Sparse Measurement Systems :

Applications, Analysis, Algorithms and Design

Submitted in partial fulfillment of the requirements for

the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

Balakrishnan Narayanaswamy

B.E., Electronics and Communication Engineering, M.S.R.I.T. under V.T.U.

M.S., Electrical and Computer Engineering, Carnegie Mellon University

Carnegie Mellon University

Pittsburgh, PA

May, 2011

ABSTRACT

This thesis deals with 'large-scale' detection problems that arise in many real world applications such as sensor networks, mapping with mobile robots and group testing for biological screening and drug discovery. These are problems where the values of a large number of inputs need to be inferred from noisy observations and where the transformation from input to measurement occurs because of a *physical* process.

In particular, we focus on sparse measurement systems. We use the term *sparse* measurement system to refer to applications where every observation is a (stochastic) function of a small number of inputs. Here, small is relative to the total input size. Such a system can conveniently be represented by a (sparse) structured graphical model. We study the fundamental limits of performance of these sparse measurement systems through an information theoretic lens and analyze robustness to noise, model mismatch and parameter uncertainty. We also look at these problems from an algorithmic point of view and develop practical algorithms, aided by the representation of the system as a graphical model. We analyze how the computational cost of detection with sparse measurements changes with various system parameters. Finally, we will show how to use both these analyses to *design* sparse measurement systems.

We show that, in addition to sensor parameters such as the measurement function and the noise model, which describe how the physical measurement transforms the inputs, the *structure* of the measurements critically effects the information theoretic and computational properties associated with a measurement *system*. The term 'structure' here refers to properties such as the number of times each input is measured and the pattern in which inputs are sensed - parameters that become important when information from many measurements has to be fused to reconstruct the input. The measurement 'system' extends

beyond the inputs and the measurements to include other factors such as how the inputs were generated, and how the measurements are fused to detect the input. We will show in this thesis that looking at a measurement system as a whole, in this way, leads to insights that can be used to improve system performance and to design better systems. The way we handle the correlations introduced by the fixed, sparse measurement structure and given measurement function, is what differentiates our use of the probabilistic method from other work in coding theory.

The aim of this thesis is to develop tools for designers and users of sparse measurement systems. Given the characteristics of the inputs to be detected and constrained choices of measurement types and structures, we help answer the following types of questions for sparse measurement systems

- how many measurements are required to detect the input ?
- what algorithms can be used for detection ?
- how can the algorithm be modified to suit a particular application ?
- how should the system design be changed to improve performance and robustness ?
- how do these answers change with different sensing functions, measurement structures, application requirements and detection algorithms ?

Providing an answer to these questions allows an engineer to design a sparse measurement system that meets specified performance objectives while respecting various physical constraints.

To demonstrate the impact of the ideas presented in this thesis we show results on two distinct real-world applications, by a combination of data driven modeling and simulation. We study two algorithms in some detail, the Cross Entropy (CE) method developed for

rare event simulation and the Sequential Decoding (SD) algorithm from coding theory. In the first experiment we show how to use a sparse measurement system for a group testing based drug discovery application. We demonstrate how group testing with CE decoding can reduce the amount of noise while limiting the number of tests. In the second experiment we show that a simple Infra-Red sensor can be used to detect multiple hot targets - motivated by the important problem of finding people in disaster areas or in collapsed buildings. We validate the feasibility of our SD approach in an experiment using an actual Infra-Red temperature sensor.

ACKNOWLEDGMENTS

This thesis would never have been possible without the continuous support of my family, my advisors and my friends. It gives me great pleasure to finally thank them, for all the help they have given me over the years.

First and foremost, I would like to express my gratitude to my advisor Prof. Rohit Negi for his guidance during my Ph.D. studies. His door, both figuratively and literally, was always open. The knowledge that he would be there to help me climb any obstacles in my research helped carry me through even the toughest years at CMU. I learned a lot from his rigor, insight, clarity of thought and his ability to apply all of these to multiple problems at the same time. Working out problems with him on the whiteboard was the best part of my life as a graduate student.

I was fortunate to have the support of my advisor, Prof. Raj Reddy, from the first day I entered CMU. His ability to define research directions that are both socially useful, important and scientifically interesting always amazes me. The freedom he gave me to explore different research directions and the support and advice he gave me in each and every one of them, helped make graduate school a great experience.

I will also forever be grateful for the guidance of Prof. Pradeep Khosla, who was my unofficial advisor. His support and solidarity when I changed research directions, was a great source of confidence. His advice on all aspects of my career, including what to work on, where to go once I was done and what to do there, helped make the hardest decisions much easier. Despite their busy schedule, I was honored to be one of the few people for whom Prof. Khosla and Prof. Reddy would always make my time, even if it was 8 AM on a Sunday morning.

I would also like to thank Prof. Jose Moura, Prof. Venkatesan Guruswami and Prof.

Edmund Yeh for being on my thesis committee. Taking Prof. Moura's courses at CMU first interested me in estimation and detection and led me to my research topic. Prof. Guruswami's course on coding theory taught me how to look at problems from a computer science point of view and I hope that I will be able to use this in my work in the future. Finally, the diversity of Prof. Yeh's work and his quick understanding of my research struck me from the first time I spoke to him. I would like to thank my committee for all their suggestions, which helped me focus my research after my proposal.

The best part of my CMU experience was the great friends I made here. Dr. Vyas 'Guru' Sekar and Dr. Satashu Goel were among the first students I met at CMU. Vyas was also the primary reason I was plugged into the vibrant CMU CS student community. His help 'spinning' my research talks and proposal made me forever lose faith in the research process. Satashu was my most trusted friend, and was always the one I turned to in times of need (or hunger).

Out of all the students I met at CMU, Varun 'Limji' Gupta was probably the person I interacted with the most. His shining example convinced me to take courses well outside my comfort zone, which turned out to be some of the most fun I have had in a classroom (with the instructor present). Despite all the things going on in his life, he was always there to help proof read my papers, proposal and thesis, even at the last minute, for which I am very thankful.

I was fortunate to be part of a great research group at CMU, making our group meetings the highlight of my week. Arjunan Rajeswaran was always a great example of how to make the most of graduate life, from which I consistently failed to learn. I am especially grateful for the guidance of Yaron Rachlin when I first joined the group. I would like to thank Satashu Goel, Gyouhwan Kim and Sungchul Han for the company, the random conversations over coffee, the warnings about married life, and for being great friends. The

almost daily discussions with Euiseok Hwang made for a great break from the grind. I also learned a great deal from Qiao Li, especially during the weekly meetings.

The length of my stay at CMU makes the next sentence much longer than is grammatically proper. I would like to thank Srivatsan 'Shane' Narayanan for many interesting discussions, Kaushik 'KL' Lakshminarayanan for all the help and the distractions, Vivek 'Sesha' Seshadri, Ravishankar 'G-Man' Krishnaswamy, Sivaraman 'RawT' Balakrishnan and Bodicherla Aditya 'BAP' Prakash for the many hours of foozball, Satashu Goel and Abhishek Jajoo for all the great food and even better company, Swapnil 'Tommy' Patil for rides to the airport, quizzes and showing me how to grow old without dignity, Srinath Sridhar for the many late night conversations, and finally, Hetunandan 'God' Kamisetty for more rides to the airport and for seeing to it that I was never the oldest one in the room.

I am deeply indebted to Ryan Kerekes for all the help during my qual preparations, and our very productive research discussions. I would also like to thank Thomas 'Tad' Merryman and Jason 'J-Dawg' Thornton for teaching me what it means to be an American. Our flag football victory is the third greatest non-research moment during my stay at CMU.

I would like to thank Vivian Lee for all the food during my early years at CMU and Helen Higgins for helping me stay legal in the US during my studies. I would also like to thank Elaine Lawrence and Lynn Philibin for helping me with all the technicalities involved with being a student in the ECE department.

I would also like to acknowledge the National Science Foundation (award CNS-0347455), the Army Research Office (through Cylab, CMU), the Industrial Technology Research Institute of Taiwan and the Army Research Office (award W911NF0710287) for financial support during my stay at CMU.

I would especially like to thank my wife and constant companion, Rashmi Gangadharaiah, for her support throughout my studies, for putting up with my research related mood

swings and for staying with me despite having much better things to do. This thesis would not have been possible without her and without the support of my family. This thesis is dedicated to Amma for the start, to Appa for the opportunity, to Gayathri and Gangadharaiah for giving me their most precious possession and for taking care of mine, to Rashmi for the strength to keep going, and finally, to my little Advika, for giving me a reason to finish.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Rapid advances in the material sciences and in manufacturing technologies have decreased the cost and size of thermal sensors, sonar detectors, biological micro-arrays, Micro-Electro Mechanical Systems (MEMS) and other measurement devices. There are also new measurement modalities such as robotic assays, Next-Generation Sequencing Technologies (NGST) and ubiquitous nano-sensors that can create a substantially larger number of measurements at a much finer resolution than was ever possible. Unfortunately, this decrease in cost and size often comes with an increase in noise, jitter and parameter uncertainty. In this thesis, we will explore how we can fuse noisy sources of data to reason about the state of the world, using the power of probabilistic models.

We will focus probabilistic models of large-scale *detection* problems. In these problems we have an input to be detected using many measurements. These problems are large-scale because the input could be in any one of an exponential (in the length of the input) number of possible states. In particular, we study *sparse* measurements systems for large-scale detection. In sparse measurement systems each measurement is a stochastic function of a (different) small subset of the input. We will see that these sparse measurements are of practical interest since they model many of the practical constraints inherent in detection

applications

We show that, in addition to parameters such as the measurement function and noise model, which describe how the physical measurement transforms the inputs, the *structure* of the measurements critically effects the information theoretic and computational properties associated with a sparse measurement *system*. The term 'structure' here refers to properties such as the number of times each input is measured and the pattern in which inputs are sensed - parameters that become important when information from many measurements has to be fused to reconstruct the input. The measurement 'system' extends beyond just the inputs and the measurements, and includes other factors such as how the inputs were generated and how the measurements are fused to detect the input. We will show in this thesis how looking at a measurement system as a whole, in this way, leads to insights that can be used to improve system performance and to design better systems. The way we handle the correlations introduced by the fixed, sparse measurement structure and given measurement function, is what differentiates our analysis from other work in coding theory.

In this thesis, we study four aspects of sparse measurement systems.

- **Applications :** We study how different practical problems can be modeled as sparse measurement systems by developing an understanding of the physics and application requirements in sensor networks and group testing for drug discovery.

- **Analysis :** We draw parallels between sparse measurement systems and Shannon's model of a communication system. We develop an information theoretic analysis of sparse measurement systems using the probabilistic method. This will require suitable modifications to handle the correlations that arise because of the constraints of physical measurement systems. We develop a general result that can be used to ana-

lyze the performance achievable with different kind of measurements, measurement structures and sources of noise. We also analyze the robustness of sparse measurement systems to model mismatch and parameter uncertainty at the detector.

- **Algorithms :** We study the Sequential Decoding (SD) algorithm from coding theory and the Cross Entropy (CE) Method from rare event simulation. We show that they can be applied to the problem of detection in sparse measurement systems. We analyze both the accuracy and computational properties of specific algorithms. We show that the *computational* properties of these algorithms depends on system parameters like noise level, measurement function, measurement structure, and most critically, on the number of measurements available to the detector. We combine this analysis with the theory developed to create algorithms with superior performance and low complexity.

- **System Design :** Our work has an engineering motivation, and so, we highlight the practical utility of our results. We show applications of our algorithms on real data from a thermal sensing and a group testing application. We describe methods by which the theoretical and algorithmic analysis can guide practical system design.

In this chapter we provide a flavor of our study and results.

1.1 APPLICATIONS : SOME MOTIVATING EXAMPLES

Our interest in the analysis of sparse measurement systems is motivated by new applications of group testing, such as, pooling for genetic screening and drug discovery [1] and the application of graphical models to sensor networks proposed by Moura et. al. in [2]. We first describe the model we study and then show how the model represents the physical process behind these different applications.

In this thesis, we study the general problem of detection in a sparse measurement system. An illustration of our model is shown in Figure 1.1. The fundamental problem we are interested in is that of detecting k discrete random variables (inputs) $\mathbf{V} = \left[V^1, \ldots, V^k\right]$ given n *noisy and sparse* measurements $\mathbf{Y} = [Y^1, \ldots, Y^n]$.

The graphical representation in Figure 1.1 is called a factor graph [3]. It consists of circles, one for each variable and squares representing factors. Each factor f_i is a function of the form

$$f_i(y^i, v^{\mathcal{N}_i^1}, \ldots, v^{\mathcal{N}_i^c}) \tag{1.1}$$

where $v^{\mathcal{N}_i^1}, \ldots, v^{\mathcal{N}_i^c}$ are values of the c input variables connected to the factor f_i. Each factor encodes the relationship between the variables it is connected to.

The factor graph is essentially a way to represent a joint distribution over the random variables as a product of all the factors.

$$p(\mathbf{v}, \mathbf{y}) \propto \prod_{i=1}^{n} f_i(y^i, v^{\mathcal{N}_i^1}, \ldots, v^{\mathcal{N}_i^c}) \prod_{j=1}^{k} h_j(v^j) \tag{1.2}$$

For sensing applications we start with a generic illustration of a sensor network (perhaps a thermal or seismographic sensing application) in Figure 1.2. In sensor networks the aim is to detect the state of the environment by fusing the information from noisy sensor measurements. In our model of sparse measurement system in Figure 1.1, each node V^t is a random variable that represents the state of a part of the environment. We assume that each sensor i returns a noisy estimate Y^i of a function Ψ of inputs that it measures.

Suppose that, in a thermal sensing application, the variable V^t is 1 if the t^{th} region is hot and 0 otherwise. A thermal sensor returns the (noisy) average temperature in its field of view. The factor f_i would then be,

$$f_i(y^i, v^{\mathcal{N}_i^1}, \ldots, v^{\mathcal{N}_i^c}) = P_{\text{noise}}(y^i | \frac{1}{c} \sum_{j=1}^{c} v^{\mathcal{N}_i^j}) \tag{1.3}$$

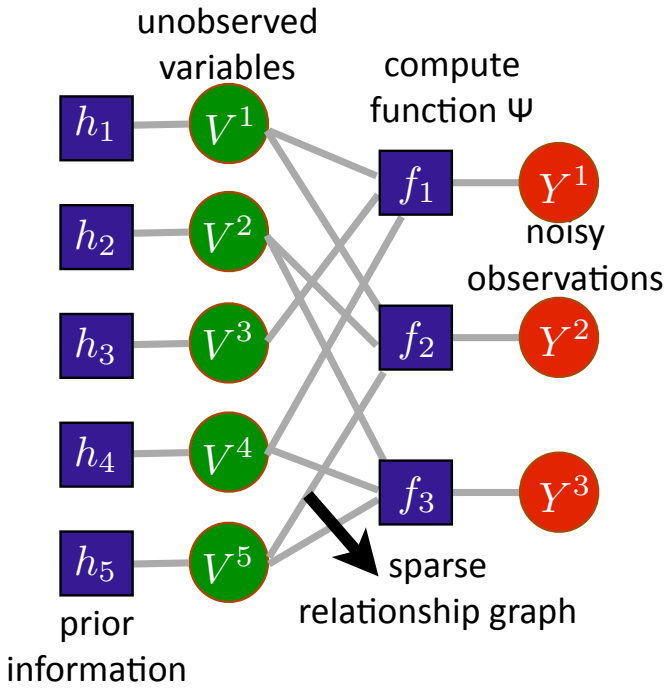


Figure 1.1: Factor graph representation of a sparse measurement system

where P_{noise} is the noise model of the thermal sensor.

Sparse measurement models are also appropriate for a large number of detection problems in the area of generalized group testing [4]. Group testing was introduced in [5] to reduce the number of tests required to detect men affected with syphilis out of inductees in the US army during World War II. Starting from blood testing the idea that grouping can reduce the number of tests required has found numerous applications [4], such as, finding faulty microprocessors [6], DNA screening [7], monitoring computer networks [8], spectrum sensing [9] and drug discovery [1].

The generic group testing problem is one of finding defective or 'positive' items among a large group of normal items. The naive method of testing each input separately would require as many tests as there are inputs. However, if the number of positives is small a significant reduction in the number of tests is possible by 'pooling' or testing groups of

Figure 1.2: Illustration of a multi sensor system

items together. In generalized group testing small pool sizes are often appropriate either because of system constraints or because they provide the best performance.

We now map this problem to our sparse measurement model in Figure 1.1. The random variable V^t is 1 if the t^{th} input is defective or 0 otherwise. The measurements Y^j are the result of different group tests. In conventional group testing, a factor f_i would then be,

$$f_i(y^i, v^{\mathcal{N}_i^1}, \ldots, v^{\mathcal{N}_i^c}) = P_{\text{noise}}(y^i | \max_j v^{\mathcal{N}_i^j}]) \tag{1.4}$$

In *generalized* group testing other measurement functions, input alphabets and general noise models are possible.

Another large class of applications where our ideas may find use is the problem of Distributed Source Coding (DSC) [10]. In these problems separated sources attempt to transmit correlated information to a receiver. The transmitted symbols are the measurements of the input (or message). The side information plays the role of the prior in our model. The receiver attempts to decode the measurements by detecting the true input. Versions of this problem are also called the Slepian-Wolf problem [11] and the Wyner-Ziv problem [12]. Parity check matrices of good codes form good encoding matrices for DSC [13]. Low-Density Parity Check codes [14] are good codes leading to sparse measurement systems. This corresponds to our model with inputs and measurements binary

($\mathcal{V} = \mathcal{X} = \mathcal{Y} = \{0, 1\}$) and each measurement Ψ is an XOR of its inputs.

The critical structural fact in all these examples is that each measurement is a noisy function of only a small subset of the inputs while the intent of the system designer is to detect the state of the entire environment. In this thesis, we use the term *sparse* measurement system to refer such applications. Accounting for this sparse nature of the measurement structure is critical to ensure that our theoretical analyses match the results of practical experiments. This will also allow our results to be used in the design of practical sparse measurement systems.

We are interested in detection problems where the hidden variables of interest are transformed by a physical process into the measurements. Understanding the dependencies between the inputs and measurements requires understanding the physics of the system. For example, the success of our experiments in Chapter 5 with thermal sensors, critically depends on developing an accurate physics-based model of the measurements, in order to understand how measurements are affected by multiple objects in their field of view [1, 15]. Modeling is an essential step required in order to translate the theory and algorithms developed for sparse measurement systems into practical, real-world results.

In addition to expert knowledge system parameters may also be learnt from data [16]. In applications such as sensor networks [15], we generally obtain initial estimates of system parameters from our knowledge about the system and where sensors are placed, but improve the estimates through calibration. In order to understand the effect of model mismatch and uncertainty that arise from using estimated models, we extend our theoretical results to these situations and also suggest some practical methods to ameliorate these adverse affects.

In this thesis, we demonstrate the practical applicability of detection algorithms for sparse measurement systems. We use measurement data from thermal sensors [15] to detect

hot targets - an experiment motivated by the real problem of detecting humans in a disaster area of collapsed building where visual sensors do not work [17, 18]. We also show results on a group testing experiment for a drug discovery application [1].

While the applications described above fit quite naturally into the sparse measurement framework, there are many other problems where similar ideas and techniques have been applied and where our work may find a place. For example, compressed sensing [19, 20] has at its core a linear measurement problem. In contrast to the compressed sensing problem, our work is for discrete inputs and outputs with general non-linear measurement functions. We focus on the effect of noise and prior information on the number of measurements required. We pursue an information theoretic, as opposed to an algorithmic approach, to the problem. We show that modified versions of algorithms designed for compressed sensing, perform well for discrete input sparse measurements with low noise [21]. Pursuing this connection further could lead to efficient algorithms for sparse measurement systems with provable guarantees, in special scenarios.

Finally, we hope that our analysis techniques, intuition, algorithms and the overall flavor of our approach would be applicable to more general problems on probabilistic models such as learning the structure of graphical models [22], observation selection [23] and constraint satisfaction [24], search [25] and optimization problems.

1.2 ANALYSIS : INFORMATION THEORETIC ANALYSIS OF SPARSE MEASUREMENT SYSTEMS

The primary theoretical focus of this thesis is the performance limits of sparse measurement systems, independent of the detection algorithm. The advantage of using the sparse measurement framework is that they can be used to model the structural features of different measurement problems. The results can then be specialized to different applications

such as sensor networks [26, 27], communication [28] or group testing [21].

We study the general problem of detecting k discrete random variables (inputs) $\mathbf{V} = \left[V^1, \ldots, V^k\right]$. Each V^j is drawn independently, identically distributed (i.i.d.) from P_V. Each position in the vector represents a discrete input at that 'location', which takes values in an alphabet $\mathcal{V}$. We are given $n = \frac{k}{R}$ (where R is the 'rate') *noisy and sparse* measurements $\mathbf{Y} = [Y^1, \ldots, Y^n]$. A higher rate R means that we have fewer measurements. A measurement u is sparse if it is a function of a small ($c = \Theta(1)$) number of the inputs V^j.

We consider the measurement function to be a composition of two functions. Each noiseless function value $X^u \in \mathcal{X}$, is a function Ψ of the inputs it measures, $X^u = \Psi\left(V^{l(u,1)}, \ldots, V^{l(u,c)}\right)$. These $l\,(u,i)$s together define the structure of the sparse measurements. Each measurement is corrupted independently according to a probability mass function (p.m.f.) $P_{Y|X}\left(Y^u = y | X^u = x\right)$ resulting in the observed measurements $\mathbf{Y} \in \mathcal{Y}^n$. An example measurement system is shown in the factor graph notation [3] in Figure 1.1.

Given the measurement $\mathbf{Y}$, we use a decoder $q : \mathcal{Y}^n \to \mathcal{V}^k$ to obtain an estimate $\widehat{\mathbf{v}}$ of the input. A fundamental parameter of interest will be the distance or distortion D between the input $\mathbf{v}$ and the reconstruction $\widehat{\mathbf{v}}$. For example, the normalized Hamming distortion between the two vectors is defined as

$$d_H(\mathbf{v}, \widehat{\mathbf{v}}) = \frac{1}{k} \sum_{i=1}^{k} 1_{v^i = \widehat{v}^i} \tag{1.5}$$

We study the problem in the large system limit as $n \to \infty$ and answer the questions, (i) 'What rates R are sufficient to detect the input to within a distortion D using sparse noisy measurements?' which we call an *achievability* result and (ii) 'What rates R are required to detect the input to within a distortion D using sparse noisy measurements?' which we call a *converse* result. Based on parallels to information theory we use a random measurements analogue of Shannon's random coding argument and derive bounds for these quantities.

The motivation for our analysis is the work of Shannon [29, 30] where most of the problems in information theory and their solutions [1] can be found. His analysis was the first to show that it is possible to transmit information *reliably* even in the presence of noise, at rates below channel *capacity*. He gave sharp bounds on the system parameters under which this is possible. While the information theoretic bounds were non-constructive and derived in the large system limit, they laid the foundations for the digital communication revolution.

Achievability results, which are lower bounds on channel capacity in communication theory, tell us first of all that reliable communication is possible across noisy channels, something that was not known before the work of Shannon. Together with converse results, which are upper bounds on capacity, they give us a good idea of how sub-optimal any particular practical coding or decoding scheme is. This allows us to focus on improving systems in situations where there is a large performance gap to the bounds. In addition, the bounds give us some intuition into the properties of good codes and demonstrate that larger message and code lengths can be used to improve performance.

In this thesis we will try to develop a theory to similarly guide the design of sparse measurement systems, continuing the work of [31, 32]. We do this by defining the 'Sensing Capacity' that plays the role of channel capacity for sparse measurement systems. We analyze bounds on the sensing capacity for different measurement structures and system parameters. These bounds result in estimates of the number of measurements required for reliable detection from sparse measurements. Our results will be different from the information theoretic results for communication because of the special constraints present in sparse measurement structures. Showing how to handle the correlation between measurements, induced by the fact that they arise from a fixed sparse measurement structure and

[1] To quote David Forney, information theory was "created almost fully grown from the brow of Zeus"

given measurement function, is the primary theoretical contribution of this thesis. It also differentiates our work from many other applications of the random coding method in coding theory. We will also show how the analysis can be used to guide and inform the design of sparse measurement systems.

While information theoretic analysis of models similar to our sparse measurement systems is fairly recent [21, 26, 33] the general analysis methods we use have a rich history in probability theory [34], information theory [29] and combinatorics [35]. In addition each of the application areas we mentioned in Section 1.1 has a large literature, including theoretical analysis. One of the most important theoretical contributions of this thesis is the generality of our final result that can be applied to many of these special cases of interest.

We now describe some of the prior work, related to our the work in this thesis, in some of these different application areas. We begin with the group testing problem [4] where the objective is to identify a small number 'positive' elements from a much larger set. The basic result in group testing is that the number of tests required can be reduced substantially by testing groups of elements together. In our model this corresponds to the case where each measurement is the Boolean OR of a subset of the inputs.

There has been a lot of theoretical work on the basic 'OR' model of group testing - both information theoretic [36, 37] and combinatorial [4]. Sparse pools, which correspond to our sparse measurements, have been shown to be useful for group testing [7]. We will analyze the performance of sparse 'regular' designs, where every input is measured the same number of times. These structures are motivated by the need for fair coverage in some group testing problems and the excellent performance of LDPC code structures developed in coding theory [38]. We will also extend the group testing model to more general measurement functions, noise models, priors and input alphabets. Our focus will be on obtaining the most general results possible, which will allow us to analyze new sparse group

testing designs for different applications [39] that do not fall into the classical group testing framework. We study the effect of noise, which is usually an after thought in the usual group testing work [4, 36] (though there are a few exceptions [40, 41]).

Compressed sensing [19, 20] is a linear measurement problem that has received much attention in the recent past. There has been work on information theoretic analysis [42] of the compressed sensing problem, especially from a graphical models learning framework [43]. In addition, sparse designs have received recent attention [44, 45]. In contrast to this line of work we consider more general (possibly non-linear) measurement functions and focus on the effect of using different sparse measurement structures and different sources of noise on the performance. We also consider situations that arise in some group testing problems, where the sparsity of the measurements is actually required to improve performance.

Keeping with the practical focus of our work and motivated by our experiments with real sensor network and group testing applications, we also study the loss in performance of sparse measurement systems when there is model mismatch or parameter uncertainty at the detector. In sensor networks, this could be the result of drift in sensor parameters [46], calibration errors [47] or manufacturing defects [48]. In group testing problems, sources of mismatch could range from uncertainty in the concentrations and re-activities of the chemicals being tested to approximations made to the pooling measurement function [49]. We should not expect to have perfect knowledge of measurement functions and noise models, and so, understanding the effects of model mismatch and parameter uncertainty is worthwhile. In our analysis we borrow from [50] who looked at similar problems in the framework of communication. We, however, have to make modifications to account for the sparsity of and correlations between sparse measurements.

Finally, in this thesis we also develop a converse that bounds the minimum number of

measurements required to reconstruct the input to within a given fidelity. This converse is based on standard information theoretic techniques. We will see that the converse rates are sometimes close to the achievability bounds, indicating the tightness of our analysis in some scenarios.

1.3 ALGORITHMS : DETECTION IN SPARSE MEASUREMENT SYSTEMS

Given a general factor graph - such as the one in Figure 1.1 - inference is the process of reasoning about hidden nodes or variables ($\mathbf{V}$) in a graphical model given the observed information ($\mathbf{Y}$). There are two kinds of inference we are usually interested in - Marginal Probabilities (MP) and Maximum A-Posteriori (MAP). In the first case we wish to find the distribution of a hidden variable conditioned on the observations:

$$p(v^i|\mathbf{Y}) = \sum_{\mathbf{v}'=\mathbf{v}\setminus v^i} p(\mathbf{v}|\mathbf{Y}) \tag{1.6}$$

In MAP or Most Likely Explanation (MLE) inference we seek an assignment to the hidden variables that maximizes the conditional probability, that is we are looking for a $\mathbf{v}_*$, such that

$$\mathbf{v}_* = \arg\max_{\mathbf{v}} p(\mathbf{v}|\mathbf{Y}) \tag{1.7}$$

We will focus on MAP inference, which we call detection, for most of this thesis.

One advantage of the factor graph framework is that we can draw on the large body of work on inference algorithms in graphical models while designing detection algorithms for sparse measurement systems. Inference algorithms [3, 51] can be exact, like the Variable Elimination algorithm, or approximate. It has been shown that exact inference in general graphical models is NP-Hard [52]. Known algorithms for exact inference have computational complexity exponential in the tree-width of the graphical model. Since many of the sparse measurement structures we look at have high tree-width [53] we focus on approxi-

mate inference algorithms in this thesis.

Approximate algorithms for inference in graphical models fall into four classes - local message passing algorithms like belief propagation (BP) [54], search algorithms like A* search [55], sampling algorithms like Gibbs sampling [56] and variational algorithms [57]. Each of these classes of algorithms has been studied extensively in the machine learning and coding theory communities [58] and there exists a good understanding of conditions under which performance of specific algorithms like BP perform well [59]. For other algorithms this understanding is less well developed and those will be the focus of our work.

In this thesis we will look at two detection algorithms in some detail and mention others as appropriate. One algorithm we look at is a search-based algorithm developed in the coding theory literature called Sequential Decoding (SD) [60]. SD was first suggested for sensor networks by [61]. We demonstrate the practical applicability of SD in a thermal sensing application [15]. We focus on SD because it has interesting computational properties [62], in particular, the existence of a computational cut-off rate. Computational cut-off is a phenomenon where increasing the number of measurements above a threshold *substantially* reduces the computational complexity of an algorithm. This suggests interesting trade-offs for sparse measurement systems with limited computational resources. We rigorously analyze the computational complexity of SD and demonstrate the existence of the computational cut-off for sensor network applications.

The second algorithm we look at is called the Cross Entropy (CE) method which was developed for rare-event simulation, but later found applications as a general Monte-Carlo approach to many different optimization problems [63]. Our theory suggests a stopping criterion for the CE method that is shown to be useful in practice. We show the applicability of this algorithm on data from a group testing experiment for drug discovery from [1]. In addition, we demonstrate that the CE method displays a computational cut-off phenomenon

similar to SD.

1.4 SYSTEM DESIGN : FROM THEORY TO PRACTICE

An engineer designing a measurement system has a number of decisions to make, such as, what kind of sensing devices to use, how to deploy them and how to fuse the measurements to extract the important information. We believe that *information theory is a rich source of answers to questions about sparse measurement systems : theoretical, algorithmic and, perhaps most importantly, practical*.

However, while borrowing intuition from the theory of communication, this thesis also aims to understand how measurement systems are different from communication systems, where the encoder is under the control of the system designer. The focus of our work is on how to model, analyze and fuse information from measurements that are constrained because of costs involved or the physics inherent to a system. For example, we study *sparse* measurement systems because this sparsity was inherent in the applications from Section 1.1. Our experiments with thermal sensor networks and group testing for drug design led us to the conclusion that model mismatch and parameter uncertainty are important aspects of practical measurement systems. Based on this observation, we study the effect of uncertainty and mismatch, both from a theoretical, as well as an algorithmic viewpoint.

Imposing constraints on the measurement systems also has important algorithmic implications we wish to study. An understanding of the effect of sensing functions, noise models and measurement structures on the performance of different algorithms allows us not only to choose the right algorithm for a particular sensing task but also design measurements that are 'easy' to decode. This line of work also differentiates this thesis from other work in coding and communication and also highlights our focus on practical system design.

Based on the above discussion the aim of this thesis is to build tools for users and designers of real sensing systems by,

- Developing an information theoretic understanding of measurement structures when practical constraints are incorporated
- Studying the performance of different algorithms when applied to detection when measurements have inherent constraints
- Showing the applicability of our model, theoretical results and algorithmic analysis with experiments with real sparse measurement systems
- Using the analyses to design measurement systems that have good performance, are easy to decode and are robust to noise, mismatch and uncertainty.

1.5 CONTRIBUTIONS

We now summarize the primary contributions of this thesis.

Theory

- For a sparse measurement model, we developed a general expression for the number of measurements required to reconstruct an input with a required fidelity, by specifying a lower bound on 'sensing capacity'. The expression derived shows explicitly the dependence of number of measurements required on the measurement structure, noise model, prior and measurement function for different measurement structures. In particular, we derived lower bounds on the sensing capacity of sparse regular measurement structures and showed a good match between the bounds and performance of practical algorithms.

- Motivated by practical considerations we derived expressions for the performance of sensing systems with model mismatch and parameter uncertainty.

- We derived a converse that showed fundamental limits on the performance of particular sparse measurements. This converse is shown to be tight in some special cases.

Algorithms

- Compared the performance of algorithms for detection in sparse measurement systems like Belief Propagation (BP), the Cross Entropy (CE) method and Sequential Decoding (SD). We demonstrated that the theoretical analysis is useful in identifying the sub-optimality of practical algorithms and that the CE method shows performance that follows the trends indicated by the theoretical bounds.

- Demonstrated the 'computational cut-off' phenomenon displayed by the CE method where the algorithm becomes computationally efficient once a sufficient number of measurements are collected. This demonstrated that collecting additional measurements can alter the trade-off between accuracy and computational complexity resulting in algorithms that are both accurate and have low complexity. We also showed that the SD algorithm also has a similar computational cut-off for sensor networks and empirically demonstrated this for a thermal sensing system. We analyzed SD to make the case that collecting more measurements can reduce the computational complexity of algorithms.

Applications

- Showed the applicability of the CE method in a group testing experiment for drug discovery.

- Demonstrated the practicality of SD with experiments using real thermal sensor measurements.

System Design

- Used the theoretical analysis to outline how diverse measurements can make a sensing system robust to mismatch and uncertainty.
- Suggested theoretically justified modifications to the sequential decoding algorithm that improve performance for sensing applications.
- Showed how to modify the measurement functions and pre-process data in a sparse measurement system to improve the computational properties of its decoder.

CHAPTER 2

APPLICATIONS OF SPARSE MEASUREMENT SYSTEMS

In this chapter we motivate the study of sparse measurement systems by describing a few applications of the model. We focus on two distinct classes of applications, group testing for screening applications and sensor networks for detection and monitoring. We also outline some other relevant applications. We first explain the abstract model and then show how it maps to the different applications. We will return to these applications in Chapter 5 and show how the methods described in the intervening chapters can be used to design systems for real-world problems.

2.1 WHAT ARE SPARSE MEASUREMENT SYSTEMS?

The problem of detecting a discrete vector from noisy measurements appears in many different guises in signal processing and information theory. Some examples include, the problem of communicating over a noisy channel, group testing [4], communicating across a multiple access channel [64] and detection in sensor networks [65].

In this thesis we are interested in the study of *sparse* measurement systems. Each *sparse* measurement is a function of a small number of the variables to be detected (a

formal definition is provided later). Sparse measurements are studied because of their performance, because of real world constraints or because they are associated with efficient detection algorithms. For example, Rachlin et. al. [26] model sensor measurements as being sparse since real sensors have limited field of view. Guo et. al. [66] suggest the use of sparsely-spread Code-Division Multiple Access (CDMA) for multi-user detection because they can be decoded efficiently. Bruno et. al. [7] used 'low-weight pools' for group testing, where each pool is the mixture of very few samples, because they have better performance. Montanari et. al. [33] suggest the use of sparse statistics for lossless compression because they have a smoothness property (changing one input changes only a few measurements) which is important in some data storage applications.

We will see that, in addition to system parameters such as the measurement function and noise model, that describe how the physical measurement transforms the inputs, the *structure* of the measurements critically effects the information theoretic and computational properties associated with a measurement *system*. The term 'structure' refers to properties, such as, the number of times each input is measured and the pattern in which inputs are sensed, which become important when information from many measurements has to be fused to reconstruct the input. The measurement 'system' includes other factors such as how the inputs were generated and how the measurements are fused to detect the input. We will show in this thesis how looking at a measurement system as a whole leads to insights that can be used in system design.

The study of sparse measurement systems is ongoing and can be approached from several different directions. Montanari [67] studies the performance of message passing algorithms [3] for detection with sparse measurements using density evolution [58]. Guo et. al. [66] analyze the problem of decoding sparsely spread CDMA from a statistical physics perspective. In [33], lossless compression using sparse noiseless measurements with an

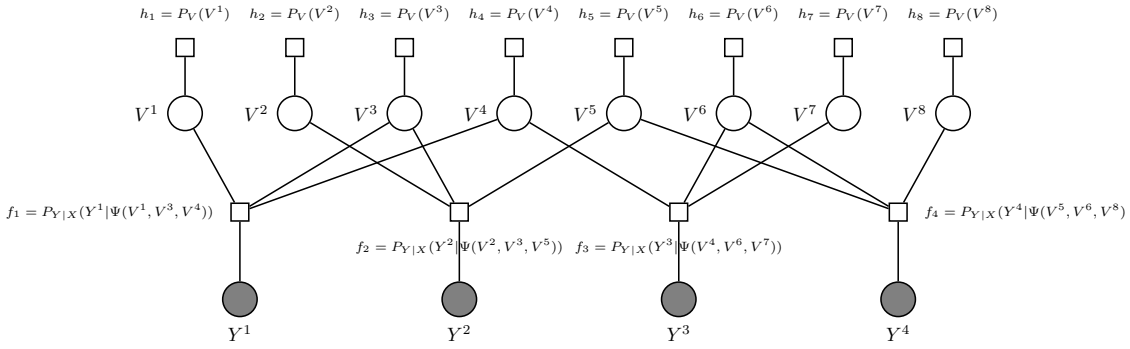


Figure 2.1: Factor graph of detecting variables from sparse measurements with $c = 4$, $d = 2$, $k = 8$, $n = 4$, and the rate $R = 2$. f_u are measurement factors and h_j factors correspond to prior information.

optimal decoder is studied and analyzing the effect of measurement noise is left as an open problem.

In this thesis, we study the problem of detecting k discrete random variables (inputs) $\mathbf{V} = \left[V^1, \ldots, V^k\right]$. Each V^j is assumed to be drawn i.i.d. from a distribution P_V. Each V^j represents a discrete input at 'location' j, which takes values in an alphabet $\mathcal{V}$. We are given n *noisy* measurements $\mathbf{Y} = [Y^1, \ldots, Y^n]$. Each measurement Y^u is a stochastic function of c inputs $Z^u = [V^{l(u,1)}, \ldots, V^{l(u,c)}] \in \mathcal{V}^c$. We can decompose this function into a composition of two functions. The first is a noiseless function value $X^u \in \mathcal{X}$ computed as $X^u = \Psi\left(V^{l(u,1)}, \ldots, V^{l(u,c)}\right) = \Psi(Z^u)$. Then, each X^u is corrupted independently according to a probability mass function (p.m.f.) $P_{Y|X}\left(Y^u = y | X^u = x\right)$ resulting in the observed measurements $\mathbf{Y} \in \mathcal{Y}^n$. The set of $l(u, i)$s is called the measurement structure $\mathcal{C}_n$. This can be represented as a factor graph [3] as shown in in Figure 2.1. The goal is to detect the input $\mathbf{V}$ from the measurements $\mathbf{Y}$. In this thesis we study sparse measurement structures, which means that $c = \Theta(1)$. Thus, each measurement Y^u is a function of a small number of variables V^j.

We now describe how the sparse measurement systems defined above are a good model

for some applications.

2.2 GROUP TESTING FOR HIGH THROUGHPUT SCREENING

The first application of the sparse measurement model that we look at is the problem of group testing for high throughput screening [1]. **Group testing** [4] is an old technique that was used to reduce the number of blood tests required to detect syphilis in soldiers during World War II. The blood from multiple people is mixed together to form a 'pool' that is tested. If any of the soldiers in that pool are 'positive' for syphilis the sample will test positive. This pooling was shown to substantially reduce the number of tests required [5]. To fit this application in our model in Figure 2.1, we associate a random variable V^j with each solider j, which takes a value 1 if he is infected and 0 otherwise. The function Ψ is the Boolean OR since the output of a group test is positive if any sample in the pool is positive. Sources of errors such as human or machine error and chemical inhibitors, can be modeled by the noise model $P_{Y|X}$.

While pooling for blood screening is still relevant today [68] (and may be useful again in the future [1]), similar problems have been studied for a variety of applications including DNA clone library screening [7], Protein-protein interaction mapping [69], high-throughput drug screening [1], genetic screening [70], retrieval [71] and circuit testing. A number of these new applications in biological screening do not fit the conventional group testing framework because of non-binary inputs, different measurement functions and unique noise models. We describe a couple of these new applications in detail and show how our more general model is appropriate for these applications as well.

[1] See "Human or Cylon? : Group testing on Battlestar Galactica" by Christopher R. Bilder, http://statistics.unl.edu/faculty/bilder/grouptesting/Mizzou.pdf

2.2.1 Drug discovery

Kainkaryam [1] described the use of pooling strategies for drug discovery, using a detailed chemical model of pooling. This modeling was done to improve the efficiency of High-Throughput Screening (HTS). HTS is a scientific experimentation process by which thousands or even millions of biochemical, genetic or pharmacological tests are conducted in a short time frame. Advances in biology and chemistry continuously produce increasing number of molecules that may be suitable for therapeutic intervention. This combined with the methods for combining multiple sources into a single reagent has meant that traditional high-throughput screening (HTS) has had to evolve into ultraHTS (uHTS) [72] in order to screen available targets. This is usually done through miniaturization and the use of robotics, although this alone may not be sufficient [73].

Another way to improve the efficiency of HTS is group testing, where pools of compounds are tested. Group testing (or pooling) has been shown to reduce the number of tests required if very few of the inputs are active [5]. Group testing has received much attention in theoretical computer science [74], and more recently signal processing [75]. However, most of the work is for the Boolean OR measurement model and sub-linear number of positives in the input at the low noise limit. These aspects of the model need to be generalized in order to study the applications we are interested in.

We start by describing the physics underlying the pooling process. We then build a model for the pooling process. We consider scintillation proximity assay (SPA) [76] as a representative technique for detecting chemical reactivity for drug discovery applications. The model for SPA type pooling should be applicable with small changes to other chemical and biological HTS problems. SPA uses a fluorescent bead (that emits light when stimulated) that can anchor with the target in the reaction. If, because of the pooling process, the reagents in the pool occupy some of the anchor points, then they will not be available

to anchor the fluorescent markers, resulting in a decrease in the signal - a process called inhibition [1].

Suppose I_{MAX} and I_{MIN} are the maximum and minimum possible intensities that a test can produce. If $V^1, \ldots, V^c$ are the association constants of the c chemicals mixed in a pool then the intensity from a pooled test Y_{TEST} can be approximated as

$$Y_{TEST} \propto I_{MIN} + (I_{MAX} - I_{MIN}) * \frac{L}{L + \sum_a V^a} \tag{2.1}$$

where the proportionality constants and L depend on the system parameters such as the ambient temperature and the concentrations and re-activities of the reagents used in the test. Since we are interested in detection problems we consider the case where V^a is 0 or 1, corresponding to each chemical being inactive or active, respectively.

This physics based model can be mapped directly to our sparse measurement model in Figure 2.1. The input alphabet is $\mathcal{V} = \{0, 1\}$, with 1 corresponding to active and 0 corresponding to an inactive chemical. The length of input k is the number of chemicals to be tested and n is the number of pools. The fraction of chemicals we expect to be active represents the prior, and the measurement noise is due to failed tests, as well as errors in the florescence detector. Assuming that the amount of each chemical pooled is the same, larger pool sizes lead to lower florescence levels, with the noise level essentially constant since it is due to shot noise in the detector. The actual noise is signal dependent and larger pools lead to more noise in the output. We will describe the noise model in more detail in Section 2.2.3. Different noise models are possible, and our analysis will be general enough to study them.

2.2.2 Genetic screening

Another application of group testing that has recently received some attention is the use of pooling for genetic screening of carriers of disease causing mutations. In order to explain this problem and develop a model, we outline the fundamentals of the genetic theory of inheritance. For more details, see [77].

A gene is a unit of heredity in a living organism. It consists of a region of DNA that codes for a protein or RNA chain that has a particular function in an organism. Genetic disorders occur when there is a mutation in the DNA region corresponding to a gene. These variants of a gene are called alleles. While many variants are harmless, such as the mutations that cause variation in human eye-color, others result in serious diseases such as Sickle Cell Anemia and Color-blindness.

Humans are diploid, since they carry two copies of every gene. Dominance is the preferential expression of one variant (also called an allele) of a gene over another variant of the same gene, when both are present in the organism. Many disorders, such as Cystic Fibrosis and Sickle Cell Anemia, are recessive (the opposite of dominant). The disorder is manifested only in individuals that have two non-functioning copies of a gene. Organisms that have one non-functioning copy are called carriers and do not display any symptoms of the disease. However, if two carriers have mutual offspring, there is a 25% chance that the offspring will suffer from the disease. So, it may be of interest to screen populations to find carriers of prevalent diseases, as suggested by studies, such as, [78, 79].

We are interested in sparse measurement structures as models for the design of pooling designs to screen for rare genetic variants [80, 81]. Many disorders (e.g, sickle-cell anemia [77]) are caused by changes in specific locations in the genome and can be detected by sequencing small parts of the genome. Since the prevalence rate of these alternate alleles is low, it is possible to pool together samples from many individuals and sequence the pools

to reduce the number of tests required. The results from the pools are then used to detect carriers and affected individuals [80]. This is reminiscent of the group testing problem [4], except that the input alphabet is ternary (as opposed to binary in the group testing problem) and that the measurement function is a weighted sum (as opposed to a binary OR).

According to the Hardy Weinberg principle, both allele and genotype frequencies in a population remain constant (in equilibrium) from generation to generation [77]. This means that for most diseases there is a constant fraction of people who are carriers. This gives us the priors P_V for our model, and is different from the conventional group testing problem, where some fixed number of inputs are active (as opposed to a fixed fraction). This leads to very different asymptotics. For constant fraction of positive inputs and non-zero noise level, perfect reconstruction may not be possible, unlike the claims of conventional group testing. In addition, a constant pool size c will be the optimal even with growing number of inputs n as long as the fraction of positives p is fixed. Thus, sparse measurement structures are optimal, unlike the model assumed by conventional group testing. Our model is designed to account for these effects and this again highlights the applicability of the model we study, since we focus on sparse measurement structures with constant c.

Parallels have also been made between pooling for genetic screening and the problem of compressed sensing in [81]. However, inputs in the compressed sensing problem are continuous and the suitability of the pooling matrices for compressed sensing algorithms is hard to test. In addition, the peculiar noise model that arises in Next-Generation Sequencers (NGS) is hard to incorporate into the analysis and algorithms related to compressed sensing. Our model is a better match to the genetic screening applications since it allows for general measurements, over any discrete alphabet with arbitrary noise models.

2.2.3 Noise models for high throughput screening

In the applications of high throughput screening in Sections 2.2.1 and 2.2.2 it turns out that a similar noise model is applicable. We call this the reads noise model, and explain it with respect to a genetic screening application. In order to understand the basis for the noise model we explain the underlying physics of a genetic sequencer.

