NORWEGIAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Faculty of Information Technology, Mathematics and Electrical Engineering

Department of Telematics

UNIVERSITY OF TARTU

Faculty of Mathematics and Computer Science

Institute of Computer Science

Informatics Curriculum

NordSecMob

Sushanta Paudyal

MULTI-SYMBOL LOCALLY REPAIRABLE CODES

Master's Thesis (30 ECTS)

Supervisor: Vitaly Skachek, PhD

Co-supervisor: Colin Boyd, PhD

2015

Why Mr. Anderson?

Why, why? Why do you do it? Why, why get up? Why keep fighting?

Do you believe you're fighting for something, for more than your survival?

Can you tell me what it is? Do you even know?

Is it freedom, or truth, perhaps peace, could it be for love? Illusions, Mr. Anderson, vagaries of perception, temporary constructs of a feeble human intellect trying desperately to justify an existence that is without meaning or purpose. And all of them as artificial as the Matrix itself, although only a human mind could invent something as insipid as love.

You must be able to see it Mr. Anderson, you must know it by now. You can't win, it's pointless to keep fighting.

Why Mr. Anderson, why, why do you persist?

Because I choose to.

The Wachowskis

MULTI-SYMBOL LOCALLY REPAIRABLE CODES

Abstract

Locally Repairable Codes (LRCs) have seen an increase in interest because of their applicability in distributed storage systems. In this thesis, we define and study a generalization of LRCs that we name Multi-symbol Locally Repairable Codes (MLRCs). MLRCs can be useful in situations where multiple users request data from a number of failed servers, concurrently. We derive an upper bound on the minimum distance of such MLRCs.

Keywords: distributed storage systems, locally repairable codes, codes with locality and availability, erasure correcting codes.

Mitme sümboliga lokaalselt parandatavad kodeeringud

Kokkuvõte

Lokaalselt parandatavad kodeeringud (LRC-kodeeringud) on järjest suurema huvi all seoses nende rakendatavusega hajutatud salvestussüsteemides. Käesolevas magistritöös defineerime ja uurime mitme sümboliga lokaalselt parandatavaid kodeeringuid (MLRC-kodeeringuid), mis on üldistus üle LRC-kodeeringute. MLRC-kodeeringud võivad olla kasulikud olukordades kus rohkelt kasutajaid pärivad samaaegselt andmeid mitmelt vigaselt serverilt. Me tuletame ülemise tõkke selliste MLRC-kodeeringute minimaalsele kaugusele.

Märksõnad: hajutatud salvestussüsteemid, lokaalselt parandatavad kodeeringud, kodeeringud lokaalsusega ja saadavusega, kustutusi parandavad kodeeringud.

Translation: Ivo Kubjas

Contents

List of Symbols

$\mathbb{F}_q$	A finite field with q elements.
$\mathbb{F}_2$	A binary finite field.
$\mathbb{N}$	A set of natural numbers: $\{1, 2, \cdots\}$.
$[k]$	The set: $\{1, 2, \cdots, k\}$.
$[n, k, d]$	An expression used to denote the length, dimension and minimum distance (respectively) of a linear code.
$d_{\min}(\mathcal{C})$	The minimum distance of a code $\mathcal{C}$.
$\lvert\mathcal{C}\rvert$	The size of a code $\mathcal{C}$, i.e. the number of codewords in $\mathcal{C}$.
$\vec{a}$	A vector a that is written as a tuple of its n coordinates: $\vec{a} = (a_1, a_2, \cdots, a_n)$.
$\lvert\vec{a}\rvert$	The number of coordinates in $\vec{a}$.
$\vec{x}\rvert_{R_j}$	Vector $\vec{x}$ restricted to a set of coordinates R_j. If $\vec{x} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ and $R_j = \{1, 4, 6\}$, then $\vec{x}\rvert_{R_j} = (x_1, x_4, x_6)$.
$\delta(\vec{a}, \vec{b})$	The Hamming distance between vectors $\vec{a}$ and $\vec{b}$.

Chapter 1

Introduction

1.1 Background

This thesis deals with the problem of data storage in distributed systems; in such systems, data is stored in a number of servers that operate independently. Let us take an example of a distributed storage system consisting of n servers: $S_1, S_2, \cdots, S_n$. Let us assume that one user wants data from server S_2 and multiple users want data from S_n. Further, consider a scenario where both the servers are down at the moment. In such a situation, user requests can be satisfied if data has been stored redundantly in other servers.

Situations like the one described in the preceding paragraph are formalized through Locally Repairable Codes (LRCs); LRCs are codes that allow local repair of lost symbols. Local repair is a process of recovering an erased symbol through a small set of other symbols. If each server in the distributed system is setup to hold an encoded symbol, then the problem of recovering data from failed servers can be mapped to the problem of local repair.

1.2 Literature Review

We use standard notations from classical coding theory: n, k and d refer to the length, dimension and minimum distance of a code defined over a finite field, respectively.

The notion of locality, where a lost symbol can be recovered through a small set of other symbols, was introduced by Yekhanin et al. in [2]; the paper performed an in-depth study of relationships between redundancy, locality and minimum distance of a special case of LRCs. The following is one of the main results from [2]:

$$d \;\leq\; n - k - \left\lceil \frac{k}{r} \right\rceil + 2,$$

where r denotes the locality of *information symbols*. This means that any information symbol that is erased can be repaired from at most r other symbols.

