Graceful codes: fundamental limits and constructions

by

Hajir Roozbehani

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2019

© Massachusetts Institute of Technology 2019. All rights reserved.

Author .....
Department of Aeronautics and Astronautics
21 June, 2019

Certified by .....
Yury Polyanskiy
Associate Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Certified by .....
Pablo Parrilo
Professor of Electrical Engineering and Computer Science
Thesis Committee Member

Certified by .....
Sertac Karaman
Associate Professor of Aeronautics and Astronautics
Thesis Committee Member

Accepted by .....
Sertac Karaman
Associate Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Graceful codes: fundamental limits and constructions

by

Hajir Roozbehani

Submitted to the Department of Aeronautics and Astronautics
on 21 June, 2019, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Aeronautics and Astronautics

Abstract

A central question in information theory is to understand when and how data can be reconstructed from noisy observations. Error correcting codes are means of adding redundancy to the data to enable better recovery. Most commonly, codes are designed to recover data in a regime where the statistics of the noise are kept constant. In a number of applications, however, it is required that the quality of the reconstruction degrade gracefully as noise statistics worsen. It was known since the early work of Jacob Ziv (among others) that trade-offs between gracefullness and error correcting capability exist. We focus on characterizing these trade-offs and proposing codes that are closer to optimal than those employed today.

The information-theoretic contributions consist of three parts: combinatorial – where we study the so called alpha-beta profile of codes over large alphabets; geometric – where we show that a linear code that spreads out nearby data vectors must contract some far away data vectors as well; and probabilistic – where we show that good linear codes must necessarily experience threshold effect, i.e. degrade their performance sharply when the noise level exceeds a certain limit.

Our main coding-theoretic contribution is the introduction of a new class of non-linear sparse-graph codes that we call Low-Density Majority Codes (LDMCs). They admit efficient decoding via belief propagation and have provably superior performance compared to the best-possible linear systematic codes, in particular LDGMs. Hence, we hope that LDMCs will be able to replace LDGMs in practical applications, such as pre-coding for optical channels, tornado-raptor codes, and protograph constructions.

Thesis Supervisor: Yury Polyanskiy
Title: Associate Professor of Electrical Engineering and Computer Science

Acknowledgments

> A dream you dream alone is only a dream. A dream you dream together is reality.
>
> Yoko Ono

When I first arrived at MIT, I was rather unaware of mathematics and the beauties it has to offer. I think of this thesis as a journey of transformation from the state of being unaware to what Dan Harris might call "ten percent less unaware". Along the way, I relied on the advice, support, and kindness of many without whom my journey would have all but ended prematurely.

First and foremost, I would like to thank my advisor Yury Polyanskiy for giving me the opportunity to work on this thesis under his guidance. It is thanks to your patience and support that I have come as far as I have. You were always there to share your enthusiasm and knowledge with me; you taught me that mathematics is beautiful; you taught me to explore with no fear; and above all, you remained an ally in the hardest of times. I am grateful for the countless hours you spent to read all the proofs in this thesis, and for always finding a way to help me improve them.

I would like to thank the members of my thesis committee Pablo Parrilo and Sertac Karaman for providing valuable feedback, and pointing out many interesting connections to other research problems. I would like to thank Pablo also for his teachings. Your class on Algebraic Techniques and Semidefinite Programming was one of the first and best classes that I took at MIT. I shall always look fondly on that experience for it invoked a deep sense of appreciation for algebraic abstraction and geometric intuition. Sertac, I am grateful to you for always voicing your support, be it in the corridors of LIDS or in official meetings.

I thank the thesis readers Ziv Goldfeld and Bobak Nazer for making time to read and comment on this work. I thank Caroline Uhler for her comments on my thesis proposal and for many memorable conversations around our interests in algebraic statistics. I thank Arya Mazumdar and Ankit Rawat for our collaboration on CJSCC. I thank Frank Kschischang for interesting discussions, for his words of encouragement,

and for generously sharing early drafts of their manuscript on work related to this dissertation.

I would like to thank all of my math teachers at MIT. In particular, I greatly enjoyed the classes that I took with Paul Seidel, Richard Melrose, and Bjorn Poonen. I thank Eric Larson for always making time to educate me about algebraic geometry and intersection theory.

I am grateful to all of the LIDS faculty for their teachings and their mentorship. In particular, I would like to say thanks to Emilio Frazolli who was my first advisor at MIT. Thank you for accepting me as your student and for supporting me as I transitioned into other areas of research. I am grateful to Sasha Megretski for always welcoming me to his office. I thank John Tsitsiklis whose wonderful probability class was among my best learning experiences at MIT. I also thank him for giving me the opportunity to work as his teaching assistant. I thank Guy Bressler for giving me the opportunity to teach his recitations and for co-organizing our summer reading groups. I thank Sanjoy Mitter, Asu Ozdagler, Sasha Rakhlin, Suvrit Sra, and Devavrat Shah from whom I have learned inside and outside classrooms. I thank Munther Dahleh for generously offering his help, advice, and support at the crossroads. And thank you to Mardavij for showing me the way!

I would like to thank the LIDS administrative staff Rachel Cohen, Lynne Dell, Jennifer Donovan, Francisco Jaimes, Brian Jones, Gracie Gao, Rich Lay, and Debbie Wright, for all the help they have provided me. I am grateful to Beth Marois from the department of Aeronautics and Astronautics for her constant support over the years.

I thank my friends and colleagues Ganesh Ajjanagadde, Jason Altschuler, Ammar Ammar, Luis Castro, Pratick Chaudhari, Ali Faqigh, Bomin Jiang, Igor Kadota, Mina Karzand, Yola Katsargyri, Eren Kizildag, Raeez Lorgat, Ali Makhdoumi, Flora Meng, David Miculescu, Noel Morris, Omer Tanovic, and Mark Tobenkein for making MIT a better place for me. I thank Amir Mehrabian, Ehsan Naseri, and Hoda Sadeghian for their friendship of many years. I salute Lt. Col. Philip Root and thank him for the good times we shared in office. I thank the outstanding ranks of the coffee

hour crew Elaheh Fata, Tuhin Sarkar, Ian Schneider, and Igor Spasojevic for their camaraderie.

I thank past and current members of our research group: Flavio Calmon, Austin Collins, Ziv Goldfeld, Yuzhou Gu, Suhas Kowshik, Anuran Makur, Or Ordentlich, and Jennifer Tang. I have learned from all of you, be it in a group meeting, seminar, or a passing conversation. In particular, Anuran's work on channel comparisons has inspired some sections of this dissertation.

I once "dreamed a dream" to come to MIT for graduate school. I was fortunate to live my dream, and even more so to have had people who dreamed my dream with me. Thank you to Mom and Dad! This day would never come if it were not for you and because of you. Thank you to Mardavij, Mitra, and Mahnaz for always being there for me. I cannot imagine what my life in Boston would have look liked without you; fortunately, I never had to! Thank you to Mojgan for being the sister that you are. Thank you to Marjan, Abolfazl, and my favorite pianists Roozbeh and Ronak for giving me the best graduation gift I could ask for! Lastly, I would like to say thanks to Setareh for many precious years of support and patience. We started this journey together and no journey has ever been completed without ever being started. Thank you for the send-off! I know it was not easy.

Contents

Chapter 1

Introduction

This thesis is a study of *graceful degradation* in the context of partial data recovery. We start with a basic estimation problem in the presence of missing data:

> *Given a code $f : x \mapsto y$, recover x after observing (possibly some random subset of) the coded data $y = f(x)$.*

Roughly speaking, anytime that the amount of available data falls short of the *information theoretic* requirements of full recovery, we have a partial data recovery problem in hand. Information theoretic requirements for full recovery are easy to describe here: the number of observations must be at least equal to the number of unknowns. Once this requirement is met, there is hope to solve the following *coding theoretic* problem: design good maps f from which any unknown vector can be recovered. If however the information theoretic requirement is not met, there exist some vectors for any design that cannot be fully recovered. Let $\hat{x}$ be an estimate for the unknown x. To measure the quality of the estimate, we need a notion of similarity, i.e., a metric. In this thesis, we work exclusively with the Hamming distance

$$d_H(x, \hat{x}) = \sum_i \mathbb{1}\{x_i \neq \hat{x}_i\},$$

which simply counts the number of coordinates in which x and $\hat{x}$ differ. By a graceful code, we mean a mapping f for which the quality of estimation varies smoothly with the amount of available data from $y = f(x)$.

To construct good codes, we need to make some assumption on how the information is generated and how the coded data is erased. Since the seminal works of Claude Shannon [70] and Richard Hamming [28], the following models are now standard:

- *Shannon's stochastic model*: Assumes that the information $X_1, X_2, \cdots$ is a random sequence (typically i.i.d uniform over some alphabet) and each coded data $Y_i = f_i(X)$ is dropped randomly (and independently) with probability ϵ.

- *Hamming's combinatorial model*: Assumes that the information is an arbitrary vector x over a fixed base field. It bounds the number of missing coded bits in the data $y = f(x)$ but otherwise assumes that they are chosen adversarially.

Most commonly, codes are designed to operate efficiently in the regime of full data recovery. This often leads to some restrictive conditions. In a communication problem, this typically means that delay is not important, or somewhat equivalently, that the noise statistics are known. In many modern applications these assumption are not met. One example is that of short-packet communication [20]. In this regime, the concentration laws of probability are not fully in play yet and the channel cannot be treated as a stable medium that on average behaves in a predictable way. The uncertainty in the medium can be significant enough that, for all practical purposes, we may assume to be working with a family of channels. Hence to achieve good finite length performance, we need graceful codes that can adapt to the medium as it shifts from one channel to the next. Another setting where communication delay is critical is that of control over a noisy communication channel [76, 39, 66, 50]. Control systems are generally sensitive to delays in the feedback loop. Typically, delays in the feedback control signal are more destabilizing than noise or other forms of disturbance. Hence, it is often preferred to have partial feedback in real time than to have perfect delayed feedback. It is thus important to design coding schemes that estimate the state in real time and progressively improve when more gracedata is made available. Graceful codes are also useful in the regime of full data recovery (delay issues notwithstanding), since they can be used as an inner code in layered designs (or concatenated codes). For instance a design with two layers typically uses an outer error correcting code

(BCH, Hamming, etc) and an inner error reducing code. Ideally, the inner code must be graceful since it is to operate in the regime of partial data recovery. It only produces an estimate of the source with some distortion that is within the error correcting capability of the outer code. Even in the full recovery regime, such designs are becoming increasingly popular. Tornado-raptor codes and its many extensions are among such designs [8, 26, 4, 9, 45, 15, 14]. This is mainly due to the fact that achieving small error under iterative decoding with a single layer design is difficult. For instance, the standards in optical communication require an error rate of 10^{-15}, much lower than what a state-of-the-art low density parity check (LDPC) code can achieve. It has been observed that two layer deigns can achieve the required error rate and that significant savings in complexity and power is obtained when the inner code of the design is a graceful low density generator matrix (LDGM) code [72, 81, 5]. We have such applications in mind when we speak of graceful degradation.

This brings us to the following loose definition of a good graceful code: it is a code that has good error correction capability and can smoothly adapt to variations in its medium. Once we make this notion more precise, we set out to address the same two fundamental problems discussed earlier: 1) to determine the information theoretic limits of graceful codes, i.e., study the trade-offs between error correction and gracefulness, and 2) to solve the corresponding coding theoretic problem of constructing them. Evidently, designing practical codes that can operate closer to the fundamental trade-off than those employed in modern practice (e.g. LDGMs), can impact all or some of the applications mentioned above. We first give an informal overview of how this thesis plans to undertake this effort and outline the organization of the thesis. We then dedicate the next few sections of this chapter to explain our technical results and their connection with prior work more formally.

The trade-offs between gracefulness and error correction capability were known to exists for analog systems since the early work of Jacob Ziv [82]. However, determining the exact trade-offs even for the simple problem discussed at the beginning of this chapter is an unsolved problem both in the Hamming and Shannon senses of the definition [36, 74, 32, 60, 52, 33]. Our main information theoretic contribution is to

study what these trade-offs are for linear codes. Our results in essence show that non-linear codes are superior to linear codes from the perspective of graceful degradation. This is done by

- establishing the information theoretic trade-offs for graceful linear codes, and
- constructing practical graceful non-linear codes that surpass these limitations.

The organization of the thesis is as follows. We first formalize the graceful degradation problem both for Hamming's and Shannon's models in the next few sections and introduce our main coding theoretic contribution: a new family of sparse graph codes called Low Density Majority Codes (LDMCs). After reviewing our technical results we move on to Part I of the thesis, where we focus on the graceful degradation problem under Shannon's stochastic model. In Chapter 2, we characterize the trade-offs between error correction capability and gracefulness for linear codes. These results can be shown to improve significantly on the best known general converses [36, 32] in the case of linear codes. Such trade-offs are naturally related to the area theorem of coding [53]. In Chapter 3, we study the implications of the area theorem and show that it is not strong enough to fully characterize these trade-offs for linear codes. In Chapter 4 we provide new tools to analyze the dynamics of belief propagation (BP) for general non-linear codes using various notions of channel comparison in information theory. When applied to certain special cases of LDMCs the results accurately predict the performance. It follows from this analysis and the trade-offs of Chapter 2 that LDMCs have provably superior performance to the best possible linear systematic codes. We study the applications of LDMCs to code optimization in Chapter 5, where we show that by replacing the degree 1 nodes in LDGMs, the performance (as well as the rate of convergence) can be uniformly improved for all noise levels. In Chapter 6, we study soft-decoding properties of LDMCs when used in a concatenated design.

In Part II of the thesis we shift our focus to the Hamming model. Historically speaking, this is the first model we studied. Our original motivation was to find maps that are graceful in the Hamming sense. In §1.3.1, we breifly comment on how this

study led to the development of LDMCs.

In Chapter 7, we characterize the trade-offs for linear codes in the Hamming sense and show that linear codes with good distance are not graceful. It follows from these results that LDMCs are superior to linear codes in a (weak) Hamming sense as well whenever the bandwidth expansion factor is not an integer. In Chapter 8 we study codes over large alphabets and prove some combinatorial results about their so called alpha-beta profile in this regime. Chapter 9 contains some of our exhaustive attempts at constructing graceful codes (in the Hamming sense) prior to the development of LDMCs. We propose general methods to find the best possible short linear codes. These methods use various structures including algebraic, symmetry, linearity, etc. to construct graceful codes.

1.1 Shannon's model

We start by describing more formally a version of the graceful degradation (or joint source-channel) problem for a binary unbiased source and a memoryless erasure channel. Let $X = (X_1, X_2, \cdots, X_k) \sim \text{Ber}(1/2)^{\otimes k}$ be *information bits*. An *encoder* $f : \{0,1\}^k \to \{0,1\}^n$ maps X to a (possibly longer) sequence $Y = (Y_1, \cdots, Y_n)$ where each Y_i is called a *coded bit* and Y is a codeword. The rate of the code f is denoted by $R = k/n$ and its bandwidth expansion by $\rho = n/k$. A channel BEC_ϵ takes Y and produces $Z = (Z_1, \ldots, Z_n)$ where each $Z_j = Y_j$ with probability $(1-\epsilon)$ or $Z_j =?$ otherwise. In this thesis we are interested in performance of the code simultaneously for multiple values of ϵ, and for this reason we denote Z by $Z(\epsilon)$ to emphasize the value of the erasure probability.

Upon observing the distorted information $Z(\epsilon)$, *decoder* g maps $Z(\epsilon)$ into $\hat{X}(\epsilon)$. We measure quality of the decoder by the data bit error rate (BER):

$$\text{BER}_f(\epsilon) := \frac{1}{k}\sum_{i=1}^{k} \mathbb{P}[X_i \neq \hat{X}_i(\epsilon)] = \frac{1}{k}\mathbf{E}[d_H(X, \hat{X}(\epsilon))]\,,$$

where d_H stands for Hamming distance.[1]

Consider the setting of point-to-point (or many) communication where a single user needs to transmit the information bits to interested part(ies). Suppose for now that there is a single user who is interested in the information source, and that the communication takes place over a fixed BEC_ϵ. As mentioned before, a central question in information theory is to determine the amount of data needed at the user's end to recover the source data with some guaranteed fidelity. In this case, we are interested in the best achievable performance for the given channel. We define the information theoretic limit of a family $\mathcal{F}$ of codes for partial recovery w.r.t capacity-to-rate ratio $x = C/R$ as follows

$$\omega_{\mathcal{F}}(x) := \inf_{f \in \mathcal{F}} \mathrm{BER}_f(1 - xR).$$

Note that at $C/R = x$ we have $\epsilon = 1 - xR$. When $C = R$, on average, we observe k coded bits, i.e., the number of available observations on average matches the number of source bits to be estimated. In other words, the ratio C/R measures the excess (or lack thereof) in the average number of available observations for recovering X. The reason to define $\omega_{\mathcal{F}}$ in terms of the ratio C/R (as opposed to the erasure probability ϵ) is to have a unified way of quantifying the information theoretic limit of a family that may contain codes of different rates. We can always restrict a family to sub-codes of fixed rate, or block-length, and study the corresponding partial recovery limit separately.

For linear maps $\mathcal{L}$, it is easy to find a bound for $\omega_{\mathcal{L}}(x)$. Indeed, we have a simple counting problem in hand. To recover any m source bits, we need to observe at least m linear equations (associated with coded bits). The remaining coordinates cannot be guessed better than random (see Prop. 1 below). Therefore

$$\omega_{\mathcal{L}}(x) \geq \frac{1 - x}{2}. \tag{1.1}$$

This lower bound has a nice geometric interpretation that is worth noting. The kernel

[1] We remark that $\mathrm{BER}_f(\epsilon)$ depends on the choice of the decoder as well. We specify the choice of decoder if it is not clear from the context.

of the linear system associated with the observed coded bits specifies the region of uncertainty in which X lies. All points in this region are equally likely to occur and contribute to the distortion in recovering X. We thus need to find a point (not necessarily inside the kernel) that minimizes the average distance to all the points in the kernel, i.e., we want to find the Chebyshev center of the kernel. The above lower bound is tight for sub-cubes. We may thus interpret the bound as follows: among all linear sub-spaces of the Hamming cube with the same dimension, the sub-cubes have the smallest Chebyshev radius.

For general codes, we can again reduce the matters to a counting problem by applying the entropy functional. Roughly speaking, since on average we observe $(1-\epsilon)n$ equations, we can only hope to reduce the entropy of X by $(1-\epsilon)n$ bits upon observing $Z(\epsilon)$. Then it follows from Fano's inequality (and concavity of entropy) that for any family $\mathcal{F}$

$$h(\omega_{\mathcal{F}}(x)) \geq 1 - x, \tag{1.2}$$

where h is the binary entropy function. We call this lower bound the information theoretic limit of partial recovery for a general family of codes. Likewise, (1.1) is called the information theoretic limit for linear codes. The two bounds are shown in Fig. 1-1. It follows from Shannon's achievability theorems for coding [70] and rate distortion [69] that the above two bounds are tight asymptotically, i.e., there exist encoders and decoders that operate close to the curves when n and k are large. When $C > R$, the curves for linear and non-linear codes coincide. We call this regime the *error correction* regime. However, there is a gap between the two curves when $C < R$. We call this regime the *error reduction* regime. The bounds show that non-linear codes are more capable than linear codes in the regime of error reduction.

The gap between $\omega_{\mathcal{L}}$ and $\omega_{\mathcal{F}}$ for $C/R < 1$ has a geometric explanation. The pre-image of a point under a linear map is an affine space, and affine spaces have relatively large diameter (in the Hamming sense). However, the pre-image of a point under a non-linear map can be a set with small diameter. Such sets are known as anticodes and over the binary cube Hamming balls are the optimal anticodes. Indeed

the general lower bound can be achieved by first packing points inside balls in the source space and then encoding the centers optimally. We call the codes that can achieve the information theoretic limit of partial recovery the *Shannon codes*.

When multiple parties are interested in the information source, we need to consider the behavior of $\omega_{\mathcal{F}}$ at different points. Intuitively, we want to say that a family of codes is graceful if $\mathrm{BER}_f(\epsilon)$ varies smoothly with ϵ for some f, while satisfying some required fidelity criteria by users. To formalize this notion, we can fix an erasure probability ϵ_1 and a minimum admissible recovery quality δ_1. Then among all the codes (in the family) satisfying $\mathrm{BER}(\epsilon_1) \leq \delta_1$, we look for one that has the lowest possible BER at some ϵ_2, i.e., a code that gives the best possible improvement (resp. least possible degradation) as more (resp. less) data becomes available. We thus introduce the two point trade-off function as follows.

Definition 1. *Given a family $\mathcal{F}$ of codes, the two point trade-off function of f at (ϵ_1, δ_1) is defined as*

$$\eta_{\mathcal{F}}(\epsilon_2; \epsilon_1, \delta_1) = \inf_{f \in \mathcal{F}} \{\mathrm{BER}_f(\epsilon_2) : \mathrm{BER}_f(\epsilon_1) \leq \delta_1\}$$

where the BER *functions are computed w.r.t to the optimal (bitwise* MAP*) decoder.*

It follows from our results in Chapter 2 (see Theorem 5) that linear codes are not graceful, i.e., their trade-off function has a threshold like behavior. That is to say, if a linear code is efficient for partial recovery of one user it performs poorly for the other. For instance, consider the case with two users where user 1 is interested in 50% of the source bits and user 2 is interested in 25% of the source bits. Can we design linear codes so that, on average, user 1 can reach his goal by observing around $0.5k$ coded bits and user 2 can achieve his by observing close to $0.25k$ coded bits? Unfortunately, the answer is no as shown in Fig.1-1. Similarly, separation codes of Shannon suffer from the same issue. However, we shall see that there exist non-linear codes that can provide a graceful degradation in performance while staying close or even below the fundamental line of linear codes. A prevalent barrier in using non-linear codes is their decoding complexity. Indeed the idea of solving linear systems

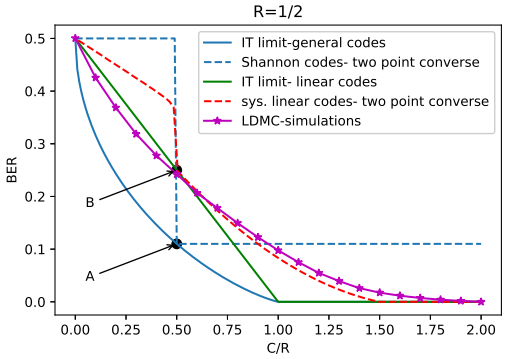


Figure 1-1: The lower bounds for codes of rate $R = 1/2$ vs achievability with systematic LDMCs as defined in §1.3. Here $C = 1 - \epsilon$ is the capacity of the erasure channel and BER is computed w.r.t the source distortion $\mathbf{E}[d(X, \hat{X})]/k$, where d is the Hamming distance and k is the number of source bits. We note that on average k coded bits are returned by the channel at $C/R = 1$. Shannon codes that achieve the information theoretic limit suffer from an ungraceful collapse. The two point converse for systematic linear codes is from Theorem 5 and is computed for codes that can achieve the point $B = (0.5, 0.2501)$, i.e., they satisfy BER ≤ 0.2501 at $\epsilon = 0.75$. This means that they can, on average, recover $0.499k$ coordinates from $0.5k$ observations. The bound is stable, i.e., a small perturbation on the location of point B cannot prevent the step-like behavior of the code. The lower bound shows that almost no unobserved coordinates can be recovered when $C/R < 0.5$. Furthermore, separation codes that pass through point $A = (0.5, 0.1101)$ suffer from the same problem. The LDMCs can however achieve a graceful decline while surpassing the fundamental limitations of linear codes when $C/R \leq 0.5$.

of equations should in general be more appealing than solving non-linear equations with no structure. The codes that we shall present shortly are, however, efficiently decodable and can surpass capabilities of linear codes for partial recovery. We call these codes Low Density Majority Codes(LDMCs) and describe them in §1.3. As shown in Fig.1-1, LDMCs can simultaneously achieve smaller error than any linear codes for both users.

We shall be mainly interested in the trade-off function of the family $\mathcal{L}$ of linear codes in comparison with LDMCs. The following definition is relevant.

Definition 2. *A code g is said to (ϵ_1, ϵ_2)-dominate $\mathcal{F}$ if there exists δ_1 so that* $\text{BER}_g(\epsilon_1) \leq \delta_1$ *and* $\text{BER}_g(\epsilon_2) \leq \eta_{\mathcal{F}}(\epsilon_2; \epsilon_1, \delta_1)$. *If* $\text{BER}_g(\epsilon) \leq \text{BER}_f(\epsilon)$ *for all ϵ and all $f \in \mathcal{F}$, then g is said to dominate $\mathcal{F}$.*

The question of graceful degradation for a code over a family $\{\text{BEC}_\epsilon\}_{\epsilon\in[\epsilon_1,\epsilon_2]}$ of channels can now be discussed in terms of (ϵ_1, ϵ_2)-domination w.r.t to $\mathcal{F}$ for a rich enough family of maps $\mathcal{F}$. We study the trade-off function of the family $\mathcal{L}$ of systematic linear codes in Chapter 2 and show that LDMCs dominate $\mathcal{L}$ in the error reduction regime.

1.2 Hamming's model

We now discuss some versions of graceful degradation in the Hamming sense [33, 34].

Definition 3. *A code $f : \mathbb{F}_q^k \to \mathbb{F}_q^n$ is said to be a $[n, k, D(\delta)]$ combinatorial-joint-source-channel-code (CJSCC), if*

$$\forall x, |e| \leq \delta k \implies d_H(g(f(x+e)), x) \leq Dk.$$

CJSCCs can be viewed as maps that "contract" the input (noise) error δ to output (decoder) error D. In particular, if the contraction is linear, i.e., if $D(\delta) \leq \lambda\delta$, then the CJSCC is said to be λ-error reducing [71]. In this sense, CJSCCs generalize the notion of error reducing codes.

A related notion to CJSCC is that of the (α, β)-property [59]. A mapping of k symbols to n symbols is said to have the (α, β)-property if it sends any two strings of (Hamming) distance more than αk to two strings of (Hamming) distance more than βn.

Definition 4 ([60])**.** *A map $f : \mathbb{F}_q^k \to \mathbb{F}_q^n$ is said to be $[n, k, \beta(\alpha)]$ if*

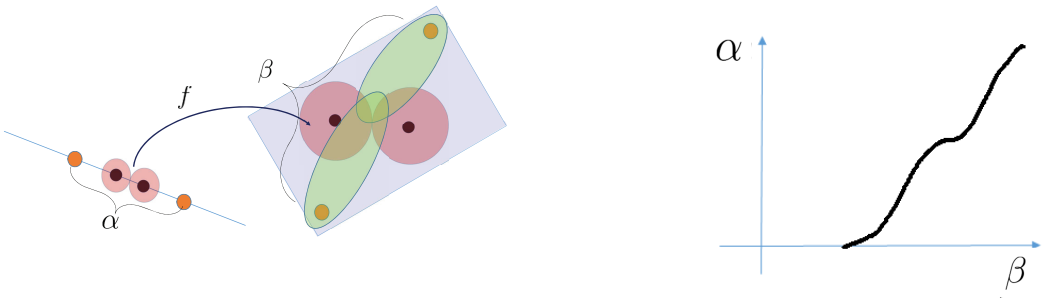
$$|x - y| > \alpha k \implies |f(x) - f(y)| > \beta n,$$

where $|\cdot|$ denotes the Hamming weight.

The (α, β)-property can be seen as a relaxation of the CJSCC property. Indeed, the (α, β)-property is a restriction of the CJSCC property where the decoder is forced to pick the estimate from the pre-image of f. For linear codes, and in the regime of

large alphabets, the two notions are equivalent. Since our combinatorial results fall in this regime, we do not distinguish between the two.

The (α, β)-property is a natural generalization of minimum distance. We think of β as the level of erasure noise needed to cause a relative distortion of α in the input space (see Fig. 1-2a). A graceful code in this sense is one with a smooth (α, β)-profile similar to Fig.1-2b. Such a code can fully recover the input when the noise level is below its error correcting capability and can paritialy recover it once the noise level exceeds its error correction capabiltiy.



(a) The (α, β)-property (b) A (fictitious) graceful code

Figure 1-2: The alpha-beta property and the corresponding notion of graceful degradation. a) The (α, β)-property. The images of the points along with the regions of disturbance required to cause confusion are shown. Any two points that are more than α-away in the input space are sent to points that are β-away in the output space. As a result the amount of noise needed in the target space to cause α-distortion is larger than minimum distance. b) The "ideal" profile in the Hamming sense. Such profile would indicate that the code can fully recover the input when the noise level is small and can partially recover it as the noise level increases. Finding codes with such profiles is a difficult task.

For a linear map f, we can equivalently define

$$\beta_f(\alpha) := \inf_x \{\frac{|f(x)|}{n} \mid |x| > \alpha k\} - \frac{1}{n}$$

and

$$\beta_f^* := \beta_f(1 - \frac{1}{k}). \tag{1.3}$$

Note that the (relative) minimum distance of f is $\beta(0) + \frac{1}{n}$.

Operationally, β^* characterizes the threshold for adversarial erasure noise beyond

which the decoder cannot guaranteed to recover a single bit. Alternatively, $1 - \beta_f^*$ is the fraction of equations needed such that f can always recover at least one input symbol. Likewise, $1 - \beta(0)$ measures the minimum number of equations needed so that f can fully recover the input. It is thus ideal to have a code with large minimum distance and monotonically increasing $\beta(\alpha)$. Such a code can fully recover the input when the number of erasures is less than its minimum distance, and as the number of erasures exceeds its minimum distance, it can offer some partial recovery guarantees.

It turns out that, similar to the stochastic case, there is a trade-off between error correction and gracefulness. We study these trade-offs in detail in Chapter 7. In particular, we show that the only linear codes that can asymptotically achieve $\beta^* = 1$ are repetition like and that no such linear codes exsit when $\rho \notin \mathbb{Z}$. However, LDMCs always satisfy $\beta^* = 1$ in a stable way that easily extends to the asymptotic limits as well. Thus LDMCs can dominate linear codes in the (admittedly weak) sense of β^* as well.

1.3 The LDMC ensemble

We first define the notion of a check regular code ensemble generated by a Boolean function.

Definition 5. *Let $\mathbf{P}_\Sigma$ be a joint distribution on m-subsets of $[k]$. Given a Boolean function $f : \{0,1\}^m \to \{0,1\}$, the (check regular) ensemble of codes on $\{0,1\}^k$ generated by $(f, \mathbf{P}_\Sigma)$ is the family of random codes $f_\Sigma : x \mapsto (f(x_S))_{S \in \Sigma}$ obtained by sampling $\Sigma \sim \mathbf{P}_\Sigma$.Here x_S is the restriction of x to the coordinates indexed by S.*

Given $x \in \{0,1\}^d$, we consider the d-majority function

$$\text{d-maj}(x) = \mathbb{1}_{\{\sum_i x_i > \frac{d}{2}\}}.$$

We have the following definition:

Definition 6. *Let $\mathbf{U}_\Sigma = \text{Unif}^{\otimes n}(\{\text{d-subsets of } [k]\})$ be the uniform product distribution on the d-subsets of $[k]$. The ensemble of codes generated by $(\text{d-maj}, \mathbf{U}_\Sigma)$ is*

*called the Low Density Majority Code (*LDMC*) ensemble of degree d and denoted by* LDMC*(d). Furthermore, define the event $A = \{\sum_{S\in\Sigma} \mathbb{1}_{\{i\in S\}} = \sum_{S\in\Sigma} \mathbb{1}_{\{j\in S\}}\}$, i.e., the event that each i appears in the same number of d-subsets S. Then the ensemble generated by* $(\text{d-maj}, \mathbf{U}_{\Sigma|A})$ *is called a regular* LDMC*(d) ensemble.*

We shall also consider systematic LDMCs, which are codes of the form $x \mapsto (x, f(x))$ where f is picked from a regular LDMC ensemble. Throughout this work, we also refer to the check regular ensemble generated by the XOR function, known as the Low Density Generator Matrix codes (LDGMs).

1.3.1 A comment

Before we present our technical results on LDMCs, we briefly explain what led us to study this family of codes.

While LDMCs posses many interesting properties in the Shannon sense of graceful degradation, our original motivation lied, in fact, in finding graceful codes for the Hamming model, i.e., we wanted to construct maps with smooth (α, β)-profiles, similar to what is shown in Fig. 1-2b. It was however our expectation that a solution to the latter problem would provide at least some insight on how to make progress on the former. There seems to be a recurring theme in coding theory that codes designed on the basis of good Hamming properties turn out to be useful for the stochastic problem as well. A recent example is the work of [42] which shows that many families of good error correcting codes, which were originally designed to give protection against the combinatorial noise in the Hamming model, can achieve full recovery in the stochastic sense as well. Further evidence will come in Chapter 9 where we shall see that some small codes with smooth (α, β)-profile perform well against stochastic noise as well (see Fig. 9-2). We thus set out to find codes whose profiles looked like that of Fig. 1-2b.

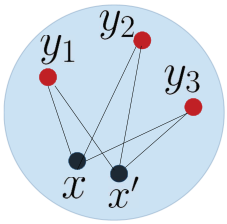
To design good codes in the combinatorial sense, one idea that appealed to us was to "geometrize" the problem, i.e., to forget momentarily that we are working over the Hamming cube and lift the problem to the Euclidean space. We then considered the

following construction. Pick some randomly chosen triangulations points $y_1, \cdots, y_n$ inside $\{0,1\}^k$. Given an input x, compute inner products (or alternatively distances) between the input and the triangulation points $f : x \mapsto ((x, y_1), \cdots, (x, y_n))$. The idea is shown in Fig. 1-3. This map is easily seen to preserve nearby distances. The principle here is that when we try to triangulate a point, our estimate of where the point is should vary smoothly with the position of the point. But there is a problem with the points that are far away: any two typical points x, x' that are maximally apart will likely land in coordinates that differ by at most $O(\sqrt{k})$. This means that the relative distance in the target space vanishes, and there seems to be no way to salvage a good code of positive rate out of this. One way around this problem is to pick our triangulation vectors to be sparse. Then the relative distance of such points in every coordinate is of the same order as the weight of the triangulating vector. Hence there is some hope to obtain smooth binary codes of non-vanishing rate after quantization. The all-important question is now this: how to quantize to get back a binary codes? In some sense if the quantization map is not itself smooth then the overall scheme fails. It may not seem like we have made much progress at this point since we have just reduced the task of finding smooth codes to finding smooth quantizers, a problem that had been observed and reported before in the context of graceful degradation [25]. After some time we learned that the right way to quantize is to use a 1-bit quantizer, i.e., to compute majorities. This was the first idea that enabled us to construct maps that dominated the repetition code (see Fig.1-4).