There are a few different commercially available Next-Generation Sequencing Technologies (NGSTs) [82]. These aim to sequence a subset of the DNA strands in the input sample. Consider a pool created by mixing the blood/DNA of c individuals. Depending on the number of people who have the mutation, the fraction of mutant DNA samples in the pool can take $c+1$ values $X \in \mathcal{X} = \{\frac{0}{c}, \frac{1}{c}, \ldots, \frac{c}{c}\}$.

Suppose that in a particular pool $x \in \mathcal{X}$ fraction of the mutated DNA sample is present with $1-x$ fraction of normal DNA. The NGST device draws strands from this mixture to sequence. The output of each sequencing run is called a read. Suppose the sequencer takes r reads. Then, the probability of each (noiseless) read being mutated is x. The probability that y out of the r reads are mutated is then,

$$P_{Y|X}(y|x) = \binom{r}{y} x^y (1-x)^{r-y} \tag{2.2}$$

Here y represents the number of output reads (out of a possible r reads) that are mutant. The output alphabet $\mathcal{Y} = \{0, 1, \ldots, r\}$.

While the reads model accounts for the noise due to concentration effects we must also include the probability that a test may fail due to preparation error or detector error. In the screening application, this could happen because of imaging errors when the output of the fluorescence detector is analysed, noise in the detector, or errors with pooling machinery. We model this with a failure probability, p_f, by assuming that the output is uniform across

all possible values when a failure occurs, resulting in the following mixture noise model.

$$P_{Y|X}(y|x) = (1-p_f)\binom{r}{y}x^y(1-x)^{r-y} + p_f\frac{1}{|\mathcal{Y}|} \tag{2.3}$$

where p_f is the test failure probability. In addition to the possible reduction in the number of tests, an added advantage for group testing is that the pools can be designed to be robust to failure noise.

We will see that for a fixed number of reads r and a given prior probability, there is an optimal pool size which is a constant c. This further emphasises the need for sparse measurement structures. For example in the reads model, non-sparse measurement structures have rate $R \to 0$ as c grows. Thus, for applications with this kind of physical process underlying them, sparse pools are actually *necessary*.

2.3 SENSOR NETWORKS

Advances in technology have created sensor network applications with new sensing modalities and novel constraints. The application of the sparse measurement structures to sensor networks, as studied in this section, is motivated by the results of [15, 31, 83], who studied a 'large scale' detection problem. In these problems, each measurement Y^u is a noisy function of a subset of the environment. Since measurements correspond to physical sensors, such as sonar sensors [84] or thermal sensors [18], we have to model the *spatial* nature of these sensing modalities. By spatial, we mean that a sensor measurement is a function of locations that are close by or contiguous in a spatial sense. We will focus on algorithms for such contiguous measurements to complement the theory developed in [32].

2.3.1 Object detection using thermal sensors

We study object detection with thermal sensors, motivated by the problem of detecting people in collapsed buildings or in darkness, where pictures and video are not suitable [85]. In such situations, it may be possible to detect humans because of their thermal signatures [17]. However, in order to distinguish humans from other sources of heat, such as fires or small animals, we must reconstruct the shape of the hot object given low-resolution thermal measurements. In this application scenario the robots and the sensors should be cheap and easily replaceable because it is possible that they could be destroyed during rescue operations. We are interested in how we can detect the shape of hot objects given the noisy, low-resolution output of cheap thermal sensors.

The IR temperature sensor modeled in our simulation and used in our experiments is the Melexis MLX90601KZA-BKA[2]. To motivate our sensor model, we briefly discuss the physics associated with this type of sensor. A common way to understand the radiation characteristics of an object is to model it as a black body. A black body is an object that absorbs and emits radiation in all wavelengths, and whose emission curves are completely determined by its temperature. The Stefan-Boltzmann law governs the total power per unit area radiated by a black body, J, which is given by,

$$J = \sigma T^4 \tag{2.4}$$

where σ is the Stefan-Boltzmann constant and T is the object temperature.

The distribution of radiation power emitted as a function of wavelength depends on its temperature. Objects at 400°C emit visible radiation. Lower temperature objects emit more power over longer wavelengths. Hence, to measure the temperature of objects at lower temperatures, we have to use sensors that are sensitive to much longer wavelengths.

[2] http://www.melexis.com

IR sensors respond to incident radiation in the IR spectrum, and hence can sense radiation from objects with lower temperatures. For example, the MLX90601KZA-BKA sensor is sensitive to temperatures from -40°C to +125°C.

The amount of energy that arrives at the sensor is a function of its field of view. In our simulation, we vary the sensor's field of view. The sensor receives different fractions of energy based on the angle between the target and the centerline of the sensor's field of view. An example of the amount of energy sensed as a function of incidence angle is shown in Figure 2.2 for a sensor with a field of view 26°. A target that is directly in front of the sensor generates a higher reading than a target at the same temperature not centered in the field of view. We model the sensor as a set of weights, one for each grid block in its field of view. The weights depend on the orientation of the sensor and the angle between its center and each grid block. The sensor weights are normalized, and are used to average the temperatures of the objects in the sensor's field of view to produce a weighted average temperature T_{av}. Thus, the power reaching the IR sensor is proportional to T_{av}^4.

However, the sensor itself generates radiation. Assuming the sensor is at the ambient temperature of the environment T_{am}, the sensor emits radiation with power proportional to T_{am}^4. Therefore, we model the output signal of the sensor as proportional to $T_{av}^4 - T_{am}^4$. The sensor's output is noisy, and to model this we corrupted the sensor model's output with zero-mean Gaussian noise, whose variance is estimated from real sensor data.

We do not model effects such as diffusion of heat from the target to the environment and we assume that the targets are ideal Lambertian sources that radiate energy equally in all directions.

This can also be mapped to our general factor graph in Figure 2.1. Each grid location to be sensed can be either binary - corresponding to hot or ambient temperature - or different values corresponding to the temperature of other possible targets. The measurement func-

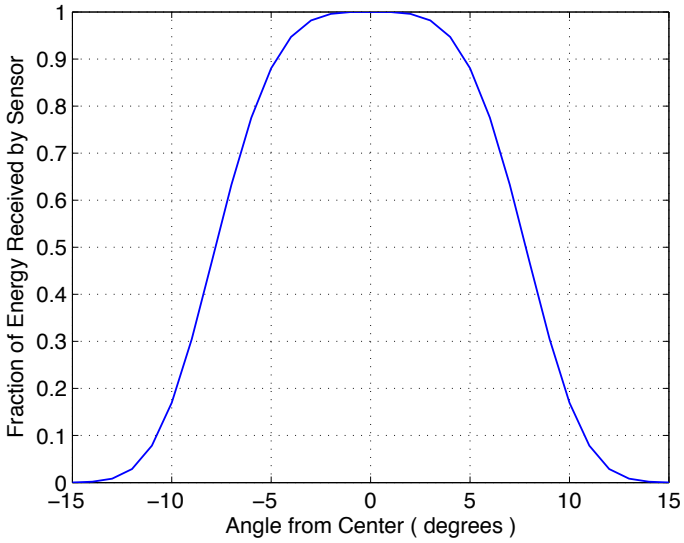
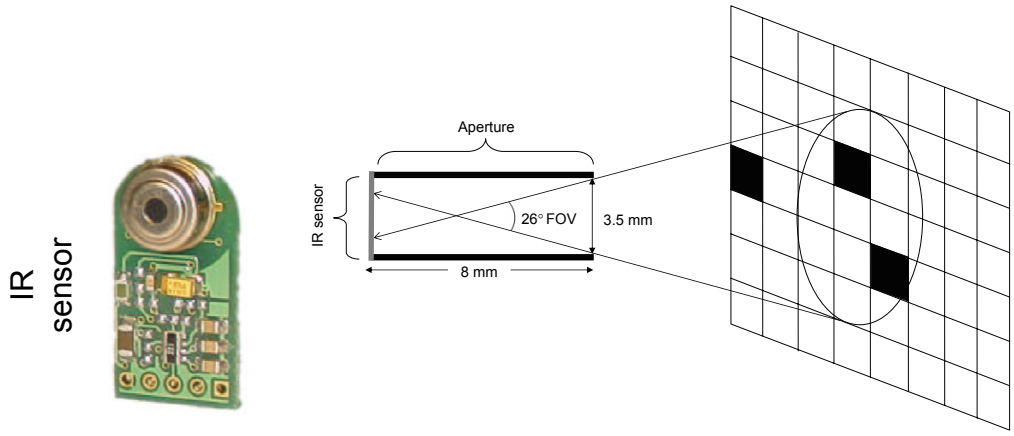


Figure 2.2: Fraction of energy received by sensor as a function of incidence angle.



(a) Thermal sensor

(b) An illustration of the sensor application. An IR temperature sensor senses a discrete temperature field. The sensor field of view is limited using an aperture. White grid blocks are at ambient temperature and black grid blocks are at a higher target temperature.

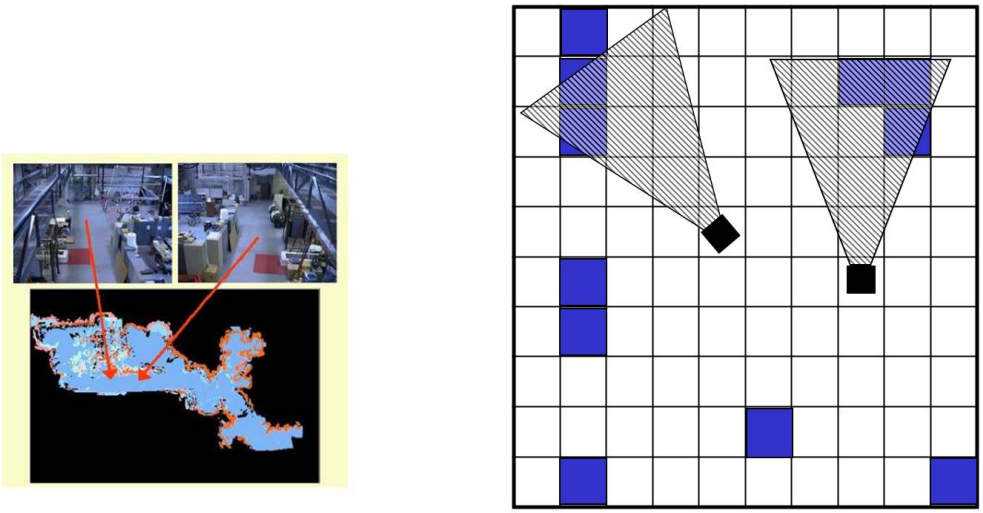
Figure 2.3: Detection with measurements : Thermal Sensor

tion is given by a combination of the thermal laws described above with the sensitivity of the sensor to incident energy at different angles of incidence as shown in Figure 2.2. This measurement function is a weighted sum of the inputs in the field of view. The noise is assumed to be Gaussian since it is mostly due to shot noise in the detector.

2.3.2 Mapping with sonar sensors

Another sensor network application where measurements have a contiguous nature arises in mapping applications. For example, in robotic mapping [86] a robot collects sensor measurements to construct a map of the environment. The robot for example, may have access to distance-to-obstacle measurements from a sonar sensor and we would like to use these measurements to map the surroundings of the robot, as shown in Figure 2.4(a). The environment can be modeled as a binary vector with 0 indicating no obstacle, and 1 indicating the presence of an obstacle. Each measurement gives us some (noisy) information about a subset of the environment. We detect the state of the environment by combining the information from these measurements.

This problem can also be mapped to our model from Figure 2.1. There are k inputs, corresponding to the size of the area to be mapped, suitably discretized. The inputs are binary $\mathcal{V} = \{0, 1\}$, corresponding to a grid block being occupied or not. Larger alphabets are also possible if we want to distinguish occupied locations based on the reflectivity of the material. The measurement function Ψ is the distance to the nearest occupied block in the cone of view of the sonar detector. Noise could occur due to unknown reflectivity of the surface or because of angled reflections failing to return to the sonar detector. See [87] for more details on the different parameters in the sonar noise model.



(a) Mapping the environment using robots with sonar sensors

(b) Idealized model

Figure 2.4: Detection with measurements : Robotic mapping

2.4 OTHER APPLICATIONS

We list a few other applications that use sparse measurement structures and map them to Figure 2.1, to further motivate our line of study.

- **Multi-user detection** [64] is the problem of decoding the transmissions of multiple users who are using a shared channel resource. In general, the channel can be modeled as a vector channel $\mathbf{H}$, with $\mathbf{Y} = \mathbf{HV} + \mathbf{N}$. In the case of binary CDMA the alphabet of the V^j variables to be estimated is $\mathcal{V} = \{1, -1\}$. Each output Y^u is then a *linear* function of the discrete V^js. This falls into our framework when $\mathbf{H}$ is row sparse, as is the case for sparsely spread CDMA [66]. Here, the output at each Y^u is a linear function of only a few of the binary V^js. The noise model $P_{Y|X}$ is usually assumed to be Gaussian, modeling the additive noise in the channel.

- **Luby Transform (LT) codes** [88] are low-density generator matrix (LDGM) codes that are universal and rate-less erasure codes. They are useful in many network cod-

ing applications. In an LT code, each encoding symbol chooses a degree c from a degree distribution. It then chooses c message bits as neighbors. The encoded symbol is then the XOR of the chosen bits. We map this into the model we study as follows. The discrete V^js are the 'message bits' in the LT code and the function Ψ is XOR. The noise model $P_{Y|X}$ is the erasure channel over which we use the LT codes. $\mathcal{Y}$ is a tertiary alphabet $\mathcal{Y} = \{0, 1, *\}$, with $*$ used for erasures.

- **Distributed sparse random projections** were suggested in [89] to store the information collected by a sensor network in a resilient and refinable manner. They suggested the use of sparse random projections to compress real valued information, using ideas from the theory of compressed sensing [19]. Sparse random projections have similarly been suggested for finding distances between pairs of points for database information retrieval by [90]. In the discrete case the data points correspond to inputs in a discrete alphabet $\mathcal{V}$ with linear (in the appropriate field) measurements with no noise and no allowed distortion (since we require perfect reconstruction). Our results may be useful when sparse 'measurements' are used and errors are introduced during storage or transmission, such as in the problem of joint source channel coding.

CHAPTER 3

ANALYSIS OF SPARSE MEASUREMENT SYSTEMS

In the previous chapter we saw some applications of sparse measurement systems that serve to motivate a number of theoretical problems for study. The analysis of systems that have sparse dependencies arise in a number of other areas including statistical physics [91], artificial intelligence [51], coding theory [14, 92] and optimization [93]. We will draw on ideas from these areas to study the fundamental limits of performance of sparse measurement systems.

Our information theoretic analysis characterizes asymptotic properties of the sparse measurement system as system size grows using ideas from coding theory, combinatorics and large deviations. We will use techniques developed for understanding the information capacity of channels to analyze the 'Sensing Capacity' of sparse measurement systems. We characterize how the performance of the sparse measurement system changes with the structure of the measurements, the noise model and the measurement function. We will also use information theoretic techniques to develop a converse. Showing how to handle the correlation between measurements, induced by the fact that they arise from a fixed sparse measurement structure and given measurement function, is the primary theoretical

contribution of this thesis. It also differentiates our work from many other applications of the random coding method in coding theory. In Chapter 4 we will see how to use the results of our analysis to improve the performance of practical decoding algorithms - a novel departure from the focus of most conventional analyses. Finally, in Chapter 5 we will show how to use our techniques and bounds to *design* sparse measurement systems.

3.1 MODELS

We recall the model we study and introduce the required notation. We then describe the key insight behind the information theoretic analysis in this thesis - the correspondence between sparse measurements and coding for noisy channels. This parallel draws on the work of [32]. We also emphasize some of the key obstacles to applying the conventional Shannon-theoretic analysis to sparse measurement structures, to convince the reader of both the novelty and generality of our results. Finally, we describe a few specializations of our results to different structures that model some of the specific applications we are interested in.

3.1.1 Basic Model Description

We now describe the general model we analyze. For the major part of this chapter we keep the measurement structure general so that the analysis holds for sparse regular measurement structures that model regular pooling designs or sparse 'arbitrary' measurements like those studied by [32]. We will later show how to specialize the results to various scenarios of interest.

Our general model can be represented using a factor graph [3] as in Figure 2.1. The factor graph is a bipartite graph with edges between two kinds of nodes. The nodes represented by a circle are variables and squares represent 'factors'. A factor is a specific

function of the variables it is connected to. The factor graph as a whole indicates how the global function, or joint distribution, can be written as the product of factors.

We are interested in factor graphs that represent sparse measurement structures. Referring to Figure 2.1, the state of the input is modeled by k discrete random variables $\mathbf{V} = \left[V^1, \ldots, V^k\right]$. Each position in the vector represents a discrete input at that 'location', which takes values in $\mathcal{V}$. Each of the n measurements, Y^u for $u \in \{1, \ldots, n\}$, is a function of c inputs $Z^u = [V^{l(u,1)}, \ldots, V^{l(u,c)}] \in \mathcal{V}^c$, resulting in a measurement structure $\mathcal{C}_n$. We define the rate of the measurement system to be $R = \frac{k}{n}$. We are interested in sparse measurement systems, by which we mean that $c = \Theta(1)$.

To model $\mathbf{Y}^n$, we assume that the overall measurement function can be represented as a composition of functions. Each noiseless function value $X^u \in \mathcal{X}$, which is a function $X^u = \Psi\left(V^{l(u,1)}, \ldots, V^{l(u,c)}\right)$ computed by the measurement system. Each X^u is corrupted independently according to a probability mass function (p.m.f.) $P_{Y|X}\left(Y^u = y | X^u = x\right)$ resulting in the observed measurements $\mathbf{Y} \in \mathcal{Y}^n$. Thus, the factor graph of a sparse measurement structure represents the factorization of the joint distribution over the random variables $\mathbf{V}$ and $\mathbf{Y}$ as

$$P(\mathbf{v}, \mathbf{y}) = \prod_{i=1}^{k} P_V(v^i) \prod_{u=1}^{n} P_{Y|X}(y^u | \Psi(z^u)) \tag{3.1}$$

Given the (noisy) measurements the central problem in sparse measurement systems is that of detection, that is, reconstructing the discrete inputs inputs $\mathbf{v}$ given the measurements $\mathbf{Y} = \mathbf{y}$. We will use a decoder $q : \mathcal{Y}^n \to \mathcal{Y}^k$ to obtain an estimate $\widehat{\mathbf{v}}$ of the input. One choice of decoder is the Maximum A-Posteriori (MAP) decoder, which finds an assignment to the hidden variables $\widehat{\mathbf{v}}$ that maximizes the conditional probability,

$$\widehat{\mathbf{v}} = q(\mathbf{y}) = \arg\max_{\mathbf{v}} p(\mathbf{v}|\mathbf{y}) \tag{3.2}$$

A fundamental parameter of interest will be the distance or distortion between a particular input $\mathbf{v}$ and the reconstruction $\widehat{\mathbf{v}}$. Perfect detection of the input may not be possible from sparse noisy measurements, even asymptotically. So, we define the tolerable distortion region around an input vector $\mathbf{v}_m$ as

$$\mathcal{D}_{\mathbf{v}_m} = \left\{ m' : \frac{1}{k} d_{GEN}\left(\mathbf{v}_{m'}, \mathbf{v}_m\right) < D \right\} \tag{3.3}$$

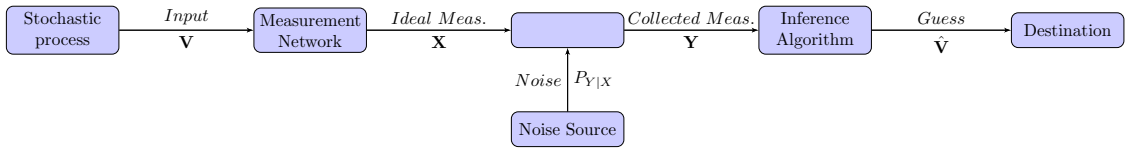
for some distortion $D \in [0, 1]$ where $d_{GEN}\left(\bullet, \bullet\right)$ is a general additive distance function, such as the Hamming distance d_H.

$$d_H(\mathbf{v}, \widehat{\mathbf{v}}) = \sum_{i=1}^{k} 1_{v^i = \widehat{v}^i} \tag{3.4}$$

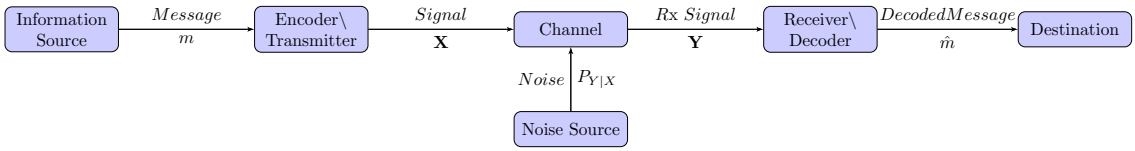
The object of our interest is the probability that the output of the decoder is outside a tolerable distortion region of the true state of input. We can define the probability of decoding failure as,

$$P_{e,\mathcal{C}_n} = P(d_{GEN}(\widehat{\mathbf{v}}, \mathbf{v}) > D) \tag{3.5}$$

Motivated by applications, we define an error event for a given measurement structure $\mathcal{C}$ and true input $\mathbf{V} = \mathbf{v}_m$ as the event $\{q(\mathbf{Y}) \notin \mathcal{D}_{\mathbf{v}_m}\}$, and the corresponding probability of error $P_{e,m,\mathcal{C}}$. We further define $P_{e,\mathcal{C}}$ to be $P_{e,m,\mathcal{C}}$ averaged across inputs m. We say that a rate R is *achievable* if there exists a sequence of measurement structures $\mathcal{C}_n$, with $P_{e,\mathcal{C}_n} \to 0$ as $n \to \infty$ for fixed R. Similar to [65], we define the sensing capacity $C(D)$ at distortion D as $\max_R R$ over achievable R. By definition, a larger sensing capacity implies that we can detect the input using fewer measurements. We are interested in lower bounds on the sensing capacity, which are achievable rates for detection.



(a) Abstract model of a measurement system



(b) Abstract model of a communication system

Table 3.1: Parallels between communicating over noisy channels and detection with noisy measurements

Communication System	Measurement System
Message m	Input $\mathbf{v}_m$
Encoder	Measurement Structure
Codeword $\mathbf{X}$	Noiseless Measurements $\mathbf{X}$
Channel Noise $P_{Y\|X}$	Measurement Noise $P_{Y\|X}$
Received Signal $\mathbf{Y}$	Noisy Measurements $\mathbf{Y}$
Decoder	Detector
Decoded Message $\hat{m}$	Detected input $\hat{\mathbf{v}}$
Capacity $\hat{m}$	Sensing Capacity $\hat{\mathbf{v}}$

3.1.2 Sparse Measurements and Communication Channels

Given a measurement system, we can represent the input, noisy measurements and the decoding process with an abstract generative model as shown in Figure 3.1(a). Similarly we illustrate the classical communication model from [29] in Figure 3.1(b). Once these two figures are placed side by side the parallels become clear, as described in [2, 32] for sensor networks and expressed in Table 3.1. We will use these terms interchangeably in this thesis, alternating between the communication view and the measurement view as appropriate.

Given these parallels, [32] studied how to extend Shannon's seminal channel capacity results [29] to models of sensor networks. Rachlin et. al. [65] showed how measurement systems are fundamentally different from communication systems because the codewords

arising from measurements are correlated with each other. In order to overcome the limitations of measurement systems they introduce a notion of 'Sensing Capacity' that is non-zero in many situations of interest. They did this by allowing for error or distortion in the reconstruction. Recall that given a tolerable distortion D, we call a rate R achievable if the sequence of sparse measurement system $\mathcal{C}_n$, satisfies $P_{e,\mathcal{C}} \to 0$ as $n \to \infty$. The sensing capacity is defined as $\max R$ over achievable R. The sensing capacity plays the same role for sparse measurement systems that channel capacity plays for communication systems.

The most important differences between channel coding and sparse measurements (also noted by [32]) that must be accounted for in our analysis are as follows,

- In a communication system, the codeword for a message can be an arbitrary function of the message because the system designer has complete control over the encoding process. On the other hand, a sparse measurement system is limited by the kinds of measurements available, and so, cannot implement arbitrary functions between the input and measurements, but only those transformations that respect physical constraints.

- The sparsity of the measurements means that each measurement can only be a function of a small, fixed subset of the inputs. This leads to the problem that *similar inputs many not be distinguishable with arbitrarily small error probability using a sparse, noisy measurement system.*

Thus, while the parallels to channel coding provide insights and analytical tools, these need to be modified to account for the above properties of sparse measurements.

3.1.3 Models of sparse measurement systems and their motivation

Borrowing from information theory, we plan to use the probabilistic method with an *ensemble* of sparse measurement structures. An ensemble is a distribution over possible measurement structures. We calculate the average probability of error, $P_{e,\mathcal{C}_n}$, based on (3.5), over the measurement ensemble. Our use of the probabilistic method can be interpreted in two different ways :

- The use of ensembles is an analytical tool that allows us to apply the probabilistic method. We want to show the existence of a sequence of measurement structures $\mathcal{C}_n$ such that $P_{e,\mathcal{C}_n} \to 0$ as $n \to 0$.

- The measurement structures are actually the result of a random process (such as, random deployment of sensors or random choice of pooling designs) and we are analyzing the average performance over randomly occurring measurement structures. We want to show that $P_e \to 0$ where $P_e = E_{\mathcal{C}_n} P_{e,\mathcal{C}_n}$.

The ensemble must be chosen carefully to reflect the physical reality by which the measurement structure arises. That is, the random deployment of physical sensors would result in a different measurement structure than in group testing designs chosen to ensure fairness. These constraints on the ensemble differentiate our work from communication theory where the system designer can choose any arbitrary ensemble, since he has complete control over the encoder. We describe three different ensembles motivated by the applications from Chapter 2.

3.1.3.1 The arbitrary connections ensemble

The first measurement structure, shown in Figure 2.1, is the arbitrary connections model motivated by [2] and studied in detail in [32]. This structure models sparse measurement

structures without spatial aspects, such as, the human olfactory system [94], chemical sensors/electronic noses [48] or LT Codes [88].

Since these models do not have any spatial constraints, each measurement can be a function of an *arbitrary* subset of the inputs. We do however impose (for now) a constraint that each measurement measures exactly c inputs, though we will show how to relax this constraint later. In order to generate a random measurement structure from this ensemble, we assume that each measurement (or sensor) chooses c locations to measure independently and uniformly at random, and then generates a function Ψ of those locations. In such a random drawing, the distribution of each noiseless output X^u for a measurement u is i.i.d. conditioned on the occurrence of a fixed input. The motivation for this ensemble was to model the procedure of dropping sensors at random to create a sensor network.

This model was studied in some detail in [26, 32] where a lower bound on the capacity of this measurement structure was derived. We will improve that result by obtaining a tighter bound. One disadvantage of this ensemble is that because the sensors choose locations to measure independently, some inputs may be measured many times, while a constant fraction may not be measured at all. This leads to a substantial loss in performance, which is apparent even when the measurements are noiseless. We will next look at a sparse measurement structure that does not suffer from this shortcoming.

3.1.3.2 The sparse regular ensemble

The second model we study is motivated by parallels to coding theory, and is shown in Figure 3.1. This ensemble is motivated by the structure of LDPC codes [58]. It is also related to the configuration model in combinatorics and graph theory [95], and to the micro-canonical ensemble in the statistical physics literature [96].

In this ensemble not only is each measurement a function of c inputs, but also every input is measured exactly d times. A measurement structure is generated from this ensemble

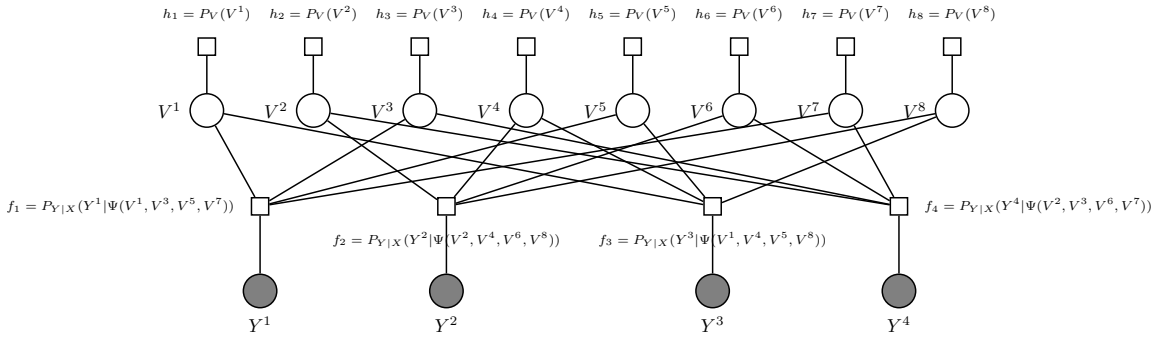


Figure 3.1: Factor graph of sparse regular measurement structure with $c = 4$, $d = 2$, $k = 8$, $n = 4$, and the rate $R = 2$. f_u are measurement factors and h_j factors correspond to prior information.

as follows. We create $d * k$ input sockets and $c * n = d * k$ output sockets. A measurement structure is generated by matching the input and output sockets with a random matching graph and connecting nodes corresponding to matched sockets. This results in (with some technicalities needed for multiple edges) a structure drawn uniformly from all (c, d) regular bipartite graphs - which we call the sparse regular ensemble.

This model has a number of advantages over the arbitrary model in that all inputs are measured the same number of times, ensuring fair and complete coverage. This is especially important in medical applications like group testing for genetic screening or drug discovery as discussed in Chapter 2. Unfortunately in such a random drawing, the distribution of each noiseless output X^u for a measurement u is not independent of other measurements, even for a fixed input, complicating the analysis substantially. The generality of our analysis (which we claim as one of the major theoretical contributions of this thesis) allows us to analyze this ensemble despite the correlations.

3.1.3.3 The contiguous ensemble

The next structure we study is motivated by sensor network applications. This model was also studied in [32] as a way to account for the spatial nature of physical sensors. Real world sensors have a locality property, in that, they cannot measure arbitrary locations but

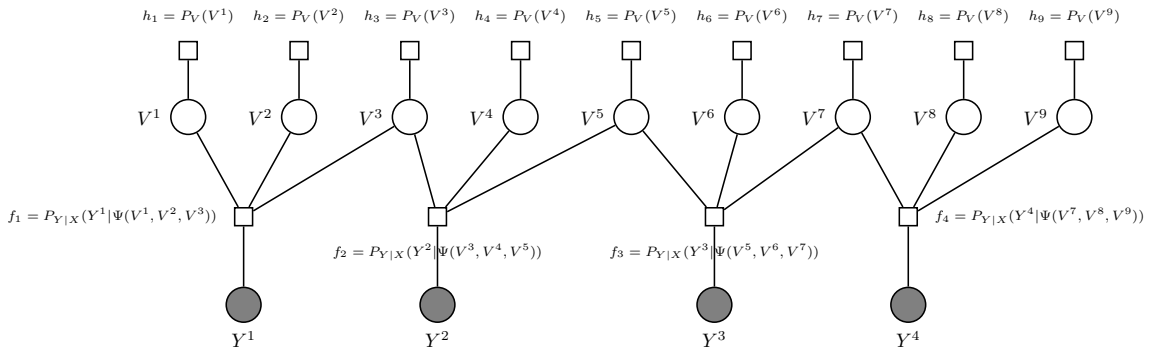


Figure 3.2: Factor graph of detecting variables from sparse measurements with $c = 3$, $k = 9$, $n = 4$. Rate $R = 2$ since the network extends further than illustrated. f_u are measurement factors and h_j factors correspond to prior information.

sense locations that are near each other in a spatial sense. In this model each measurement chooses a location uniformly from among all locations and then measures c contiguous locations. This is illustrated in Figure 3.2.

Since this model was studied in detail in [32] we do not consider a capacity analysis of this measurement structure in this thesis. However, we will show how the analysis technique and the resulting expression for sensing capacity can be used to design *algorithms* for inference in such measurement systems in Chapter 4.

3.1.3.4 Decoders with mismatch and uncertainty

While the previous sub-sections described different measurement structures, in this thesis, we will also look at how using a different decoder can affect performance. This problem is motivated by our experiences with trying to use the theory and algorithms in real applications like thermal measurements [15] and group testing for drug discovery [1].

As described in Chapter 2, a major problem in real sensor networks and other physical applications is the lack of exact knowledge of system parameters. For example, in a thermal sensor network, the ambient temperature may drift. In a sonar or laser based mapping application, the reflectivity of the environment may change as the sensor is moved. In

applications such as group testing for drug discovery, the concentrations and re-activities of various substances may be unknown and may depend on changing ambient conditions. This leads to the problem of parameter uncertainty. In addition, even if these parameters are estimated during a calibration phase, they may not be estimated perfectly or may drift with time, leading to the problem of parameter mismatch.

We will show that these problems, parameter uncertainty and model mismatch, are fundamentally different and have different effects on the capacity of a measurement system. In addition our analysis will suggest new decoders which are useful in practice, as discussed in Chapter 4.

3.2 LOWER BOUNDS ON THE CAPACITY OF SPARSE MEASUREMENT SYSTEMS

In this thesis, we will analyze the fundamental information theoretic limits of the performance of sparse measurement systems. We will use random coding ideas [29] and Chernoff bounding techniques [97] developed in the information theory community, which are traditionally used to analyze codes over noisy channels. This is possible because of the parallels described in Section 3.1.2. Our analysis will be based on the probabilistic method [35], augmented with combinatorial calculations and large deviations theory [34].

3.2.1 The Probabilistic Method for sparse measurement ensembles

Following [32] and our own work in [21, 27] we will use a random measurement argument, paralleling Shannon's random coding argument in his model of a communication system, to derive a lower bound on the capacity of sparse measurement systems. In Shannon's proof of the capacity of channels, the codewords corresponding to the messages are drawn i.i.d. from a particular distribution, which is optimized to result in the highest

achievable rate. However, this is not a good model for our purposes, since we want to constrain the mappings $\mathbf{V}_m \rightarrow \mathbf{X}_m$ to be only those mappings that can be implemented using a sparse measurement structure $\mathcal{C}_n$. Such mappings cause dependencies between the codewords $\mathbf{X}_m$ for different messages m, and additionally, among the different measurements X_m^u of a codeword $\mathbf{X}_m$. To account for these dependencies we will use the method of types [98], motivated by the work of [32]. These dependencies differentiate our work from conventional applications of the random coding method.

Based on the above discussion, before we apply the probabilistic method to the analysis of sparse measurement systems, we must define an appropriate distribution (or ensemble) over sparse measurement structures. This is contrast to the distribution over codewords used in Shannon theory. In Section 3.1.3, we specified three different ensembles that are of practical interest. Rather than derive results separately for each ensemble, we will first derive a general result, and the show how the theorem holds, as long as the measurement ensembles satisfy a certain permutation invariance condition. Since all the three ensembles satisfy this condition we can specialize the theorem to these cases later. This general approach has the added benefit that it allows us to see how our final result depends on the chosen measurement ensemble, the noise model, prior information, measurement function and decoder.

3.2.2 Permutation invariance and the method of types

As indicated in the previous sub-section, we will apply Shannon's probabilistic method while respecting the constraints imposed by measurement systems. In order to do this we will use the ensembles described in Section 3.1.3. We will now define an important property of these ensembles, permutation invariance, which allows our asymptotic analysis. Permutation invariance is a property of the distribution over measurements induced by a

ensemble, and so, we define these distributions below.

There are two distributions that will arise during our analysis that depend on the particular measurement ensemble. Before we define and characterize these distributions we need to define some concepts. Because of the symmetries introduced by the random generation of the measurement structure the macroscopic parameters we are interested in will depend on the 'type' of the input vector $\mathbf{v}$ [98]. The type of an input vector $\mathbf{v}$, $\gamma = \left(\gamma^1, \ldots, \gamma^{|\mathcal{V}|}\right)$, is defined as the empirical p.m.f. of $\mathbf{v}$. The joint type of two input vectors $\mathbf{v}_m$ and $\mathbf{v}_{m'}$, $\lambda = \left(\lambda^{11}, \ldots, \lambda^{|\mathcal{V}||\mathcal{V}|}\right)$ is a p.m.f. of length $|\mathcal{V}|^2$ with λ^{ab} denoting the fraction of locations $\{i : v_m^i = a, v_{m'}^i = b\}$. The type of $\mathbf{z}_m$ is δ, where δ^e is the fraction of u such that $z_m^u = e$, that is, the empirical p.m.f. of $\mathbf{z}_m$. Finally the joint type of $\mathbf{z}_m$ and $\mathbf{z}_{m'}$, $\kappa = (\ldots, \kappa(e, f), \ldots)$ where $e, f \in \mathcal{V}^c$ and $\kappa(e, f)$ is the fraction of outputs $\{u : z_m^u = e, z_{m'}^u = f\}$.

We illustrate these concepts with an example. Suppose $k = 3$ is the number of inputs, and consider the case where we take $n = 3$ measurements. Let $\mathcal{V} = \{0, 1\}$ and m and m' be such that $\mathbf{v}_m = [0\ 0\ 1]$ and $\mathbf{v}_{m'} = [0\ 1\ 0]$. Measurement 1 measures locations 1 and 2, Measurement 2 measures locations 2 and 3 and Measurement 3 measures locations 3 and 1. Once the measurement structure is fixed we can define $\mathbf{z}_m = \begin{pmatrix} 00 \\ 01 \\ 10 \end{pmatrix}$ and $\mathbf{z}_{m'} = \begin{pmatrix} 01 \\ 10 \\ 00 \end{pmatrix}$. The type δ of $\mathbf{z}_m$ would be $[1/3, 1/3, 1/3, 0]$ since there are no occurrences of 11 in $\mathbf{z}_m$. Similarly the joint type of $\mathbf{z}_m$ and $\mathbf{z}_{m'}$ is $\kappa(00, 01) = 1/3$, $\kappa(01, 10) = 1/3$, $\kappa(10, 00) = 1/3$ with all other entries being 0. Finally, suppose that each measurement computes a simple sum of its inputs. Then $\mathbf{x}_m = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$ and $\mathbf{x}_{m'} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$. $\mathbf{Y}$ would be the noisy version of the true $\mathbf{x}_m$, drawn from the noise model $P_{Y|X}$.

Stated in terms of types, the first distribution that is important to us is $P_m(\mathbf{z})$, which is the probability over the measurement ensemble of the vector $\mathbf{Z}_m = \mathbf{z}$. The other distribution is $P_{m,m'}(\mathbf{z}, \mathbf{z}')$, which is the probability that $\mathbf{Z}_m = \mathbf{z}$ and $\mathbf{Z}_{m'} = \mathbf{z}'$.

We define the ensemble to be input permutation invariant if $P_m(\mathbf{z}) = P^{\gamma}(\mathbf{z})$ depends only on the type γ of $\mathbf{v}_m$, $\forall$ $\mathbf{v}_m$ of type γ and if $P_{m,m'}(\mathbf{z}, \mathbf{z}') = P^{\lambda}(\mathbf{z}, \mathbf{z}')$ depends only on the joint type λ of $\mathbf{v}_m$ and $\mathbf{v}_{m'}$, $\forall$ $\mathbf{v}_m, \mathbf{v}_{m'}$ of joint type λ.

We define the ensemble to be output permutation invariant as well, if $P^{\gamma}(\mathbf{z}) = P^{\gamma,\delta}$ depends only on the type δ of $\mathbf{z}$, $\forall$ $\mathbf{z}$ of type δ, and $P^{\lambda}(\mathbf{z}, \mathbf{z}') = P^{\lambda,\kappa}$ depends only on the joint type κ of $\mathbf{z}$ and $\mathbf{z}'$, $\forall$ $\mathbf{z}, \mathbf{z}'$ of joint type κ.

Thus, input permutation invariance is a characteristic of an ensemble of measurement structures that ensures that changing the ordering of the inputs does not change any of the probability distributions that arise in our analysis. Similarly output permutation invariance states that changing the ordering of the measurements does not change the distributions. We note that the arbitrary and sparse regular models are both input and output permutation invariant while the contiguous ensemble is output permutation invariant but not input permutation invariant. However, this can be accounted for by using the idea of c-order types as was done in [32].

Our main theorem in the next section is a lower bound on the capacity of sparse measurement systems that depends on these distributions. The general theorem can then be specialized to a particular permutation invariant ensemble by computing the appropriate distributions.

3.2.3 Achievable rates for detection : A lower bound on the sensing capacity

We now answer a natural question that arises with sparse measurement systems - how many measurements are sufficient to guarantee that we can detect the random variables $\mathbf{V}$ to within a specified distortion D? i.e what is the sensing capacity $C(D)$? Rachlin [32] derived a lower bound on sensing capacity $C_{LB}^{ML}(D)$ using Gallager's [97] Chernoff bounding technique. However, [32] assumed that optimal ML or MAP decoders are used,

and furthermore, that all the problem model and parameters are known exactly. In addition [32] studied only the arbitrary connections and contiguous model but not the sparse regular ensemble. So, [32] could exploit the independence of X^u for different u, for a given input $\mathbf{V} = \mathbf{v}_m$. We will show that this independence is not necessary for the analysis, but rather the more general permutation invariance condition described in the previous section suffices. We provide a simpler, yet more general derivation of C_{LB}, which allows us to go beyond the results in [32], by addressing some of these limitations. This allows us to extend the analysis to the sparse regular ensemble and also analyze a more general class of decoders.

We assume a decoder that maximizes some similarity score $s^{\mathcal{C}}$ (which depends on the specific measurement structure $\mathcal{C}$) between the received $\mathbf{Y}$ and possible noiseless measurements $\mathbf{x}_m$ for different $\mathbf{v}_m$. Since there is a one-to-one correspondence between m and $\mathbf{v}_m$, we can write such a decoder as $g(\mathbf{Y}) = \arg\max_m s^{\mathcal{C}}(\mathbf{x}_m, \mathbf{Y})$. For example, a Maximum Likelihood(ML) decoder has $s^{\mathcal{C}}(\mathbf{x}_m, \mathbf{Y}) = p_{\mathbf{Y}|\mathbf{X}}(\mathbf{Y}|\mathbf{x}_m)$. However, the analysis also holds for decoders maximizing other (possibly sub-optimal) scores. We define

$$\mathcal{S}^{\gamma}(D) = \{\lambda : \sum_{a\in\mathcal{V}} \sum_{b\in\mathcal{V}, b\neq a} \lambda^{ab}\}, \ \sum_{b\in\mathcal{V}} \lambda^{ab} = \sum_{b\in\mathcal{V}} \lambda^{ba} = \gamma^a \ \forall\, a \in \mathcal{V} \tag{3.6}$$

as the set of joint types λ corresponding to vectors $\mathbf{v}_{m'}$ at distortion D from a vector $\mathbf{v}_m$ of type γ. $H(\bullet)$ is the entropy function.

We define a tilting function

$$f^{\lambda,w}(\mathbf{X}_m, \mathbf{Y}) = E_{\mathbf{X}_{m'}|\mathbf{X}_m}\left(s^{\mathcal{C}}(\mathbf{X}_{m'}, \mathbf{Y})\right)^w \tag{3.7}$$

This is the expected value of the score for an incorrect input $\mathbf{v}_{m'}$ which has a joint type λ with the true input $\mathbf{v}_m$. The form of this tilting function depends on the measurement structure and measurement function, and is different from the tilting function that arises in

coding theoretic analyses. Intuitively, if this tilting metric is small compared to the value of $s^{\mathcal{C}}(\mathbf{x}_m, \mathbf{Y})$ for the true input $\mathbf{v}_m$, then our decoder is unlikely to output a guess $\mathbf{v}_{m'}$ that has joint type λ with $\mathbf{v}_m$.

Our theorem then concerns the large deviations of $Z_n^{\lambda,w}$

$$Z_n^{\lambda,w} = \frac{1}{n}\log\left(\frac{\left(s^{\mathcal{C}}(\mathbf{X}_m, \mathbf{Y})\right)^w}{f^{\lambda,w}(\mathbf{X}_m, \mathbf{Y})}\right) \tag{3.8}$$

$Z_n^{\lambda,w}$ is the ratio of these two quantities - the *empirical* score for the correct input and the *expected* score for incorrect inputs with joint type λ. We define the log-moment generating function $\Lambda_n^{\lambda,w}(s)$,

$$\Lambda_n^{\lambda,w}(s) = \log E[e^{sZ_n^{\lambda,w}}] \tag{3.9}$$

Given these definitions we can state our general theorem.