Recognizing that local repair would be unsuccessful if one of the r symbols were also erased, Wang and Zhang introduce in [7] the notion of availability, where a symbol can be repaired through t disjoint sets consisting of up to r symbols each. They derive an upper bound on the minimum distance of *linear* codes that have information symbol locality and availability. We re-write their result below:

$$d \;\leq\; n - k - \left\lceil \frac{t(k-1)+1}{t(r-1)+1} \right\rceil + 2. \tag{1.1}$$

The same bound as in (1.1) is derived by Dimakis et al. in [4]; they show that the bound holds for *non-linear* codes as well.

Tamo and Barg take things a step further by deriving an upper bound on the minimum distance of codes that have *all symbol* locality and availability in [6]. They obtain the following bound:

$$d \;\leq\; n - \sum_{i=0}^{t} \left\lfloor \frac{k-1}{r^i} \right\rfloor.$$

The notion of locality and availability is generalized further by Mazumdar et al. in [3]. They introduce the idea of cooperative local repair where a set of

symbols is used to recover another disjoint set of erased symbols. Mazumdar et al. obtain the following bound for the minimum distance:

$$d \leq n - k + 1 - l\left(\left\lceil \frac{k}{z} \right\rceil - 1\right),$$

where l denotes the number of erasures and z denotes the maximum number of symbols required for repair.

1.3 Our Contribution

In this thesis, we build upon the work of [3] (and of [4]) by introducing a new generalization of LRCs named Multi-symbol Locally Repairable Codes (MLRCs). Our generalization is different from the construction in [3] because we allow for the repair of a collection of symbols where repetitions are allowed. Further, each symbol is repaired independently from a set of symbols of size at most r. We derive an upper bound on the minimum distance of MLRCs and compare our bound with existing bounds in the literature.

1.4 Organization

Chapter 2 introduces different models used to address the problem of local repair. Section 2.1.1 gives a brief overview of the necessary notions of coding theory for readers unfamiliar with classical coding theory. We introduce the concepts relevant to local repair in Section 2.1.2. We introduce the generic and well established model of local repair in Section 2.2. Section 2.3 introduces a generalization of the model described in Section 2.2. We introduce our new model, which is yet another generalization of the model of Section 2.2, in Section 2.4.

Chapter 3 presents our main work. We derive an upper bound on the minimum distance of our new model in Section 3.1, as the title suggests. We compare the bound derived in Section 3.1 with other existing bounds in Section 3.2 and briefly comment on further research directions in Section 3.3.

We suggest the reader unfamiliar with coding theory to read in the following sequence: Section 2.1.1, Section 2.1.2, Chapter 1, and then sequentially from Section 2.2. For the advanced reader who is up to speed with local repair, we suggest to skip to Sections 2.3-2.4. The remaining readers are encouraged to read sequentially from Section 2.1.2.

Chapter 2

Different Repair Models

In this chapter, we introduce three different models that are used to address different generalizations of the problem of local repair; these models are abbreviated as LRCs, CLRCs and MLRCs. We begin the chapter with some preliminaries.

2.1 Preliminaries

2.1.1 Classical Coding Theory

Consider the following: Avia wants to send a message to Brock; let us represent the message as a k bit vector $\vec{x} = (x_1, x_2, \cdots, x_k)$, $\vec{x} \in \mathbb{F}_2^k$. Ideally, the vector $\vec{x}$ would be sent through a medium without errors and Brock would receive the intended message. However, in practice, mediums are noisy and it is very likely that some of the bits will be flipped/ lost during transmission.

Coding theory accounts for the issue described in the preceding paragraph through *error correcting codes*. Error correcting codes (or simply codes) introduce redundancy into the the data transmitted to account for errors. A **code** can be thought of as a collection of vectors over a finite field. In particular, a **linear code** is a collection of vectors over a finite field where each vector/codeword belongs to the row space of a matrix called the *generator matrix* of the code.

Example 1. *Let us suppose Avia sends the following message to Brock:* $\vec{x} =$

$(1, 0, 1)$. *We will show how we can introduce redundancy into the message to account for errors, in a process called encoding.*

Avia's message could be *encoded* to a codeword of a so-called Simplex code in the following way:

$$\underset{\vec{x}}{\begin{bmatrix}1 & 0 & 1\end{bmatrix}} \underset{G}{\begin{bmatrix}1 & 0 & 0 & 0 & 1 & 1 & 1\\ 0 & 1 & 0 & 1 & 0 & 1 & 1\\ 0 & 0 & 1 & 1 & 1 & 0 & 1\end{bmatrix}} = \underset{\vec{y}}{\begin{bmatrix}1 & 0 & 1 & 1 & 0 & 1 & 0\end{bmatrix}}.$$

Put in words, the vector $\vec{x}$ is encoded using the generator matrix of a Simplex code (the matrix G) to generate a codeword of a Simplex code (vector $\vec{y}$). The reader should immediately notice that the first three bits of $\vec{y}$ are the same as that of the message/information vector (viz. $\vec{x}$); this is no coincidence. Codes that encode information symbols directly into the codewords are referred to as **systematic codes**. ■

Linear codes are normally described by three parameters within brackets: $[n, k, d]$.