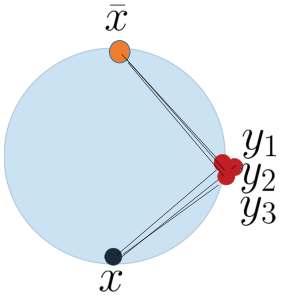
1.4 Main contributions

Our main results are as follows:

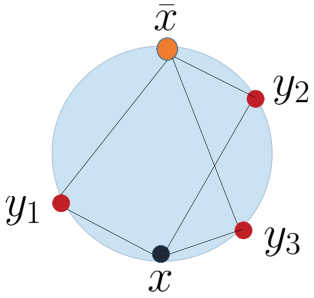
- We establish two-point lower bounds for the partial recovery trade-off function of systematic linear codes in the stochastic setting in Chapter 2. Together with the bounds of Chapter 4, these bounds show that systematic LDMCs are provably more capable than any systematic linear code for partial recovery. These bounds also improve on existing bounds (cf. [36]) for the stochastic



(a) A triangulation code



(b) High dimensional picture



(c) Sparse triangulation code

Figure 1-3: The notion of a random triangulation code. a) The map $f : x \mapsto ((x, y_1), (x, y_2), (x, y_3))$ is smooth, i.e., it preserves nearby distances. b) A problem emerges in high dimensions as distances concentrate for complimentary points. Indeed if $\bar{x}$ is the compliment of x then its (relative) inner product with a randomly chosen point different from that x by at most $O(1/\sqrt{n})$. c) The problem is resolved once the triangulation points are chosen to be low weight vectors.

broadcast JSCC problem in the case of linear codes. They may also be used to derive informative bounds for the finite length analysis of linear codes.

- We study the implications of the area theorem in the stochastic setting in Chapter 3. We conclude that our bounds are tighter than those obtained via area theorem. In retrospect, this is not surprising since in the case of input BER there are no conservation laws (see Example 1 in Chapter 3).

- We use various notions of channel comparison to provide a general method for analysis of information propagation for BP in Chapter 4. In the process, we prove a variant of Mrs. Gerber's Lemma, which may prove useful in convexifying information measures for parametric distributions.

- In particular, we apply our tools to compute upper and lower bounds for BP

error of LDMC(3) and LDMC(5). A data processing argument shows that the lower bound is universal, i.e., it holds for the optimal (bitwise MAP) decoder as well. In the case of LDMC(3), the lower bound is very close to our empirical results, which means there can only be a small gap between the optimal and BP decoder for LDMC(3) for any erasure level. In the case of systematic LDMC(3), our upper and lower bounds match fairly well.

- We construct an asymptotic upper bound for BP error of systematic LDMC(d) of large degrees. The bound does not depend on degree and relies on propagating messages in just 1 iteration of BP. The bound tightly fits our simulation results, demonstrating that asymptotics in d kicks in early.

- We show that LDGM constructions can be uniformly improved by replacing repetition code with LDMCs in Chapter 5. A joint optimization over LDGM and LDMC is shown to improve on partial recovery for all noise levels.

- In the combinatorial setting, we establish the trade-offs between minimum distance and the recoverability threshold β^* for linear codes in Chapter 7. We show that linear codes cannot achieve $\beta^* = 1$ when the bandwidth expansion factor ρ is not an integer. It follows that no linear repetition like code exists when $\rho \notin \mathbb{Z}$, answering a question asked in [60]. LDMCs on the other hand can achieve $\beta^* = 1$.

- We provide tight bounds for the (α, β)-limits of general codes over large alphabets in Chapter 8 and present explicit (short) codes with good (α, β)-properties in Chapter 9 .

1.5 Prior work

Rateless codes

To solve the multi-cast problem over the internet, the standard TCP protocol uses feedback to deal with erasures, i.e., each lost packet gets re-transmitted. This scheme

is optimal from a data recovery point of view. From any k received coded data bits, k source bits can be recovered. Hence it can achieve every point on the fundamental line of Fig. 1-1. However, a separate feedback line is not always available, and using the same channel to implement feedback has other complications. For instance, when many packets are likely to get dropped, feedback has a large overheard (or the excess in information bits required to reconstruct the source). Alternatively, a forward error correcting code can be used to deal with data loss. A preliminary analysis in [44] shows that forward error correction can save up to 25% in overhead compared to a feedback approach over a typical Internet network.

In particular, Fountain codes have been introduced to solve the problem of multi-casting over the erasure channel [12]. They are a family of linear error correcting codes that can recover k source bits from any $k + o(k)$ coded bits with small overhead. A special class of fountain codes, called systematic Raptor codes, have been standardized and are used for multi-casting in 3GPP [8, 26, 4, 9, 45]. Various extensions and applications of Raptor codes are known [15, 14]. However, as observed in [67], these codes are not able to adapt to the user demands and temporal variations in the network.

As less data becomes available at the user's end, it is inevitable that our ability to recover the source deteriorates. However, we may still need to present some meaningful information about the source to the user, i.e., we want to partially recover the source. For instance, in sensor networks it becomes important to maximize the throughput of the network at any point in time since there is always a high risk that the network nodes fail and become unavailable for a long time [30]. In such applications it is important for the codes to operate *gracefully*, i.e., to partially recover the source and improve progressively as more data comes in. We show in Chapter 2 that Fountain codes, and more generally linear codes, are not graceful for forward error correction. Hence, it is not surprising that many authors have tried to develop graceful linear codes by using partial feedback [30, 27, 6, 13]. However, we shall challenge the idea that graceful degradation (or the online property) is not achievable without feedback [13]. Indeed LDMCs give a family of efficient (non-linear) error reducing

codes that can achieve graceful degradation and can perform better than any linear code in the sense of partial recovery (see Fig.1-1).

Raptor codes are essentially concatenation of a rateless Tornado type error-reducing code with an outer error correcting pre-coder. Forney [23] observed that concatenation can be used to design codes that come close to Shannon limits with polynomial complexity. Forney's concatenated code consisted of a high rate error correcting (pre)-coder that encodes the source data and feeds it to a potentially complicated inner error correcting code. One special case of Raptor codes, called pre-code only Raptor code is the concatenation of an error correcting code with the repetition code. Recently, such constructions are becoming popular in optics. In these applications it is required to achieve 10^{-15} ouput BER, much lower than the error floor of LDPC. Concatenation with a pre-coder to clean up the small error left by LDPCs is one way to achieve the required output BER [72]. It was shown recently however that significant savings in decoding complexity (and power) can be achieved if the inner code is replaced with a simple error reducing code and most of the error correction is left to the outer code [81, 5].

These codes, as all currently known examples of concatenated codes, are linear. They use an outer linear error correcting code (BCH, Hamming, etc) and an inner error reducing LDGM. The LDGM code however operates in the regime of partial data recovery. It only produces an estimate of the source with some distortion that is within the error correcting capability of the outer code. To achieve good error reduction, however, LDGMs still need rather long block-length and a minimum number of successful transmissions. In other words, they are not graceful codes (see Fig. 5-4). We shall see in Chapter 5 that LDMCs can uniformly improve on LDGMs in this regime. Thus, we expect that LDMCs appear in applications where LDGMs are currently used for error reduction.

Joint Source-Channel Coding

The problem discussed in this work can be viewed as an example of broadcasting with a joint source-channel code (JSCC), which is considered one of the challenging open

problems is network information theory [36, 35, 64, 74, 32]. In general it is known that the users have a conflict of interests, i.e., there is a trade-off between enhancing the experience of one user and the others. For instance, if we design the system to work well for the less resourceful users, others suffer from significant delay. Likewise, if we minimize the delay for the privileged users, others suffer significant loss in quality. Naturally, there are two questions we are interested in: 1) what are the fundamental trade-offs for partial recovery 2) how do we design codes to achieve them?

Many achievability and converse bounds are available for the two user case under various noise models [64, 58, 2, 22, 18]. In turns out that in most cases there is a gap between achievablity and converse bounds. In a sense, the theory and practice of partial recovery so far are much less developed compared with the classic setting of full recovery with one user (also known as point-to-point communication). For the classic problem, Shannon provided a converse for full recovery and showed that it is asymptotically tight using a non-constructive (random coding) argument. Over the years many practical codes were developed that can achieve good performance in the sense of full recovery and admit efficient decoding. These codes mostly rely on the idea that linear systems of equations with proper structure (symmetry, sparsity, etc) can be solved efficiently. However, for the two user case the best achievability results are either non-constructive [22], or involve complicated non-linearities (e.g., compression at different scales [24][37]). Shannon also developed the rate distortion theory of partial recovery for one user and showed that separation is asymptotically optimal. In practice, however, the codes are finite and it is known that in this regime separation is not optimal (see [36] and references therein). Furthermore, lossy compression is inherently nonlinear and separating it from coding adds another layer of complexity to the system. This is the problem that JSCCs attempt to solve.

A classic error correction solution is not completely satisfactory here. Indeed for error correction to work, we need to know the channel quality. If we design the code to work well in the worst case situation, we suffer significant delay. If we assume a best case channel, we suffer significant loss in recovery once the channel quality drops below the design rate. This sudden drop in quality is known as the "cliff

effect" [24] and shown in Fig.1-4 for LDPC codes. Roughly speaking, there is a phase transition in the BER performance of LDPCs or any capacity achieving code. When the noise level is below a certain threshold the input can be recovered with small error. When the noise level exceeds that threshold the input cannot be recovered with good fidelity. This is a consequence of the so called area theorem and will be visited later. Our results show that the "cliff effect" persists in the range of partial recovery as well. That is, any linear code that comes close to the fundamental limits of partial recovery cannot be graceful. This latter result cannot be inferred from the area theorem (see Chapter 3) or the general converses known for the JSCC problem.

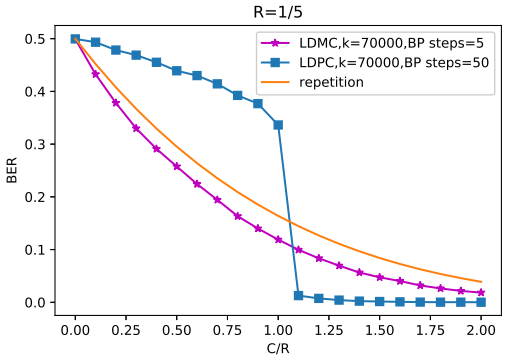


Figure 1-4: Comparing BER at different erasure channels for three codes with rate 1/5: an LDPC code with $k = 70000$ data bits using 50 iterations of BP, the repetition code, and LDMC(3) with $k = 70000$ information bits using 5 iterations of BP. The LDMC code does not have any systematic bits. The LDPC code suffers from the cliff effect. Here C is the capacity of the channel and R is the rate of the code.

The repetition code, on the other hand, can recover the input bits partially at all channel noise levels. Of course, its performance degrades as the channel capacity drops but it does so in a graceful way.

Much work has been done recently to find graceful codes in the literature [37, 77, 24, 10, 54, 25]. Such approaches can broadly be categorized into JSCC solutions. It was known since the early days of communication that trade-offs exist between error correction and gracefulness of a code [82]. We characterize these trade-offs for linear codes under erasure noise. Our bounds on the trade-off functions give new converses for broadcasting with linear codes. These bounds are stronger than those inferred

from the area theorem (see Chapter 3) or the general converses known for JSCC (cf. [36]). Our results reveal that, unlike the classic setting, the converse bounds cannot be achieved with linear codes. Hence, to find good practical codes for broadcasting we need to cosider non-linear codes. Our proposed codes, LDMCs, may prove to be helpful in this regard.

Non-linear codes

Codes whose computational graph (see Fig.4-2) are sparse are known as sparse graph codes. Many such codes are known [47] and can achieve near Shannon limit performance. With a few exceptions, these codes are mostly linear. One problem with linear codes is that BP cannot be initiated without the presence of low degree nodes. In [16], the authors observe that non-linear functions do not have this problem and use random sparse non-linear codes to achieve near optimal compression using BP. However, using non-linear functions in this setting is mainly due to algorithmic considerations, namely, to enable the use of BP. Otherwise, similar compression results can be obtained by using LDGMs under different message passing rules[79]. In [56], the authors use special non-linear sparse graph codes to build optimal smooth compressors. In all of these works, however, the focus is on point-wise performance and a result the codes are optimized to operate at a particular rate. As such, they are unlikely to achieve graceful degradation.

Another relevant work in this area is that of random constraint-satisfaction problems (CSPs) with a planted solution [40]. It appears that the CSP literature mostly focused on geometric characterization of spaces of solutions and phase transitions thereof. These do not seem to immediately imply properties interesting to us here (such as graceful degradation).

Part 1

Shannon's stochastic model

Chapter 2

Trade-offs for linear codes

Systematic linear codes form a vast majority of the codes that are used in practice. In this section, we work towards proving that LDMCs are optimal w.r.t to this family. In the following, by $\ker(A)$ we refer to the left kernel of A, that is the subspace of vectors x satisfying $xA = 0$.

2.1 hrank

Definition 7. *Given a matrix A define*

$$\mathrm{hrank}(A) = |\{j : \ker(A) \subset \{x : x_j = 0\}\}|$$

Definition 8. *Given a matrix A, define $\tilde{A}(p, q)$ to be a random sub-matrix of A that is obtained by sampling each row of A with probability p and each column of A with probability q independently of other rows/columns.*

The following proposition is well known (cf. [65]).

Proposition 1. *Consider a system of equations $xG = y$ over $\mathbb{F}_2$. If $\ker(G) \subset \{x : x_i = 0\}$, then x_i is uniquely determined from solving $xG = y$. Otherwise, there is a bijection between the set of solutions $\{x : xG = y, x_i = 0\}$ and $\{x : xG = y, x_i = 1\}$. In particular, if exactly t coordinates are uniquely determined by the above equations, then $\mathrm{hrank}(G) = t$.*

Our next proposition relates BER and hrank.

Proposition 2. *Let $G = [I\ A]$ be the generator matrix of a systematic linear code f with rate R. Then* $\mathrm{BER}_f(\epsilon) \leq \delta$ *if and only if*

$$\mathbf{E}[\mathrm{hrank}\left(\tilde{A}(\epsilon, 1-\epsilon)\right)] \geq (\epsilon - 2\delta)k.$$

Proof. If BER is bounded by δ, there are, on average, at most $2\delta k$ bits that are not uniquely determined by solving $x\tilde{G}(1, 1-\epsilon) = y$. For a systematic code, the channel returns $\mathrm{Bin}(k, 1-\epsilon)$ systematic bits. The remaining systematic bits x_r are to be determined from solving $x_r\tilde{A}(\epsilon, 1-\epsilon) = \tilde{y}$ where $\tilde{y}$ is some vector that depends on the channel output y and the returned systematic bits. If t additional systematic bits are recovered, then $\mathrm{hrank}(\tilde{A}(\epsilon, 1-\epsilon)) = t$ by Proposition 1. Since on average at least $(\epsilon - 2\delta)k$ additional systematic bits are recovered, the claim on the average hrank follows. ■

2.2 Trade-offs between rank and hrank

The next proposition shows how matrices with positive hrank behave under row sub-sampling. Our main observation is that row sub-sampled matrices of a (thin) matrix with large hrank have bounded rank. In particular, if a (thin) matrix has full hrank, its sub-sampled matrices cannot have full rank.

Proposition 3. *Consider and arbitrary field $\mathbb{F}$ and let $\epsilon_1 > \epsilon_2$. Given a $k \times m$ matrix A,*

$$\mathbf{E}[\mathrm{rank}\left(\tilde{A}(\epsilon_2, 1)\right)] \leq \mathrm{rank}(A) - (1 - \frac{\epsilon_2}{\epsilon_1})\mathbf{E}[\mathrm{hrank}\left(\tilde{A}(\epsilon_1, 1)\right)],$$

and

$$\mathbf{E}[\mathrm{hrank}\left(\tilde{A}(\epsilon_2, 1)\right)] \geq \frac{\epsilon_2}{\epsilon_1}\mathbf{E}[\mathrm{hrank}\left(\tilde{A}(\epsilon_1, 1)\right)].$$

Therefore, if $\mathbf{E}[\mathrm{rank}\left(\tilde{A}(\epsilon_2, 1)\right)] = \mathrm{rank}(A) - o(k)$, *then* $\mathbf{E}[\mathrm{rank}\left(\tilde{A}(\epsilon_1, 1)\right)] = o(k)$.

Proof. Suppose that $\text{hrank}\left(\tilde{A}(\epsilon_1, q)\right) = t$. This means that there are at least t rows a_j in $\tilde{A}(\epsilon_1, q)$ such that a_j is not in the span of $\{a_i : i \neq j\}$. Let B be the row-submatrix of $\tilde{A}(\epsilon_1, q)$ associated to these t rows, and B^c be its compliment, i.e., the matrix with rows $\{a_j : a_j \in \tilde{A}(\epsilon_1, q), a_j \notin B\}$. We claim that the compliment of B is a matrix of rank $\text{rank}(A) - t$. To see this, note that $\text{Im}(B) \cap \text{Im}(B^c) = \{0\}$, for otherwise we get linear dependencies of the form $h = \sum_i \alpha_i b_i \neq 0$ where $b_i \in B$ and $h \in \text{Im}(B^c)$, which contradicts the construction of B. This means that $\text{rank}(B^c) + \text{rank}(B) = \text{rank}(A)$. The claim now follows since $\text{rank}(B) = t$. Under row sub-sampling, each row of B is selected with probability ϵ_2/ϵ_1 independently of other rows. Thus,

$$\mathbf{E}[\text{hrank}\left(\tilde{A}(\epsilon_2, q)\right) | \text{hrank}\left(\tilde{A}(\epsilon_1, q)\right) = t] \geq \frac{\epsilon_2}{\epsilon_1} t$$

The rows selected from B^c can contribute at most $\text{rank}(A) - t$ to the rank of $\tilde{A}(\epsilon_2, q)$. Hence

$$\mathbf{E}[\text{rank}\left(\tilde{A}(\epsilon_2, q)\right) | \text{hrank}(\tilde{A}(\epsilon_1, q)) = t] \leq \frac{\epsilon_2}{\epsilon_1} t + \text{rank}(A) - t$$

Taking the average over the hrank of $\tilde{A}(\epsilon_1, q)$ proves the first two results. The last inequality follows by re-arranging the terms.

■

Remark 1. In general the above bound cannot be improved up to $o(k)$ deviations. Indeed we can partition the matrix $\tilde{A}(\epsilon_1, 1)$ in the form

$$\begin{bmatrix} B \\ O \\ F \end{bmatrix}$$

where B is a basis with $\text{hrank}(\tilde{A}(\epsilon_1, 1))$ many rows, O is the zero matrix, and F is a redundant frame with $f > 1 - \epsilon_1 - t$ rows that span the co-kernel of B. This means that any $1 - \epsilon_1 - t$ rows in F form a basis for the image of F. Now for any $\epsilon_2 < \epsilon_1$, if $\frac{\epsilon_2}{\epsilon_1} f = 1 - \epsilon_1$, then we sub-sample a basis from f with high probability. Thus the *hrank* of the sub-sampled matrix $\tilde{A}(\epsilon_2, 1)$ can jump up with high probability for large

k.

The next Proposition shows that rank is well behaved under column sub-sampling.

Proposition 4. *Consider an arbitrary field $\mathbb{F}$ and let $p > q$. Given a $k \times m$ matrix A over $\mathbb{F}$,*

$$\mathbf{E}[\text{rank}\left(\tilde{A}(1,p)\right)] \leq \min\{pm, \frac{p}{q}\mathbf{E}[\text{rank}\left(\tilde{A}(1,q)\right)]\}.$$

Proof. Pick a column basis for $\tilde{A}(1,p)$. We can realize $\tilde{A}(1,q)$ by sub-sampling columns of $\tilde{A}(1,p)$. In this way, each column in the basis of $\tilde{A}(1,p)$ is selected with probability q/p independently of other columns. In other words,

$$\mathbf{E}[\text{rank}\left(\tilde{A}(1,q)\right)] \geq \frac{q}{p}\mathbf{E}[\text{rank}\left(\tilde{A}(1,p)\right)].$$

The desired result follows. ■

2.3 Converse

We are now ready to prove our main result.

Theorem 5. *Let $f : x \mapsto xG$ be a systematic linear code of rate $1/\rho$ with generator matrix $G = [I \mid A]$ over $\mathbb{F}_2$. Fix $\epsilon_1 > \epsilon_2$ and $\delta_1 \leq \frac{\epsilon_1}{2}$. If $\text{BER}_f(\epsilon_1) \leq \delta_1$, then*

$$\text{BER}_f(\epsilon_2) \geq \kappa(\epsilon_1, \delta_1, \rho) \triangleq \frac{\epsilon_2 - \frac{1-\epsilon_2}{1-\epsilon_1}\left[\frac{\epsilon_2}{\epsilon_1}\gamma + (\rho-1)(1-\epsilon_1) - \gamma\right]}{2}$$

with $\gamma = \epsilon_1 - 2\delta_1$. In particular, if $\text{BER}(\epsilon_2) = \epsilon_2 - \frac{1}{2} + o(1)$, then $\text{BER}(\epsilon_1) = \frac{\epsilon_1}{2} - o(1)$. Furthermore, if $\epsilon_2 > \epsilon_1$

$$\text{BER}_f(\epsilon_1) \geq \inf_{\delta_2}\{\delta_2 : \kappa(\epsilon_2, \delta_2, \rho) \leq \delta_1\}.$$

Proof. By Proposition 2, we have $\mathbf{E}[\text{hrank}\left(\tilde{A}(\epsilon_1, 1-\epsilon_1)\right)] \geq \gamma k$. By Proposition 3, we have

$$\mathbf{E}[\text{rank}\left(\tilde{A}(\epsilon_2, 1-\epsilon_1)\right)] \leq (\frac{\epsilon_2}{\epsilon_1}\gamma + (\rho-1)(1-\epsilon_1) - \gamma)k.$$

By Proposition 4, we have

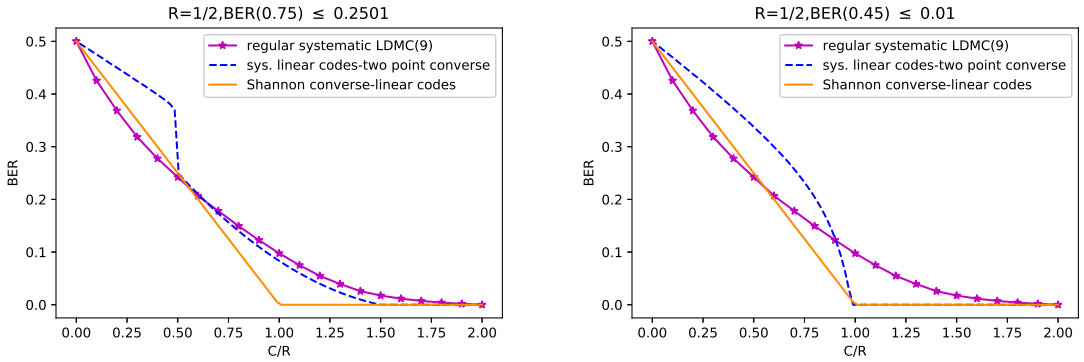
$$\mathbf{E}[\text{rank}\left(\tilde{A}(\epsilon_2, 1-\epsilon_2)\right)] \leq \frac{1-\epsilon_2}{1-\epsilon_1}(\frac{\epsilon_2}{\epsilon_1}\gamma + (\rho-1)(1-\epsilon_1) - \gamma)k.$$

The first result now follows from Proposition 2 upon observing that $\text{hrank}(\tilde{A}) \leq \text{rank}(\tilde{A})$.

The second result follows since $\text{BER}(\epsilon_2) = \epsilon_2 - \frac{1}{2} + o(1)$ implies that $\text{hrank}(\tilde{A}(\epsilon_2, 1-\epsilon_2)) = (1-\epsilon_2)k - o(k)$ by Proposition 2. By the second part of Proposition 3, we have $\text{hrank}(\tilde{A}(\epsilon_1, 1-\epsilon_1)) = o(k)$. The result follows after applying Proposition 2 again. ■

2.4 LDMCs vs linear systematic codes

Fig. 2-1 shows the lower bound for codes of rate $\frac{1}{2}$. It can be seen that regular systematic LDMCs of rate 1/2 $(0.5, \epsilon)$-dominate linear codes for all $\epsilon \leq 0.5$ and cannot be much worse when $\epsilon > 0,5$. In fact we do not believe that the lower bound for linear codes is tight and expect LDMCs to dominate all linear codes of rate 1/2 that can achieve $\text{BER}(0.75) = 0.25 + o(1)$.



(a) Lower bound for linear codes with BER(0.75) $\leq$ 0.2501

(b) Lower bound for linear codes with BER(0.45) $\leq$ 0.01

Figure 2-1: The LDMC performance along with the lower bound of Theorem 5 on the BER for the systematic linear codes of rate $\frac{1}{2}$ satisfying a) BER(0.75) $\leq$ 0.2501 and b) BER(0.45) $\leq$ 0.01. The left figure shows that any systematic linear code that comes close to the Shannon limit for linear codes suffers from ungraceful collapse. For such codes, there is a threshold such that almost any further improvement on BER for erasure probabilities below the threshold comes from the systematic observations. Furthermore, almost no unobserved bit can be recovered as the erasure probability exceeds the threshold. The right figure shows that it is not possible to attain good performance in the error reduction regime with systematic linear codes at the cost of tolerating a small error. Even at 10% overhead, systematic linear codes that achieve BER lower than 0.01 exhibit a sharp decay in performance once $C < R$.

Chapter 3

Bounds via area theorem

The lower bound of Theorem 5 states that a linear systematic code cannot have small BER for all erasure probabilities. In this sense, it has the flavor of a "conservation law". In coding theory, it is often important to understand how a code behaves over a family of parametrized channels. The main existing tool in the literature to study such questions is the so called area theorem. Here we introduce the theorem and study its consequences for two point bounds on BER. It turns out that the bound in Theorem 5 is tighter than what can be inferred from the area theorem.

Let us first provide an example to show that there is no conservation laws for input BER:

Example 1. Let f be the 2 fold repetition map $X \mapsto (X, X)$. Let g be a systematic code sending $x_i \mapsto (x_i, X_i, x_i)$ for all odd i and $x_j \mapsto (x_j)$ for all even i. Then $\mathrm{BER}_f(\epsilon) = \frac{1}{2}\epsilon^2$ and $\mathrm{BER}_g(\epsilon) = \frac{1}{2}(\frac{1}{2}\epsilon^3 + \frac{1}{2}\epsilon)$. It can be checked that f dominates g. This means that among repetition codes a balanced repetition is optimal.

3.1 Area theorem

Following [65], we define the notion of an *extrinsic information transfer* (EXIT) function.

Definition 9. *Let W be a codeword chosen from an (n, k) code C according to the*

uniform distribution. Let $Y(\epsilon)$ be obtained by transmitting W through a BEC(ϵ). Let

$$Y_{\sim i}(\epsilon) = (Y_1(\epsilon), \cdots, Y_{i-1}(\epsilon), ?, Y_{i+1}(\epsilon), \cdots, Y_n(\epsilon))$$

be obtained by erasing the i-th bit from $Y(\epsilon)$. The i-th EXIT function of C is defined as

$$h_i(\epsilon) = H(W_i|Y_{\sim i}(\epsilon))$$

The average EXIT function is

$$h(\epsilon) = \frac{1}{n}\sum_{i=1}^{n} h_i(\epsilon)$$

The area theorem states that

Theorem 6 (Area Theorem)**.** *The average EXIT function of a binary code of rate R satisfies the following property*

$$R = \int_0^1 h(\epsilon)d\epsilon.$$

Let g be a decoder acting on $Y(\epsilon)$. Then the output bit error rate associated to g can be defined as

$$p_b^g(\epsilon) = \frac{\mathbf{E}[d(W, g(Y(\epsilon))]}{n}$$

where the expectation is taken w.r.t to both the input distribution and channel realizations at erasure probability ϵ. By Proposition 1, the MAP decoder g^* either fully recovers a bit or leaves it completely unbiased. Thus the i-th EXIT function can be written as

$$H(W_i|Y_{\sim i}(\epsilon)) = H(W_i|Y_{\sim i}(\epsilon), g_i^*(Y_{\sim i}(\epsilon))) = \mathbf{P}(g_i^*(Y_{\sim i}(\epsilon)) =?).$$

This gives

$$p_b^{g^*}(\epsilon) = \frac{1}{2n}\sum_i \epsilon \mathbf{P}(g_i^*(Y_{\sim i}(\epsilon)) =?) = \frac{\epsilon h(\epsilon)}{2}. \tag{3.1}$$

Let us now find the implications of the area theorem for the input BER of linear systematic codes. To this end we define the average systematic EXIT function

$$h^{\text{sys}}(\epsilon) = \frac{1}{k}\sum_{i=1}^{k} h_i(\epsilon).$$

Likewise we can define the non-systematic EXIT function as follows:

$$h^{\text{non-sys}}(\epsilon) = \frac{1}{n-k}\sum_{i=k+1}^{n} h_i(\epsilon).$$

3.2 Behavior of BER vs EXIT function

We first prove a lemma to show that the coded bit error rate converges to 0 continuously as the input bit error rate vanishes.

Lemma 7 (Data BER vs EXIT function)**.** *Fix $\epsilon < \epsilon_0$. For any binary linear code of rate R, we have*

$$h(\epsilon) \leq \frac{2R}{\epsilon_0 - \epsilon}\text{BER}(\epsilon_0).$$

In particular, if $BER(\epsilon_0) \to 0$ for a sequence of linear codes, then $h(\epsilon) \to 0$ for all $\epsilon < \epsilon_0$.

Proof. Let X be an input codeword $X \in \{0,1\}^n$ and denote by $Z(\epsilon)$ and $Z(\epsilon_0)$ outputs of degraded binary erasure channels, i.e.:

$$X \to Z(\epsilon) \to Z(\epsilon_0)\,.$$

Notice that

$$I(X_i; Z(\epsilon_0)|Z_{\sim i}(\epsilon)) = I(X_i; Z_i(\epsilon_0)|Z_{\sim i}(\epsilon)) = (1-\epsilon_0)H(X_i|Z_{\sim i}(\epsilon))\,,$$

where the first equality follows from degradation and the second is a property of erasure channels. Rewriting this identity and summing over i we obtain

$$\sum_{i=1}^{n} H(X_i|Z_{\sim i}(\epsilon), Z(\epsilon_0)) = \epsilon_0 \sum_{i=1}^{n} H(X_i|Z_{\sim i}(\epsilon)) = \epsilon_0 n h(\epsilon)\,, \tag{3.2}$$

where $h(\cdot)$ is an EXIT function of the code X.

We now interpret the left-hand side sum in (3.2) as another EXIT function (a conditional one). Indeed, given $Z(\epsilon_0)$ denote by T_0 the set of erasures in $Z(\epsilon_0)$. Conditioned on $Z(\epsilon_0) = z_0$ we have that the joint distribution $P_{X,Z(\epsilon)|Z(\epsilon_0)=z_0}$ can be understood as follows: X_{T_0} is sampled from the distribution $P_{X_{T_0}|X_{T_0^c}}$ and then each of the $|T_0|$ entries of X_{T_0} is erased independently with probability $\omega = \frac{\epsilon}{\epsilon_0}$. Denote by $h^0(\omega; z_0)$ the EXIT function of the code X_{T_0} (note that this is a random function, dependent on values of z_0 on a set T_0^c). This discussion implies

$$h^0(\omega; z_0) = \frac{1}{|T_0|} \sum_{i=1}^{n} H(X_i|Z_{\sim i}(\epsilon), Z(\epsilon_0) = z_0) \tag{3.3}$$

(note that terms corresponding to $i \notin T_0$ are zero.) From the area theorem and monotonicity of the EXIT function we obtain

$$h^0(\omega; z_0)(1-\omega) \leq \frac{1}{|T_0|} H(X|Z(\epsilon_0) = z_0)\,, \tag{3.4}$$

where the right-hand side is an effective rate of the code. In all, from (3.2)-(3.4) we obtain (after taking expectation over z_0)

$$n h(\epsilon) \leq \frac{1}{\epsilon_0 - \epsilon} H(X|Z(\epsilon_0))\,. \tag{3.5}$$

So far we have not used the fact that the code is binary, but now we will. Let $k(T_0) \leq nR$ be the number of unrecoverable information bits given a set T_0 of erasures. Notice that

$$H(X|Z(\epsilon_0) = z_0) \leq k(T_0)\,,$$

and thus taking the expectation, we obtain

$$H(X|Z(\epsilon_0)) \leq \mathbf{E}[k(T_0)] = 2nR \times \mathrm{BER}(\epsilon_0)\,.$$

Together with (3.5) this completes the proof.