Theorem 1 (Lower bound on sensing capacity of permutation invariant measurement ensembles) *Suppose we use a decoder that maximizes $s^{\mathcal{C}}$. Further, suppose that the limit*

$$\Lambda^{\lambda,w}(s) = \lim_{n\to\infty}\frac{1}{n}\Lambda_n^{\lambda,w}(ns) \tag{3.10}$$

exists, and

$$\Lambda^{*\lambda,w}(x) = \sup_s\left(sx - \Lambda^{\lambda,w}(s)\right) \tag{3.11}$$

is the Fenchel-Legendre transform. If

$$\mathcal{D}_{\Lambda^{\lambda,w}} = \left\{s \in \mathcal{R} : \Lambda^{\lambda,w}(s) < \infty\right\} \tag{3.12}$$

and the origin belongs to the interior of $\mathcal{D}_{\Lambda^{\lambda,w}}$, then, there exists a sequence of sparse measurement structures $\mathcal{C}_n$, such that the average probability of error, $P_{e,\mathcal{C}_n} \to 0$ as $n \to \infty$, for fixed R, as long as $R < C_{LB}(D)$. Here $C_{LB}(D)$ is a lower bound on sensing capacity for a given distortion D, given by,

$$C_{LB(D)} = \min_{\lambda\notin\mathcal{S}^{\gamma}(D)}\sup_{w\geq 0}\frac{\sup_{T(\lambda)}\left\{T(\lambda) : \inf_{x\in(-\infty,T(\lambda)]}\Lambda^{*\lambda,w}(x) > 0\right\}}{[H(\lambda) - H(\gamma)]} \tag{3.13}$$

Proof: Let S_m^λ be the set of inputs $\mathbf{v}_{m'}$ at joint type λ with $\mathbf{v}_m$.

$$P_{e,m} = P\left(\bigcup_{m' \notin \mathcal{D}_{\mathbf{v}_m}} \left\{s^{\mathcal{C}}\left(\mathbf{X}_{m'}, \mathbf{Y}\right) > s^{\mathcal{C}}\left(\mathbf{X}_m, \mathbf{Y}\right)\right\}\right) \tag{3.14}$$

$$= P\left(\bigcup_{\lambda \notin \mathcal{S}^\gamma(D)} \bigcup_{m' \in S_m^\lambda} \left\{s^{\mathcal{C}}\left(\mathbf{X}_{m'}, \mathbf{Y}\right) > s^{\mathcal{C}}\left(\mathbf{X}_m, \mathbf{Y}\right)\right\}\right) \tag{3.15}$$

We introduce $f^{\lambda,w}\left(\mathbf{X}_m, \mathbf{Y}\right)$, which is a λ specific tilting function to be optimized later, and $w \geq 0$ is a parameter that can also be optimized.

$$\leq \sum_{\lambda \notin \mathcal{S}^\gamma(D)} P\left(\bigcup_{m' \in S_m^\lambda} \left\{\frac{\left(s^{\mathcal{C}}\left(\mathbf{X}_{m'}, \mathbf{Y}\right)\right)^w}{f^{\lambda,w}\left(\mathbf{X}_m, \mathbf{Y}\right)} > \frac{\left(s^{\mathcal{C}}\left(\mathbf{X}_m, \mathbf{Y}\right)\right)^w}{f^{\lambda,w}\left(\mathbf{X}_m, \mathbf{Y}\right)}\right\}\right) \tag{3.16}$$

$$\tag{3.17}$$

We define a 'bad noise' event using a λ specific threshold $e^{nT(\lambda)}$

$$\mathcal{Y}_{b,\lambda} = \left\{\frac{\left(s^{\mathcal{C}}\left(\mathbf{X}_m, \mathbf{Y}\right)\right)^w}{f^{\lambda,w}\left(\mathbf{X}_m, \mathbf{Y}\right)} < e^{nT(\lambda)}\right\} \tag{3.18}$$

We now union bound (3.16) over good and bad noise events. For the good noise condition we have reduced the problem to a tilted threshold test.

$$P_{e,m} \leq \sum_{\lambda \notin \mathcal{S}^\gamma(D)} P\left(\mathcal{Y}_{b,\lambda}\right) + \tag{3.19}$$
$$\sum_{\lambda \notin \mathcal{S}^\gamma(D)} \sum_{m' \in S_m^\lambda} P\left(\left\{\frac{\left(s^{\mathcal{C}}\left(\mathbf{X}_{m'}, \mathbf{Y}\right)\right)^w}{f^{\lambda,w}\left(\mathbf{X}_m, \mathbf{Y}\right)} \geq e^{nT(\lambda)}\right\}\right)$$

Using results from [98], $\left|\mathcal{S}^\gamma\left(D\right)\right| \leq n^{|\mathcal{V}|^2}$ since there are only a polynomial number of joint types, and $|S_m^\lambda| \leq e^{nR[H(\lambda)-H(\gamma)]}$. Considering the second term in (3.19),

$$Term\ 2 \leq n^{|\mathcal{V}|^2} \max_{\lambda \notin \mathcal{S}^\gamma(D)} [e^{nR[H(\lambda)-H(\gamma)]} \tag{3.20}$$
$$P\left(\left\{\frac{\left(s^{\mathcal{C}}\left(\mathbf{X}_{m'}, \mathbf{Y}\right)\right)^w}{f^{\lambda,w}\left(\mathbf{X}_m, \mathbf{Y}\right)} \geq e^{nT(\lambda)}\right\}\right)]$$

Applying Markov's inequality,

$$\leq \quad n^{|\mathcal{V}|^2} \max_{\lambda \notin \mathcal{S}^\gamma(D)} e^{nR[H(\lambda)-H(\gamma)]} e^{-nT(\lambda)} \tag{3.21}$$
$$E_{\mathbf{X}_m} E_{\mathbf{Y}|\mathbf{X}_m} E_{\mathbf{X}_{m'}|\mathbf{X}_m} \frac{\left(s^{\mathcal{C}}\left(\mathbf{X}_{m'}, \mathbf{Y}\right)\right)^w}{f^{\lambda,w}\left(\mathbf{X}_m, \mathbf{Y}\right)}$$

We now choose the tilting function $f^{\lambda,w}\left(\mathbf{X}_m, \mathbf{Y}\right) = E_{\mathbf{X}_{m'}|\mathbf{X}_m}\left(s^{\mathcal{C}}\left(\mathbf{X}_{m'}, \mathbf{Y}\right)\right)^w$ resulting in

$$Term\ 2 \ \leq n^{|\mathcal{V}|^2} \max_{\lambda \notin \mathcal{S}^\gamma(D)} e^{-n[T(\lambda)-R[H(\lambda)-H(\gamma)]]} \tag{3.22}$$

If the measurement ensembles are input permutation invariant, as discussed in Section 3.2.2, the distributions only depend on the type γ of $\mathbf{v}_m$, and the pairwise joint types λ of $\mathbf{v}_m$ and $\mathbf{v}_{m'}$. This probability goes to zero if $R < \min_{\lambda \notin \mathcal{S}^\gamma(D)} \frac{T(\lambda)}{[H(\lambda)-H(\gamma)]}$.

For the first term in (3.19), we have

$$Term\ 1 \ \leq n^{|\mathcal{V}|^2} \min_{\lambda \notin \mathcal{S}^\gamma(D)} P\left(Z_n^{\lambda,w} < T\left(\lambda\right)\right) \tag{3.23}$$

where we used the definition of $Z_n^{\lambda,w}$ and its log-m.g.f $\Lambda_n^{\lambda,w}\left(s\right)$,

$$Z_n^{\lambda,w} = \frac{1}{n} \log\left(\frac{\left(s^{\mathcal{C}}\left(\mathbf{X}_m, \mathbf{Y}\right)\right)^w}{f^{\lambda,w}\left(\mathbf{X}_m, \mathbf{Y}\right)}\right), \Lambda_n^{\lambda,w}\left(s\right) = \log E[e^{sZ_n^\lambda}] \tag{3.24}$$

Suppose the limit $\Lambda^{\lambda,w}\left(s\right) = \lim_{n\to\infty} \frac{1}{n}\Lambda_n^{\lambda,w}\left(ns\right)$ exists, and $\Lambda^{*\lambda,w}\left(x\right) = \sup_s\left(sx - \Lambda^{\lambda,w}\left(s\right)\right)$ is the Fenchel-Legendre transform. Define $\mathcal{D}_{\Lambda^{\lambda,w}} = \left\{s \in \mathcal{R} : \Lambda^{\lambda,w}\left(s\right) < \infty\right\}$, and suppose the the origin belongs to the interior of $\mathcal{D}_{\Lambda^{\lambda,w}}$, then, according to the Gartner-Ellis theorem [34], the probability in (3.23) goes to 0 with exponent $-\inf_{x\in(-\infty,T(\lambda)]} \Lambda^{*\lambda,w}\left(x\right)$. If this exponent is positive, then Term 2 in (3.19) also goes to 0 as $n \to \infty$. Suppose that inputs are generated independently at random with p.m.f. $\gamma = p_V\left(v = V\right)$. The average

probability of error, P_e

$$\begin{aligned} P_e &= E_{\mathcal{C}} P_{e,\mathcal{C}} = \sum_{m=1}^{2^{nR}} P_{\mathbf{V}}\left(\mathbf{V} = \mathbf{v}_m\right) E_{\mathcal{C}} P_{e,m,\mathcal{C}} \\ &= \sum_{m=1}^{2^{nR}} P_{\mathbf{V}}\left(\mathbf{V} = \mathbf{v}_m\right) P_{e,m} \end{aligned} \tag{3.25}$$

We divide the inputs $\mathbf{v}_m$ based on their types γ'. There are $e^{nRH(\gamma')}$ target vectors of each type γ'. According to (3.22) the error rate for an input m is only a function of its type γ', which we denote by $P_{e,m} = P_{e,\gamma'}$.

$$\begin{aligned} P_e &= \sum_{\gamma'} e^{nRH(\gamma')} P^{\gamma}(\mathbf{v}\ of\ type\ \gamma') P_{e,\gamma'} \\ &= \sum_{\gamma'} e^{nRH(\gamma')} e^{-nR(H(\gamma')+D(\gamma||\gamma'))} P_{e,\gamma'} \end{aligned} \tag{3.26}$$

where γ is the type corresponding to the true distribution P_V. In the above sum, in terms for which $\gamma' \neq \gamma$, the $e^{-n(D(\gamma||\gamma'))}$, causes their contribution to go to zero as $n \to \infty$, using the fact that there only a polynomial number of types [98]. And so, if the $P_{e,\gamma}$ (for typical environments m of type γ) goes to zero exponentially, then $P_e \to 0$. The condition for this to happen was stated after (3.24). If this condition is satisfied, the average probability of error across measurement structures goes to zero, and there exists a sequence of structures $\mathcal{C}_n$ for which the error rate goes to zero. This completes the proof. ■

3.2.4 Interpreting the proof of the main theorem

The theorem states that if we have a sufficient number of measurements then there exists an appropriate measurement structure and a decoder using the score $s^{\mathcal{C}_n}$ that can guarantee detection to within a distortion D with vanishingly small probability of error. *The generality of this theorem is one the primary theoretical contributions of this thesis*. In the next section we specialize this result to different measurement ensembles and decoders.

Some discussion of the intuition behind this result may be helpful. The original large scale detection problem is a hypothesis testing problem with an exponential number of competing hypotheses. Using a union bound, we group the competing hypotheses into a polynomial number of joint types, with $e^{nR[H(\lambda)-H(\gamma)]}$ hypotheses for each joint type λ. Each hypothesis test is solved using a threshold of the form $e^{nT(\lambda)}$. See (3.18). The error exponent for each joint type is a function of the largest threshold that still allows detection, for that joint type. We union bound over all the hypotheses with a particular joint type, each of which has the same probability of error. The probability of error is the same because of the permutation invariance of the measurement ensemble. The average probability of error is bounded by the worst exponent, which can be interpreted as the most confusable joint type λ. We now give the intuition behind different parts of the proof.

Equations (3.14) to (3.16) : We use the random measurement argument outlined in Section 3.2.1. We bound $P_{e,m}$, which is the expected probability of error over the random sensor networks. To do this we wish to follow the classical union bound arguments, where we union bound over all incorrect messages, as in the channel capacity proof [97]. However, because of the non-i.i.d. nature of the codewords (as explained in Section 3.2.1) we first split the union bound based on the type of the incorrect environments as suggested in [32].

Equations (3.18) to (3.24) : We split the error events based on two sources of error - i) the noise is so large that the measurements $\mathbf{Y}$ are not similar to the noiseless measurements $\mathbf{X}_m$ corresponding to the true input $\mathbf{v}_m$ (a 'bad noise' event) and ii) we draw a measurement network in which the $\mathbf{X}_{m'}$ corresponding to a different $\mathbf{v}_{m'}$ is similar to $\mathbf{Y}$. The conditions under which the probability of both these events goes to zero as $n \to \infty$ is our main large deviation result.

Equations (3.25) to (3.26) : We complete the proof by showing that if we can guarantee

that the average error probability for 'typical' inputs goes to zero exponentially fast in n, (using the large deviation result), then the average error probability across all environments goes to zero.

3.3 SPECIALIZING THE GENERAL THEOREM

We now show how to calculate lower bounds on the capacity of specific sparse measurement structures by specializing Theorem 1 to various cases of interest. The large deviations exponent calculations involve some combinatorics and analysis that is structure specific. Comparing the final results allows us to understand the effect of measurement structure on the performance of sparse measurement systems.

3.3.1 The sparse regular ensemble

The sparse regular ensemble is useful in the modeling of group testing designs that find application in biological screening and drug discovery. The primary advantage of this structure, in applications, is coverage fairness, since each input is measured the same number of times. Another advantage is the control over the pool sizes c, which can be optimized in system design as outlined in Chapter 5.

We recall the two important distributions that are ensemble specific. Stated in terms of types, the first distribution that is important to us is $P_m(\mathbf{z})$, which is the probability (over the measurement ensemble) that $\mathbf{Z}_m = \mathbf{z}$. The other distribution is $P_{m,m'}(\mathbf{z}, \mathbf{z}')$, which is the probability that $\mathbf{Z}_m = \mathbf{z}$ and $\mathbf{Z}_{m'} = \mathbf{z}'$. Using the input and output permutation symmetries from Section 3.2.2 (both of which apply to the sparse regular ensemble) and some combinatorics, $P_m(\mathbf{z}) = P^{\gamma}(\mathbf{z}) = P^{\gamma,\delta}$, $\forall$ $\mathbf{v}_m$ of type γ and $\mathbf{z}$ of type δ. Also, $P_{m,m'}(\mathbf{z}, \mathbf{z}') = P^{\lambda}(\mathbf{z}, \mathbf{z}') = P^{\lambda,\kappa}$, $\forall$ $\mathbf{v}_m, \mathbf{v}_{m'}$ with joint type λ and $\mathbf{z}$ and $\mathbf{z}'$ of joint type κ.

For the above probability distributions to be defined, δ has to be consistent with γ and

κ has to be consistent with λ, as defined below.

$$P_m(\mathbf{z}) = P^{\gamma,\delta} = \begin{cases} 2^{-ncH(\gamma)} & \text{if } \sum_{\mathbf{e}\in\mathcal{V}^c} \frac{|\{i:e^i=a\}|}{c}\delta(\mathbf{e}) = \gamma^a,\ \forall\, a \in \mathcal{V} \\ 0 & \text{otherwise} \end{cases} \tag{3.27}$$

$$P_{m,m'}(\mathbf{z},\mathbf{z}') = P^{\lambda,\kappa} = \begin{cases} 2^{-ncH(\lambda)} & \text{if } \sum_{\mathbf{e},\mathbf{f}\in\mathcal{V}^c} \frac{|\{i:e^i=a,f^i=b\}|}{c}\kappa(\mathbf{e},\mathbf{f}) = \lambda^{ab},\ \forall\, a,b \in \mathcal{V} \\ 0 & \text{otherwise} \end{cases} \tag{3.28}$$

Using these expressions we obtain Corollary 2.

Corollary 2 (Lower bound on sensing capacity of sparse regular measurements) *There exists a sequence of sparse, regular measurement structures $\mathcal{C}_n$, such that the average probability of error, $P_{e,\mathcal{C}_n} \to 0$ as $n \to \infty$, with ML decoding, for fixed R, as long as $R < C_{LB}^{REG}(D)$, where*

$$\begin{aligned} C_{LB}^{REG}(D) \;\leq\; & \inf_{\lambda\in\mathcal{S}^\gamma(D)} \inf_{\kappa \text{ consistent with } \lambda} \frac{-H(Y|Z) + cH(\lambda)}{H(\lambda) - H(\gamma)} \\ & + \frac{-H(\kappa) - \sum_{a,b,o}\kappa(a,b)P_{Y|Z}(o|a)\log P_{Y|Z}(o|b)}{H(\lambda) - H(\gamma)} \end{aligned} \tag{3.29}$$

Proof: We now specialize Theorem 1 to the special case of sparse regular ensembles give the distributions described in Section 3.3.1. In order to do we need to characterize the large deviations using the threshold $T(\lambda)$. We use T for the optimal threshold.

$$T \;=\; \sup_{T(\lambda)} \left\{ T(\lambda) : \inf_{x\in(-\infty\ T(\lambda))} \Lambda^{*\lambda}(x) > 0 \right\} \tag{3.30}$$

The condition can be rewritten as,

$$T \;=\; \sup_{T(\lambda)} \left\{ T(\lambda) : \inf_{x\in(-\infty\ T(\lambda))} [\sup_s \left(sx - \Lambda^{\lambda,w}(s) \right)] > 0 \right\} \tag{3.31}$$

Since $\left(sx - \Lambda^{\lambda,w}(s) \right)$ is convex in x and concave in s, we can interchange the sup and the

inf.

$$T = \sup_{T(\lambda)} \left\{ T(\lambda) : \sup_{s \in (-\infty\ T(\lambda))} [\inf_x \left(sx - \Lambda^{\lambda,w}(s) \right)] > 0 \right\} \tag{3.32}$$

For any $s > 0$ the infimizing x will be $-\infty$ and $T = -\infty$. So $s \leq 0$. If $s \leq 0$, because the term is linear in x, the infimizing x will be $T(\lambda)$.

$$T = \sup_{T(\lambda)} \left\{ T(\lambda) : \sup_{s \leq 0} \left(sT(\lambda) - \Lambda^{\lambda,w}(s) \right) > 0 \right\} \tag{3.33}$$

The supremizing $T(\lambda)$ attains equality above. This supremizing value is T and satisfies

$$\left(s^* T - \Lambda^{\lambda,w}(s^*) \right) = 0 \tag{3.34}$$

where s^* is the supremizing s. This can be rewritten as

$$T = \sup_{s \leq 0} \frac{\Lambda^{\lambda,w}(s)}{s} \tag{3.35}$$

We can lower bound by choosing a particular value of s. We choose $\lim_{s \uparrow 0}$.

$$T \geq \lim_{s \uparrow 0} \frac{\Lambda^{\lambda,w}(s)}{s} \tag{3.36}$$

We define ξ as the empirical joint type of $\mathbf{Z}_m$ and $\mathbf{Y}$. This is defined to be consistent with δ as follows,

$$\sum_{o \in \mathcal{Y}} \xi(a,o) = \delta(a) \ \forall\ a \in \mathcal{V}^c \tag{3.37}$$

We return to (3.36). We do not optimize over w setting it to 1. We now replace the calculated distributions from (3.27) and (3.28).

$$T \geq \lim_{s \uparrow 0} \lim_{n \to \infty} \frac{1}{n} \frac{1}{s} \log[\sum_{\xi} 2^{nH(\xi)} 2^{-ncH(\gamma)} 2^{n \sum_{a,o} \xi(a,o) \log P_{Y|Z}(o|a)} \tag{3.38}$$

$$\left(\sum_{\kappa} 2^{nH(\kappa)} 2^{-nH(\delta)} 2^{-ncH(\lambda)} 2^{ncH(\gamma)} 2^{n \sum_{b,o} \rho(b,o) \log P_{Y|Z}(o|b)} 2^{-n \sum_{a,o} \xi(a,o) \log P_{Y|Z}(o|a)} \right)^{-s}] \tag{3.39}$$

where $\rho(b,o) = \sum_a \kappa(a,b)\frac{\xi(a,o)}{\delta(a)}$, is the empirical p.m.f. between $\mathbf{Z}_{m'}$ and $\mathbf{Y}$. Let ξ^* be the ξ that *supremizes* over ξ the term in the summation. Let δ^* be the corresponding type for $\mathbf{Z}_m$. Then, replacing the $\sum_\xi$ by a larger term (since we are dividing by s which is negative) we continue with the bound,

$$T \geq \lim_{s\uparrow 0}\lim_{n\to\infty}\frac{1}{n}\frac{1}{s}\log[(n+1)^{|\mathcal{V}|^c|\mathcal{Y}|}2^{nH(\xi^*)}2^{-ncH(\gamma)}2^{n\sum_{a,o}\xi^*(a,o)\log P_{Y|Z}(o|a)} \tag{3.40}$$

$$\left(\sum_\kappa 2^{nH(\kappa)}2^{-nH(\delta^*)}2^{-ncH(\lambda)}2^{ncH(\gamma)}2^{n\sum_{b,o}\rho(b,o)\log P_{Y|Z}(o|b)}2^{-n\sum_{a,o}\xi^*(a,o)\log P_{Y|Z}(o|a)}\right)^{-s}] \tag{3.41}$$

Splitting the $\log$ term into two separate terms we get,

$$T \geq \lim_{s\uparrow 0}\lim_{n\to\infty}\frac{1}{s}|\mathcal{V}|^c|\mathcal{Y}|\frac{\log(n)}{n} + \frac{1}{s}[H(\xi^*) - cH(\gamma) + \sum_{a,o}\xi^*(a,o)\log P_{Y|Z}(o|a)]$$

$$-\frac{1}{n}\log\left(\sum_\kappa 2^{n[H(\kappa)-H(\delta^*)-cH(\lambda)+cH(\gamma)}2^{n\sum_{b,o}\rho(b,o)\log P_{Y|Z}(o|b)}2^{-n\sum_{a,o}\xi^*(a,o)\log P_{Y|Z}(o|a)}\right) \tag{3.42}$$

We see that unless $[H(\xi^*) - cH(\gamma) + \sum_{a,o}\xi^*(a,o)\log P_{Y|Z}(o|a)] \geq 0$, the first term goes to ∞ as $s \uparrow 0$. There is only one choice of ξ that achieves this, and that must be the supremizing ξ^*. This means that $\xi^* = [\gamma^c \times P_{Y|Z}] = [\gamma \times \gamma \times \ldots \times \gamma \times P_{Y|Z}]$. The corresponding $\delta^* = [\gamma \times \gamma \times \ldots \times \gamma]$.

Then in the second term, $2^{nH(\delta^*)}$ equals $2^{ncH(\gamma)}$.Choosing a supremizing κ^* and the fact that there are only a polynomial number of joint types κ, gives something larger than the sum - now required because of the '$-$' sign before the $\log$. κ^* is,

$$\kappa^* = \arg\sup_\kappa H(\kappa) + \sum_{b,o}\rho(b,o)\log P_{Y|Z}(o|b) \tag{3.43}$$

Replacing the supremizing κ^* from (3.43) in (3.42)

$$T \geq \lim_{s\uparrow 0} \lim_{n\to\infty} \frac{1}{s} |\mathcal{V}|^c |\mathcal{Y}| \frac{\log(n)}{n} \tag{3.44}$$

$$- \frac{1}{n} \left(|\mathcal{V}|^{2c} \log(n+1) + nH(\kappa^*) - ncH(\lambda) + n \sum_{b,o} [\rho^*(b,o) - \xi^*(b,o)] \log P_{Y|Z}(o|b) \right) \tag{3.45}$$

where $\rho^*(b,o) = \sum_a \kappa^*(a,b) \frac{\xi^*(a,o)}{\delta^*(a)}$.

$$T \geq \lim_{s\uparrow 0} \lim_{n\to\infty} \frac{1}{s} |\mathcal{V}|^c |\mathcal{Y}|] \frac{\log(n)}{n} + \frac{|\mathcal{V}|^{2c}}{n}$$
$$- \left(H(\kappa^*) - cH(\lambda) + \sum_{b,o} [\rho^*(b,o) - \xi^*(b,o)] \log P_{Y|Z}(o|b) \right) \tag{3.46}$$

Taking $\lim_{n\to\infty}$ the first two terms go to zero.

$$T \geq -H(Y|Z) + cH(\lambda) - H(\kappa^*) - \sum_{b,o} \rho^*(b,o) \log P_{Y|Z}(o|b) \tag{3.47}$$

where

$$\kappa^* = \arg\sup_{\kappa} H(\kappa) + \sum_{b,o} \rho^*(b,o) \log P_{Y|Z}(o|b) \tag{3.48}$$

Replacing this value of κ^* in (3.47) gives us the large deviation threshold that is the numerator of (3.13) in Theorem 1. Replacing this value of T in (3.13) gives us our result.

■

3.3.2 The arbitrary connections ensemble

Moving on to the arbitrary connections model, we first consider the situation where all the measurement functions are identical and c is constant. Each measurement is a function of a subset of c locations as described in Section 3.1.3.1. We assume a Maximum Likelihood (ML) decoder that maximizes $s^{\mathcal{C}}(\mathbf{x}_m, \mathbf{Y}) = P_{\mathbf{Y}|\mathbf{X}}(\mathbf{Y}|\mathbf{x}_m)$.

Corollary 3 ([32] ML decoding) *There exists a sequence of sparse measurement structures $\mathcal{C}_n$, such that the average probability of error, $P_{e,\mathcal{C}_n} \to 0$ as $n \to \infty$, with ML*

decoding, for fixed R, as long as $R < C_{LB}^{ML}(D)$, where

$$C_{LB}^{ML}(D) = \min_{\lambda \notin \mathcal{S}^{\gamma}(D)} \sup_{w \geq 0} \frac{\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} P^{\gamma}(x) P_{Y|X}(y|x) \log \left(\frac{\left(P_{Y|X}(y|x)\right)^{w}}{Q^{\lambda,w}(y|x)} \right)}{[H(\lambda) - H(\gamma)]} \tag{3.49}$$

Proof: We define the various terms that occur in Theorem 1:

$$s^{\mathcal{C}}(\mathbf{x}_m, \mathbf{Y}) = \prod_{u=1}^{n} P_{Y|X}(Y^u | x_m^u) \tag{3.50}$$

$$Q^{\lambda,w}(y|x) =$$

$$\sum_{x' \in \mathcal{X}} P^{\lambda}(X' = x' | X = x) \left(P_{Y|X}(Y = y | x = x')\right)^{w} \tag{3.51}$$

$$f^{\lambda,w}(\mathbf{X}_m, \mathbf{Y}) = \prod_{u=1}^{n} Q^{\lambda,w}(Y^u | X_m^u) \tag{3.52}$$

$$Z_n^{\lambda} = \frac{1}{n} \sum_{u=1}^{n} \log \left(\frac{\left(P_{Y|X}(Y^u | X_m^u)\right)^{w}}{Q^{\lambda,w}(Y^u | X_m^u)} \right) \tag{3.53}$$

Each of the terms in Z_n^{λ}, $Z^u = \log \left(\frac{\left(P_{Y|X}(Y^u|X_m^u)\right)^{w}}{Q^{\lambda,w}(Y^u|X_m^u)} \right)$ are i.i.d. Thus, we are seeking a large deviations result for the average of i.i.d terms and we can use Cramer's theorem [34]. Hence, the condition in Theorem 1 is met and $\Lambda^{\lambda,w}(s) = \log E[e^{sZ^1}]$. Now, if $x < E(Z^1)$, then $\Lambda^{*\lambda}(x) \geq 0$. So the resulting condition is that $T(\lambda) \leq E\left(\log\left(\frac{\left(P_{Y|X}(Y|X)\right)^{w}}{Q^{\lambda,w}(Y|X)}\right)\right)$. Using this threshold gives us the result. ■

We consider bounds on (3.49). Rather than optimizing over w if we set $w = 1$

$$C_{LB}(D) = \min_{\lambda \notin \mathcal{S}^{\gamma}(D)} \frac{\mathcal{D}(P_{Y|X} || Q^{\lambda})}{[H(\lambda) - H(\gamma)]} \tag{3.54}$$

where $\mathcal{D}(p||q)$ is the KL-divergence [97] between two distributions p and q. Our bound in (3.49) is tighter than the bound (3.54) which was derived in [32].

For a further specialization of the theorem we assume the ideal coding case, where the codewords $\mathbf{X}$ are actually i.i.d. Then the KL-divergence becomes a mutual information

$I(X;Y)$, independent of the joint type λ, resulting in

$$C_{LB} = I(X;Y) \; \forall \; D < \min_{a \in \mathcal{V}} \gamma^a \tag{3.55}$$

which is Shannon's expression for channel capacity [97]. Thus, the sensing capacity in Theorem 1 and in [32] can be considered generalizations of the channel capacity.

Before we continue we introduce a mapping that will be useful in generalizing some of our results. This mapping illustrates how we can combine the measurement function and the noise model into a single distribution. Suppose we have a measurement function Ψ and a noise model $P_{Y|X}$. We define a single vector output measurement function $\Psi^G\left(v^{l(1)}, \ldots, v^{l(c)}\right) = \left[v^{l(1)}, \ldots, v^{l(c)}\right] = Z$ which has an output alphabet $\mathcal{Z} = \mathcal{V}^c$. We can then define

$$P^G_{Y|Z}\left(Y = y | Z = z\right) = P_{Y|X}(y|\Psi^t(Z)) \tag{3.56}$$

to be the noise model for this measurement Ψ^G. The noise model $P_{Y|Z}$ with measurement Ψ is equivalent to the noise model $P^G_{Y|Z}$ with measurement Ψ^G.

3.3.3 Model mismatch at the decoder in the arbitrary ensemble

The next specialization we consider is the case where there is mismatch of the decoder to system parameters which was described in Section 3.1.3.4. We consider the arbitrary ensemble. Each measurement is a function of a subset of c locations. However, instead of the true noise model $P_{Y|X}$, the decoder uses a similarity score $s^{\mathcal{C}}\left(\mathbf{x}, \mathbf{y}\right) = \prod_{i=1}^n s(x^i, y^i)$. For example, if the noise model is mis-calibrated, then $s(x,y) = \widehat{P}_{Y|X}(y|x)$. We run a decoder using this score. The decoder returns the index m of the input $\mathbf{v}_m$ that maximizes $s^{\mathcal{C}}\left(\mathbf{x}, \mathbf{y}\mathbf{Y}\right) = \prod_{u=1}^n s\left(x^u_m, Y^u\right)$.

Corollary 4 (Decoding with a mismatched score) *There exists a sequence of sparse measurement structures $\mathcal{C}_n$, such that the average probability of error, $P_{e,\mathcal{C}_n} \rightarrow 0$ as $n \rightarrow \infty$, with mismatched decoding using a score $s\left(\mathbf{x}_m, \mathbf{Y}\right)$, for fixed R, as long as $R < C^{MIS}_{LB}(D)$,*

where

$$C_{LB}^{MIS}(D) = \min_{\lambda \notin \mathcal{S}^{\gamma}(D)} \sup_{w \geq 0} \frac{\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} P^{\gamma}(x) P_{Y|X}(y|x) \log\left(\frac{s^{w}(x,y)}{Q_d^{\lambda,w}(y|x)}\right)}{[H(\lambda) - H(\gamma)]} \tag{3.57}$$

Proof: In this case we define ,

$$\begin{aligned} s^{\mathcal{C}}(\mathbf{x}_m, \mathbf{Y}) &= s(\mathbf{x}_m, \mathbf{Y}) = \prod_{u=1}^{n} d(x_m^u, Y^u) \\ Q_d^{\lambda,w}(y|x) &= \sum_{x' \in \mathcal{X}} P^{\lambda}(X' = x'|X = x) s^{w}(x', y) \\ f^{\lambda,w}(\mathbf{X}_m, \mathbf{Y}) &= \prod_{u=1}^{n} Q_d^{\lambda,w}(Y^u|X_m^u) \\ Z_n^{\lambda} &= \frac{1}{n} \sum_{u=1}^{n} \log\left(\frac{(s(X_m^u, Y^u))^{w}}{Q_d^{\lambda,w}(Y^u|X_m^u)}\right) \end{aligned} \tag{3.58}$$

Since Z_n^{λ} is again an average of i.i.d terms, and so we now apply Cramer's theorem [34] to calculate the large derivation threshold, which gives us the result. ■

The Csiszar-Korner-Hui bound [99] in coding theory provides a bound on achievable rates for communicating over noisy channels with mismatched decoder. Equation (3.57) is the equivalent of this bound modified to account for the correlations between codewords $\mathbf{X}$ that is inherent in sparse measurement systems. The bound in (3.57) is smaller than the sensing capacity $C_{LB}^{ML}(D)$ in (3.49), indicating that we require more measurements to compensate for model mismatch. However, the bound shows that we can, in fact, still guarantee arbitrarily small probability of error $P_{e,\mathcal{C}_n}$ as long as a sufficient number of measurements are available. This is an interesting fact that can be exploited in sensing applications of interest - take more measurements than required to make the system robust to mismatch at the decoder. We mention that using the mapping in (3.56), we can also analyze the case where the measurement function Ψ is mis-calibrated. The bound can also be extended to the sparse regular model.

3.3.4 Parameter uncertainty at the decoder in the arbitrary ensemble

The next scenario we study is the case where we have parameter uncertainty.

Corollary 5 (ML decoding with parameter uncertainty) *Suppose that the decoder does not know the true measurement noise model $P_{Y|X}$, but only that it belongs to a finite set $\{P^{\theta}\}$, indexed by $\theta \in \Theta$. Then, there exists a sequence of sparse measurement structures $\mathcal{C}_n$, such that the average probability of error, $P_{e,\mathcal{C}_n} \to 0$ as $n \to \infty$, for fixed R, as long as $R < C_{LB}^{UN}(D)$ where $C_{LB}^{UN}(D)$ is a lower bound on sensing capacity with uncertainty given by,*

$$C_{LB}^{UN}(D) = \min_{\lambda \notin \mathcal{S}^{\gamma}(D)} \frac{\min_{\theta_t \in \Theta} \min_{\theta' \in \Theta} E^{\theta_t}\left(\log\left(\frac{\left(P_{Y|X}^{\theta_t}(Y|X)\right)}{Q_{\theta'}^{\lambda}(Y|X)}\right)\right)}{[H(\lambda) - H(\gamma)]} \tag{3.59}$$

Proof: We choose a decoder that maximizes the following score.

$$s^{\mathcal{C}}(\mathbf{x}_i, \mathbf{Y}) = \frac{1}{|\Theta|}\sum_{\theta \in \Theta} P_{\theta}(\mathbf{Y}|\mathbf{x}_i) = \frac{1}{|\Theta|}\sum_{\theta \in \Theta}\prod_{u=1}^{n} P_{\theta}(Y^u|x_i^u) \tag{3.60}$$

This results in,

$$f^{\lambda}(\mathbf{X}_i, \mathbf{Y}) = \frac{1}{|\Theta|}\sum_{\theta \in \Theta}\prod_{u=1}^{n} Q_{\theta}^{\lambda}(Y^u|X_i^u) \tag{3.61}$$

$$Z_n^{\lambda} = \frac{1}{n}\log\left(\frac{\sum_{\theta \in \Theta}\prod_{u=1}^{n}\left(P_{\theta}(Y^u|X_i^u)\right)}{\sum_{\theta' \in \Theta}\prod_{u=1}^{n} Q_{\theta'}^{\lambda}(Y^u|X_i^u)}\right) \tag{3.62}$$

This is not the average of i.i.d. terms, and so, we use a bounding procedure to simplify the expression. Let θ_t be the true θ that actually occurred.

$$Z_n^{\lambda} \geq \frac{1}{n}\log\left(\frac{\prod_{u=1}^{n}\left(P_{\theta_t}(Y^u|X_i^u)\right)}{\sum_{\theta' \in \Theta}\prod_{u=1}^{n} Q_{\theta'}^{\lambda}(Y^u|X_i^u)}\right) \tag{3.63}$$

$$\geq \frac{1}{n}\log\left(\frac{\prod_{u=1}^{n}\left(P_{\theta_t}(Y^u|X_i^u)\right)}{|\Theta|\max_{\theta'}\prod_{u=1}^{n} Q_{\theta'}^{\lambda}(Y^u|X_i^u)}\right) \tag{3.64}$$

$$\geq -\frac{\log(|\Theta|)}{n} + \min_{\theta' \in \Theta}\frac{1}{n}\sum_{u=1}^{n}\log\left(\frac{P_{\theta_t}(Y^u|X_i^u)}{Q_{\theta'}^{\lambda}(Y^u|X_i^u)}\right) \tag{3.65}$$

$$\geq -\epsilon + \min_{\theta' \in \Theta} S_n^{\lambda,\theta_t,\theta'} \tag{3.66}$$

The last inequality is true for any ϵ for n large enough. Each term in $S_n^{\lambda,\theta_t,\theta'}$ is i.i.d as before. Since $Z_n^{\lambda} \geq \min_{\theta' \in \Theta} S_n^{\lambda,\theta_t,\theta'} - \epsilon$, $P\left(Z_n^{\lambda} < T(\lambda)\right) \leq P\left(\min_{\theta' \in \Theta} S_n^{\lambda,\theta_t,\theta'} < T(\lambda) + \epsilon\right)$, for

n large enough such that $\frac{\log(|\Theta|)}{n} \leq \epsilon$. Using a union bound over θ', this goes to zero if $T(\lambda) < \min_{\theta_t \in \Theta} \min_{\theta' \in \Theta} E^{\theta_t}\left(\log\left(\frac{\left(P^{\theta_t}_{Y|X}(Y|X)\right)}{Q^{\lambda}_{\theta'}(Y|X)}\right)\right)$, and so, we have proved the corollary. ■

Let θ^* be the minimizing θ_t in Corollary 5. If the set of noise models $P^{\theta}_{Y|X}$ is convex, the corollary suggests a simplified decoder that just assumes the model $P^{\theta^*}_{Y|X}$. This decoder would achieve the same *worst-case* performance as the suggested decoder with uncertainty, but would have a lower computational complexity. We will also show in Section 5.4 how to use the proof technique from this corollary to design a decoding algorithm that performs well under uncertainty for the case where the set of noise models is not convex.

Note that, to simplify notation, we chose $w = 1$. w can be optimized to obtain better bounds. Also, using the mapping in (3.56), we can analyze the case where the measurement function Ψ is uncertain.

3.3.5 Heterogeneous arbitrary ensembles

We now derive an extension that is useful for systems that have different kinds of measurement such as measurements with different values of c.

Corollary 6 ([32] ML decoding, heterogeneous measurements) *Assume that there are M classes of measurements, and each class $t \in \{1, \ldots, M\}$ has its own measurement function $\Psi^t\left(v^{l(1)}, \ldots, v^{l(c)}\right)$ and corresponding noise model $P^t_{Y|X}$. Let there be a constant fraction $\alpha_t = \frac{n_t}{n}$ of measurements of class t. There exists a sequence of heterogeneous sparse measurement systems $\mathcal{C}^{HET}_n$, such that the average probability of error, $P_{e,\mathcal{C}^{HET}_n} \to 0$ as $n \to \infty$, with ML decoding, for fixed R, as long as $R < C^{HET}_{LB}(D)$, where*

$$C^{HET}_{LB}(D) = \min_{\lambda \notin \mathcal{S}^{\gamma}(D)} \sup_{w_t \geq 0} \sum_{t=1}^{M} \alpha_t \frac{\sum_{z \in \mathcal{Z}} \sum_{y \in \mathcal{Y}} P^{\gamma}(x) P^t_{Y|Z}(y|z) \log\left(\frac{\left(P^t_{Y|Z}(y|z)\right)^{w_t}}{Q^{t,\lambda,w_t}(y|z)}\right)}{[H(\lambda) - H(\gamma)]} \quad (3.67)$$

where $Z = \left[v^{l(1)}, \ldots, v^{l(c)}\right]$ and

$$P^t_{Y|Z}(Y = y|Z = z) = P^t_{Y|X}(y|\Psi^t(Z)) \quad (3.68)$$

Proof: We outline the proof. In Theorem 1, we can account for the heterogeneity using the mapping in (3.56). We define a single vector output measurement function $\Psi\left(v^{l(1)},\ldots,v^{l(c)}\right) = \left[v^{l(1)},\ldots,v^{l(c)}\right] = Z$ which has an output alphabet $\mathcal{Z} = \mathcal{V}^c$. Then we assume M noise functions - one for each kind of measurement. We now define $P^t_{Y|Z}\left(Y = y|Z = z\right) = P^t_{Y|X}(y|\Psi^t(z))$ as in the corollary. This mapping is useful because it unifies the noise model $P^t_{Y|X}$ and the measurement function Ψ^t into one noise model $P^t_{Y|Z}$. We use an ML decoder that returns the index $\hat{m} = \arg\max_m P_{\mathbf{Y}|\mathbf{X}}\left(\mathbf{Y}|\mathbf{x}_m\right)$. We then define the various terms that occur in Theorem 1. Instead of n i.i.d. terms we have M independent quantities each of which is the average of n_t i.i.d. terms, for $t = 1,\ldots,M$. We apply Cramer's theorem and the result follows. ■

3.3.6 Comparison of the lower bound on sensing capacity and algorithm performance

We check the utility of the derived lower bounds on sensing capacity by comparing them with the performance of a practical decoder in simulations. We consider the application of sparse regular structures to group testing for genetic screening as described in Section 2.2.2 and use the L1 decoder [19] as an example of a practical decoder motivated by the work of [81].

We recall the model. In our simulation of a genetic screening application the inputs are 0,1 and 2, corresponding to normal, carriers and effected individuals. We consider the limiting case where there are no errors in the pooling outputs, that is, $P_{Y|X}(Y = y|X = x) = \mathbf{1}(x - y)$. In Figure 3.3 we show the performance of the sparse pooling design when the fraction of normal individuals is $p_V(V = 0) = 1 - p = 0.9$ and when it is $p_V(V = 0) = 1 - p = 0.99$. The alternate alleles are distributed according to the Hardy-Weinberg Law [77]. We define an error only when a 1 or 2 is decoded to a 0 (or vice versa),

since these are the most serious errors. So the distortion between $\mathbf{v}$ and $\mathbf{v}'$ is

$$d_G(\mathbf{v}, \mathbf{v}') = \sum_{i=1}^{k} [\mathbf{1}_{v^i=0, v'^i \neq 0} + \mathbf{1}_{v^i \neq 0, v'^i=0}] \tag{3.69}$$

Note that our bound is general enough to include such non-Hamming definitions of distortion. We simulate a range of pools sizes c to show that our bounds are meaningful. We compare the performance of the L1 decoder with our bound from (3.29).

Two interesting conclusions can be drawn from Figure 3.3. The first is that the bound and algorithm show the same trends across a range of model parameters. The second conclusion is that as the probability of alternate alleles falls, the L1 decoder performs almost as well as (our analysis of) the ML decoder. This indicates that our bound can be used to design practical pooling strategies to evaluate the effectiveness of various design choices such as (i) pool size, (ii) use of bar-coding technologies [100] or (iii) the number of reads per lane [82]. From another point of view these results also indicate the power of the L1 decoder with sparse inputs in low-noise conditions. Identifying when algorithm performance is 'good' is one of the most important uses of our theoretical bounds.

3.4 A CONVERSE : AN UPPER BOUND ON THE SENSING CAPACITY

This section presents a simple converse to complement the general achievability result in Theorem 1. A converse is an upper bound on rates for which arbitrarily small error rates $P_{e,\mathcal{C}_n}$ are possible. Converses are usually hard to prove because we must show that no structure can have better performance. At the end of the section we compare the converse to the (achievable) lower bound on measurement capacity.

Theorem 7 (Sensing capacity : Converse) *Suppose inputs V^i are drawn i.i.d. according to a p.m.f. γ and the measurement function is Ψ. If there exists a sequence of measurement structures drawn from a sparse measurement ensemble such that the average probability of error, $P_{e,\mathcal{C}_n} \to 0$ as $n \to \infty$ for fixed R, then $R < C_{UB}(D)$, where $C_{UB}(D)$ is an upper*

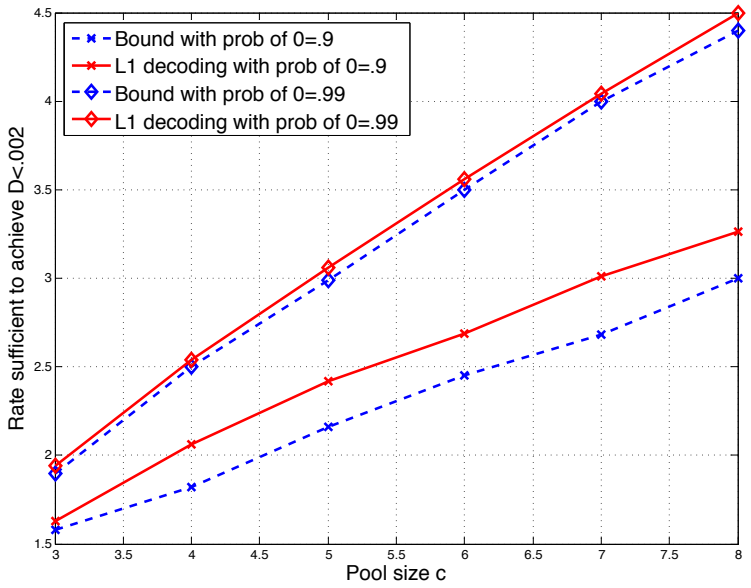


Figure 3.3: Comparison of the bound in Theorem 2 with the performance of the L1 decoder for a range of model parameters. All results are for allowable distortion $D = .002$ with no noise.

bound on sensing capacity, given by,

$$C_{UB}(D) = \begin{cases} \max_{\delta} \frac{I(X;Y)}{[H(\gamma)-H(D)]} & \text{, if } H(\gamma) > H(D) \\ \infty & \text{, otherwise} \end{cases} \tag{3.70}$$

Here $X \sim \Psi(Z)$ and $Z \sim \delta$. δ is a distribution over $\mathcal{V}^c$ constrained by the choice of measurement structure. For example, for sparse regular measurement structures,

$$\sum_{\mathbf{e} \in \mathcal{V}^c} \frac{|\{i : e^i = a\}|}{c} \delta(\mathbf{e}) = \gamma^a \forall\ a \in \mathcal{V} \tag{3.71}$$

Proof: We outline the proof. Consider the Markov chain corresponding to the sparse measurement system,

$$\mathbf{V} \rightarrow \mathbf{X} \rightarrow \mathbf{Y} \rightarrow \widehat{\mathbf{V}} \tag{3.72}$$

By the source channel separation theorem (with distortion) [101],

$$R \leq \begin{cases} \frac{I(\mathbf{X};\mathbf{Y})}{[H(\gamma)-H(D)]} & \text{if } H(\gamma) > H(D) \\ \infty & \text{otherwise} \end{cases} \tag{3.73}$$

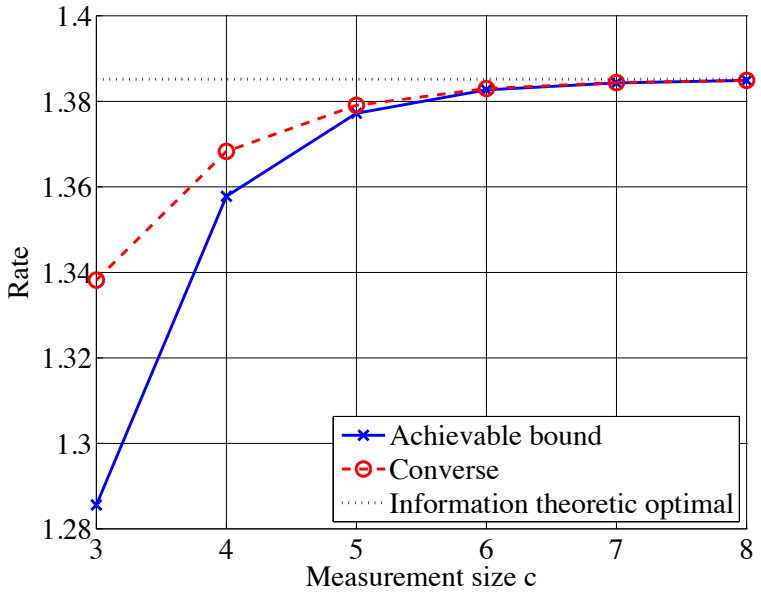


Figure 3.4: Comparison of converse and achievable bound of noiseless XOR with no distortion as measurement size c changes. The information theoretic optimal for lossless compression at $P_V(V = 1) = p = 0.2$, which is $\frac{1}{H(p)}$ is also shown.