The **length** of each codeword of a code is represented by n; n equals 7 for the Simplex code described in Example 1.

The number of information symbols is represented by k; notice that k is also the number of rows of the generator matrix and is usually referred to as the **dimension** of the code. The generator matrix is a $k \times n$ matrix whose rows are linearly independent.

The *Hamming distance* of two vectors is the number of coordinates at which they differ; for instance, the Hamming distance of vectors $(1, 0, 1)$ and $(0, 1, 0)$ is 3 because they differ in all three coordinates. This can be formally written as:

$$\delta\left(\vec{a}, \vec{b}\right) = |\{i : a_i \neq b_i\}|,$$

where $\delta\left(\vec{a}, \vec{b}\right)$ denotes the Hamming distance of vectors $\vec{a}$ and $\vec{b}$; $|\vec{a}| = |\vec{b}|$.

The **minimum distance** of a code (denoted by d) is the minimum of the Hamming distances of all codewords of the code.

Example 2. *We will give an example of the minimum distance through a (non-linear) code $\mathcal{C}$ consisting of only three codewords; each codeword is a linear combination of the rows of the following matrix:*

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

The code $\mathcal{C}$ can be represented as a ***set of codewords***: $\mathcal{C} = \{\vec{y_1}, \vec{y_2}, \vec{y_3}\}$, where:

$$\begin{aligned} \vec{y_1} &= \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}, \\ \vec{y_2} &= \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}, \\ \vec{y_3} &= \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}. \end{aligned}$$

We invite the reader to think about the fact that each of the three preceding vectors are indeed linear combinations of rows of the matrix G. The Hamming distances of the codewords of $\mathcal{C}$ are:

$$\begin{aligned} \delta(\vec{y_1}, \vec{y_2}) &= 4, \\ \delta(\vec{y_1}, \vec{y_3}) &= 4, \\ \delta(\vec{y_2}, \vec{y_3}) &= 4. \end{aligned}$$

Since the minimum of the distances is 4, $d = 4$ for the code $\mathcal{C}$. ■

The relationship between the length, dimension and minimum distance of a linear code is captured by the following bound [5]:

$$d \leq n - k + 1.$$

This bound, named after Richard Collom Singleton, is referred to as the **Singleton bound** for linear codes.

2.1.2 Codes with Locality

We will introduce the idea of locality through the binary Simplex code example of Section 2.1.1. Consider the following encoding of $\vec{x} = (x_1, x_2, x_3)$:

$$\underbrace{\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}}_{\vec{x}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}}_{G} = \underbrace{\begin{bmatrix} x_1 & x_2 & x_3 & x_2+x_3 & x_1+x_3 & x_1+x_2 & x_1+x_2+x_3 \end{bmatrix}}_{\vec{y}}.$$

Let us assume that the first bit of $\vec{y}$ is erased during transmission. Of the remaining bits of $\vec{y}$, we may use a combination of two bits to reconstruct the erased symbol (viz. x_1). If we denote by $\Gamma_i(x_1)$, the i-th set of indices of symbols that repair x_1, then we have the following (this is one of many possibilities):

$$\Gamma_1(x_1) = \{2, 6\}, \qquad \Gamma_2(x_1) = \{3, 5\}, \qquad \Gamma_3(x_1) = \{4, 7\}.$$

This is because, over a binary finite field, the following equations are satisfied:

$$\begin{aligned} x_1 &= x_2 + (x_1 + x_2), \\ x_1 &= x_3 + (x_1 + x_3), \\ x_1 &= (x_2 + x_3) + (x_1 + x_2 + x_3). \end{aligned}$$

$\Gamma_1(x_1)$, $\Gamma_2(x_1)$ and $\Gamma_3(x_1)$ are referred to as the **repair groups** of the symbol x_1, and a union of these non-intersecting repair groups is called the **repair set**.

The maximum size that any repair group can have is defined as the **locality** of the symbol under consideration; in other words, the maximum number of symbols needed to repair a lost symbol is the locality of that lost symbol. Thus, the first symbol of $\vec{y}$ has locality equal to 2. We represent locality by the letter r.

The total number of non-intersecting repair groups for a symbol is defined as the **availability** of the symbol. In our example, the availability of x_1 (the first symbol) is equal to 3 since there are a total of three disjoint repair groups that

repair the first symbol. We represent availability by the letter t.

