathcal{D}_i}, Y_i\right) \quad \text{such that} \quad \ell'_{j\to\mathcal{A}} \neq m_{j\to\mathcal{A}} \\ & \forall (j \to \mathcal{A}) \in \mathcal{C}_i \text{ and } \quad \ell'_{j\to\mathcal{A}} = m_{j\to\mathcal{A}} \, \forall (j \to \mathcal{A}) \in \mathcal{D}_i \setminus \mathcal{C}_i \Big\} \end{aligned}$$

Consider the code corresponding to a particular $(j \to \mathcal{A}) \in \mathcal{C}_i$. There are $(|\mathcal{B}_{j\to\mathcal{A}}| - 1)$ sequences in the bin $\mathcal{B}_{j\to\mathcal{A}}(b_{j\to\mathcal{A}})$ such that $\ell_{j\to\mathcal{A}} \neq m_{j\to\mathcal{A}}$, where $|\mathcal{B}_{j\to\mathcal{A}}|$ is the size of the bin. Thus for a particular $\mathcal{C}_i$, the number of possible

events in which $\boldsymbol{\ell}_{j\to\mathcal{A}} \neq \boldsymbol{m}_{j\to\mathcal{A}}$ (the number of events $E_{3,\mathcal{C}_i}(\boldsymbol{\ell}'_{\mathcal{C}_i})$) is $\prod_{(j\to\mathcal{A})\in\mathcal{C}_i}(|\mathcal{B}_{j\to\mathcal{A}}|-1)$. Note that the probabilities of these events are equal. Thus the probability of error P_e is bounded by

$$P_e \leq \sum_{\mathcal{C}_i\subseteq\mathcal{D}_i}\left(\prod_{(j\to\mathcal{A})\in\mathcal{C}_i}(|\mathcal{B}_{j\to\mathcal{A}}|-1)\right) Pr\left[E_{3,\mathcal{C}_i}(\boldsymbol{\ell}'_{\mathcal{C}_i}) \mid \left(\boldsymbol{U}_{\mathcal{D}_i}(\boldsymbol{m}_{\mathcal{D}_i}), \boldsymbol{Y}_i\right) \in A^*_\epsilon(U_{\mathcal{D}_i}, Y_i)\right]$$

We have the following

$$\begin{aligned} & Pr[E_{3,\mathcal{C}_i}(\boldsymbol{\ell}'_{\mathcal{C}_i}) \mid \left(\boldsymbol{U}_{\mathcal{D}_i}(\boldsymbol{m}_{\mathcal{D}_i}), \boldsymbol{Y}_i\right) \in A^*_\epsilon(U_{\mathcal{D}_i}, Y_i)] \\ \leq\ & 2^{N\left[H(U_{\mathcal{C}_i}|U_{\mathcal{D}_i\setminus\mathcal{C}_i},Y_i)-\sum_{(j\to\mathcal{A})\in\mathcal{C}_i}H(U_{j\to\mathcal{A}})+\epsilon_3+\epsilon_4\right]} \end{aligned}$$

where $\epsilon_3, \epsilon_4 \to 0$ as $\epsilon \to 0$. We bound

$$\prod_{(j\to\mathcal{A})\in\mathcal{C}_i}(|\mathcal{B}_{j\to\mathcal{A}}|-1) < \prod_{(j\to\mathcal{A})\in\mathcal{C}_i}|\mathcal{B}_{j\to\mathcal{A}}| = 2^{N\sum_{(j\to\mathcal{A})\in\mathcal{C}_i}(\tilde{R}_{j\to\mathcal{A}}-R_{j\to\mathcal{A}})}$$

Hence

$$P_e \leq \sum_{\mathcal{C}_i\subseteq\mathcal{D}_i} 2^{-N\left[\sum_{(j\to\mathcal{A})\in\mathcal{C}_i}R_{j\to\mathcal{A}}-\sum_{(j\to\mathcal{A})\in\mathcal{C}_i}\tilde{R}_{j\to\mathcal{A}}+\sum_{(j\to\mathcal{A})\in\mathcal{C}_i}H(U_{j\to\mathcal{A}})-H(U_{\mathcal{C}_i}|U_{\mathcal{D}_i\setminus\mathcal{C}_i},Y_i)-\epsilon_3-\epsilon_4\right]}$$

Thus, by selecting sufficiently large N and the rates such that

$$\sum_{(j\to\mathcal{A})\in\mathcal{C}_i}R_{j\to\mathcal{A}} > \sum_{(j\to\mathcal{A})\in\mathcal{C}_i}\tilde{R}_{j\to\mathcal{A}} - \sum_{(j\to\mathcal{A})\in\mathcal{C}_i}H(U_{j\to\mathcal{A}}) + H(U_{\mathcal{C}_i}|U_{\mathcal{D}_i\setminus\mathcal{C}_i},Y_i) + \epsilon_3+\epsilon_4$$

P_e can be made arbitrarily small. Note that $Pr(E_3\cap E_0^c) \leq \sum_{i\in[M]}Pr(E_{3,\mathcal{D}_i}\cap$

E_0^c). Thus, $Pr(E_3 \cap E_0^c)$ is small.

At node $i \in [M]$ when $\left(\boldsymbol{U}_{\mathcal{D}_i}(\boldsymbol{\ell}_{\mathcal{D}_i}), \boldsymbol{Y}_{\mathcal{E}(\mathcal{D}_i)}, \boldsymbol{Y}_i, \boldsymbol{T}\right) \in A_\epsilon^*(U_{\mathcal{D}_i}, Y_{\mathcal{E}(\mathcal{D}_i)}, Y_i, T)$, the empirical distribution on $\mathcal{T} \times \mathcal{Y}_i \times \mathcal{U}_{\mathcal{D}_i}$ is approximately equal to the true distribution, and, thus the expected distortion is approximately D_i. From Lemma 1, the probability of succesful coding $\to 1$ as $N \to \infty$. Thus, the overall expected distortion is approximately D_i at node $i \in [M]$. This completes the proof, i.e. $\mathbf{conv}\,(\mathcal{RD}_{in}) \subseteq \mathcal{RD}$. $\square$