Since $\mathbf{X}$ is generated from $\mathbf{V}$ by a sparse measurement ensemble, $X^u = \Psi(Z^u) \; \forall \; u$. Recall that Z^u was defined in Section 3.2.2. For $P_{e,\mathcal{C}_n} \to 0$, we must have

$$\frac{1}{nRH(\gamma)} \sum_{m:\mathbf{v}_m \text{of type } \gamma} P_{e,\mathcal{C}_n,m} \to 0 \tag{3.74}$$

That is, probability of error must go to zero for typical inputs. For typical inputs, the type of $\mathbf{Z}$ is constrained by the measurement ensemble as described in Section 3.3. This results in the theorem. ■

3.4.1 Comparison of bounds on achievability and the converse

We now compare the converse in Theorem 7 and the achievable result for sparse regular measurements in Corollary 2, in two cases of interest.

Example A : Lossless and noiseless compression with XORs. We study the problem of linear (in $GF(2)$) lossless compression because it is of interest in the problems of dis-

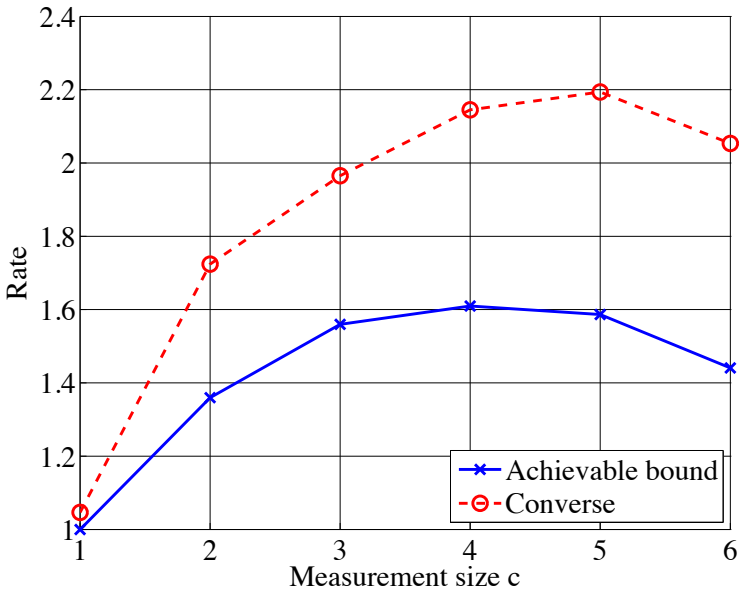


Figure 3.5: Comparison of converse and achievable bound of SUM measurements with $p_V(V = 1) = p = 0.1$, 90 reads noise model and allowed distortion $D = .002$.

tributed compression [11] and network data compression [33]. We consider the following scenario. The inputs to the measurements V^i are binary, $\mathcal{V} = \{0, 1\}$. They are distributed i.i.d. according to p_V with $p_V(V = 1) = p = 0.2$. We consider the use of sparse regular measurements with the function Ψ chosen as the XOR of inputs. We look at the case of no noise, to allow for lossless compression.

Figure 3.4 compares the achievable bound, the converse for sparse XOR measurements and information theoretic bound for lossless compression, which is the inverse of the entropy $= \frac{1}{H(p)}$. We see that our bounds for sparse measurements are very close together (by $c = 8$, they are numerically identical) and both are very close to the information theoretic bound of $\frac{1}{H(p)}$ for large enough c. Notice that the information theoretic bounds assumes optimal codes which may be computationally inefficient, whereas our measurement structure us a sparse linear code, which allows efficient encoding and Belief Propagation (BP) decoding. See Appendix B for a discussion of BP as it applies to sparse measurement

systems.

Example B : Sparse regular pooling designs for genetic screening. We return to the genetic screening application from Section 2.2.2. For this simulation we consider the case where inputs are binary 0 or 1 corresponding to normal people and carriers. Thus, $\mathcal{V} = \{0, 1\}$. This is realistic for diseases where symptoms make effected people easy to recognize without blood tests. We use the reads noise model from Chapter 2.

We recall the definition of the reads noise model. Suppose that in a particular pool $x \in \mathcal{X}$ fraction of the mutated DNA sample is present with $1 - x$ fraction of normal DNA. In this case, for a pool size c, $\mathcal{X} = \{\frac{0}{c}, \ldots, \frac{c}{c}\}$. The NGST device draws strands from this mixture to sequence. The output of each sequencing run is called a read. Suppose the sequencer takes r reads. Then, the probability of each (noiseless) read being mutated is x. This corresponds to our model with measurement function $\Psi(v^1, \ldots, v^c) = \frac{1}{c}\sum_{i=1}^{c} v^i$ - the AVERAGE function. The probability that y out of the r reads are mutated is then,

$$P_{Y|X}(y|x) = \binom{r}{y} x^y (1-x)^{r-y} \tag{3.75}$$

Here y represents the number of output reads (out of a possible r reads) that are mutant. Thus, $\mathcal{Y} = \{0, \ldots, r\}$.

We compare the bounds for the following representative parameter settings in Figure 3.5 - $r = 90$ reads, $p_V(V = 1) = p = 0.1$ and allowed distortion $D = 0.002$. Here we see that both the converse and achievable bounds show similar trends with pool size c. The most important conclusion is that there exists an optimal pool size. This suggests that sparse pools are indeed required, since dense pools would have a low rate, thus requiring a large number of measurements.

3.5 COMPLEXITY OF COMPUTING THE BOUNDS

The bounds in Sections 3.2.3 and 3.3 provide analytical insights into the performance of sparse measurement systems as compared to a pure simulation based approach. Furthermore, they are computationally cheaper to compute than running time consuming simulations. This reduction in complexity is because they do not involve any explicit decoding operations. Thus, the complexity of computing the bounds is independent of the size of the problem (n, k), which is significant saving when compared to the simulation based approach.

The main complexity in computing the bounds arises from storing the joint distribution between the inputs $Z \in \mathcal{V}^c$ and the noisy outputs Y of a single measurement, and computing the various extremal distributions in the bounds (such as the optimal λ and κ in (3.29)). Since in general, even storing the table of an arbitrary measurement function $\Psi(v^1, \ldots, v^c)$ is exponential in c, the complexity of computing the bounds can be exponential in c.

Fortunately, many measurement functions that occur in practice such as the SUM function in pooling designs, the OR function in group testing, and the THRESHOLD function in sensor networks are 'type' based. A function Ψ is type based if

$$\Psi(\mathbf{v}) = \Psi(\gamma) \tag{3.76}$$

where γ is the type (or empirical p.m.f.) of $\mathbf{v} \in \mathcal{V}^c$. This means that the output depends only on the type (number of 0s, 1s and so on) of the input vector seen by the measurement and not on the actual vector value. The type of a vector can take only a polynomial (in its length) number of values [98]. In particular the type of a vector $\mathbf{v}$ of length c takes less than $(c+1)^{|\mathcal{V}|}$ values. This reduces the description length of the measurement function substantially. However, note that the bounds in (3.29) and (3.49) also depend on the distortion between pairs of c-length vectors. So, we also need to show that the constraints and optimization

objectives in these two expressions can be written in terms of a polynomial number of variables.

We now proceed to describe a choice of variables for (3.29). The original optimization problem (for a fixed λ) is over the joint κ of two vectors $\mathbf{v}, \mathbf{v}'$ of length c. This means that κ is a distribution over $|\mathcal{V}|^{2c}$. Instead we introduce one variable for each pair wise type and distortion between them resulting in at most $(c+1)^{3|\mathcal{V}|}$ variables, for the case of Hamming distortion. The key fact is that the optimization problem is a maximum entropy problem, where we are trying to maximize $H(\kappa)$ subject to constraints. Any two vectors with joint type λ have the same distortion and will be appear in the same constraints. They will also have equal probability in the optimizing κ since the *uniform distribution maximizes the entropy when subject to constraints only on the marginals*.

Without this symmetry, offered by types, we can compute the bound for values of c upto 6, say, as seen in the examples in [32]. Using the symmetry we can compute the bound for SUM measurements (where the output alphabet increases with c) for c upto 20. For binary measurements like XOR or OR we can compute the bound for c upto 40. The type-based speedup can also be extended to measurements where the output depends on types of a small number of subsets (such as WEIGHTED SUM with few different weights) with a corresponding increase in complexity. We will also see how to exploit type symmetric functions in algorithm design when we look at the Belief Propagation (BP) algorithm in Appendix B.

CHAPTER 4

ALGORITHMS FOR SPARSE MEASUREMENT SYSTEMS

In Chapter 2, we saw how sparse measurement systems modeled by factor graphs provide a common framework for the problems we are interested in. In Chapter 3, we showed how information theoretic and combinatorial techniques could be used to analyze the fundamental limits of these sparse measurement systems. Unfortunately, the decoders that we assumed in obtaining these limits are practically infeasible for reasonable problem sizes. In this chapter we study practical decoding algorithms for large-scale detection problems.

The primary advantage of the factor graph framework is that we can draw on the large body of work on inference algorithms in graphical models to design detection algorithms for sparse measurement systems. Inference algorithms [3, 51] can be exact or approximate. We discuss optimal inference algorithms in Appendix A and an approximate algorithm from the graphical model literature, called Belief Propagation (BP) in Appendix B. In Appendix B we also discuss how the method of Density Evolution can be used to analyze the performance of BP in sparse measurement systems and how to significantly speed up the BP calculations for sparse measurements that have a certain 'type' based symmetry. This allows us to compare the performance of BP with other approximate algorithms for

large measurement sizes c.

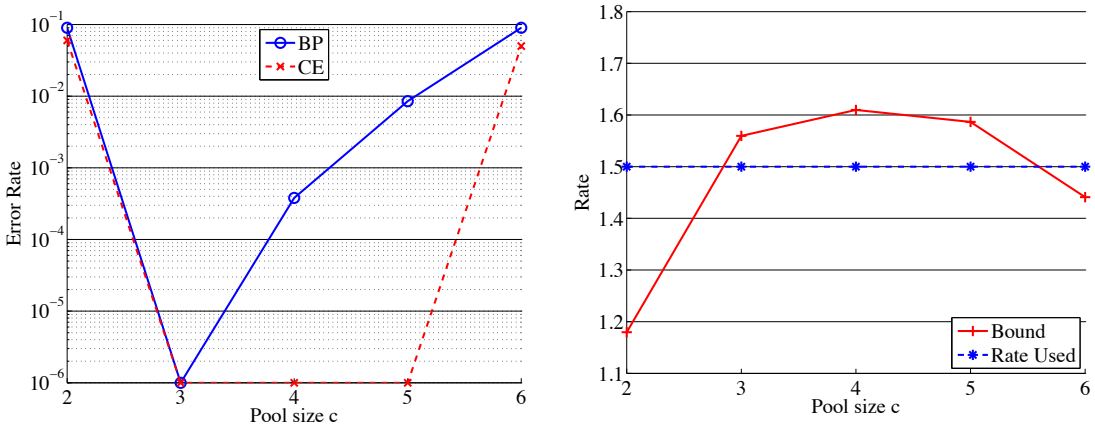
As opposed to BP, we focus on two different algorithms in this chapter. The first algorithm we look at is the Cross Entropy (CE) method, in Section 4.1. This algorithm was initially developed in the area of optimization and rare event simulation [102] and we discuss how it can be applied to the problem of detection in sparse measurement systems. We then study how to use the information theoretic analysis of Chapter 3 to design a stopping criterion for the CE method. We study the performance of the CE method in simulated group testing experiments.

In Section 4.2 we study sequential decoding (SD), a search algorithm originally developed for the decoding of convolutional codes [60]. We show that SD is particularly suited for detection in problems involving contiguous sparse measurement structures, such as those that arise in sensor networks. We then develop an expression for the computational properties of SD [28, 103]. This expression clearly shows how the computational complexity of SD depends on various system parameters like the noise level, the priors and the measurement function used.

4.1 THE CROSS ENTROPY METHOD

Sampling algorithms have a rich history and have found applications in many areas including artificial intelligence [104], statistical physics [105] and combinatorial optimization [106]. Sampling and Markov Chain Monte Carlo (MCMC) methods are general, fast and accurate numerical methods for calculating the expectation of some function with respect to a probability distribution. Detection problems can be written as expectations, and so, can be solved using sampling methods [104].

Before we get into details of the sampling method we study in this thesis, we discuss why we need a new algorithm, given that there exist popular message passing (local) algo-



(a) Comparison of performance of BP and the CE method

(b) Bound for the same parameter values of p and reads

Figure 4.1: Comparing algorithm performance with the bound indicates that BP may be suboptimal for larger c resulting in the search for new algorithms for inference with sparse measurements.

rithms such as Belief Propagation (BP) (described in Appendix B). One reason is shown in Figure 4.1, which displays the performance of BP in a simulated pooling problem with sparse regular measurements. We simulate binary inputs, with $p_V(V = 1) = p = 0.1$, and the reads noise model from Section 2.2.3, with reads $r = 90$ and distortion $D = 0.002$. We compare the performance of the algorithms with our bound from (3.29).

In Figure 4.1(b) we see that the bound indicates that a pool size of $c = 6$ is optimal and that there is no substantial difference between $c = 4, 5$ and 6 at these parameter settings. However, the performance of BP in Figure 4.1(a) does not follow this trend. If BP is sub-optimal, then perhaps other algorithms can be found with performance closer to the bound. This search leads us to the Cross Entropy (CE) method, an algorithm whose performance is also shown in Figure 4.1(a). We see that this algorithm follows the trends illustrated in the bounds.

Several sampling algorithms have been proposed in the literature, such as, the Metropolis algorithm from [105], the Gibbs sampling algorithm from [107] and many others [108].

In this thesis we select a relatively new sampling algorithm called the Cross Entropy (CE) method, which was developed for the problem of rare event simulation by [106]. The reasons for this are two-fold. The first is pragmatic, in that, the CE method turned out to be fast, easy to implement and far more accurate than other sampling algorithms in our simulations. The second is that the CE method has a tunable parameter N that can be used to trade-off computational complexity and accuracy. Further we will be able to use our analysis of Chapter 3 to develop a stopping criterion for the CE method applied to detection with sparse measurement systems.

4.1.1 The Cross Entropy method for combinatorial optimization

The CE method was motivated by an adaptive algorithm originally developed for the problem of rare event simulation by [102] and then modified for general optimization problems, as described in some detail in [106]. This was done by converting the deterministic optimization problem into a stochastic problem and then using the rare event simulation algorithm. We take our description of the general CE method from [106].

Suppose we want to maximize a function $S(\mathbf{v})$. The CE method involves translating the problem, by introducing a family of densities $f(:, \mathbf{p})$ parametrized by $\mathbf{p}$. Each sample from $f(:, \mathbf{p})$ represents a possible solution to the optimization problem. We fix a stopping criterion $Stop(\mathbf{p}_t, \mathbf{p}_{t-1}, \mathbf{v}_*) < \epsilon$, where ϵ is a chosen threshold. The algorithm is described in Algorithm 1.

In the particular case of MAP detection in sparse measurement systems, the optimization function is $S(\mathbf{v}) = P(\mathbf{v}|\mathbf{Y})$. Suppose $\mathcal{S}$ is the set of samples $|mathbfV_t$ drawn, that are in the top $1 - \rho$ quantile. The update rule will be

$$p_t^j = \frac{1}{|\mathcal{S}|} \sum_{t \in \mathcal{S}} V_t^j \tag{4.2}$$

Input: N, $\mathbf{p}$, $f(:,\bullet)$, $S(\bullet)$,ρ,$Stop(\bullet)$,ϵ
Output: $\mathbf{v}_*$ with high value of $S(\mathbf{v})$
Define $\mathbf{p}_0 = \mathbf{p}$ as the initial parametrization.
Set $t = 1$ (iteration counter), CONVERGED=0 (flag), $S_{max} = -\infty$, $v_* = \mathbf{0}_{k\times 1}$.
while *!CONVERGED* **do**
  Generate $\mathbf{V}_1, \ldots, \mathbf{V}_N$ i.i.d. from $f(:, \mathbf{p}_{t-1})$.
  foreach $i = 1, \ldots, N$ **do**
    Calculate $S(\mathbf{V}_i)$
  end
  Order them from smallest to biggest $S_1, \ldots, S_N$.
  Let $\hat{\gamma}_t$ be the $(1-\rho)$ sample quantile of performances $\hat{\gamma}_t = S_{\lceil (1-\rho)N \rceil}$.
  if $\hat{\gamma}_t < \gamma$ **then**
    Set $\hat{\gamma}_t = \gamma$.
  end
  if $S_{max} < S_N$ **then**
    $S_{max} = S_N$;
    $\mathbf{v}_* = \mathbf{V_N}$;
  end
  foreach $j = 1, \ldots, n$ **do**

$$p_t^j = \frac{\sum_{i=1}^{N} \mathbb{1}_{S(\mathbf{V}_i)\geq \hat{\gamma}_t} \mathbb{1}_{V_i^j = 1}}{\sum_{i=1}^{N} \mathbb{1}_{S(\mathbf{V}_i)\geq \hat{\gamma}_t}} \tag{4.1}$$

  end
  if $Stop(\mathbf{p}_t, \mathbf{p}_{t-1}, \mathbf{v}_*) < \epsilon$ **then**
    CONVERGED=1;
  end
  $t = t + 1$;
end

Algorithm 1: Basic CE method for combinatorial optimization.

The stopping criterion can be any choice of convergence, such as, $Stop(\mathbf{p}_t, \mathbf{p}_{t-1}, \mathbf{v}_*) = |\mathbf{p}_t - \mathbf{p}_{t-1}|_1$.

We illustrate the performance of the CE method by comparing it against the BP algorithm of Appendix B for $\rho = 0.2$ and $N = 100,000$. We see in Figure 4.1(a) that the CE method substantially outperforms BP for pool size $c > 3$. BP performs well for $c \leq 3$, since, for small pool sizes the sparse measurement structures are essentially tree like to large depths and BP is optimal for trees. However, for larger pool sizes the CE method seems to be a better choice.

The primary concern when using the CE method is the choice of stopping criterion and the choice of the tuning parameter N. In the next section we draw intuition from the asymptotic analysis in Chapter 3 to derive a stopping criterion and develop a CE method for sparse measurement systems.

4.1.2 A stopping criterion for the cross entropy method

In our large deviation analysis in Chapter 3 we showed that for permutation invariant ensembles, using a score function $s^{\mathcal{C}}(\mathbf{X}_m, \mathbf{Y}) = P(\mathbf{Y}|\mathbf{X}_m)$,

$$\frac{1}{n} \log \frac{P(\mathbf{Y}|\mathbf{X}_m)}{E^{\lambda^*}_{\mathbf{X}_{m'}|\mathbf{X}_m}[P(\mathbf{Y}|\mathbf{X}_{m'}))]} \to T(\lambda^*) \tag{4.3}$$

for the true input m.

We also showed that as long as the rate $R < C_{LB}(D)$, all inputs at a distortion greater than D have a likelihood less than that of the true input. So, at rates below $C_{LB}(D)$ if we find an input with likelihood greater than the threshold indicated by (4.3) with high probability we have found an input within distortion D of the true input. Thus, we can

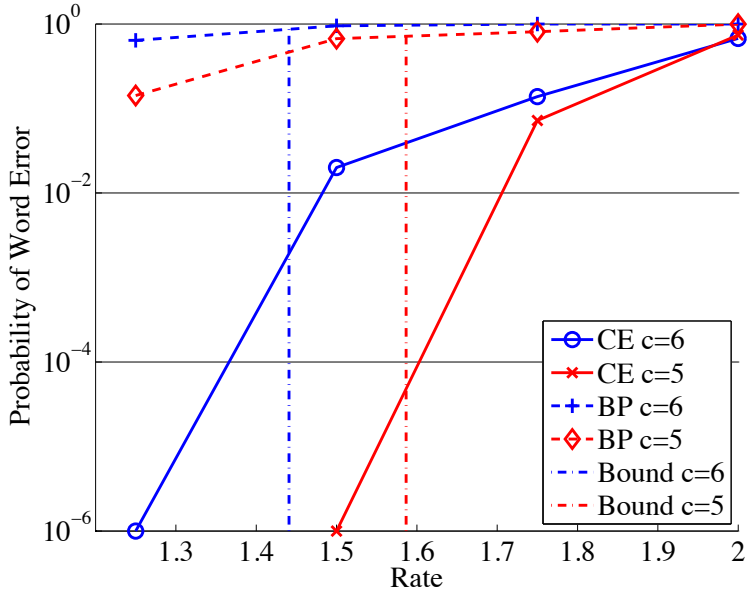


Figure 4.2: Comparison of the performance of the Cross Entropy Method and the bound at different cs with $p_V(V = 1) = p = 0.1$, $k = 200$, 90 reads and $D = .002$. The bottom of the graph corresponds to no word errors in 10^6 runs

choose the following stopping criterion and threshold for the CE method,

$$Stop(\mathbf{p}_t, \mathbf{p}_{t-1}, \mathbf{v}_*) = -\frac{1}{n} \log \frac{P(\mathbf{y}|\mathbf{x}_*)}{E^{\lambda^*}_{\mathbf{X}_{m'}|\mathbf{x}_*}[P(\mathbf{y}|\mathbf{X}_{m'}))]} \tag{4.4}$$

where $\mathbf{x}_*$ is the noiseless measurement corresponding to $\mathbf{v}_*$. The threshold is,

$$\epsilon = -T(\lambda^*) \tag{4.5}$$

This choice results in the modified CE method specified in Algorithm 2.

We now explore how this algorithm performs through simulations.

We already saw in Section 3.3.6 how the lower bound on sensing capacity of sparse regular measurements (3.29) is an excellent predictor of algorithm performance, at least in the noiseless case for sparse inputs. We move on to a noisy case, where the L1 decoder from the previous chapter and [21] fails. We consider simulations of sparse regular measurement structures for the pooling problem from Section 2.2. We simulate the case with binary

Input: N_0, α,$\mathbf{p}$, $f(:,\bullet)$, $S(\bullet)$,ρ,$Stop(\bullet)$,ϵ,NITER
Output: $\mathbf{v}_*$ with high value of $S(\mathbf{v})$
Define $\mathbf{p}_0 = \mathbf{p}$ as the initial parametrization.
Set $t = 1$ (iteration counter), CONVERGED=0 (flag), $S_{max} = -\infty$, $v_* = \mathbf{0}_{k\times 1}$, $N = N_0$.
while *!CONVERGED* **do**
  while $t <$*NITER* **do**
    Generate $\mathbf{V}_1, \ldots, \mathbf{V}_N$ i.i.d. from $f(:, \mathbf{p}_{t-1})$.
    foreach $i = 1, \ldots, N$ **do**
      Calculate $S(\mathbf{V}_i)$
    end
    Order them from smallest to biggest $S_1, \ldots, S_N$.
    Let $\hat{\gamma}_t$ be the $(1-\rho)$ sample quantile of performances $\hat{\gamma}_t = S_{\lceil (1-\rho)N \rceil}$.
    if $\hat{\gamma}_t < \gamma$ **then**
      Set $\hat{\gamma}_t = \gamma$.
    end
    if $S_{max} < S_N$ **then**
      $S_{max} = S_N$;
      $\mathbf{v}_* = \mathbf{V_N}$;
    end
    foreach $j = 1, \ldots, n$ **do**

$$p_t^j = \frac{\sum_{i=1}^{N} \mathbb{1}_{S(\mathbf{V}_i)\geq \hat{\gamma}_t} \mathbb{1}_{V_i^j = 1}}{\sum_{i=1}^{N} \mathbb{1}_{S(\mathbf{V}_i)\geq \hat{\gamma}_t}} \tag{4.6}$$

    end
    $t = t + 1$;
  end
  if $Stop(\mathbf{p}_t, \mathbf{p}_{t-1}, \mathbf{v}_*) < \epsilon$ **then**
    CONVERGED=1;
  end
  $N = \alpha N$;
end

Algorithm 2: Modified CE method, using the theoretically inspired new stopping criterion.

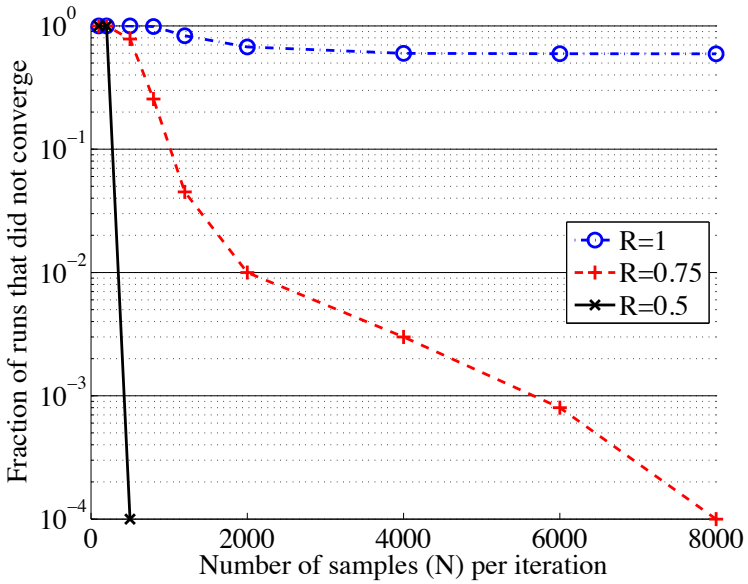


Figure 4.3: Convergence of the modified Cross Entropy method at different rates with $p_V(V=1) = p = 0.4$, $p_f = 0.1$, $k = 200$ and $c = 3$. The bound for these parameters is $C_{LB}^{REG}(D) = 0.76$. The bottom of the figure corresponds to no errors in 10^4 runs. The line for $R = 1 > C_{LB}^{REG}(D)$ corresponds to absolute word error.

inputs, $\mathcal{V} = \{0, 1\}$ which have $p_V(V = 1) = p = 0.1$ and Ψ chosen as SUM. We use the reads noise model, from (2.2) in Section 2.2.3 with reads $r = 90$, input length $k = 200$ and distortion $D = 0.002$. We also plot the bound (3.29) from Chapter 3 in Figure 4.2.

From the results in Figure 4.2, we can see how the performance of BP from Appendix B and that of the modified CE method become highly differentiated at larger values of c. For such values of c, the CE method is by far the better algorithm. We can also see that the bound continues to be an excellent predictor of performance of the modified CE method for the sparse regular structures for larger values of c. We draw this conclusion because for rates below the bound $C_{LB}^{REG}(D)$ the error rate drops substantially while the error rate of BP remains high.

We are interested in not just the performance but also the computational complexity of algorithms in detection for sparse measurement applications. The complexity of the CE

method is determined by the number of samples N used in each iteration, since $N >> k$ we write the complexity as $O(N \log N)$. As a measure of complexity we explore what N is required in our modified CE method.

We simulate the case with binary inputs, $\mathcal{V} = \{0, 1\}$ and $p_V(V = 1) = p = 0.4$ and the measurement Ψ is the SUM of inputs. We use the reads noise model with failures from (2.3) in Section 2.2.3 with reads $r = \infty$ and $p_f = 0.1$, input length $k = 200$ and $c = 3$. Since the number of reads is large, the only source of error is failure of the test. The bound for $C_{LB}^{REG}(D)$, in (3.29) from Chapter 3, for these parameters was 0.76.

In Figure 4.3 we show the fraction of runs for which $Stop(\bullet)$ did not reach the convergence threshold ϵ. We emphasize that in each run the CE method as a whole converged (in that the estimates do not change substantially with more iterations) but may not have found a vector within distortion D of the true input. In the original CE method we would not have known that the optimization had failed but thanks to our analysis and carefully chosen convergence criterion, we can reject the output $\mathbf{v}_*$ in such cases, increase N and try again. An important point to note is that *all runs that met the convergence criterion found a vector within D of the true vector*. This reinforces the importance and usefulness of our asymptotic analysis from Chapter 3, even for structures with k as small as 200.

Another interesting computational property of the CE method can be seen in Figure 4.3. At rates close to the capacity bound our modified CE method requires many runs of the outer 'increase N' loop to converge, whereas at lower Rs - away from capacity - the algorithm found an environment within a distortion D at much smaller Ns - smaller by almost two orders of magnitude. Thus, the computational complexity of the CE algorithm decreases *substantially* once a sufficient number of measurements are collected. This type of 'computational cut-off' phenomenon is reminiscent of the properties of sequential decoding of convolutional codes [62], which we discuss in the next section.

Finally, for rates above capacity we note that we have actually plotted the error rate, as opposed to the fraction that converge, since our threshold is only meaningful at rates below $C_{LB}^{REG}(D)$. Here we see that even for very large N, the CE method fails to find an environment within distortion D on most runs. This hints that our information theoretic analysis is a meaningful predictor of algorithm performance.

From the tone of this section, the reader may get the impression that CE is always the algorithm of choice when compared to BP. However, this is definitely not the case. For many of the results presented here, especially at rates close to the capacity C_{LB}^{REG}, the CE method converges only for a large number of samples - depending on the length of the input k, noise levels and required distortion. However, the complexity of our fast BP (described in Appendix B) is fixed for a given number of iterations and is only linear in the number of measurements. So in the case of small $c = 3$, near zero distortion and rates R near capacity or under strict computational constraints, BP may be preferred. The popularity of BP in decoding of LDPC codes is an excellent illustration of such a situation.

To illustrate this, we simulate the case with binary inputs, $\mathcal{V} = \{0, 1\}$ and $p_V(V = 1) = p = 0.1$ and Ψ chosen as the SUM function. We use the reads noise model, from (2.2) in Section 2.2.3 with reads $r = 90$ and input length $k = 200$. In Figure 4.4 we have plotted average word error rate for BP and CE with small $c = 3$. Here we see that for higher rates, BP does better than CE especially when N (a measure of computation) is small.

The primary advantage of CE from our point of view, is the tunable nature of its performance. We can achieve improved performance for large N at the cost of increased computation. The CE method is fast for non-type based measurement functions where the BP speed up (see Appendix B) is not possible. We will return to the CE method when we (briefly) discuss algorithms for handling uncertainty in Section 4.3. For now we move on to the next class of algorithms for inference in sparse measurement systems, which we will

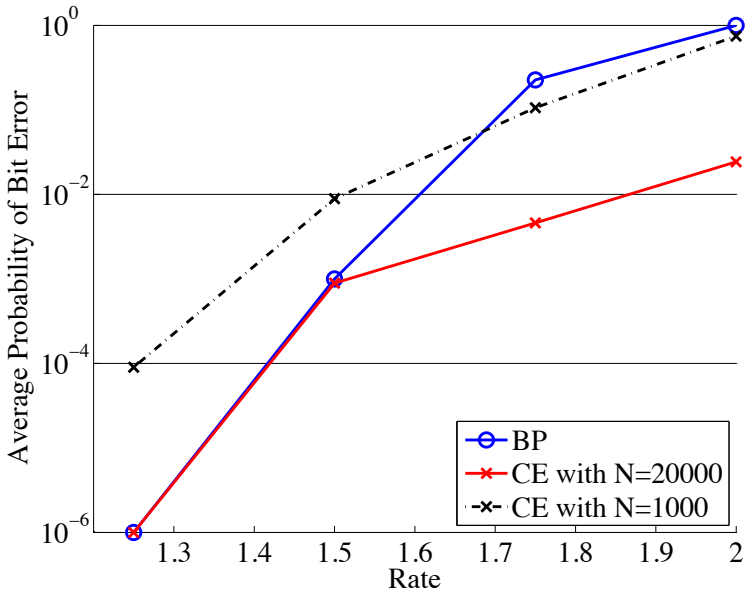


Figure 4.4: Comparison of Belief Propagation and the Cross Entropy Method at different rates with $p = 0.1$, $reads = 90$, $k = 200$ and $c = 3$. The bottom of the figure corresponds to no errors in 10^6 runs.

see are especially suited to sensor network applications.

4.2 SEQUENTIAL DECODING

The term 'large-scale detection' was used in [32] to characterize sensor network detection problems where the number of hypotheses is exponentially large. Examples of such applications include the use of seismic sensors to detect vehicles [109], the use of sonar sensors to map a large room [84] and the use of thermal sensors for high-resolution imaging [15]. In these applications the environment can be modelled as a discrete grid, and each sensor measurement is effected by a large number of grid blocks simultaneously ('field of view' c), sometimes by more than 100 grid blocks [15]. Conventionally, there have been two types of approaches to such problems. The first set of algorithms includes computationally expensive algorithms such as the optimal (in this case also called Viterbi)

decoding as discussed in Section A and belief propagation as discussed in Section B. These algorithms are at least exponential in the size of the field of view of the sensor c, making them infeasible for many common sensing problems. The second set of algorithms make approximations (such as independence of sensor measurements [84]) to make the problem computationally feasible, at the cost of accuracy. Thus, there has existed a trade-off between computational complexity and error rate of detection, with low error rate requiring high complexity algorithms.

Our study in this section is motivated by a completely different trade-off first shown for convolutional codes by [62] and for sensor networks in [32]. We saw one interesting instance of this trade-off in Figure 4.3 for the CE method where increasing the number of measurements (decreasing the rate R) made the algorithm substantially more efficient. Motivated by this we will now study Sequential Decoding (SD) [110], which has been shown to exhibit a similar computational cut-off phenomenon in communication problems. Our interest in this measurements for computation exchange, is that it is a trade-off that may make sense in measurement problems, where we do not have control over the measurement functions, and sensor network applications where fast detection is of the essence [111], as opposed to coding problems.

While [61] developed the possibility of using sequential decoding in sensor networks, the metric used there and in [15], the Fano metric, originated in the decoding of convolutional codes for communication [112]. We will see that the Fano metric [113] is not appropriate for sensor networks. The main contribution of Section 4.2.3 is the derivation of a sequential decoding metric for sensor networks from first principles, following arguments analogous to those used by Fano [113] for convolutional codes. The derived metric differs from the Fano metric, due to the dependence between the measurements that is inherent in sensor networks [32]. We analyze the behavior of this metric and show that it has the

requisite properties for use with sequential decoding.

While the discussion in Section 4.2.3 is somewhat heuristic, relying on the intuition obtained from the analysis of Chapter 3, we then provide a more rigorous understanding of the *computational* properties of sequential decoding. Research in areas such as sensor networks [15, 103] and decoding over ISI channels [114] have shown that many real world search problems can be expressed as decoding on a tree. In Section 4.2.4 we analyze the computational complexity of a particular kind of sequential decoding, called a stack decoder, for the case of specific convolutional code/measurement function/tree code transmitted over a memoryless Gaussian channel.

While the results in this section are analytical, we will return to sequential decoding in Chapter 5. There we will use this analysis to design systems that are easy to decode using SD [115] and also to design sequential decoders that are better suited to a particular sensing or measurement system [103].

4.2.1 Sequential decoding with a stack

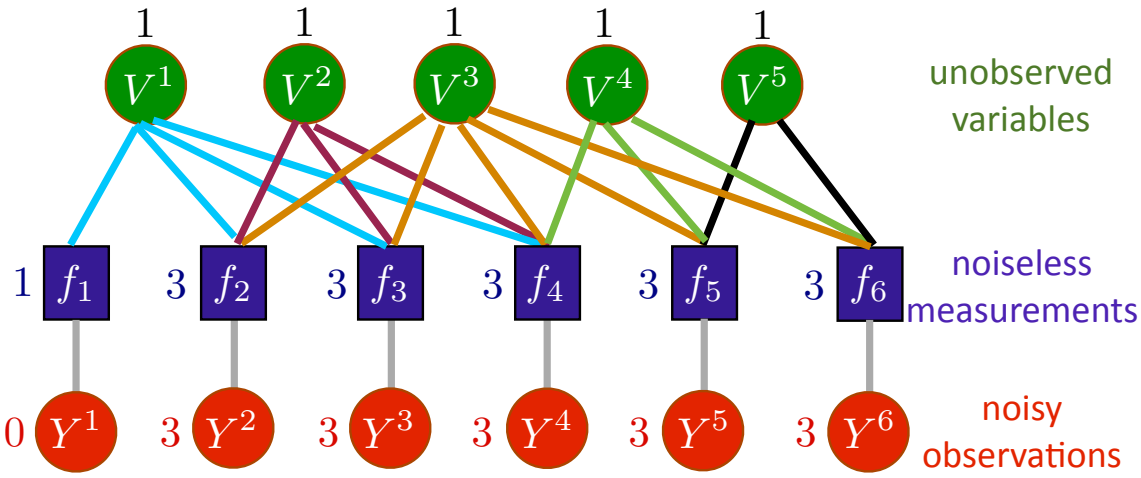
In this thesis, we consider a particular kind of sequential decoding, called stack decoding [60], since it does not require an arbitrary threshold that other variants of SD require. We consider that the measurements form a tree code (such as a convolutional code with large memory or a sensor network using sensors with large field of view c). The code symbols transmitted over a noisy channel correspond to noisy measurements in a sensor network. An example of a sparse measurement system and its corresponding tree are shown in Figures 4.5(a) and 4.5(b). The tree begins at the root node. Each node has $|\mathcal{V}|^U$ children. With a node in the tree we associate the U inputs to the measurement network required to complete at least one measurement. To each branch emanating from that node, we associate one of the $|\mathcal{V}|^U$ possible assignments for the U inputs and the corresponding assignments to

W noiseless outputs. Here W is the number of measurements completed by considering the U inputs. For example, in Figure 4.12, $U = 2$ and $W = 3$. Suppose, that the input length is k and that the number of measurements is n. We define the rate of the tree code exactly as we defined the rate of a measurement network in Chapter 3, as $R = \frac{k}{n} = \frac{U}{W}$. Each possible input vector $\mathbf{v}$ corresponds to one path through the tree. The output vector, read from the branches of the path can be considered the codeword, or noiseless measurements, $\mathbf{x}$. This codeword is transmitted across a channel (such as an Additive White Gaussian Noise (AWGN) channel) to form the noisy measurement vector $\mathbf{Y}$ of length n. In the language of coding theory, $\mathbf{Y}$ would be called the received symbol vector.

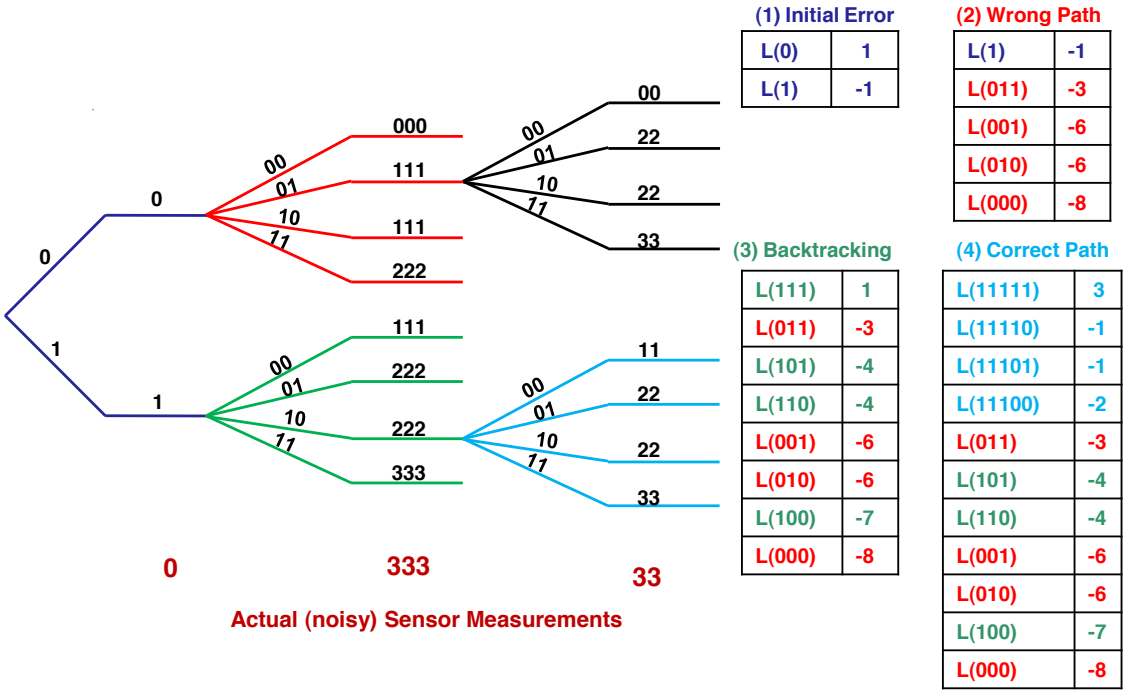
Given a received symbol vector, the decoding problem is to find the transmitted codeword $\mathbf{x}$, or rather the transmitted message $\mathbf{v}$. Thus, we want to reconstruct the input $\mathbf{v}$ given the noisy measurements $\mathbf{y}$. Consider a partial path consisting of the first P transmitted symbols. We first select a metric $\mathbf{\Delta}$ for a partial path

$$\mathbf{\Delta}([x^1, \ldots, x^P], [y^1, \ldots, y^P]) = \sum_{i=1}^{P} \Delta(x^i, y^i) \tag{4.7}$$

which assigns a higher value to the partial paths that are more likely to have generated the partial sequence of the first P measurements $[y^1, \ldots, y^P]$. Having chosen the metric, the stack decoder works as follows. Initially the stack contains just the root node with a metric 0. The decoder removes the top most path from the stack and extends it by hypothesizing all possible choices for the next U bits, resulting in $|\mathcal{V}|^U$ new paths, which are extensions of the original partial path. This is called a node extension of the node under consideration. Each of these new partial paths is assigned an updated metric according to (4.7). These are then inserted into the stack. The stack is re-sorted based on the metric, such that the path with the highest metric is at the top of the stack. The process continues until the topmost path on the stack has the same length as the input vector of the measurement network.



(a) An illustration of a simple contiguous sensor network with noisy measurements



(b) Tree code corresponding to the same sensor network and (possible) first few stack states

Figure 4.5: Stack decoding for sensor networks

We try to explain this more clearly using the simple example in Figures 4.5(a) and 4.5(b). In this case all the inputs were active, denoted by a value of 1. The noiseless output measurements, or codeword, is $\mathbf{x} = [1\ 3\ 3\ 3\ 3\ 3]$. The noisy measurements are $\mathbf{y} = [0\ 3\ 3\ 3\ 3\ 3]$, because of noise in the first measurement. The stack decoder first hypothesizes possible values for V^1. In this example, the partial path with $V^1 = 0$ results in a better match of $\mathbf{x}'$ to $\mathbf{y}$, compared to the true input, which has $V^1 = 1$. This results in an initial error, as shown in Figure 4.5(b).

Next the decoder must hypothesize values for V^2 and V^3. However, none of the assignments result in a $\mathbf{x}'$ that matches the observations y^2, y^3, y^4 very well, and so, the path with assignment $V^1 = 1$ comes to the top of the stack. Thus, re-ordering the stack causes the algorithm to *backtrack*, reassigning a value to V^1 ($V^1 = 1$, which is correct). The decoder now continues to extend this correct path and eventually obtains the correct estimate for the true input $\mathbf{v}$.

It is clear that the choice of metric is important in guiding the search performed by the algorithm and hence plays a very important role in deciding its computational and error properties. There has been a lot of work on designing the metric for use with sequential decoding for different applications [116]. Before we study how the performance depends on the metric chosen, we explain why we are interested in the sequential decoding algorithm for problems involving sparse measurement structures.

4.2.2 Sequential decoding and the computational cut-off rate

For a tree code, the decoding complexity of the optimal ML decoder (also called the Viterbi decoder), is exponential in the memory of the code. This could become unbounded if the memory of the code is unbounded, as required for optimal tree codes. However, the sequential decoder, is a sub-optimal decoder that exhibits an interesting computational

phenomenon. The computational cut-off rate (R_{comp}) of a sequential decoder is defined as the rate below which the average complexity of decoding becomes linear in the number of measurements. Savage [62] and others [117] have established analytically the existence of the computational cut-off rate for convolutional codes. However, it is not obvious that a similar phenomenon should exist for sensor networks, especially since there is no computational cut-off for turbo codes on symmetric channels [118]. Rachlin et. al. [61] studied the stack decoder and showed that for simulated sonar range measurements, increasing the number of measurements reduces the detection complexity. In this section, we consider a slightly different version of the stack decoder for thermal measurements, motivated by our application from Section 2.3.1, to clearly demonstrate the computational cut-off phenomenon. We then compare sequential decoding with Belief Propagation from Section B, optimal algorithms from Section A and a simple bit-flipping heuristic based on Gallager's Algorithm A [58].

We now consider SD for the thermal target detection. For more details of the physics based sensor model the reader is referred to Section 2.3.1. To recall, in this application we wanted to detect the shape of hot targets on a wall using a low resolution, cheap thermal sensor. Here we empirically demonstrate the concept of a computational cut-off rate and evaluate our sequential decoding algorithm in a simulated large-scale detection application. We also compare the performance of sequential decoding to other algorithms. Experiments on real thermal sensor data will be presented in Section 5.2. The results in this section have appeared in [15].

In our simulations, we use an IR temperature sensor to sense a binary field, as shown in Figure 4.6. The field consists of grid blocks that are either at ambient temperature or at some target temperature. We set the ambient temperature at 27°C and targets at 127°C. Using our simulated IR sensor, we take measurements of the field from different locations

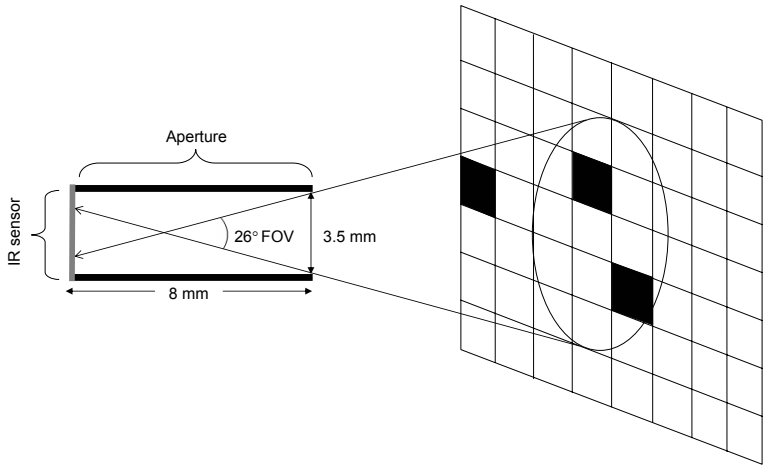


Figure 4.6: An illustration of the sensor application. An IR temperature sensor senses a discrete temperature field. The sensor field of view is limited using an aperture. White grid blocks are at ambient temperature and black grid blocks are at a higher target temperature.

and orientations. Detecting the state of the grid using these sensor measurements poses a significant computational challenge due to the large number of grid blocks that affect each sensor measurement. Since the number of grid states is exponentially large, this is a large-scale detection problem. For example, let us consider a discrete 250mm $\times$250mm field composed of 25mm$\times$25mm blocks. A sensor with field of view 26° at a distance of 250mm from the plane is affected by as many as 50 grid blocks at a time. Further, the number of possible grid states is 2^{100}. As a result, detecting the state of the binary field is a computationally difficult problem.

We ran simulations using the sensor model described earlier. For all graphs of simulation results, the points represent average values. For each point in a graph of simulation results, we generated 10 random sensor deployments and measured performance on 10 randomly generated target configurations for each deployment, resulting in 100 trials per data point. Each target configuration consisted of a 10×10 grid, and was generated by placing a target in each grid block with probability 0.3.