2.2 Locally Repairable Codes (LRCs)

Definition 1. *[4] An (n, k, r, t) LRC satisfies the following properties.*

1. *t* ***repair groups each:*** *For each encoded information (systematic) symbol y_i, there exist t sets $\Gamma_1(y_i), \Gamma_2(y_i), \cdots, \Gamma_t(y_i)$, such that y_i is a function of the encoded symbols indexed by $\Gamma_j(y_i)$; $i \in [k], j \in [t]$ and $\Gamma_j(y_i) \subseteq [n] \backslash \{i\}$.*
2. ***Locality of systematic symbols:*** *$|\Gamma_j(y_i)| \leq r$, for all $i \in [k]$, $j \in [t]$.*
3. ***Non-intersecting repair groups:*** *$\Gamma_j(y_i) \cap \Gamma_l(y_i) = \varnothing$ for all $i \in [k]$ and $j \neq l \in [t]$.*

Example 3. *[4] Consider a systematic code that encodes three information symbols (x_1, x_2, x_3) to a codeword $\vec{y}$ such that $\vec{y} = (x_1,\ x_2,\ x_3,\ x_1,\ x_1 + x_2,\ x_2 + x_3,\ x_1 + x_3)$. We will show that this codeword belongs to a $(7, 3, 2, 2)$ LRC.*

It should be obvious that $n = 7$ (the length of the codeword $\vec{y}$) and $k = 3$ (the number of information symbols). We now show that the code under consideration has locality and availability equal to 2 each by showing that all three properties listed in Definition 1 are satisfied.

Property 1: *t* ***repair groups each.*** The following are the repair groups of the first symbol x_1:

$$\begin{aligned} \Gamma_1(x_1) &= \{4\}, \\ \Gamma_2(x_1) &= \{2, 5\}, \end{aligned}$$

which we can read as: the second repair group of code symbol x_1 contains the second and fifth symbols of the code (viz. x_2 and $x_1 + x_2$).

The repair groups of x_2 and x_3 follow:

$$\Gamma_1(x_2) = \{1, 5\}, \qquad \Gamma_2(x_2) = \{3, 6\},$$
$$\Gamma_1(x_3) = \{2, 6\}, \qquad \Gamma_2(x_3) = \{1, 7\}.$$

We see that all three information symbols (viz. x_1, x_2, x_3) have two repair groups each.

Property 2: Locality of systematic symbols. We see, from Property 1, that each group is of size at most 2. Thus, $r = 2$.

Property 3: Non-intersecting repair groups. Again, from the discussion on Property 1, we see that no two repair groups of a given symbol intersect with one another. ■

It is shown in [4] that the following bound holds for an (n, k, r, t) LRC:

$$d \;\leq\; n - k - \left\lceil \frac{t(k-1)+1}{t(r-1)+1} \right\rceil + 2.$$

A special case of LRCs, where $t = 1$, was studied in [2].

2.3 Cooperative Locally Repairable Codes (CLRCs)

Definition 2. *[3] An (n, k, z, l) CLRC satisfies the following properties.*

1. ***Locality of erased symbols:*** *for any codeword $\vec{y} = (y_1, y_2, \cdots, y_n)$, any l code symbols indexed by a set $S \subset [n]$ are functions of up to z other code symbols indexed by another set Γ_S.*
2. ***Non intersecting sets of erased and repair symbols:*** *$S \cap \Gamma_S = \emptyset$, where $|S| = l$ and $|\Gamma_S| \leq z$.*

Example 4. *We will consider the same codeword from Example 3 and show that it also belongs to a $(7, 3, 2, 1)$ CLRC.*

The following is the codeword under consideration: $\vec{y} = (x_1,\ x_2,\ x_3,\ x_1,\ x_1 + x_2,\ x_2 + x_3,\ x_1 + x_3)$. It should again be obvious that $n = 7$ and $k = 3$.

What remains to be shown is that $z = 2$ and $l = 1$. In other words, every code symbol can be repaired with up to 2 other code symbols from the codeword. We demonstrate this below.

x_1 can be repaired with the fourth code symbol which is also x_1.

x_2 can be repaired with the fourth and fifth symbols: x_1 and $x_1 + x_2$.

x_3 can be repaired with x_1 and $x_1 + x_3$.

x_1 (the fourth symbol) can be repaired with the first symbol.

$x_1 + x_2$ can be repaired with x_1 and x_2.

$x_2 + x_3$ is repaired with x_2 and x_3.

$x_1 + x_3$ can be repaired with the first and third symbols: x_1 and x_3.

■

The following bound holds for CLRCs, as shown in [3]:

$$d \ \leq \ n - k + 1 - l\left(\left\lceil \frac{k}{z} \right\rceil - 1\right).$$

With a minor amendment of the proof of Theorem 1 in [3], we obtain a tighter bound:

$$d \ \leq \ n - k + 1 - l\left\lceil \frac{k-l}{z} \right\rceil.$$

2.4 Multi-symbol Locally Repairable Codes (MLRCs)

An MLRC is a new generalization of LRCs that we introduce in this thesis. We will perform a detailed study on MLRCs in the next chapter. In this section, we introduce the reader to the basics of MLRCs.

Definition 3. *An (n, k, r, t) MLRC satisfies the following properties.*

1. *t* ***non-intersecting repair groups in total****: A tuple of t symbols $(y_{i_1}, y_{i_2}, \cdots, y_{i_t})$, where the elements in the tuple are not necessarily distinct,*

can be reconstructed from t disjoint sets $\Gamma_j(y_i)$ such that $\Gamma_j(y_i) \subseteq [n] \setminus \{i_1, i_2, \cdots, i_t\}$; $i \in [n]$, $j \in [t]$.