■

3.3 Converse

Proposition 8. *Let $\epsilon_2 < \epsilon_1$. For any binary code with $\mathrm{BER}(\epsilon_2) \leq \delta_2$ we have*

$$\mathrm{BER}(\epsilon_1) \geq \sup_{\{\epsilon_0:\epsilon_0<\epsilon_2\}} \frac{\epsilon_1}{2R}\left(\frac{1}{(\epsilon_1-\epsilon_0)}(R-(1-\epsilon_1)-\epsilon_0\frac{2\delta_2(\epsilon_0/\epsilon_2)R}{\epsilon_2-\epsilon_0})-1+R\right)$$

In particular, if $\mathrm{BER}(\epsilon_2) = o(1)$, then

$$\mathrm{BER}(\epsilon_1) \geq \frac{\epsilon_1}{2R}\left(\frac{1}{(\epsilon_1-\epsilon_2)}(R-(1-\epsilon_1))-1+R\right)+o(1)$$

Proof. To prove the lower bound on $h(\epsilon_2)$, we may approximate $h(\epsilon_1)$ in a worst-cast fashion as a piece-wise constant function. To do this, note that $h(\epsilon) \leq h(\epsilon_2)$ for all $\epsilon \leq \epsilon_2$, and $h(\epsilon) \leq h(\epsilon_1)$ for all $\epsilon \in (\epsilon_2, \epsilon_1]$, and $h(\epsilon) \leq 1$ for all $\epsilon > \epsilon_1$. Then the area theorem gives that

$$1-\epsilon_1+h(\epsilon_1)(\epsilon_1-\epsilon_2)+h(\epsilon_2)\epsilon_2 \geq R$$

We note that

$$h(\epsilon) = Rh^{\mathrm{sys}}(\epsilon)+(1-R)h^{\mathrm{non-sys}}(\epsilon)$$

Using the above two relations, we have

$$Rh^{\mathrm{sys}}(\epsilon_1) \geq \frac{1}{\epsilon_1-\epsilon_2}(R-(1-\epsilon_1)-h(\epsilon_2)\epsilon_2)-(1-R)h^{\mathrm{non-sys}}(\epsilon_1)$$

Using $h^{\text{non-sys}} \leq 1$, we get

$$h^{\text{sys}}(\epsilon_1) \geq \frac{1}{R(\epsilon_1 - \epsilon_2)}(R - (1 - \epsilon_1) - h(\epsilon_2)\epsilon_2) - (\frac{1}{R} - 1)$$

If $\text{BER}(\epsilon_2) \to 0$ then $h(\epsilon_2') \to 0$ for any $\epsilon_2' < \epsilon_2$ by Lemma 7. In this case, we can write

$$h^{\text{sys}}(\epsilon_1) \geq \frac{1}{R(\epsilon_1 - \epsilon_2')}(R - (1 - \epsilon_1)) - (\frac{1}{R} - 1)$$

Since $\epsilon_1 > \epsilon_2$, the right hand is continuous for all $\epsilon_2' < \epsilon_2$. Thus we may take the limit as $\epsilon_2' \to \epsilon_2$ to obtain the desired result.

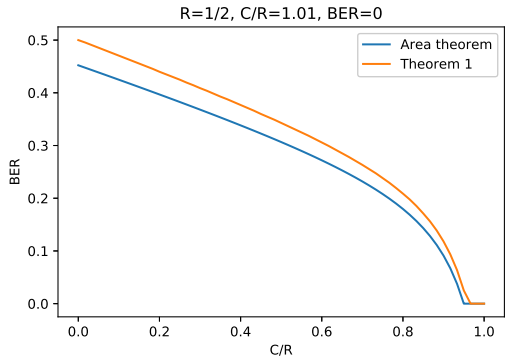
The bounds on BER follow from Lemma 1 and the above two inequalities upon noticing that for a linear systematic code

$$\text{BER}(\epsilon) = \frac{\epsilon h^{\text{sys}}(\epsilon)}{2}.$$

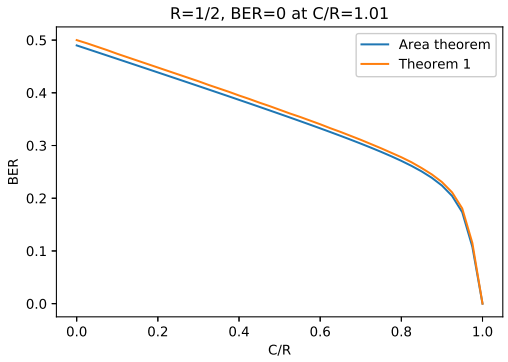
■

3.4 Comparing the bounds

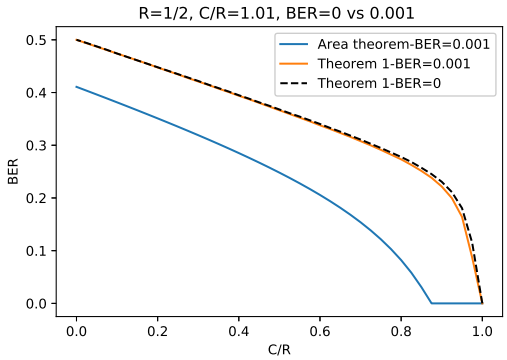
The above bound is compared with that of Theorem 5 in Fig.3-1. It can be seen that the former bound is tighter and more stable.



(a) Lower bounds for codes with BER(0.475) = 0



(b) Lower bounds for codes with BER(0.495) = 0



(c) Stability of the bounds around BER(0.495) = 0

Figure 3-1: Comparing the lower bounds of Theorem 5 and Proposition 8 for linear systematic codes of rate 1/2 satisfying a) BER(ϵ) = 0 at $\epsilon = 0.475$ b) BER(ϵ) = 0 at $\epsilon = 0.495$ c) BER(ϵ) = 0 vs BER(ϵ) = 0.001 at $\epsilon = 0.495$. We note that the bounds from Theorem 5 are tighter and more stable as BER moves away from 0.

Chapter 4

Analysis of Belief Propogation

In this chapter we provide tools to study the error dynamics under BP for general codes and apply them to bound the BER of LDMC(3). The same tools can be used to bound the error under the optimal (bitwise MAP) decoder as well. Our analysis shows that for LDMC(3) the gap between BP and optimal decoder is small.

4.1 Review of BP

We recall the notion of a code ensemble generated by a Boolean function $f : \{0,1\}^m \to \{0,1\}$ from Chapter 1.3. We also briefly review the notion of a (bipartite) factor graph associated with a code from the ensemble (cf. [65], Chapter 2). Consider a code defined on $\{0,1\}^k$. To every coordinate $i \in [k]$, we associate a *variable node* and represent it by a circle. We further associate random variables $X_i \overset{i.i.d}{\sim} \text{Ber}(1/2)$ to the variable nodes. Likewise, to every subset $\Delta_j \in \Delta$, we associate a *check node* and represent it by a square. Every such node represents a constraint of the form $y_j = f(X_{\Delta_j})$, where y_j's are the realized (unerased) coded bits and X_{Δ_j} is the restriction of X to the coordinates in Δ_j. We connect a variable node i to a check node Δ_j if and only if $i \in \Delta_j$ (see Fig. 4-1a). We remark that most references associate a separate node with y_j's to model the channel likelihoods [41, 65, 78]. In the language of [41], our description is a cross section of the full factor graph parametrized by y_j's. We do not make this distinction in the sequel as our primary interest is to analyze the

decoding error for erasure noise. In this case, we can simply restrict to the sub-graph associated with the observed bits and do not need to consider the channel likelihoods.

Given a target bit X_i, the decoding problem is to estimate (or approximate) the marginal probabilities $p_{X_i|Y}(\cdot|y)$ for a realization y of the (observed) coded bits. Here we denote such an estimate by the function π_{X_i} and refer to it as a message. A message should be thought of as an approximation to the true marginal computed by the decoder. To study the behavior of iterative decoding methods, it is helpful to consider the notion of a local neighborhood. Given a target bit X_i, we denote by $\Delta(i)$ the set of its neighbor nodes among the factors, that is, the set of check nodes whose constraint involves X_i. We further define the local neighborhood $\partial(i)$ among the variable nodes to be the set of variables (other than i) that appear in $\Delta(i)$ (see Fig. 4-1b). Given a vector $v \in \{0,1\}^k$, we define $\partial v_i := v_{\partial(i)}$ to be the restriction of v to the coordinates in $\partial(i)$. Likewise, if $v \in \{0,1\}^n$, then $\Delta v_i := v_{\Delta(i)}$ denotes the restriction of v to $\Delta(i)$. The j-th node in $\Delta(i)$ is denoted by $\Delta_j(i)$. The variable nodes other than i that are connected to Δ_j are denoted by $\partial_j(i)$. Similarly, we define the j-th order local neighborhood $\partial^j(i)$ by recursively unfolding the local neighborhoods at the boundary $\partial^{j-1}(i) := \partial(\partial^{j-1}(i)) - \partial^{j-1}(i)$. In other words, the ℓ-th order boundary is the set of nodes (not in $\partial^{j-1}(i)$) that are in the local neighborhood of $\partial^{j-1}(i)$. Likewise, $\Delta^j(i) := \Delta(\partial^{j-1}(i)) - \Delta^{j-1}(i)$ (see Fig. 4-2a). The compliment of $\Delta(i)$ inside Δ is denoted by $\Delta^{\sim}(i)$. Finally, we define $\Delta^{(j)}(0) := \cup_{i=1}^{j} \Delta^i(0)$.

With this notation we can describe a generic iterative algorithm to compute π_{X_i}. Let $\pi_{\partial X_0}$ be the message (or approximation) for $p_{\partial X_0|\Delta^{\sim} Y_0}$. This is the conditional estimate of the random variables in the local neighborhood of X_0 given all the observed bits outside the neighborhood. By d-separation, the computation of the marginals for X_0 can be decomposed as follows:

$$\pi_{X_0}(x_0) = \sum_{\partial x_0} \pi_{\partial X_0}(\partial x_0) p_{X_0|\partial X_0}(x_0|\partial x_0) \propto \sum_{\partial x_0} \pi_{\partial X_0}(\partial x_0) \prod_{\Delta_j \in \Delta(0)} \mathbb{1}_{\{y_j = f(x_0, \partial_j x_0)\}}. \tag{4.1}$$

In this way, we obtain an iterative procedure where the messages $\pi_{\partial X_0}$ flow into the local neighborhood and the posterior estimates π_{X_0} flow out to the target node (see

Fig. 4-1b). To iterate such a procedure ℓ-times, one needs to first approximate the marginals at the ℓ-th order boundary. Once this is done, (4.1) can be applied iteratively to compute π_{X_0}. The factor (sub-)graph obtained after ℓ unfoldings represents the natural order of recursive computations needed to compute π_{X_0}, and hence, we refer to it as a *computational graph*. Fig. 4-2 shows the case where the computational graph is a tree.

Belief propagation (BP) is a special case of such iterative procedure where the input messages are assumed to factorize into a product:

$$\pi_{\partial X_0} = \prod_i \pi_{\partial_i X_0}$$

The number of iterations of BP determine the depth of the computational graph, i.e., the order of the local neighborhood on which we condition. We denote by π^ℓ the message corresponding to $p_{X_0|\Delta^{(\ell)}Y_0}$. This is the approximate marginal given observations revealed in the computational graph of depth ℓ. After ℓ iterations, the marginals under BP can be written more efficiently (compared with (4.1)) as

$$\pi^\ell_{X_0}(x_0) \propto \prod_{\Delta_j \in \Delta(0)} \sum_{\partial_j x_0} \pi^{\ell-1}_{\partial_j X_0}(\partial_j x_0) \mathbb{1}_{\{y_j = f(x_0, \partial_j x_0)\}}, \tag{4.2}$$

with the initial conditions $\pi^0_{X_i}(0) = \pi^0_{X_i}(1) = 1/2$ for all bits.

It can be checked that when the computational graph is a tree, BP is exact, i.e., it computes the correct marginals $p_{X_0|\Delta^{(\ell)}Y_0}$ given the observations in the depth ℓ graph. We also refer to the correct marginal $p_{X_0|Y}$ as the (bitwise) MAP estimate of X_0. When the computational graph is a tree, the only difference between MAP and BP estimates is the input messages into the ℓ-th order local neighborhood. In other words, if the initial messages along the boundary are the correct marginals $p_{\partial^l X_0|\Delta^{\sim(\ell)}Y_0}$, then BP iterations recover the (bitwise) MAP estimate. Here $\Delta^{\sim(\ell)}$ is the set of check nodes in Δ that do no appear in the computational tree of depth ℓ.

4.2 E-functions

We recall that, in general, a computational graph of small depth ($o(\log(k))$) corresponding to a (check-regular) code ensemble is with high probability a tree (cf. [65], Exercise 3.25). For such ensembles, we want to study the dynamics of the decoding error along the iterations of BP. Hence, we need to understand how the error flows in and out of the local neighborhood of a target node. In other words, we want to understand how the BP dynamics contracts the input error.

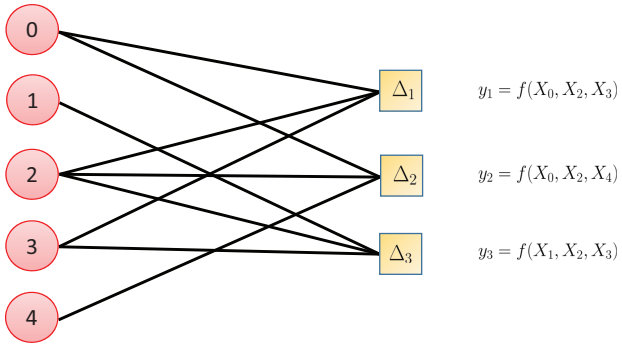
We define E-functions for this purpose. They can be viewed as a mapping of the input error (at the beginning of a decoding iteration) to the output error (at the end of the iteration). There are two types of E-functions studied in this work: the erasure functions and the error functions.

Definition 10 (Erasure function)**.** *Consider a code ensemble generated by a Boolean function* $f : \{0,1\}^m \to \{0,1\}$ *with variable node degrees sampled from* Deg*. Fix* $\alpha = C/R$ *and consider a computational tree of depth 1 as in Fig. 4-2b corresponding to the target bit* X_0*. Let* $M_j = f(X_0, X^{(j)})$*,* $j = 1, \cdots, d$*, where* $X^{(j)} \sim \mathrm{Ber}(1/2)^{\otimes(m-1)}$ *are the boundary nodes. Suppose that each boundary node is observed through a (memoryless)* BEC *channel, i.e.,* $Y^{(j)} = \mathrm{BEC}_{\bar{q}}(X^{(j)})$ *where* $\bar{q} = 1 - q$ *is the probability of error. The function*

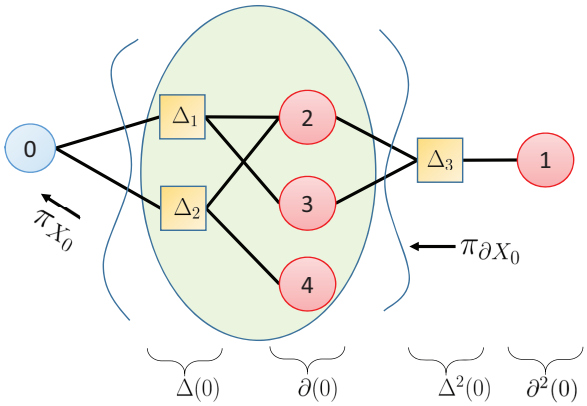
$$E_d^{\mathrm{BEC}}(q) = \mathbf{E}[\mathbf{P}(X_0 = 1 | M_1, \cdots, M_d, Y^{(1)}, \cdots, Y^{(d)}) | X_0 = 0]$$

is called the d-th erasure polynomial of the ensemble. Here the expectation is taken with respect to the ensemble distribution as well the randomization over bits. The erasure function is defined as

$$E^{\mathrm{BEC}}(\alpha, q) = \sum_k \mathbf{P}(\mathrm{Deg} = k) E_k^{\mathrm{BEC}}(q).$$



(a) Factor graph representation of the (observed) equations



(b) Local neighborhood of 0 in the unfolded factor graph

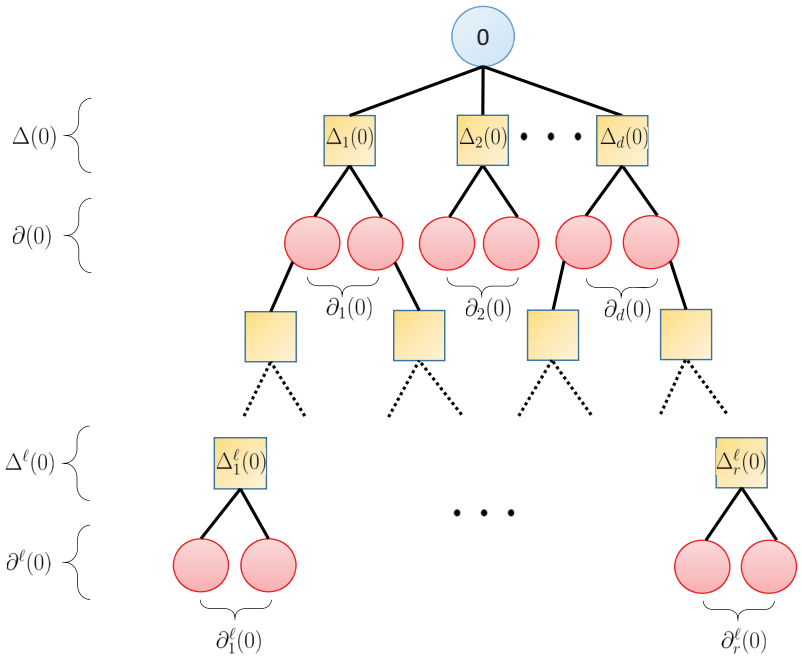
Figure 4-1: The factor graph of a code and the local neighborhood of a target node are shown. a) The check nodes correspond to observed (unerased) coded bits and represent the constraints imposed by such observations. b) The factor graph can be unfolded with respect to a target node. The immediate (variable) neighbors of the target nodes in such unfolding form its local neighborhood. A recursive algorithm can first estimate the marginal probabilities $\pi_{\partial X_0}$ for the local neighborhood and then compute the posterior π_{X_0} using (4.1). Here we recall that $\partial X_0 = X_{\partial(0)}$.

The d-th truncated easure polynomial is

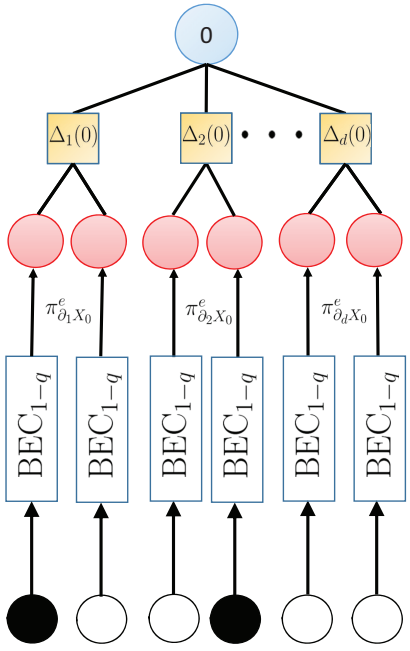
$$E_{\leq d}^{\mathrm{BEC}}(\alpha, q) = \sum_{k \leq d} \mathbf{P}(\mathrm{Deg} = k) E_k^{\mathrm{BEC}}(q).$$

Similarly, we can define the notion of an error function.

Definition 11 (Error function)**.** *In the setup of* Definition 10*, let* $Y^{(j)} = \mathrm{BSC}_q(X^{(j)})$ *be the result of passing* $X^{(j)}$ *through a (memoryless)* BSC *channel with crossover*



(a) Computational tree of depth ℓ for a (check-regular) ensemble of degree 3



(b) Local neighborhood of a variable node with BEC inputs

Figure 4-2: a) A computational tree for a (check-regular) ensemble of degree 3 obtained after ℓ unfoldings w.r.t a target node along with the (local) indexing used in the analysis of BP. We refer to $\partial^j(0)$ as the j-th order neighborhood of 0 and $\partial^\ell(0)$ as the boundary of the tree. b) The local neighborhood of the target node with leaves observed through BEC channels. This local graph is used to define the erasure function.

probability q. The function

$$E_d^{\mathrm{BSC}}(q) = \mathbf{E}[\mathbf{P}(X_0 = 1 | M_1, \cdots, M_d, Y^{(1)}, \cdots, Y^{(d)}) | X_0 = 0]$$

is called the d-th error polynomial of the ensemble. Likewise, the error function is defined as

$$E^{\mathrm{BSC}}(\alpha, q) = \sum_k \mathbf{P}(\mathrm{Deg} = k) E_k^{\mathrm{BSC}}(q)$$

The d-th truncated error polynomial is

$$E_{\leq d}^{\mathrm{BSC}}(\alpha, q) = \sum_{k \leq d} \mathbf{P}(\mathrm{Deg} = k) E_k^{\mathrm{BSC}}(q) + \frac{1}{2} \sum_{k > d} \mathbf{P}(\mathrm{Deg} = k)$$

Remark 2. We briefly discuss the effect of truncating the E-functions here. Clearly $E^{\mathrm{BEC}} \geq E_{\leq d}^{\mathrm{BEC}}$ holds pointwise since we drop some non-negative terms from E^{BEC} to obtain $E_{\leq d}^{\mathrm{BEC}}$. Likewise $E^{\mathrm{BSC}} \leq E_{\leq d}^{\mathrm{BSC}}$ since we assume all the high degree nodes are in error when computing $E_{\leq d}^{\mathrm{BSC}}$. In fact, due to monotonicity, a better upper bound on E^{BSC} would be

$$E_{\leq d}^{\mathrm{BSC}}(\alpha, q) \leq \sum_{k \leq d} \mathbf{P}(\mathrm{Deg} = k) E_k^{\mathrm{BSC}}(q) + E_{d+1}^{\mathrm{BSC}}(q) \sum_{k > d} \mathbf{P}(\mathrm{Deg} = k).$$

In practice, we choose the truncation degree to be large enough that makes this adjustment not so crucial.

Remark 3. For linear codes, iterative decoding is often studied in terms of the input-output entropy or the so called EXIT charts [65] instead of error probability. For linear codes, the two methods are equivalent as the EXIT function is proportional to the probability of error. For general codes, however, we would need to invoke a Fano type inequality to relate the two and this step is often lossy. For instance, in the case of LDMCs, we can obtain much better bounds by analyzing the probability of error directly.

4.3 Bounds via comparison lemmas

The motivation to compute the E-functions comes from various comparison lemmas in information theory. The idea is to approximate the incoming messages to the local neighborhoods of BP as if they were induced by simpler to analyze (BEC or BSC) channels, while preserving certain properties of the inputs. Then we carry out the local contraction analysis on these simpler message and apply comparison lemmas to control the error dynamics for the original inputs. To this end, we define some partial orders on the space of channels with common input alphabets.

Definition 12 ([21, Chapter 5.6]). *Given two channels $P_{Y|X}$ and $P_{Y'|X}$ with common input alphabet, we say that $P_{Y'|X}$ is*

- less noisy *than $P_{Y|X}$, denoted by $P_{Y|X} \preceq_{\text{l.n.}} P_{Y'|X}$, if for all joint distributions P_{UX} we have*
$$I(U;Y) \le I(U;Y')$$

- more capable *than $P_{Y|X}$, denoted by $P_{Y|X} \preceq_{\text{m.c.}} P_{Y'|X}$, if for all marginal distributions P_X we have*
$$I(X;Y) \le I(X;Y').$$

- less degraded *than $P_{Y|X}$, denoted by $P_{Y|X} \preceq_{\text{deg}} P_{Y'|X}$, if there exists a Markov chain $Y - Y' - X$.*

We refer to [49, Sections I.B, II.A] and [62, Section 6] for alternative useful characterizations of the less-noisy order. In particular, it is known (cf. [62, Proposition 14],[38]) that

$$P_{Y|X} \preceq_{\text{l.n.}} P_{Y'|X} \iff D(P_Y\|Q_Y) \le D(P_{Y'}\|Q_{Y'}) \tag{4.3}$$

where the output distributions correspond to common priors P_X, Q_X. The following implications are easy to check

$$P_{Y|X} \le_{\text{deg}} P_{Y'|X} \implies P_{Y|X} \le_{\text{l.n.}} P_{Y'|X} \implies P_{Y|X} \le_{\text{m.c.}} P_{Y'|X}.$$

Proposition 9. *Consider the dynamical system*

$$q_{t+1}^{\mathrm{BEC}}(x_0) = 1 - 2E_{\leq d}^{\mathrm{BEC}}(\alpha, q_t^{\mathrm{BEC}}) \tag{4.4}$$

initialized at $q_0^{\mathrm{BEC}} = x_0$ *with* $\alpha = C/R$. *Similarly, define*

$$q_{t+1}^{\mathrm{BSC}}(x_0) = E_{\leq d}^{\mathrm{BSC}}(\alpha, q_t^{\mathrm{BSC}}) \tag{4.5}$$

with $q_0^{\mathrm{BSC}} = x_0$. *Let* $\delta_\ell^{\mathrm{BP}}$ *be the* BER *of a (check regular) ensemble under* BP *after* ℓ *iterations. Likewise, let* δ^{MAP} *be the* BER *under the optimal (bitwise* MAP*) decoder. Then*

$$\frac{1 - q_\ell^{\mathrm{BEC}}(1)}{2} - o(1) \leq \delta^{\mathrm{MAP}} \leq \delta_\ell^{\mathrm{BP}}.$$

Furthermore,

$$\frac{1 - q_\ell^{\mathrm{BEC}}(0)}{2} - o(1) \leq \delta_\ell^{\mathrm{BP}} \leq q_\ell^{\mathrm{BSC}}(1/2) + o(1)$$

with $o(1) \to 0$ *and* $k \to \infty$.

To prove the proposition, we need several definitions and two lemmas.

Definition 13 (BMS [65, Chapter 4.1])**.** *Let* W *be a memoryless channel with binary input alphabet* $\mathcal{X}$ *and output alphabet* $\mathcal{Y}$. *Let the two element cyclic group act on* $\mathcal{X}$ *and* $\mathcal{Y}$. *Denote by* $-$ *the action of its generator (transposition). We say that* W *is a binary memoryless symmetric channel (*BMS*) if it is invariant under* $-$*, i.e., if* $W(y|x) = W(-y| - x)$ *for all* $y \in \mathcal{Y}$.

We also define the total variation distance (TV) and χ^2-divergence between two probability measures P and Q as follows:

$$(P, Q) \triangleq \frac{1}{2} \int |dP - dQ|,$$
$$\chi^2(P, Q) \triangleq \int (\frac{dP}{dQ})^2 dQ - 1.$$

Lemma 10. *Let W be a* BMS *channel and define its probability of error, capacity and χ^2-capacity as follows*

$$P_e(W) = \frac{1 - (W(\cdot|0), W(\cdot|1))}{2}, \tag{4.6}$$

$$C(W) = D(W(\cdot|0)\|P_Y), \qquad P_Y = \frac{1}{2}(W(\cdot|0) + W(\cdot|1)) \tag{4.7}$$

$$I_{\chi^2}(W) = \chi^2(W(\cdot|0)\|P_Y)\,. \tag{4.8}$$

The following holds:

1. *Among all* BMS *channels with the same value of $P_e(W)$ the least degraded is* BEC *and the most degraded is* BSC*, i.e.*

$$\mathrm{BSC}_\delta \preceq_{deg} W \preceq_{deg} \mathrm{BEC}_{2\delta}\,, \tag{4.9}$$

   where $\preceq_{deg}$ denotes the (output) degradation order.

2. *Among all* BMS *with the same capacity C the most capable is* BEC *and the least capable is* BSC*, i.e.:*

$$\mathrm{BSC}_{1-h^{-1}(C)} \preceq_{mc} W \preceq_{mc} \mathrm{BEC}_{1-C}\,, \tag{4.10}$$

   where $\preceq_{mc}$ denotes the more-capable order, and $h^{-1} : [0,1] \to [0,1/2]$ is the functional inverse of the (base-2) binary entropy function $h : [0,1/2] \to [0,1]$.

3. *Among all* BMS *channels with the same value of χ^2-capacity $\eta = I_{\chi^2}(W)$ the least noisy is* BEC *and the most noisy is* BSC*, i.e.*

$$\mathrm{BSC}_{1/2-\sqrt{\eta}/2} \preceq_{ln} W \preceq_{ln} \mathrm{BEC}_{1-\eta}\,, \tag{4.11}$$

   where $\preceq_{ln}$ denotes the less-noisy order.

The next lemma states that if the incoming messages to BP are comparable, then the output messages are comparable as well.

Lemma 11. *Fix some random transformation $P_{Y|X_0,X_1^m}$ and m* BMS *channels $W_1, ..., W_m$. Let $W : X_0 \mapsto (Y, Y_1^m)$ be a (possibly non-*BMS*) channel defined as follows. First, $X_1, ..., X_m$ are generated as i.i.d* Ber$(1/2)$*. Second, each Y_j is generated as an observation of X_j over the W_j, i.e. $Y_j = W_j(X_j)$ (observations are all conditionally independent given X_1^m). Finally, Y is generated from all X_0, X_1^m via $P_{Y|X,X_1^m}$ (conditionally independent of Y_1^m given X_1^m). Define the $\tilde{W}$ channel similarly, but with W_j's replaced with $\tilde{W}_j$'s. The following statements hold:*

1. *If $\tilde{W}_j \preceq_{deg} W_j$ then $\tilde{W} \preceq_{deg} W$*
2. *If $\tilde{W}_j \preceq_{ln} W_j$ then $\tilde{W} \preceq_{ln} W$*

Remark 4. An analogous statement for more capable channels does not hold. To this see, let $Y = X + X_1 + X_2$ be a parity constraint. Then the channel $X \mapsto (0, Y_1, Y_2)$ is equivalent to $U \mapsto (Y_1, Y_2)$ in the setting of Example 2 in §4.4.

The lemmas are proved in Appendix A.

Proof of Prop. 9. We sample codes from the family and consider the (local) computational graph of a fixed bit X_0 with depth ℓ. It is known that for large codes, the computational graph of depth ℓ has a tree structure with high probability. Hence, we may assume that the graph is a tree.

Consider the depth ℓ tree emanating from X_0. The channel

$$T_\ell : X_0 \mapsto (\text{computational tree of depth } \ell, \Delta^{(\ell)} Y_0)$$

is a BMS (recall that $\Delta^{(\ell)} Y_0 = Y_{\Delta^{(\iota)}(0)}$ denotes all the coded bits observed in the tree of depth ℓ). We note that running ℓ-steps of BP is equivalent to decoding X_0 from the output of T_ℓ. In other words $\delta^{\text{BP}} = P_e(T_\ell)$ is the error we want to bound. Further note that the structure of the tree is included as part of the channel, so that $P_e(T_\ell)$ is computed by randomizing over possible realizations of the graph as well.

Now condition on the first layer of the tree. If the number of variable nodes in $\partial(0)$ is m, then the restriction of T_ℓ to the first layer has the structure of Lemma 11. Indeed for each choice of $X_0 = x_0, \partial X_0 = \partial x_0$, $P_{\Delta Y_0|X_0,\partial X_0}$ simply indicates whether or not

all the constraints in the local neighborhood are satisfied: $P_{\Delta Y_0|X_0,\partial X_0}(\Delta y_0|x_0,\partial x_0) = \prod_{j\in\Delta(0)} \mathbb{1}_{\{y_j=f(x_0,\partial_j x_0)\}}$. Furthermore, if we set $W_j = T_{\ell-1}$ to be the corresponding tree channel emanating from X_j's (with $j \in \partial(0)$), then due to the locally tree assumption their observations are independent.

Now assume by induction that $T_{\ell-1} \preceq_{\deg} \mathrm{BEC}_{\bar{q}_{\ell-1}}$. Then by Lemma 11, we have $T_\ell \preceq_{\deg} \tilde{T}_\ell$ where $\tilde{T}_\ell$ is the tree of depth ℓ in which the channels W_j are replaced with $\mathrm{BEC}_{\bar{q}_{\ell-1}}$. Note that if we condition on the degree d of X_0, then the $\tilde{T}$ channel has error $E_d^{\mathrm{BEC}}(\alpha, q_{\ell-1})$. By averaging over the degrees, we obtain

$$P_e(\tilde{T}_\ell) = E^{\mathrm{BEC}}(\alpha, q_{\ell-1}) \geq E_{\leq d}^{\mathrm{BEC}}(\alpha, q_{\ell-1}) = \bar{q}_\ell/2,$$

where the inequality is due to truncation (recall that in $E_{\leq d}^{\mathrm{BEC}}$ all nodes of degree larger than d are assumed to have zero error–see Remark 2). To complete the induction step, note that $\tilde{T}_\ell \preceq_{\deg} \mathrm{BEC}_{\bar{q}_\ell}$ by Lemma 10. We thus have $P_e(T_\ell) \geq \bar{q}_\ell/2$ as desired.

The proof of the BSC upper bound is obtained in a similar manner after replacing the input channels to $\tilde{T}_\ell$ with BSCs and invoking the reverse sides of Lemmas 11,10 again.

Finally, BP and MAP decoding differ only by the initialization of beliefs at the leaf nodes. Since the MAP channel at the leaves is a degradation of BEC_0, the lower bound on MAP follows as well.

■

4.4 A counter-example

A counter-example is presented in [19, Problem 15.11] to show that the less noisy property is strictly stronger than less capable. The example is instructive but involves non-BMS channels. Here we give a more natural counter-example (from a coding-theoretic perspective) using the parity function and BSC/BEC channels. The purpose of the example is to show that some coded bits may be easier to recover from a less capable channel. It also serves to show that an analogous statement of Lemma 11

above for more capable channels does not hold.

Example 2. By Lemma 10 below, if $\epsilon \leq h(\delta)$, then BEC_ϵ is more capable than BSC_δ and if $\epsilon \leq 1-(1-2\delta)^2$ then BEC_ϵ is less noisy than BSC_δ. We show that for some $\epsilon \leq h(\delta)$, the BEC_ϵ channel is not less noisy by giving an explicit construction.