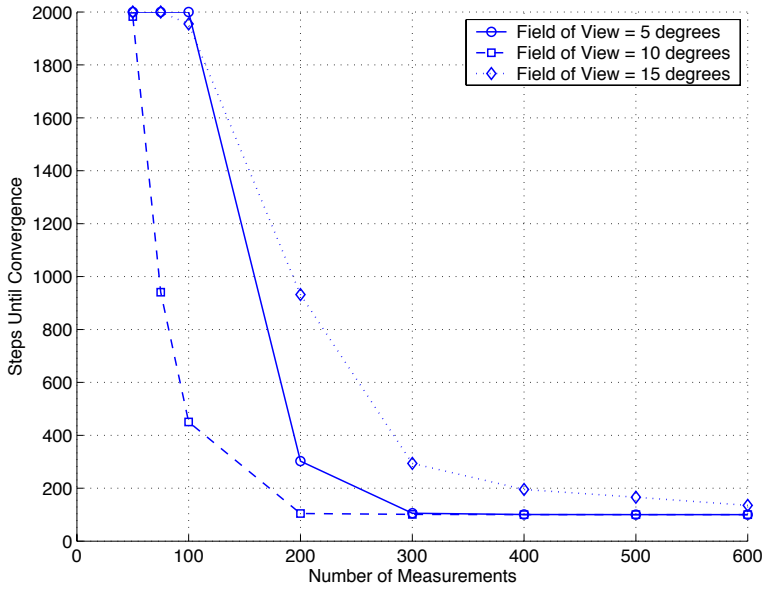


Figure 4.7: Average number of steps until convergence of sequential decoding as a function of the number of sensor measurements for sensors with different fields of view.

Figure 4.7 demonstrates the computational complexity of sequential decoding as a function of the number of sensor measurements, for sensors with different fields of view. For a small number of sensor measurements, the algorithm does not converge on average within the allotted number of maximum steps, which was set to two thousand. Above a certain number of measurements, we observe a sharp drop in the average number of steps required for the algorithm to converge. The number at which this drop occurs depends on the sensor field of view. Thus, the algorithm's complexity *decreases* with an increase in the number of sensor measurements. This behaviour is similar to the computational cutoff phenomenon observed when using sequential decoding for convolutional codes in communications. For a sufficiently large number of sensor measurements, the average number of steps required for convergence approaches 100, which is equal to the number of grid blocks in the field. This indicates that the algorithm does not backtrack often for a sufficient number of sensor measurements. Interestingly, sensors with a ten degree field of view experience this tran-

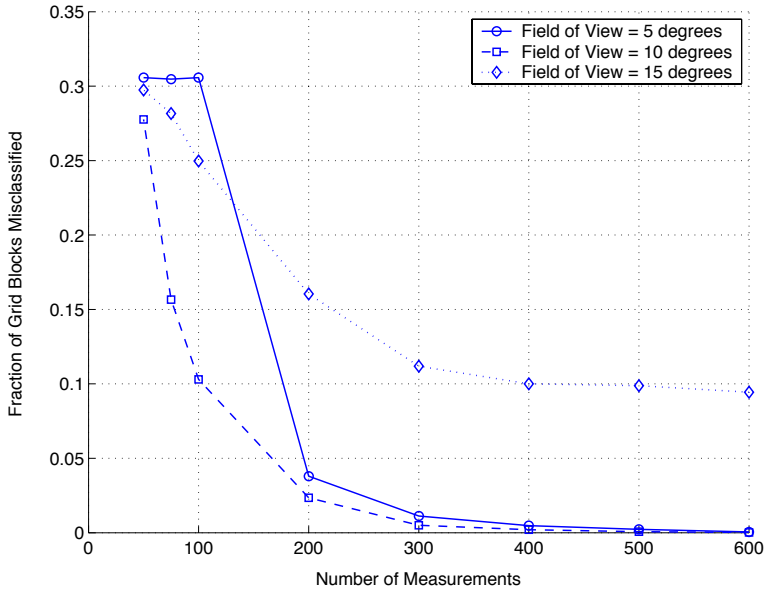


Figure 4.8: Empirical detection error of sequential decoding as a function of the number of sensor measurements for sensors with different fields of view.

sition earlier than sensors with either a five degree or a fifteen degree field of view. The detection accuracy of sequential decoding for these simulations is shown in Figure 4.8. This graph also shows an interesting relationship, with sensors with a field of view of ten degrees achieving the lowest error rate as we vary the number of sensor measurements. A priori, one might conjecture that sensors with the smallest field of view would have an advantage by enabling low estimation complexity. One might also conjecture that sensors with a large field of view would have an advantage in error rate by observing grid blocks multiple times due to overlapping sensor observations. Our simulation reveals that an intermediate field of view could have both a lower computational complexity and a higher accuracy.

To provide a context for the performance of sequential decoding, we study the effect of different parameters on the running time and accuracy of bit-flipping, loopy BP, and sequential decoding. We implemented all algorithms in Matlab.

The first algorithm, bit-flipping, is a simple and fast algorithm. This algorithm starts with a random estimate of the field and progressively improves this estimate by flipping one grid block at a time. For each field location, we compare the likelihoods of that location having a target versus the likelihood of it being unoccupied, assuming that our estimate of all other field locations is correct. We then set that field location to the value that gave us a higher likelihood. We cycle through all the locations in the field based on a random schedule. This bit-flipping can be repeated iteratively until convergence or until a fixed maximum number of iterations is reached.

The second algorithm, loopy belief propagation, is an algorithm for approximate inference in graphical models. We first convert the sensor network into a graph, with one node for each sensor and one node for each block in the field. The graph has directed edges from each field node to every sensor that observes that field location. We also specify Gaussian probability distributions for each sensor, for every configuration of sensed nodes. We use the Bayes Net Toolbox [119], which is free to download and has a fast implementation of BP in Matlab, one version of which is optimized for the conditional Gaussian nodes required for this simulation. Another standard algorithm for inference in sensor network mapping algorithms is the junction tree algorithm. However, we found that we were unable to run this algorithm using the Bayes Net Toolbox for even modest fields of view due to memory problems. Hence, we do not compare junction tree with sequential decoding. We will compare with optimal algorithms for smaller field of view in Section 4.2.3.

We study the effect of changing the field of view and the number of sensor measurements to compare the various algorithms described earlier. When the field of view is increased, the number of grid blocks sensed by the sensor with each measurement increases. Figure 4.9 shows that while the running time of belief propagation increases exponentially with the field of view of the sensor, the time taken by sequential decoding increases only

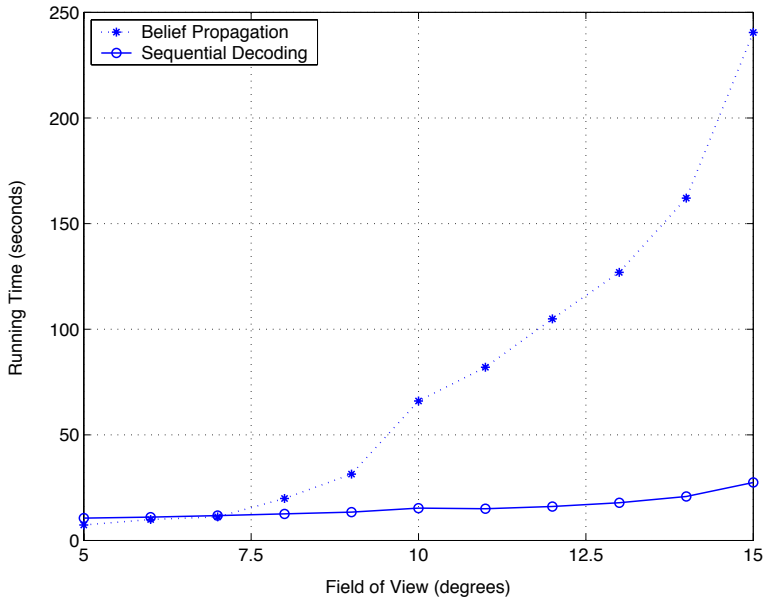


Figure 4.9: Running times of belief propagation and sequential decoding as a function of the field of view.

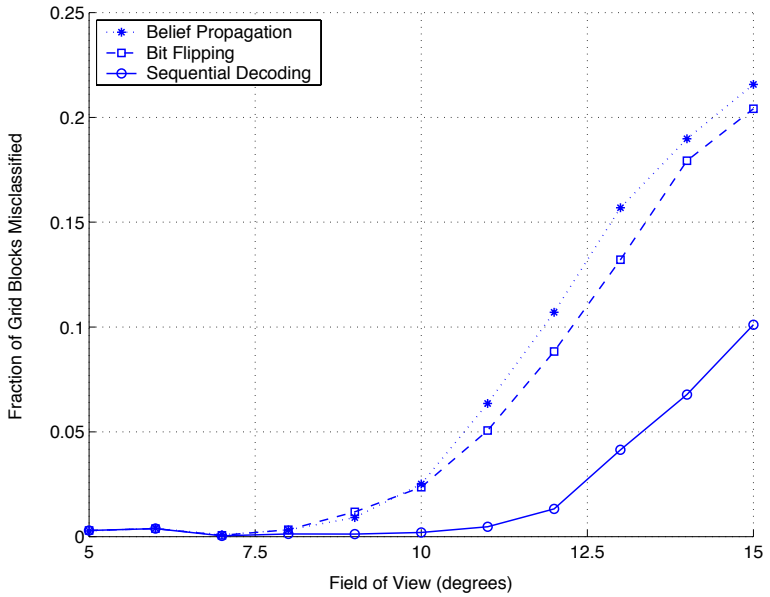


Figure 4.10: Empirical error rates of belief propagation, bit flipping, and sequential decoding as a function of field of view.

linearly. Bit-flipping was limited to ten iterations and was therefore ran in constant time (15 seconds). In these experiments the number of measurements was fixed at 400. We also analyse the error rates of the two algorithms. Significantly, despite having a much smaller computational complexity, sequential decoding achieves lower error rates than belief propagation. Similarly, sequential decoding outperforms the bit-flipping algorithm. The disparity in accuracy between sequential decoding and the other two algorithms increases with increasing field of view. As in the previous results, we observe that the relationship between field of view and error rate is not monotonic. As shown in Figure 4.10, sensors with a field of view of 9 degrees achieved the lowest error among the fields of view tested when using sequential decoding. Thus, choosing sensors with either the smallest or largest field of view may not be optimal.

Given that we now have convincing evidence that the computational cut-off phenomenon exists and can be exploited in sensor networks we will now try to develop an better understanding of the sequential decoding algorithm and its computational properties.

4.2.3 The sequential decoding metric

In this section we parallel the reasoning in [113], where a metric for sequential decoding was derived for decoding convolutional codes and extend that reasoning to the problem of detection in sensor networks. To emphasize the parallels we use codewords and messages (from the coding theory literature) interchangeably with measurements and inputs, respectively. We use an argument similar to the random coding argument [29], but which deals with the complications of inter-codeword dependence using the method of types [98]. We first describe the contiguous sensor network model [32] that we will study in this section.

4.2.3.1 Sensor Network Model

We consider the problem of detection in one-dimensional sensor networks. While a heuristic sequential decoding procedure has been applied to complex 2-D problems [15, 61], we present the 1-D binary case for ease of understanding and analysis. Motivated by parallels to communication theory and prior work, we model a contiguous sensor network as shown in Figure 3.2. In this model, the environment is modeled as a k-dimensional discrete vector $\mathbf{v}$. Each position in the vector can represent any binary phenomenon such as presence or absence of a target. Possible target vectors are denoted by $\mathbf{v_i}$ $i \in 1, \ldots, 2^k$. There are n sensors and each sensor is connected to (senses) c contiguous locations $t, \ldots, t+c-1$. The noiseless output of the sensor is a value $x \in \mathcal{X}$ that is an arbitrary function of target bits to which it is connected, $X = \Psi(v^t, \ldots, v^{t+c-1})$. For example, this function could be a weighted sum, as in the case of thermal sensors, or location of the nearest target, as in the case of range sensors. The sensor output is then corrupted by noise according to an arbitrary p.m.f. $P_{Y|X}$ to obtain a vector of noisy observations $\mathbf{y} \in \mathcal{Y}$. We assume that the noise in each sensor is identical and independent so that observed sensor output vector is related to the noiseless sensor outputs as $P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) = \prod_{l=1}^{n} P_{Y|X}(y^l|x^l)$.

Using this output $\mathbf{Y}$, the decoder must detect which of the 2^k target vectors actually occurred. Many applications also have a distortion constraint that the detected vector must be less than a distortion $D \in [0, 1]$ from the true vector, i.e., if $d_H(\mathbf{v_i}, \mathbf{v_j})$ is the Hamming distance between two target vectors, the tolerable distortion region of $\mathbf{v_i}$ is $\mathcal{D}_{\mathbf{v_i}} = \{j : \frac{1}{k} d_H(\mathbf{v_i}, \mathbf{v_j}) < D\}$. Detection using this contiguous sensor model is hard because adjacent locations are often sensed together and have a combined effect on the sensors sensing them, which is hard to separate.

Before we go into the details of our analysis of sequential decoding for such problems, we introduce the notation related to the circular c-order types that we use related to the

contiguous measurement ensemble we analyze. Some of these terms were introduced in the analysis of Chapter 3.

4.2.3.2 Definitions

As in Chapter 3, the rate R of a sensor network is defined as the ratio of number of target positions being sensed, k, to number of sensor measurements, n, $R = \frac{k}{n}$. $\mathcal{D}(p||q)$ represents the Kullback-Leibler distance between two distributions p and q. A sensor network is created as follows. Each sensor independently connects to c target locations according to the contiguous sensor network model as described in Section 4.2.3.1. When we consider the ensemble of such sensor networks the noiseless sensor outputs $\mathbf{X_i}$ associated with a particular target configuration $\mathbf{v_i}$ is random. This allows to use a random coding argument to analyze the performance of the decoder. We can write this probability as $P_{\mathbf{X_i}}(\mathbf{x_i}) = \prod_{l=1}^{n} P_{X_i}(x_i^l)$. A crucial fact is that the sensor outputs are not independent of each other, and are only independent given the true target configuration. The random vectors $\mathbf{X_i}$ and $\mathbf{X_j}$ corresponding to different target vectors $\mathbf{v_i}$ and $\mathbf{v_j}$ are not independent. Because of the sensor network connections to the target vector, two target vectors, $\mathbf{v}_i$ and$\mathbf{v}_j$, that are similar to each other, are expected to result in similar sensor outputs ($\mathbf{X}_i$ and $\mathbf{X}_j$).

Unlike the arbitrary and regular ensembles defined in Sections 3.1.3.1 and 3.1.3.2, this contiguous ensemble is not input permutation invariant. So, the types, γ and λ defined in Chapter 3 cannot be used to analyze this situation. Instead, we define the concepts of *circular c-order types* and *circular c-order joint types* [98] to apply to our problem. A circular sequence is one where the last element of the sequence precedes the first element of the sequence. The circular c-order type γ of a binary target vector sequence $\mathbf{v_i}$ is defined as 2^c dimensional vector where each entry corresponds to the frequency of occurrence of one possible subsequences of length c. For example for a binary target vector and $c = 2$, $\gamma = (\gamma^{00}, \gamma^{01}, \gamma^{10}, \gamma^{11})$, where γ^{01} is the fraction of two-bit subsequences in $\mathbf{v_i}$ of value

01. Since all c-order types are assumed to be circular, we omit the word circular in the remainder of this section for brevity.

Since each sensor selects the target positions it senses independently and uniformly across the target positions, $P_{\mathbf{X_i}}(\mathbf{x_i})$ depends only on the c-order type γ of $\mathbf{v_i}$ and can be written as $P_{\mathbf{X_i}}(\mathbf{x_i}) = P^{\gamma,n}(\mathbf{x_i}) = \prod_{l=1}^{n} P^{\gamma}(x_i^l)$ and is the same for each $\mathbf{v}$ of the same c-order type γ. We define λ as the vector of $\lambda^{(\mathbf{a})(\mathbf{b})}$, the fraction of positions $\mathbf{v_i}$ has subsequence $\mathbf{a}$ and $\mathbf{v_j}$ has subsequence $\mathbf{b}$. For example when $c = 2$, $\lambda = (\lambda^{(00)(00)}, \ldots, \lambda^{(11)(11)})$, where $\lambda^{(01)(11)}$ is the fraction of to-bit subsequences of value 01 in $\mathbf{v_i}$ and 11 in $\mathbf{v_j}$, at the same positions. The joint probability $P_{\mathbf{X_i X_j}}(\mathbf{x_i}, \mathbf{x_j})$ depends only on the joint type of target vectors $\mathbf{v_i}$ and $\mathbf{v_j}$ and we can write $P_{\mathbf{X_i X_j}}(\mathbf{x_i}, \mathbf{x_j}) = P^{\lambda,n}(\mathbf{x_i}, \mathbf{x_j}) = \prod_{l=1}^{n} P^{\lambda}(x_i^l, x_j^l)$.

There are two other important probability distributions that arise in the discussions to follow. The first is the joint distribution between the ideal output $\mathbf{x_i}$ when $\mathbf{v_i}$ occurs, and its corresponding noisy output $\mathbf{y}$. Assuming an i.i.d. noise distribution $P_{Y|X}$, this is written as

$$\mathbf{P}_{\mathbf{X_i Y}}(\mathbf{x_i}, \mathbf{y}) = \prod_{l=1}^{n} \mathcal{P}_{X_i Y}(x_{il}, y_{il}) = \prod_{l=1}^{n} P_{X_i}(x_i^l) P_{Y|X}(y_l | x_i^l) \tag{4.8}$$

The second distribution is the joint distribution between the ideal output $\mathbf{x_j}$ corresponding to $\mathbf{v_j}$ and the noisy output $\mathbf{y}$ generated by the occurrence of $\mathbf{x_i}$ corresponding to a different target vector $\mathbf{v_i}$. This is denoted as

$$\mathbf{Q}^{(i)}_{\mathbf{X_j Y}}(\mathbf{x_j}, \mathbf{y}) = \prod_{l=1}^{n} \mathcal{Q}^{i}_{X_j Y}(x_j^l, y^l) = \prod_{l=1}^{n} \sum_{a \in \mathcal{X}} P_{X_j X_i}(x_j^l, x_i^l = a) P_{Y|X}(y^l | x_i^l = a) \tag{4.9}$$

Again the important fact should be noted that even though $\mathbf{Y}$ was produced by $\mathbf{X_i}$, $\mathbf{Y}$ and $\mathbf{X_j}$ are dependent because of the dependence between $\mathbf{X_i}$ and $\mathbf{X_j}$.

We can reduce c-order type over a binary alphabet to a 2-order type over a sequence with symbols in an alphabet $\mathcal{V}^{(c-1)}$ of cardinality $2^{2(c-1)}$ by mapping each shifted binary subsequence of length $c-1$ to a single symbol in this new alphabet. We define

$\lambda^{'} = \{\lambda^{(\mathbf{a})(\mathbf{b})}, \forall \mathbf{a}, \mathbf{b} \in \mathcal{V}^{(c-1)}\}$ as a marginal type (marginalized over the last co-ordinate), $\lambda'^{(\mathbf{a})(\mathbf{b})} = \sum_{a,b \in \mathcal{V}} \lambda^{(\mathbf{a}a),(\mathbf{b}b)}$ and $\tilde{\lambda} = \{\lambda^{(\mathbf{a})(a)(\mathbf{b})(b)}, \forall \mathbf{a}, \mathbf{b} \in \mathcal{V}^{(c-1)}, \forall a, b \in \mathcal{V}\}$ as a conditional type $\tilde{\lambda}^{(\mathbf{a}a)(\mathbf{b}b)} = \frac{\lambda^{(\mathbf{a}a)(\mathbf{b}b)}}{\lambda^{(\mathbf{a})(\mathbf{b})}}$. Correspondingly, we define $\gamma^{'} = \{\gamma'^{\mathbf{a}}, \forall \mathbf{a} \in \mathcal{V}^{(c-1)}\}$ with $\gamma'^{\mathbf{a}} = \sum_{a \in \mathcal{V}} \gamma^{\mathbf{a}a}$ and a conditional type $\tilde{\gamma} = \{\gamma^{\tilde{\mathbf{a}}a}, \forall \mathbf{a} \in \mathcal{V}^{(c-1)}, \forall a \in \mathcal{V}\}$ with $\gamma^{\tilde{\mathbf{a}}a} = \frac{\gamma^{(\mathbf{a})(a)}}{\gamma^{(\mathbf{a})}}$.

4.2.3.3 Metric derivation

While deriving the metric to be used for detection in sensor networks, significant changes need to be made from [113] because the codewords ($\mathbf{X}_i$) are not independent or identically distributed. The optimum decoding procedure would be the MAP (Maximum A Posteriori) decoder, which seeks to find the codeword $\mathbf{X}$ that maximizes the joint probability $\mathbf{P_{X_i Y}}$. The MAP decoder for the 1-D contiguous sensor network case reduces to the Viterbi algorithm. The Viterbi algorithm is feasible only for sensors with smaller field of view c, since it requires the evaluation of each element in a state space of size 2^{c-1} at each step with number of steps linear in the length of the target vector. The computation grows exponentially with c, and so we must use a sub-optimal decoder for large c. The SD algorithm relies on the fact that if probability of error is sufficiently small, then the *a posteriori* most probable $\mathbf{X}$ is expected to be much more probable than all the other codewords. We try to select a condition under which we may expect the performance of an algorithm to be good. Given the received sequence $\mathbf{y}$, suppose that there exists a $\mathbf{x_i}$ such that

$$p(\mathbf{x_i}, \mathbf{y}) \geq \sum_{j \neq i} p(\mathbf{x_j}, \mathbf{y}) \tag{4.10}$$

If the sensor networks were generated according to the random scheme described in Section 4.2.3.1, then we can approximate the left and right hand sides of (4.10) with their expected value over the random ensemble of sensor network configurations, along the lines of the

random coding argument [29] as applied to sensor networks [32].

$$\mathbf{P}_{\mathbf{X_i Y}}(\mathbf{x_i}, \mathbf{y}) \geq \sum_{j \neq i} \mathbf{Q}^{(i)}_{\mathbf{X_j}, \mathbf{Y}}(\mathbf{x}_j, \mathbf{y}) \tag{4.11}$$

The right hand side of (4.11) has an exponential number of terms because of the exponential number of incorrect codewords indexed by j. However, as described in Section 4.2.3 these terms can be represented in terms of the type γ of $\mathbf{v_i}$ and joint type λ of $\mathbf{v_i}$ and $\mathbf{v_j}$. If we can tolerate errors up to a distortion D we should require the posterior probability of the correct codeword to be much greater than that of all other codewords that have a distortion greater than D. Let $\mathcal{S}_\gamma(\mathcal{D})$ be the set of all joint types corresponding to codewords of target vectors $\mathbf{v}$ at a distortion greater than D.

$$\prod_{l=1}^{n} P^{\gamma}_{X_i}(x_i^l) P_{Y|X}(y^l|x_i^l) \geq \sum_{\lambda \in \mathcal{S}_\gamma(\mathcal{D})} \beta(\lambda, k) \prod_{l=1}^{n} \sum_{a \in \mathcal{X}} P^{\lambda}_{X_j X_i}(x_j^l, X_i^l = a) P_{Y|X}(y^l | X_i^l = a) \tag{4.12}$$

where $\beta(\lambda, k)$ is the number of vectors $\mathbf{v_j}$ having a given joint type λ with $\mathbf{v_i}$, bounded by $\beta(\lambda, k) \leq 2^{k[H(\tilde{\lambda}|\lambda') - H(\tilde{\gamma_j}|\gamma_j')]}$ [32]. Since $\mathbf{v_i}$ and $\mathbf{v_j}$ have the joint type λ their types are the corresponding marginals of λ (because of our definition of types being circular) i.e $\gamma_{j\mathbf{b}} = \sum_{\mathbf{a} \in 0,1^c} \lambda_{(\mathbf{a})(\mathbf{b})}$ and $\gamma_{i\mathbf{a}} = \sum_{\mathbf{b} \in 0,1^c} \lambda_{(\mathbf{a})(\mathbf{b})}$.There are only a polynomial number of types $C(k) = 2^{2(c-1)} k^{2^{c-1}} (k+1)^{2^{2(c-1)}}$ [32, 98], and so, (4.12) reduces to,

$$\begin{aligned} &\prod_{l=1}^{n} P^{\gamma}_{X_i}(x_i^l) P_{Y|X}(y^l|x_i^l) - C(k) 2^{k[H(\tilde{\lambda^*}|\lambda^{*'}) - H(\tilde{\gamma^*}|\gamma^{*'})]} \\ &\prod_{l=1}^{n} \sum_{a \in \mathcal{X}} P^{\lambda^*}_{X_j X_i}(x_j^l, X_i^l = a) P_{Y|X}(y_l | X_i^l = a) \geq 0 \end{aligned} \tag{4.13}$$

for an appropriate choose of λ^* from the set of all λs and where γ^* is the marginal type corresponding to the joint type λ^*. Taking $\log_2()$ and applying a limiting argument as

$n \to \infty$, we obtain the general form of the metric,

$$\sum_{l=1}^{n} \log_2[P^{\gamma}(x_i^l)P_{Y|X}(y^l|x_i^l)] - \sum_{l=1}^{n} \log_2[P^{\gamma^*}(x_i^l)P_{Y|X}^{\lambda^*}(y_l|x_i^l)] - k[H(\tilde{\lambda^*}||\lambda^{*'}) - H(\tilde{\gamma^*}|\gamma^{*'})] \tag{4.14}$$

The key property of this metric is that it is expected to be positive when the probability of the correct codeword dominates erroneous codewords.

The choice of λ^* is based on the properties desired of the metric and is discussed in Section 4.2.3.5. This leads us to the choice of appropriate metric as $S_n = \sum_{i=1}^{n} \Delta_i$, where Δ_i is the increment in the metric as sensor i (along the 1-D space) is decoded.

$$\Delta_i = \log_2[P^{\gamma}(x_i^l)P_{Y|X}(y^l|x_i^l)] - \sum_{l=1}^{n} \log_2[P^{\gamma^*}(x_i^l)P_{Y|X}^{\lambda^*}(y^l|x_i^l)] - R[H(\tilde{\lambda^*}||\lambda^{*'}) - H(\tilde{\gamma^*}|\gamma^{*'})] \tag{4.15}$$

There are two requirements of a metric to be appropriate for sequential decoding. The first is that the expected value of the metric increase as the decoding of the target vector proceeds down the correct branches of the binary hypothesis tree. The second is that the metric should decrease along any incorrect path in the tree that is more than a distortion D from the true target vector. We analyse the metric derived and prove that it satisfies these requirements for appropriate choices of rate R and representative joint type λ^*.

4.2.3.4 Analysis of the metric along the correct path

We calculate the expected value of the increment in the metric Δ_i along the correct decoding path and derive the condition on the sensor network for the expected value of these increments to be positive along this path.

$$E_{X,Y}\Delta_i = E_{X,Y} \log_2[P^{\gamma}(x)P(y|x)] - E_{X,Y} \log_2[P^{\gamma^*}(x)P^{\lambda^*}(y|x)] - R[H(\tilde{\lambda^*}||\lambda^{*'}) - H(\tilde{\gamma^*}|\gamma^{*'})] \tag{4.16}$$

Here, the expectation is over X and Y, the noiseless outputs and the noisy measurements, respectively, along the correct path. This reduces to

$$E_{X,Y}\Delta_i = \mathcal{D}(\mathbf{P_{XY}}||\mathbf{Q}^{(\lambda^*)}_{\mathbf{X_iY}}) - R[H(\tilde{\lambda^*}||\lambda^{*'}) - H(\tilde{\gamma^*}|\gamma^{*'})] \tag{4.17}$$

which will be greater than or equal to 0 if we chose rate R such that

$$R \leq \min_{\substack{\lambda \\ \sum_{a\neq b}\lambda^{ab}\geq D \\ \sum_b \lambda_{ab}=\gamma^{ia}}} \frac{\mathcal{D}(\mathbf{P_{XY}}||\mathbf{Q}^{(\lambda)}_{\mathbf{X_iY}})}{[H(\tilde{\lambda}||\lambda') - H(\tilde{\gamma}|\gamma')]} \tag{4.18}$$

We note that the right hand side of (4.18) is the expression for a lower bound on sensing capacity $C^{ML}_{LB}(D)$ for a contiguous sensor network derived in [32]. Thus as long as the number of sensors used is sufficiently high so that the rate R is below $C^{ML}_{LB}(D)$ the metric will increase along the correct path.

4.2.3.5 Analysis of the metric along an incorrect path

We now analyse the change in metric over an incorrect path corresponding to a vector $\mathbf{v_j}$ having a joint type λ with the true target vector. We calculate the expected value of the increment in the metric Δ_i along this path.

$$\begin{aligned} &E_{X',Y}\Delta_i = E_{X',Y}\log_2 P^{\gamma}(x)P(y|x) - \\ &E_{X',Y}\log_2 P^{\gamma^*}(x)P^{\lambda^*}(y|x) - R[H(\tilde{\lambda^*}||\lambda^{*'}) - H(\tilde{\gamma^*}|\gamma^{*'})] \end{aligned} \tag{4.19}$$

where X' is the noiseless sensor outputs along the wrong path and Y is the noise corrupted true sensor outputs.

$$E_{X',Y}\Delta_i = \sum_{X'Y} Q^{\lambda}(x,y)\log_2(\frac{P^{\gamma^*}(x,y)}{Q^{\lambda^*}(x,y)}) - R[H(\tilde{\lambda^*}||\lambda^{*'}) - H(\tilde{\gamma^*}|\gamma^{*'})] \tag{4.20}$$

$$= -\mathcal{D}(Q^{\lambda}||P^{\gamma^*}) + \mathcal{D}(Q^{\lambda}||Q^{\lambda^*}) - R[H(\tilde{\lambda^*}||\lambda^{*'}) - H(\tilde{\gamma^*}|\gamma^{*'})] \tag{4.21}$$

$$\leq -\mathcal{D}(Q^{\lambda^*}||P^{\gamma^*}) - R[H(\tilde{\lambda^*}||\lambda^{*'}) - H(\tilde{\gamma^*}|\gamma^{*'})] \tag{4.22}$$

$$\leq 0 \tag{4.23}$$

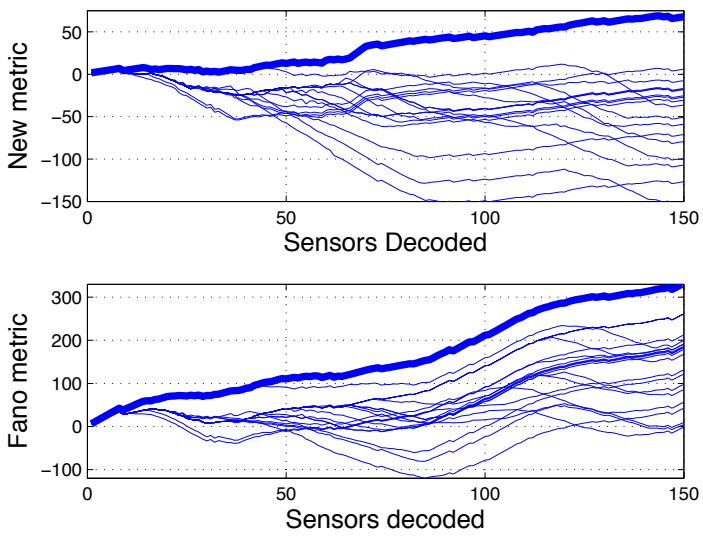


Figure 4.11: Behavior of new metric and Fano metric along the correct path(in bold) and incorrect paths at a distortion D=0.02 for $c = 15$

Inequality (4.22) is true if we choose,

$$\lambda^* = \arg \min_{\lambda \in \mathcal{S}_\gamma(\mathcal{D})} \mathcal{D}(Q^\lambda || P^{\gamma^*}) \tag{4.24}$$

Inequality (4.22) arises from (4.21) because the set $\mathcal{S}_\gamma(\mathcal{D})$ is closed and convex, and using Theorem 12.6.1 in [101]. Equation (4.23) is from the positivity of KL divergence [101] and since $[H(\tilde{\lambda^*}||\lambda^{*'}) - H(\tilde{\gamma^*}|\gamma^{*'})] > 0$ since each γ is the marginal of the corresponding λ.

The convexity of $\mathcal{S}_\gamma(\mathcal{D})$ can be reasoned as follows. The set of $\lambda \in \mathcal{S}_\gamma(\mathcal{D})$ are defined by 1) a normalization constraint to ensure that they are true probability distributions 2) a set of Markov constraints so that they are associated with possible 1-D sequences and 3) a distortion constraint so that they correspond to sequences more than a distortion D from the true target vector. Since all these constraints are linear equalities and finite-dimensional, the set $\mathcal{S}_\gamma(\mathcal{D})$ is convex and closed. Each probability distribution Q^λ is linear in the elements of λ and hence the set of Q^λ is convex, and so, Theorem 12.6.1 in [101] can be applied.

4.2.3.6 *Understanding the behavior of the new metric*

We simulate a random sensor network at a specified rate R with sensors drawn randomly according to the contiguous sensor model. Error rates and runtimes for sequential decoding are averaged over 5000 runs. The output of each sensor is the weighted sum of inputs in its field of view $x = \sum_{j=1}^{c} w_j v_{t+j}$. This output is discretized into one among 50 levels, uniformly spaced between 0 and the maximum possible value, and corrupted with exponentially distributed noise to obtain the sensor outputs $\mathbf{y}$. A sequential decoder (stack decoder) decodes $\mathbf{y}$ to the estimate $\hat{\mathbf{v}}$. If the detected vector $\hat{\mathbf{v}}$ is more than a distortion D from the true target vector an error occurs. We initially simulate the growth of the metric along correct and wrong paths. This is to verify that the properties derived in Sections 4.2.3.4 and 4.2.3.5, which were for the expected value of the metric, do in fact hold, for a specific sensor network configuration. The wrong paths are chosen to be at a distortion D from the true path. The behavior of the new metric is compared to that of the Fano metric, which is used to decode convolutional codes, where the increment is defined to be [113]:

$$\Delta_{F_i} = \log_2[P_{Y|X}(y^l|x_i^l)] - \log_2 P_{Y^l}(y^l) \tag{4.25}$$

Figure 4.11 shows the important difference between the new metric and the Fano metric. Even when we proceed along an erroneous path that is at a distortion D from the true path, the Fano metric continues to increase. This is because the Fano metric was derived assuming that if a path diverges from the true path at any point, the bits of the two codewords will be independent from that point onwards. While this is approximately true for strong convolutional codes with large memory, it is not a good assumption in sensor networks, due to the strong dependence between the codewords, arising from choice of sensors. Thus, if noise causes the Fano metric based decoder to take one of these erroneous paths, it may continue to explore it, increasing the path length, since the metric continues to increase

even though the path is now incorrect. In contrast, when our new metric is used, the metric for the wrong path at distortion D decreases after the error, while the metric of the correct path is still increasing. Thus, we expect that the new metric would perform better than the Fano metric when used in a sequential decoder.

We will see in the next chapter that these results are in fact useful, when we return to applications of this result to algorithm design. Now we move on to a different method of analyzing sequential decoding that is more closely tied to its origins in the coding theory literature and that rigorously derives expressions related to the computational complexity and accuracy of sequential decoding for specific, fixed measurement functions. These results will be used in Chapter 5 to design sensors and decoders that are computationally efficient and accurate.

4.2.4 Sequential decoding for specific measurement functions

Sequential decoding is a sub-optimal method for decoding tree codes. We now consider stack decoding of tree codes with real symbols transmitted over a Gaussian channel. The tree corresponding to a code is show in Figure 4.12. We described a similar code tree when we discussed the SD algorithm at the beginning of this section. Each input vector corresponds to a path through the tree and the output symbols on the corresponding branches form the codeword $\mathbf{X}$ that is transmitted across an AWGN channel. This forms the received symbols $\mathbf{Y} = \mathbf{X} + \mathbf{N}$ where $\mathbf{N}$ is the additive noise in the channel, whose elements n_k are i.i.d., distributed as $\mathcal{N}(0, \sigma^2)$.

Tree codes, such as convolutional codes, that are generated by encoders that have finite memory can be decoded optimally by the Viterbi algorithm. The computation required by the Viterbi algorithm to decode a codeword is exponential in the memory of the code. Sequential decoders such as the stack decoder [60], assign a value (or "metric") to each

partial path through the tree. They then perform a metric first search of the tree.

Research in areas such as sensor networks [15, 103] and decoding over ISI channels [114] have shown that many other real world search problems can also be expressed as tree codes. These 'encoders' may have very large memory, making sequential decoding a practical alternative to the Viterbi algorithm. For example [120] and more recently [121] suggest the use of sequential decoding for decoding lattice codes. These applications have resulted in renewed interest in sequential decoding algorithms, especially in the case of real-valued (as opposed to binary) symbols transmitted over an additive white Gaussian noise (AWGN) channel. While there has been work on construction of good codes for sequential decoding over AWGN channels, such as [122] for convolutional codes, [123] for trellis codes and [116] for linear block codes, that used the results in [124] - which was for discrete alphabets over DMCs - for motivation, there has been no analysis of sequential decoding over AWGN channels. We analyse a particular form of the metric (which has a bias parameter) and show that there exists a bias for which the computational effort decreases exponentially with the column distance function (calculated as a squared Euclidean distance) of the tree code. Since the proof does not require independence of the transmitted symbols the proof holds for (i) convolutional codes with anti-podal signaling, (ii) sequence detection over Gaussian ISI channels and (iii) some sensor networks.

The number of paths explored by these SD algorithms depends on the growth of the metric along the correct and incorrect paths, which in turn depends on the particular noise instantiation for that channel use. Thus, the computational effort of sequential decoding is a random variable. Random coding arguments have been used [125] to show that for rates R less than some value R_{comp}, the average computational effort of sequential decoders is small and essentially independent of the size of the memory of the code. This makes sequential decoding an attractive alternative to Viterbi decoding for tree codes with large

memory. Seminal work by Chevillat and Costello [124] developed an upper bound on the computational effort and error probability for a specific convolutional code (as opposed to the existence proofs using random coding arguments) over the binary symmetric channel (BSC). These bounds decrease exponentially with the so called "column distance function" of the convolutional code.

A commonly used metric for sequential decoding of convolutional codes is the Fano metric [113], which is optimal for minimum error probability decoding of variable length codes [126]. The Fano metric for a path of length rW symbols is,

$$M_f(r) = \log_2 \frac{P(\mathbf{Y}_{(1,rW)}|\mathbf{X}_{(1,rW)})}{P(\mathbf{Y}_{(1,rW)})} - rWR \tag{4.26}$$

where $\mathbf{Y}_{(1,rW)}$ is the first rW received symbols and $\mathbf{X}_{(1,rW)}$ is the first rW transmitted symbols. For a specific tree code (decoded over a Gaussian channel), without other constraints, computing $P(\mathbf{Y}_{(1,rW)})$ is not feasible and it needs to be approximated at the decoder. One approximation is to assume that all the received symbols are equi-probable, in which case, after appropriate scaling, the metric can be written as

$$M(r) = -d(r) + \gamma r \tag{4.27}$$

where $d(r)$ is the squared Euclidean distance between the first rW symbols of $\mathbf{Y}$ and $\mathbf{X}$, and γ is the bias term in the metric. Note that this is not the Fano metric for the Gaussian channel since we have approximated $P(\mathbf{Y}_{(1,rW)})$ by a constant for all $\mathbf{Y}$. The rest of this sub-section sets up the quantity to be bounded based on arguments made in [125] and uses notation from [124].

4.2.4.1 Computational effort in stack decoding

Let $\mathbf{X}^i$ be an infinitely long codeword transmitted over a Gaussian memoryless channel. The incorrect subset S_u is defined as the set of codewords $\mathbf{X}^j$ that coincide with the correct

codeword in the first u branches but diverge in branch $u+1$. The number of node extensions C_u performed by the decoder in each incorrect subset S_u can be used as a measure of decoding effort [124, 125]. Thus, in this sub-section we are interested in bounding $P(C_u > N_u)$, i.e., the probability that the stack decoder will perform more than N_u node extensions in S_u.

In analysing each S_u, path length is measured from the origin of S_u. Using the pigeon hole principle, we see that before extending for the first time an incorrect path j in S_u beyond L the decoder has made at most $N_u = 2^{VL}$ computations in S_u. Let $[L^j > L]$ be the event that path j is extended beyond length L.

$$P(C_u > N_u) < P(\cup_j [L^j > L]) \tag{4.28}$$

where the union is over all paths j in S_u. In the stack sequential decoder, a path can be extended only if reaches the top of the stack. If M^i_{min} be the minimum metric along the correct path, an incorrect path j can be extended beyond length L only if its metric at length L, $M^j(L)$, is above M^i_{min}. Another way of stating this is that the metric $M^j(L)$ should be greater than any metric along path i.

$$P([L^j > L]) \leq P(\cup_{L' \geq 0} \left(M^j(L) \geq M^i(L')\right) \tag{4.29}$$

Based on (4.27) we see that the metric value difference is related to the difference in squared Euclidean distance between the two paths (i up to length L' and j upto length L), and the received symbol sequence. We seek to bound (4.29) by a value that decreases exponentially with the "column distance function" of the code, which we now proceed to define. $d^{ij}(L)$ is the squared Euclidean distance between $\mathbf{X}^i_{(1,WL)}$ and $\mathbf{X}^j_{(1,WL)} \in S_u$, that is $d^{ij}(L) = \sum_{k=1}^{WL}(x^i_k - x^j_k)^2$ where x^i_k is the k^{th} symbol in sequence $\mathbf{X}^i$. We define the column distance

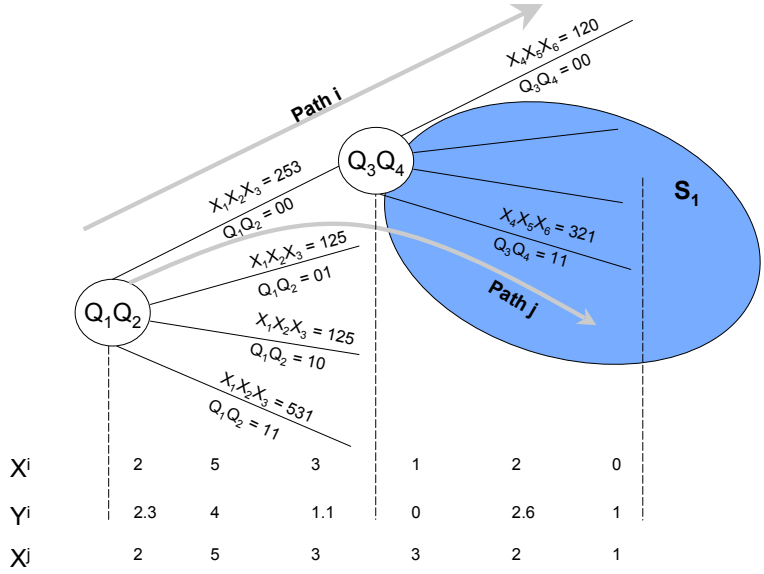


Figure 4.12: First two levels of a code tree for a measurement network. Path i corresponds to the transmitted symbol sequence $\mathbf{X}_i$ and j corresponds to $\mathbf{X}_j$ in the incorrect subset S_1 for $u = 1$. All distances are measured from the root of S_u, $d^i(L = 1) = 2.36$, $d^j(L = 1) = 9.36$, $d^{ij}(L = 1) = 5$.

function (CDF) of the code to be

$$d_c(r) = \min_{i,\mathbf{X}_j \in S_u} d^{ij}(r) = \min_{i,\mathbf{X}_j \in S_u} \sum_{k=1}^{WL} (x_k^i - x_k^j)^2 \tag{4.30}$$

This differs from the Hamming distance based CDF defined in [124]. $d_c(r)$ is a monotonically increasing function of r. We define $d^i(L')$ to be the squared Euclidean distance between the first L' branches of path i and the received sequence $\mathbf{Y}$ beyond u. $d^j(L)$ is the squared Euclidean distance between the first L branches of path j and the received

sequence $\mathbf{Y}$ beyond u. Examples are in Figure 4.12. Using (4.27) in (4.29),

$$\begin{aligned} P([L^j > L]) \leq & \sum_{L'=0}^{\infty} P(d^i(L') - d^j(L) > \gamma(L' - L)) \\ = & \sum_{L'=0}^{L} P(d^i(L') - d^j(L) > \gamma(L' - L)) \\ & + \sum_{L'=L+1}^{\infty} P(d^i(L') - d^j(L) > \gamma(L' - L)) \end{aligned} \tag{4.31}$$

4.2.4.2 *Upper bound on number of computations for the AWGN channel*

We now proceed to analyse (4.31). The differences in the rest of the proof, as compared to [124], arise due to the real valued transmitted symbols and the Gaussian channel as opposed to binary symbols transmitted over the BSC. We analyse the two sums separately, and show that each one decreases exponentially with the column distance function $d^{ij}(L)$.