Appendix B. Proof of Theorem 2

For each $(j \to \mathcal{A}) \in \mathcal{S}$, let $\boldsymbol{X}_{j\to\mathcal{A}} = f^N_{j\to\mathcal{A}}(\boldsymbol{Y}_j)$. Then, for each $i \in [M]$ and $\mathcal{C}_i \subseteq \mathcal{D}_i$

$$
\begin{aligned}
& N \sum_{(j\to\mathcal{A})\in\mathcal{C}_i} R_{j\to\mathcal{A}} \\
\geq & \sum_{(j\to\mathcal{A})\in\mathcal{C}_i} H(\boldsymbol{X}_{j\to\mathcal{A}}) \\
\geq & H\left(\boldsymbol{X}_{\mathcal{C}_i}\right) \\
\geq & H\left(\boldsymbol{X}_{\mathcal{C}_i} \,\middle|\, \boldsymbol{Y}_i,\ \boldsymbol{X}_{\mathcal{D}_i\setminus\mathcal{C}_i}\right) \\
\geq & H\left(\boldsymbol{X}_{\mathcal{C}_i} \,\middle|\, \boldsymbol{Y}_i,\ \boldsymbol{X}_{\mathcal{D}_i\setminus\mathcal{C}_i}\right) - H\left(\boldsymbol{X}_{\mathcal{C}_i} \,\middle|\, \boldsymbol{Y}_{[M]},\ \boldsymbol{X}_{\mathcal{D}_i\setminus\mathcal{C}_i}\right) \\
= & I\left(\boldsymbol{Y}_{[M]\setminus i};\ \boldsymbol{X}_{\mathcal{C}_i} \,\middle|\, \boldsymbol{Y}_i,\ \boldsymbol{X}_{\mathcal{D}_i\setminus\mathcal{C}_i}\right) \\
= & H\left(\boldsymbol{Y}_{[M]\setminus i} \,\middle|\, \boldsymbol{Y}_i,\ \boldsymbol{X}_{\mathcal{D}_i\setminus\mathcal{C}_i}\right) - H\left(\boldsymbol{Y}_{[M]\setminus i} \,\middle|\, \boldsymbol{Y}_i,\ \boldsymbol{X}_{\mathcal{D}_i}\right) \\
= & \sum_{n=1}^{N} \left[H\left(Y^{(n)}_{[M]\setminus i} \,\middle|\, \boldsymbol{Y}^{n-1}_{[M]\setminus i},\ \boldsymbol{Y}_i,\ \boldsymbol{X}_{\mathcal{D}_i\setminus\mathcal{C}_i}\right) - H\left(Y^{(n)}_{[M]\setminus i} \,\middle|\, \boldsymbol{Y}^{n-1}_{[M]\setminus i},\ \boldsymbol{Y}_i,\ \boldsymbol{X}_{\mathcal{D}_i}\right)\right]
\end{aligned}
$$

Define

$$
\begin{aligned}
Z^{(n)}_{j\to\mathcal{A}} &\triangleq \left(\boldsymbol{Y}^{n-1}_{[M]}, \boldsymbol{X}_{j\to\mathcal{A}}\right) \\
W^{(n)}_i &\triangleq \left(\boldsymbol{Y}_i^{[n+1,N]}\right)
\end{aligned}
$$

where $\boldsymbol{Y}_i^{[n+1,N]} := \left(Y^{(n+1)}, \ldots, Y^{(N)}\right)$. Hence

$$
\begin{aligned}
N\sum_{n=1}^{N} R_{j\to\mathcal{A}} &\geq \sum_{n=1}^{N} \left[H\left(Y^{(n)}_{[M]\setminus i} \,\middle|\, Y^{(n)}_i, W^{(n)}_i, Z^{(n)}_{\mathcal{D}_i\setminus\mathcal{C}_i}\right) - H\left(Y^{(n)}_{[M]\setminus i} \,\middle|\, Y^{(n)}_i, W^{(n)}_i, Z^{(n)}_{\mathcal{D}_i}\right)\right] \\
&= \sum_{n=1}^{N} I\left(Y^{(n)}_{[M]\setminus i}; Z^{(n)}_{\mathcal{C}_i} \,\middle|\, Y^{(n)}_i, W^{(n)}_i, Z^{(n)}_{\mathcal{D}_i\setminus\mathcal{C}_i}\right)
\end{aligned}
$$

We now show that $\left(W_{[M]}^{(n)}, Z_{\mathcal{S}}^{(n)}\right) \in \Psi$. Note that for each node $i \in [M]$, $\hat{T}_i^{(n)}$ can be expressed as a function of $Z_{\mathcal{D}_i}^{(n)}, W_i^{(n)}$ and $Y_i^{(n)}$, i.e. $\hat{T}_i^{(n)} = g_i^{(n)}\big(Y_i^{(n)}, W_i^{(n)}, Z_{\mathcal{D}_i}^{(n)}\big)$. Thus there exists a decoding function $g_i^{(n)}$ such that

$$E\big[d_i\big(T^{(n)}, g_i^{(n)}\big(Y_i^{(n)}, W_i^{(n)}, Z_{\mathcal{D}_i}^{(n)}\big)\big)\big] \leq D_i^{(n)}$$

It can be easily seen that $p(t^{(n)}, y_{[M]}^{(n)}, w_{[M]}^{(n)}) = p(t^{(n)}, y_{[M]}^{(n)})\ p(w_{[M]}^{(n)})$. It remains to show that $T^{(n)}, Y_{[M]\setminus j}^{(n)} \leftrightarrow Y_j^{(n)} \leftrightarrow Z_{j\to\mathcal{A}}^{(n)}$.

$$\begin{aligned}
& I\Big(Z_{j\to\mathcal{A}}^{(n)}; T^{(n)}, Y_{[M]\setminus j}^{(n)} \,|\, Y_j^{(n)}\Big) \\
= \;& I\Big(T^{(n)}, Y_{[M]\setminus j}^{(n)}; Y_j^{(n)}, Z_{j\to\mathcal{A}}^{(n)}\Big) - I\Big(T^{(n)}, Y_{[M]\setminus j}^{(n)}; Y_j^{(n)}\Big) \\
= \;& I\Big(T^{(n)}, Y_{[M]\setminus j}^{(n)}; Y_j^{(n)}, Y_{[M]}^{n-1}, X_{j\to\mathcal{A}}^{N}\Big) - I\Big(T^{(n)}, Y_{[M]\setminus j}^{(n)}; Y_j^{(n)}\Big) \\
\leq \;& I\Big(T^{(n)}, Y_{[M]\setminus j}^{(n)}; Y_j^{(n)}, Y_{[M]}^{n-1}, Y_j^{N}\Big) - I\Big(T^{(n)}, Y_{[M]\setminus j}^{(n)}; Y_j^{(n)}\Big) \\
= \;& I\Big(T^{(n)}, Y_{[M]\setminus j}^{(n)}; Y_j^{(n)}\Big) - I\Big(T^{(n)}, Y_{[M]\setminus j}^{(n)}; Y_j^{(n)}\Big) \\
= \;& 0
\end{aligned}$$

Hence, $\left(W_{[M]}^{(n)}, Z_{\mathcal{S}}^{(n)}\right) \in \Psi$.