Let X_1, X_2 be independent $\mathrm{Ber}(1/2)$ random variables. Let $U = \mathrm{XOR}(X_1, X_2)$ be their parity and $\tilde{Y}_i = \mathrm{BEC}_\epsilon(X_i)$, $Y_i = \mathrm{BSC}_\delta(X_i)$ be their observations. By [19, Problem 6.18], the property of being more capable tensorizes, as does that of being less noisy [62, Proposition 16], [73, Proposition 5]. It thus suffices to show that for some $\epsilon \leq h(\delta)$ we have

$$I(U; Y_1, Y_2) > I(U; \tilde{Y}_1, \tilde{Y}_2).$$

We denote by N_δ the number of flips (resp. by N_ϵ the number of erasures) in the BSC (resp. BEC) channel. It follows that

$$I(U; Y_1, Y_2) = 1 - h(\mathbb{P}(\{N_\delta \text{ is even}\}) = 1 - h(\delta^2 + (1-\delta)^2)$$

while

$$I(U; \tilde{Y}_1, \tilde{Y}_2) = \mathbb{P}(\{N_\epsilon = 0\}) = (1-\epsilon)^2.$$

We can easily check that the inequality

$$(1-\epsilon)^2 < 1 - h(\delta^2 + (1-\delta)^2)$$

holds for $\epsilon \in (1-\sqrt{1-h(\delta^2+(1-\delta)^2)}, h(\delta)]$. We note that the interval is non-empty since $\delta^2 + (1-\delta)^2 > \delta$ for all $\delta < 1/2$.

In fact, parity bits can become noisier as soon as BEC_ϵ loses its less noisy property. The next example illustrates this point.

Example 3. Let $\tilde{Y}_i = \mathrm{BEC}_\epsilon(X_i)$ and $Y_i = \mathrm{BSC}_\delta(X_i)$. Set $\delta := \lambda/n$ and $\epsilon := \lambda'/n$. Then by Lemma 10 below, if $\lambda' < 4\lambda - \tau$, then $\mathrm{BEC}_{\lambda'/n}$ is less noisy than $\mathrm{BSC}_{\lambda/n}$ for large enough n and $\tau > 0$. We show that for some $\lambda' \geq 4(1+\mu)\lambda$ with $\mu > 0$, the channel $\mathrm{BEC}_{\lambda'/n}$ is asymptotically (for large n) more capable but not less noisy than

$\mathrm{BSC}_{\lambda/n}$.

First note that $\lambda'/n < h(\lambda/n)$ holds asymptotically if $\lambda' = c\lambda$ for any fixed c. Indeed we can easily check that $\lim_{x\downarrow 0} h(x)/x = \infty$. Thus for all $\lambda' = c\lambda$, the $\mathrm{BEC}_{\lambda'/n}$ channel is more capable (for large enough n). To see that it is not necessarily less noisy, let $X_i \sim \mathrm{Ber}(1/2)$ be an i.i.d sequence. Let $U_n = \sum_{i=1}^{n} X_i \pmod 2$ be the parity of the first n bits. Again since the two properties tensorize, it suffices to show that for some $\lambda' = c\lambda$ we have

$$\lim_n I(U_n; Y^n) > \lim_n I(U_n; \tilde{Y}^n).$$

To show this, we note that the number N_δ of flips (resp. N_ϵ of erasures) in the BSC (resp. BEC) channel is asymptotically distributed as $\mathrm{Poi}(\lambda)$ (resp. $\mathrm{Poi}(\lambda')$). It follows that

$$\lim_n I(U_n; Y^n) = 1 - \lim_n h(\mathbb{P}(\{N_\delta \text{ is even}\}) = 1 - h(\frac{1}{2} + \frac{1}{2}e^{-2\lambda}).$$

Using the taylor series expansion of the binary entropy function around $1/2$, we have that

$$\lim_n I(U_n; Y^n) = e^{-4\lambda}(\frac{1}{2\ln 2} + O(e^{-\lambda})),$$

whereas

$$\lim_n I(U_n; \tilde{Y}^n) = \lim_n \mathbb{P}(\{N_\epsilon = 0\}) = e^{-\lambda'}.$$

Our claim now reduces to checking that the inequality

$$e^{-4\mu\lambda} < \frac{1}{2\ln 2} + O(e^{-\lambda})$$

holds for some λ and all $\mu > 0$, which follows easily after passing to the limits.

4.5 Computing E-functions for LDMC(3)

In the rest of this section, we provide an algorithm to compute the E-functions for LDMC(3) and use Proposition 9 to obtain upper and lower bounds for BP and

bitwise MAP decoders for this family of codes. The degree distribution of LDMC(3) is asymptotically Poi(3α) distributed where $\alpha = C/R$. In this case, the truncated erasure polynomial is

$$E_{\leq d}^{\mathrm{BEC}}(\alpha, q) = \sum_{k=0}^{d} \mathbf{P}(\mathrm{Poi}(3\alpha) = k) E_k^{\mathrm{BEC}}(q).$$

Computing the erasure polynomials is more involved for LDMC(3) than LDGMs since the BP messages are more complicated. For LDGMs, the messages are trivial in the sense that every uncoded bit remains unbiased after each BP iteration. This does not hold for LDMCs, and it is in fact this very principle that allows BP decoding to initiate for LDMCs without systematic bits. Hence, to analyze BP locally, we need to randomize over all possible realizations of the bits in the local neighborhoods. This is a computationally expensive task in general, but one that can be carried out in some cases by properly taking advantage of the inherent symmetries in the problem.

The BP update rules are easy to derive for LDMCs. Let Y_j be the majority of 3 bits X_0, X_1, X_2. Then if $Y_j = 0$, the check to bit message is

$$m_j = \frac{\mathbf{P}(X_0 = 0 | Y_j = 0)}{\mathbf{P}(X_0 = 1 | Y_j = 0)} = 1 + \frac{1}{r_1} + \frac{1}{r_2}, \tag{4.12}$$

where $r_i = \mathbb{P}(X_i = 0)/\mathbb{P}(X_i = 1)$ are the priors (or input messages to the local neighborhood). The posterior likelihood ratio for X_0 is $r_0 = \prod_{j \in \Delta(0)} m_j$. We now use these update rules to compute the E-polynomials for LDMC(3). In Appendix B, we provide a Python generated list of the erasure polynomials for LDMC(3), which are used in various places throughout this chapter and in §5.2 for code optimization.

Let $\bar{q} = 1 - q$ be the probability of erasure at the boundary. For bits of degree zero, the probability of error is clearly $\frac{1}{2}$ and for bits of degree 1 the probability of error is $\frac{1}{4}$ independent of q. To see this, consider the computational tree of a degree 1 bit X_0 at depth 1. There are two leaf bits in tree. Suppose that neither of the leaf bits is erased. This happens with probability q^2. Conditioned on this, only when the two leaf bits take different values can X_0 be fully recovered and this conditional probability is

$\frac{1}{2}$. Otherwise, the bit remains unbiased and must be guessed randomly. The overall contribution of this configuration to the probability of error for X_0 is $q^2/2$. One other possible configuration is when only one leaf bit is erased. In this case the target bit is determined whenever the unerased bit disagrees with the majority, which happens with probability $\frac{1}{4}$. When the unerased bit agrees with the majority, it weakens the (likelihood ratio) message sent from the majority to the target bit. In this case, the message passing rule in (4.12) shows that the probability of error is $\frac{1}{3}$. Overall, the contribution of this configuration to the probability of error is $2q(1-q)/4$. Finally, if both bits are erased, which happens with probability $(1-q)^2$, then the probability of error is again $\frac{1}{4}$. Adding up all the error terms, we see that $E_1(q) = \frac{1}{4}$. It is true for any monotonic function that $E_1(q)$ is a constant. Indeed if f is monotonic, then the decision rule for estimation of any degree 1 node depends in a deterministic fashion on the value of f and not on the distribution of local beliefs.

The first interesting case where the error probability depends on the erasure probability $1-q$ happens for degree 2 nodes. We work out the computation of $E_2(q)$, which contains the main ideas to compute the full error polynomial.

Example 4 (Computing $E_2(q)$ for LDMC(3)). For a degree 2 bit X_0, there are two majorities connected to the bit, and four leaf bits in the tree. We need to compute what message is sent from each majority to the target bit along each realization and compute the corresponding error probability. Consider the case where all four bits at the leaves are unerased. Since the erasure events are independent, the probability that all four leaves are unerased is q^4 and we incur an error of $1/2$ in recovering X_0 only when all the neighboring leaves agree, i.e., the leaf configuration $\partial^0 X$ takes values in $\{0000, 0011, 1100, 1111\}$. The message sent from each majority to the bit in this case is $\mathbb{P}(x=0)/\mathbb{P}(x=1) = 1$. The leaf configurations above each have a $1/16$ chance of being realized. The overall contribution to the probability of error is $\frac{q^4}{8}$ in this case. When all the bits are erased, the message sent to the target bit is either $1/3$ or 3 by each majority. If the two majorities agree, which happens with probability $5/8$, the messages amplify and give the target bit a 0.9 chance of correct recovery. If they do not, we get two conflicting messages which cancel out each other, and that leaves the

target bit unbiased. Overall, the error in this case is $(5/8 \times 1/10 + 3/8 \times 1/2)(1-q)^4$. When three bits are unerased, we consider the majority with only one unerased bit. The bit that is recovered correctly disagrees with its majority with probability 1/4, in which case the target bit will be recovered correctly. The remaining 3/4 of the time, the message sent upward in the tree to the target bit is 2, giving the bit a 2/3 chance of being recovered. The other majority either determines the bit with probability 1/2 or sends it a message of 1 otherwise. Overall, the error incurred from such configurations is $4q^3(1-q)/8$. Next, consider the case where two bits are unerased. If they belong to the same majority, say Δ_1^0, they can recover the target bit with probability 1/2, hence, the probability of error is $\mathbf{P}(\hat{X}_0 \neq \Delta_2^0)/2 = 1/8$. The contribution to the probability of error from such configurations is $2q^2(1-q)^2/8$. If they belong to different majorities, which happens with probability $4q^2(1-q)^2$, then each majority determines the bit 1/4 of the time independently of the other majority. If neither majority fully recovers the bit, then each majority sends a message of 2 or 1/2 upward. The messages agree with probability 5/9 and disagree with probability 4/9. Hence an error can happen with probability $9/16 \times (1/5 \times 5/9 + 1/2 \times 4/9)$. The total error incurred from this contribution is $9/16 \times (1/5 \times 5/9 + 1/2 \times 4/9) 4q^2(1-q)^2$. Finally, when only 1 bit is unerased, its majority can determine the target bit with probability 1/4. When it does not determine the bit, it sends a message of 2 or 1/2 upward. The other majority sends a message of 3 or 1/3 upward. The two majorities agree with probability 7/12 in which case the message upward is either 6 or 1/6, giving an error of 1/7. If the majorities disagree, the message upward is either 3/2 or 2/3 giving an error of 2/5. Putting things together, the error polynomial as a function of q for a degree 2 bit is:

$$E_2(q) = \frac{q^4}{8} + \frac{q^3(-q+1)}{2} + q^2(-q+1)^2 + \frac{3q(-q+1)^3}{4} + \frac{(-q+1)^4}{4}$$

For the general case, the ideas are the same. Consider the message sent from the a majority check to a target bit modulo inversion. This means that we identify a message m and its inverse $1/m$ as one group of messages. This is a random variable

that depends on the erasure patterns as well as the realized values at the leaves. Let us first condition on the erasure patterns. In this case the message is either in $\{0, \infty\}$, $\{1\}$, $\{2, 1/2\}$, or $\{3, 1/3\}$. In the first case, the conditional error is zero, hence, we assume that one of the latter messages is sent. Let M_i be the message sent from the i-th majority to the target bit modulo inversion. If we represent $\{1\}$ with a constant, $\{2, 1/2\}$ with variable s, and $\{3, 1/3\}$ with variable t, then the distribution of M_i (modulo inversion) can be represented by the following polynomial

$$f(s, t, q) = q^2/2 + 2q(1-q)s + (1-q)^2 t \tag{4.13}$$

where $1-q$ is the erasure probability at the leaves. For a target node of degree d, the joint distributions of messages $M_1, \cdots, M_d$ is given by a product distribution $\prod_i p_{M_i}$. Modulo permutation of messages, these can be represented by

$$f(s, t, q)^d = \sum_{j,k:j+k\leq d} f_{jk}^d(q) s^j t^k. \tag{4.14}$$

Define $S_i = \mathbb{1}_{\{M_i \in \{2,1/2\}\}}$ and $T_i = \mathbb{1}_{\{M_i \in \{3,1/3\}\}}$ to be the indicators that either $\{2, 1/2\}$ or $\{3, 1/3\}$ are sent, respectively. Let $S = \sum_{i=1}^d S_i, T = \sum_{i=1}^d T_i$. Note that $\mathbf{P}(S = j, T = k) = f_{jk}^d(q)$, i.e., the coefficient of $s^j t^k$ in the above expansion of $f(s, t, q)^d$ is the probability of the event $\{S = j, T = k\}$. If we find the conditional error E_{jk} associated with each monomial term in f, then we can conveniently represent the erasure polynomial as follows

$$E_d^{\mathrm{BEC}}(q) = \sum_{j,k:j+k\leq d} f_{jk}^d(q) E_{jk}. \tag{4.15}$$

To this end, define $M(j, k) = (M_i(j, k))$ for all $i \leq d$ with

$$M_i(j,k) = \begin{cases} 2 & 0 \leq i \leq j \\ 3 & j < i \leq j+k \\ 1 & \text{otherwise,} \end{cases} \tag{4.16}$$

to map S, T back to a realization of the incoming messages to X_0. By symmetry

$$\mathbf{P}(\hat{X}_0 \neq X_0 | S = j, T = k) = \mathbf{P}(\hat{X}_0 \neq X_0 | M(j,k)).$$

Let $A = (A_i)$ with $A_i = \mathbb{1}_{\{\Delta_i Y_0 = X_0\}}$ being the indicator that the i-th majority agrees with the target bit. Let $p_{a|jk} = \mathbf{P}(A = a | M(j,k))$ be the conditional probability that a is realized given the incoming messages. Since the events $\{\Delta_i Y_0 = X_0\}$ are independent conditioned on M_i's we have

$$p_{a|jk} = \prod_i \mathbf{P}(A_i = a_i | M_i(j,k)) = \prod_i \frac{1}{1 + M_i(j,k)^{2a_i - 1}}. \tag{4.17}$$

The conditional probability of error given the joint realization of messages and majority votes is given by

$$E_{jk|A} = \min\left(\frac{1}{1 + \prod M_i(j,k)^{1-2a_i}}, \frac{\prod M_i(j,k)^{1-2a_i}}{1 + \prod M_i(j,k)^{1-2a_i}}\right) \tag{4.18}$$

It is convenient to define

$$E_{jk} = \sum_{a \in \{0,1\}^d} p_{a|jk} E_{jk|A} \tag{4.19}$$

and think of it as the error associated to the monomial $y^j z^k$ in (4.14). Algorithm 1 summarizes the proposed procedure to compute the erasure polynomial. For instance, for degree 4 nodes we have the following erasure polynomial:

$$\begin{aligned} E_4^{\mathrm{BEC}}(q) = {} & 0.03125q^8 + 0.25q^7(-q+1) + 1.25q^6(-q+1)^2 \\ & + 2.875q^5(-q+1)^3 + 4.6875q^4(-q+1)^4 \\ & + 4.4375q^3(-q+1)^5 + 2.84375q^2(-q+1)^6 \\ & + 0.9375q(-q+1)^7 + 0.15625(-q+1)^8. \end{aligned}$$

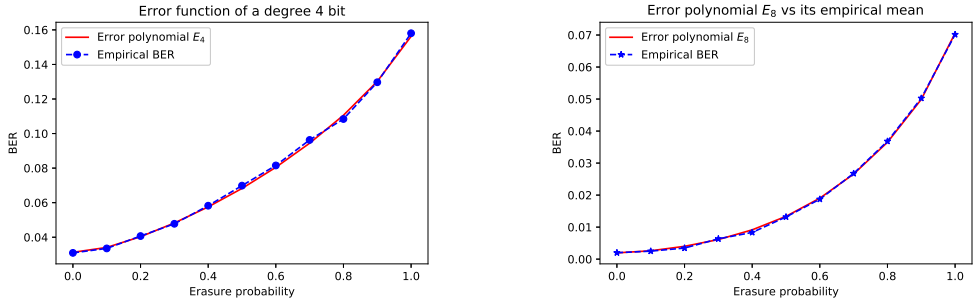
Fig.4-3 compares E_d^{BEC} with the empirical BER of degree d nodes across samples

Algorithm 1 Compute $E_d(q)$

0: **function** ErrorPoly(d):
0: Define $f(s,t,q) = q^2/2 + 3/2q(1-q)s + (1-q)^2 t$
0: Expand the d-th power of f

$$f^d(q) = \sum_{j,k} f_{jk}^d s^j t^k$$

0: Initialize $E := 0$
0: **for** k:=1 to d and $j \leq k$ **do**
0: Compute E_{jk} using (4.16)-(4.19)
0: Update $E := E + E_{jk} f_{jk}^d$
0: **end for**
return E
0: **end function**=0



(a) $d = 4$ (b) $d = 8$

Figure 4-3: Comparing the erasure polynomials $E_4^{\text{BEC}}, E_8^{\text{BEC}}$ with their empirical means. The empirical curves are obtained using 50000 samples from the computational trees of depth 1 for target nodes of degrees 4 and 8, respectively, with leaves observed through BEC_ϵ as in Fig. 4-2b.

from its depth 1 computational tree with BEC inputs for d=4,8[1]. For many code ensembles an exact computation of E_d^{BEC} is often computationally prohibitive. In such cases, one can sample from the computational tree and find E_d^{BEC}'s by solving a regression problem. Such functions are useful in optimizing codes as we will see in the next sections.

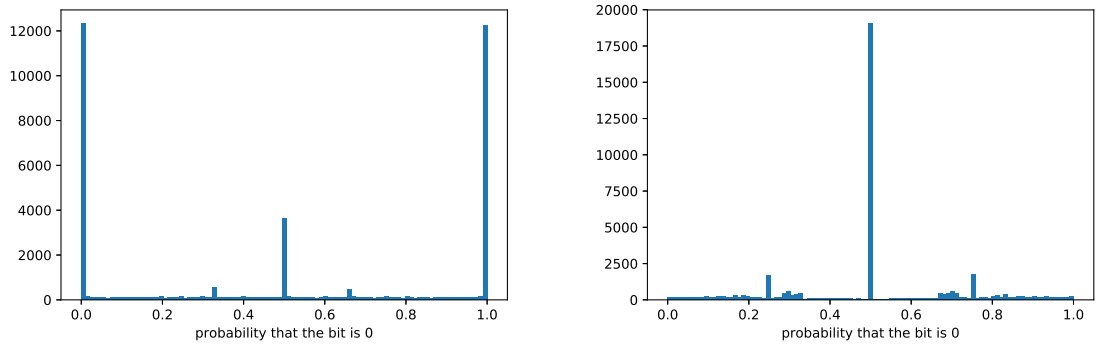
Recall the definition of $q_t^{\text{BEC}}(x_0)$ from (4.4)-(4.5). Once we compute the E-polynomials, we iterate the dynamical system in (4.4)-(4.5) to find bounds on the

[1]In some references, E-polynomials are called EXIT functions and the corresponding plots are called EXIT curves.

C/R	$(E_2^{\text{BEC}}, \text{BER}_2)$	$(E_3^{\text{BEC}}, \text{BER}_3)$	$(E_4^{\text{BEC}}, \text{BER}_4)$	$(E_5^{\text{BEC}}, \text{BER}_5)$
0.25	(0.194,0.202)	(0.127,0.146)	(0.097,0.117)	(0.068,0.093)
0.5	(0.166,0.177)	(0.106,0.124)	(0.070,0.090)	(0.047,0.066)
1	(0.137,0.139)	(0.077,0.081)	(0.044,0.047)	(0.025,0.028)

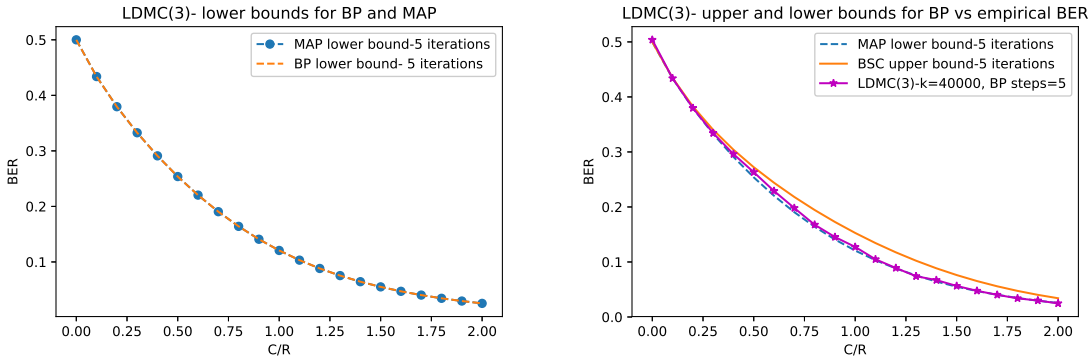
Table 4.1: Comparing BER_d, the empirical bit error rate of degree d nodes after 10 iterations of BP, with the theoretical lower bounds E_d^{BEC} at various C/R's. The lower bounds are computed at $1 - 2\text{BER}$ for each C/R where BER is obtained empirically.

decoding error. We compare the bounds with the empirical performance of BP in Fig.4-5 for LDMC(3). We see a good agreement between the two. In particular, we see that the lower bound for LDMC(3) is almost tight. To explain this, we need to consider the distribution of posterior beliefs in LDMC(3). As shown in Fig.4-4, the empirical histogram of beliefs after convergence of BP at $C/R = 1$ has three major spikes: two spikes at $p = 0, 1$ and one at $p = 0.5$. The rest of the beliefs are almost uniformly distributed across the range $[0, 1]$. It thus seems reasonable to approximate the posteriors obtained by BP as if they were induced by erasure channels. We emphasize that this phenomenon is specific to ensembles of degree 3. For larger degrees, the histogram has a pronounced uniform component (see Fig. 4-7). Thus one cannot expected a similar agreement between the BEC lower bound and the BER performance (see Fig. 4-6).



(a) $C/R = 1$ (b) $C/R = 0.25$

Figure 4-4: The empirical histogram of belief distributions for LDMC(3) with $k = 40000$ bits. The number of bits that are 0 with probability close to p are shown as a function of p for a) $C/R = 1$ b) $C/R = 0.25$.



(a) Density evolution lower bounds for MAP and BP

(b) Bounds from density evolution vs empirical BER

Figure 4-5: The LDMC(3) performance with 5 iterations of BP along with the bound of Proposition 9 using $\ell = 5$ and $E_{d\leq 10}$-functions. a) The density evolution dynamics of (4.4) has a unique fixed point. Hence both bitwise MAP and BP lower bounds converge to the same point. We remark that this property does not hold for general codes (see Conjecture 1 and Remark 5 below). b) The BP performance is compared against the bitwise MAP lower bound. The lower bound is almost tight since the empirical histogram of beliefs in LDMC(3) is much closer to one induced by an erasure channel than BSC (see Fig. 4-4).

For smaller degrees, it is possible to lower bound E_d with a simpler to compute polynomial A_d for all $q \in [0, 1]$. We describe this idea next.

Table 4.1 compares the values of $E_d^{\text{BEC}}(1 - 2\text{BER})$ with the empirical BER of degree d nodes in the LDMC(3) ensemble after 10 iterations of BP.

The ideas to compute the BSC upper bound are similar. Recall that in (4.19), E_{jk} is the error associated to the monomial $s^j t^k$ (meaning that j of type 1 and k of type 2 messages are received) for LDMC(3). In general we can re-write (4.19) in the form

$$\sum E_{i_1,\cdots i_s} f^d_{i_1,\cdots i_s}(q)$$

where again $E_{i_1,\cdots i_s}$ is a channel-independent term that corresponds to the conditional error given the input types at the boundary. The only term that depends on the channel is $f^d_{i_1,\cdots i_s}(q)$. Thus for any channel once we find the corresponding f-polynomial we can construct upper/lower bounds as above.

Let us construct the f-polynomial of LDMC(3) for BSC. Again consider the local

neighborhood of a target node connected to one majoiry. Note that for the two leaf nodes in the boundary, each realization $00, 01, 10, 11$ is equally likely (after possible flips by BSC). We need to compute the likelihood that they agree with their majority given the realization. Let ∂X_0 be the boundary bits, $\partial \tilde{X}$ be their observations through BSC(p), and Y the majority. We proceed as follows.

- The observed value is $\tilde{\partial} X_0 = 00$:

$$\begin{aligned}\mathbf{P}(Y = 0|\partial \tilde{X}_0 = 00) \propto \mathbf{P}(\tilde{\partial} X_0 = 00|\partial X_0 = 00)\mathbf{P}(Y = 0|\partial X_0 = 00)\mathbf{P}(\partial X_0 = 00) \\ + \mathbf{P}(\partial \tilde{X}_0 = 00|\partial X_0 = 01)\mathbf{P}(Y = 0|\partial X_0 = 01)\mathbf{P}(\partial X_0 = 01) \\ + \mathbf{P}(\partial \tilde{X}_0 = 00|\partial X_0 = 10)\mathbf{P}(Y = 0|\partial X_0 = 10)\mathbf{P}(\partial X_0 = 10).\end{aligned}$$

  We can check that normalization constant is 4. Hence

$$\mathbf{P}(Y = 0|\partial \tilde{X}_0 = 00) = 4((1-p)^2 \times 1/4 + p(1-p)1/2 \times 1/4 + p(1-p)1/2 \times 1/4) = (1-p)^2 + p(1-p).$$

  The message corresponding to this event is

$$\mathbf{P}(X_0 = 0|Y = 0, \partial \tilde{X}_0 = 00) = 1 + 2/\alpha$$

  with $\alpha = \frac{1-p}{p}$. The complimentary event $\mathbf{P}(\Delta Y_0 = 1|\partial \tilde{X}_0 = 00)$ has probability $p(1-p) + p^2$ and the corresponding message sent to the target node is

$$\frac{\mathbf{P}(X_0 = 0|Y = 1, \partial \tilde{X}_0 = 00)}{\mathbf{P}(X_0 = 1|Y = 1, \partial \tilde{X}_0 = 00)} = \frac{1}{1 + 2\rho}.$$

  Let s represent $1 + 2/\alpha$ and t represent $1 + 2\alpha$ (modulo inversion). Then so far we have $\mathbf{P}(\partial \tilde{X}_0 = 00) = 1/4$ and

$$\frac{\mathbf{P}(X_0 = 0|\partial \tilde{X}_0 = 00)}{\mathbf{P}(X_0 = 1|\partial \tilde{X}_0 = 00)} = t(p(1-p) + p^2) + s((1-p)^2 + p(1-p)).$$

- Suppose that $\partial\tilde{X}_0 = 11$ is observed. By symmetry

$$\mathbf{P}(Y = 0|\partial\tilde{X}_0 = 11) = \mathbf{P}(Y = 1|\partial\tilde{X}_0 = 00) = p(1-p) + p^2.$$

  The corresponding message is $1 + 2\alpha$. Likewise

$$\mathbf{P}(Y = 1|\partial\tilde{X}_0 = 11) = \mathbf{P}(Y = 0|\partial\tilde{X}_0 = 00) = p(1-p) + (1-p)^2$$

  with message $\frac{1}{1+2/\alpha}$. Thus $\mathbf{P}(\partial\tilde{X}_0 = 11) = 1/4$ with

$$\frac{\mathbf{P}(X_0 = 0|\partial\tilde{X}_0 = 11)}{\mathbf{P}(X_0 = 1|\partial\tilde{X}_0 = 11)} = t(p(1-p) + p^2) + s((1-p)^2 + p(1-p)).$$

- Suppose that $\partial\tilde{X}_0 = 01$ or $\partial\tilde{X}_0 = 10$ is observed. We have $\mathbf{P}(\partial\tilde{X}_0 = 01) = \mathbf{P}(\partial\tilde{X}_0 = 10) = 1/4$ with

$$\mathbf{P}(Y = 0|\partial\tilde{X}_0 = 10) = \mathbf{P}(Y = 1|\partial\tilde{X}_0 = 10) = 1/2$$

  by symmetry. The corresponding messages in each case are, $1 + \alpha + 1/\alpha$ for $Y = 0$ and $\frac{1}{1+\alpha+1/\alpha}$ for $Y = 1$, which we represent by z.

- Adding up all the terms, we get the following f-polynomial to compute E^{BSC}:

$$f = \frac{z}{2} + \frac{1}{2}(t(p(1-p) + p^2) + s((1-p)^2 + p(1-p))). \tag{4.20}$$

We use this polynomial to compute $E^{\mathrm{BSC}}_{\leq 10}$ and obtain an upper bound on BP error using Proposition 9. The upper bound is compared with the simulation results in Fig.4-5.

4.6 Comparing LDMC(3) with LDMC(5)

It is natural to ask how the BER curves behave for LDMC(d) as d grows. This question is in general computationally difficult to answer. The girth of the compu-

tational graph grows exponentially fast with d and BP iterations do not seem to stabilize quickly enough when d is large. Hence, one needs to consider codes of large length and more iterations of BP. Here we compare the performance of LDMC(5) with LDMC(3). We also compute the erasure function of LDMC(5) and compare the corresponding bound with simulations. As mentioned before, the spiky nature of histogram observed in Fig. 4-4 is specific to the ensembles of degree 3 and hence one cannot expect the BEC lower bound of Proposition 9 to give equally good predictions on BER for higher degrees.

We first work out the computation of E^{BEC} for LDMC(5). As before, we need to consider various cases for realization of erasures at the input layer:

- No input bits are erased. This case occurs with probability q^4. If the input bits are balanced, no error occurs. The complimentary event in which the bits are not balanced has probability $5/8$, in which case the message to the target bit is $\mathbb{P}(X_0 = 0)/\mathbb{P}(X_0 = 1) = 1$ and error is $1/2$. The corresponding term is $5/8q^4$.

- One bit is erased. This happens with probability $4q^3(1-q)$. There are two cases to consider in which an error may occur: 1) all three unerased bits agree with the majority; this happens with probability $1/4$, and the corresponding message is 1. 2) Two unerased bits agree with the majority; this happens with probability $9/16$; the corresponding message is 2, which we represent with u. Overall, the error term is $4q^3(1-q)(1/4 + 9/16u)$.

- Two bits are erased. This happens with probability $6q^2(1-q)^2$. There are two cases in which an error may occur: 1) both unerased bits agree wit the majority; this happens with probability $7/16$, and the corresponding message is $4/3$, denoted by w. 2) one unerased bit agrees with the majority; this happens with probability $1/2$, and the corresponding message is 3, denoted by z. Overall, the error term is $6q^2(1-q)^2(7/16w + 1/2z)$.

- Three bits are erased. This happens with probability $4q(1-q)^3$. Two cases need to be considered: 1) the unerased bit agree with the majority, which

happens with probability 11/16, in which the message is 7/4; we represent this message by v. 2) the unerased bit disagrees with the majority, which happens with probability 5/16 and gives a message of 4, represented here by s. The corresponding term is $4q(1-q)^4(11/16v + 5/16s)$.

- All bits are erased. This happens with probability $(1-q)^4$ in which case the message is 11/5. We represent this message by t. The corresponding term is $(1-q)^4 t$.

- Adding up all the terms, we get the following polynomial

  f(q)=$5/8q^4 + 4q^3(1-q)(1/4 + 9/16u) + 6q^2(1-q)^2(7/16w + 1/2z)$
  $+ 4q(1-q)^3(11/16v + 5/16s) + (1-q)^4 t.$

Using (4.15) we compute $E^{\mathrm{BEC}}_{\leq 10}$ for LDMC(5) and then apply Proposition 9 to compute a lower bound on BER. The results are shown in Fig. 4-6 along with comparisons between ensembles of degree 3 and 5 for 5 iterations of BP. We note that the effect of truncation is of a lower order than the scale of the plots in Fig. 4-6. Since $E_d(q)$ is monotonically decreasing in d and q, we can deduce for all $\alpha \leq 1$ that

$$|E^{\mathrm{BEC}}(\alpha, q) - E^{\mathrm{BEC}}_{\leq 10}(\alpha, q)| \leq E^{\mathrm{BEC}}_{10}(0)\mathbb{P}(\mathrm{Poi}(5) > 10) = 0.001.$$

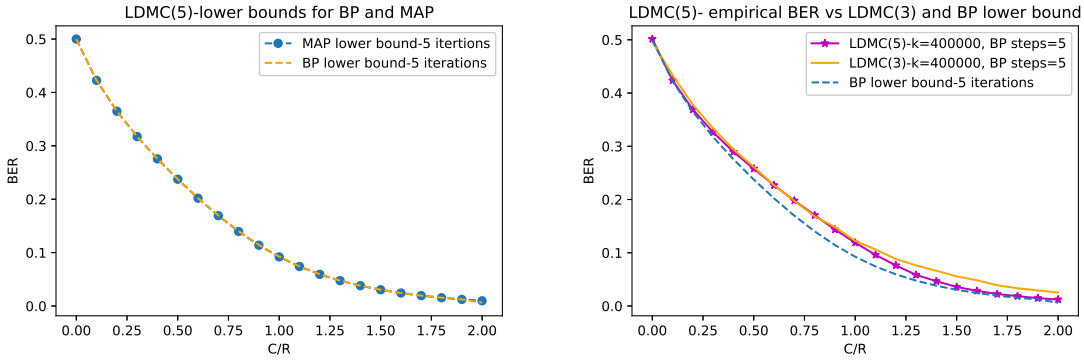
Thus the gap between q_ℓ and δ^{BP} for the degree 5 ensemble cannot be attributed to the truncation, but rather to the role of the "uniform" component of the belief histogram shown in Fig. 4-7.

We still see in Fig. 4-6a that q_l^{BEC} converges to a unique point regardless of the initial condition for LDMC(5). We remark that the same holds for the error dynamics of the large d limit obtained below in (4.25). In the view of these observations, we put forth the following conjecture:

Conjecture 1. For any ensemble generated by a monotone function, $q_\ell^{\mathrm{BEC}}(x)$ converges

to a unique fixed point independent of x.