4.2.4.3 *Case A:* $L + 1 \leq L' < \infty$

y_k is the k^{th} received symbol $y_k = x_k^i + n_k$.

$$d^i(L') - d^j(L) = \sum_{k=1}^{WL'} (x_k^i - y_k)^2 - \sum_{k=1}^{WL} (x_k^j - y_k)^2 \tag{4.32}$$

$$= -\sum_{k=1}^{WL} (x_k^i - x_k^j)^2 - 2\sum_{k=1}^{WL} n_k (x_k^i - x_k^j) + \sum_{k=WL+1}^{WL'} n_k^2 \tag{4.33}$$

Define $s = d^i + d_1(L)$, $d^i = \sum_{k=WL+1}^{WL'} n_k^2$ and $d_1(L) = -2\sum_{k=1}^{WL} n_k (x_k^i - x_k^j)$. d^i is the sum of squares of $W(L'-L)$ Gaussian i.i.d zero mean random variables, and hence it has a χ^2 distribution with $N = W(L' - L)$ degrees of freedom. $d_1(L)$ is the sum of independent Gaussian random variables and is distributed as $\mathcal{N}(0, 4\sigma^2 d^{ij}(L))$. d^i is random because of the dependence on noise samples $n_k,\ k = WL + 1, \ldots, WL'$ and $d_1(L)$ depends on the noise samples $n_k,\ k = 1, \ldots, WL$. These are non-overlapping intervals and the noise is i.i.d. Thus, d^i and $d_1(L)$ are independent random variables. Hence the sum $s = d^i + d_1(L)$

has a moment generating function (mgf) that is the product of the mgfs of the two parts.

$$\phi_s(t) = \frac{1}{(1-2\sigma^2 t)^{\frac{W(L'-L)}{2}}} e^{2\sigma^2 d^{ij}(L)t^2}, \quad Re\{t\} < \frac{1}{2\sigma^2} \tag{4.34}$$

We define $\epsilon = d^{ij}(L) + \gamma(L' - L)$. Using the Chernoff bound for $t \in \mathbb{R}, t > 0$,

$$\sum_{L'=L+1}^{\infty} P(d^i(L') - d^j(L) > \gamma(L'-L)) = \sum_{L'=L+1}^{\infty} P(s > \epsilon) \tag{4.35}$$

$$\leq e^{-d^{ij}(L)(t-2\sigma^2 t^2)} \sum_{L'=L+1}^{\infty} \left(e^{t\gamma}(1-2\sigma^2 t)^{\frac{W}{2}}\right)^{-(L'-L)} \tag{4.36}$$

which is finite if $\frac{W}{2}\log_e(1-2\sigma^2 t) + t\gamma > 0$. The slope of $\frac{W}{2}\log_e(1-2\sigma^2 t) + t\gamma$ at $t = 0$ is $-W\sigma^2 + \gamma$. By choosing $\gamma > W\sigma^2$, the conditions in (4.34) and 4.36 can be satisfied.

$$\sum_{L'=L+1}^{\infty} P(s \geq \epsilon) \leq e^{-d^{ij}(L)(t-2\sigma^2 t^2)} \frac{1}{e^{t\gamma}(1-2\sigma^2 t)^{\frac{W}{2}} - 1} \tag{4.37}$$

$$= G(t) e^{-d^{ij}(L)(t-2\sigma^2 t^2)} \tag{4.38}$$

where $G(t)$ is independent of L. From (4.34), $t < \frac{1}{2\sigma^2} \implies t - 2\sigma^2 t^2 > 0$. We can minimize this bound over $0 < t < \frac{1}{2\sigma^2}$. Let the minimizing value of t be t_1. Define $E_1 = t_1 - 2\sigma^2 t_1^2) > 0$. For $t = t_1$, the bound on the sum is exponentially decreasing with $d^{ij}(L)$.

We have established two conditions on t. Equation (4.34) requires that $t < \frac{1}{2\sigma^2}$ and (4.36) requires that $\frac{W}{2}\log_e(1-2\sigma^2 t) + t\gamma > 0$. We have shown there always exists a value of the bias term γ in the metric, which results in this sum in the bound going to 0 exponentially with $d^{ij}(L)$, irrespective of the rate R. [124] used a similar condition to bound the rate R that sequential decoding can support, but actually this is only a consequence of the metric selected. In particular [124] used the Fano metric where γ is a function of R, and hence a bound on R was obtained.

4.2.4.4 *Case B:* $0 \leq L' \leq L$

We now proceed to analyse the second sum in (4.31).

$$d^i(L') - d^j(L) = \sum_{k=1}^{WL'}(x_k^i - y_k)^2 - \sum_{k=1}^{WL}(x_k^j - y_k)^2 \tag{4.39}$$

$$= -d^{ij}(L') - 2\sum_{k=1}^{WL'} n_k(x_k^i - x_k^j) - \sum_{k=WL'+1}^{WL}(n_k - (x_k^j - x_k^i))^2 \tag{4.40}$$

We define $s = d_2(L') - d_3(L, L')$. $d_2(L') = -2\sum_{k=1}^{WL'} n_k(x_k^i - x_k^j)$ is distributed as $\mathcal{N}(0, 4\sigma^2 d^{ij}(L'))$. $d_3(L, L') = \sum_{k=WL'+1}^{WL}(n_k - (x_k^j - x_k^i))^2$ and hence has a non-central χ^2 distribution with $N = W(L - L')$ degrees of freedom, non-centrality parameter $d^2 = d^{ij}(L) - d^{ij}(L')$. $d_2(L)$ and $d_3(L, L')$ depend on non-overlapping segments of the i.i.d noise and hence are independent. Thus, the mgf of s is given by the product of mgfs,

$$\phi_s(t) = e^{2\sigma^2 d^{ij}(L')t^2} \frac{e^{-(-d^{ij}(L') + d^{ij}(L))\frac{t}{(1+2\sigma^2 t)}}}{(1+2\sigma^2 t)^{\frac{W(L-L')}{2}}}, \; Re\{t\} > -\frac{1}{2\sigma^2} \tag{4.41}$$

We define $\epsilon = d^{ij}(L') + \gamma(L' - L)$. Using the Chernoff bound, for $t \in \mathbb{R}, t > 0$,

$$\sum_{L'=0}^{L} P(d^i(L') - d^j(L) > \gamma(L' - L)) = \sum_{L'=0}^{L} P(s > \epsilon) \tag{4.42}$$

$$\leq e^{-d^{ij}(L)(\frac{t}{(1+2\sigma^2 t)})} \sum_{L'=0}^{L} e^{d^{ij}(L')(-t + 2\sigma^2 t^2 + \frac{t}{(1+2\sigma^2 t)})} .e^{-(L'-L)[t\gamma - W\frac{1}{2}\log_e(1+2\sigma^2 t)]} \tag{4.43}$$

For $t > 0$, $(-t + 2\sigma^2 t^2 + \frac{t}{(1+2\sigma^2 t)}) > 0$. Further, $d^{ij}(L)$ monotonically increases with L. Thus, we can upper bound (4.43) by replacing $d^{ij}(L')$ by $d^{ij}(L)$.

$$\sum_{L'=0}^{L} P(s > \epsilon) \leq e^{-d^{ij}(L)(t-2\sigma^2 t^2)} . \sum_{L'=0}^{L} e^{-(L'-L)[t\gamma - W\frac{1}{2}\log_e(1+2\sigma^2 t)]} \quad (4.44)$$

$$\leq e^{-d^{ij}(L)(t-2\sigma^2 t^2)} \frac{e^{L[t\gamma - \frac{W}{2}\log_e(1+2\sigma^2 t)]}}{1 - e^{-[t\gamma - \frac{W}{2}\log_e(1+2\sigma^2 t)]}} \quad (4.45)$$

$$\leq H(t) e^{L\phi} e^{-d^{ij}(L)(t-2\sigma^2 t^2)} \quad (4.46)$$

where $H(t)$ is independent of L and $\phi = t\gamma - \frac{W}{2}\log_e(1 + 2\sigma^2 t)$. Note that $\phi \geq 0$ since $\gamma > W\sigma^2$ and that ϕ is convex in t. For $t < \frac{1}{2\sigma^2}$, $(t - 2\sigma^2 t^2) > 0$. We can optimize this bound over $t < \frac{1}{2\sigma^2}$. Let t_2 be the optimizing value. Define $E_2 = t_2 - 2\sigma^2 t_2^2 > 0$. For $t = t_2$ this bound decreases exponentially with $d^{ij}(L)$.

4.2.4.5 Combined Bound

Since (4.38) and (4.46) show that each of the two sums in (4.31) decreases exponentially with $d^{ij}(L)$, so does the LHS of (4.31). Define $\mu = \min(E_1, E_2)$.

$$P([L^j > L]) \leq H(t_2) e^{L\phi} e^{-d^{ij}(L)(E_2)} + G(t_2) e^{-d^{ij}(L)(E_1)} \quad (4.47)$$

$$\leq e^{-d^{ij}(L)(\mu) + L\phi} (H(t_2) + G(t_1)) \quad (4.48)$$

$$\leq \beta e^{-\mu d^{ij}(L) + L\phi} \quad (4.49)$$

where our choice of t_1, t_2 and γ ensures that $\mu > 0$. We see that the bound does not depend on the specific path j but only on its distance $d^{ij}(L)$. So, we define $[L^d > L]$ to be the event that a particular path of length L in S_u at a distance d from path i is extended. Since there are only a countable number of paths $j \in S_u$, the distance $d = d^{ij}(L)$ for some path j can only take a countable number of values. We form a discretization of the real line from $d_c(r)$ to ∞, every δ step length. We define the summation to be over $d_c(r), \ldots, d_c(r) + k\delta, \ldots$. Then n_d denotes the number of incorrect paths in S_u whose distance from the correct path

at length L lies between d and $d+\delta$. We assume that $n_d \leq e^{d\xi} < e^{d\mu}$ (i.e., $\xi < \mu$).

$$P(\cup_j [L^j > L]) \leq \sum_{d=d_c(L)}^{\infty} n_d P([L^d > L])$$
$$\leq \sum_{d=d_c(L)}^{\infty} \beta e^{-(\mu-\xi)d+L\phi} \leq \beta \frac{e^{-(\mu-\xi)d_c(L)+L\phi}}{1-e^{-(\mu-\xi)\delta}} \tag{4.50}$$

For this bound to be practical, we require that $(\mu-\xi)d_c(L) - L\phi$ be positive, that is $d_c(L) > r_d L$, where $r_d = \frac{\phi}{\mu-\xi}$. This presents a lower bound on how fast the CDF must grow. Based on our initial definitions, $L = \frac{1}{V}\log_2(N_u)$, and so we have the final bound from (4.28).

$$P(C_u > N_u) < \beta \frac{e^{-\frac{(\mu-\xi)}{V} d_c(\log_2(N_u)) + \frac{1}{V}(\log_2(N_u))\phi}}{1-e^{-(\mu-\xi)\delta}} \tag{4.51}$$

Increasing γ beyond $W\sigma^2$ in Case A increases μ but also increases ϕ. These two equations show the effect of the choice of bias on computational complexity of sequential decoding. The bias γ could be chosen to balance the computations in Case A (4.38) and Case B (4.46) (perhaps by performing a line search). The optimal value will depend on the specific CDF of the code. We do not optimize γ in this sub-section, but use R. Another choice of γ was derived in the previous sub-section and that can be analysed using the methods below as well.

4.2.4.6 Upper Bound on Error Probability

We have reduced the bound in (4.49) to exactly the same form as in [124]. We can use the same arguments as [124] to bound the probability of error. We present their arguments for completeness.

A proto-error EP_u is the event that an incorrect path j in S_u has a higher metric at the point of merger with the correct path i than the minimum metric along path i beyond u [125], [124]. No decoding error can occur without a proto-error. E_u is the event that

error occurs in S_u. Let path j merge with path i at depth L_m.

$$\begin{aligned} P_{E_u} &\leq P_{EP_u} \\ &\leq \sum_{j \in S_u} P(M^j(L_m) > M^i_{min})) \\ &\leq \sum_{j \in S_u} \beta e^{-\mu d^{ij}(L_m) + L_m \phi} \end{aligned} \tag{4.52}$$

where we used (4.49). Now, from the lower bound on growth $d_c(L_m) > r_d L_m$

$$P_{E_u} < \sum_{j \in S_u} \beta e^{-d\omega} = \sum_{d=d_{free}}^{\infty} w_d \beta e^{-d\omega} \tag{4.53}$$

where $\omega = r_d \mu - \phi$. We discretize d again from d_{free} with some parameter δ', and define w_d to be the number of paths in S_u with merging distance between d and $d + \delta'$. By definition, $d \geq d_{free}$ where d_{free} is the free distance of the tree code. If $w_d \leq e^{d\rho} < e^{d\omega}$,

$$P_{E_u} < \eta e^{-(\omega - \rho) d_{free}} \tag{4.54}$$

where $\eta = \frac{\beta}{1 - e^{-\delta'(\omega - \rho)}}$.

4.2.4.7 *Discussion of results*

As mentioned in the introduction, our proof holds even when the transmitted symbols are dependent. We consider a few special cases. Recent work in detection for sensor network applications [15] and decoding lattice codes [121] has resulted in interest in sequential decoding for new applications. An implication of our results to these problems is that rather than using the Fano metric, which works well for good convolutional codes, the metric bias should be tuned for each application for better performance (see for example [103]). Reference [121] raised the question of pre-processing such that lattice codes can be decoded efficiently by sequential decoding. We suggest that pre-processing that optimizes the squared Euclidean column distance function could be useful in improving the performance of sequential decoding. In [15], a model was suggested for sensor net-

works with the environment being discrete and each sensor has a specific field of view. The sensor output could be a real value (such as the weighted sum of sensed regions with additive Gaussian noise). Modifying the sensor to obtain a larger column distance function can result in faster and more accurate performance of sequential decoding. The accuracy increase is because any real system must have a practical bound on the number of allowed computations as shown in Chapter 5 [115].

The case of transmission of real values across Gaussian ISI channels is also of interest in applications such as wireless communication and large memory ISI channels in data storage systems [127]. One example is shown in Figure 4.13. Two channels were chosen as shown in Figure 4.13(a) to have the same partial energy sums. The reason for highlighting the partial energy sums is that this is the metric used to evaluate channels for performance with sequential detection, such as the minimum phase processing front end for Decision Feedback Equalizers(DFEs). Partial energy sums are important for DFEs since DFEs cannot backtrack and the probability of error is determined by the probability of first error. Sequential decoders can backtrack and hence their performance is determined by the CDF. Analysing (4.51), we see that if we require the computation to be small (N_u small) this depends on $d_c(\frac{1}{V}\log_2(N_u))$, which is the initial part of the CDF. This implies that the initial part of the CDF is the most important in deciding the performance of sequential decoding. Since higher rate codes can be expected to have slowly increasing CDF (because codewords are packed closer together), there will exist a maximum rate beyond which sequential decoding is inefficient. This is shown by the improved performance of the channel with one negative tap weight. This channel has a better (larger) CDF, in the beginning, which grows worse later. Yet it has the same partial energy sums as the channel with all positive weights.

In this sub-section we have proved a bound on computation and error rate for a par-

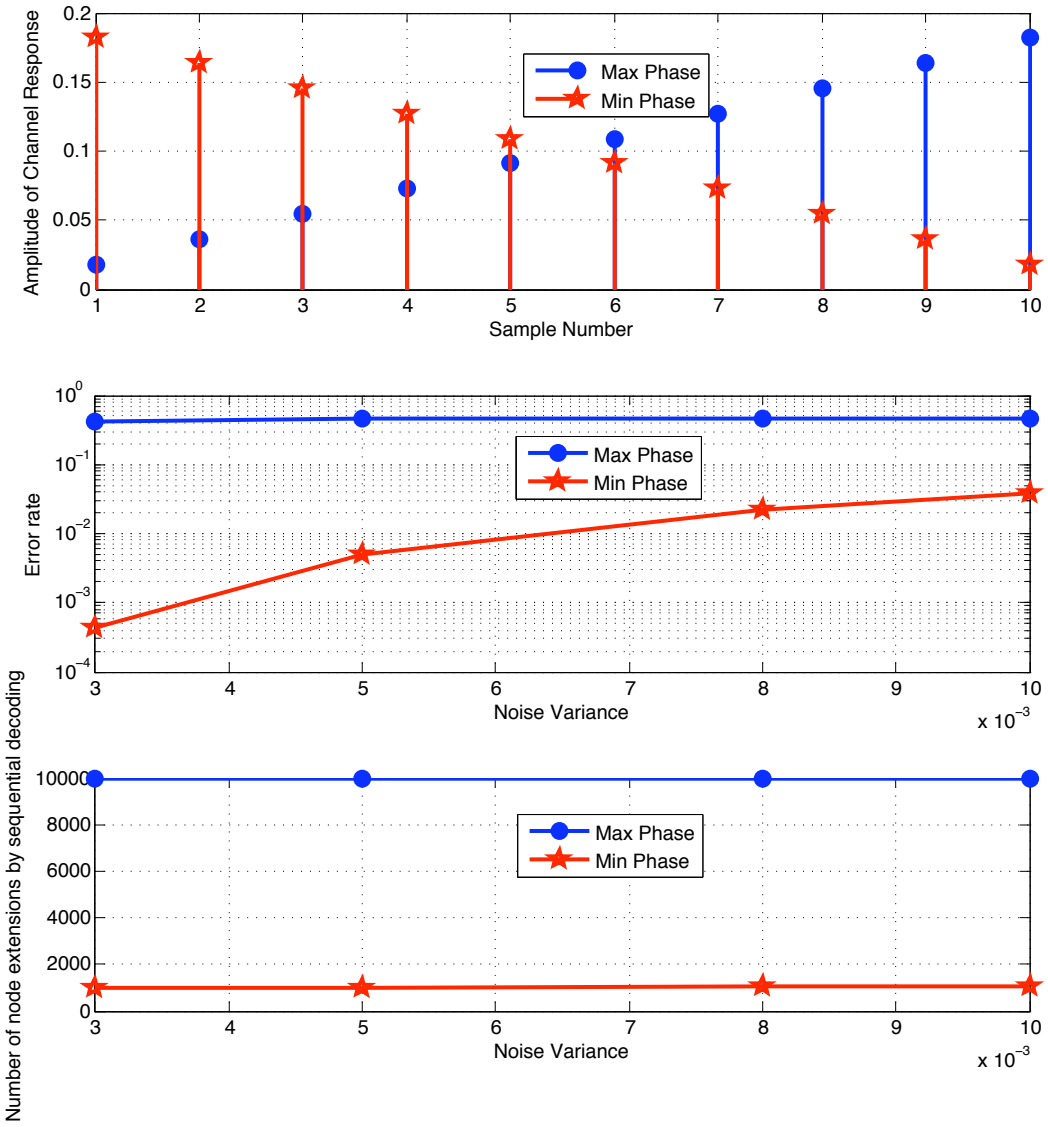


Figure 4.13: Performance improvement of sequential decoding with rapid initial CDF over an ISI channel - a comparison of minimum phase and maximum phase channel response. Number of node extensions was limited to 10000 for a length 1000 target vector

ticular tree code with real symbols transmitted over an AWGN channel. While we have followed [124], we made substantial changes to account for the continuous Gaussian channel. We proved that for a metric of the form of (4.27) (parameterized by γ), there exists a value of γ such that the number of computations in each incorrect subset is bounded by a quantity which exponentially decreases with the column distance function $d^{ij}(L)$, as long as the code meets certain criteria. The criteria are that (i) $d^{ij}(L)$ grows faster than a specified linear function of L and (ii) the number of partial paths j of length L having $d^{ij}(L)$ between d and $d+\delta$ be limited.

In [124] a bound on rate R was obtained, but we point out that this was a consequence of their particular choice of metric, the Fano metric. Comparing the conditions on t in (4.34), (4.36) and (4.41), we note that the parameter γ represents a trade-off which determines the performance of sequential decoding. Optimizing γ for particular codes would be an interesting direction of future research.

While here we have demonstrated the usefulness of the CDF in predicting algorithm computational complexity in Chapter 5 we will show how to use this results to design measurements that can be decoded efficiently with sequential decoding.

We finally mention that almost all the results here about sequential decoding can, in some form, be extended to cases with mismatch and uncertainty. Since these results will again only be useful in a case-by-case basis depending on situation specific parameters, we do not display these results here.

4.3 ALGORITHMS FOR HANDLING UNCERTAINTY

In this last section we present some suggestions on how to use the intuition behind the analysis of Chapter 3 to design practical decoders that work with model and parameter uncertainty. We are motivated by physical applications such as sensor networks [83], [111]

CDMA systems [66] or DNA microarrays [128], where all the system parameters may not be known and have to be learned from data or provided by an expert.

Sensor network applications, such as, habitat monitoring [129], or border monitoring [111] are susceptible to variations in sensor parameters [130] and other uncertainties. Calibration in these networks, requires a controlled input or knowledge of the correlations between sensor measurements from a model of the phenomenon being sensed [131]. However, there are often model errors due to mismatches in control input, error in physical model, errors in reference sensors, or absence of good sensor models [130]. Perfect calibration can become expensive or infeasible, especially in large networks. In addition, sensor networks have a large number of sensors, and in these must be individually calibrated before accurate large-scale detection can be performed. Even if calibration is possible, environment perturbations can cause drift and uncertainty over time. For sensor networks to gain widespread use, we still require application oriented performance guarantees.

In this section, we will study two algorithmic approaches to the problem of detection with uncertain model parameters. The first approximates the computationally infeasible decoder analyzed in Section 3.3.4 using the CE method from Section 4.1. We will see that decoding with uncertainty is possible at the cost of some increase in decoding complexity. Our second approach is a variant of the Expectation Maximization (EM) algorithm [132] that has found many uses in the communication [133] and coding [134] when parameter uncertainty is a problem. We present a slight variant that directly estimates the input (which is what we are interested in) as opposed to estimating the uncertain parameter (which is the output of the conventional EM algorithm). This approach seems cleaner since it directly estimates the object of interest and must have been discovered in the past. Since we were unable to find it in the literature we present the derivation here for completeness.

4.3.1 The cross entropy method with uncertainty

The CE method from Section 4.1 reduces to the ML decoder as the number of samples at each iteration $N \to \infty$. The analysis in Section 3.3.3 suggested a computationally infeasible decoder, which maximized the average of likelihoods (4.55), for situations with parameter uncertainty. We will now see how to use the CE method to approximate this new decoder as well.

In Section 3.3.3 we analyzed a decoder that maximized

$$s^{\mathcal{C}}(\mathbf{x}_i, \mathbf{Y}) = \frac{1}{|\Theta|} \sum_{\theta \in \Theta} P_\theta(\mathbf{Y}|\mathbf{x}_i) = \frac{1}{|\Theta|} \sum_{\theta \in \Theta} \prod_{u=1}^{n} P_\theta(Y^u|x_i^u) \tag{4.55}$$

and showed that this decoder was (in some sense) minimax optimal. Since the CE method is a general discrete optimization algorithm, we design a CE method to maximize (4.55). Note that we can also derive a threshold or stopping criterion for the CE method as we did in Section 4.1. We now study its performance and compare it to the CE method without uncertainty, which uses the score $P_{\theta_*}(\mathbf{Y}|\mathbf{x}_i)$ with the correct value θ_* of the parameter θ. A simple simulation experiment demonstrating some interesting computational properties is shown below.

We simulate the case with binary inputs, $\mathcal{V} = \{0, 1\}$ and probability of 1, $p = 0.4$ and the function Ψ is the SUM of measured inputs. We use two noise models. In $P^{(1)}_{Y|X}$, $P^{(1)}_{Y|X=1}$ (the first row) is $[.7\ .15\ .05\ .05\ .03\ .02]$ and all other rows are circular shifts of the first row. In $P^{(2)}_{Y|X}$, $P^{(2)}_{Y|X=1}$ (the first row) is $[.02\ 7\ .15\ .05\ .05\ .03]$ and all other rows are circular shifts of the first row. We choose these odd looking $P_{Y|X}$s so that there is a substantial difference in the noise models, yet both still have the same capacity. We use input vector length $k = 200$.

In Figure 4.14 we see a validation of the analysis of decoding with uncertainty in Section 3.3.3. The CE method with the metric from (4.55) approaches the performance of the

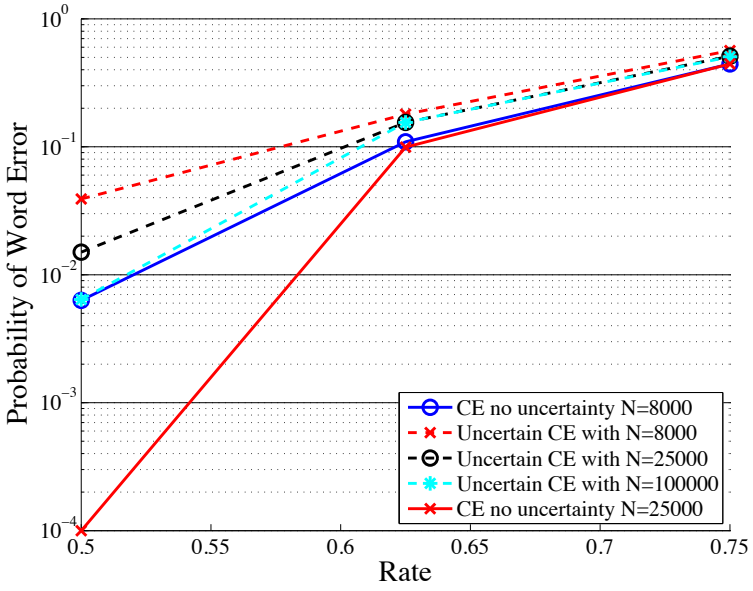


Figure 4.14: Performance of CE with uncertainty decoder for $p = .4$, $D = .04$, $c = 5$, SUM measurements and $P_{Y|X}$ consists of circular shifts of $[.7\ .15\ .05\ .05\ .03\ .02]$. We see that an increase in complexity, as measured by number of samples N, is required to achieve performance of CE decoder without uncertainty.

CE method. This handling of uncertainty does not come for free in that the complexity of the CE method with uncertainty, as measured by number of samples N in each CE iteration, is orders of magnitude larger when there is parameter uncertainty at the decoder, even when they both achieve the same error rate. Developing a better understanding of how uncertainty effects the computation, accuracy and number of measurements trade-offs is a very interesting direction of future research.

4.3.2 Expectation Maximization for detection with parameter uncertainty

In Chapter 3, we analyzed optimal MAP or ML decoders for decoding with uncertainty, which are too computationally complex to be practical in general. In the previous subsection, we looked at the CE method that used the intuition behind the analyzed decoders to develop a practical algorithm for handling uncertainty. We also showed some of its

properties with simulations.

In this sub-section we take a different approach based on the EM algorithm [132]. But our derivation is different, in that, we do not estimate the unknown (channel/measurement) parameter but directly detect the input with uncertainty. We have included the derivation here for completeness since we could not find a similar approach in the literature. Evaluating this algorithm and comparing it with other similar algorithms is not undertaken here.

We first outline the basic version of the EM algorithm. The 'data' is divided into two parts the good data D_g and the missing data D_b. We want to estimate a parameter θ, and start with an initial estimate θ^0. We define a quality function

$$Q(\theta;\theta^i) \;=\; E_{D_g}[\log p(D_g, D_b;\theta|D_g;\theta^i] \tag{4.56}$$

We then choose as our next estimate of θ, $\theta^{i+1} = \arg\max_\theta Q(\theta;\theta^i)$.

The EM algorithm is used to estimate uncertain parameters in channel coding, using exactly the formulation above. However, in our case we are not interested in estimating the parameter θ, but rather detecting the input vector $\mathbf{v}$. This leads to a different form of the EM algorithm, which we derive below.

In our model, the unknown sensor parameter θ corresponds to the missing data D_b. The noisy measurements $\mathbf{Y}$ correspond to the good data D_g, and the parameter to be estimated (the original θ) correspond to the environment $\mathbf{V}$. With this in mind,

$$\begin{aligned} Q(\theta;\theta^i) &= E_{D_g}[\log p(D_g, D_b;\theta|D_g;\theta^i] && (4.57)\\ &= \sum_{\theta\in\Theta} p(\theta|\mathbf{Y},\mathbf{V}_i)\log p(\theta,\mathbf{Y}|\mathbf{V}) && (4.58)\end{aligned}$$

Consider the first term,

$$\begin{aligned} p(\theta|\mathbf{Y},\mathbf{V}_i) &= \frac{p(\mathbf{Y},\mathbf{V}_i|p(\theta))}{p(\mathbf{Y},\mathbf{V}_i)} && (4.59)\\ &\propto p(\mathbf{V}_i)p(\mathbf{Y}|\mathbf{V}_i,\theta)p(\theta) && (4.60)\end{aligned}$$

We define this as,

$$r_t(\theta) = \prod_{j=1}^{k} p(V_i^j) \prod_{t=1}^{n} p(y^t|x_i^t, \theta)p(\theta) \tag{4.61}$$

which can be calculated, for fixed θ by counting the locations t with $y^t = y$ and $x^t = x$, $\forall\, y \in \mathcal{Y}$ and $\forall\, x \in \mathcal{X}$. Now,

$$\begin{aligned}
\mathbf{V}_{i+1} &= \arg\max_{\mathbf{V}} \sum_{\theta\in\Theta} r_t(\theta) \log p(\theta, \mathbf{Y}|\mathbf{V}) && (4.62)\\
&= \arg\max_{\mathbf{V}} \sum_{\theta\in\Theta} r_t(\theta) \log p(\mathbf{Y}|\mathbf{V}, \theta) && (4.63)\\
&= \arg\max_{\mathbf{V}} \sum_{\theta\in\Theta} r_t(\theta) \log \prod_{m=1}^{M} f_m(Y^m, V^m, \theta) && (4.64)\\
&= \arg\max_{\mathbf{V}} \log \prod_{m=1}^{M} \prod_{\theta} f_m^{r_t(\theta)}(Y^m, V^m, \theta) && (4.65)
\end{aligned}$$

where $p(\mathbf{Y}|\mathbf{V}, \theta)$ factorizes with M factors $f_m(Y^m, V^m, \theta)$.

We recall that in our measurement/sensor network problem these factors are

$$f_m(Y^m, V^m, \theta) = p_\theta(Y^m|\Psi(V^{m_1}, \ldots, V^{m_c})) \tag{4.66}$$

The maximization in (4.65) is an inference step. Exploring the performance of different inference algorithms, such as BP from Appendix B or the CE method, when combined with the EM algorithm, is a very interesting direction of future research.

While this chapter alternated between heuristics which draw intuition from our previous analyses, asymptotic analysis of algorithm performance and rigorous computational characterization of algorithms we hope that it sheds some light on the (complex) question of algorithm selection for sparse measurement systems. In the next chapter we will see examples of how these algorithms can be applied to real problems and, going further, show how the results of this chapter can be used to guide algorithm design.

CHAPTER 5

PRACTICAL EXPERIMENTS AND THE DESIGN OF SPARSE MEASUREMENT SYSTEMS

In this chapter we show how the analysis from Chapter 3 and the algorithmic understanding developed in Chapter 4, can be used fruitfully to study some of the applications we described in Chapter 2. In particular, we are interested in *designing* sparse measurement systems for these applications. We view a measurement system as including the nature of the input, the physics of the measurement process, the structure of the measurement graph and the noise introduced in the sensing. In addition, our definition of a system encompasses the algorithm used to detect the inputs being measured. We will see in this chapter how thinking about the measurement system as including both the physical parts and the detection algorithm can be used profitably in the design of sensing systems.

This chapter can be broadly divided into two parts. In the first part, consisting of Sections 5.1 and 5.2, we show how some of the structures, analyzed in Chapter 3, and the algorithms, developed in Chapter 4, perform in real sparse measurement applications. This establishes that sparse measurements arise in practice and that the algorithms we have studied have practical merit. Then, in Sections 5.3, 5.4, 5.5, 5.6, we discuss how our asymptotic

theory from Chapter 3 and analysis of algorithms from Chapter 4, lead to ways to improve the practicality and performance of these measurement systems. This will demonstrate how we can use the results of this thesis to inform the design of practical measurement systems.

5.1 PRACTICAL EXPERIMENT : GROUP TESTING FOR DRUG DISCOVERY

The use of group testing for drug discovery was described in Section 2.2. Our data is from a pooling experiment for a drug discovery application [1]. We describe the data collection from [1] below.

For these experiments, [1] used the MicroSource Spectrum 2000 collection. The compounds in the SPECTRUM Collection provide a range of biological activities and structural diversity especially for screening programs. All 2000 compounds were screened at the University of Michigan Center for Chemical Genomics to identify small molecules that inhibit the binding of $RGS4C$ with $G\alpha_0$. From these compounds a subset of size 316 of the most active compounds was used in the proof-of-concept pooling experiment. A mosquito robot (TTP LabTech, Melbourn UK) was used to perform the tests.

These 316 compounds were run in singleplex (each compound tested individually) and in multiplex (group testing). The group testing strategy or design was based on the Shifted Transversal Design (STD) [135] with pool size $c = 4$ (which corresponds to measurement size in our sparse measurement model) and rate $R = 1$. However, for our experiments we will use $R = \frac{4}{3}$ by dropping a fraction of the tests. This results in a sparse regular graph with $c = 4$ and $d = 3$. We will return to a comparison of this deterministic STD design with our random sparse regular ensemble design in Section 5.1.2.

The activities of the 316 compounds were tested individually twice (Singleplex 1 and 2) and are shown in Figure 5.1, reprinted with permission from [1]. The connection between

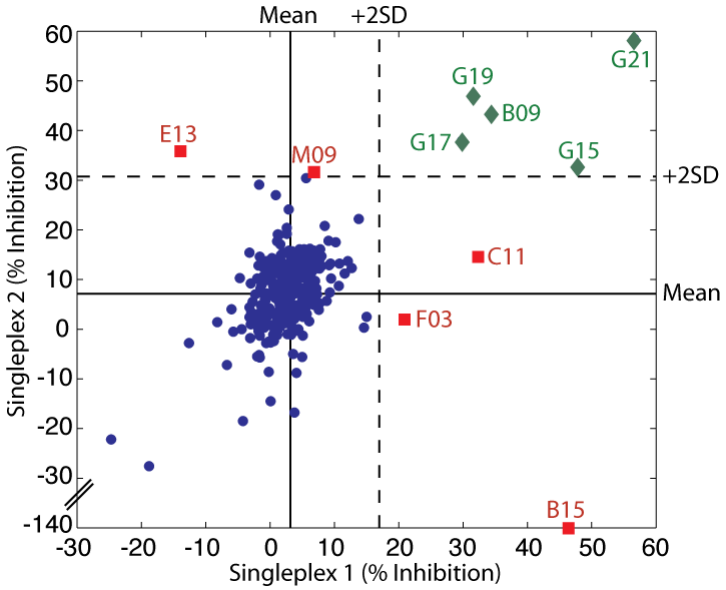


Figure 5.1: The %Inhibition (per-plate) for 316 compounds from the MS-2000 library from two separate singleplex (one-compound-one-well) runs. Reprinted with permission from [1].

the %Inhibition shown there and the activity of the compounds is explained in detail below. Compounds that showed activity in both runs were classified as 'consensus hits' (highlighted with green squares). The differences between the two indicates how noisy drug discovery testing can be, and why even rate $R = 1$ pooling may be useful if a reduction in the noise can be shown. We show results on rate $R = \frac{4}{3}$ pooling to demonstrate how group testing can *both* reduce the number of tests and compensate for noise.

In the rest of this section, we compare a few detection algorithms that aim to find which input chemicals are active given the florescence output from the pools. Unfortunately, even though we have the two singleplex results we do not know which chemicals are actually active because of the disagreement between the two individual tests. We can only hope that the output of our algorithms is consistent with the consensus hits from Figure 5.1. In addition, we do not have enough details on pooling process to construct good models for the noise. For the detection algorithms we will use simple Gaussian or Laplacian noise models.

So, given a difference in result, it will be unclear if this is because of model mismatch or because of the algorithm. We get around this in two ways.

The first approach is to compare the likelihood or cost of the input found by each algorithm, assuming that the model is correct. A better detection algorithm should return a lower cost estimate. The second approach is to consider the stability of the result. We see what happens to the solution when more measurements are added. We will actually see that all algorithms find the same two inputs as positive (G15 and G21) when all 316 measurements ($R = 1$) are used but have different outputs with $R = \frac{4}{3}$. Finally, we will also do some simulations, motivated by the experiments, to compare algorithms and structures.

5.1.1 CE vs. BP for practical pooling

We now compare the performance of two of the algorithms, the Cross Entropy method and Belief Propagation, on this real data. We already saw that CE was the better choice for many pooling-type problems with $c > 3$. Since that is the case here ($c = 4$) we expect to see this benefit here as well.

We are interested in finding which inputs are active and which are not - as opposed to finding the activity levels of all the inputs. This is realistic since any chemical marked as active will be re-screened many times in future tests so that exact activities levels can be assessed later. To model this we consider the input to be discretized to two levels, corresponding to active and inactive. Thus the input alphabet is $\mathcal{V} = \{0, 1\}$. The expected output florescence reading in each pool X^js for $j = 1, \ldots, n$ can be written in terms of the input activities V^is for $i = 1, \ldots, k$ as

$$\Sigma^j = \sum_{i \in \mathcal{N}(Y^j)} V^i \tag{5.1}$$

$$X^j = a\frac{\Sigma^j + b}{1 + \Sigma^j} \tag{5.2}$$

where $\mathcal{N}(Y^j)$ are the inputs pooled in pool j and a and b are unknown constants. This is just a re-writing of (2.1) which models the non linear measurement function inherent in the chemistry of pooling for drug discovery. The constant of proportionality is in general unknown and depends on the concentrations and re-activities of the chemicals involved in the test.

We estimate these constants from the positive and negative control wells that were a part of the test [1]. Given these constants we now have a sparse measurement network with $c = 4$ and a non-linear measurement function of the form of (5.2).

We do not have a way to estimate the noise model or the prior on the input being active or inactive. From the data we see that there are a number of spikes (both positive and negative) indicating that the noise is unlikely to be Gaussian in distribution. We enforce a sparsity prior on the inputs and a Laplacian penalty on the reconstruction error, leading to the following optimization problem, with regularization parameter λ.

$$\min_{\mathbf{V}} ||\mathbf{V}||_0 + \lambda||\mathbf{X} - \mathbf{Y}||_1 \tag{5.3}$$

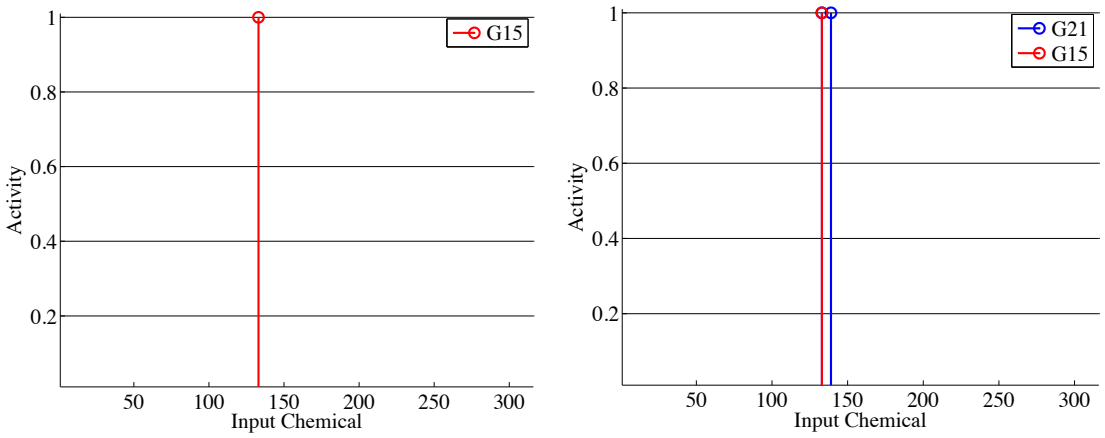
subject to

$$X^j = a\frac{\Sigma^j + b}{1 + \Sigma^j} \quad \forall j \in 1, \ldots, n \tag{5.4}$$

$$V^i \in \{0, 1\} \quad \forall i \in 1, \ldots, k \tag{5.5}$$

$$\Sigma^j = \sum_{i \in \mathcal{N}(Y^j)} V^i \tag{5.6}$$

We do not have estimates of the noise model and we set $\lambda = 1$ in (5.3) above. Because the above optimization problem factors according to the (local) constraints, we can use BP to solve it. In addition, we can also use the CE method to solve such discrete optimization problems. Hence, we proceed to compare these two algorithms. Another way to look at the above problem is stating it as an detection problem with a Laplace noise distribution,



(a) Result of using BP to solve Problem 5.3. Cost of solution 41.5964

(b) Result of using CE to solve Problem 5.3. Cost of solution 40.7535

Figure 5.2: A comparison of CE and BP on real pooling data from a drug discovery experiment

with appropriately chosen parameters. Different choices of the Laplacian parameters and the prior on inputs lead to different values of λ in the above problem.

The results of running BP and the CE method on the data are shown in Figures 5.2(a) and 5.2(b). We see that one of the chemicals $G15$ was found as active by both algorithms while CE found another $G21$ as an additional hit. The cost of the assignment found (according to (5.3)) by BP was 41.5964 and for CE was 40.7535 indicating that CE is the better search algorithm since it found a better assignment. Also, note that $G15$ and $G21$ were active in both the singleplex experiments in Figure 5.1.

5.1.2 CE vs. L1 for practical pooling

An alternative to the non-linear approach taken in the previous sub-section is to re-parametrize the above problem as a linear one [1] and apply ideas from Compressed Sensing [19]. Rather than deal with the output florescence directly, [1] estimated the unknown constants to calculate new measurements $\mathbf{T}$ that are noisy versions of the SUM of the inputs

exactly like our SUM model of pooling from the previous chapters.

$$T^j = \frac{ab - Y^j}{Y^j - a} \tag{5.7}$$

where $\mathcal{N}(Y^j)$ specify the same measurement structure as before. If we assume the noise is Gaussian distributed, we end up with the following problem,

$$\min_{\mathbf{V}} ||\mathbf{V}||_0 + \lambda||\mathbf{\Sigma} - \mathbf{T}||_2^2 \tag{5.8}$$

subject to

$$\Sigma^j = \sum_{i \in \mathcal{N}(Y^j)} V^i \tag{5.9}$$

$$V^i \in [0, 1] \;\; \forall i \in 1, \ldots, k \tag{5.10}$$

Kainkaryam [1] applied L1-regularized L2-minimization, solving the following relaxed problem.

$$\min_{\mathbf{V}} ||\mathbf{V}||_1 + \lambda||\mathbf{\Sigma} - \mathbf{T}||_2^2 \tag{5.11}$$

subject to

$$\Sigma^j = \sum_{i \in \mathcal{N}(Y^j)} V^i \tag{5.12}$$

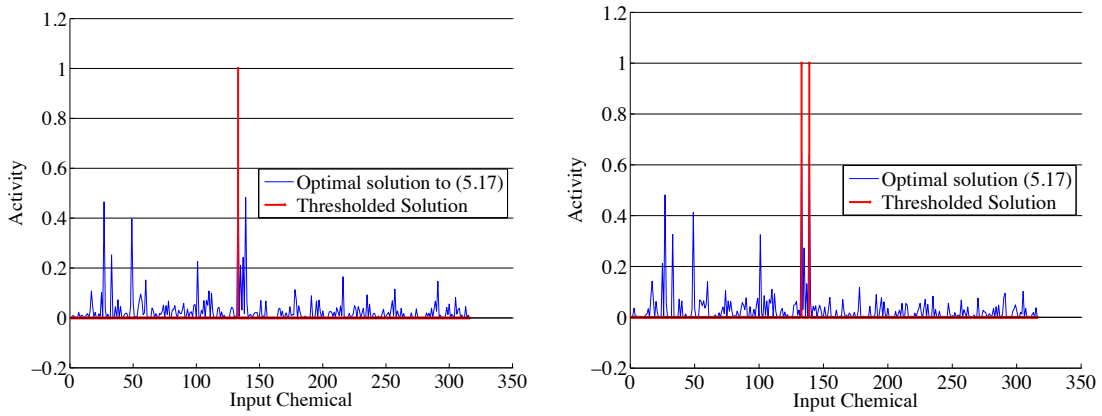
We note that the noise in the data does not seem to be Gaussian, especially after the non-linear transforms applied above and that the activities should lie between 0 and 1 in general. So we instead focus on the problem below.

$$\min_{\mathbf{V}} ||\mathbf{V}||_0 + \lambda||\mathbf{\Sigma} - \mathbf{T}||_1 \tag{5.13}$$

subject to

$$\Sigma^j = \sum_{i \in \mathcal{N}(Y^j)} V^i \tag{5.14}$$

$$V^i \in \{0, 1\} \;\; \forall i \in 1, \ldots, k \tag{5.15}$$



(a) Result with $\frac{3}{4}$ of the measurements. (b) Result with all the measurements.

Figure 5.3: Change in output of rounding the convex optimizer solution to Problem 5.16 as the number of measurements is increased

and its convex relaxation,

$$\min_{\mathbf{V}} ||\mathbf{V}||_1 + \lambda||\mathbf{\Sigma} - \mathbf{T}||_1 \tag{5.16}$$

subject to

$$\Sigma^j = \sum_{i \in \mathcal{N}(Y^j)} V^i \tag{5.17}$$

$$V^i \in [0, 1] \;\; \forall i \in 1, \ldots, k \tag{5.18}$$

This is a convex problem and can be solved efficiently [136]. Once we have a solution we round it using a threshold of $\frac{1}{2}$. We emphasize that this may not be completely fair to the convex optimization approach since output of the algorithm actually gives a much more detailed picture of the input. However, since we are, in general, interested in detection problems we follow this line of reasoning for now.

The discretized output of the convex optimizer is shown in Figure 5.3(a). We see there that it found the higher cost solution that BP also found in the previous sub-section. We also conducted an experiment where all 316 measurements are used, corresponding to a $R = 1$ pooling. In this case the output of CE and BP remained the same - as shown

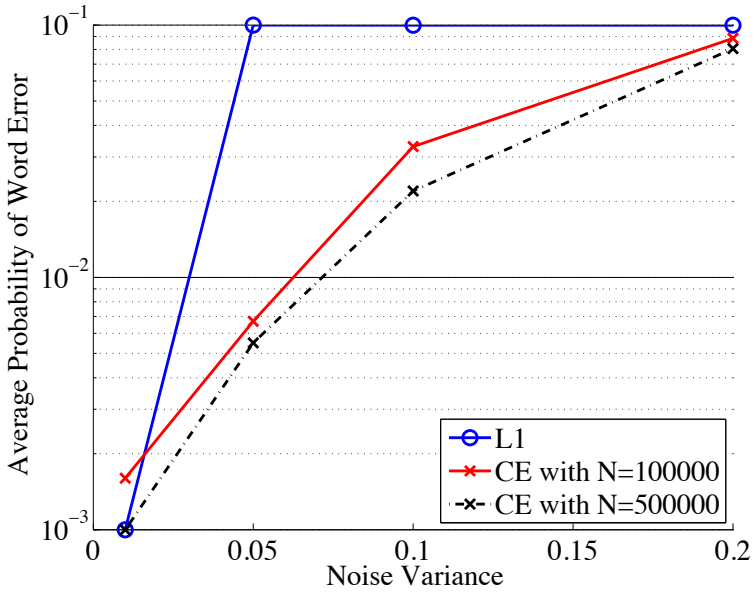


Figure 5.4: Comparison of CE and rounded L1 for Problem 5.8.

in Figures 5.2(a) and 5.2(b) - but the output of the rounded convex optimizer changed as shown in Figure 5.3(b). When all the measurements are used the convex optimizer returns $G21$ as well.

Since these results are somewhat unsatisfactory in terms of separating the algorithms in the absence of ground truth, we turn to simulations. We consider the case where the inputs are binary $V^i \in \{0, 1\}$, and the probability $p_V(V = 1) = 0.1$. The noise is Gaussian and we study the effect of changing the noise variance, corresponding to Problem 5.8. In Figure 5.4, we compare the output of the CE method and the solution obtained by rounding the solution of (5.11) to solve the non-convex problem (5.8). We see that here that for a fixed number of samples N rounding the convex solution has better performance for very small noise levels though the CE method is better for larger noise levels. We see that a substantially larger number of samples N is required for the CE method to perform well in low-noise situations, indicating high computational complexity compared to the convex optimization based decoder.

The results in this section demonstrate the applicability of the CE method to group testing problems. We now move on to a thermal sensing application to demonstrate the effectiveness of the SD algorithm.