To complete the outer bound proof, it remains to show that there exist random vectors $(T, Y_{[M]}, W_{[M]}, Z_{\mathcal{S}})$ such that at each node $i \in [M]$

$$I\Big(Y_{[M]\setminus i}; Z_{\mathcal{C}_i} \,|\, Y_i, W_i, Z_{\mathcal{D}_i\setminus\mathcal{C}_i}\Big) = \frac{1}{N}\sum_{n=1}^{N} I\Big(Y_{[M]\setminus i}^{(n)}, Z_{\mathcal{C}_i}^{(n)} \,|\, Y_i^{(n)}, W_i^{(n)}, Z_{\mathcal{D}_i\setminus\mathcal{C}_i}^{(n)}\Big)$$

and

$$\begin{aligned}
E[d_i(T, g_i(Y_i, W_i, Z_{\mathcal{D}_i}))] &= D_i \\
&= \frac{1}{N}\sum_{i=1}^{N} D_i^{(n)}
\end{aligned}$$

Let Q be a random variable independent of $\boldsymbol{T}, \boldsymbol{Y}_{[M]}, \boldsymbol{W}_{[M]}$ and $\boldsymbol{Z}_{\mathcal{S}}$ which takes values in the set $\{1, \ldots, N\}$ with probability

$$Pr[Q = n] = \lambda_n \quad \forall\, n \in \{1, \ldots, N\}$$

Define the following random variables for all $(j \to \mathcal{A}) \in \mathcal{S}$ and $i \in [M]$

$$\begin{aligned} T &:= (Q, T^{(Q)}) \\ Y_i &:= (Q, Y_i^{(Q)}) \\ W_i &:= (Q, W_i^{(Q)}) \\ Z_{j\to\mathcal{A}} &:= (Q, Z_{j\to\mathcal{A}}^{(Q)}) \end{aligned}$$

Let $g_i(Y_i, W_i, Z_{\mathcal{D}_i}) = g_i^{(Q)}\left(Y_i^{(Q)}, W_i^{(Q)}, Z_{\mathcal{D}_i}^{(Q)}\right)$. For each $n \in [N]$, $g_i(Y_i, W_i, Z_{\mathcal{D}_i}) = g_i^{(n)}\left(Y_i^{(n)}, W_i^{(n)}, Z_{\mathcal{D}_i}^{(n)}\right)$ with probability λ_n. Let $E[d_i(T, g_i(Y_i, W_i, Z_{\mathcal{D}_i}))] = D_i$. Then, we have

$$\begin{aligned} D_i &= E\left[d_i(T, g_i(Y_i, W_i, Z_{\mathcal{D}_i}))\right] \\ &= \sum_{n=1}^{N} \lambda_n E\left[d_i\left(T^{(n)}, g_i^{(n)}\left(Y_i^{(n)}, W_i^{(n)}, Z_{\mathcal{D}_i}^{(n)}\right)\right)\right] \\ &= \sum_{n=1}^{N} \lambda_n D_i^{(n)} \end{aligned}$$

Also, we have

$$
\begin{aligned}
& I\left(Y_{[M]\backslash i}; Z_{\mathcal{C}_i} \,\middle|\, Y_i, W_i, Z_{\mathcal{D}_i\backslash\mathcal{C}_i}\right) \\
= \;& H\left(Y_{[M]\backslash i} \,\middle|\, Y_i, W_i, Z_{\mathcal{D}_i\backslash\mathcal{C}_i}\right) - H\left(Y_{[M]\backslash i} \,|\, Y_i, W_i, Z_{\mathcal{D}_i}\right) \\
= \;& H\left(Y^{(Q)}_{[M]\backslash i}, Q \,\middle|\, Y^{(Q)}_i, W^{(Q)}_i, Z^{(Q)}_{\mathcal{D}_i\backslash\mathcal{C}_i}, Q\right) - H\left(Y^{(Q)}_{[M]\backslash i}, Q \,\middle|\, Y^{(Q)}_i, W^{(Q)}_i, Z^{(Q)}_{\mathcal{D}_i}, Q\right) \\
= \;& \sum_{n=1}^{N} \lambda_n \left[H\left(Y^{(n)}_{[M]\backslash i} \,\middle|\, Y^{(n)}_i, W^{(n)}_i, Z^{(n)}_{\mathcal{D}_i\backslash\mathcal{C}_i}\right) - H\left(Y^{(n)}_{[M]\backslash i} \,\middle|\, Y^{(n)}_i, W^{(n)}_i, Z^{(n)}_{\mathcal{D}_i}\right)\right] \\
= \;& \sum_{n=1}^{N} \lambda_n I\left(Y^{(n)}_{[M]\backslash i}, Z^{(n)}_{\mathcal{C}_i} \,\middle|\, Y^{(n)}_i, W^{(n)}_i, Z^{(n)}_{\mathcal{D}_i\backslash\mathcal{C}_i}\right)
\end{aligned}
$$

Let $\lambda_n = \frac{1}{N}$. This completes the outer bound proof, i.e. $\mathbf{conv}\left(\mathcal{RD}_{out}\right) \supseteq \mathcal{RD}$. $\square$

Appendix C. Proof of Theorem 3

We provide a detailed proof of our inner bound in this section.

Probability of Error Analysis:

Assuming encoding and decoding are done without error upto $k-1$ rounds, we analyze the probability of error of encoding and decoding at round k.

Define the following events.