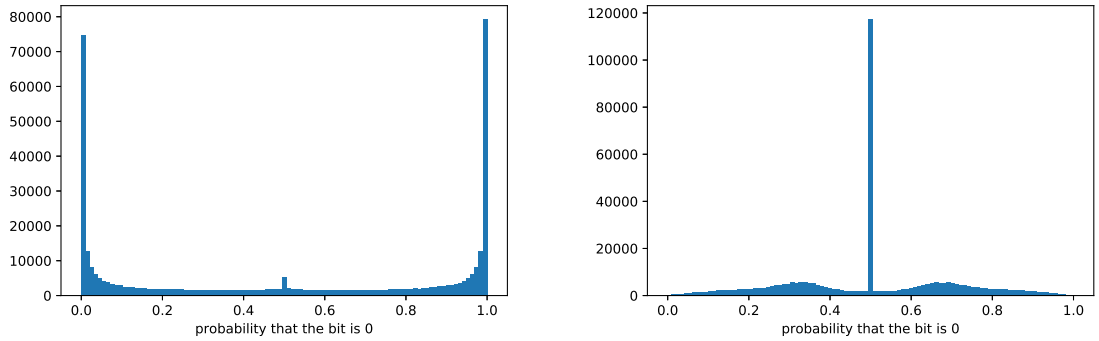
Remark 5. We note that the conjecture does not hold for general ensembles. For instance, we have $q_\ell^{\mathrm{BEC}}(1) > 0$ for ensembles generated by XOR whereas $q_\ell^{\mathrm{BEC}}(0) = 0$ for all ℓ. In fact, Mackay showed in [46] that the (check regular) ensembles generated by XOR are *very good*, meaning that for large enough degree they can asymptotically achieve arbitrarily small error for rates close to capacity under MAP decoding. Evidently, such performance cannot be achieved by BP since for any degree larger than 1 $q = 0$ is a fixed point of BP, i.e., BER is 1/2 for all ℓ. This point shall be explained further in Chapter 5 (see Fig. 5-2).



(a) Density evolution lower bounds for MAP and BP

(b) Bounds from density evolution vs empirical BER

Figure 4-6: The LDMC(5) performance with 5 iterations of BP along with the lower bound of Proposition 9 using $\ell = 5$ and $E_{\leq 10}^{\mathrm{BEC}}$-function. a) The density evolution dynamics of (4.4) has a unique fixed point. Hence both bitwise MAP and BP lower bounds converge to the same point. b) The BP performance is compared against LDMC(3) and the BP lower bound. The lower bound can be seen to be looser compared with the scenario in Fig.4-5. This can be attributed to the fact that the empirical histogram of beliefs in LDMC(5) shown in Fig. 4-7 is less spiky compared with 4-4 and can no longer be well approximated by one parameter (i.e., the erasure probability). Furthermore, LDMC(5) can be seen to perform better than LDMC(3) for all erasure probabilities. However, we note that longer codes are needed to avoid short cycles in the computational graph and achieve good decoding performance for $d = 5$.



(a) $C/R = 1$ (b) $C/R = 0.25$

Figure 4-7: The empirical histogram of belief distributions for LDMC(5) with $k = 400000$ bits after 5 iterations of BP. The number of bits that are 0 with probability close to p are shown as a function of p for a) $C/R = 1$ b) $C/R = 0.25$.

4.7 Tighter bounds for systematic LDMC(d) with $d = 3, 5$

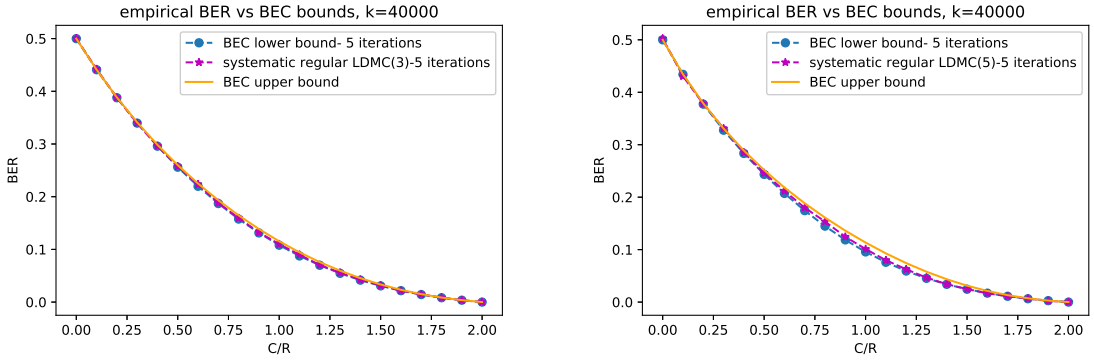
It is possible to obtain tighter bounds for systematic ensembles. Here we study the case of systematic regular LDMC(3). The next section extends the analysis to the large d limit for LDMC(d).

Consider a regular ensemble of (check) degree d. Let ϵ be the probability of erasure and R be the rate of the code with variable degree $1 + d(1-R)/R$. Note that we need $d(1-R)/R \in \mathbb{Z}$ to ensure that a regular systematic code exists. As before let $\alpha = C/R$. For a regular systematic ensemble of rate R, we have the following erasure function:

$$E^{\mathrm{BEC}}(q, \alpha) = (1 - \alpha R) \sum_{i \leq d(1-R)/R} \mathbb{P}(\mathrm{Bin}(\frac{d(1-R)}{R}, \alpha R) = i) E_i^{\mathrm{BEC}}(q, \alpha). \qquad (4.22)$$

The key observation is that BP can be initially loaded with the information we obtain from systematic bits. In other words we can iterate the dynamical system in (4.4) with $x_0 = 1 - \alpha R$ and E^{BEC} as above. Clearly, $q_1^{\mathrm{BEC}}(x_0)$ gives an exact estimate for the first iteration of BP and can serve as an upper bound for the error $\delta_\ell^{\mathrm{BP}}$. The results are shown in Fig. 4-8 for $R = 1/2$ and $d = 3$. The bounds can be seen to

be rather tight. We note that the accuracy of these bounds depend primarily on the rate and the check degree of the ensemble and not the regularity assumption.



(a) $C/R = 1$ (b) $C/R = 0.25$

Figure 4-8: The performance of systematic regular LDMC(d) of rate $R = 1/2$ using 5 steps of BP along with the upper and lower bounds obtained from the erasure functions for a) d=3 b) d=5. The upper bound uses one iteration of (4.4) with the erasure function as in (4.22) and initialized at $x_0 = 1 - \alpha R$. The initial point is the fraction of unerased bits observed in the systematic portion of the code.

4.8 Upper bound for systematic LDMC(d) as $d \to \infty$

Now we consider the case where the node degree tends to infinity for systematic LDMC(d) of rate $\frac{1}{2}$. To get an upper bound for LDMC codes in this case, we can analyze one step of BP. To do this, we first need to understand what a typical majority to bit message looks like as degree increases.

Consider a majority Y of $d+1$ bits $X_0, \cdots, X_d$. Let $r_i = \frac{\mathbf{P}(X_i=0)}{\mathbf{P}(X_i=1)}$. Then the BP update rules for X_0 are as follows:

$$\frac{\mathbf{P}(X_0 = 0|Y = 0)}{\mathbf{P}(X_0 = 1|Y = 0)} = 1 + \frac{\sum_{|I|=d/2} \prod_{i\in I} 1/r_i}{\sum_{|I|<d/2} \prod_{i\in I} 1/r_i}. \tag{4.23}$$

Set $x = C/R$. Initially, around $p = x/2$ fraction of the bits are return by the channel. We have that of the $d-1$ nodes that x_0 is connected to, around dp are recovered perfectly. In this case, roughly $dp/2$ send a message of $r_i = \infty$ and the rest send $r_i = 0$. There are around $(1-p)d$ nodes that are undecided and send a message of

1 into the local neighborhood. Then if we group the terms in the numerator that contain the strong $1/r_i = \infty$ messages with the terms that send the uninformative $r_i = 1$, we get the dominating terms in both the numerator and denominator of (4.23). Let S' be the subset of nodes with $r_i = 1$. Given that $|S'| \approx d(1-p)$, the majority to bit message is asymptotically as follows:

$$\frac{\mathbf{P}(X_0 = 0|Y = 0)}{\mathbf{P}(X_0 = 1|Y = 0)} \approx 1 + \frac{\sum_{I \subset S', |I| = d(1-p)/2} 1}{\sum_{I \subset S', |I| \leq (d-2)(1-p)/2} 1} = 1 + \frac{\binom{d(1-p)}{d(1-p)/2}}{\sum_{j \leq (d-2)(1-p)/2} \binom{d(1-p)}{j}}.$$

By Stirling's approximation, the numerator behaves as:

$$2^{d(1-p)} \sqrt{\frac{2}{d(1-p)\pi}}$$

and the denominator is roughly

$$2^{d(1-p)}/2.$$

Then the triangle to bit message when $Y = 0$ is

$$1 + 2\sqrt{\frac{2}{d(1-p)\pi}}.$$

Some of the incoming messages to x_0 will cancel each other and the rest will amplify. If N_0 is the number of majorities that evaluate to 0 and N_1 is the number of majorities that evaluate to 1, then the decoding error at x_0 is

$$\frac{1}{1 + (1 + 2\sqrt{\frac{2}{d(1-p)\pi}})^{|N_0 - N_1|}}. \tag{4.24}$$

If we integrate this expression w.r.t the distribution of $N_0 - N_1$ then we get the average error at x_0. One can show that the probability that a node agrees with its majority is:

$$\frac{1}{2}(1 + \sqrt{\frac{2}{\pi d}}).$$

Note that $N_0 - N_1$ is asymptotically normal by the CLT. When $Y = 0$, $N_0 - N_1$ has

mean $d'x\sqrt{\frac{2}{\pi d}}$ and variance $d'x$ where $d' = d(1-r)$. When $r = 1/2$ we get $d' = d/2$ and initially we have $p = 1/2$. Thus $N_0 - N_1 \sim d'x\sqrt{\frac{2}{\pi d(1-p)}} + \sqrt{d'x}Z$ where Z is standard normal. We can write this as $N_0 - N_1 \sim \sqrt{d'x}(\sqrt{d'x}\sqrt{\frac{2}{\pi d}} + Z)$.

We can integrate (4.24) w.r.t to this density to find the average decoding error after one iteration of BP. Setting $d' = d(1-r)$ and taking the limit as $d \to \infty$, we find that

$$\lim_{d\to\infty} \int_{-\infty}^{\infty} \frac{1}{1 + (1 + 2\sqrt{\frac{2}{d(1-p)\pi}})^{(\sqrt{dx/2}(z+\sqrt{\frac{x}{\pi}}))}} f(z)dz = \int_{-\infty}^{\infty} \frac{1}{1 + e^{2\sqrt{\frac{2x(1-r)}{\pi(1-p)}}z + \frac{4x(1-r)}{\pi\sqrt{1-p}}}} f(z)dz. \tag{4.25}$$

This integral gives an upper bound on the decoding error of BP in the asymptotic regime of large d. Fig. 4-9 shows the above bound versus the empirical performance of LDMC(17). BP converges fast for systematic LDMCs, which explains the accuracy of this one step prediction.

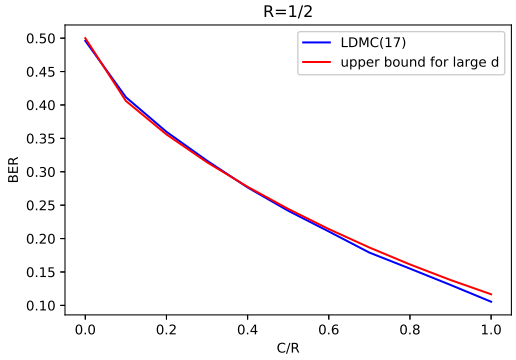


Figure 4-9: The empirical performance of LDMC(17) after 5 iterations along with the predicted on step error in the large d limit obtained from (4.25).

Chapter 5

Applications in code optimization

In this chapter we study optimization of LDGMs. Recall that LDGM(d) is the (check regular) ensemble of degree d generated by the parity function. We show that a joint design over LDGMs and LDMCs can uniformly improve the performance of LDGMs in some simple settings.

As discussed in the introduction, LDGMs are some of the most widely used families of linear codes. They are known to be good both in the sense of coding [46] and compression [79]. In fact, [46] shows that LDGM(d) (for odd[1] $d \geq 3$) enjoys, from a theoretical perspective, almost all the good properties of random codes. Indeed as shown in Fig. 5-1, even relatively short LDGMs can achieve reasonably small error under MAP decoding. As the codes get longer, and the degrees grow, the error can be made arbitrarily small for all $C/R > 1$. From a practical perspective, however, their decoding is problematic. The problem is that MAP decoding is not easy to implement in practice even for moderate size codes. BP decoding is not feasible either since for such codes, as generated, BP has a trivial local minima in which all bits remain unbiased. One may hope that adding a small number of degree 1 nodes would enable BP to get around this initial fixed point and achieve near optimal performance. Unfortunately, this is not the case. Improving the performance of BP for LDGMs is a non-trivial task that often involves some careful code optimization

[1] When d is even the all one vector is in the kernel of the generator matrix. This implies that BER is $1/2$.

with many relevant parameters. We briefly discuss this matter next.

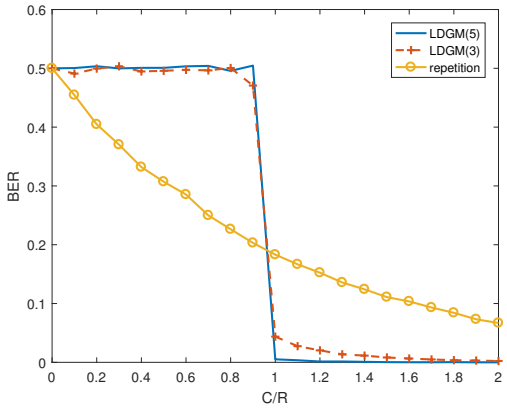


Figure 5-1: The empirical performance of LDGM(d) for $d = 1, 3, 5$ under (block) MAP decoding for rate $R = 1/2$ and $k = 1000$. It can be seen that the codes quickly achieve a threshold-like performance close to the IT limit. In the view of Theorem 5, bitwise MAP decoding cannot yield any major improvements for either codes.

To understand how LDGM(d) behaves under BP, we first construct its erasure function and then appeal to Proposition 9. With the notation of Fig. 4-2, we note that a parity check Δ_j of degree d can determine a target bit X_0 if all of its $d-1$ leaf bits $\partial_j X_0$ in the local neighborhood are unerased. Otherwise, it sends an uninformative message. Thus if q is the probability of erasure coming into the local neighborhood after some iterations of BP, then at the next iteration the target bit remains erased with probability $(1-q^{d-1})^i$. This gives the i-th erasure polynomial $E_i^{\mathrm{BEC}}(q) = 1/2(1-q^{d-1})^i$. Since the variable node degrees are Poisson distributed (with parameter αd), we obtain the following erasure function

$$E^{\mathrm{BEC}}(\alpha, q) = \frac{1}{2}\sum_i \mathbb{P}(\mathrm{Poi}(\alpha d) = i)(1-q^{d-1})^i. \tag{5.1}$$

5.1 D-curves

We can now study, as before, the local dynamics of error under BP. Let $q_\ell^{\mathrm{BEC}}(x_0)$ be computed as done in Proposition 9. We note that $q_\ell^{\mathrm{BEC}}(0)$ in this case is (asymptotically) exact for predicting BP error, meaning that whenever the computational graph is a tree the average error is equal (and not just lower bounded) by q_ℓ^{BEC}. This is

due to the fact that parity checks preserve the BEC structure of the input messages, hence, we can write

$$\delta_\ell^{\mathrm{BP}} = q_\ell^{\mathrm{BEC}}(0) + o(1).$$

Thus for BP to make any progress during decoding, we can impose the following (necessary) contraction constraint, called the D-function of the ensemble:

$$D(\alpha, q) := \frac{1-q}{2} - E^{\mathrm{BEC}}(\alpha, q) \geq 0. \tag{5.2}$$

Similarly, we can define the truncated D-function:

$$D_{\leq d}(\alpha, q) := \frac{1-q}{2} - E_{\leq d}^{\mathrm{BEC}}(\alpha, q) \geq 0. \tag{5.3}$$

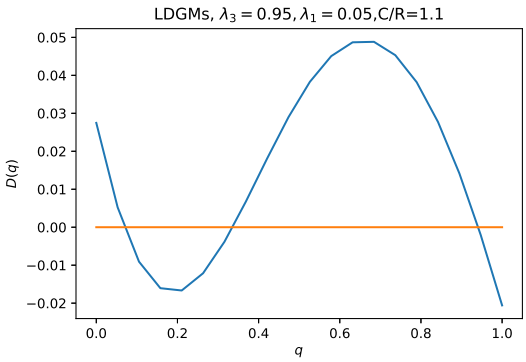
In other words, we simply want the outgoing error to be less than the error flowing in. For iterative decoding to take off, we need $D(0) > 0$. Then the first point where $D(q) = 0$ occurs determines the limiting performance of BP. Fig. 5-2 shows the results for LDMC(3) and a mixed LDGM ensemble, which is defined as follows. Let Λ be a degree distribution over check degrees. An LDGM ensemble is said to be Λ-mixed if each check node in the code is selected i.i.d from Λ.

5.2 Improving LDGMs via LDMCs

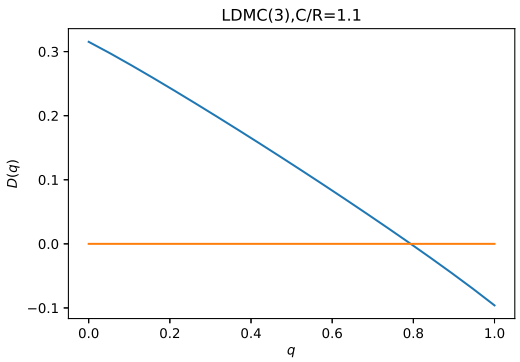
It is easy to define the erasure function of such an ensemble in terms of the erasure function of its regular components. Let $\lambda_i := \mathbb{P}(\Lambda = i)$. Suppose that Λ has finite support with cardinality m. Then the erasure polynomial of an Λ-mixed LDGM ensemble is simply

$$E_\Lambda^{\mathrm{BEC}}(\alpha, q) = 2^{m-1} \prod_{i=1}^{m} E_{\mathrm{LDGM}(i)}^{\mathrm{BEC}}(\alpha\lambda_i, q). \tag{5.4}$$

We note that this expression is half the probability that a variable node receives an uninformative message from each component of the code in the ensemble. The



(a) D-curve for mixed LDGM at $\alpha = 1.1$, truncated at $d = 10$



(b) D-curve for LDMC(3) at $\alpha = 1.1$, truncated at $d = 10$

Figure 5-2: The truncated D-function from (5.3) viewed as a function of q for $\alpha = 1.1$. The first zero of $D(q)$ determines the fixed points of BP dynamics in (4.4). The fixed points remain stable with respect to small variations in the truncation degree $d = 10$. Two ensembles are considered: a) A mixed LDGM ensemble using $\lambda_1 = 0.05$ fraction of degree 1 nodes and $\lambda_3 = 0.95$ fraction of degree 3 nodes. The degree 1 nodes are needed so that $D(0) > 0$ is satisfied. It can be seen that BP has fixed point near 0. Thus small perturbations in degree distributions cannot help BP reach the MAP level of performance shown in Fig. 5-1, which corresponds to the right most zero of the D-function. More sophisticated optimization is required to improve the performance. b) The LDMC(3) ensemble. As expected from the observations in Figs. 4-5-4-6, $D(q)$ has a unique fixed point. See also Conjecture 1. Furthermore, the relatively large value of $D(0)$ suggests that the convergence is fast for this ensemble.

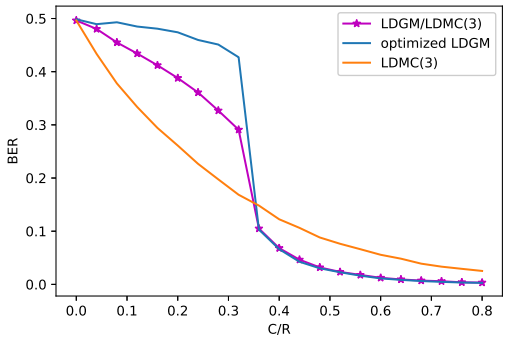
code optimization problem now can be formulated in terms of the dynamical system in (4.4) associated with this E-function. Suppose that we want to run ℓ iterations of BP to decode an LGDM. Let $q_{\ell,\alpha}^{\mathrm{BEC}}(0)$ be density of unerased bits after ℓ iterations with $C/R = \alpha$. If we are interested in minimizing the BP error at two different C/R's, say α_1 and α_2, then the following optimization problem becomes relevant

$$\begin{aligned} &\text{maximize}_{\Lambda} \; q_{l,\alpha_1}^{\mathrm{BEC}}(0) + q_{l,\alpha_2}^{\mathrm{BEC}}(0) \\ &\sum_i \lambda_i = 1 \\ &\lambda_i \geq 0. \end{aligned}$$

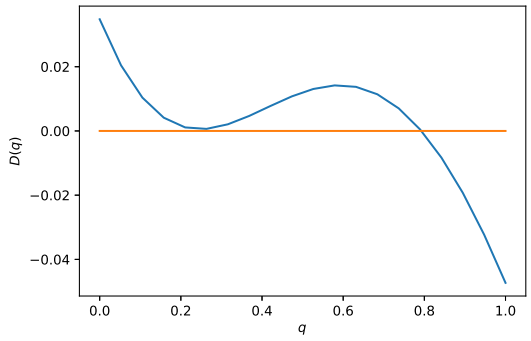
This is a non-convex optimization problem. We can solve it up to local optimality using gradient descent. Solving for $\alpha_1 = 0.9, \alpha_2 = 1.1$ over LDGM(d) with $d \leq 3$,

we find that $\lambda_1 = 0.08, \lambda_2 = 0.22, \lambda_3 = 0.7$. The D-curve and the corresponding performance are shown in Fig. 5-4. The same figure demonstrates the performance after we simply remove the lower degree parities by setting $\lambda_1 = 0, \lambda_2 = 0$, and replace them with an LDMC(3). Since LDMC(3) dominates the repeition code everywhere, we expect this new LDGM/LDMC ensemble to have lower error than the pure LDGM ensemble. We can see that the LDGM family exhibits a sharp transition at the end point $C/R = 0.9$ while the combined ensemble degrades more smoothly beyond this point while maintaining smaller error everywhere else. It can also be seen that the D-curve with optimal parameters almost touches the x-axis for some small q when $\alpha = 0.9$. This is an artifact of the optimal designs. Such proximity with zero induces a near fixed point, from which BP requires many iterations to escape until it reaches good performance.

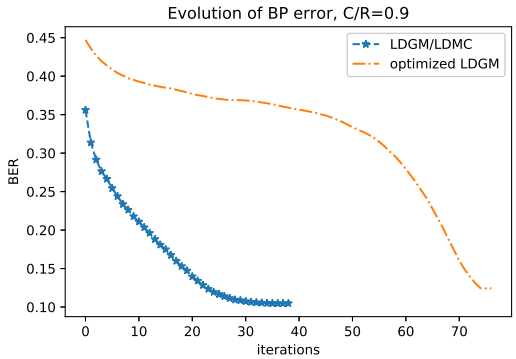
We can also optimize jointly over the LDGM/LDMC ensemble by computing the erasure polynomial as before. Solving the optimization problem at $\alpha_1 = 0.8, \alpha_2 = 1.1$ for the joint LDGM(d)/LDMC(3) ensemble (with $d \leq 3$) gives $\lambda_1 = 0.0, \lambda_2 = 0.261, \lambda_3 = 0.482$ and $\lambda_{\text{LDMC(3)}} = 0.257$ and for the LDGM ensemble (with $d \leq 3$) we get $\lambda_1 = 0.001, \lambda_2 = 0.669, \lambda_3 = 0.33$. The results are shown in Fig.5-3.



(a) LDGM (optimized over degrees ≤ 3) vs combined LDGM/LDMC.



(b) D-curve at $\alpha = 0.9$ for the optimized LDGM, truncated at $d = 10$.



(c) BP error vs time steps.

Figure 5-3: a) Code performance is compared for optimized LDGMs (over degrees ≤ 3) and combined LDGM/LDMC ensembles using 80 iterations of BP for $k = 50000$. The optimized ensemble has parameters $\lambda_1 = 0.08$, $\lambda_2 = 0.22$, $\lambda_3 = 0.7$. The LDGM/LDMC ensemble is obtained by replacing the degree 1 and 2 components of the optimized ensemble with LDMC(3). Since LDMC(3) dominates repetition for all noise levels, it is reasonable to expect that the combined LDGM/LDMC ensemble performs better. b) The D-curve for the optimized LDGM ensemble is shown. The low values of D (relative to the position of the fixed point) and the near zero point at $q \approx 0.25$ indicate that BP requires many iterations to converge. This behavior is typical for optimal designs shown at the bottom figure. The fixed point near 0.8 is compatible with the error of 0.105 obtained at $\alpha = 0.1$ on the left. c) The progress of BP error is shown at $C/R = 0.9$. The nearly flat region in the error curve can be explained by the presence of a near fixed point in the D-curve. Overall, the experiments suggest that the joint ensemble converges much faster and achieves better performance uniformly for all erasure levels.

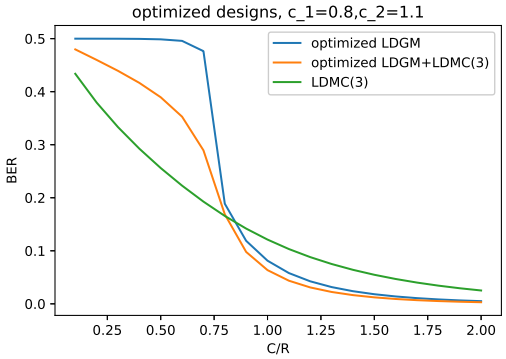


Figure 5-4: BER curves for optimized LDGM/LDMC and is compared with optimized LDGM. The optimized joint design has degree distributions $\lambda_1 = 0.0, \lambda_2 = 0.0, \lambda_3 = 0.62$ and $\lambda_{\text{LDMC(3)}} = 0.38$ and the optimized LDGM has degree distributions $\lambda_1 = 0.001, \lambda_2 = 0.669, \lambda_3 = 0.33$. The codes are optimized to minimize the sum of BERs at $\alpha_1 = 0.8$ and $\alpha_2 = 1.1$.

Chapter 6

Codes as channel transforms

In this chapter we study LDMCs from the perspective of a channel transform. This notion arises when one employs a concatenated code. Concatenated codes are the codes that act on pre-coded information. This means that the input to the code is not an arbitrary point in the alphabet space, but rather the codeword of an outer code. This technique is often used to design codes with high performance and low decoding complexity. For instance, to approach the capacity of the erasure channel with LDPCs one needs to use high degree variable nodes. These in turn create short cycles in the computational graph of the BP decoder, which is problematic for accuracy of BP. To mitigate the impact of cycles, one needs to use very large codes and many iterations of BP, leading to long delays in the communication system as well as an expensive decoding procedure. A common method to circumvent these difficulties is to employ a two (or more) layer design. A low complexity inner code $f_i : \mathcal{A}^k \to \mathcal{A}^n$ is used to reduce the channel error, without necessarily correcting any erasure pattern. Then an outer error correcting code $f_o : \mathcal{A}^m \to \mathcal{A}^k$ cleans up the remaining error. The outer code here can be an LDPC but one that faces a weakened channel, hence, it requires fewer BP iterations and can be made to be shorter. It can also be a (short) error correcting code that relies on syndrom decoding. In either

case, the overall communication path looks like the following

$$\mathcal{A}^m \xrightarrow{f_o} \mathcal{A}^k \underbrace{\xrightarrow{f_i} \mathcal{A}^n \xrightarrow{\mathrm{BEC}_\epsilon} Y \xrightarrow{g_i}}_{Q(\epsilon)\text{:channel transform}} \mathcal{B}^k \xmapsto{g_o} \mathcal{A}^m.$$

Here Y is the outcome of the channel, g_i is inner decoder and g_o is the outer decoder. The domain of the outer decoder is chosen to be different from the alphabet of the message space on purpose. This is to accommodate various decoding messages that maybe transmitted from the inner decoder to the outer decoder. Two common choices in the literature are: 1) hard decision decoding ($\mathcal{B} = \mathcal{A}$); in this case the inner decoder can only transmit a hard decision on each bit to the outer decoder corresponding to its best estimate of what the bit value is. 2) soft-decision decoding($\mathcal{B} = \mathbb{R}$); in this case the inner decoder is allowed to send the bitwise probabilities of error to the outer decoder. In either case, we can view the action of the inner code together with its decoder as one channel Q.

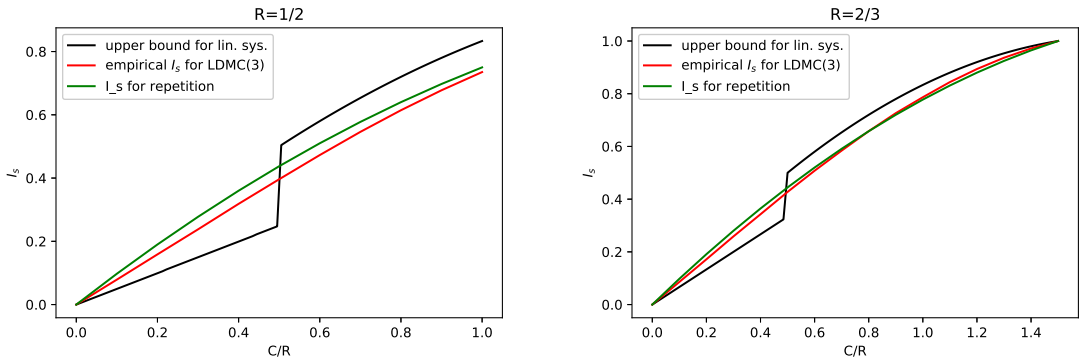
For hard decision decoding, it is clear that the channel (after interleaving) is a BSC with crossover probability equal to BER. For soft-decision decoding, the output of the channel transform is a sequence of probabilities. We view this channel as a product channel that sends the marginals on every bit to the outer decoder $Q(\epsilon) : \mathcal{A}^k \to \prod_{i=1}^k \pi_i$. In practice, often an interleaver is placed between the inner and outer decoder to ensure that the bit errors are not correlated, hence, it makes sense to model the action of the inner code with a product channel. To study the performance of codes as channel transforms under erasures we introduce the notion of soft information

$$I_s(\epsilon) = 1 - \mathbf{E}[\frac{1}{k}\sum_{i=1}^{k} h_i(\epsilon)]$$

where $h_i = h(\pi_i)$ is the binary entropy of the i-th marginal produced by $Q(\epsilon)$. The soft information can be seen as the average per-bit information sent from the inner code to the hard decision (outer) decoder. If the inner code is wrapped with an interleaver, I_s will closely approximate the capacity of the inner channel $Q(\epsilon)$. In this case, two information bits of the inner code are likely to fall in different blocks of the

outer error correcting code. Hence, the possible dependencies between the bits is not relevant.

We note that for a linear code $I_s(\epsilon) = 1-2\text{BER}(\epsilon)$. Thus we can use the bounds of Theorem 5 together with Proposition 1 to obtain similar bounds on soft information. For LDMCs we can measure the soft information empirically. The results are shown in Fig. 6-1



(a) $R = 1/2$ (b) $R = 2/3$

Figure 6-1: Empirical soft information for LDMC(3) with $k = 20000$ compared with repetition and linear codes satisfying BER $= 0.25$ at $C/R = 0.5$ for three different rates. The codes are systematic.

Part 2

Hamming's combinatorial model

Chapter 7

Combinatorial trade-offs for linear codes

7.1 Introduction

In this part of the thesis, we study the graceful degradation problem for the Hamming model. As discussed in the introduction, the goal is to understand what codes can achieve smooth (α, β)-profiles similar to what is shown in Fig. 1-2b. Henceforth, we assume familiarity with the material discussed in §1.2.

We first briefly review the relevance of the (α, β)-property for graceful degradation and explain the main results of this chapter. One often encodes a message by a map f to build tolerance against external noise. For instance, one may map x to $f(x)$ and save the outcome on a storage device. Then noise may act by erasing some of stored bits in an adversarial manner. One then observes the non-erased bits and provides an estimate $\hat{x}$ for x. With the conventions of §1.2, a map can fully recover the input from $\beta(0)n$ erasures. As the number of erasures exceeds $\beta(0)n$, it is desired that x be recovered with good fidelity, that is, we want $|x - \hat{x}|$ to be as small as possible. In general, $\beta(\alpha)n$ erasures on the output can cause at most αk distortions in the input. Indeed if we let $\hat{x}$ be an arbitrary point in the pre-image, it is guaranteed that $|x - \hat{x}| \leq \alpha k$. In some cases, this arbitrary point is the best estimate available for the input. For instance, if q is large and f is linear, then one cannot find an estimate

with provably lower error [1]. Thus it makes sense to think of $1-\alpha$ as the quality of estimation in recovering x against adversarial noise with intensity β. In this sense, β^* (see (1.3) for definition) can be thought of as a measure for the ability of the code to partially recover the input in the presence of strong erasure noise.

A related concept is that of unequal protection (UEP) codes [51, 11, 55, 63]. A code with minimum distance d is said to have the UEP if, for some fixed i, it can always recover the i-th input bit from more than d erasures. In this sense, the UEP codes are often said to have the graceful degradation property. A map with the (α, β)-property does not necessarily provide this type of biased protection. If d erasures occur there is no guarantee that any specific bit can be recovered exactly. However, more can be said about the joint estimates. For instance, if a code has $\beta(1/k) > \beta(0)$ and exactly d erasures occur, then the symbol error rate (SER) on estimating m bits from d erasures can be shown to be at most $\frac{1}{m}$. In other words, the (α, β)-property does not provide unequal protection for any specific bit but it can still ensure graceful degradation of overall SER as the noise level exceeds the error correction capabilities of the code depending on how fast β increases with α.