5.2 PRACTICAL EXPERIMENT : OBJECT DETECTION USING THERMAL SENSORS

In this section we study an application of the contiguous model in a sensor network application. We consider an experiment using thermal sensors. We conduct an experiment using the MLX90601KZA-BKA thermal sensor as described in detail in Section 2.3.1.

We set the sensor field of view at 26°C by adding an aperture as shown in Figure 4.6. We used the model shown in Figure 2.2 for the sensor's sensitivity to radiation at different incidence angles. We obtained this model by fitting a function to the points provided in the sensor data sheet corresponding to the aperture we used.

In our experiments we used chemical heat pads as targets and pinned them on a flat surface. These pads achieved a temperature of about 50°C for an extended period of time, while the ambient temperature in the room was 23.7°C. The sensor was placed at a distance of 250mm from the plane and collected 1600 measurements of a 500mm$\times$500mm region. The sensor was mounted on two servo motors that panned and tilted the sensor. An example scan is shown in Figure 5.5 and Figure 5.6. Due to heat diffusion from the hot targets to nearby regions of the board, which is absent from our model, we apply a threshold to the data.

We processed the measurements using sequential decoding to obtain the detected field shown in Figure 5.7. The ground truth is shown in Figure 5.8. Using sequential decoding, only 2.7% of the grid blocks were misclassified with a sensor whose field of view included up to 100 grid blocks in each measurement.

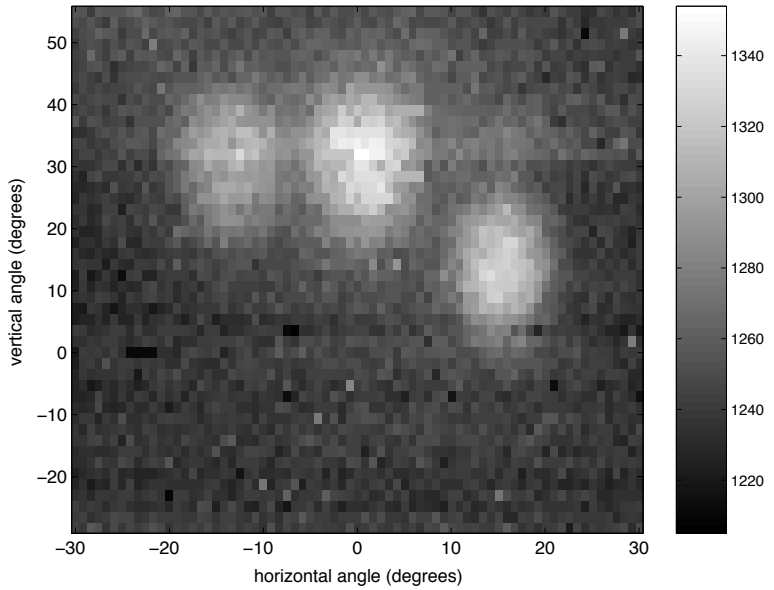


Figure 5.5: Raw thermal sensor data indicating presence of 3 targets.

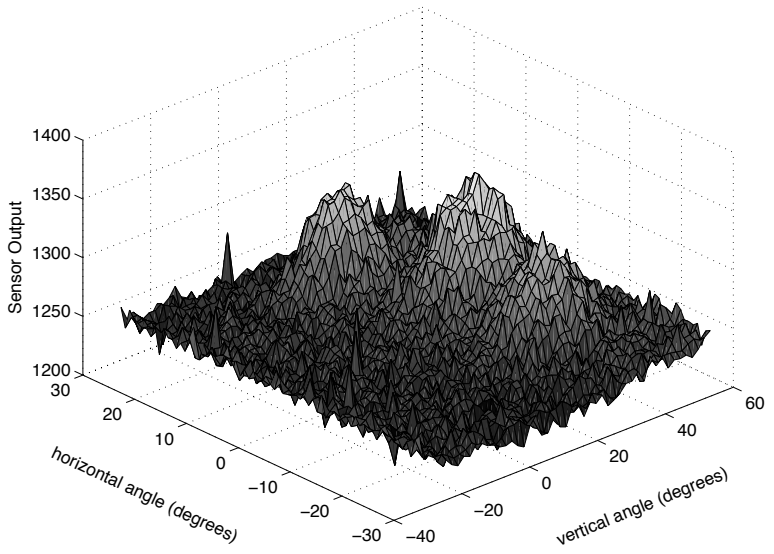


Figure 5.6: Raw thermal sensor data showing how noisy the readings are.

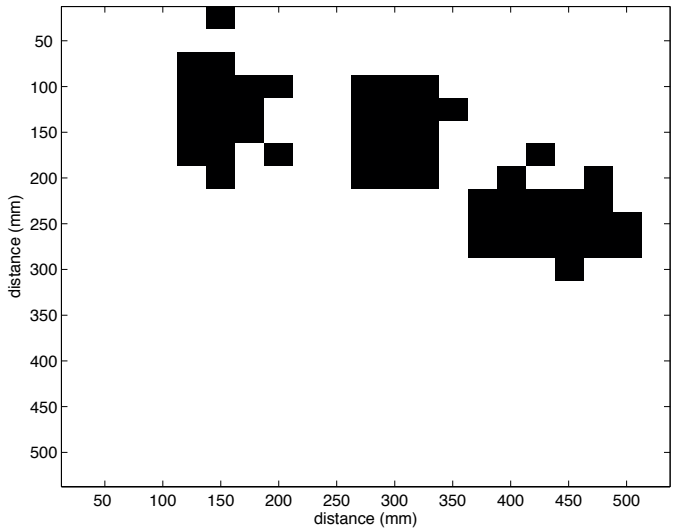


Figure 5.7: Sequential decoding estimate of target configuration using experimental data.

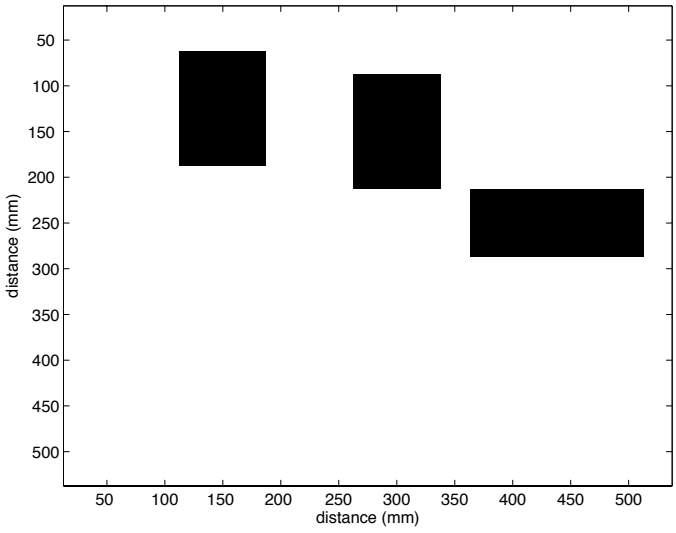


Figure 5.8: Ground truth from our experimental setup.

Now that we have demonstrated the applicability of our algorithms in very different sparse measurement problems we will see how the theory and algorithmic analysis we have developed can be used to guide better system design. We will show (i) how the random measurement analysis can be used to design better pooling structures and robust sensor networks and (ii) how our analysis of sequential decoding can be used to design algorithms and measurements with improved performance.

5.3 SYSTEM DESIGN : GROUP TESTING

Given a group testing problem, such as the problem of drug discovery discussed in Sections 2.2.1 and 5.1, the measurement function and most parameters like the prior and noise model are fixed. An experiment designer can control only some aspects of the problem, such as which compounds are pooled in which wells and how many compounds should be mixed in a single pool. We will now explore (through simulations) how the theory we have developed could be useful in the design of pooling strategies for drug discovery or genetic screening.

5.3.1 Random vs. Shifted Transversal Designs for pooling

In Section 5.1, compounds were pooled based on the Shifted Transversal Design (STD) of [135]. The STD is a combinatorial pooling design that aims to minimize the number of compounds that co-occur in different pools and to equalize the size of the intersection of different pools. It also has the added bonus that as long as a fixed (small) number of positives and read errors occur during the experiment, the true input can be unambiguously decoded.

We note that almost exactly the same intuition used by [135] has been used in the coding theory community [137, 138] for designing LDPC codes for BP based decoding and in the

design of measurement matrices for compressed sensing [139]. The algorithms described in [135] are also very similar to BP. We expect that the design and decoding strategies used in the coding theory community will be a good source of pooling designs in the future. However, for the LDPC code constructions to be practical in pooling applications, parameters like c, d and R need to be specified, which is exactly what our bounds provide.

We compare the STD design and the sparse regular measurement structures we described in Chapter 3. To recall, in order to generate a pooling design from the sparse regular ensemble we fix the pool size c and the number of times each input is pooled d (chosen based on the given number of input samples k, prior probability, noise model and most importantly, our bounds) and choose uniformly from all bipartite graphs in the configuration model. Structures drawn from this ensemble do not have any worst-case guarantees of robustness to noise. However, we note that the guarantees for deterministic designs, such as the STD, are substantially worse than performance on average. Thus, the utility of most pooling strategies is demonstrated by simulation to determine average error rates [70, 135]. The utility of using our bounds is that we can determine when the performance of a group testing experiment will be sufficient, without running time consuming simulations.

We consider in Figure 5.9 the case where the pooling outputs are very noisy, and inputs are sparse. In our terminology we consider the case where pool size is $c = 4$, rate $R = 1$, probability of positive is $p_V(V = 1) = p = .05$ and allowable distortion is $D = 0.008$. However, based on the study from Section 5.1, we consider the relatively short input lengths $k = 316$ as more representative of practical problem sizes. We see that random designs have excellent performance for noisy pooling with CE decoding. We can also see the improvement in BP performance when random pools are used instead of STD designs. In addition, we expect that if BP decoding is used, then for designs based on LDPC codes, at appropriate R, performance would be excellent (for large enough k) . Note that the bound

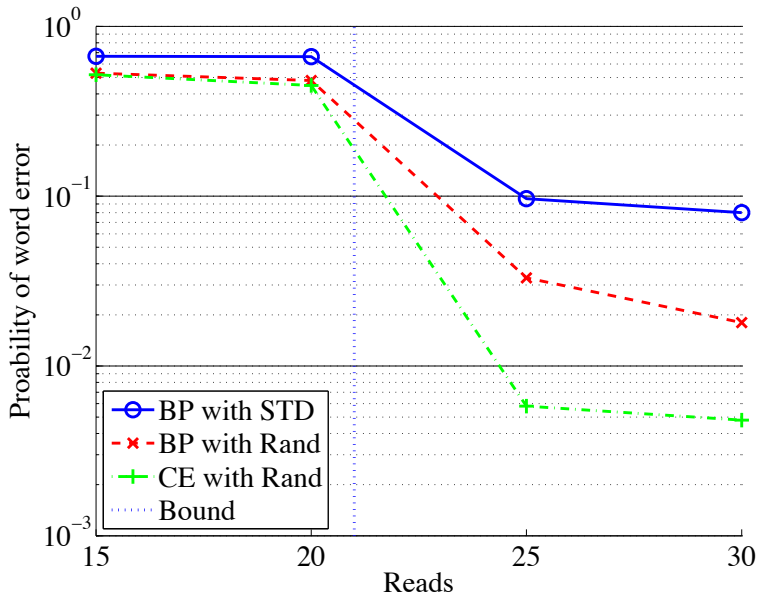


Figure 5.9: A comparison of random and Shifted Transversal Designs and an illustration of the benefit of CE decoding.

in Figure 5.9 is a good predictor of algorithm performance, though the error rate does not drop to near zero because of a combination of the high noise level and short input lengths.

Finally, we note that random designs have been suggested for different group testing problems [7, 140, 141] and derandomization strategies have been recently explored [41] leading to more choices for pooling designs for group testing. These results cannot be directly ported to general sparse measurement systems, such as the drug discovery application, without estimates of the number of measurements and pool size required for the particular noise and prior. Our bounds, $C_{LB}^{REG}(D)$, provide the rate R needed for a certain reconstruction fidelity D. For example, in a drug discovery pooling design the concentration of the chemicals can be changed at some increase in the cost of the experiments resulting in less noise in the pooling output. Similarly, in a genetic screening experiment increasing the number of reads (as discussed in Section 2.2.2) can reduce the errors at the cost of increasing time and resource usage. One fundamental question, that our bounds can

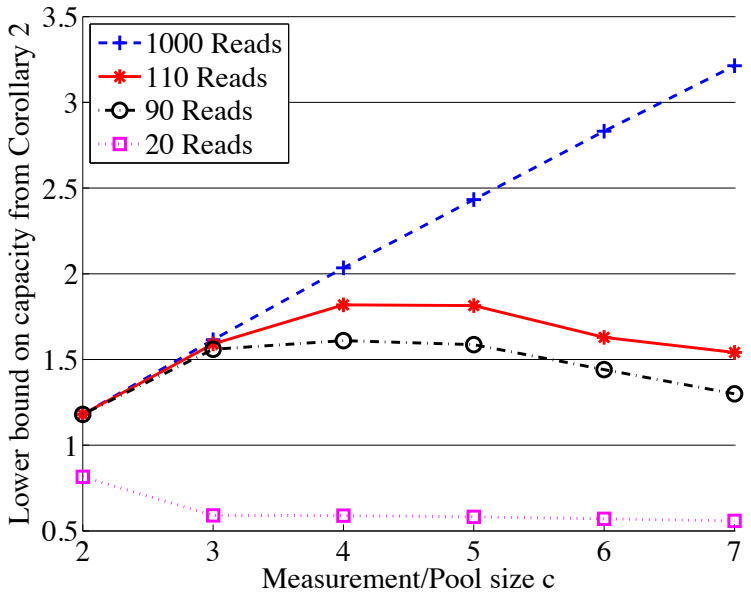


Figure 5.10: Optimal pool size for different reads or concentrations for $p_V(V = 1) = p = 0.1$ and $D = 0.002$. More reads correspond to lower noise. We assume no other source of noise, so a pool size of 1 would achieve a rate of $R = \frac{1}{1-\frac{D}{p}} = 1.0204$.

answer is, when is pooling useful in terms of the cost benefit trade-off ?

We demonstrate how our bounds can answer such questions for a particular setting of the parameters in Figure 5.10 with $p_V(V = 1) = p = 0.1$ and $D = 0.002$. Here again we consider the reads noise model in (2.2) from Section 2.2 with different reads r. Recall that for this model a pool size of $c = 1$ is considered to be noiseless - definitely not the case based on our experiments - but we choose this to illustrate that pooling can be used to reduce the number of tests required even in this situation. In Figure 5.10 we see how the optimal pool size increases as the number of reads increases. This is expected, since for low reads this signal dependent noise model penalizes larger pool sizes. This is useful in system design, since it tells us what would be the benefit of pooling and what pooling strategy to use in different situations without running time consuming simulations.

We look at the reads model with failures from Section 2.2.3 and (2.3) in particular. We

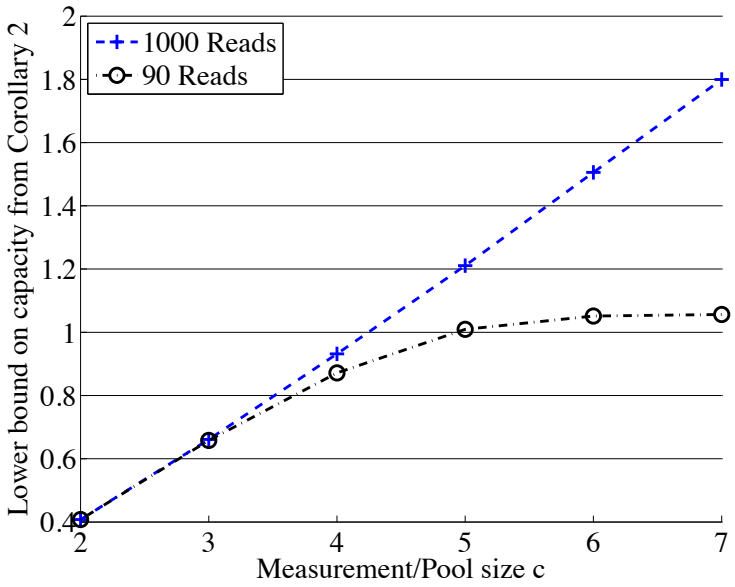


Figure 5.11: Optimal pool size for different reads or concentrations for $p_V(V = 1) = p = 0.1$ and $D = 0.002$. More reads correspond to lower noise. Here we assume a failure probability of .05 so a pool size of 1 would achieve a rate of $R = .25$.

expect that the benefits from pooling will increase as other noise sources - such as system noise and mechanical read errors - are included in the noise model. This is shown in Figure 5.11, where we see that even for moderate reads, $r = 90$, pooling reduces the number of tests more than four fold as compared to testing each sample individually which would require a rate of $R = .25$ to achieve a distortion $D = 0.002$. This gain increases as each test becomes more error prone, in terms of (2.3), as the failure probability p_f increases. Thus, pooling can not only reduce the number of tests required but also increase robustness to experimental noise.

5.4 SYSTEM DESIGN : SENSOR NETWORKS ROBUST TO UNCERTAINTY

From the practical experiments - both pooling and sensor network related - we learned that we cannot expect to have perfect knowledge of system parameters such as priors or

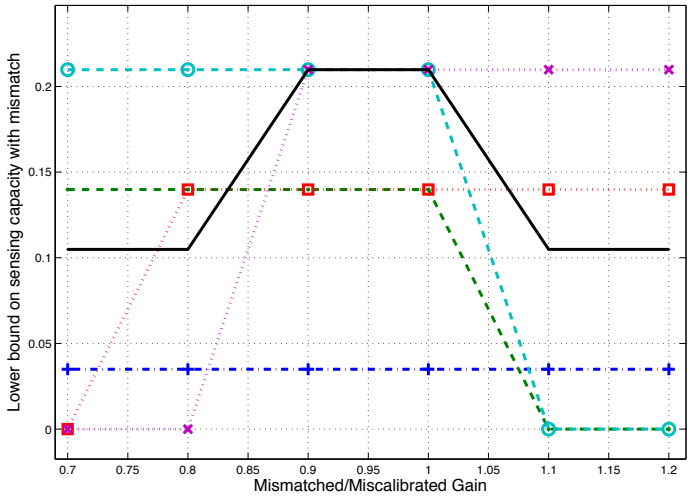


Figure 5.12: The use of heterogeneous measurements to compensate for mismatch. The dash/dot lines are rates achievable by measurements with different thresholds, at different mismatches in the gain. The true gain is 1. The solid line represents the rates achievable using a mixture of measurements.

measurement noise. We looked at the problem of mismatch and uncertainty from an information theoretic viewpoint in Section 3.3. We now consider measurement structures drawn from the arbitrary connections model. We will show how the insights from that theory can be used to design measurement structures robust to mismatch.

We look at a simple example where mismatch can cause a sharp drop-off in the performance of a measurement system (or equivalently a large increase in the number of measurements required to achieve a specified distortion), and where the bounds from the previous section can be used to guide the design of measurement systems. Each measurement measures $c = 5$ random locations. The measurement function takes an average of values within it's field of view and then applies a threshold. The detector assumes that the environment takes values $\{0, 1\}$, but actually the values are $\{0, g\}$, where g is a mismatched gain. The required rates for measurements with different thresholds, to achieve a

distortion of $D = .005$ are plotted with dash/dot lines in Figure 5.12. The sharp drop-off in rate indicates that the sensors are especially susceptible to the gain mismatch, which is not surprising given the threshold model. To ensure that the error criterion is always met despite the mismatches, we would need to use the measurements with the pessimistic 'blue dash-dot' threshold (near the bottom of the figure), which can achieve a rate of less than $.035$ which would require about 30 measurements per input. Alternatively, we can use a heterogeneous network, composed of n_t measurements with threshold t, for each measurement type t. We use Corollary 6 combined with Corollary 5 which allows us to calculate the achievable rate for such a heterogeneous ensemble. So, if we know that mismatches are possible, and if we know the rates achievable with mismatch (using the results of the previous section), then we could use a mixture of measurements with different thresholds, which achieves the black solid curve in the figure - a rate of more than $.1$.

This is a simple example where we can use the formulas derived to determine a combination of measurements so that the measurement network is robust to mismatch. Other kinds of mismatches can be analyzed similarly, and in many cases it can be seen that a mixture of heterogeneous measurements is more robust to mismatch and uncertainty than networks with a single kind of measurement.

5.5 SYSTEM DESIGN : MODIFYING THE ALGORITHM TO SUIT THE MEASUREMENTS

Returning to contiguous measurement networks, we show how the *theory* developed in Section 4.2.3.3 can be used to *improve algorithm performance* in a sensor network application. To recall, using Theorem 1, we first developed an intuitive understanding of the working of sequential decoding. This led us to the choice of appropriate metric as

$S_n = \sum_{i=1}^{n} \Delta_i$, where Δ_i is the increment in the metric as each sensor is decoded.

$$\Delta_i = \log_2[P^{\gamma}(x_{il})P_{Y|X}(y_l|x_i^l)] - \tag{5.19}$$

$$\sum_{l=1}^{n} \log_2[P^{\gamma^*}(x_i^l)P_{Y|X}^{\lambda^*}(y_l|x_i^l)] - R[H(\tilde{\lambda^*}||\lambda^{*'}) - H(\tilde{\gamma^*}|\gamma^{*'})] \tag{5.20}$$

There are two requirements of a metric to be appropriate for sequential decoding. The first is that the expected value of the metric increase as the decoding of the target vector proceeds down the correct branches of the binary hypothesis tree. The second is that the metric should decrease along any incorrect path in the tree that is more than a distortion D from the true target vector. We analyzed the metric derived and prove that it satisfies these requirements for appropriate choices of rate R and representative joint type λ^*.

The optimal decoding strategy in a contiguous sensor network application is Viterbi decoding. The computational complexity of Viterbi decoding increases exponentially in the size of the measurement c. However the *computational complexity* of sequential decoding is empirically seen to be largely independent of the measurement size c. This is illustrated in Figure 5.13, which compares the average running times of sequential decoding and Viterbi decoding. While Viterbi decoding is not feasible for measurements with large c, sequential decoding is practical. Many real world sensors such as thermal sensors or sonar sensors have precisely such a large measurement sizes, demonstrating the need for sequential decoding algorithms.

The primary property of sequential decoding is the existence of a computational cutoff rate. In communication theory, the computational cutoff rate is the rate below which average decoding time of sequential decoding is bounded. The complexity of Viterbi decoding on the other hand, is almost independent of the rate. We demonstrate through simulations that such a phenomenon exists for detection in large-scale sensing problems using the new metric. This leads us to an alternative to the conventional trade-off between computational

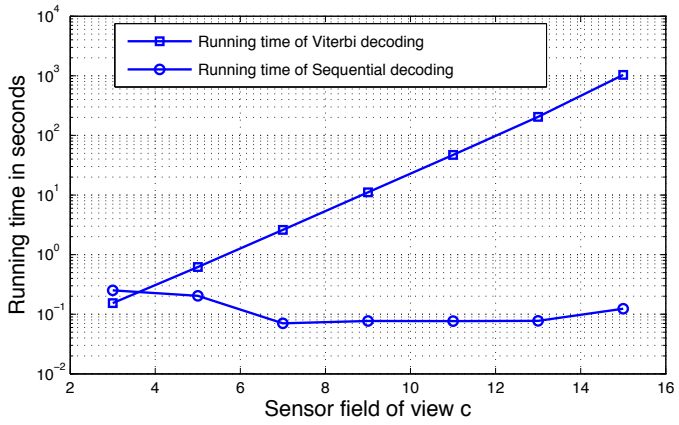


Figure 5.13: Comparison of running times of optimal Viterbi decoding and sequential decoding for different sensor fields of view c

complexity and accuracy of detection, where this trade-off can be altered by collecting additional sensor measurements, leading to algorithms that are both accurate and computationally efficient. The results of these simulations are shown in Figure 5.14.

Finally we compare the performance of sequential decoding when the new metric from (5.20) is used and when the Fano metric is used as a function of the tolerable distortion in Figure 5.15. The Fano metric does not account for the allowed distortion in any way and we can see that its performance is much worse. Our theoretical analysis showed that perfect reconstruction may not be possible from noisy measurements, requiring the specification of an allowed distortion. The results here show how to *use the distortion to design algorithms*. In addition, they demonstrate the applicability of the new metric we derived for SD.

5.6 SYSTEM DESIGN : MODIFYING MEASUREMENTS TO SUIT THE ALGORITHM

In this section, we continue our study of the contiguous model of sensor networks with a focus on the performance of SD for linear measurement functions Ψ. In these problems a network of linear sensors are used to solve a detection problem. The environment can be

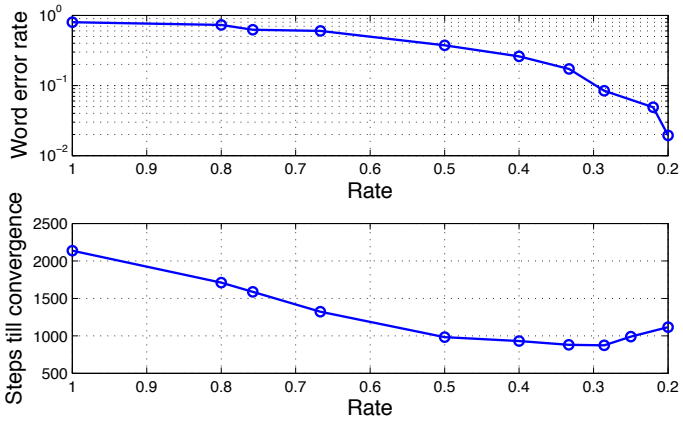


Figure 5.14: Running times and error rates for sequential decoding at different rates for $c = 15$

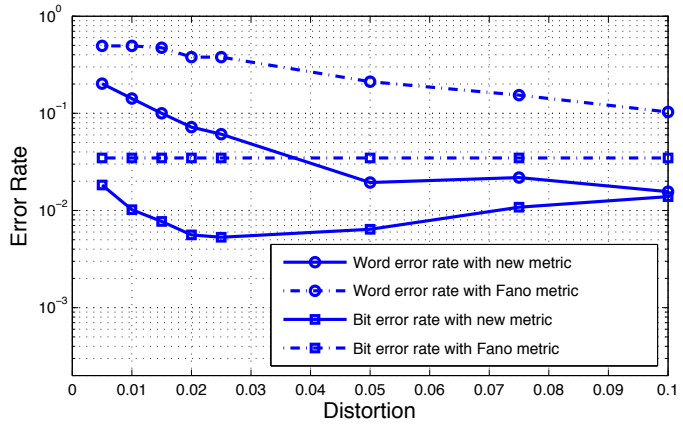


Figure 5.15: Comparison of error rates of new metric and Fano metric for different tolerable distortions at rate 0.2 for $c = 15$

modeled as a discrete grid, and each sensor measurement is effected by a large number of grid blocks simultaneously (defined by the field of view c of the sensors).

We demonstrate how we can pre-process the measured data or the sensors to suit the properties of a particular algorithm - in this case SD. We note that in general any processing of sensor measurements cannot improve the performance of optimal ML decoding (by the data processing inequality [101]). We show that certain processing can however improve the performance of a particular sub-optimal algorithm substantially. Since we are limited to sub-optimal algorithms in practice this is a useful/pragmatic approach to take.

Based on the analysis in Section 4.2.4, we see that sequential decoding will have better computational properties when used with sensors that have larger initial Column Distance Functions (CDFs). One possible approach to modifying the CDF is to pre-process sensor measurements to somehow 'simulate' sensors with the desired CDF.

As shown in Figures 5.16 and 5.17 certain sensors require a much larger SNR to achieve low error rate (and fast detection). Given sensor measurements from sensors of Type 2 or 3, we can process the data (using a minimum phase pre-processing explained later) to simulate sensors of Type 4 or 5, and improve performance. If an optimal detection strategy, in this case a dynamic program (Viterbi decoding) were to be used, both sensor configurations would have the same error rate at equal SNRs. Sensors of Type 2 or 3 are obtained by zeroing out some parts of the field of view of sensors of Type 1, and would have a higher error rate under optimal detection, but perform better under sequential decoding (for some SNRs). The pre-processing is purely for computational reasons to better suit the algorithm, leading to very interesting computational properties.

Our results from Section 4.2.4 derived results for sequential decoding of specific (i.e not random) convolutional codes with real symbols over Gaussian memoryless channels, or uncoded transmission over Gaussian Inter-Symbol Interference (ISI) channels. We note

that the contiguous sensor network with equal shifts can be viewed as just such a convolutional code. The results provide an upper bound for the distribution determined by the number of nodes C explored in each incorrect subset S as

$$P(C > N) < \beta e^{((k)d_c(\log_2(cN)))}, \tag{5.21}$$

for positive constants β and k. We recall that we define the column distance function (CDF) $d_c(r)$ to be the minimum over all pairs of environments of the Euclidean distance between the sensor measurement representations of two environments that differ in at least the first of r locations. The upper bound decreases exponentially with increasing CDF. Also, because of the $\log$ function, if lower computation is required the initial part of the CDF should increase rapidly. This can be intuitively justified as follows. A rapidly increasing initial CDF means that as we search the tree, if noise causes us to start along an incorrect sub-tree, within a few steps it looks very different from the correct path. The algorithm uses its metric to detect this and returns to the correct path without wasting too much computation. So sequential decoding will have better computational properties when used with sensors that have larger initial CDFs.

Finding the optimal pre-processing that maximizes the CDF of the code would be an ill-posed problem unless constraints are placed on what the pre-processing can do to the noise. One constraint would be that the SNR of the resulting code should be maintained the same. Another could be that the noise should remain white. Also, a search over all possible transformations of the data would be infeasible. So, in this initial work, we restrict ourselves to linear pre-processing of the sensor measurements.

At this point it would be interesting to draw a parallel with another algorithm that is used in communication receivers, the Decision Feedback Equalizer (DFE). The DFE is derived assuming that the feedback (consisting of previously estimated symbols) is correct.

However, there is no way for the decoder to backtrack and correct past mistakes. Thus, the error probability of the DFE is determined by the probability of first error. The DFE searches for the best linear pre-processing that keeps the noise white while maximizing the column distance when only one error is made. In contrast, the sequential decoder is allowed to backtrack and hence other error vectors need to be considered. Thus, sequential decoding seeks to maximize the initial CDF, where CDF is calculated against all error vectors.

We have found that maximizing the initial CDF is computationally complex, requiring computation exponential in the field of view of the sensor. However, when only error vectors with errors in just the first location are considered, the problem becomes much easier. The best linear pre-processing that keeps the noise white while maximizing the minimum Euclidean distance between two environments that differ in only the first location, is the min-phase solution i.e, the processing of measurements $\mathbf{Y}$ that converts the effective sensing function into a minimum phase signal. We show some results in Figure 5.17. If we use the min-phase processing, sensors of Type 3 can be converted into sensors of the Type 5. This gives us substantial improvement in the performance of sequential decoding, even though we have found a sub optimal preprocessing. The optimal filter to obtain a min phase equivalent is acausal and IIR and is stable if filtering is performed sequentially in a reverse direction. If the filtering is considered too computationally complex then there exist finite approximations such as the FIR-DFE and MMSE-DFE which are computationally much less demanding and as an added bonus are more suitable for distributed detection. The results in Figures 5.16 and 5.17 shows how filtering to produce a min phase filter can substantially improve the performance of SD.

The method presented works very well for many sensor functions. However, we discuss a special case where it fails because of an interesting property of functions. The solution turned out to be the min-phase spectral factorization of the PSD of the linear sensing func-

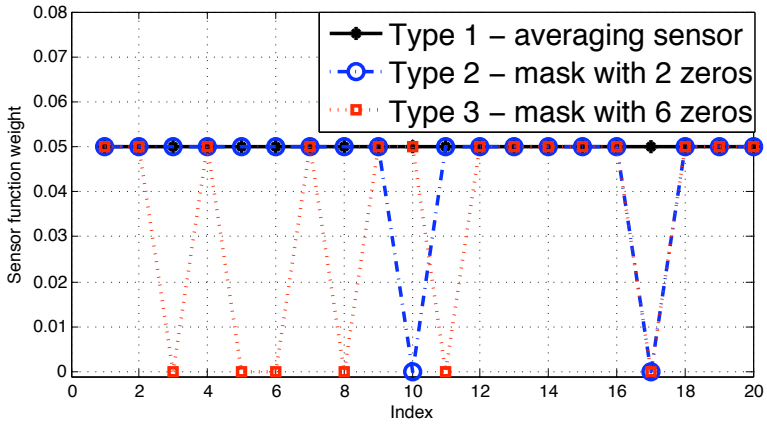


Figure 5.16: Performance improvement of sequential decoding with rapid initial CDF over an ISI channel - a comparison of minimum phase and maximum phase channel response. Number of node extensions was limited to 10000 for a length 1000 target vector

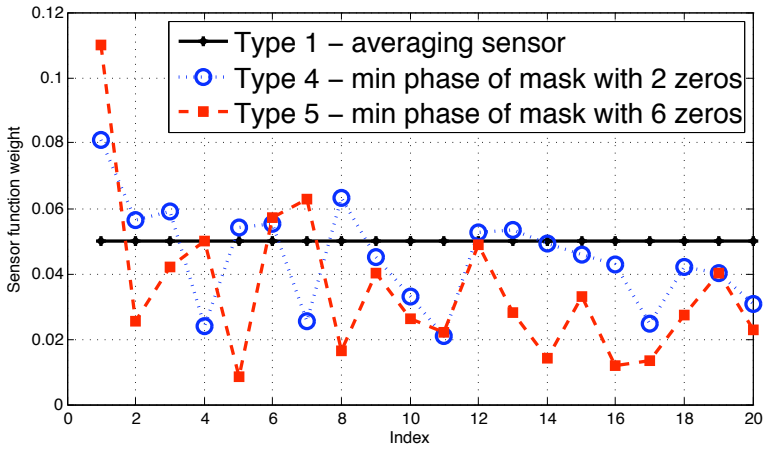


Figure 5.17: Performance improvement of sequential decoding with rapid initial CDF over an ISI channel - a comparison of minimum phase and maximum phase channel response. Number of node extensions was limited to 10000 for a length 1000 target vector

tion. One interpretation of the min phase solution is that it is obtained by moving all zeros of the original transfer function inside the unit circle. In situations where the zeros of the sensing function are close to or on the unit circle then the min-phase equivalent and the original sensing function will be largely similar and there is not much gain in the min-phase modification. Some sensor functions are symmetric in the spatial dimension, and coefficients do not grow very rapidly from one coefficient to the next. The z transform of such sensing functions results in a special kind of palindromic equation. From the fundamental theorem of palindromic polynomials, such equations have all their zeros on the unit circle and hence the min-phase solution will be the same as the original symmetric function. One example is the sensing function of Type 1 (the averaging sensor). In such cases, we need to modify the sensor to break the symmetry of the sensor.

In many 'real' situations we cannot modify the weights arbitrarily. We can however sometimes use masks in front of the sensor to zero out certain weights. Once the symmetry is broken the min-phase processing can be performed as before. For example zeroing out 6 weights (2 weights) in sensors of Type 1 leads to sensors of Type 3 (Type 2) and the min phase processing leads to sensors of Type 5 (Type 4) resulting in substantial improvements in error rate and speed.

An important point needs to be made here. At the beginning of the section it was mentioned that the motivation for the processing was to improve the performance of the sequential decoding algorithm not the performance of optimal ML decoding. The processing we are introducing (by masking weights) actually *reduces* the SNR of the system. Thus the new modified sensor could perform worse if optimal ML decoding were used. For some SNRs we see that masked sensors of Type 3 perform worse than the original with sequential decoding. However, the results in Figure 5.17 show that we can gain a substantial improvement in speed and accuracy of sequential decoding with the modified sensors after

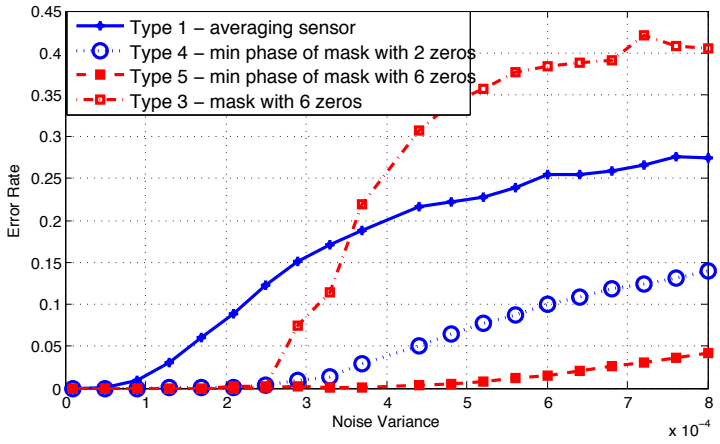


Figure 5.18: Performance improvement of sequential decoding with rapid initial CDF over an ISI channel - a comparison of minimum phase and maximum phase channel response. Number of node extensions was limited to 10000 for a length 1000 target vector

min-phase processing. Previously, we tried to find the optimal pre-processing that did not change the SNR. The results of this section lead to the conclusion that in many cases a better trade-off of performance and accuracy can be obtained if we are allowed to decrease the SNR. This is another direction of future research - finding the optimal CDF while trading off SNR. Thus, the performance of sensor network detection with sequential decoding can be improved significantly by (a) modifying sensors and then (b) pre-processing the data to suit the sequential decoding algorithm.

We now briefly discuss preliminary extensions of these ideas to 2-D sensing tasks. In Section 5.2 a thermal sensor was set up to scan a vertical surface with some regions at a higher temperature than the background. The thermal sensor is very low resolution, and gives a weighted average of the temperature of all regions in its field of view. We wished to resolve blocks at a much higher resolution. One way of handling such 2-D tasks is to convert them to 1-D sensing problems by scanning the 2-D environment with a (i) horizontal raster scan (ii) vertical raster scan or (iii) a zig-zag scan. Once the 2-D problem

has been reduced to a 1- D problem sequential decoding can be applied. Raster scans preserve 2-D convolution as a 1-D convolution when appropriately padded.

The intuition from our results here is that the choice of the scanning procedure can be made such that the column distance function is maximized. This change based on the shape and size of the environment being sensed and the shape and size and weights of the field of view of the sensor. It is interesting to note that this choice of scanning direction is a form of data processing to suit the properties of sequential decoding and would not change the performance of the sensor network with optimal maximum likelihood detection. Another approach to 2-D task is to apply the min-phase approach from the previous section. However, there is no spectral factorization theorem for 2-D matrices, and so arbitrary 2-D sensing matrices cannot be converted to min-phase form. One feature we can exploit in this setting is the fact that in some sensors the z-transform of the sensing function can be approximately factored into a product of two functions, one for each dimension. Each dimension can then be converted into a min-phase form, resulting in a 2-D min- phase sensor. The best direction to scan would now depend on the CDFs of each 1-D factor. In cases where the sensing function cannot be factorized or when either one or both functions are palindromic sequences, we can modify the sensing function with the use of masks to obtain a min-phase or factorizable sensing function.

We close by repeating the message of this chapter - considering the entire set of measurements and the decoding algorithm as a single system to be *jointly* designed allows us to achieve better performance with limited resources. While this chapter had a few examples of how such system design could be achieved we expect that a similar approach could work in many other problems of practical interest.

CHAPTER 6

CONCLUSIONS, DISCUSSIONS AND FUTURE WORK

This thesis was motivated by practical detection problems that arise in applications such as sensor networks and high throughput screening. In particular the focus was on using sparse measurement structures to model these applications. We made the case that modeling the sparse nature of the measurements, where every observation is a (stochastic) function of a small number of inputs, is critical. The work described in this thesis can be grouped into three parts : Analysis, Algorithms and finally Applications and System Design. We discuss each one separately below.

6.1 ANALYSIS OF SPARSE MEASUREMENT SYSTEMS

In this thesis, we analyzed the fundamental limits of performance of sparse measurement systems. Our primary theoretical insight came from a parallel between sparse measurement systems and communication systems. We see that the measurements form a kind of channel code, a parallel first suggested by Rachlin et. al. This insight allowed us to use the probabilistic method to analyze the limits of performance of sparse measurement systems motivated by the information theoretic analysis of Shannon. However, extensive

modifications were required to handle correlations that arise due to the constrained structures and fixed measurement functions. This correlation structure of the measurements, which arises because of the fixed sparse measurement structure and the given measurement function, differentiates our methods from many other applications of the probabilistic method in coding theory. Our primary *theoretical* contribution was an expression for 'sensing capacity' that is general enough (much more so than [32]) to allow for many of the constraints that arise in practical systems, such as fixed field of view in a sensor network, limited amount of per person sample in a biological assay, fixed measurement functions and parameter uncertainty or model mismatch. This expression can be used to characterize the number of measurements sufficient to overcome these practical limitations and meet specified performance requirements.

The first specific contribution of this thesis, which is a special case of the general theorem, is a lower bound on the sensing capacity of sparse regular measurements. These structures were shown to be important in group testing experiments. This bound was not possible using the techniques in [32]. We compared the predictions of the bound to simulations of practical algorithms, such as the L1 decoder and the CE method, and saw that the bound is an excellent predictor of algorithm performance.

Using the general expression we also obtained bounds on the capacity of different sparse measurement structures. We showed that the structure of the measurements - the number of times each input is measured and the pattern in which inputs are sensed - is important in deciding the capacity of sparse measurement systems.

Based on our experiments with real sensor networks and group testing for drug discovery, we saw that measurement calibration is an important problem in practical detection applications. Motivated by this, we used our general theorem to analyze the performance of sparse measurement systems with model mismatch and parameter uncertainty.

Our second specific contribution, was a definition of the sensing capacity for systems with mismatch and uncertainty. We gave lower bounds on the sensing capacity of these systems as a special case of our general theorem. This demonstrated that it is still possible to detect an input reliably, *provided a sufficient number of measurements are available*. This presents a useful design choice in measurement systems where increasing the number of measurements can make the system *robust* to calibration errors. We also showed how using a mixture of sensors can make a measurement system more robust to parameter uncertainty than when a single kind of sensor is used.

Our third theoretical contribution was a converse, or upper bound on sensing capacity, which was left as an open problem in [32]. This converse was shown to be tight in some interesting situations. In other scenarios, the gap between the achievable rates and the converse were not substantial. Together with the lower bounds on capacity, the converse provided a way to estimate the sub-optimality of particular algorithms.

This thesis demonstrates that the parallels between communication systems and measurement systems extends beyond the problems explored in [32]. However, all the theoretical results were asymptotic and the guarantees were probabilistic. This may not be suitable for applications such as screening for diseases, which are matters of life and death. An interesting direction of future research is drawing from the extensive work in coding and information theory to design and analyze sparse measurement systems with provable worst case guarantees at finite lengths.

6.2 ALGORITHMS FOR SPARSE MEASUREMENT SYSTEMS

Motivated by the successful use of the theoretical insights from parallels to communication theory, we studied algorithmic connections between detection in sparse measurement systems and decoding of channel codes. In this direction, we extended the idea of sequen-

tial decoding of convolutional codes and the Cross Entropy (CE) method for rare event simulation to detection with sparse measurements. The primary *algorithmic* contributions of this thesis were the analysis of the accuracy and computational complexity of sequential decoding for specific sparse measurements, a method to speed up Belief Propagation for a large class of commonly occurring measurements and a demonstration that the theoretical bounds are an excellent predictor of the performance of the CE method.

One striking algorithmic insight that arose in our work is the *computational cut-off* phenomenon in sparse measurement systems. We showed that when a sufficiently large number of measurements are available, both sequential decoding and the CE method have a sharp empirical performance transition, becoming computationally efficient *and* accurate. This computational cut-off suggests a novel trade-off in the design of sparse measurement systems, where the number of measurements can be increased to *substantially* reduce the computational cost of detection. We rigorously characterized the computational performance of sequential decoding for sparse measurements and analytically showed the presence of a computational cut-off. We analyzed the computational properties of SD in detail and derived an expression for the complexity of SD, that highlights the effect of the measurement function on the complexity of detection.

We demonstrated, by simulations, that the Cross Entropy method often performs substantially better than inference algorithms such as Belief Propagation for sparse, regular measurements. We used our theoretical analysis to provide a stopping criterion for the CE method and showed how to use the analysis to set parameters in the algorithm. This provides an interesting example of how the theoretical analysis can guide algorithm design. We also showed through simulations that the computational cut-off phenomenon extends to the CE method as well.

An important direction of future research here is developing an analytical characteriza-

tion of the appearance of the computational cut-off rate for different measurement structures, such as the sparse regular structure, and for different algorithms, such as the CE method.

6.3 APPLICATIONS AND SYSTEM DESIGN

To demonstrate the impact of the ideas presented in this thesis we showed results on two diverse real-world applications. In particular, this thesis we successfully apply the theoretical and algorithmic results to real measurement data. Our results show why sparse measurement systems are both feasible and necessary for different tasks. These practical demonstrations of sparse measurement systems is one of the most important contributions of this thesis.

The first application we showed results for, was the use of sparse measurements structures in a pooling experiment for high throughput drug discovery. We showed that the CE method performed well on real data from the screening experiment. In simulations we showed that sparse regular measurement structures had much better performance than the STD designs used in [1]. We also showed that the performance of CE was substantially better than BP in many scenarios of interest, especially when noise levels are high.

In the second application we demonstrated how a cheap thermal sensor, which is designed to estimate the gross temperature of a large object, could be used to estimate multiple targets. This was achieved by taking many measurements from different locations and fusing the information using the SD algorithm. To enable this application we developed a realistic physics based model of the thermal sensor.

We used the expression we derived for the complexity of SD, to design measurement systems that are 'easy' to decode. We also showed how to pre-process data to substantially improve the computational properties of SD.

The primary *engineering* insight of this thesis was showing that the theory can be used to improve algorithm performance and design systems with better guarantees. This insight came from our view of the measurement system as including not only the measurements but also their structure, the physics of the input and the particular detection algorithm used. For example, the theory tells us that perfect reconstruction may not be possible, given noisy measurements, and also which types of errors are the most likely. We show how incorporating this information into inference algorithms can detection performance.

The critical contribution of this part of our work is the idea that systems can be designed together with detection algorithms not just to improve performance, but also decrease detection complexity. An important direction of future research is developing a better understanding of how to design measurement systems for other algorithms such as BP or the CE method and for other measurement structures.

Appendices

APPENDIX A

OPTIMAL ALGORITHMS FOR INFERENCE IN GRAPHICAL MODELS

We now proceed to describe some optimal algorithms for detection in sparse measurement systems. This section contains no novel contributions and is presented for completeness and as a motivation for local message passing algorithms studied later in Appendix B. The reader who is familiar with work on graphical models [51] can safely skip this section.