1. $E_1 = \{$There does not exist a $\boldsymbol{W}_k(i_k)$ such that $(\boldsymbol{Y}_{\mu(k)}, \{\boldsymbol{W}_\ell(i_\ell)\}_{\ell=1}^{k-1}, \{\boldsymbol{U}_\ell(i_\ell, j_\ell)\mathbf{1}[\lambda(\ell) = \mu(k)]\}_{\ell=1}^{k-1}, \boldsymbol{W}_k(i_k)) \in \mathcal{A}_\epsilon^N\}$

2. $E_2 = \{$There does not exist a $\boldsymbol{U}_k(i_k, j_k)$ such that $(\boldsymbol{Y}_{\mu(k)}, \{\boldsymbol{W}_\ell(i_\ell)\}_{\ell=1}^{k}, \{\boldsymbol{U}_\ell(i_\ell, j_\ell)\mathbf{1}[\lambda(\ell) = \mu(k)]\}_{\ell=1}^{k-1}, \boldsymbol{U}_k(i_k, j_k)) \in \mathcal{A}_\epsilon^N\}$

3. $E_3 = \{$There does not exist a pair $(\boldsymbol{W}_k(i_k), \boldsymbol{U}_k(i_k, j_k))$ such that $(\boldsymbol{Y}_1, \boldsymbol{Y}_2, \boldsymbol{Y}_3, \{\boldsymbol{W}_\ell(i_\ell), \boldsymbol{U}_\ell(i_\ell, j_\ell)\}_{\ell=1}^{k-1}, \boldsymbol{W}_k(i_k), \boldsymbol{U}_k(i_k, j_k)) \in \mathcal{A}_\epsilon^N\}$

4. $E_4 = \{(\boldsymbol{Y}_{\phi(k)}, \{\boldsymbol{W}_\ell(i_\ell)\}_{\ell=1}^{k-1}, \{\boldsymbol{U}_\ell(i_\ell, j_\ell)\mathbf{1}[\lambda(\ell) = \phi(k)]\}_{\ell=1}^{k-1}, \boldsymbol{W}_k(\tilde{i}_k)) \in \mathcal{A}_\epsilon^N,$ but $\tilde{i_k} \neq i_k\}$

5. $E_5 = \{(\boldsymbol{Y}_{\lambda(k)}, \{\boldsymbol{W}_\ell(i_\ell)\}_{\ell=1}^{k-1}, \{\boldsymbol{U}_\ell(i_\ell, j_\ell)\mathbf{1}[\lambda(\ell) = \lambda(k)]\}_{\ell=1}^{k-1}, \boldsymbol{W}_k(i'_k)) \in \mathcal{A}_\epsilon^N,$ but $i'_k \neq i_k\}$

6. $E_6 = \{(\boldsymbol{Y}_{\lambda(k)}, \{\boldsymbol{W}_\ell(i_\ell)\}_{\ell=1}^{k-1}, \{\boldsymbol{U}_\ell(i_\ell, j_\ell)\mathbf{1}[\lambda(\ell) = \lambda(k)]\}_{\ell=1}^{k-1}, \boldsymbol{W}_k(i'_k), \boldsymbol{U}_k(i'_k, j'_k)) \in \mathcal{A}_\epsilon^N,$ but $j'_k \neq j_k\}$

The probability of error $P(e_k)$ at round k

$$\begin{aligned} P(e_k) \;\leq\; & P(E_1) + P(E_2|E_1^c) + P(E_3|E_1^c \cap E_2^c) + \\ & P(E_4|E_1^c) + P(E_5|E_1^c) + P(E_6|E_1^c \cap E_2^c \cap E_5^c) \end{aligned}$$

We first show $P(E_1) \to 0$ as $N \to \infty$.

$$
\begin{aligned}
& Pr[\boldsymbol{W}_k \in \mathcal{A}_\epsilon^N(W_k|Y_{\mu(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell) = \mu(k)]\}_{\ell=1}^{k-1}) \mid \boldsymbol{W}_k \in \mathcal{A}_\epsilon^N(W_k)] \\
= & \frac{|\mathcal{A}_\epsilon^N(W_k|Y_{\mu(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell) = \mu(k)]\}_{\ell=1}^{k-1})|}{|\mathcal{A}_\epsilon^N(W_k)|} \\
= & \frac{2^{NH(W_k|Y_{\mu(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell)=\mu(k)]\}_{\ell=1}^{k-1})}}{2^{NH(W_k)}} \\
= & 2^{-NI(W_k; Y_{\mu(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell)=\mu(k)]\}_{\ell=1}^{k-1})}
\end{aligned}
$$

$$
\begin{aligned}
& P(E_1) \\
= & \left(1 - Pr[\boldsymbol{W}_k \in \mathcal{A}_\epsilon^N(W_k|Y_{\mu(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell) = \mu(k)]\}_{\ell=1}^{k-1}) \mid \boldsymbol{W}_k \in \mathcal{A}_\epsilon^N(W_k)]\right)^{\tilde{L}_1^k} \\
\leq & \exp\{-\tilde{L}_1^k Pr[\boldsymbol{W}_k \in \mathcal{A}_\epsilon^N(W_k|Y_{\mu(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell) = \mu(k)]\}_{\ell=1}^{k-1}) \mid \boldsymbol{W}_k \in \mathcal{A}_\epsilon^N(W_k)]\} \\
= & \exp[-\tilde{L}_1^k 2^{-NI(W_k; Y_{\mu(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell)=\mu(k)]\}_{\ell=1}^{k-1})}] \\
= & \exp\{-2^{N[\tilde{R}_1^k - I(W_k; Y_{\mu(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell)=\mu(k)]\}_{\ell=1}^{k-1})]}\}
\end{aligned}
$$

If $\tilde{R}_1^k \geq I(W_k; Y_{\mu(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell) = \mu(k)]\}_{\ell=1}^{k-1})$, $P(E_1) \to 0$ as $N \to \infty$. By similar argument, we can show $P(E_2|E_1^c) \to 0$ as $N \to \infty$ if

$$\tilde{R}_2^k \geq I(U_k; Y_{\mu(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell) = \mu(k)]\}_{\ell=1}^{k-1} \mid W_k)$$

We now use the Markov lemma to show $P(E_3|E_1^c \cap E_2^c) \to 0$ as $N \to \infty$. According

to the Markov lemma, if

$$W_k, U_k \leftrightarrow Y_{\mu(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell) = \mu(k)]\}_{\ell=1}^{k-1} \leftrightarrow$$
$$Y_{\lambda(k)}, Y_{\phi(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell) = \lambda(k)]\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell) = \phi(k)]\}_{\ell=1}^{k-1}$$
$$P[(\boldsymbol{Y}_1, \boldsymbol{Y}_2, \boldsymbol{Y}_3, \{\boldsymbol{W}_\ell(i_\ell), \boldsymbol{U}_\ell(i_\ell, j_\ell)\}_{\ell=1}^{k-1}) \in \mathcal{A}_\epsilon^N] \to 1 \text{ as } N \to \infty$$
$$P[(\boldsymbol{Y}_{\mu(k)}, \{\boldsymbol{W}_\ell(i_\ell)\}_{\ell=1}^{k-1}, \{\boldsymbol{U}_\ell(i_\ell, j_\ell)\mathbf{1}[\lambda(\ell) = \mu(k)]\}_{\ell=1}^{k-1}, \boldsymbol{W}_k(i_k), \boldsymbol{U}_k(i_k, j_k)) \in \mathcal{A}_\epsilon^N] \to 1$$
$$\text{as } N \to \infty$$

then

$$P[(\boldsymbol{Y}_1, \boldsymbol{Y}_2, \boldsymbol{Y}_3, \{\boldsymbol{W}_\ell(i_\ell), \boldsymbol{U}_\ell(i_\ell, j_\ell)\}_{\ell=1}^{k-1}, \boldsymbol{W}_k(i_k), \boldsymbol{U}_k(i_k, j_k)) \in \mathcal{A}_\epsilon^N] \to 1 \text{ as } N \to \infty$$

Hence, $P(E_3|E_1^c \cap E_2^c) \to 0$ as $N \to \infty$.