2. ***Locality of encoded symbols**: $|\Gamma_j(y_i)| \leq r$, for all $i \in [n], j \in [t]$.*

Example 5. *In this example, we discuss that the following encoding of (x_1, x_2, x_3) belongs to a $(10, 3, 2, 2)$ MLRC: $\vec{y} = (x_1, x_2, x_3, x_1, x_2, x_3, x_1, x_1 + x_2, x_2 + x_3, x_1 + x_3)$.*

Since the length of $\vec{y}$ is 10 and $\vec{y}$ encodes three information bits, it should be no surprise that $n = 10$ and $k = 3$. We will argue that the codeword under discussion has locality and availability equal to 2 each by discussing that the properties listed in Definition 3 are satisfied.

Property 1: t non-intersecting repair groups in total. Since there are 10 symbols in the codeword $\vec{y}$, we have $\binom{11}{2} = 55$ choices of tuples with 2 symbols (allowing for repetition). To show that $t = 2$, we need to consider each of these 55 tuples and show that each symbol in the tuple can be repaired with disjoint sets of symbols; we will show this for a few tuples only and invite the reader to think about why our claim would hold for the remaining (many) tuples.

- A tuple of the first symbol and its repetition: (x_1, x_1) can be repaired with the fourth and seventh symbols of the codeword.
- A tuple of the second and eighth symbols: $(x_2, x_1 + x_2)$ can be repaired with the fifth, ninth and tenth symbols. We write this formally as:

$$\Gamma_1(x_2) = \{5\}, \qquad \Gamma_2\left(x_1 + x_2\right) = \{9, 10\}.$$

- A tuple of the eighth symbol and its repetition: $(x_1 + x_2, x_1 + x_2)$ can be repaired in the following way:

$$\Gamma_1(x_1 + x_2) = \{1, 2\}, \qquad \Gamma_2(x_1 + x_2) = \{4, 5\}.$$

Property 2: Locality of encoded symbols. From the discussion on Property 1, we see that each symbol could be repaired with no more than 2 other symbols.

We would need to show that this holds for each of the 55 tuples mentioned in Property 1 to convince the reader that each encoded symbol has locality, $r = 2$.

■

In the next chapter, we will show that the following bound holds for MLRCs:

$$d \ \leq \ n - k + 1 - \lambda \left\lceil \frac{k - \lambda}{rt} \right\rceil,$$

for any $\lambda \in \mathbb{N}$, $\lambda \leq t$ and $\lambda \leq k$.

Chapter 3

The Multiple Repair Model

3.1 An Upper Bound on the Minimum Distance of MLRCs

Our multiple repair model (MLRC) allows for the repair of a collection of symbols where repetitions are allowed and each symbol is repaired independently from a set of other symbols of size at most r; this is different from other models that have appeared in the literature. The following theorem derives an upper bound on the minimum distance of MLRCs. We introduce, and later optimize over, a new parameter λ that denotes the number of distinct symbols in our collection of symbols that require repair.

Theorem 1. *Let $\mathcal{C}$ be an (n, k, r, t) MLRC. Then, the minimum distance of $\mathcal{C}$ satisfies*

$$d \;\leq\; n - k + 1 - \lambda \left\lceil \frac{k-\lambda}{rt} \right\rceil, \tag{3.1}$$

for any $\lambda \in \mathbb{N}$, $\lambda \leq t$ and $\lambda \leq k$.

Proof. We derive an upper bound on the minimum distance of $\mathcal{C}$ by constructing a sub-code $\mathcal{C}'$ in such a way that a number of coordinates in each codeword of $\mathcal{C}'$ are fixed. This allows us to remove those coordinates from $\mathcal{C}'$ to obtain a smaller code $\mathcal{C}''$ with $|\mathcal{C}'| = |\mathcal{C}''|$ and $d_{\min}(\mathcal{C}') = d_{\min}(\mathcal{C}'')$.

From the definition of the minimum distance of a code, we have

$$d_{\min}(\mathcal{C}) \leq d_{\min}(\mathcal{C}'),$$

which then means $d_{\min}(\mathcal{C}) \leq d_{\min}(\mathcal{C}'')$; we will use this inequality in our analysis.

We describe the construction of the sub-code $\mathcal{C}'$ in Algorithm 1, which is based on techniques from [4].

The idea of Algorithm 1 is to project a given code onto the repair set of the t symbols chosen in the given iteration. From the projection, the most 'popular' vector is chosen, and codewords that correspond to that vector are put together to construct a sub-code for the particular iteration. This is repeated until the *while* loop breaks.

For the analysis, we define a set $\mathcal{I}_j$ as the union of symbols chosen *up to* the j-th iteration and each symbol's repair group:

$$\mathcal{I}_j = \bigcup_{j' \in [j]} \Big(R_{j'} \cup \big\{ i_1^{j'}, i_2^{j'}, \cdots, i_\lambda^{j'} \big\} \Big).$$

Next, we define $\mathcal{A}_j$ as the set consisting of symbols chosen *in* the j-th iteration, along with each symbol's repair group:

$$\begin{aligned} \mathcal{A}_j &= \mathcal{I}_j \setminus \mathcal{I}_{j-1}, \\ \mathcal{A}_j &\subseteq R_j \cup \big\{ i_1^j, i_2^j, \cdots, i_\lambda^j \big\}, \\ a_j &= |\mathcal{A}_j|. \end{aligned}$$

Notice that $\mathcal{I}_j$ is a union of sets $\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_j$:

$$\mathcal{I}_j = \bigcup_{j' \in [j]} \mathcal{A}_{j'}.$$

We describe the relations between sets $\mathcal{I}_j$ and $\mathcal{A}_j$ in figure 3.1.