We describe a general method to derive optimal algorithms that exploit the factorization represented in factor graphs, such as the one in Figure 2.1. Though we are only interested in sparse measurement systems, we describe the general algorithm since it motivates the BP algorithm of Appendix B.

Given a factor graph, we know that it represents a joint distribution over the random variables of the form

$$p(\mathbf{v}) \propto \prod_{i=1}^{n} f_i(y^i, v^{\mathcal{N}_i^1}, \ldots, v^{\mathcal{N}_i^c}) \tag{A.1}$$

where y^i and $v^{\mathcal{N}_i^1}, \ldots, v^{\mathcal{N}_i^c}$ are the assignments to the variables connected to factor f_i. We can use this value of $p(\mathbf{v})$ in (1.6) or (1.7) to compute the marginals or the MAP assignments. The direct approach involves computing and marginalizing over a joint distribution over $|\mathcal{V}|^k$ variables which is computationally impractical (for large k) *irrespective* of the

underlying sparse measurement structure. We would prefer algorithms that used the underlying structure to perform inference as efficiently as possible.

Variable elimination [51] is one such algorithm (or procedure) that exploits the factorization structure to interleave the product operations from (A.1) with the sum operations in (1.6) or the product operations from (1.7). This is achieved by exploiting the distribution of products over sums and products [142]. Below is an outline of the variable elimination algorithm for a particular choice of input variable V^t.

1. Choose an ordering of variables $V^1, \ldots, V^{t-1}, V^{t+1}, \ldots, V^k$.

2. Pick the next variable from the ordering to eliminate. Call this variable V^{now}.

3. Collect all factors involving V^{now}. Multiply them together to create a new factor f_{now}. Delete the original factors that were collected.

4. Marginalize V^{now} out of f_{now} by computing $g_{now} = \sum_{V^{now}} f_{now}$. Add this factor back into the set of all factors.

5. As long we still have variables to eliminate, return to step 2.

Note that larger and larger factors could be generated in Step 3 above. The size of the factors actually generated depends on the choice of order in which variables are eliminated. The minimum factor size over all possible choices of variable orderings corresponds to the tree-width of the original graph which can be thought of as a measure of complexity of inference in a particular graphical model. Unfortunately, even if the tree-width of a graph is small (and correspondingly inference is *easy*) finding the optimal ordering is NP-complete.

Ignoring this small setback, since heuristics for choosing elimination orderings are quite good in practice [143], we will now describe how to generalize the above algorithm to perform inference over all V^ts simultaneously. A junction tree is a computational structure

derived from the probabilistic structure represented by a graphical model or Bayes net that essentially pre-compiles the joint distribution represented by the graphical model into a data structure that supports the simultaneous execution of a large class of queries. For a complete discussion of this algorithm, see [51]. This data structure can be thought of as a specific kind of tree (with the Running Intersection Property or RIP) where each node corresponds to groups of variables from the original measurement system.

Given a junction tree the Shafer-Shenoy algorithm [144] is an iterative algorithm that updates probability estimates by exchanging 'messages' locally among nodes connected in the junction tree. This leads to the following definition for the message updates. Suppose we have two clusters $\mathcal{B}$ and $\mathcal{C}$ connected in the junction tree. Let $\mathbf{u}$ be the set of variables assigned to some some factor in $\mathcal{B}$ and $\mathbf{v}$ be the set of variables assigned to some factor in $\mathcal{C}$. Let $\mathbf{E}$ be the set of edges in the junction tree, and $\phi_{\mathcal{B}}$ be the product of all the factors assigned to node $\mathcal{B}$. Note that since $\phi_{\mathcal{B}}$ is the product of factors it is a function of all variables that occur in any of the factors assigned to $\mathcal{B}$. The message sent from $\mathcal{B}$ to $\mathcal{C}$ is a function of $\mathbf{u}$ which are the variables that occur both in factors assigned to $\mathcal{B}$ and in factors assigned to $\mathcal{C}$.

$$\mu_{\mathcal{B}\to\mathcal{C}}(\mathbf{u}) = \sum_{v\in\mathcal{B}\setminus\mathcal{C}} \phi_{\mathcal{B}}(\mathbf{u}\bigcup\mathbf{v}) \prod_{(\mathcal{A},\mathcal{B})\in\mathbf{E},\mathcal{A}\neq\mathcal{C}} \mu_{\mathcal{A}\to\mathcal{B}}(\mathbf{u}_{\mathcal{A}}\bigcup\mathbf{v}_{\mathcal{A}}) \tag{A.2}$$

where $\bigcup$ denotes the union operation, and the notation $\mathbf{v}_{\mathcal{A}}$ indicates the subset of the variables in $\mathbf{v}$ that are present in some factor assigned to $\mathcal{A}$. The cluster beliefs at a cluster $\mathcal{C}$ after any iteration, or in particular at the end after convergence are,

$$\beta_{\mathcal{C}}(\mathbf{u}) = \phi_{\mathcal{C}}(\mathbf{u}) \prod_{(\mathcal{B},\mathcal{C})\in\mathbf{E}} \mu_{\mathcal{B}\to\mathcal{C}}(\mathbf{u}_{\mathcal{B}}) \tag{A.3}$$

It can be shown that these updates results in correct and convergent local beliefs that solve (1.6), that is $p_{\mathcal{C}}(\mathbf{u}) \propto \beta_{\mathcal{C}}(\mathbf{u})$. Because of the structural properties of the junction tree, we

can show that Shafer Shenoy algorithm is equivalent to running Variable Elimination in all directions simultaneously. In general the space-time complexity is exponential in the width (size of largest cluster) of the junction tree, which again relates back to the tree-width of the original sparse measurement factor graph.

For most sparse measurement structures, such the ones analyzed in Chapter 3, lower bounds on the tree width are known to be quite large (of the order 50-80 for nominal k and n, see for example [53]), and so the junction tree algorithm is infeasible. It is however possible to run for some sensor network or contiguous measurement structures where sensors have very limited field of view - where it is equivalent to the well-known BCJR [145] or Viterbi dynamic programming decoding algorithms. We will discuss the performance and computational properties of the Viterbi algorithms when we study sequential decoding in Section 4.2.

Since optimal algorithms (like Viterbi decoding) are computationally infeasible we move on to practical detection algorithms. In general we would like to develop an understanding of when each kind of detection algorithm is suitable, and in particular we are interested in how they perform on the different sparse measurement structures that have practical applications.

APPENDIX B

BELIEF PROPAGATION

We now consider Belief Propagation (BP) an algorithm that performs inference by the exchange of messages between variables and their factors until convergence. Our main contribution here is the simplification of the calculations of BP, for certain measurement functions Ψ, which is described in Section B.1. We also indicate how the asymptotic analysis of BP based on Density Evolution [58] can be extended to sparse measurement systems using the idea of 'Perfect Projection' from [146].

There is a special measurement structure for which the junction tree algorithm from Appendix A is computationally feasible - when the original factor graph is a tree with low tree-width. In this case the nodes in the junction tree correspond to merging a measurement with all its measured inputs. This resulting tree satisfies all the required properties a junction tree must satisfy. The tree width of this (derived) graphical model is the maximum number of children of any parent in the tree.

The Shafer-Shenoy updates can now be broken down into messages exchanged between factors and variables in the original factor graph resulting in the following messages passed between nodes in the original factor graph [147]. The message passed from variable to

factor is

$$\mu_{V^t \to f_h}(v) = \prod_{h' \in \mathcal{N}_t} \mu_{f_{h'} \to V^t}(v) \tag{B.1}$$

and the message passed between a factor and a connected variable is

$$\mu_{f_h \to V^t}(v) = \sum_{\mathbf{v}' \setminus v^{t'} : t' \in \mathcal{N}_h \setminus t} \left(f_h(\mathbf{v}', v) \prod_{v^{t'} : t' \in \mathcal{N}_h \setminus t} \mu_{V^{t'} \to f_h}(v^{t'}) \right) \tag{B.2}$$

Since these updates are now just in terms of messages exchanged between nodes in the original graph there is no procedural reason that the same updates cannot be run on a factor graph that is not a tree. This algorithm is often called 'loopy BP'. We no longer have guarantees about the convergence or correctness of this algorithm like we did for the Shafer-Shenoy algorithm, but the algorithm has been seen to work well in practice [148].

For large enough input length k, sparse regular graphs (like the ones in Chapter 3) are tree-like up to a fixed constant depth. In addition BP has been shown to be near optimal in Low-Density Parity Check (LDPC) codes [92]. Most LDPC codes, as we discussed in Section 2.4, are a special case of our sparse regular model with no noise and no allowed distortion. So we expect that BP will perform well when the input length is large and the pool size c is small (since BP is exact for trees). A comparison of the bound and the performance of BP for small $c = 3$ is shown in Figure B.1. These results also show that the bounds we derived in Chapter 3 are meaningful since they reasonably predict the trends of BP showing that they can be profitably used in system design. However, we can see that the bound predicts that a rate of around 1.63 is achievable while BP requires a rate less than 1.5 for the word error rate to drop significantly. We could draw the conclusion that BP is sub-optimal and that searching for better algorithms could beneficial and in fact we will demonstrate such an algorithm in Section 4.1. This illustrates another aspect of the utility of our information theoretic bounds - they can be used as indicators of the sub-optimality of heuristic, computationally feasible or practically used algorithms.

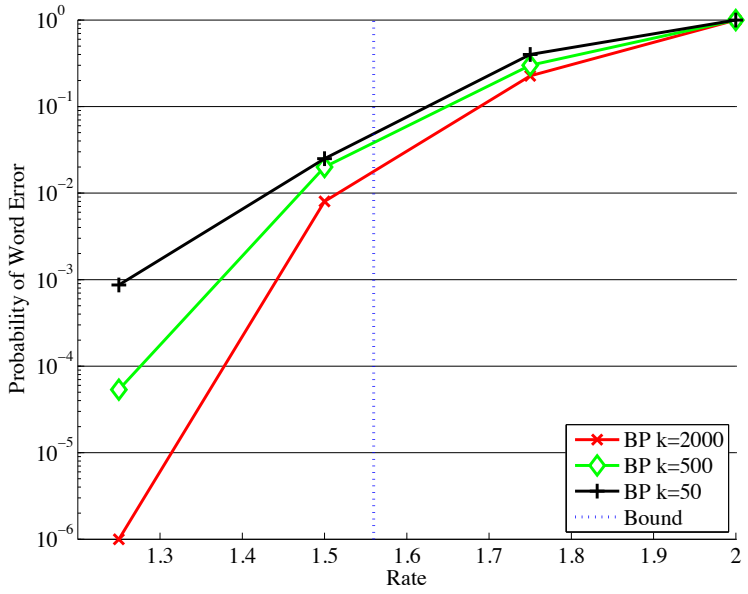


Figure B.1: Demonstration that the bound is a reasonable predictor of BP performance for small $c = 3$ at distortion $D = .002$ with $p = .1$ and 90 reads. The bottom of the plot indicates no errors in 10^6 runs.

We are especially interested in the computational aspects of algorithms and not just their accuracy. From (B.1), the computational complexity of the BP updates is exponential in the measurement size c. This is a problem when measurements with large c are used. The next section shows that for measurement functions that have a certain symmetry property the updates can be calculated *exactly* without requiring any approximations even for large values of c.

B.1 FASTER BP CALCULATIONS FOR TYPE BASED MEASUREMENTS

While loopy BP is much more efficient that the optimal Junction Tree algorithm its complexity is still exponential in the size of a measurement c which quickly becomes a problem as c grows.

A potential solution to this problem, pursued in recent work that explored the pooling

problem as a compressed sensing problem [70], is to approximate the computation in (B.1) with a Monte-Carlo sampling step. However, in this section we show that for some kinds of measurements (including those that arise in pooling) these updates can be computed exactly with a substantial speed-up which renders the Monte-Carlo sampling approximation unnecessary.

In order to achieve this speed-up we exploit a symmetry present in certain measurement functions, such as $\Psi =$ SUM, OR, MAX, THRESHOLD and many others, which are type-based measurements. A measurement function Ψ is type based if,

$$\Psi(Z) = \psi(t) \tag{B.3}$$

for some function ψ where t is the type, (empirical p.m.f.) of Z. These are Ψ functions for which the output only depends on the type of the input t (the number of 0s, 1s, and so on) and not on the actual ordering of the inputs. In our factor graphs this means that each factor $f(\mathbf{v}, y) = P_{Y|X}(y|\Phi(\mathbf{v})) = P_{Y|X}(y|\Psi(t)) = f(t, y)$ where t is they type of the particular input $\mathbf{v}$.

We first simplify (B.2) using this type -based symmetry,

$$\mu_{f_h \to V^t}(v) = \sum_{\mathbf{v}' \setminus v^{t'} : t' \in \mathcal{N}_h \setminus t} \left(f_h(\mathbf{v}', v) \prod_{v^{t'} : t' \in \mathcal{N}_h \setminus t} \mu_{V^{t'} \to f_h}(v^{t'}) \right) \tag{B.4}$$

$$= \sum_{t=0}^{(c-1)^{|\mathcal{X}|}} f(t, v) \prod_{v^{t'} : t' \in \mathcal{N}_h \setminus t, \mathbf{v} \text{ of type t}} \mu_{V^{t'} \to f_h}(v^{t'}) \tag{B.5}$$

So now we need to calculate $\prod_{v^{t'} : t' \in \mathcal{N}_h \setminus t, \mathbf{v} \text{ of type t}} \mu_{V^{t'} \to f_h}(v^{t'})$ for all possible values of the type t efficiently.

In order to exploit this symmetry we use the properties of generating functions [149]. Consider a set of polynomials in a formal variable A, with one equation for each of the c

neighbors of the measurement factor f^h.

$$P_{t'}(A) = \sum_{i=1}^{|\mathcal{V}|} \mu_{V^{t'} \to f_h}(v^{t'}) A^{i-1} \tag{B.6}$$

Now, each $P_{t'}(A)$ is a polynomial of degree $|\mathcal{V}| - 1$ and $\prod_{t' \in \mathcal{N}_h \setminus t} P_{t'}(A)$ is a polynomial of degree $c(|\mathcal{V}| - 1)$. Each term in this product polynomial corresponds to one type t, $\prod_{v_{t'}: t' \in \mathcal{N}(h) \setminus t, \mathbf{v} \text{ of type t}} \mu_{V_{t'} \to f^h}(v_{t'})$.

We now need to compute the formal polynomial $\prod_{t' \in \mathcal{N}_h \setminus t} P_{t'}(A)$.

The sequence of coefficients of this product polynomial can be calculated as the convolution of the coefficient sequences of the original polynomials $P_{t'}(A)$. This can be done efficiently using Fast Fourier Transforms. Thus, these messages can be computed much faster - a speed-up which becomes critical in many applications. For example, in our pooling example using the original update equations would only allow us to use c of about 5. The new updates would allow us to go up to $c = 40$ for binary measurements and $c = 20$ for SUM measurements on a desktop computer. This allows us to use the best possible c which can be large for very sparse priors p.

B.2 DENSITY EVOLUTION FOR MESSAGE PASSING ALGORITHMS

Recently a technique called Density Evolution (DE) was used to analyze the performance of BP for decoding LDPC codes [58]. Section 3.1.3 argued that sparse regular measurement structures are an appropriate model for some pooling applications. So, DE may be applicable to analyze the performance of BP in such measurement structures. However, DE was originally proposed for LDPC codes. This section outlines the differences between DE for sparse regular measurement systems and LDPC codes and compares the prediction of DE with simulations of BP for a pooling application.

We use the fact that, as shown in [146] the DE technique is also applicable for code

trees where each node satisfies a 'Perfect Projection' property. This property states that each variable node is drawn independently from a given fixed distribution. This is true in our sparse measurement systems since inputs are assumed to be drawn i.i.d from p_V. Thus, the predictions of DE will be good predictions of BP in the large system limit.

We now proceed to outline the density evolution equations for our sparse measurement systems. For simplicity we describe the case of binary inputs and SUM function. The same procedure can be directly used for larger input and output alphabets with a corresponding increase in computational complexity.

DE studies the distribution of the messages exchanged between nodes during message passing in the limiting case of infinitely long structures [150]. Since we are working with binary input alphabets ($|\mathcal{V}| = 2$) it is sufficient to track the distribution of two log-likelihood ratios (LLRs),

$$p^l_{LLR_0}(r) = Pr(\log(\frac{m^l_{f\to v}(0)}{m^l_{f\to v}(1)}) = r|v = 0) \tag{B.7}$$

$$p^l_{LLR_1}(r) = Pr(\log(\frac{m^l_{f\to v}(0)}{m^l_{f\to v}(1)}) = r|v = 1) \tag{B.8}$$

These LLRs conditioned on the variable *receiving* the message being 0 and 1 respectively. We need to track two LLRs because *we cannot assume w.l.o.g. that the true input is the all zeros*, which is a departure from the density evolution analysis of LDPC codes in [58].

We use DE to calculate the distributional fixed points of (B.7) and (B.8). We outline the DE technique. We start with the distribution over LLRs $p^l_{LLR_0}(r)$ and $p^l_{LLR_1}(r)$ based on the prior p_V. These form the initial distribution of messages from variables to factors. Using the update in (B.2) we calculate the new distribution of messages from factor to variable for each of 2^d possible configurations of true value variables connected to the factor. We average these to calculate the distribution of factor to variable messages. We repeat the process with the update in (B.1), resulting in new distributions over the LLRs $p^{l+1}_{LLR_0}(r)$ and

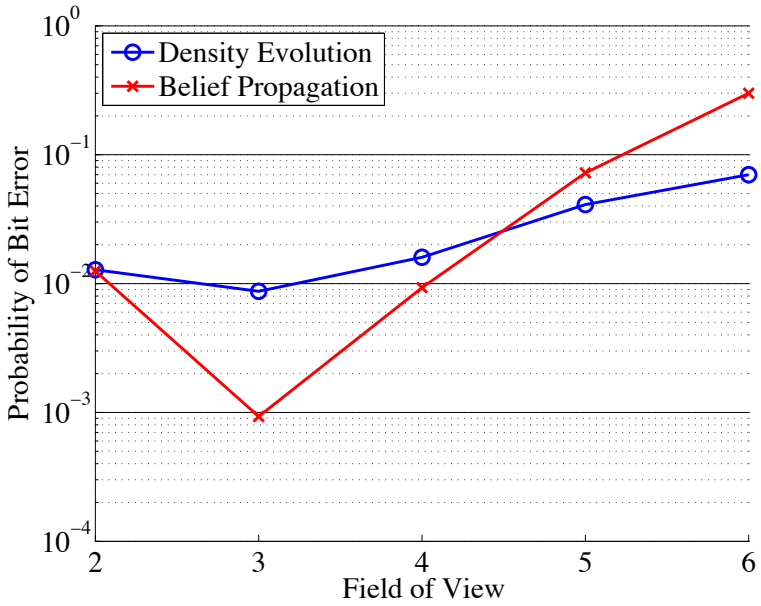


Figure B.2: A comparison of the performance of BP and the prediction made by DE for a range of c with SUM measurements, $R = 1$, $p = .05$, flip probability $pFlip = .05$ and input length $k = 200$

$p^{l+1}_{LLR_1}(r)$. Both these steps are repeated L times, where L is the number of BP iterations.

We simulate BP for SUM measurements, rate $R = 1$ and probability of error $pFlip = .05$ in Figure B.2. We compare the results to the result of DE. We see that DE is not a good indicator of the performance of BP in sparse measurement systems with small input length $k = 200$.

There are some interesting conclusions we could draw from Figure B.2 though they are not based on any rigorous analysis. For many different parameter settings - such as those in the Figure - we see that BP performs better than the DE prediction for small c and worse for higher c. It would thus appear that having loops in the graph helps the performance of BP for smaller c when most loops are not too short but hurts performance for larger c when many short loops occur for relatively small k. An analysis of this effect, perhaps using the loop calculus of [59] would be a very interesting direction of future research.

In general BP could be a very useful algorithm if the graph is tree like for large depths or if some sort of correlation decay property [151] holds. Finally, note that these results are for the average bit-error rate (since this is what DE analysis gives us) as opposed to the word error rate that we are usually interested in and hence are largely incomparable with our bounds.

In conclusion, our bound is a reasonable predictor of BP performance even at shorter lengths for word error rates, but the DE prediction of bit error rates is very far off - to the extent that it is possibly unusable for system design. We present another algorithm in Section 4.1, called the CE method, which has excellent agreement with our bounds and may be a better choice especially if the bounds are used to develop and understand practical systems.

We also mention here that BP implementation and DE analysis can be extended to the cases of mismatch discussed in previous chapters. However, since the base case results have so much discrepancy this does not seem to be a useful result to present.

REFERENCES

[1] R. M. Kainkaryam, "Pooling designs for high-throughput biological experiments," Ph.D. dissertation, The University of Michigan, 2010.

[2] J. M. F. Moura, R. Negi, and M. Pueschel, "Distributed sensing and processing: a graphical model approach," *DARPA ACMP Integrated Sensing and Processing Workshop,*, 2002.

[3] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, Feb 2001.

[4] D. Du and F. Hwang, *Combinatorial group testing and its applications*. World Scientific, 2000.

[5] R. Dorfman, "The detection of defective members of large populations," *The Annals of Mathematical Statistics*, vol. 14, no. 4, pp. pp. 436–440, 1943. [Online]. Available: http://www.jstor.org/stable/2235930

[6] M. Karpovsky, K. Chakrabarty, and L. Levitin, "On a new class of codes for identifying vertices in graphs," *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 599 –611, Mar. 1998.

[7] W. J. Bruno, E. Knill, D. J. Balding, D. C. Bruce, N. A. Doggett, W. W. Sawhill, R. L. Stallings, C. C. Whittaker, and D. C. Torney, "Efficient pooling designs for library screening," *Genomics*, vol. 26, no. 1, pp. 21 – 30, 1995.

[8] G. Cormode and S. Muthukrishnan, "What's hot and what's not: tracking most frequent items dynamically," *ACM Trans. Database Syst.*, vol. 30, pp. 249–278, March 2005. [Online]. Available: http://doi.acm.org/10.1145/1061318.1061325

[9] G. Atia, S. Aeron, E. Ermis, and V. Saligrama, "Cooperative sensing in cognitive radios," in *Allerton Conference on Communication, Control, and Computing*, 2007.

[10] P. L. Dragotti and M. Gastpar, *Distributed Source Coding: Theory, Algorithms and Applications*. Academic Press, 2009.

[11] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *Information Theory, IEEE Transactions on*, vol. 19, no. 4, pp. 471–480, Jul 1973.

[12] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *Information Theory, IEEE Transactions on*, vol. 22, no. 1, pp. 1–10, Jan 1976.

[13] A. Wyner, "Recent results in the shannon theory," *Information Theory, IEEE Transactions on*, vol. 20, no. 1, pp. 2–10, Jan 1974.

[14] R. G. Gallager, *Low Density Parity Check Codes*. M.I.T. Press, 1963.

[15] Y. Rachlin, N. Balakrishnan, R. Negi, J. Dolan, and P. Khosla, "Increasing sensor measurements to reduce detection complexity in large-scale detection applications," in *Military Communications Conference, 2006. MILCOM 2006. IEEE*, Oct. 2006, pp. 1–7.

[16] P. Abbeel, D. Koller, and A. Y. Ng, "Learning factor graphs in polynomial time and sample complexity," *Journal of Machine Learning Research*, vol. 7, pp. 1743–1788, 2006.

[17] A. Himoto, H. Aoyama, O. Fuchiwaki, D. Misaki, and T. Sumrall, "Development of micro rescue robot-human detection," in *IEEE International Conference on Mechatronics*, 2005, pp. 526 – 531.

[18] Q. Hao, D. Brady, B. Guenther, J. Burchett, M. Shankar, and S. Feller, "Human tracking with wireless distributed pyroelectric sensors," *Sensors Journal, IEEE*, vol. 6, no. 6, pp. 1683 –1696, 2006.

[19] E. J. Candès, J. K. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[20] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[21] B. Narayanaswamy, R. Negi, and P. Khosla, "Robust lossy detection using sparse measurements: The regular case," in *IEEE International Symposium on Information Theory*, 2010, pp. 1583 –1587.

[22] N. Santhanam and M. Wainwright, "Information-theoretic limits of graphical model selection in high dimensions," in *IEEE International Symposium on Information Theory*, 2008, pp. 2136 –2140.

[23] A. Krause and C. Guestrin, "Optimal value of information in graphical models," *J. Artif. Int. Res.*, vol. 35, pp. 557–591, July 2009. [Online]. Available: http://portal.acm.org/citation.cfm?id=1641503.1641516

[24] T. Dimitriou, "SAT distributions with planted assignments and phase transitions between decision and optimization problems," *Discrete Appl. Math.*, vol. 153, pp. 58–72, December 2005. [Online]. Available: http://dx.doi.org/10.1016/j.dam.2005.05.006

[25] T. Hogg, B. A. Huberman, and C. P. Williams, "Phase transitions and the search problem," *Artif. Intell.*, vol. 81, pp. 1–15, March 1996. [Online]. Available: http://portal.acm.org/citation.cfm?id=233339.233343

[26] Y. Rachlin, R. Negi, and P. K. Khosla, "The sensing capacity of sensor networks," *CoRR*, vol. abs/0901.2094, 2009.

[27] B. Narayanaswamy, R. Negi, and P. Khosla, "Robust detection of random variables using sparse measurements," in *Proceedings of the 47th annual Allerton conference on Communication, control, and computing*, ser. Allerton'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 1468–1475. [Online]. Available: http://portal.acm.org/citation.cfm?id=1793974.1794225

[28] ——, "An analysis of the computational complexity of sequential decoding of specific tree codes over gaussian channels," in *IEEE international Symposium on Information Theory*, 2008, pp. 2508 –2512.

[29] C. E. Shannon, "A mathematical theory of communication," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 5, no. 1, pp. 3–55, January 2001. [Online]. Available: http://dx.doi.org/10.1145/584091.584093

[30] ——, "Coding theorems for a discrete source with a fidelity criterion," in *IRE Nat. Conv. Rec., Pt. 4*, 1959, pp. 142–163.

[31] Y. Rachlin, R. Negi, and P. Khosla, "Sensing capacity for discrete sensor network applications," in *Fourth International Symposium on Information Processing in Sensor Networks*, April 2005, pp. 126–132.

[32] Y. Rachlin, "On the interdependence of sensing, accuracy, and complexity in large-scale detection applications," Ph.D. dissertation, Pittsburgh, PA, USA, 2007, aAI3262829.

[33] A. Montanari and E. Mossel, "Smooth compression, gallager bound and nonlinear sparse-graph codes," in *IEEE International Symposium on Information Theory*, July 2008, pp. 2474–2478.

[34] A. Dembo and O. Zeitouni, *Large Deviations Techniques and Applications*. Jones and Bartlett, Boston, 1993.

[35] N. Alon and J. Spencer, *The Probabilistic Method*. John Wiley, 1992.

[36] T. Berger and V. Levenshtein, "Asymptotic efficiency of two-stage disjunctive testing," *IEEE Transactions on Information Theory,*, vol. 48, no. 7, pp. 1741 –1749, July 2002.

[37] G. Atia and V. Saligrama, "Noisy group testing: An information theoretic perspective," in *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*, 30 2009.

[38] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 399–431, 1999.

[39] R. Kainkaryam, A. Bruex, A. Gilbert, J. Schiefelbein, and P. Woolf, "poolMC: Smart pooling of mRNA samples in microarray experiments," *BMC Bioinformatics*, vol. 11, no. 1, p. 299, 2010. [Online]. Available: http://www.biomedcentral.com/1471-2105/11/299

[40] H. Q. Ngo and D. zhu Du, "A survey on combinatorial group testing algorithms with applications to dna library screening," 2000.

[41] M. Cheraghchi, "Derandomization and group testing," *CoRR*, vol. abs/1010.0433, 2010.

[42] M. Akcakaya and V. Tarokh, "Shannon-theoretic limits on noisy compressive sampling," *IEEE Transactions on Information Theory*, vol. 56, no. 1, pp. 492 –504, 2010.

[43] M. J. Wainwright, "Information-theoretic limits on sparsity recovery in the high-dimensional and noisy setting," *IEEE Trans. Inf. Theor.*, vol. 55, pp. 5728–5741, December 2009. [Online]. Available: http://dx.doi.org/10.1109/TIT.2009.2032816

[44] W. Wang, M. J. Wainwright, and K. Ramchandran, "Information-theoretic limits on sparse signal recovery: dense versus sparse measurement matrices," *IEEE Trans. Inf. Theor.*, vol. 56, pp. 2967–2979, June 2010. [Online]. Available: http://dx.doi.org/10.1109/TIT.2010.2046199

[45] A. Gilbert and P. Indyk, "Sparse recovery using sparse matrices," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 937 –947, 2010.

[46] I. Sasaki, M. Josowicz, J. Janata, and A. Glezer, "Enhanced performance of a filter-sensor system," *Analyst*, vol. 131, pp. 751–756, 2006.

[47] V. Bychkovskiy, S. Megerian, D. Estrin, and M. Potkonjak, "A collaborative approach to in-place sensor calibration," in *IPSN*, 2003, pp. 301–316.

[48] J. W. Gardner and P. N. Bartlett, "Electronic Noses. Principles and Applications," *Measurement Science and Technology*, vol. 11, no. 7, p. 1087, 2000. [Online]. Available: http://stacks.iop.org/0957-0233/11/i=7/a=702

[49] M. A. Sheikh, S. Sarvotham, O. Milenkovic, and R. G. Baraniuk, "DNA array decoding from nonlinear measurements by belief propagation," in *Proceedings of the 2007 IEEE/SP 14th Workshop on Statistical Signal Processing*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 215–219. [Online]. Available: http://portal.acm.org/citation.cfm?id=1524876.1525173

[50] A. Lapidoth and P. Narayan, "Reliable communication under channel uncertainty," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2148–2177, Oct 1998.

[51] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[52] G. F. Cooper, "The computational complexity of probabilistic inference using bayesian belief networks," *Artificial Intelligence*, vol. 42, no. 2-3, pp. 393 – 405, 1990. [Online]. Available: http://www.sciencedirect.com/science/article/B6TYF-47X29XG-3H/2/f8a895d2ec1685901f93b201c47348e2

[53] B. O. Lucena, "Dynamic programming, tree-width, and computation on graphical models," Brown University, Tech. Rep., 2002.

[54] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1988.

[55] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100 –107, 1968.

[56] M. I. Jordan and Y. Weiss, "Graphical models: Probabilistic inference," in *Handbook of Neural Networks and Brain Theory*, 2nd ed., M. Arbib, Ed. Cambridge, MA, USA: MIT Press, 2002.

[57] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Found. Trends Mach. Learn.*, vol. 1, pp. 1–305, January 2008. [Online]. Available: http://portal.acm.org/citation.cfm?id=1498840.1498841

[58] T. Richardson and R. 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.

[59] M. Chertkov and V. Y. Chernyak, "Loop series for discrete statistical models on graphs," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2006, no. 06, p. P06009, 2006. [Online]. Available: http://stacks.iop.org/1742-5468/2006/i=06/a=P06009

[60] F. Jelinek, "Fast sequential decoding algorithm using a stack," *IBM Journal of Research and Development*, vol. 13, no. 6, p. 675, 1969.

[61] Y. Rachlin, R. Negi, and P. Khosla, "On the interdependence of sensing and estimation complexity in sensor networks," in *Proceedings of the 5th international conference on Information processing in sensor networks*, ser. IPSN '06. New York, NY, USA: ACM, 2006, pp. 160–167.

[62] J. Savage, "Sequential decoding-the computation problem," *Bell Syst. Tech. J*, 1966.

[63] R. Y. Rubinstein and D. P. Kroese, *The Cross Entropy Method: A Unified Approach To Combinatorial Optimization, Monte-carlo Simulation (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2004.

[64] S. Verdu, *Multiuser Detection*. Cambridge University press, 1998.

[65] Y. Rachlin, R. Negi, and P. Khosla, "Sensing capacity for target detection," in *IEEE ITW*, Oct. 2004, pp. 147–152.

[66] D. Guo and C.-C. Wang, "Multiuser detection of sparsely spread CDMA," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 3, pp. 421–431, April 2008.

[67] A. Montanari, "Estimating random variables from random sparse observations," *European Transactions on Telecommunications*, vol. 19, no. 4, pp. 385–403, 2008.

[68] D. J. Westreich, M. G. Hudgens, S. A. Fiscus, and C. D. Pilcher, "Optimizing Screening for Acute Human Immunodeficiency Virus Infection with Pooled Nucleic Acid Amplification Tests ," *J. Clin. Microbiol.*, vol. 46, no. 5, pp. 1785–1792, 2008. [Online]. Available: http://jcm.asm.org/cgi/content/abstract/46/5/1785

[69] N. Thierry-Mieg, "Pooling in systems biology becomes smart," *Nat Meth*, vol. 3, pp. 161–162, Mar.

[70] Y. Erlich, A. Gordon, M. Brand, G. Hannon, and P. Mitra, "Compressed genotyping," *IEEE Transactions on Information Theory*, vol. 56, no. 2, pp. 706 –723, feb. 2010.

[71] W. Kautz and R. Singleton, "Nonrandom binary superimposed codes," *IEEE Transactions on Information Theory*, vol. 10, no. 4, pp. 363 – 377, Oct. 1964.

[72] R. P. Hertzberg and A. J. Pope, "High-throughput screening: new technology for the 21st century," *Current Opinion in Chemical Biology*, vol. 4, no. 4, pp. 445 – 451, 2000.

[73] A. Dove, "Drug screening-beyond the bottleneck," *Nature Biotechnology*, vol. 17, no. 9, pp. 859–863, 1999.

[74] A. C. Gilbert, S. Guha, P. Indyk, Y. Kotidis, S. Muthukrishnan, and M. J. Strauss, "Fast, small-space algorithms for approximate histogram maintenance," in *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, ser. STOC '02. New York, NY, USA: ACM, 2002, pp. 389–398.

[75] A. Gilbert, M. Iwen, and M. Strauss, "Group testing and sparse signal recovery," in *42nd Asilomar Conference on Signals, Systems and Computers*, 2008, pp. 1059 –1063.

[76] S. Wu and B. Liu, "Application of scintillation proximity assay in drug discovery," in *BioDrugs*, vol. 10, no. 19, 2005, p. 383392.

[77] L. Hartwell, L. Hood, M. Goldberg, A. Reynolds, L. Silver, and R. Veres, *Genetics: From Genes to Genomes 3/e*. McGraw-Hill, 2006.

[78] G. Rosner, S. Rosner, and A. Orr-Urtreger, "Genetic Testing in Israel: An Overview," *Annual Review of Genomics and Human Genetics*, vol. 10, no. 1, pp. 175–192, 2009, pMID: 19453249.

[79] J. Ekstein and H. Katzenstein, "The Dor Yeshorim story: community-based carrier screening for Tay-Sachs disease," *Adv. Genet.*, vol. 44, pp. 297–310, 2001.

[80] S. Prabhu and I. Pe'er, "Overlapping pools for high-throughput targeted resequencing." *Genome research*, vol. 19, no. 7, pp. 1254–1261, July 2009.

[81] Y. Erlich, N. Shental, A. Amir, and O. Zuk, "Compressed sensing approach for high throughput carrier screen," in *2009 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, September 2009, pp. 539–544.

[82] E. R. Mardis, "Next-Generation DNA Sequencing Methods," *Annual Review of Genomics and Human Genetics*, vol. 9, no. 1, pp. 387–402, 2008.

[83] Y. Rachlin, R. Negi, and P. Khosla, "Temporal sensing capacity," in *Proc. Allerton Conference on Communication, Control, and Computing*, 2006.

[84] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *IEEE International Conference on Robotics and Automation*, vol. 2, Mar. 1985, pp. 116 – 121.

[85] R. Murphy, "Trial by fire," *IEEE Robotics Automation Magazine*, vol. 11, no. 3, pp. 50 – 61, 2004.

[86] S. Thrun, "Robotic mapping: a survey," pp. 1–35, 2003.

[87] M. Paskin, C. Guestrin, and J. McFadden, "A robust architecture for distributed inference in sensor networks," in *Proceedings of the 4th international symposium on Information processing in sensor networks*, ser. IPSN '05. Piscataway, NJ, USA: IEEE Press, 2005. [Online]. Available: http://portal.acm.org/citation.cfm?id=1147685.1147697

[88] M. Luby, "LT codes," in *The 43rd Annual IEEE Symposium on Foundations of Computer Science*, 2002, pp. 271–280. [Online]. Available: http://dx.doi.org/10.1109/SFCS.2002.1181950

[89] W. Wang, M. Garofalakis, and K. Ramchandran, "Distributed sparse random projections for refinable approximation," in *IPSN '07: Proceedings of the 6th international conference on Information processing in sensor networks*. New York, NY, USA: ACM, 2007, pp. 331–339.

[90] P. Li, T. J. Hastie, and K. W. Church, "Very sparse random projections," in *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2006, pp. 287–296.

[91] E. Ising, "Beitrag zur theorie des ferromagnetismus," *Zeitschrift für Physik A Hadrons and Nuclei*, vol. 31, pp. 253–258, 1925, 10.1007/BF02980577.

[92] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, March 2008.

[93] D. Aldous and J. M. Steele, "The objective method: Probabilistic combinatorial optimization and local weak convergence," in *Discrete and Combinatorial Probability*. Springer, 2003, pp. 1–72.

[94] L. B. Buck, "Nobel lecture," http://nobelprize.org/nobel_prizes/medicine/laureates/2004/buck-lecture.html, December 2004.

[95] S. Janson, D. E. Knuth, T. Łuczak, and B. Pittel, "The birth of the giant component," *ArXiv Mathematics e-prints*, Sept. 1993.

[96] M. Newman, A. L. Barabsi, and D. J. Watts, *The Structure and Dynamics of Networks*. Princeton University Press, 2006.

[97] R. G. Gallager, *Information Theory and Reliable Communication*. New York, NY, USA: John Wiley & Sons, Inc., 1968.

[98] I. Csiszar, "The method of types," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2505–2523, Oct 1998.

[99] I. Sason and S. Shamai, "Performance analysis of linear codes under maximum-likelihood decoding: A tutorial," *Foundations and Trends in Communications and Information Theory*, vol. 3, no. 1/2, 2006.

[100] J. Shendure, G. J. Porreca, N. B. Reppas, X. Lin, J. P. McCutcheon, A. M. Rosenbaum, M. D. Wang, K. Zhang, R. D. Mitra, and G. M. Church, "Accurate multiplex polony sequencing of an evolved bacterial genome," *Science*, vol. 309, pp. 1728–1732, Sep 2005.

[101] T. M. Cover and J. A. Thomas, *Elements of information theory*. New York, NY, USA: Wiley-Interscience, 1991.

[102] R. Y. Rubinstein, "Optimization of computer simulation models with rare events," *European Journal of Operations Research*, vol. 99, pp. 89–112, 1996.

[103] B. Narayanaswamy, Y. Rachlin, R. Negi, and P. Khosla, "The sequential decoding metric for detection in sensor networks," in *IEEE International Symposium on Information Theory*, 2007, pp. 1611 –1615.

[104] R. M. Neal, "Probabilistic inference using markov chain monte carlo methods," 1993.

[105] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, pp. 1087–1092, 1953.

[106] R. Y. Rubinstein and D. P. Kroese, *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning (Information Science and Statistics)*. Springer, 2004. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/038721240X

[107] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," pp. 564–584, 1987.

[108] A. E. Gelfand and A. F. M. Smith, "Sampling-based approaches to calculating marginal densities," *Journal of the American Statistical Association*, vol. 85, no. 410, pp. 398–409, 1990. [Online]. Available: http://dx.doi.org/10.2307/2289776

[109] D. Li, K. Wong, Y. H. Hu, and A. Sayeed, "Detection, classification, and tracking of targets," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 17 –29, Mar. 2002.

[110] R. Johannesson and K. S. Zigangirov, *Fundamentals of Convolutional Coding*. Wiley-IEEE Press, 1999.

[111] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y. Choi, T. Herman, S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita, "A line in the sand: a wireless sensor network for target detection, classification, and tracking," *Comput. Netw.*, vol. 46, no. 5, pp. 605–634, 2004.

[112] J. Wozencraft and B. Reiffen, *Sequential Decoding*. Cambridge, MA, USA: MIT Press, 1961.

[113] R. Fano, "A heuristic discussion of probabilistic decoding," *IEEE Transactions on Information Theory*, vol. 9, no. 2, pp. 64–74, Apr 1963.

[114] F. Xiong, "Sequential decoding of convolutional codes in channels with intersymbol interference," *IEEE Transactions on Communications*, vol. 43, no. 234, pp. 828 – 836, 1995.

[115] B. Narayanaswamy, R. Negi, and P. Khosla, "Preprcessing measurements and modifying sensors to improve detection in sensor networks"," in *Proceedings of the Sensor, Signal and Information Processing (SenSIP) Workshop*, Sedona, AZ, May 2008.

[116] V. Sorokine and F. Kschischang, "A sequential decoder for linear block codes with a variable bias-term metric," *IEEE Transactions on Information Theory*, vol. 44, no. 1, pp. 410 –416, Jan. 1998.

[117] E. Arikan, "An inequality on guessing and its application to sequential decoding," *IEEE Transactions on Information Theory*, vol. 42, no. 1, pp. 99–105, 1996.

[118] O. Etesami, "Relations between belief propagation on erasure and symmetric channels," in *IEEE International Symposium on Information Theory*, 2004.

[119] K. P. Murphy, "The Bayes Net Toolbox for MATLAB," *Computing Science and Statistics*, vol. 33, p. 2001, 2001.

[120] V. Tarokh, A. Vardy, and K. Zeger, "Sequential decoding of lattice codes," in *IEEE International Symposium on Information Theory*, 1997.

[121] A. D. Murugan, H. E. Gamal, M. O. Damen, and G. Caire, "A unified framework for tree search decoding: rediscovering the sequential decoder," *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 933–953, 2006.

[122] J. Massey and J. Costello, D., "Nonsystematic convolutional codes for sequential decoding in space applications," *IEEE Transactions on Communication Technology*, vol. 19, no. 5, pp. 806 –813, 1971.

[123] F.-Q. Wang and J. Costello, D.J., "Robustly good trellis codes," *IEEE Transactions on Communications*, vol. 44, no. 7, pp. 791 –798, July 1996.

[124] P. Chevillat and J. Costello, D., "An analysis of sequential decoding for specific time-invariant convolutional codes," *IEEE Transactions on Information Theory*, vol. 24, no. 4, pp. 443–451, Jul 1978.

[125] G. D. Forney, Jr., "Convolutional codes III. Sequential decoding," *Information and Control*, vol. 25, no. 3, pp. 267–297, July 1974.

[126] J. Massey, "Variable-length codes and the Fano metric," *IEEE Transactions on Information Theory*, vol. 18, no. 1, pp. 196 – 198, Jan. 1972.

[127] A. Weathers, "An analysis of the truncated Viterbi algorithm for PRML channels," *IEEE International Conference on Communications*, vol. 3, pp. 1951–1956 vol.3, 1999.

[128] M. A. Sheikh, S. Sarvotham, O. Milenkovic, and R. G. Baraniuk, "DNA array decoding from nonlinear measurements by belief propagation," in *IEEE/SP 14th Workshop on Statistical Signal Processing*, Aug. 2007, pp. 215–219.

[129] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. New York, NY, USA: ACM, 2002, pp. 88–97.

[130] G. van der Horn and J. H. Huijsing, "Integrated Smart Sensor Calibration," *Analog Integr. Circuits Signal Process.*, vol. 14, no. 3, pp. 207–222, 1997.

[131] J. Feng, S. Megerian, and M. Potkonjak, "Model-based calibration for sensor networks," *Proceedings of IEEE Sensors*, vol. 2, pp. 737–742 Vol.2, Oct. 2003.

[132] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, vol. 39, no. 1, pp. 1–38, 1977.

[133] M. Feder and E. Weinstein, "Parameter estimation of superimposed signals using the EM algorithm," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36, no. 4, pp. 477 –489, Apr. 1988.

[134] A. Eckford, "The factor graph EM algorithm: applications for LDPC codes," in *IEEE 6th Workshop on Signal Processing Advances in Wireless Communications*, 2005, pp. 910 – 914.

[135] N. Thierry-Mieg, "A new pooling strategy for high-throughput screening: the Shifted Transversal Design," *BMC Bioinformatics*, vol. 7, no. 1, p. 28, 2006. [Online]. Available: http://www.biomedcentral.com/1471-2105/7/28

[136] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[137] C. Di, D. Proietti, I. Telatar, T. Richardson, and R. Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel," *Information Theory, IEEE Transactions on*, vol. 48, no. 6, pp. 1570–1579, Jun 2002.

[138] N. Kashyap and A. Vardy, "Stopping sets in codes from designs," in *IEEE International Symposium on Information Theory*, 2003.

[139] R. A. DeVore, "Deterministic constructions of compressed sensing matrices," *J. Complex.*, vol. 23, pp. 918–925, August 2007. [Online]. Available: http://portal.acm.org/citation.cfm?id=1322590.1323073

[140] M. Mezard and C. Toninelli, "Group Testing with Random Pools: optimal two-stage algorithms," *ArXiv e-prints*, June 2007.

[141] F. K. Hwang, "Random k-set pool designs with distinct columns," *Probab. Eng. Inf. Sci.*, vol. 14, no. 1, pp. 49–56, 2000.

[142] S. Aji and R. McEliece, "The generalized distributive law," *Information Theory, IEEE Transactions on*, vol. 46, no. 2, pp. 325 –343, mar. 2000.

[143] U. Kjrulff, "Triangulation of graphs – algorithms giving small total state space," Tech. Rep., 1990.

[144] G. R. Shafer and P. P. Shenoy, "Probability propagation," *Annals of Mathematics and Artificial Intelligence*, vol. 2, pp. 327–351, 1990, 10.1007/BF01531015. [Online]. Available: http://dx.doi.org/10.1007/BF01531015

[145] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (corresp.)," *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284 – 287, mar. 1974.

[146] C.-C. Wang, S. R. Kulkarni, and H. V. Poor, "Density evolution for asymmetric memoryless channels," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4216–4236, 2005.

[147] D. J. C. MacKay, *Information Theory, Inference & Learning Algorithms*. New York, NY, USA: Cambridge University Press, 2002.

[148] K. P. Murphy, Y. Weiss, and M. I. Jordan, "Loopy belief propagation for approximate inference: An empirical study," in *UAI*, 1999, pp. 467–475.

[149] H. S. Wilf, *generatingfunctionology*, 2nd ed. Academic Press, Inc., 1994.

[150] T. Richardson and R. Urbanke, "Fixed points and stability of density evolution." *Commun. Inf. Syst.*, vol. 4, no. 1, pp. 103–116, 2004.

[151] T. Weber, "Correlation decay and decentralized optimization in graphical models," Ph.D. dissertation, MIT Sloan School of Management, Cambridge, Massachusetts, USA, 2010.