We next show $P(E_4|E_1^c) \to 0$ as $N \to \infty$.

$$\begin{aligned}
& Pr[\boldsymbol{W}_k(\tilde{i}_k) \in \mathcal{A}_\epsilon^N(W_k|Y_{\phi(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell) = \phi(k)]\}_{\ell=1}^{k-1}) \mid \boldsymbol{W}_k(\tilde{i}_k) \in \mathcal{A}_\epsilon^N(W_k)] \\
= & \frac{|\mathcal{A}_\epsilon^N(W_k|Y_{\phi(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell) = \phi(k)]\}_{\ell=1}^{k-1})|}{|\mathcal{A}_\epsilon^N(W_k)|} \\
= & \frac{2^{NH(W_k|Y_{\phi(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell)=\phi(k)]\}_{\ell=1}^{k-1})}}{2^{NH(W_k)}} \\
= & 2^{-NI(W_k; Y_{\phi(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell)=\phi(k)]\}_{\ell=1}^{k-1})}
\end{aligned}$$

$$
\begin{aligned}
& P(E_4|E_1^c) \\
\leq\ & \frac{\tilde{L}_1^k}{L_1^k} Pr[\boldsymbol{W}_k(\tilde{i}_k) \in \mathcal{A}_\epsilon^N(W_k|Y_{\phi(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell)=\phi(k)]\}_{\ell=1}^{k-1}) \mid \boldsymbol{W}_k(\tilde{i}_k) \in \mathcal{A}_\epsilon^N(W_k)] \\
=\ & \frac{\tilde{L}_1^k}{L_1^k} 2^{-NI(W_k;Y_{\phi(k)},\{W_\ell\}_{\ell=1}^{k-1},\{U_\ell \mathbf{1}[\lambda(\ell)=\phi(k)]\}_{\ell=1}^{k-1})} \\
=\ & 2^{-N[R_1^k-(\tilde{R}_1^k-I(W_k;Y_{\phi(k)},\{W_\ell\}_{\ell=1}^{k-1},\{U_\ell \mathbf{1}[\lambda(\ell)=\phi(k)]\}_{\ell=1}^{k-1}))]}
\end{aligned}
$$

$P(E_4|E_1^c) \to 0$ as $N \to \infty$ if

$$
\begin{aligned}
R_1^k \ \geq\ & \tilde{R}_1^k - I(W_k; Y_{\phi(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell)=\phi(k)]\}_{\ell=1}^{k-1}) \\
=\ & I(W_k; Y_{\mu(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell)=\mu(k)]\}_{\ell=1}^{k-1}) - \\
& I(W_k; Y_{\phi(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell)=\phi(k)]\}_{\ell=1}^{k-1}) \\
=\ & I(W_k; Y_{\mu(k)}, \{U_\ell \mathbf{1}[\lambda(\ell)=\mu(k)]\}_{\ell=1}^{k-1} \mid Y_{\phi(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell)=\phi(k)]\}_{\ell=1}^{k-1})
\end{aligned}
$$

Similarly , we can show $P(E_5|E_1^c) \to 0$ as $N \to \infty$ if

$$
R_1^k \geq I(W_k; Y_{\mu(k)}, \{U_\ell \mathbf{1}[\lambda(\ell)=\mu(k)]\}_{\ell=1}^{k-1} \mid Y_{\lambda(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell)=\lambda(k)]\}_{\ell=1}^{k-1})
$$

Hence,

$$
\begin{aligned}
& R_1^k \\
\geq\ & \max[I(W_k; Y_{\mu(k)}, \{U_\ell \mathbf{1}[\lambda(\ell)=\mu(k)]\}_{\ell=1}^{k-1} \mid Y_{\phi(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell)=\phi(k)]\}_{\ell=1}^{k-1}), \\
& \quad I(W_k; Y_{\mu(k)}, \{U_\ell \mathbf{1}[\lambda(\ell)=\mu(k)]\}_{\ell=1}^{k-1} \mid Y_{\lambda(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell)=\lambda(k)]\}_{\ell=1}^{k-1})] \\
=\ & I(W_k; Y_{\mu(k)}, \{U_\ell \mathbf{1}[\lambda(\ell)=\mu(k)]\}_{\ell=1}^{k-1} \mid Y_{\phi(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell)=\phi(k)]\}_{\ell=1}^{k-1})
\end{aligned}
$$

Finally, we show $P(E_6|E_1^c \cap E_2^c \cap E_5^c) \to 0$ as $N \to \infty$.

$$
\begin{aligned}
& Pr[\boldsymbol{U}_k(j_k') \in \mathcal{A}_\epsilon^N(U_k|Y_{\lambda(k)}, \{W_\ell\}_{\ell=1}^k, \{U_\ell \mathbf{1}[\lambda(\ell)=\lambda(k)]\}_{\ell=1}^{k-1})|\ \boldsymbol{U}_k(j_k') \in \mathcal{A}_\epsilon^N(U_k|W_k)] \\
= & \frac{|\mathcal{A}_\epsilon(U_k|Y_{\lambda(k)}, \{W_\ell\}_{\ell=1}^k, \{U_\ell \mathbf{1}[\lambda(\ell)=\lambda(k)]\}_{\ell=1}^{k-1})|}{|\mathcal{A}_\epsilon(U_k|W_k)|} \\
= & \frac{2^{NH(U_k|Y_{\lambda(k)}, \{W_\ell\}_{\ell=1}^k, \{U_\ell \mathbf{1}[\lambda(\ell)=\lambda(k)]\}_{\ell=1}^{k-1})}}{2^{NH(U_k|W_k)}} \\
= & 2^{-NI(U_k; Y_{\lambda(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell)=\lambda(k)]\}_{\ell=1}^{k-1}\ |\ W_k)}
\end{aligned}
$$