Algorithm 1 Construction of a sub-code $\mathcal{C}'(\subseteq \mathcal{C})$.

Input:
- an (n, k, r, t) MLRC $\mathcal{C}$ over $\mathbb{F}_q$,
- $\lambda \in \mathbb{N}$ such that $\lambda \leq t$ and $\lambda \leq k$.

1: $\mathcal{C}_0 = \mathcal{C}$.
2: $\mathcal{C}' = \mathcal{C}_0$.
3: $j = 0$.
4: **while** $|\mathcal{C}_j| > q^\lambda$ **do**
5: $j = j + 1$.
6: Choose a set of indices $\Lambda = \{i_1^j, i_2^j, \cdots, i_\lambda^j\}$ such that there exist at least 2 codewords in $\mathcal{C}_{j-1}$ that differ at the i_m^j-th coordinate, for every $m \in [\lambda]$; $i_m^j \in [n]$.
7: Define a collection of $t - \lambda$ indices: $i_{\lambda+1}^j, i_{\lambda+2}^j, \cdots, i_t^j$, such that each of the indices belong to Λ. $i_1^j, i_2^j, \cdots, i_t^j$ denote t indices chosen from Λ where repetitions are allowed.
8: Let R_j be the index of at most rt code symbols that repair the t symbols: $i_1^j, i_2^j, \cdots, i_t^j$.
9: Let $\vec{y}$ $(\in \mathbb{F}_q^{|R_j|})$ be the most frequent element in the set $\left\{\vec{x}|_{R_j} : \vec{x} \in \mathcal{C}_{j-1}\right\}$.
10: Define $\mathcal{C}_j = \left\{\vec{x} : \vec{x} \in \mathcal{C}_{j-1} \text{ and } \vec{x}|_{R_j} = \vec{y}\right\}$.
11: **if** $1 < |\mathcal{C}_j| \leq q^\lambda$ **then**
12: $\mathcal{C}' = \mathcal{C}_j$.
13: **end while**.
14: **else if** $|\mathcal{C}_j| = 1$ **then**
15: Pick a maximal subset $\tilde{R}_j \subset R_j$ and go to line 8; replace occurrences of R_j by $\tilde{R}_j$.
16: **end if**
17: **end while**

Output: $\mathcal{C}'$.

Since $\mathcal{A}_j$ consists of all symbols chosen in the j-th iteration, $|R_j| \leq a_j$; R_j is the set of code symbols needed for repair in the j-th iteration. However, notice that there are λ symbols chosen in the j-th iteration that do not appear in R_j, which means $|R_j| \leq a_j - \lambda$.

There are $q^{|R_j|}$ possibilities for $\vec{y}$ (see Algorithm 1), and thus the code $\mathcal{C}_{j-1}$ can

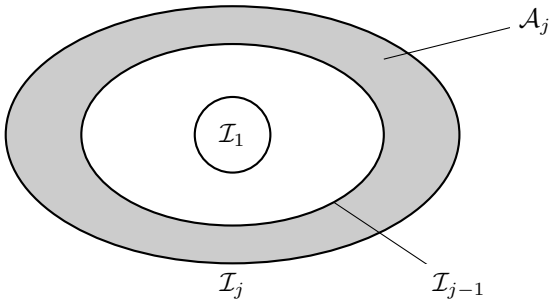

Figure 3.1: Relationships between $\mathcal{I}_j$ and $\mathcal{A}_j$.

be truncated by this factor in the j-th iteration:

$$|\mathcal{C}_j| \;\geq\; \frac{|\mathcal{C}_{j-1}|}{q^{|R_j|}} \;\geq\; \frac{|\mathcal{C}_{j-1}|}{q^{a_j-\lambda}}.$$

Notice that, through induction on j (base case: $j = 1$), we obtain the following:

$$|\mathcal{C}_{j'}| \;\geq\; \frac{|\mathcal{C}_0|}{q^{\sum_{j=1}^{j'}\left(a_j-\lambda\right)}}. \tag{3.2}$$

Assuming that the while loop (Algorithm 1) terminates when $j = \tau$, we have $|\mathcal{C}_\tau| \leq q^\lambda$. This, along with (3.2), gives us the following:

$$\begin{aligned}
\lambda &\geq \log_q |\mathcal{C}_\tau| \\
&\geq \log_q |\mathcal{C}_0| - \log_q q^{\sum_{j=1}^{\tau}\left(a_j-\lambda\right)} \\
&= k - \sum_{j=1}^{\tau}\left(a_j - \lambda\right), \\
k - \lambda &\leq \sum_{j=1}^{\tau}\left(a_j - \lambda\right). \qquad (3.3)
\end{aligned}$$