It is a classic problem in coding theory to find maps with large $\beta(0)$. It is thus useful to have estimates on how large $\beta(0)$ can be. The answer to this question is not yet known unless the alphabet size is large, though various upper bounds on $\beta(0)$ exist (cf. [48]). The recent work has extended this problem to finding estimates on $\beta(\alpha)$ [59, 61]. Again the exact answer is known only when the alphabet size is large. We shall see in the next chapter that $\beta(\alpha) \leq 1 - \frac{1-\alpha}{\rho}$ with $\rho := \frac{n}{k}$, where equality can be achieved if $q \geq n$. In this chapter, we focus on a different problem.

The above discussion motivates the need for a code with large minimum distance and monotonically increasing $\beta(\alpha)$. Such a code can fully recover the input when the number of erasures is less than its minimum distance, and as the number of erasures exceeds its minimum distance, it can offer some partial recovery guarantees. It turns out, however, that there is a trade-off between full and partial recovery. In the (α, β)-spectrum, we can fix one point, namely, the minimum distance (or

[1] When $q > n$, the Chebyshev radius of a linear subspace of $\mathbb{F}_q^n$ is equal to its diameter.

equivalently $\beta(0)$), and ask how large $\beta(\alpha)$ can be at some other point? We give some results in this direction for linear codes. Our results show that there is a trade-off between the minimum distance δ of a linear code and its β^* (see (1.3) for definition). We characterize the optimal trade-off between δ and β^* (over large alphabets) and construct some optimal codes that can achieve it. We further show that optimal codes are not graceful in the sense that they must send some input vectors with large weight to codewords with minimal weight δ. A priori, the (α, β)-property asks for the mapping of dissimilar messages to be also dissimilar and as such is a relaxation of the locality sensitive hashing (LSH) property ([59]). Our results show, however, that at least in the case of linear codes there is a stronger connection between the two in the sense that if a code sends dissimilar messages to dissimilar codewords, it must also send some similar messages to similar codewords (see Theorem 15).

7.2 Geometric systems

We briefly review the notion of an (α, β)-geometric system, which will be used in the proofs and ensuing discussions. We refer the reader to [59] for further details.

The (α, β)-property of f is determined by its image as well as a choice of an embedding. If we write $f(x) = xG$, then we can think of the columns of G as elements of projective space $\mathbf{P}^{k-1}$, which we will call β-points, while projective images of the k standard basis vectors are going to be called α-points. In this language, for example, $1 - \beta^*$ is the largest fraction of β-points through which we can pass a hyperplane avoiding all α-points We denote the set of α, β-points, respectively, by $\Gamma_\alpha, \Gamma_\beta$. We also define the sets of α-only points $\Gamma_{\alpha\backslash\beta} := \Gamma_\alpha \backslash \Gamma_\beta$, and β-only points $\Gamma_{\beta\backslash\alpha} := \Gamma_\beta \backslash \Gamma_\alpha$.

We remark that the minimum distance of a map is a property of its image, hence, it depends only upon the configuration of its β-points. On the other hand, β^* is a property of both the image and the embedding and as such depends on the arrangements of both α-points and β-points. The bounds in this section are thus to be interpreted as follows: fixing a property (the minimum distance) of the image, bound β^* for all possible embeddings, i.e., any configuration of α-points.

7.3 MDS codes

Here we show that linear MDS codes have $\beta^* = 1 - \frac{1}{\rho}$ when $n \geq 2k - 1$ (see (1.3) for definition of β^*). We remark that this result can be seen as a generalization of Theorems 8 and 9 in [52]. Recall that a linear map is MDS if and only if the points in Γ_β are in general linear position. We start with a simple observation to show that the bound $n \geq 2k - 1$ is sharp:

Proposition 12. *If $n < 2k - 1$ then the there exists a linear MDS code $f : \mathbb{F}_q^k \to \mathbb{F}_q^n$ for which $|f(x)| \leq n - k + 1$ implies $|x| \leq k - 1$. In other words, $\beta > 1 - \frac{1}{\rho}$ can be achieved at $\alpha = 1 - \frac{1}{k}$.*

Proof. Pick the α-points such that $\Gamma_\alpha \subset \Gamma_\beta$. Then any hyperplane containing $k - 1$ β-points must contain at least one α-point. ■

Conversely, when $\Gamma_\alpha \subset \Gamma_\beta$ one can easily check that if $\rho \geq 2$ then $\beta(\alpha) \leq 1 - \frac{1}{\rho}$ for all $\alpha < 1$. Indeed the hyperplane containing any $k - 1$ points in $\Gamma_{\beta \setminus \alpha}$ cannot contain any α-points due to the general position property of Γ_β. One can ask whether $\beta > 1 - \frac{1}{\rho}$ can be achieved for some $\alpha < 1$ for a different configuration of α-points? Let us consider another simple configuration of α-points before we prove the general result. One can place each α-point on a line between two β-points. We can now construct a graph whose nodes are the n β-points and there is an edge between two β-points if the line connecting them does not contain an α-point. The fact that the β-points are in general position implies that every α-point can be in the span of exactly one pair of β-points when $k > 3$ (two general lines do not meet). The graph is thus missing k edges compared to the complete graph on n nodes. By Turan's theorem, it must contain a $(k - 2)$-clique. Then a hyperplane containing the clique (and no further β-point) has a relative (asymptotic) weight of $1 - \frac{1}{\rho}$ and does not pass through any α-points (otherwise, the line connecting the β-points spanning this α-point and the $(k - 2)$-space spanned by the clique points will intersect, violating the general position assumption). Hence, asymptotically, we must have $\beta \leq 1 - \frac{1}{\rho}$ for all $\alpha < 1$.

In the same manner one can show that placing α-points in the span of $o(k)$ β-points will not improve on β^* asymptotically. But proving the result for general

configurations of α-points and finite n requires different ideas.

Theorem 13. *Suppose the image of $f : \mathbb{F}^k \to \mathbb{F}^n$ is a linear MDS code. If $n \geq 2k-1$ then there exists $x \in \mathbb{F}^k$ with $|x| = k$ such that $|f(x)| \leq n-k+1$. This implies that $\beta^* = 1 - \frac{1}{\rho}$.*

Proof. Suppose we have a collection B of l points in general position and an arbitrary collection A of m points inside $\mathbf{P}^r$. We claim that if $l \geq r+m$, there exists an $\mathbb{F}$-rational hyperplane containing r points in B and no points in A. Note that the desired result follows from this claim upon setting $m = k, r = k-1$.

We prove the claim by induction on r. When $r = 1$, each hyperplane is a point in $\mathbf{P}^1$. If $l \geq m+1$ there must exist a point in B that is not a point in A. Now suppose that the claim holds in dimension r. Take a collection B of $l \geq r+m+1$ points in general position inside $\mathbf{P}^{r+1}$. Since $l \geq r+m+1$, there must exist a point $p \in B$ such that $p \notin A$. We project the sets A and B from p down to $\mathbf{P}^r$. The image of B under this projection is a set B' consisting of $l-1$ points in general position. We have $l-1 \geq r+m$. Hence, by the inductive hypothesis, there exists a hyperplane in $\mathbf{P}^r$ that contains r points in B' and no points in the image of A. Lift this hyperplane by taking the cone over it that passes through p. This gives a hyperplane inside $\mathbf{P}^{r+1}$ that contains $r+1$ points of B and no points of A. This proves the claim. ■

Remark 6. Consider solving a system of linear equations $y = xG$ where G is a $k \times n$ matrix with Kruskal rank k (i.e., any k columns of G span a k-dimensional space). It is possible to find x with $|x| = k$ that satisfies some $k-1$ of the constraints.

For MDS codes of length $n \leq 2k-2$ we have the following result:

Theorem 14. *Suppose the image of $f : \mathbb{F}_q^k \to \mathbb{F}_q^n$ is a linear MDS code with $q > k$. If $k+1 \leq n < 2k-1$, there exists $x \in \mathbb{F}_q^k$ with $|x| \geq k - sk$ for all $0 \leq sk \leq 2k-n-1$ such that $|f(x)| \leq k - sk$. In other words, $\beta(\alpha) \leq \frac{1-s}{\rho}$ for all $\alpha < 1-s$.*

Proof. Consider the sets A of m arbitrary points and B of l points in general position inside $\mathbf{P}^r_{\mathbb{F}_q}$, where $m \leq l \leq m+r$. We claim that there exists a hyperplane containing $l-m$ points in B and no points in A if $q > m$.

Project the sets A, B successively from a point in B that is not in A. We repeat this step by projecting from a point in (the image of) B that is not in (the image of) A until no further such point exists. Note that we can project at least $l - m$ times (since the points of B are in general position). After this, we land in a projective space of dimension $r' \leq r - l + m$. The image A' of A under this projection is a set of cardinality m (counted with multiplicity). If $q > m$, there exists a hyperplane inside $\mathbf{P}^{r'}_{\mathbb{F}_q}$ that contains no point of A'. To see this, note that there are $\frac{q^{r'}-1}{q-1}$ hyperplanes inside $\mathbf{P}^{r'}_{\mathbb{F}_q}$. For a fixed point $p \in A'$, there are $\frac{q^{r'-1}-1}{q-1}$ hyperplanes that pass through p. By the union bound, if $\frac{q^{r'}-1}{q-1} > m\frac{q^{r'-1}-1}{q-1}$, there must exist a hyperplane that passes through no point of A'. We lift this hyperplane back into $\mathbf{P}^{r}_{\mathbb{F}_q}$. This will give a hyperplane passing through at least $l - m$ points of B and no points of A.

It further follows that for $s \leq r - (l - m)$ there exists a hyperplane containing $l - (m - s)$ points in B and no more than s points in A. Indeed one can remove a point $p \in A$ and apply the above argument to $A \backslash \{p\}$. ■

Remark 7. The bound $\beta(\alpha) \leq \frac{1-s}{\rho}$ is tight and is achieved if $\Gamma_\alpha \subset \Gamma_\beta$, that is, if the code is systematic. In fact, this result, combined with Theorem 13, can be used to characterize which $k \times (k-1)$ sub-matrices of G have full-weight elements in their left null space (as mentioned above) over large alphabets: if $G = [I|A]$ has full Kruskal rank, then a $k \times (k-1)$ sub-matrix of G has a full weight element in its left null space if and only if it is a submatrix of A. This follows from Theorems 1 and 2 and the fact that a shortened MDS code is still an MDS code.

7.4 Linear codes

In this section we give a converse bound on β^* for definition) as a function of δ for linear codes. Our bound is alphabet independent, and can be tight (over large alphabets). We prove some further (α, β)-limitations of the codes that achieve the bound and construct some examples of such codes. In particular, we show that if a code with positive distance achieves the bound, then there exists some x with relatively large weight for which $|f(x)| = \delta n$.

Theorem 15. *Let $f : \mathbb{F}_q^k \to \mathbb{F}_q^n$ be a linear code of relative minimum distance δ with $q > k$. Then there exists $x \in \mathbb{F}_q^k$ with $|x| = k$ such that $|f(x)| \leq \frac{n}{2}(1 + \sqrt{1 - \frac{4\delta}{\rho(1-\frac{1}{n})^2} + \frac{4}{n(1-\frac{1}{n})^2}}) + 1$. In other words,*

$$\beta^* \leq \frac{1}{2} + \frac{1}{2}\sqrt{1 - \frac{4\delta}{\rho}},$$

More generally, for $\alpha < 1$ we asymptotically have

$$\beta(\alpha) \leq 1 - \frac{1 + \frac{1}{\rho} - \frac{\alpha}{\rho}}{2}\big(1 - \sqrt{1 - \frac{4(1 - \alpha(1-\delta))}{\rho(1 + \frac{1}{\rho} - \frac{\alpha}{\rho})^2}}\big)$$

Furthermore, if $|f(x)| \geq n - t$ for all x with $|x| = k$ and some $t < k$, then there exists x with $|x| \geq t$ such that $|f(x)| \leq (n-t)\frac{t+1}{k}$. In other words, for all $\alpha < \rho(1 - \beta^*)$ we have $\beta(\alpha) \leq \rho\beta^*(1 - \beta^*)$. In particular, if a code achieves the above bound on β^*, then for all $\alpha < \frac{\rho}{2}(1 - \sqrt{1 - \frac{4\delta}{\rho}})$ we have $\beta(\alpha) = \delta$.

Proof. Consider two sets A of m arbitrary points and B of l points inside $\mathbf{P}^r_{\mathbb{F}_q}$ with the property that any hyperplane contains at most $l(1-\delta)$ fraction of the points in B. Consider successive projections of A, B from the points in (the image of) B that are not in (the image of) A. Note that, to project from a point p, we draw a line from p to every point (except for p) in A, B, and map that point to the intersection of the line with $\mathbf{P}^r$. Suppose that after s projections we can no longer find any B-point to further project from. We say that a B-point is lost in projection if its image is not defined (i.e., it lies on the point from which we project). Let λ be the number of points in $B \backslash A$ that are lost in the projections after t steps. Suppose the image of A contains m' unique points $p_1, \cdots, p_{m'}$ inside $\mathbf{P}^{r'}$ where $r' := r - t$. The image of B contains $l - \lambda$ points counted with multiplicities. Let b_i be the number of points in B that get mapped to p_i in the image of A. We may assume that $b_1 \geq b_2 \geq ... \geq b_{m'}$. On average, there are $c = \frac{l-\lambda}{m'} \geq \frac{l-\lambda}{m}$ points of B lying on top of a point in A. If we pick a hyperplane that passes through $p_1, \cdots, p_{r'}$ inside $\mathbf{P}^{r'}_{\mathbb{F}_q}$, it must contain at least $r'\frac{l-\lambda}{m}$ points in the image of B. We can lift this hyperplane back to $\mathbf{P}^r_{\mathbb{F}_q}$ to get a hyperplane containing at least $\lambda + \frac{(l-\lambda)(r-t)}{m}$ points in B. The assumption on B

requires that

$$\delta l \leq l - \left((l-\lambda)\frac{(r-t)}{m} + \lambda \right) \tag{7.1}$$

Using $\lambda \geq t$, we can write this as

$$t \geq \frac{\delta l m}{l-t} - m + r \tag{7.2}$$

This implies:

$$t \geq \frac{r+l-m}{2}\left(1 - \sqrt{1 - \frac{4l(-m(1-\delta)+r)}{(r+l-m)^2}}\right) \tag{7.3}$$

If $q > m$, there exists a hyperplane inside $\mathbf{P}^{r'}_{\mathbb{F}_q}$ that contains no point of A'. To see this, note that there are $\frac{q^{r'}-1}{q-1}$ hyperplanes inside $\mathbf{P}^{r'}_{\mathbb{F}_q}$. For a fixed point $p \in A'$, there are $\frac{q^{r'-1}-1}{q-1}$ hyperplanes that pass through p. By the union bound, if $\frac{q^{r'}-1}{q-1} > m\frac{q^{r'-1}-1}{q-1}$, there must exist a hyperplane that passes through no point of A'. Setting $l := n, m := k, r := k-1$, we get

$$\beta^* \leq 1 - \frac{t}{n} \leq \frac{1}{2} + \frac{1}{2}\sqrt{1 - \frac{4\delta}{\rho}} \tag{7.4}$$

as desired. In general, we can remove s points from A and apply the above argument to $A\backslash\{p\}$ so that for $\alpha < 1 - \frac{s}{k}$ we have

$$\beta(\alpha) \leq 1 - \frac{1+\frac{1-\alpha}{\rho}}{2}[1 - \sqrt{1 - \frac{4(1-\alpha(1-\delta))}{\rho(1+\frac{1}{\rho}-\frac{\alpha}{\rho})^2}}] \tag{7.5}$$

Now suppose that $f(x) \geq n - t$ for all x with $|x| = k$. Then the above sequence of projections must stop after t steps. Applying the same argument as above will prove the second part. ■

Remark 8. This result shows that there is a trade-off between the "smoothness" of a code and its ability to correct errors. The trade-off stems from two opposing tendencies: to correct errors a code needs to spread out messages while smoothness requires local structures (cf. [7]).

Remark 9. This result strengthens the connection between the (α, β)-property and the locality sensitive hashing (LSH) property. A priori, the (α, β)-property is only a

relaxation of the LSH condition (see [59]), in the sense that a map that is good in the (α, β)-sense sends far away messages to faraway codewords. This result suggests that such map must send some nearby messages to nearby codewords as well.

Remark 10. For MDS codes the above Theorem states that $\beta^* \leq 1 - \frac{1}{\rho}$ for $\rho \geq 2$ and $\beta^* \leq \frac{1}{\rho}$ for $\rho \leq 2$, which agrees with Theorems 13,14. The repetition code can asymptotically achieve $\delta = 0$ and $\beta^* = 1$. Thus the bound is tight at the two extreme points $\delta = 0, \delta = 1 - \frac{1}{\rho}$. The bound can be achieved at other values of δ as well.

Remark 11. It follows from the above proof that any linear code achieving $\beta^* = 1$ in the asymptotic regime (as $k \to \infty$) must be repetition-like, that is almost all columns of the generator matrix must have weight 1. Indeed the depth of the above projection sequence can be at most $o(k)$ for any such code. LDMCs on the other hand can achieve $\beta^* = 1$ asymptotically (or otherwise). Therefore they are superior to linear codes in this (admittedly weak) sense as well.

Remark 12. It is asked in [59] what codes can (asymptotically) achieve $\alpha = \beta$ when ρ is not an integer. It follows from our proof that such codes, if they exist, cannot be linear (over large alphabets). Indeed one can check that there are no repetition-like codes achieving $\alpha = \beta$ for non-integral ρ and any linear code achieving $\beta^* = 1$ is repetition-like as discussed above.

Remark 13. The Theorem states that the codes achieving the bound on β^* must send some heavy weight vectors to low weight codewords. This need not be true for codes in general. The second example below gives codes of relative distance $\delta > 0$ for which $|f(x)| > \delta n$ for all x with $|x| > 2$.

Problem 1. The bound of Theorem 15 can be tight when the alphabet size is large. It is a (hard) open problem to improve the bound over small alphabets.

Example 5. Here we present a non-MDS code with positive minimum distance that achieves the bound on β^* from Theorem 15. The proof of Theorem 15 suggests that such codes must look like a repetition code after a certain number of projections from β-only points. Take $\frac{k}{2}$ lines in $\mathbf{P}^{k-1}$ that are in general linear position. This means that any s lines are not contained in a $2s-2$-dimensional subspace. Place

three β-points and one α-point on each line. Place the other $\frac{k}{2}$ α-points in general position w.r.t the lines and place two β-points on each of them. The code has length $3\frac{k}{2} + 2\frac{k}{2} = \frac{5k}{2}$ and dimension k. One can check that this code has, asymptotically, $\delta = \frac{2}{5}$. By Theorem 15, $\beta^* \leq \frac{4}{5}$. A hyperplane that contains no α-points passes through at most one β-point from each of the lines. We can thus find a hyperplane passing through $\frac{k}{2}$ β-points and no α-point, but we cannot find a hyperplane passing through more β-point without containing an α-point. This gives $\beta^* = \frac{4}{5}$, which agrees with the bound of Theorem 15. Note that after $\frac{k}{2}$ projections from the chosen β-only points the code looks like the repetition code with $\rho = 2$.

Example 6. The Theorem above shows that the optimal codes (i.e., codes that achieve the bound on β^*) send some heavy weight messages to codewords of weight δn. This property need not hold for non-optimal codes. Here we give examples of codes for which $|f(x)| > \delta n$ for $|x| > 1$.

Consider maps f_1, f_2 where $f_1 : \mathbb{F}_q^k \to \mathbb{F}_q^n$ has distance $\delta_1 n$ and $f_2 : \mathbb{F}_q^{k-1} \to \mathbb{F}_q^n$ has distance $\delta_2 n$. Extend f_2 to a map on $\mathbb{F}_q^k$ by adding a zero row to its generator matrix. Construct a map $f : \mathbb{F}_q^k \to \mathbb{F}_q^{2n}$ sending $x \mapsto (f_1(x), f_2(x))$. Then f has distance $\delta_1 n$ but $|f(x)| \geq (\delta_1 + \delta_2)n$ for all x with $|x| > 1$.

Chapter 8

Maps over large alphabets

In this chapter we address the question of finding the best parameters that can be achieved for an $(\alpha, \beta)_q$-map. First, we present a converse result for such maps. Next, we present an achievability scheme for $(\alpha, \beta)_q$-maps which employ the extremal configuration characterized by Ahlswede and Khachatrian [3]. For large enough q this scheme is optimal as it attains the converse bound. Then we utilize Reed-Solomon codes to construct explicit optimal $(\alpha, \beta)_q$-maps for $q > n$.

8.1 Converse for $(\alpha, \beta)_q$-maps

Theorem 16. *Let $q \stackrel{\mathcal{N}}{=} 2$. Then, for an $(\alpha, \beta)_q$-map to exist, we must have*

$$h_q(\alpha) \stackrel{\mathcal{N}}{=} 1 - \min\{\rho R^q_{LP1}(\beta), \rho(1-\beta)\}. \tag{8.1}$$

Furthermore, for a sufficiently large field size q, we must have

$$\alpha \geq 1 - \rho + \rho\beta + o_q(1). \tag{8.2}$$

Proof. Let f be an $(\alpha, \beta)_q$-map. Assume that $\mathcal{C} \in \mathbb{F}_q^k$ is the maximum size code with relative distance α, i.e., $|\mathcal{C}| = A_q(n, \alpha n)$. Encoding each codeword in $\mathcal{C}$ with the map f, we get a set of vectors $f(\mathcal{C}) \in \mathbb{F}_q^k$ where any two vector are at Hamming distance

at least βn. Therefore, $f(\mathcal{C})$ is a code with relative distance β. This implies that

$$A_q(n, \alpha n) = |f(\mathcal{C})| \leq A_q(n, \beta n). \tag{8.3}$$

From the GV bound we know that

$$A_q(n, \alpha n) \stackrel{\mathcal{N}}{=} q^{k(1-h_q(\alpha))+o(k)}. \tag{8.4}$$

On the other hand, the linear programming bound in [1] ensures that

$$A_q(n, \beta n) \leq q^{nR^q_{LP1}(\beta)+o(n)}, \tag{8.5}$$

where

$$R^q_{LP1}(\delta) = h_q(\frac{q-1}{q} - \beta\frac{q-2}{q} - \frac{2}{q}\sqrt{\beta(1-\beta)(q-1)}).$$

Now, the bound in (8.1) follows by using (8.4) and (8.5) in (8.3). We obtain the bound in (8.2) from the fact that for a large enough q we have $h_q(a) = a + o_q(1)\ \forall a \in [0, 1]$. ■

8.2 Achievability scheme for $(\alpha, \beta)_q$-map

Here we present an achievability scheme to construct an $(\alpha, \beta)_q$-map. One approach to construct (α, β)-maps over $\mathbb{F}_q$ is as follows. Cover $\mathbb{F}_q^k$ with configurations of diameter at most αk. Pack in $\mathbb{F}_q^n$ as many points (codewords) of pairwise distance more than βn as there are configurations in the cover. Then map configurations to codewords. To obtain good (α, β) properties, it makes sense to look for configurations that contain a large number of points, for having fewer codewords leads to better separation. A natural choice is to cover with Hamming balls. The Hamming balls are too small when $q > 2$ and, hence, do not give satisfactory (α, β) properties. Thus, one can ask for the shape and cardinality of extremal configurations. This was a long-standing combinatorial problem and settled by the diametric theorem of Ahlswede and Khachatrian [3], which we quote below.

We are interested in large subsets with bounded diameter. The cardinality of the

largest such subset is

$$N_q(d,k) = \max\{|\mathcal{A}| : \mathcal{A} \subset \mathbb{F}_q^k \text{ s.t. } \operatorname{diam}(\mathcal{A}) \leq d\}.$$

For $\boldsymbol{x} \in \mathbb{F}_q^k$, define $J(\boldsymbol{x}) = \{j : x_j = 0\}$ and

$$\mathcal{U}_i = \{\boldsymbol{x} \in \mathbb{F}_q^k : |J(\boldsymbol{x}) \cap [1, k-d+2i]| \geq k-d+i\}.$$

Note that each set $\mathcal{U}_i$ can be written as a cartesian product of some $(k-2d+i)$-dimensional ball of radius i with $\mathbb{F}_q^{d-2i}$. In particular, $\mathcal{U}_0$ is a low dimensional cube $\mathbb{F}_q^\rho$ inside $\mathbb{F}_q^k$. We are now ready to state the diametric theorem:

Proposition 17 (The diametric theorem[3])**.** *Let r be the largest integer such that*

$$k-d+2r < \min\{k+1, k-d+2\frac{k-d-1}{q-2}\}.$$

Then $N_q(k,d) = |\mathcal{U}_r|$.

We will make use of the extremal configurations that appear in the theorem in the covering step mentioned above. We state the achievable parameters in the following result. For large q, this result establishes the tightness of the converse bound in Theorem 16.

Theorem 18. *Fix $q \geq 2$ and set*

$$\bar{\rho} := \begin{cases} \frac{\rho(q-2)}{q(1-h_q(1/q))} & q > 2 \\ \rho & q = 2 \end{cases}$$

Then for all $\beta \leq 1 - \frac{1}{q}$ and $k_i, n_i \to \infty$ with $\frac{n_i}{k_i} \to \rho$, there exists (α_i, β_i)-maps $f : \mathbb{F}_q^{k_i} \to \mathbb{F}_q^{n_i}$ with $(\alpha_i, \beta_i) \to (\alpha, \beta)$ if

$$\alpha \geq \max\{\frac{2}{q}, 1 - \bar{\rho} + \bar{\rho} h_q(\beta)\}\} \text{ or } h_q(\alpha/2) \geq 1 - \rho + \rho h_q(\beta) \tag{8.6}$$

Proof. Let $r(k,d)$ be the integer as in the diametric theorem. We cover $\mathbb{F}_q^k$ with

$\mathcal{U}_{r(k,d)}$'s. Write

$$\mathcal{U}_{r(k,d)} = B_{r(k,d)} \times \mathbb{F}_q^{d-2r(k,d)}.$$

Define $t(k,d) = k - d + 2r(k,d)$. Now we can mod out the second factor so that covering $\mathbb{F}_q^k$ with translates of $\mathcal{U}_{r(k,d)}$ reduces to covering $\mathbb{F}_q^{t(k,d)}$ with Hamming balls of radius $r(k,d)$. Define

$$K(d,X) = \min\{m : \cup_{i=1}^m S_i = X, \operatorname{diam}(S_i) = d\}$$

and

$$W(r,X) = \min\{m : \cup_{i=1}^m B_i = X, \operatorname{rad}(B_i) = r\}.$$

Given a linear code $C \subset X$, denote its covering radius by $r_{\text{cov}}(C)$ and further define

$$w(r,X) = \min\{m : \text{there is an } m\text{-dimensional linear code } C \subset X \text{ with } r_{\text{cov}}(C) \leq r\}.$$

We have $W(r,\mathbb{F}_q^t) \leq q^{w(r,\mathbb{F}_q^t)} \leq q^{t(1-h_q(r/t))+O(\log t)}$ where the second inequality is from [17]. We can thus bound the number of configurations of diameter d needed to cover $\mathbb{F}_q^k$ as follows

$$K(d,\mathbb{F}_q^k) \leq W(r(k,d),\mathbb{F}_q^{t(k,d)}) \leq q^{w(r(k,d),\mathbb{F}_q^{t(k,d)})} \leq q^{t(k,d)(1-h_q(\frac{r(k,d)}{t(k,d)}))+O(\log t(k,d))}$$

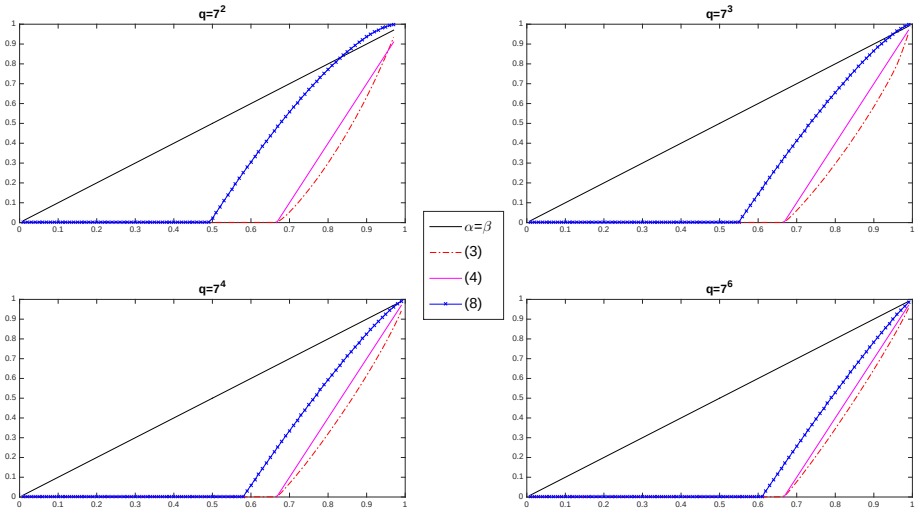
Using the GV bound, we can see that (α,β) is achievable asymptotically if

$$t(k,\alpha k)(1 - h_q(\frac{r(k,\alpha k)}{t(k,d)})) \leq n(1 - h_q(\beta))$$

holds as $k \to \infty$. Setting $\overline{r} := r(k,d)/k$, we can rewrite the above in the form

$$(1 - \alpha + 2\overline{r})(1 - h_q(\frac{\overline{r}}{1-\alpha+2\overline{r}})) \leq \rho(1 - h_q(\beta))$$

Furthermore, note that for $\alpha \geq 2/q$ and $q > 2$, we have $\bar{r} = \frac{1-\alpha}{q-2}$ for large k. Hence



the achievable region contains the curve

$$\frac{q(1-\alpha)}{q-2}(1-h_q(\frac{1}{q})) = \rho(1-h_q(\beta))$$

when $\alpha \geq \frac{2}{q}$. In other words

$$h_q(\beta^*(\alpha,\rho,q)) \geq 1 - \frac{1}{\bar{\rho}} + \frac{\alpha}{\bar{\rho}} \tag{8.7}$$

if $\alpha \geq \frac{2}{q}$. On the other hand, when $\alpha < \frac{2}{q}$ we have $\bar{r} := \frac{\alpha k}{2}$. Covering $\mathbb{F}_q^k$ with Hamming balls of radius $\alpha k/2$ gives a lower bound

$$h_q(\alpha/2) \leq 1 - \rho + \rho h_q(\beta) \tag{8.8}$$

on achievable β's. This proves (8.6). ■

Fig. 1 shows the bounds in (8.6)-(8.2) for some finite values of q.

8.3 Truncated Reed-Solomon codes

Here we give an explicit family of codes that achieve optimal (α,β)-trade-offs for $q \geq \rho k$. Set $\bar{\alpha} := 1-\alpha$ and consider the Reed-Solomon code $f_{RS} : V \to \mathbb{F}_q^{\rho k}$ where V

is the subspace of $\mathbb{F}_q^k$ formed by its first $\bar{\alpha}k$ coordinates. Note that f_{RS} is a $(0, 1-\frac{\bar{\alpha}}{\rho})$-map. Now let π_V be the projection to V map and define the truncated Reed-Solomon (TRS) code $f_{TRS} : \mathbb{F}_q^k \to \mathbb{F}_q^{\rho k}$ as follows: $f_{TRS}(\boldsymbol{x}) = f_{RS}(\pi_V(\boldsymbol{x}))$. Any vector $\boldsymbol{x} \in \mathbb{F}_q^k$ with $\mathrm{wt}(\boldsymbol{x}) > \alpha k$ projects to a non-zero vector in V. Hence

$$\mathrm{wt}(\boldsymbol{x}) > \alpha k \implies |f_{TRS}(\boldsymbol{x})| > (1 - \frac{\bar{\alpha}}{\rho})\rho k,$$

which means that f_{TRS} is a $(\alpha, 1-\frac{\bar{\alpha}}{\rho})$-map. Furthermore, when $q \geq \rho k$, one can check that the bound in (8.2) is sharp even for finite k. Thus, the TRS parameters are optimal over large enough alphabets.

Chapter 9

Explicit constructions for short codes

In this chapter we study short linear codes. These are some of our attempts at constructing good graceful codes prior to the development of LDMCs. After presenting the preliminary background, we first discuss some natural notions of (α, β)-optimality that are relevant for graceful degradation. We then propose a method for constructing linearly optimal short codes using generalized Macwilliams identities. We apply this method to construct optimal codes. We also construct an optimal code using algebraic ideas. The codes that we construct here are short and have smooth (α, β)-profiles. We observe empirically that they are graceful for stochastic noise as well. We discuss some algebraic geometric codes at the end.

9.1 Preliminaries

In this section we briefly review the background material needed for this chapter. Both our designs and analysis rely heavily on generalized Macwilliams identities and the linear programming bound of coding.