$$
\begin{aligned}
& P(E_6|E_1^c \cap E_2^c \cap E_5^c) \\
\leq & \frac{\tilde{L}_2^k}{L_2^k} Pr[\boldsymbol{U}_k(j_k') \in \mathcal{A}_\epsilon^N(U_k|Y_{\lambda(k)}, \{W_\ell\}_{\ell=1}^k, \{U_\ell \mathbf{1}[\lambda(\ell)=\lambda(k)]\}_{\ell=1}^{k-1})|\ \boldsymbol{U}_k(j_k') \in \mathcal{A}_\epsilon^N(U_k|W_k)] \\
= & \frac{\tilde{L}_2^k}{L_2^k} 2^{-NI(U_k; Y_{\lambda(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell)=\lambda(k)]\}_{\ell=1}^{k-1}\ |\ W_k)} \\
= & 2^{-N[R_2^k - (\tilde{R}_2^k - I(U_k; Y_{\lambda(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell)=\lambda(k)]\}_{\ell=1}^{k-1}\ |\ W_k))]}
\end{aligned}
$$

$P(E_6|E_1^c \cap E_2^c \cap E_5^c) \to 0$ as $N \to \infty$ if

$$
\begin{aligned}
R_2^k \geq & \tilde{R}_2^k - I(U_k; Y_{\lambda(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell)=\lambda(k)]\}_{\ell=1}^{k-1}\ |\ W_k) \\
= & I(U_k; Y_{\mu(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell)=\mu(k)]\}_{\ell=1}^{k-1}|\ W_k) \\
& - I(U_k; Y_{\lambda(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell)=\lambda(k)]\}_{\ell=1}^{k-1}|\ W_k) \\
= & I(U_k; Y_{\mu(k)}, \{U_\ell \mathbf{1}[\lambda(\ell)=\mu(k)]\}_{\ell=1}^{k-1}\ |\ W_k, Y_{\lambda(k)}, \{W_\ell\}_{\ell=1}^{k-1}, \{U_\ell \mathbf{1}[\lambda(\ell)=\lambda(k)]\}_{\ell=1}^{k-1})
\end{aligned}
$$

Thus, if we select the rates $\tilde{R}_1^k, \tilde{R}_2^k, R_1^k, R_2^k$ such that above conditions are satisfied $P(e_k) \to 0$ as $N \to \infty$. $\square$

Bibliography

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks (Elsevier)*, vol. 38, no. 4, pp. 393–422, Mar. 2002.

[2] H. L. Van Trees, *Detection, Estimation, and Modulation Theory (Part 1)*. John Wiley & Sons, 1968.

[3] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer Science + Business Media, LLC, 2006.

[4] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3122–3136, July 2008.

[5] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1865–1877, May 2008.

[6] C. C. Moallemi and B. Van Roy, "Consensus propagation," *IEEE Transactions on Information Theory*, vol. 52, no. 11, pp. 4753–4766, Nov. 2006.

[7] I. D. Schizas and A. Ribeiro and G. B. Giannakis, "Consensus in Ad Hoc WSNs with noisy links-Part I: Distributed estimation of deterministic signals," *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 350–364, Jan. 2008.

[8] I. D. Schizas and G. B. Giannakis and S. I. Roumeliotis and A. Ribeiro, "Consensus in Ad Hoc WSNs with noisy links-Part II: Distributed estimation and smoothing of random signals," *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1650–1666, Apr. 2008.

[9] T. P. Minka, "A family of algorithms for approximate bayesian inference," PhD Thesis, Massachusetts Institute of Technology, 2001.

[10] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. John Wiley & Sons, Inc, 2006.

[11] R. L. Dobrushin and B. S. Tsybakov, "Information transmission with additional noise," *IEEE Transactions on Information Theory*, vol. IT-8, no. 5, pp. 293–304, September 1962.

[12] A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Transactions on Information Theory*, vol. IT-22, no. 1, pp. 1–10, January 1976.

[13] S. C. Draper and G. W. Wornell, "Side information aware coding strategies for sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, pp. 966–976, August 2004.

[14] T. Berger, Z. Zhang, and H. Viswanathan, "The CEO Problem," *IEEE Transactions on Information Theory*, vol. 42, no. 3, pp. 887–902, May 1996.

[15] J. Chen, X. Zhang, T. Berger, and S. B. Wicker, "An Upper Bound on the Sum-Rate Distortion Function and Its Corresponding Rate Allocation Schemes for the CEO Problem," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, pp. 977–987, August 2004.

[16] A. B. Wagner and V. Anantharam, "An improved outer bound for multiterminal source coding," *IEEE Transactions on Information Theory*, vol. 54, no. 5, pp. 1919–1937, May 2008.

[17] Yasutada Oohama, "Rate-distortion theory for Gaussian multiterminal source coding system with several side informations at the decoder," *IEEE Transactions on Information Theory*, vol. 51, no. 7, pp. 2577–2593, Jul. 2005.

[18] V. Prabhakaran and D. Tse and K. Ramchandran, "Rate region of the quadratic Gaussian CEO problem," *Proceedings of International Symposium on Information Theory*, p. 117, Jun. 2004.

[19] A. A. E. Gamal and T. M. Cover, "Achievable rates for multiple descriptions," *IEEE Transactions on Information Theory*, vol. IT-28, no. 6, pp. 851–857, November 1982.

[20] L. Ozarow, "On a source-coding problem with two channels and three receivers," *The Bell System Technical Journal*, vol. 59, no. 10, pp. 1909–1921, December 1980.

[21] W. H. R. Equitz and T. M. Cover, "Successive Refinement of Information," *IEEE Transactions on Information Theory*, vol. 37, no. 2, pp. 269–275, Mar. 1991.

[22] Y. Steinberg and N. Merhav, "On successive refinement for the Wyner-Ziv problem," *IEEE Transactions on Information Theory*, vol. 50, no. 8, pp. 1636–1654, Aug. 2004.

[23] C. Heegard and T. Berger, "Rate distortion when side information may be absent," *IEEE Transactions on Information Theory*, vol. 31, no. 6, pp. 727–734, November 1985.

[24] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. IT-19, no. 3, pp. 471–480, March 1973.

[25] J. Garcia-Frias and Y. Zhao, "Compression of correlated binary sources using turbo codes," *IEEE Communications Letters*, vol. 5, no. 10, pp. 417–419, Oct. 2001.