Recall that $a_j = |\mathcal{A}_j| \leq \left|R_j \cup \{i_1^j, i_2^j, \cdots, i_\lambda^j\}\right| \leq rt + \lambda$. So $\left(a_j - \lambda\right) \leq rt$, which plugged into (3.3) gives us

$$\begin{aligned}
k - \lambda &\leq rt\tau, \\
\tau &\geq \left\lceil \frac{k-\lambda}{rt} \right\rceil. \qquad (3.4)
\end{aligned}$$

From the inequalities leading up to (3.3) and noting that $\mathcal{C}_\tau = \mathcal{C}'$, we have

$$\begin{aligned} \log_q |\mathcal{C}'| &= \log_q |\mathcal{C}_\tau| \\ &\geq k - \sum_{j=1}^{\tau} \big(a_j - \lambda\big) \\ &= k - |\mathcal{I}_\tau| + \tau\lambda, \end{aligned} \tag{3.5}$$

where $\mathcal{I}_\tau$ is the union of sets $\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_\tau$.

We now define a code $\mathcal{C}''$ that is formed by puncturing $\mathcal{C}'$ at symbols indexed by $\mathcal{I}_\tau$. The length of $\mathcal{C}''$ is $n - |\mathcal{I}_\tau|$ and $|\mathcal{C}''| = |\mathcal{C}'|$. Applying the Singleton bound to $\mathcal{C}''$ gives us

$$\begin{aligned} d_{\min}(\mathcal{C}) \leq d_{\min}(\mathcal{C}'') &\leq n - |\mathcal{I}_\tau| - \log_q |\mathcal{C}''| + 1 \\ &\overset{a}{\leq} n - |\mathcal{I}_\tau| - (k - |\mathcal{I}_\tau| + \tau\lambda) + 1 \\ &\leq n - k + 1 - \tau\lambda, \end{aligned} \tag{3.6}$$

where 'a' follows from (3.5) and the fact that $|\mathcal{C}'| = |\mathcal{C}''|$.

By combining (3.4) and (3.6), we finally get

$$d \leq n - k + 1 - \lambda \left\lceil \frac{k-\lambda}{rt} \right\rceil.$$

□

Corollary 1. *Let $\mathcal{C}$ be an (n, k, r, t) MLRC. Then, the following bound is satisfied:*

$$t \leq n - k - \lambda \left\lceil \frac{k-\lambda}{rt} \right\rceil,$$

for any $\lambda \in \mathbb{N}$, $\lambda \leq t$ and $\lambda \leq k$.

Proof. Noting that the minimum distance of $\mathcal{C}$ must be greater than t to allow for the repair of t erasures, we have

$$d_{\min}(\mathcal{C}) \geq t + 1,$$

which plugged into (3.1) yields

$$t \leq n - k - \lambda \left\lceil \frac{k-\lambda}{rt} \right\rceil.$$

□

3.2 Analysis

We begin this section with a discussion on optimizing the bound presented in Theorem 1. We then compare said bound with existing bounds for LRCs and CLRCs.

Example 6. *In this example, we estimate a value of λ that optimizes the bound in Theorem 1.*

Notice that said bound can be seen as an adjustment of the Singleton bound ($d \leq n - k + 1$) for MLRCs. We see that the right hand side of (3.1) is smaller than that of the Singleton bound by the following term:

$$f(\lambda) = \lambda \left\lceil \frac{k-\lambda}{rt} \right\rceil,$$

where $f(\lambda)$ can be thought of as a penalty for adding local repair properties atop classical error correcting codes. By taking a derivative of $f(\lambda)$ with respect to λ and ignoring the ceiling operation, we obtain

$$\frac{d(f(\lambda))}{d\lambda} = \frac{k}{rt} - \frac{2\lambda}{rt}.$$

Equating the derivative to zero gives us a value of λ that optimizes the bound in Theorem 1:

$$\begin{aligned} \frac{k}{rt} - \frac{2\lambda}{rt} &= 0, \\ \lambda &= \frac{k}{2}. \end{aligned}$$

Plugging in $\lambda = \frac{k}{2}$ into the bound under discussion, we have:

$$d \leq n - k + 1 - \left\lceil \frac{k}{2} \left\lceil \frac{k}{2rt} \right\rceil \right\rceil.$$

■

From the bound established in Theorem 1, we see that the following penalty is incurred with MLRCs:

$$p_{\text{MLRC}} \;=\; \lambda\left\lceil\frac{k-\lambda}{rt}\right\rceil.$$

The following upper bound for LRCs has been established in [4]:

$$d \;\leq\; n-k+1-\left\lceil\frac{t(k-1)+1}{t(r-1)+1}\right\rceil+\,1,$$

so, we have the following penalty for LRCs:

$$p_{\text{LRC}} \;=\; \left\lceil\frac{kt-t+1}{rt-t+1}\right\rceil-\,1.$$

Since MLRCs are more restrictive than LRCs, we expect the following:

$$\begin{aligned} p_{\text{MLRC}} &\geq p_{\text{LRC}}, \\ \lambda\left\lceil\frac{k-\lambda}{rt}\right\rceil &\geq \left\lceil\frac{kt-t+1}{rt-t+1}\right\rceil-1. \end{aligned} \tag{3.7}$$

We discuss further on (3.7) in Example 7.