9.1.1 Macwilliams identities for (α, β)-maps

Let C be the graph of an (α, β)-map inside $\mathbb{F}_2^k \times \mathbb{F}_2^n$. Define

$$g(u) = x^{\mathrm{wt}(u_\alpha)} y^{\mathrm{wt}(u_\beta)}$$

where wt denotes the Hamming weight. Let

$$W_C(x,y) = \sum_{u_\alpha, u_\beta \in C} g(u)$$

be the bi-weight enumerator of the map. Then

Proposition 19. *The following (Macwilliams') identity holds:*

$$W_{C^\perp}(x,y) = \frac{1}{|C|}(1+x)^k(1+y)^n W_C(\frac{1-x}{1+x}, \frac{1-y}{1+y})$$

Proof. Proceeding in the same manner as in Macwilliams' proof, we first compute the Hadamard transform of g

$$\begin{aligned}
\hat{g}(u) &= \sum_{v\in\{0,1\}^n} (-1)^{u.v} x^{\mathrm{wt}(v_\alpha)} y^{\mathrm{wt}(v_\beta)} \\
&= \sum_{v\in\{0,1\}^n} (-1)^{\sum_i u_{\alpha_i}.v_{\alpha_i} + \sum_j u_{\beta_j}.v_{\beta_j}} x^{\sum_i v_{\alpha_i}} y^{\sum_j v_{\beta_j}} \\
&= \sum_{v\in\{0,1\}^n} \prod_{i\le k} (-1)^{u_{\alpha_i}.v_{\alpha_i}} x^{v_{\alpha_i}} \prod_{i>k} (-1)^{u_{\beta_i}.v_{\beta_i}} y^{v_{\beta_i}} \\
&= \sum_{v\in\{0,1\}^n} \prod_{i\le k} (-1)^{u_{\alpha_i}.v_{\alpha_i}} x^{v_{\alpha_i}} \prod_{i>k} (-1)^{u_{\beta_i}.v_{\beta_i}} y^{v_{\beta_i}} \\
&= \sum_{v\in\{0,1\}^n} \prod_{i,j} (-1)^{u_{\alpha_i}.v_{\alpha_i}} x^{v_{\alpha_i}} (-1)^{u_{\beta_j}.v_{\beta_j}} y^{v_{\beta_j}} \\
&= \prod_{i\le k} \sum_{s\in\{0,1\}} (-1)^{s u_i} x^s \prod_{i>k} \sum_{s\in\{0,1\}} (-1)^{s u_i} y^s
\end{aligned}$$

The inner sums are $1+x, 1+y$ when $u_i = 0$ and $1-x, 1-y$ otherwise. Hence

$$\hat{g}(u) = (1+x)^{n-\mathrm{wt}(u_\alpha)}(1+y)^{n-\mathrm{wt}(u_\beta)}(1-x)^{\mathrm{wt}(u_\alpha)}(1-y)^{\mathrm{wt}(u_\beta)}.$$

Replace this in the Fourier inversion formula

$$\sum_{u \in C^\perp} g(u) = \frac{1}{|C|} \sum_{u \in C} \hat{g}(u)$$

to attain

$$W_{C^\perp}(x,y) = \frac{1}{|C|}(1+x)^k(1+y)^n W_C(\frac{1-x}{1+x}, \frac{1-y}{1+y})$$

■

9.1.2 Bivariate Krawchouk polynomials

We may write the bivariate Macwilliams identity as follows:

$$\sum_{l=0,m=0}^{n-k,n} A'_{lm} x^l y^m = \frac{1}{2^n} \sum_{i=0,j=0}^{k,n} A_{ij}(1+x)^{k-i}(1+y)^{n-j}(1-x)^i(1-y)^j$$

We can expand the inner summand on the right hand side:

$$(1-x)^i(1+x)^{k-i}(1-y)^j(1+y)^{n-j} = \sum_{l,m=0}^{k,n} P_{lm}(i,j)x^l y^m$$

where $P_{lm}(i,j) = P_l(k,i)P_m(n,j)$. In this notation, $P_l(k,x)$ is the Krawchouk polynomial defined here:

$$P_l(k,x) = \sum_{s=0}^{i} (-1)^s \binom{i}{s} \binom{k-x}{l-s}$$

One can thus write

$$A'_{lm} = \frac{1}{2^n} \sum_{i,j=0}^{k,n} P_{lm}(i,j) A_{ij}$$

This implies that a certain linear combination of A_{ij}'s need be non-negative for a linear (α, β)-map to exist.

9.1.3 Generalized linear programming bounds

By the above discussion, the spectrum of any linear (α, β)-map must satisfy the following set of constraints:

$$\begin{aligned} &\sum_{i,j=0}^{k,n} P_{lm}(i,j)A_{ij} \geq 0, \quad \forall l \leq n-k, m \leq n \\ &A_{00} = 1, \quad A_{ij} \geq 0, \quad \sum_{j} A_{ij} = \binom{k}{i}, \quad \sum_{j \leq \beta(\frac{i}{k})n} A_{ij} = 0; \end{aligned} \tag{9.1}$$

To bound the size of a candidate code one can vary k and check the feasibility of the above set of linear constraints. In the next section, we use this technique to prove optimality of a certain quasi-cyclic construction. Before that we shall present the linear programming problem in its dual form. Relaxing $(*)$, we define

$$\mathcal{L} = 1 + \sum_{i \geq 1, j > \beta(i)}^{i \leq k, j \leq n} A_{ij} + \sum_{i \geq 1, j > \beta(i)}^{i \leq k, j \leq n} A_{ij} \sum_{l \leq n-k, m \leq n} \lambda_{lm} P_{lm}(i,j) + \sum_{l \leq n-k, m \leq n} \lambda_{lm} P_{lm}(0,0)$$

The dual is

$$\begin{aligned} &\min\ 1 + \sum_{l \leq n-k, m \leq n} \lambda_{lm} P_{lm}(0,0) \\ &s.t.\ 1 + \sum_{lm} \lambda_{lm} P_{lm}(i,j) \leq 0 \quad \forall i \geq 1, j > \beta(i) \end{aligned}$$

Thus, if one finds a polynomial of the form $Q(x,y) = 1 + \sum_{l \leq n-k, m \leq n} \lambda_{lm} P_l(x) P_m(y)$ with $\lambda_{l,m} \geq 0$ such that $Q(i,j) \leq 0$ for $i \geq 1, j > \beta(i)$ then the value of the linear program is bounded above by $Q(0,0)$ (using the duality theorem). it is easy to deduce the linear programming bound from here upon noticing that $\alpha(y)$ with

$$\alpha(y) = \frac{1}{a-y}\{P_t(a)P_{t+1}(y) - P_{t+1}(a)P_t(y)\}^2$$

for propers choices of t and $a \leq \beta(0)$ satisfies these conditions. In other words we can take $\lambda_{lm} = 0$ for $l > 0$ (noting that $P_0(x) = 1$).

9.2 Strong and weak (α, β)-optimality

We consider two notions of (α, β)-optimality. One of them requires one to compare (α, β)-properties of codes with different dimension. In such settings, it becomes useful to have an absolute version of $\beta(\alpha)$. We define

$$A_i^*(f) := \inf\{|f(x) - f(y)| : |x - y| \geq i\} \tag{9.2}$$

Definition 14 (Weakly optimal maps)**.** *A code $f : \mathbb{F}_q^k \to \mathbb{F}_q^n$ is said to be weakly (α, β)-optimal if there does not exist $f' : \mathbb{F}_q^{k+1} \to \mathbb{F}_q^n$ such that*

$$A_i^*(f') \geq A_i^*(f) \quad \forall i \leq k$$

In other words, a code f is weakly optimal if no code with larger dimension can achieve the same or better $A_i^*(f)$'s.

Definition 15 (Strongly optimal maps)**.** *A code $f : \mathbb{F}_q^k \to \mathbb{F}_q^n$ is said to be strongly (α, β)-optimal if it is not dominated by any other code, i.e., there does not exist an code $f' : \mathbb{F}_q^k \to \mathbb{F}_q^n$ such that*

$$A_i^*(f') \geq A_i^*(f) \quad \forall i \leq k$$

where at least one inequality is strict.

The examples below show that weak optimality is indeed strictly weaker than strong optimality. For the reverse direction, we have the following result:

Proposition 20. *A strongly optimal map is weakly optimal.*

In other words, if there exist a larger code that achieves the same (α, β)-profile as f, then f cannot be strongly optimal. Before we present the proof we remark that the analogous statement for minimum distance is false. Indeed, a code maybe optimal in the sense of minimum distance, yet, there may exists a larger code that achieves

the same minimum distance. For instance, Tanner [75] constructed a binary [12,4,6]-code. The linear programming (LP) bound rules out the existence of a [12,3,7]-code. Thus any [12,3,6]-subcode of the Tanner code is still optimal in the sense of minimum distance. In general, one can expect such codes to exists over any field where the singleton bound is not tight. Over such fields, the existence of an $[n, k+1, d]$-code need not imply the existence of an $[n, k, d+1]$-code. However, the above proposition states that the existence of an $[n, k+1]$-code implies the existence an $[n, k]$-code with improved A_i^*'s.

Proof (of Proposition 20). Suppose a strongly optimal $f : \mathbb{F}_q^k \to \mathbb{F}_q^n$ is weakly dominated by $f' : \mathbb{F}_q^{k+1} \to \mathbb{F}_q^n$. Take the 1st coordinate and select the most common symbol among the codewords of f'. Take all the codewords of f' that start with this common symbol and remove the rest of the codewords. Now shorten the code by removing the first coordinate. This gives an $(n-1, k)$-subcode of f' with the same A_i^*'s as f. Now define an extension of f' as follows: $f''(x) := (f'(x), x_1)$ where x_1 is the first input coordinate. Clearly, all messages x, x' with $d(x, x') = k$ are sent to codewords that have distance $|f''(x) - f''(x')| = 1 + |f(x) - f(x')|$. This violates strong optimality of f.

■

Remark 14. The proof essentially relies on the fact that $A_k^*(f)$ can be improved if f is not weakly optimal. Conversely, it can be shown that any map f that achieves $A_k^*(f) = n$ is weakly optimal.

Remark 15. The same result can be proved within the class of linear maps, i.e., for a linear $[n, k]$-map f there exists a linear $[n-1, k-1]$-map f' such that $A_i^*(f) = A^*(f')$ for $i \leq k-1$.

9.3 A weakly optimal quasi-cyclic code

Here we present a code that is optimal in the weak sense. Let $C_\beta \subset \mathbb{F}_2^7$ be the [7,4]-cyclic code generated by the primitive polynomial $x^3 + x + 1$. Similarly, C_{β^3} denotes

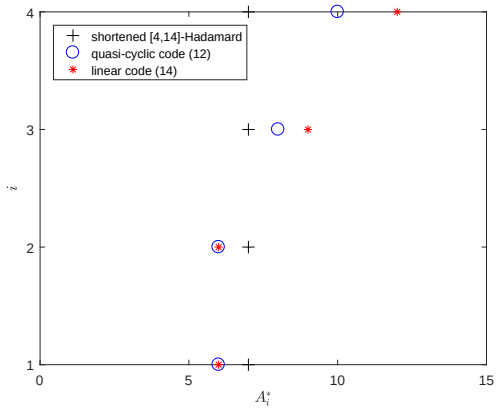


Figure 9-1: The A_i^* profiles for three codes: 1) the strongly optimal shortened [4,14,7] Hadamard code 2) the weakly optimal quasi-cyclic code of (9.4) 3) the strongly optimal linear code of (9.6).

the code generated by the primitive polynomial of β^3 (which is x^3+x^2+1). Consider the code

$$C = \{(x, y)|x \in C_\beta, y \in C_{\beta^3}\} \tag{9.3}$$

The code has a minimum distance of 6. One can check that after applying a linear transform $x \to x + x^2$, the resulting spectrum contains the following (α, β) pairs: $A_1^* = A_2^* = 6, A_3^* = 8, A_4^* = 10$ (see (9.2) for the definition of A_i^*), with the following generating matrix:

$$G = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{9.4}$$

Under these (α, β)-constraints, the LP in (9.1) becomes infeasible for $k = 5$. This implies that no $[14, 5]$-code exists with the same (or better) (α, β)-properties. We note that relaxing any of the (α, β)-constraints in the linear program will render the LP feasible with $k = 5$. This code is optimal in the weak sense but not in the strong sense as the construction below shows. We extend this code by appending the column $c := [0, 1, 0, 1]'$ to its generating matrix so it has comparable length with the

Hadamard code. It becomes a $[15, 4]$-code with the following generating matrix:

$$G_e = \left[\begin{array}{c|c} G & c \end{array} \right] \tag{9.5}$$

After the extension, the code contains the following (α, β)-pairs:$A_1^* = 6, A_2^* = 7, A_3^* = 9, A_4^* = 11$.

9.4 A strongly optimal [14,4]-linear code

We ask if there exists a $[14, 4]$-code that dominates the quasi-cyclic code of (9.4). The LP in (9.1) is infeasible if we set $A_2^* = 7$ while keeping the rest of A_i^*'s from above unchanged. However, one can ask if there exists a code with the following profile $A_1^* = 6, A_2^* = 6, A_3^* = 9, A_4^* = 12$. The space of $[14, 4]$-linear codes is too big to search over. The LP in (9.1) can help reduce the size of the search space by severely restricting A_{1j}'s. With the above A_i^*'s, it turns out that the LP is infeasible when $A_{16} < 3$. This means that such a linear code can exists only if at least three of the rows in its generating matrix have weight 6. We can now efficiently search over the space of linear codes with $A_{16} = 3$ after taking out the symmetries. Here is the generator matrix of a code that was found using computer search over the reduced search space:

$$G = \left[\begin{array}{cccccccccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{array} \right] \tag{9.6}$$

The LP and some mild extra work suffice to prove that this code is optimal in strong sense. We also extend this code by adding a column $c = [1, 0, 0, 0]'$ to its generating matrix to make it have the same length as the Hadamard code:

$$G_e = \left[\begin{array}{c|c} G & c \end{array} \right] \tag{9.7}$$

The corresponding BER profile when used in communication over BSC is shown in Fig. 9-2. It can be seen that for a wide range of channel parameters p the code of (9.7) outperforms both the quasi-cyclic code of (9.5) and the Hadamard code. We note that the $[15, 4, 8]$ Hadamard code is also optimal in the strong sense, as is the shortened $[14, 4, 7]$ Hadamard code. While the BER differences may seem marginal, we expect to see more significant improvements for larger codes.

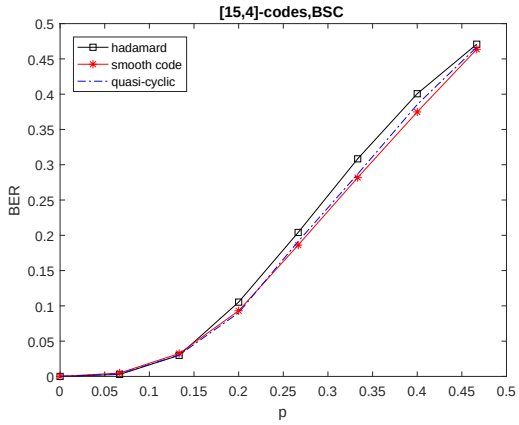


Figure 9-2: The BER profiles under bitMAP decoding for: 1) the [15,4] Hadamard code 2) the [15,4] extended quasi-cyclic code of (9.5) 3) the extended linear code of (9.7).

9.5 Impossiblity results for quasi-cyclic codes

Let $n = \rho k$ with $\rho \in \mathbb{Z}$. We consider linear maps $f : \mathbb{F}_q^k \to \mathbb{F}_q^n$ that are equivariant w.r.t cyclic shifts $T_1 : \mathbb{F}_q^k \to \mathbb{F}_q^k$ and $T_2 : \mathbb{F}_q^n \to \mathbb{F}_q^n$, i.e., $f \circ T_1 = T_2 \circ f$. Any such map, under $\varphi_f : c \mapsto (c, f(c))$, gives rise to a k-dimensional subspace of $\mathbb{F}_q^{n+k}$ that is stable under (T_1, T_2) and vice versa. Set $A_j := \mathbb{F}_q[x]/(x^j - 1)$. We are thus led to study submodules of $R := A_k \oplus A_n$ generated by $1 \mapsto (1, g(x))$ as an A_k-module. We remark that $1 \mapsto (1, g)$ generates an A_k-module inside R if and only if $(x^k - 1)g$ is zero in A_n. Thus we get a correspondence between (T_1, T_2)-stable subspaces of $\mathbb{F}_q^{n+k}$ and polynomials $g \in F_q[x]$ such that $g(x^k - 1) = h(x^n - 1)$. We next list some (α, β) properties of φ_f.

Proposition 21. *If f has the same image as the repetition map, then $\beta \leq \alpha$.*

Proof. f can be written as a composition $r \circ \pi$ where r is the repetition map and $\pi : \mathbb{F}_q^k \to \mathbb{F}_q^k$ is an automorphism. This means that every $c \in \mathbb{F}_q^k$ is sent to $r(c')$ for some $c' = \pi(c)$. Thus $\beta > \alpha$ under f if and only if $\beta > \alpha$ under π. However, no π can achieve $\beta > \alpha$ since the set $\{x : |x| \geq \beta\}$ has smaller cardinality than $\{x : |x| \geq \alpha\}$ when $\beta > \alpha$. ■

By Eisenstein's criterion, $\frac{x^p-1}{x-1}$ is irreducible over $\mathbb{Z}$ for any prime p. The next conjecture states that it will be irreducible for infinitely many primes over any fixed finite field.

Conjecture 2. There are infinitely many primes p such that $\frac{x^p-1}{x-1}$ is irreducible over $\mathbb{F}_q$ (here q is arbitrary, not necessarily a power of p).

The conjecture is not true for arbitrary q, and when q is an odd power, it boils down to a classic conjecture in number theory, known as the Artin's conjecture.

It is a basic fact that a polynomial $f(x) \in F[x]$ is irreducible over a splitting field E/F iff the Galois group $G(E/F)$ acts transitively on the roots of f (reason: the roots that are conjugate under the Galois action have the same minimal polynomial. We thus need all the roots to be in the same orbit under $\zeta \to \zeta^q$). The Artin map gives an isomorphism of the Galois group of the cyclotomic field $G(\mathbb{Q}(\zeta_p)/\mathbb{Q})$ with $(\mathbb{Z}/p\mathbb{Z})^*$ sending the Frobenius elements corresponding to (unramified) primes $q \neq p$ to $\bar{q} \in (\mathbb{Z}/p\mathbb{Z})^*$. Now to study the irreducibility of ϕ_p over $\mathbb{F}_q$ we need to work with Galois group of the residue field of the cyclotomic extension (mod q). This is a subgroup (known as the decomposition group) of $G(\mathbb{Q}(\zeta_p)/\mathbb{Q})$ and is generated by the Frobenius element corresponding to q (note that raising to power of q leaves $\mathbb{F}_q$ fixed). Thus, to get a transitive action we need all the p-th roots of unity to fall in the same orbit of under $\zeta \to \zeta^q$. Hence, for the conjecture to be true, we want q to be a primitive root modulo infinitely many primes p, which is the statement of Artin's conjecture when q is an odd power. This is known to be true under GRH (shown in [29]). The unconditional results are much weaker, e.g. [57] shows that the set of integers $E(x) = \{q < x | q \text{ is not a perfect square}\}$ for which Artin's conjecture fails has cardinality $O(\log^6 x)$ and contains no more than 6 primes. It is not known which

6 these are.

Remark. There are infinitely many q's for which $\frac{x^p-1}{x-1}$ is irreducible for a fixed prime p (reason: all we need is that q has maximal order modulo p, i..e, $q^r \not\equiv 1 \bmod p$ for any $r < p-1$. It follows from Dirichlet's theorem on arithmetic progressions that infinitely many such q's exist.). Here we need the converse.

Part b of the following is a consequence of the conjecture.

Proposition 22. *(a) Suppose that f is injective. Then f has the same image as the repetition map.*

(b) There are infinitely many primes k such that $\alpha \geq \beta$ for all $f : \mathbb{F}_q^k \to \mathbb{F}_q^n$.

Proof. Write $g = \frac{x^n-1}{x^k-1}h$. Injectivity of f requires that $x^k - 1|i$ whenever $x^n - 1|ig$. Hence $\gcd(x^k - 1, h) = 1$ and thus we can replace g with g/h without changing the image of the code. But $g/h = \frac{x^n-1}{x^k-1} = (x^k)^{\rho-1} + \cdots + 1$, which corresponds to the repetition map. This proves part (a).

To prove part (b), suppose that $\gcd(h, x^k - 1) = 1$. Then again we can replace g with g/h and observe, as in part (a), that the code has the same image as the repetition map and thus $\alpha \geq \beta$ by Proposition 1. Otherwise, by the above conjecture, there are infinitely many primes k such that either $\gcd(h, x^k - 1) = x - 1$ or $\gcd(h, x^k - 1) = x^{k-1} + \cdots + 1$. In the former case, the codeword $x^{k-1} + \cdots + 1$ has Hamming weight $k-1$ and is sent to zero, which gives $\beta < \alpha$ for all $\alpha < \frac{k-1}{k}$. In the latter case, $(x-1)$ and hence all codewords of the form $i = (x-1)c(x)$ are sent to zero. In particular, we can take $c(x) = x^{k-1} + x^{k-3} + \cdots + 1$. Then $c(x)(x-1) = x^{k-1} + x^{k-2} + \cdots + x$ has hamming weight $k-2$ and is sent to zero. In either case, we have $\beta < \alpha$ for all $\alpha < 1$ asymptotically as $k \to \infty$. ∎

9.6 Codes from Cayley-Bacharach

We start with a simple statement:

Theorem 23. *Let C be a smooth plane curve of degree d. Let $D = \sum_{i=1}^{e} p_i$ be a divisor of degree e. Suppose that $D' = \sum_{i=1}^{e} q_i$ is rationally equivalent to D. If $e \leq d-2$, then we have $p_i = q_i$.*

Proof. Set $D := \sum p_i$ and $E := \sum q_i$. Suppose D and E are distinct divisors with $\mathrm{supp}D \cap \mathrm{supp}E = \emptyset$. If $D \sim E$ then both D and E can realized as the zero locus of two different sections of the line bundle $O(E)$. In other words, $H^0(O(E)) \geq 2$. This implies, by Riemann-Roch, that

$$H^0(O(K_C - E)) > g - e$$

Thus the statement follows if we show that for all divisors E of degree at most $d-2$ we have

$$H^0(O(K_C - E)) \leq g - e$$

The adjunction formula gives that

$$O(K_C) = O_{\mathbb{P}^2}(d-3)|_C = O_C(d-3)$$

We thus have an injection

$$H^0(O_{\mathbb{P}^2}(d-3)) \hookrightarrow H^0(O(K_C))$$

Note further that

$$H^0(O_{\mathbb{P}^2}(d-3)) = \binom{d-1}{2} = g.$$

We also know that $O(K_C)$ has g global sections. By dimension count, we see that the restriction map induces an isomorphism on global sections. Hence differential forms on C that vanish along E come from restricting plane curves of degree $d-3$ that pass through E. Thus we need to count how many conditions E imposes on curves of degree $d-3$.

We claim that E imposes exactly e conditions provided that $e \leq d-2$. Suppose $d-2$ points are given in the plane. Fix a subset $\Gamma \subset E$ with $|\Gamma| = d-3$. It suffices to

find a curve of degree $d-2$ that passes through Γ but not $E-\Gamma$. By induction, we can pass a curve of degree $d-3$ through all but one point of Γ. Then we can easily pass a line through the remaining point that does not intersect $E-\Gamma$. This proves the claim. Hence $H^0(O(K_C-E)) = H^0(O_{\mathbb{P}^2}(d-3)) - e = g - e$ as desired. ■

The next example shows that the bound $e \leq d-2$ is sharp.

Example 7. Let p_i's be three collinear points on a smooth quartic C. Let L be the line through p_i's and $r \in C$ be the remaining point where the L meets C. Take a line distinct from L that passes through r. Let q_i's be the three other points where the line meets C. Then $\sum p_i + r \sim \sum q_i + r$.

We get a version of Cayley-Bacharach's theorem as an immediate corollary of our theorem:

Corollary 24. *Let $\Gamma = C \cap C'$ be the intersection of two smooth plane curves of degrees d, d', respectively. Then if some smooth plane curve C'' with $\deg C'' = d'$ passes through all but $d-2$ points of Γ, it must contain Γ.*

Proof. We have that $[C'' \cap C] \sim \Gamma$. Now apply Theorem 1. ■

The above result maybe useful in designing short (α, β)-maps.

Example 8. Let Γ be the intersection of two smooth quartics. Embed $\mathbb{P}^2$ into $\mathbb{P}^{14}$ via the Veronese embedding. Take the image of Γ to be the set of β-points of a code. Take a maximal subset of Γ and complete it to a basis for $\mathbb{P}^{14}$. Let these be the α-points. Then any hyperplane containing at least 14 β-points must contain Γ. The resulting code has $\rho = \frac{16}{15}$,$\alpha = \frac{1}{15}$,$\beta = \frac{1}{8}$. We note that the parameters lie on the asymptotically optimal line $\alpha = 1 - \rho + \rho\beta$.

While the above construction does not give any better parameters than the non-injective RS coding (RS coding on a subset), it does achieve the same performance over a much smaller field. It is possible to make the above construction work over finite fields so long as there are enough points on the plane. We can check this happens when $q = 4$ in the above example. An RS type code would require $q = 15$. In general, we can see that q grows with $\sqrt{n}$ instead of n in this construction.

Naturally, one can ask if there are similar constructions that work for larger codes. Unfortunately, this cannot be done with Cayley-Bacharach type theorems. The above construction, in the worst case, can have $\rho \to 2$ but $\beta \to 0$ as d grows. The problem lies in the fact that we only need some smooth curves and some sets of good divisors on them to construct good codes, whereas Cayley-Bacharach type theorems concern all curves and all divisors! We explain the difference next.

Given a smooth curve C of degree d, we can associate a variety to its pairs of effective divisors of degree $W_e := C^e \times C^e$. We are interested in pairs $\tilde{W}_e \subset W_e$ that correspond to equivalent divisors. Clearly, $\tilde{W}_e$ contains the diagonal $\Delta_e := \{(x, y) : x, y \in C^e, x = y\}$. When $e \leq d-2$, our theorem shows that $\tilde{W}_e = \Delta_e$, and example 1 shows that when $e > d-2$ we have $\Delta \subsetneq \tilde{W}_e$. Note however, that the three divisors for which we found a counter-example are special in that they are collinear. We cannot find a distinct equivalent pair for most other effective divisors of degree 3. In general, we claim that for a generic divisor of degree $e \leq g$, there are no non-trivial equivalent divisors.

Theorem 25. *Let C be a smooth plane curve of degree d. Let $D = \sum_{i=1}^{e} p_i$ be a generic divisor of degree e. Suppose that $D' = \sum_{i=1}^{e} q_i$ is rationally equivalent to D. If $e \leq g$, then we have $p_i = q_i$.*

Let us first prove this in the special case where C is a quartic curve.

Theorem 26. *Let C be a smooth quartic plane curve. Let p_1, p_2, p_3 be three points in general position and let $D = p_1 + p_2 + p_3$ be a divisor. Suppose that $D' = \sum_{i=1}^{3} q_i$ is rationally equivalent to D. Then we have $p_i = q_i$.*

Proof. Again we need to check $l(D) \geq 2$ or $l(K - D) \geq 1$ by Riemann-Roch. Note that $K \in [H]$ where $[H]$ denotes the hyper-plane class (reason: the curve is embedded in the plane by the canonical class, which has degree 4 and dimension 3). We make a choice of K that depends on D: $K = p_1 + p_2 + r_1 + r_2$. Then $K - D = r_1 + r_2 - p_3$. If $l(K - D) > 0$ then there exists a non-constant (rational) function f on C with two poles along r_1, r_2. This function can be used to represent C as a degree 2 cover of $\mathbb{P}_1$

(use $x \mapsto (1 : f(x))$). But this implies that the curve is hyper-elliptic, contradicting the fact that the canonical class embeds the curve in $\mathbb{P}^{g-1}$. ■

We omit the proof for now but instead prove a weaker version that still suffices for our purposes.

Theorem 27. *There exists a smooth plane curve C of degree d and an effective divisor D of degree $\binom{d+2}{2} - 2$ on C such that for all divisors $D' \leq D$ with $\deg D' \leq g$ and $E \geq 0$, we have $E \sim D$ if and only if $E = D$.*

Proof. We modify the proof of theorem 1 as follows. We need to show that we can find an effective divisor D on some smooth curve C so that all sub-divisors $D' \leq D$ of degree g impose g conditions on $H^0(O_{\mathbb{P}^2}(d-3))$. The latter is a g-dimensional vector space. We first pick the support of D one point at a time. Once we pick the n-th point, we only need to look for the $(n+1)$-th point to be outside the span of finitely many $g-1$ dimensional subspaces. Now we interpolate a smooth curve of degree d through all the chosen points. Note that there is a one dimensional family of plane curves passing through any collection of $\binom{d+2}{2} - 2$ points. We pick a smooth curve within this family (this is possible by generic smoothness). ■

Putting aside the issues of working over finite fields, we can now start to build long codes with good properties.

Example 9. We pick a family of curves and divisors with growing degree as in Theorem 2. We take the support of D to be the set of β-points and embed the curve into $\mathbb{P}^g$. We pick any subset of size g on the curve and add one point to it to form a basis. The resulting code has length $\sim d^2$ and dimension $\sim d^2/2$. It has relative minimum distance $\beta \to \frac{1}{2}$. Thus it is an asymptotically optimal code.

Two questions that arise are whether we need any field extensions to place the points, and whether embedding the curve in higher dimensions would help with further reducing the size of the base field.

Problem 2. Implement the code over $\mathbb{F}_q$, i.e., prove that there are enough points in the projective plane to find the divisor D. Generalize the construction to curves/varieties

embedded in higher dimensions. Investigate when it is possible to thread curves of a given degree and genus through a given set of points. Use this to construct good long codes over relatively small fields.

9.7 Generalized Kasami codes

In [31], Kasami constructed a family of cyclic type codes that achieve the GV bound. The construction can be presented as follows. Given a projective curve X, fix a divisor G. Recall that the Riemann-Roch space of G, denoted by $L(G)$, is the k-vector space

$$L(G) := \{f \in k(X)^\times : \mathrm{div}(f) + G \geq 0\} \cup \{0\}$$

where $k(X)$ denotes the fraction field of X. Construct on X a prime divisor D of degree r. Then define the Kasami code $\mathscr{K}_X(G, D)$ as the image of the natural embedding

$$L(G - D) \hookrightarrow L(G)$$

Kasami considered the case $X = \mathbb{P}^1$ and $G = (n-1)P_\infty$. It is easily seen that $\mathscr{K}_X(G, D)$ is an $[n, n-r]_q$ code in this case. He then showed that there are sequences of the form $\mathscr{K}_X(G, D)$ that achieve the GV bound as $\deg(G) \to \infty$. To see this, we first bound the number of low weight vectors in $L(G)$. Given a minimum distance w, there are

$$\sum_{i=0}^{w-1} \binom{n}{i} (q-1)^i \leq q^{nh_q(\frac{w-1}{n})}$$

homogeneous polynomials of weight less than w in $L(G)$. Note that each homogeneous polynomial of degree n defines a finite set with no more than n/r closed points of degree r. The polynomial is contained in $\mathscr{K}_X(G, D)$ if and only if the finite set it defines contains D as one of its points. On the other hand, for large enough r, there are more than $\frac{q^r}{2r}$ closed points of degree r. Thus a large code of weight w exists if

$$\frac{n}{r} q^{nh_q(\frac{w-1}{n})} \leq \frac{q^r}{2r} \tag{9.8}$$

holds. Passing to the limits and comparing the exponents, we see in particular that the rates $1 - h_q(w/n)$ are achievable asymptotically by Kasami codes.

One can obtain a version of $\mathscr{K}_X(G, D)$ for more general varieties. As we shall see below, the choice of the variety is not important to achieve the GV bound. There is a sequence of Kasami codes over any variety X that achieves the GV bound. Furthermore, the inequality in (1) is fundamental, in the sense that it ties together the parameters of the code with some fundamental invariant of X, namely, its Zeta function, in a manner that depends heavily on the choice of the divisors on X. To emphasize the latter point, we next work out the general form of (1) for smooth projective curves.

For a projective curve $X \subset \mathbb{P}^m$, we denote its homogeneous coordinate ring by

$$S(X) := k[x_0, \cdots, x_m]/I(X)$$

This ring can be endowed with a grading $S(X) := \oplus_d S(X)^{(d)}$. The function field of X takes the form $k(X) = \{\frac{h}{g}; h, g \in S^{(d)}(X), g \neq 0\}$.

We note that $L(G)$ is a k-vector space. It is spanned by ratios $\frac{f}{g}$ of certain homogenous polynomials of the same degree. We pick a basis $\{\frac{h_i}{g_i}\}$ for $L(G)$ and write $f = \sum_i c_i \frac{h_i}{g_i}$. We then define the weight $w(f)$ to be the number of non-zero coefficients c_i that appear in its representation w.r.t the chosen basis.

As an example, let C be an elliptic curve embedded in the plane and take $G = p+q$. Let l be the unique line passing through p and q and s be the other intersection point of l with C. Then $L(G)$ is two dimensional by Riemann-Roch. It is isomorphic to the space of lines passing through s. It is indeed generated as a k-vector space by $\frac{l}{l}$ and $\frac{l'}{l}$ where l' is any other line that contains s. While the weight of a line in $L(G)$ depends on the choice of l', we are interested in the number of low-weight lines, which is independent of the basis.

Given a rational function $f = h/g$, we let $Z_f = X \cap V(h)$ be the (scheme theoretic) intersection of $X \subset \mathbb{P}^m$ with the hyper-surface defined by h. With the preceding

notation, we may now replace the inequality in (1) with

$$\frac{1}{r}\sum_{f\in L(G):w(f)<t} \#Z_f(\mathbb{F}_{q^r}) < \frac{1}{r}\#X(\mathbb{F}_{q^r}) \tag{9.9}$$

To proceed in a manner similar to Kasami's, we first need an upper bound on $\#Z_f(\mathbb{F}_q)$ that is invariant as Z_f varies over hyper-surfaces of degree n in $\mathbb{P}^m$. When X is a curve of degree d, Bezout's theorem gives a natural upper bound of

$$\#Z_f(\mathbb{F}_{q^r}) \leq nd$$

In this case the inequality in hand reduces to

$$ndq^{l(G)h_q(t/l(G))} < \#X(\mathbb{F}_{q^r}) \tag{9.10}$$

If X is fixed, we need

$$ndq^{l(G)h_q(t/l(G))} \leq q^r \tag{9.11}$$

to hold asymptotically. This again gives the GV bound. Thus we do not gain anything by working over fixed curves.

For schemes in higher dimension, we would like to associate codes to their general irreducible subschemes. Let X be a smooth projective irreducible scheme over $\mathbb{F}_q$ and $\mathscr{I}_D$ be the ideal sheaf of a subscheme $D \subset X$.