[26] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Compression of Binary Sources With Side Information at the decoder Using LDPC Codes," *IEEE Communications Letters*, vol. 6, no. 10, pp. 440–442, Oct. 2002.

[27] S. Sandeep Pradhan and K. Ramchandran, "Distributed Source Coding Using Syndromes (DISCUS): Design and Construction," *IEEE Transactions on Information Theory*, vol. 49, no. 3, pp. 626–643, Mar. 2003.

[28] Z. Xiong, A. D. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *IEEE Signal Processing Magazine*, vol. 21, no. 5, pp. 80–94, Sep. 2004.

[29] Y. Yang, S. Cheng, Z. Xiong, and W. Zhao, "Wyner-ziv coding based on tcq and ldpc codes," *IEEE Transactions on Communications*, vol. 57, no. 2, pp. 376–387, Feb. 2009.

[30] Y. Yang, V. Stankovic, Z. Xiong, and W. Zhao, "On Multiterminal Source Code Design," *IEEE Transactions on Information Theory*, vol. 54, no. 5, pp. 2278–2302, May 2008.

[31] C. Yu and G. Sharma, "Distributed estimation and coding: A sequential framework based on a side-informed decomposition," *IEEE Transactions on Signal Processing*, vol. 59, no. 2, pp. 759–773, Feb. 2011.

[32] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *IEEE Transactions on Information Theory*, vol. 51, no. 7, pp. 2282–2312, July 2005.

[33] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[34] T. P. Minka, "Expectation Propagation for approximate Bayesian inference," in *Proceedings of the 17th Conference in Artificial Intelligence (UAI)*, 2001, pp. 362–369.

[35] H. Yamamoto and K. Itoh, "Source coding theory for multiterminal communication systems with a remote source," *IEEE Transactions of the IECE of Japan*, vol. E 63, no. 10, pp. 700–706, Oct. 1980.

[36] H. Yamamoto, "Wyner-Ziv theory for a general function of the correlated sources," *IEEE Transactions on Information Theory*, vol. 28, no. 5, pp. 803–807, Sep. 1982.

[37] T. Berger, "Multiterminal Source Coding," in *The Information Theory approach to Communications*, vol. 229. New York: Springer-Verlag, 1978, pp. 171–231.

[38] S.-Y. Tung, "Multiterminal source coding," Ph.D. dissertation, Cornell University, 1978.

[39] T. S. Han and K. Kobayashi, "A unified achievable rate region for a general class of multiterminal source coding systems," *IEEE Transactions on Information Theory*, vol. IT-26, no. 3, pp. 277–288, May 1980.

[40] Jun Chen and Toby Berger, "Successive Wyner-Ziv coding coding scheme and its application to the quadratic Gaussian CEO problem," *IEEE Transactions on Information Theory*, vol. 54, no. 4, pp. 1586–1603, Apr. 2008.

[41] V. Doshi and D. Shah and M. Medard and M. Effros, "Functional compression through graph coloring," *IEEE Transactions on Information Theory*, vol. 56, no. 8, pp. 3901–3917, Aug. 2010.

[42] H. Witsenhausen, "The zero-error side information problem and chromatic numbers," *IEEE Transactions on Information Theory*, vol. 22, no. 5, pp. 592–593, Sep. 1976.

[43] R. Ahlswede, "The rate-distortion region for multiple descriptions without excess rate," *IEEE Transactions on Information Theory*, vol. 31, no. 6, pp. 721–726, Nov. 1985.

[44] Z. Zhang and T. Berger, "New Results in Binary Multiple Descriptions," *IEEE Transactions on Information Theory*, vol. 33, no. 4, pp. 502–521, Jul. 1987.

[45] R. Venkataramani, G. Kramer, and V. K. Goyal, "Multiple description coding with many channels," *IEEE Transactions on Information Theory*, vol. IT-49, no. 9, pp. 2106–2114, September 2003.

[46] Jun Chen, "Rate Region of Gaussian Multiple Description Coding With Individual and Central Distortion Constraints," *IEEE Transactions on Information Theory*, vol. 55, no. 9, pp. 3991–4005, Sep. 2009.

[47] R. Puri, S. S. Pradhan, and K. Ramchandran, "n-Channel Symmetric Multiple Descriptions-Part II: An Achievable Rate-Distortion Region," *IEEE Transactions on Information Theory*, vol. IT-51, no. 4, pp. 1377–1392, April 2005.

[48] C. Tian and S. Mohajer and S. N. Diggavi, "Approximating the Gaussian multiple description rate region under symmetric distortion constraints," *IEEE Transactions on Information Theory*, vol. 55, no. 8, pp. 3869–3891, Aug. 2009.

[49] S. N. Diggavi and V. A. Vaishampayan, "On multiple description source coding with decoder side information," in *IEEE Information Theory Workshop*, 2004, pp. 88–93.

[50] Jia Wang and Songyu Yu and Jun Sun, "New results on multiple descriptions in the wyner-ziv setting," *IEEE Transactions on Information Theory*, vol. 55, no. 4, pp. 1701–1708, Apr. 2009.

[51] F.-W. Fu and R. W. Yeung, "On the rate-distortion region for multiple descriptions," *IEEE Transactions on Information Theory*, vol. 48, no. 7, pp. 2012–2021, July 2002.

[52] R. W. Yeung and Zhen Zhang, "Distributed source coding for satellite communications," *IEEE Transactions on Information Theory*, vol. 45, no. 4, pp. 1111–1120, May 1999.

[53] J. R. Roche and R. W. Yeung and Ka Pun Hau, "Symmetrical multilevel diversity coding," *IEEE Transactions on Information Theory*, vol. 43, no. 3, pp. 1059–1064, May 1997.

[54] R. W. Yeung and Zhen Zhang, "On symmetrical multilevel diversity coding," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 609–621, Mar. 1999.

[55] Bixio Rimoldi, "Successive Refinement of Information: Characterization of the Achievable Rates," *IEEE Transactions on Information Theory*, vol. 40, no. 1, pp. 253–259, Jan. 1994.

[56] T. R. Fischer and M. Wang, "Entropy-constrained trellis-coded quantization," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 415–426, Mar. 1992.

[57] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, 2002.

[58] R. Gray, "Vector quantization," *IEEE ASSP Magazine*, vol. 1, no. 2, pp. 4–29, Apr. 1984.

[59] M. W. Marcellin and T. R. Fischer, "Trellis Coded Quantization of Memoryless and Gauss-Markov Sources," *IEEE Transactions on Communications*, vol. 38, no. 1, pp. 82–93, Jan. 1990.