Example 7. *We show that inequality (3.7) holds for $k=2t$ and $\lambda=t$.*

Plugging in $k=2t$ and $\lambda=t$ into (3.7), we obtain

$$\begin{aligned} t\left\lceil\frac{2t-t}{rt}\right\rceil &\geq \left\lceil\frac{2t^2-t+1}{rt-t+1}\right\rceil-1, \\ t\left\lceil\frac{1}{r}\right\rceil &\geq \left\lceil\frac{2t^2}{rt}\right\rceil-1, \\ t &\geq \left\lceil\frac{2t}{r}\right\rceil-1. \end{aligned} \tag{3.8}$$

To show that (3.8) holds, we notice that for $r \geq 2$,

$$\begin{aligned} \frac{2}{r} &\leq 1, \\ \left\lceil \frac{2t}{r} \right\rceil &\leq \lceil t \rceil, \\ \left\lceil \frac{2t}{r} \right\rceil - 1 &\leq \lceil t \rceil - 1, \end{aligned}$$

and since $t + 1 \geq \lceil t \rceil$,

$$t \geq \lceil t \rceil - 1 \geq \left\lceil \frac{2t}{r} \right\rceil - 1.$$

Thus, (3.8) holds. ■

Example 8. *In this example, we make another comparison of our bound in Theorem 1 with a bound for CLRCs.*

The following bound was established for the minimum distance of CLRCs in Section 2.3:

$$d \ \leq \ n - k + 1 - l \left\lceil \frac{k-l}{z} \right\rceil. \tag{3.9}$$

Noting that l represents the number of erasures and z represents the maximum number of symbols required to repair the erasures, we can translate (3.9) to our model of MLRCs to give the following penalty for CLRCs :

$$p_{\text{CLRC}} = t \left\lceil \frac{k-t}{rt} \right\rceil.$$

We would like to show that $p_{\text{MLRC}} \geq p_{\text{CLRC}}$. That is,

$$\lambda \left\lceil \frac{k-\lambda}{rt} \right\rceil \geq t \left\lceil \frac{k-t}{rt} \right\rceil. \tag{3.10}$$

If $t = k$ and $\lambda = \frac{k}{2}$, (3.10) becomes

$$\frac{k}{2} \left\lceil \frac{k}{2rt} \right\rceil \geq 0,$$

which is always true. ■

3.3 Further Directions

The reader may have noticed that our model of MLRCs considers a worst-case scenario where each server under consideration has mandatorily failed; in other words, every symbol to be repaired has necessarily been erased. A reasonable extension to our model would be to consider situations where not all the servers have failed, yet load balancing requirements in the distributed system necessitate the use of repair groups.

Further, all work on locality and availability thus far *have* concentrated on adapting the Singleton bound to the new concept of local repair. There are many other bounds in classical coding theory where the notion of local repair has not been introduced. A possible research direction would be to study such bounds in light of local repair requirements.

Acknowledgments

I would like to express sincere gratitude to the European Commission for funding my participation in the NordSecMob program. Thanks to Dr. Vitaly Skachek for his guidance and Zhang Hui for helpful discussions. I appreciate all your support baba, mamu, shubu.

Bibliography

[1] A.G. Dimakis, P.B. Godfrey, Y. Wu, M.J. Wainwright, and K. Ramchandran. Network coding for distributed storage systems. *Information Theory, IEEE Transactions on*, 56(9):4539–4551, Sept 2010.

[2] Parikshit Gopalan, Cheng Huang, Huseyin Simitci, and Sergey Yekhanin. On the locality of codeword symbols. *Information Theory, IEEE Transactions on*, 58(11):6925–6934, Nov 2012.

[3] Ankit Singh Rawat, Arya Mazumdar, and Sriram Vishwanath. On cooperative local repair in distributed storage. In *Information Sciences and Systems (CISS), 2014 48th IEEE Annual Conference on*, pages 1–5.

[4] Ankit Singh Rawat, Dimitris S. Papailiopoulos, Alexandros G. Dimakis, and Sriram Vishwanath. Locality and availability in distributed storage. *CoRR*, abs/1402.2011, 2014.

[5] Richard C Singleton. Maximum distance q-nary codes. *Information Theory, IEEE Transactions on*, 10(2):116–118, 1964.

[6] Itzhak Tamo and Alexander Barg. Bounds on locally recoverable codes with multiple recovering sets. In *Information Theory (ISIT), 2014 IEEE International Symposium on*, pages 691–695.

[7] Anyu Wang and Zhifang Zhang. Repair locality with multiple erasure tolerance. *CoRR*, abs/1306.4774, 2013.

Non-exclusive license to reproduce thesis and make thesis public

I, Sushanta Paudyal,

1. herewith grant the Norwegian University of Science and Technology and the University of Tartu a free permit (non-exclusive license) to

   (a) reproduce for the purpose of preservation, including for addition to the DSpace digital archives, until expiry of the term of validity of the copyright and

   (b) make available to the public via the web environment of the Norwegian University of Science and Technology and the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright:

   Multi-symbol Locally Repairable Codes, supervised by Vitaly Skachek and Colin Boyd.

2. I am aware of the fact that I retain the copyright.

3. I certify that granting the non-exclusive license does not infringe upon intellectual property rights or rights arising from the Personal Data Protection Act.

June 4, 2015.