We restrict for now to closed points D of X with degree r. Recall that there is an exact sequence

$$0 \to \mathscr{I}_D \to \mathcal{O}_X \to \mathcal{O}_D \to 0$$

Twisting by a divisor G, we get

$$0 \to \mathscr{I}_D(G) \to \mathcal{O}_X(G) \to \mathcal{O}_D(G) \to 0$$

Taking global sections gives an exact sequence

$$0 \to H^0(\mathscr{I}_D(G)) \to H^0(\mathcal{O}_X(G)) \to k^r \to H^1(\mathscr{I}_D(G)) \to 0$$

where we used the fact that $H^1(O_X(G)) = 0$. The first embedding

$$H^0(\mathscr{I}_D(G)) \hookrightarrow H^0(\mathcal{O}_X(G))$$

is the Kasami code $\mathscr{K}_X(G, D)$, which has length $l(G)$ and dimension $h^0(\mathscr{I}_D(G)) = l(G) - r + h^1(\mathscr{I}_D(G))$. Its parameters depend on the choice of a divisor and an ideal sheaf. Consider the case $X = \mathbb{P}^2$. Then X has $q^{2r} + q^r + 1$ points over $\mathbb{F}_{q^r}$. If r is a prime number, we see that X has $\frac{q^{2r}+q^r-q^2-q}{r}$ points of degree r over $\mathbb{F}_q$. In general, using inclusion-exclusion, we can see, for large r, that X has roughly $\frac{q^{2r}}{r}$ degree r points over $\mathbb{F}_q$.

Take $G = (n-1)H_\infty$ so that $l(G) = \binom{n+1}{2} \sim \frac{n^2}{2}$. The number of homogeneous polynomials of degree $n-1$ in $L(G)$ with weight $< t$ is roughly $q^{l(G)h_q(t/l(G))}$. We need to count how many closed points of degree r are contained on each curve defined by such polynomials. We can estimate this number by $\frac{nq^r}{r}$. Indeed over $\mathbb{F}_{q^r}$ a curve of degree n has most of its points in an affine open. To get an estimate, it thus suffices to count the number of zeroes of a polynomial in two variables of degree n. There are q^r ways to fix one variable and obtain a univariate a polynomial of degree n, which has at most n zeroes. These points contract to around $\frac{nq^r}{r}$ points of degree r over $\mathbb{F}_q$. We thus need for large r and n that

$$\frac{q^r}{r} q^{\frac{n^2}{2}h_q(2t/n^2)} \leq \frac{q^{2r}}{r} \tag{9.12}$$

As $r, n, \to \infty$, the above inequality reduces to:

$$q^{r+n^2/2h(2t/n^2)} \leq q^{2r}$$

or

$$1 - 2r/n^2 \geq 1 - h(2t/n^2)$$

which is the GV bound.

When X is a general surface, Z_f is a curve of degree dn. Its genus is bounded above by $\frac{d^2n^2}{2}$ by the Castelnouvo bound. The Hasse-Weil-Serre estimate gives

$$\#Z_f(\mathbb{F}_q) \leq q^r + 1 + \frac{n^2d^2}{2}q^{r/2}$$

The general form of (1) for surfaces is thus

$$(q^r + 1 + \frac{n^2d^2}{2}q^{r/2})q^{l(G)h_q(t/l(G))} \leq \frac{\#X(\mathbb{F}_{q^r})}{r}$$

Again $\#X(\mathbb{F}_{q^r}) = O(q^{2r})$, hence, fixing the invariants of X will not give any improvements. So in general it appears that we need to look for varieties whose invariants vary with the parameters of the code. We end this chapter by the following question. Do such varieties exist and do they posses any coding theoretic merits?

Appendix A

Proof of channel comparison lemmas

A.1 Proof of Lemma 10

Proof. The first part of lemma is well-known. The BEC part is called the erasure decomposition lemma [65, Lemma 4.78] and the BSC part is a consequence of data processing (or, more precisely, hard decision decoding) [65, Problem 4.55]. The BSC half of the second part has been shown in [68, Appendix]. The rest of the statements appear to be new.

Let Y_δ denote output of a BSC_δ applied to input X. Then BMS W can be represented as $X \mapsto (Y_\Delta, \Delta)$ where $\Delta \in [0, 1/2]$ is a random variable independent of X. To prove the BEC part of the second claim, we need to show that for any input distribution $P_X = \mathrm{Ber}(p)$ we have

$$I(X; Y_\Delta, \Delta) \le I(X; Y_E)\,,$$

where Y_E is the output of a BEC_{1-C}. Note that $I(X; Y_E) = CH(X)$. Thus, we need to show that for any distribution of Δ and for any p the following inequality holds:

$$\mathbf{E}[h(p * \Delta)] - \mathbf{E}[h(\Delta)] \le (1 - \mathbf{E}[h(\Delta)])h(p)\,, \tag{A.1}$$

where h denotes the base-2 binary entropy function and $a * b = a(1-b) + (1-a)b$

is the binary convolution function. A result known as Mrs. Gerber's Lemma (MGL), cf. [80], which states that the parametric function

$$h(p * \delta) \quad \text{vs.} \quad h(\delta), \qquad \delta \in [0, 1/2] \tag{A.2}$$

is convex. Consequently, the function must be below the chord connecting its end-points, i.e.

$$h(p * \delta) \leq (1 - h(\delta))h(p) + h(\delta)\,.$$

Clearly, the latter inequality implies (A.1) after taking expectation over δ. Note also that the BSC part of the second claim also follows from (A.2). Indeed, from convexity we have

$$\mathbf{E}[h(p * \Delta)] \geq h(p * \delta_{eff})\,, \tag{A.3}$$

where δ_{eff} is chosen so that $h(\delta_{eff}) = \mathbf{E}[h(\Delta)]$. In turn, (A.3) is equivalent to the first relation in (4.10).

To prove the third part of the Lemma, i.e. (4.11), take $P_X = \text{Ber}(p)$ and $Q_X = \text{Ber}(q)$ and let P_Y, Q_Y be the output distributions induced by W. Similarly, let P_{Y_B}, Q_{Y_B} and P_{Y_E}, Q_{Y_E} be the distributions induced by the equal-χ^2-capacity BSC and BEC, respectively. We need to show (using (4.3)) that

$$D(P_{Y_B}\|Q_{Y_B}) \leq D(P_Y\|Q_Y) \leq D(P_{Y_E}\|Q_{Y_E})\,.$$

First notice that

$$I_{\chi^2}(\text{BSC}_\delta) = (1 - 2\delta)^2 \tag{A.4}$$

$$I_{\chi^2}(\text{BEC}_\delta) = 1 - \delta \tag{A.5}$$

After representing BMS as a mixture of BSC's we have $\eta = \mathbf{E}[(1-2\Delta)^2]$. Introducing the binary divergence function $d(a\|b) = D(\text{Ber}(a)\|\text{Ber}(b))$ we need to show: For any

distribution of $\Delta \in [0, 1/2]$ and $p, q \in [0, 1]$ we have

$$d(p * \delta_{eff} \| q * \delta_{eff}) \le \mathbf{E}[D(p * \Delta \| q * \Delta)] \le \mathbf{E}[(1-2\Delta)^2] d(p\|q)\,, \tag{A.6}$$

where $\delta_{eff} = \frac{1-\sqrt{\eta}}{2}$ is defined to satisfy

$$(1-2\delta_{eff})^2 = \mathbf{E}[(1-2\Delta)^2]\,.$$

Note that the right-most inequality in (A.6) follows froma a well-known fact that the strong data-processing contraction coefficient $\eta_{KL}(W)$ equals $\mathbf{E}[(1-2\Delta)^2]$ (e.g. this follows from the proof of [62, Theorem 21]).

To prove (A.6) we will establish *a variant of the MGL*, possibly of separate interest. Namely, we will show that for any fixed $p, q \in [0, 1]$ the function

$$d(p * \delta \| q * \delta) \quad \text{vs.} \quad (1-2\delta)^2 \qquad \delta \in [0, 1/2] \tag{A.7}$$

is convex. Clearly, (A.7) would imply both sides of (A.6).

To show (A.7) we proceed directly. Change parametrization to $x = (1-2\delta)^2$ and thus $\delta = \delta(x) = \frac{1-\sqrt{x}}{2}$. Letting $d(x; p, q) = d(p * \delta(x) \| q * \delta(x))$ we find

$$\partial_x d(x; p, q) = -\frac{1}{4\sqrt{x}} a(x; p, q) \tag{A.8}$$

$$a(x; p, q) = \left(\ln \frac{p * \delta}{1 - p * \delta} + \ln \frac{1 - q * \delta}{q * \delta}\right)(1-2p) + \left(\frac{1 - p * \delta}{1 - q * \delta} - \frac{p * \delta}{q * \delta}\right)(1-2q)\,. \tag{A.9}$$

For convenience, let us introduce

$$s \triangleq p * \delta, \quad \sigma \triangleq q * \delta\,.$$

Differentiating again, we get that convexity constraint $\partial_x^2 d(x; p, q) \ge 0$ is equivalent

to the following inequality:

$$2a(x;p,q)+\sqrt{x}b(x;p,q)\geq 0\,, \tag{A.10}$$

where (we used $1-2p=\frac{1-2s}{1-2\delta}$ and $1-2q=\frac{1-2\sigma}{1-2\delta}$)

$$b(x;p,q)=\frac{1}{(1-2\delta)^2}\left(\frac{(1-2s)^2}{s(1-s)}+(1-2\sigma)^2\frac{s(1-\sigma)^2+(1-s)\sigma^2}{\sigma^2(1-\sigma)^2}+\frac{4s}{\sigma(1-\sigma)}(1-2s)(1-2\sigma)\right).$$

Noticing that $\sqrt{x}=1-2\delta$ and multiplying (A.10) by $(1-2\delta)$ we get that we need to verify

$$2(1-2s)\ln\frac{s(1-\sigma)}{(1-s)\sigma}+\frac{2(1-2\sigma)}{\sigma(1-\sigma)}(\sigma+s-1)+\frac{(1-2s)^2}{s(1-s)}+(1-2\sigma)^2\frac{s(1-2\sigma)+\sigma^2}{\sigma^2(1-\sigma)^2}\geq 0\,. \tag{A.11}$$

Note that this inequality needs to hold for all values of $s,\sigma\in[\delta,1/2]$. However, due to arbitrariness of δ and since it does not appear in (A.11) (this is crucial), we need to simply establish (A.11) on the unit square $[0,1/2]^2$.

Here again, we reparameterize

$$u\triangleq 1-2s,\quad v\triangleq 1-2\sigma$$

so that $(u,v)\in[0,1]^2$ now range over the unit square. Then (A.11) is rewritten as (after dividing by u)

$$f(u,v)\triangleq 2\ln\frac{(1-u)(1+v)}{(1+u)(1-v)}-4\frac{v}{u(1-v^2)}(u+v)+\frac{4u}{1-u^2}+\frac{4v^2}{u(1-v^2)^2}(2(1-u)v+(1-v)^2)\geq 0\,. \tag{A.12}$$

It is easy to check that this inequality holds when either $u=0+,1-$ or $v=0+,1-$. Thus, we only need to rule out violations inside the $[0,1]^2$. Taking derivative over u of $f(u,v)$ we get

$$\partial_u f=0\quad\iff\quad\frac{2u^4}{(1-u^2)^2}=\frac{2v^4}{(1-v^2)^2}\,,$$

since $t\mapsto\frac{t}{1-t}$ is monotone, this implies that minimum of $f(u,v)$ is attained at $u=v$.

But $f(u,u) = 0$. Thus, we find

$$\min_{u,v} f(u,v) = \min_{u} f(u,u) = 0\,.$$

This concludes the proof of (A.12) and, hence, of (A.7). ■

A.2 Proof of Lemma 11

Proof. If $\tilde{W}_i$'s are degraded w.r.t W_i, then we have a Markov chain $X_0 - (Y, Y_1^m) - (Y, \tilde{Y}_1^m)$. This proves the first part.

To prove the second part, we may assume by induction that $W_i = \tilde{W}_i$ for all $i \geq 2$ (i.e. only one channel is replaced). Now suppose we have

$$U \perp\!\!\!\perp (X_1^m, Y_1^m, \tilde{Y}_1^m, Y) | X_0 \tag{A.13}$$

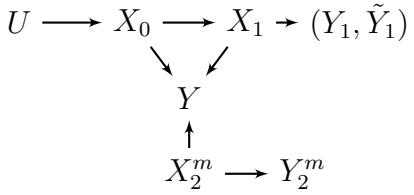
We want to show

$$I(U; Y, \tilde{Y}_1, Y_2^m) \leq I(U; Y, Y_1, Y_2^m)$$

or equivalently

$$I(U; \tilde{Y}_1 | Y, Y_2^m) \leq I(U; Y_1 | Y, Y_2^m). \tag{A.14}$$

The desired inequality follows from the definition of the less noisy order (in the conditional universe where (Y, Y_2^m) is observed) if we can show $U - X_0 - X_1 - (Y_1, \tilde{Y}_1)$ form a Markov chain conditionally on (Y, Y_2^m). Note that this is equivalent (by d-separation) to showing that the conditional independence assertions of the Lemma are representable by the following directed acyclic graphical model (DAG)



We first recall (from d-separation) that

$$A \perp\!\!\!\perp (B, C)|D \implies A \perp\!\!\!\perp B|(C, D) \tag{A.15}$$

for arbitrary random variables (A, B, C, D) (cf. (S3) in [43, Chapter 3]). From (A.13) and $(A.15)$, we see that the main assertion of interest in the above DAG is

$$U \perp\!\!\!\perp (Y_1, \tilde{Y}_1)|(X_1, Y, Y_2^m) \tag{A.16}$$

To prove this assertion, we note that

$$\begin{aligned}
p_{UY_1\tilde{Y}_1|X_1YY_2^m} &= \sum_{x_0,x_2^m} p_{UY_1\tilde{Y}_1|X_0X_2^mX_1YY_2} p_{X_0X_2^m|X_1YY_2^m} \\
&\overset{\text{by } (A.13)}{=} \sum_{x_0,x_2^m} p_{U|X_0} p_{Y_1\tilde{Y}_1|X_0X_2^mX_1YY_2} p_{X_0X_2^m|X_1YY_2^m} \\
&\propto \sum_{x_0,x_2^m} p_{U|X_0} p_{YY_1\tilde{Y}_1|X_0X_2^mX_1Y_2} p_{X_0X_2^m|X_1YY_2^m} \\
&= \sum_{x_0,x_2^m} p_{U|X_0} p_{Y_1\tilde{Y}_1|X_1} p_{Y|X_0X_1X_2} p_{X_0X_2^m|X_1YY_2^m} \\
&= p_{Y_1\tilde{Y}_1|X_1} \sum_{x_0,x_2^m} p_{U|X_0} p_{Y|X_0X_1X_2} p_{X_0X_2^m|X_1YY_2^m} \\
&= h(y_1, \tilde{y}_1, x_1) g(u, x_1, y, y_2^m).
\end{aligned}$$

This proves (A.16) and the desired inequality in (A.14) follows.

■

Appendix B

Erasure polynomials for LDMC(3)

Here we include the d-th erasure polynomials $E_d^{\mathrm{BEC}}(q)$ for $d \leq 10$ in Python form for LDMC(3). These polynomials are generated using the procedure described in §4.5 and are used to produce the bounds in Figs. 4-3-4-8 and Table 4.1, as well as for for code optimization in §5.2.

$\mathrm{E}_d^{\mathrm{BEC}}$
$d = 0$
0.5
$d = 1$
0.25
$d = 2$
0.125*q**4 - 0.25*q**3 + 0.25*q**2 - 0.25*q + 0.25
$d = 3$
0.1875*q**6 - 0.46875*q**5 + 0.46875*q**4 - 0.1875*q**3 + 4.440892e-16*q**2 - 0.09375*q + 0.15625
$d = 4$
0.46875*q**8 - 1.9375*q**7 + 3.71875*q**6 - 4.3125*q**5 + 3.28125*q**4 - 1.6875*q**3 + 0.65625*q**2 - 0.3125*q + 0.15625

$d = 5$

0.9375*q**10 - 4.58007812500001*q**9 + 10.087890625*q**8 - 13.0859375*q**7 + 10.976562500*q**6 - 6.15234375*q**5 + 2.24609375*q**4 - 0.4296875*q**3 + 0.0390625*q**2 - 0.126953125*q + 0.103515625

$d = 6$

2.2900390625*q**12 - 14.455078125*q**11 + 42.9462890624997*q**10 - 79.5214843749996*q**9 + 102.12890625*q**8 - 95.5664062499994*q**7 + 66.5722656249995*q**6 - 34.7460937499998*q**5 + 13.5791015624999*q**4 - 3.99414062499997*q**3 + 0.981445312499996*q**2 - 0.310546875*q + 0.103515625

$d = 7$

5.05517578125*q**14 - 36.368896484375*q**13 + 121.872802734375*q**12 - 251.26171875*q**11 + 354.7236328125*q**10 - 361.612548828124*q**9 + 274.061279296874*q**8 - 156.953124999999*q**7 + 68.3422851562493*q**6 - 22.3791503906245*q**5 + 5.18676757812476*q**4 - 0.68359374999993*q**3 + 0.0820312499999894*q**2 - 0.131591796874999*q + 0.070556640625

$d = 8$

12.2824707031249*q**16 - 104.389648437499*q**15 + 421.901855468746*q**14 - 1078.21191406248*q**13 + 1953.12304687496*q**12 - 2661.41503906243*q**11 + 2821.08544921866*q**10 - 2368.68652343741*q**9 + 1587.56103515619*q**8 - 849.672851562463*q**7 + 361.467285156233*q**6 - 121.303710937495*q**5 + 31.8554687499987*q**4 - 6.56933593749975*q**3 + 1.18603515624997*q**2 - 0.282226562499999*q + 0.070556640625

$d = 9$

28.517944335937*q**18 - 270.209632873528*q**17 + 1213.44797515864*q**16
- 3426.34039306621*q**15 + 6803.52593994091*q**14 - 10066.1437683096*q**13
+ 11474.6510925279*q**12 - 10284.0617065415*q**11 + 7337.84271240106*q**10
- 4200.8187103263*q**9 + 1938.88133239697*q**8 - 722.934997558378*q**7
+ 216.904724121012*q**6 - 51.2509460448973*q**5 + 8.86129760741628*q**4
- 0.913879394530327*q**3 + 0.115905761718657*q**2 - 0.122840881347652*q
+ 0.0489273071289062

$d = 10$

69.4315452575683*q**20 - 742.947502136231*q**19 + 3808.84984970093*q**18
- 12453.0257034302*q**17 + 29158.0880355837*q**16 - 52039.3605651862*q**15
+ 73535.9429168715*q**14 - 84300.01968384*q**13 + 79613.6392593413*q**12
- 62487.5754547145*q**11 + 40908.980049135*q**10 - 22326.4821624763*q**9
+ 10117.3594665529*q**8 - 3780.88119506833*q**7 + 1154.60105895993*q**6
- 285.082305908195*q**5 + 56.3512229919426*q**4 - 8.95305633544941*q**3
+ 1.28042221069343*q**2 - 0.244636535644538*q + 0.0489273071289063

Bibliography

[1] Matti Aaltonen. A new upper bound on nonbinary block codes. *Discrete Mathematics*, 83(2):139–160, 1990.

[2] Rudolf Ahlswede. The rate-distortion region for multiple descriptions without excess rate. *IEEE Transactions on Information Theory*, 31(6):721–726, 1985.

[3] Rudolf Ahlswede and Levon H Khachatrian. The diametric theorem in hamming spaces-optimal anticodes. *Advances in Applied Mathematics*, 20(4):429–449, 1998.

[4] Amjad Ali, Kyung Sup Kwak, Nguyen H Tran, Zhu Han, Dusit Niyato, Farukh Zeshan, M Talha Gul, and Doug Young Suh. Raptorq-based efficient multimedia transmission over cooperative cellular cognitive radio networks. *IEEE Transactions on Vehicular Technology*, 67(8):7275–7289, 2018.

[5] Masoud Barakatain and Frank R Kschischang. Low-complexity concatenated ldpc-staircase codes. *Journal of Lightwave Technology*, 36(12):2443–2449, 2018.

[6] Amos Beimel, Shlomi Dolev, and Noam Singer. Rt oblivious erasure correcting. *IEEE/ACM Transactions on Networking (TON)*, 15(6):1321–1332, 2007.

[7] Itai Benjamini, David Ellis, Ehud Friedgut, Nathan Keller, and Arnab Sen. Juntas in the âĎŞ1-grid and lipschitz maps between discrete tori. *Random Structures & Algorithms*, 49(2):253–279, 2016.

[8] Christos Bouras, Nikolaos Kanakis, Vasileios Kokkinos, and Andreas Papazois. Evaluating raptorq fec over 3gpp multicast services. In *2012 8th International*

Wireless Communications and Mobile Computing Conference (IWCMC), pages 257–262. IEEE, 2012.

[9] Christos Bouras, Nikolaos Kanakis, Vasileios Kokkinos, and Andreas Papazois. Embracing raptorq fec in 3gpp multicast services. *Wireless Networks*, 19(5):1023–1035, 2013.

[10] Eirina Bourtsoulatze, David Burth Kurka, and Deniz Gündüz. Deep joint source-channel coding for wireless image transmission. *IEEE Transactions on Cognitive Communications and Networking*, 2019.

[11] Shraga I Bross and Simon Litsyn. Improved upper bounds for codes with unequal error protection. *IEEE Transactions on Information Theory*, 52(7):3329–3333, 2006.

[12] John W Byers, Michael Luby, Michael Mitzenmacher, and Ashutosh Rege. A digital fountain approach to reliable distribution of bulk data. *ACM SIGCOMM Computer Communication Review*, 28(4):56–67, 1998.

[13] Yuval Cassuto and Amin Shokrollahi. Online fountain codes with low overhead. *IEEE Transactions on Information Theory*, 61(6):3137–3149, 2015.

[14] Tsung-Yi Chen, Dariush Divsalar, Jiadong Wang, and Richard D Wesel. Protograph-based raptor-like ldpc codes for rate compatibility with short block-lengths. In *2011 IEEE Global Telecommunications Conference-GLOBECOM 2011*, pages 1–6. IEEE, 2011.

[15] Tsung-Yi Chen, Kasra Vakilinia, Dariush Divsalar, and Richard D Wesel. Protograph-based raptor-like ldpc codes. *IEEE Transactions on Communications*, 63(5):1522–1532, 2015.

[16] Stefano Ciliberti, Marc Mézard, and Riccardo Zecchina. Lossy data compression with random gates. *Physical review letters*, 95(3):038701, 2005.

[17] Gerard Cohen. A nonconstructive upper bound on covering radius. *Information Theory, IEEE Transactions on*, 29(3):352–353, 1983.

[18] Thomas Cover. Broadcast channels. *IEEE Transactions on Information Theory*, 18(1):2–14, 1972.

[19] Imre Csiszar and János Körner. *Information theory: coding theorems for discrete memoryless systems*. Cambridge University Press, 2011.

[20] Giuseppe Durisi, Tobias Koch, and Petar Popovski. Toward massive, ultrareliable, and low-latency wireless communication with short packets. *Proceedings of the IEEE*, 104(9):1711–1726, 2016.

[21] Abbas El Gamal and Young-Han Kim. *Network information theory*. Cambridge university press, 2011.

[22] Abbas A El Gamal and Thomas M Cover. Achievable rates for multiple descriptions. *IEEE Trans. Information Theory*, 28(6):851–857, 1982.

[23] G David Forney. Concatenated codes. 1965.

[24] Vivek K Goyal. Multiple description coding: Compression meets the network. *IEEE Signal processing magazine*, 18(5):74–93, 2001.

[25] Vivek K Goyal, Jelena Kovačević, and Jonathan A Kelner. Quantized frame expansions with erasures. *Applied and Computational Harmonic Analysis*, 10(3):203–233, 2001.

[26] Muhammad Talha Gul, Amjad Ali, Deepak Kumar Singh, Umera Imtinan, Imran Raza, Syed Asad Hussain, Doug Young Suh, and Jong-wook Lee. Merge-and-forward: A cooperative multimedia transmissions protocol using raptorq codes. *IET Communications*, 10(15):1884–1895, 2016.

[27] Andrew Hagedorn, Sachin Agarwal, David Starobinski, and Ari Trachtenberg. Rateless coding with feedback. In *IEEE INFOCOM 2009*, pages 1791–1799. IEEE, 2009.

[28] Richard W Hamming. Error detecting and error correcting codes. *The Bell system technical journal*, 29(2):147–160, 1950.

[29] Christopher Hooley. On artinâĂŹs conjecture. *J. reine angew. Math*, 225(209-220):248, 1967.

[30] Abhinav Kamra, Vishal Misra, Jon Feldman, and Dan Rubenstein. Growth codes: Maximizing sensor network data persistence. In *ACM SIGCOMM Computer Communication Review*, volume 36, pages 255–266. ACM, 2006.

[31] Tadao Kasami. An upper bound on k/n for affine-invariant codes with fixed d/n (corresp.). *IEEE Transactions on Information Theory*, 15(1):174–176, 1969.

[32] Kia Khezeli and Jun Chen. A source-channel separation theorem with application to the source broadcast problem. *IEEE Transactions on Information Theory*, 62(4):1764–1781, 2016.

[33] Yuval Kochman, Arya Mazumdar, and Yury Polyanskiy. The adversarial joint source-channel problem. In *2012 IEEE International Symposium on Information Theory Proceedings*, pages 2112–2116. IEEE, 2012.

[34] Yuval Kochman, Arya Mazumdar, and Yury Polyanskiy. Results on combinatorial joint source-channel coding. In *2012 IEEE Information Theory Workshop*, pages 10–14. IEEE, 2012.

[35] Yuval Kochman, Or Ordentlich, and Yury Polyanskiy. Ozarow-type outer bounds for memoryless sources and channels. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 1774–1778. IEEE, 2018.

[36] Yuval Kochman, Or Ordentlich, and Yury Polyanskiy. A lower bound on the expected distortion of joint source-channel coding. *arXiv preprint arXiv:1902.07979*, 2019.

[37] Silvija Kokalj-Filipović, Emina Soljanin, and Yang Gao. Cliff effect suppression through multiple-descriptions with split personality. In *2011 IEEE International Symposium on Information Theory Proceedings*, pages 948–952. IEEE, 2011.

[38] J Korner. Comparison of two noisy channels. *Topics in information theory*, pages 411–423, 1977.

[39] Victoria Kostina and Babak Hassibi. Rate-cost tradeoffs in control. *IEEE Transactions on Automatic Control*, 2019.

[40] Florent Krzakala and Lenka Zdeborová. Hiding quiet solutions in random constraint satisfaction problems. *Physical review letters*, 102(23):238701, 2009.

[41] Frank R Kschischang, Brendan J Frey, Hans-Andrea Loeliger, et al. Factor graphs and the sum-product algorithm. *IEEE Transactions on information theory*, 47(2):498–519, 2001.

[42] Shrinivas Kudekar, Santhosh Kumar, Marco Mondelli, Henry D Pfister, Eren ŞaşoÇğlu, and Rüdiger L Urbanke. Reed–muller codes achieve capacity on erasure channels. *IEEE Transactions on information theory*, 63(7):4298–4316, 2017.

[43] Steffen L Lauritzen. *Graphical models*, volume 17. Clarendon Press, 1996.

[44] Michael Luby, Michael Mitzenmacher, Mohammad Amin Shokrollahi, Daniel A Spielman, and Volker Stemann. Practical loss-resilient codes. In *STOC*, volume 97, pages 150–159, 1997.

[45] Michael Luby, Amin Shokrollahi, Mark Watson, Thomas Stockhammer, and Lorenz Minder. Raptorq forward error correction scheme for object delivery. Technical report, 2011.

[46] David JC MacKay. Good error-correcting codes based on very sparse matrices. *IEEE transactions on Information Theory*, 45(2):399–431, 1999.

[47] David JC MacKay. Encyclopedia of sparse graph codes, 2005.

[48] Florence Jessie MacWilliams and Neil James Alexander Sloane. *The theory of error-correcting codes*. Elsevier, 1977.

[49] Anuran Makur and Yury Polyanskiy. Comparison of channels: criteria for domination by a symmetric channel. *IEEE Transactions on Information Theory*, 64(8):5704–5725, 2018.

[50] Nuno C Martins and Munther A Dahleh. Feedback control in the presence of noisy channels:âĂIJbode-likeâĂİ fundamental limitations of performance. *IEEE Transactions on Automatic Control*, 53(7):1604–1615, 2008.

[51] Burt Masnick and Jack Wolf. On linear unequal error protection codes. *IEEE Transactions on Information Theory*, 13(4):600–607, 1967.

[52] Arya Mazumdar, Yury Polyanskiy, Ankit Singh Rawat, and Hajir Roozbehani. Distance preserving maps and combinatorial joint source-channel coding for large alphabets. In *Information Theory (ISIT), 2016 IEEE International Symposium on*, pages 3067–3071. IEEE, 2016.

[53] Cyril Méasson, Andrea Montanari, Tom Richardson, and Rudiger Urbanke. Life above threshold: From list decoding to area theorem and mse. *arXiv preprint cs/0410028*, 2004.

[54] Alexander E Mohr, Eve A Riskin, and Richard E Ladner. Unequal loss protection: Graceful degradation of image quality over packet erasure channels through forward error correction. *IEEE journal on selected areas in communications*, 18(6):819–828, 2000.

[55] Alexander E Mohr, Eve A Riskin, and Richard E Ladner. Unequal loss protection: Graceful degradation of image quality over packet erasure channels through forward error correction. *IEEE journal on selected areas in communications*, 18(6):819–828, 2000.

[56] Andrea Montanari and Elchanan Mossel. Smooth compression, gallager bound and nonlinear sparse-graph codes. In *2008 IEEE International Symposium on Information Theory*, pages 2474–2478. IEEE, 2008.

[57] M Ram Murty and Seshadri Srinivasan. Some remarks on artinâĂŹs conjecture. *Canad. Math. Bull*, 30(1):80–85, 1987.

[58] L Ozarow. On a source-coding problem with two channels and three receivers. *Bell System Technical Journal*, 59(10):1909–1921, 1980.

[59] Yury Polyanskiy. On metric properties of maps between hamming spaces and related graph homomorphisms. *arXiv preprint arXiv:1503.02779*, 2015.

[60] Yury Polyanskiy. On metric properties of maps between hamming spaces and related graph homomorphisms. *Journal of Combinatorial Theory, Series A*, 145:227–251, 2017.

[61] Yury Polyanskiy and Alex Samorodnitsky. Improved log-sobolev inequalities, hypercontractivity and uncertainty principle on the hypercube. *arXiv preprint arXiv:1606.07491*, 2016.

[62] Yury Polyanskiy and Yihong Wu. Strong data-processing inequalities for channels and bayesian networks. In *Convexity and Concentration*, pages 211–249. Springer, 2017.

[63] Nazanin Rahnavard and Faramarz Fekri. Unequal error protection using low-density parity-check codes. In *Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on*, page 449. IEEE, 2004.

[64] Zvi Reznic, Meir Feder, and Ram Zamir. Distortion bounds for broadcasting with bandwidth expansion. *IEEE Transactions on Information Theory*, 52(8):3778–3788, 2006.

[65] Tom Richardson and Ruediger Urbanke. *Modern coding theory*. Cambridge university press, 2008.

[66] Anant Sahai and Sanjoy Mitter. The necessity and sufficiency of anytime capacity for stabilization of a linear system over a noisy communication linkâĂŤpart i:

Scalar systems. *IEEE transactions on Information Theory*, 52(8):3369–3395, 2006.

[67] Sujay Sanghavi. Intermediate performance of rateless codes. In *2007 IEEE Information Theory Workshop*, pages 478–482. IEEE, 2007.

[68] Eren Sasoglu. *Polar coding theorems for discrete systems*. PhD thesis, EPFL, 2011.

[69] Claude E Shannon. Coding theorems for a discrete source with a fidelity criterion. *IRE Nat. Conv. Rec*, 4(142-163):1, 1959.

[70] Claude Elwood Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.

[71] Daniel A Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Transactions on Information Theory*, 42(6):1723–1731, 1996.

[72] Kenya Sugihara, Yoshikuni Miyata, Takashi Sugihara, Kazuo Kubo, Hideo Yoshida, Wataru Matsumoto, and Takashi Mizuochi. A spatially-coupled type ldpc code with an ncg of 12 db for optical transmission beyond 100 gb/s. In *Optical Fiber Communication Conference*, pages OM2B–4. Optical Society of America, 2013.

[73] David Sutter and Joseph M Renes. Universal polar codes for more capable and less noisy channels and sources. In *2014 IEEE International Symposium on Information Theory*, pages 1461–1465. IEEE, 2014.

[74] Louis Tan, Ashish Khisti, and Emina Soljanin. Distortion bounds for broadcasting a binary source over binary erasure channels. In *2013 13th Canadian Workshop on Information Theory*, pages 49–54. IEEE, 2013.

[75] Robert Michael Tanner. A transform theory for a class of group-invariant codes. *IEEE transactions on information theory*, 34(4):725–775, 1988.

[76] Sekhar Tatikonda, Anant Sahai, and Sanjoy Mitter. Stochastic linear control over a communication channel. *IEEE transactions on Automatic Control*, 49(9):1549–1561, 2004.

[77] Andrea L Vitali. Multiple description coding-a new technology for video streaming over the internet. 2007.

[78] Martin J Wainwright, Michael I Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

[79] Martin J Wainwright and Elitza Maneva. Lossy source encoding via message-passing and decimation over generalized codewords of ldgm codes. In *Proceedings. International Symposium on Information Theory, 2005. ISIT 2005.*, pages 1493–1497. IEEE, 2005.

[80] Aaron D Wyner and Jacob Ziv. A theorem on the entropy of certain binary sequences and applications–I. 19(6):769–772, November 1973.

[81] Lei M Zhang and Frank R Kschischang. Low-complexity soft-decision concatenated ldgm-staircase fec for high-bit-rate fiber-optic communication. *Journal of Lightwave Technology*, 35(18):3991–3999, 2017.

[82] Jacob Ziv. The behavior of analog communication systems. *IEEE Transactions on Information Theory*, 16(5):587–594, 1970.