[60] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Transactions on Information Theory*, vol. IT-28, no. 1, pp. 55–67, Jan. 1982.

[61] H. Jafarkhani and V. Tarokh, "Design of successively refinable trellis-coded quantizers," *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1490–1497, Jul. 1999.

[62] D. J. C. MacKay, *Information Theory, Inference and Learning algorithms*. Cambridge University Press, 2004.

[63] ——, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.

[64] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[65] A. Ribeiro and G. B. Giannakis, "Bandwidth-Constrained Distributed Estimation for Wireless Sensor Networks - Part I: Gaussian Case," *IEEE Transactions on Signal Processing*, vol. 54, no. 3, pp. 1131–1143, Mar. 2006.

[66] S. Ramanan and J. M. Walsh, "Distributed estimation of channel gains in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3097–3107, Jun. 2010.

[67] S. Ramanan and J. M. Walsh, "Distributed estimation of channel gains in sensor networks," in *42nd Asilomar conference on signals, systems and computers*, 2008, pp. 1953–1957.

[68] A. Ephremides, "Energy concerns in wireless networks," *IEEE Wireless Communications*, vol. 9, no. 4, pp. 48–59, Aug. 2002.

[69] V. Kawadia and P. R. Kumar, "Principles and protocols for power control in wireless ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 76–88, Jan. 2005.

[70] C. fan Hsin and M. Liu, "Randomly duty-cycled wireless sensor networks: Dynamics of coverage," *IEEE Transactions on Wireless Communications*, vol. 5, no. 11, pp. 3182–3192, Nov. 2006.

[71] O. Dousse, P. Mannersalo, and P. Thiran, "Latency of wireless sensor networks with uncoordinated power saving mechanisms," in *Proceedings of the 5th ACM international symposium on Mobile ad hoc Networking and computing (MobiHoc)*, 2004.

[72] J. M. Walsh, S. Ramanan, and P. A. Regalia, "Optimality of expectation propagation based distributed estimation for wireless sensor network initialization," in *IEEE International Workshop on Signal Processing Advances for Wireless Communications*, 2008, pp. 620–624.

[73] K. Kredo II and P. Mohapatra, "Medium access control in wireless sensor networks," *Computer Networks (Elsevier)*, vol. 51, no. 4, pp. 961–994, Mar. 2007.

[74] G. Scutari, S. Barbarossa, and L. Pescosolido, "Distributed decision through self-synchronizing sensor networks in the presence of propagation delays and asymmetric channels," *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1667–1684, Apr. 2008.

[75] S. Barbarossa and G. Scutari, "Bio-inspired sensor network design," *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 26–35, May 2007.

[76] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey," *Ad Hoc Networks (Elsevier)*, vol. 3, no. 3, pp. 281–323, May 2005.

[77] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: A survey," *IEEE Network*, vol. 18, no. 4, pp. 45–50, July-Aug. 2004.

[78] J. B. Andersen, T. S. Rappaport, and S. Yoshida, "Propagation measurements and models for wireless communications channels," *IEEE Communications Magazine*, vol. 33, no. 1, pp. 42–49, Jan. 1995.

[79] D. Cassioli, M. Z. Win, and A. F. Molisch, "The ultra-wide bandwidth indoor channel: From statistical model to simulations," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 6, pp. 1247–1257, Aug. 2002.

[80] E. Green and M. Hata, "Microcellular propagation measurements in an urban environment," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 1991, pp. 324–328.

[81] A. Saleh and R. Valenzuela, "A statistical model for indoor multipath propagation," *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 2, pp. 128–137, Feb. 1987.

[82] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.

[83] A. F. Molisch, *Wireless Communications*. John Wiley & Sons, Ltd., 2005.

[84] G. L. Stuber, *Principles of Mobile Communication*, 2nd ed. Kluwer Academic Publishers, 2001.

[85] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[86] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, 1999.

[87] S. Ramanan and J. M. Walsh, "Coding perspectives for collaborative estimation over networks," in *44th Asilomar conference on signals, systems and computers*, 2010, pp. 1442–1446.

[88] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, February 2003.

[89] D. J. A. Welsh, *Matroid Theory*. Dover Publications, 2010.

[90] D. N. C. Tse and S. V. Hanly, "Multiaccess Fading Channels - Part I: Polymatroid Structure, Optimal Resource Allocation and Throughput Capacities," *IEEE Transactions on Information Theory*, vol. 44, no. 7, pp. 2796–2815, November 1998.

[91] A. A. El Gamal and T. M. Cover, "Achievable rates for multiple descriptions," *IEEE Transactions on Information Theory*, vol. IT-28, no. 6, pp. 851–857, November 1982.

[92] A. H. Kaspi, "Two-way source coding with a fidelity criterion," *IEEE Transactions on Information Theory*, vol. IT-31, no. 6, pp. 735–740, Nov. 1985.

[93] H. H. Permuter and Y. Steinberg and T. Weissman, "Two-way source coding with a helper," *IEEE Transactions on Information Theory*, vol. 56, no. 6, pp. 2905–2919, Jun. 2010.

[94] A. Orlitsky and J.R. Roche, "Coding for computing," *IEEE Transactions on Information Theory*, vol. 47, no. 3, pp. 903 – 917, 2001.

[95] N. Ma and P. Ishwar, "Two-terminal distributed source coding with alternating messages for function computation," in *IEEE International Symposium on Information Theory (ISIT 2008)*, 2008, pp. 51 – 55.

[96] Nan Ma, P. Ishwar, and P. Gupta, "Information-theoretic bounds for multiround function computation in co-located networks," in *IEEE International Symposium on Information Theory (ISIT)*, 2009, pp. 2306 – 2310.

[97] S. Cheng and Z. Xiong, "Successive Refinement for the Wyner-Ziv Problem and Layered Code Design," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 3269–3281, Aug. 2005.

[98] F. Bassi, M. Kieffer, and C. Weidmann, "Source coding with intermittent and degraded side information at the decoder," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008, pp. 2941–2944.

[99] S. Ramanan and J. M. Walsh, "Practical codes for lossy compression when side information may be absent," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 3048–3051.

[100] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1261–1271, Oct. 1996.

[101] A. A. El Gamal and E. C. Van Der Meulen, "A Proof of Marton's Coding Theorem for the Discrete Memoryless Broadcast Channel," *IEEE Transactions on Information Theory*, vol. IT-27, no. 1, pp. 120–122, January 1981.