UNIVERSITY OF CALIFORNIA, SAN DIEGO

List Decoding of Subspace Codes and Rank-Metric Codes

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering
(Communication Theory and Systems)

by

Hessam Mahdavifar

Committee in charge:

Professor Alexander Vardy, Chair
Professor Daniele Micciancio
Professor Alon Orlitsky
Professor Daniel Rogalsky
Professor Paul H. Siegel

2012

The dissertation of Hessam Mahdavifar is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2012

DEDICATION

Dedicated to my parents

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

First of all, I would like to thank my advisor Professor Alexander Vardy for invaluable guidance, encouragement and support in so many ways during the past five years. This work would not be possible without all the great help and support that he provided for me. I am also thankful to my thesis committee members Professor Daniele Micciancio, Professor Alon Orlitsky, Professor Daniel Rogalsky and Professor Paul Siegel for being wonderful mentors. I learned a lot from them by having the opportunity of taking courses with all of them as well as having personal discussions with them on various interesting topics. I would like to thank my loving and caring family for the faithful support. Although, I did not have the chance of visiting them during my PhD, I have always felt their care and love from such a long distance. I am also thankful to all my friends in San Diego. Their help and support made me feel home and gave me confidence and strength along this journey.

The results of Chapter 2 was first presented in part at ISIT 2010 in Austin and appeared in the proceedings as: H. Mahdavifar and A.Vardy, "Algebraic List-decoding on the Operator Channel", *Proceedings of the 2010 IEEE International Symposium on Information Theory*, pp. 1193-1197, Austin, Texas, June 2010. A full version of it was submitted to IEEE transactions on information theory and was recently accepted for publication. It is available on arxiv as: H. Mahdavifar and A. Vardy, "Algebraic List-decoding of Subspace Codes", available at http://arxiv.org/pdf/1202.0338.pdf. Some parts of Chapter 3 was presented at Allerton conference 2011 and appeared in proceedings as: H. Mahdavifar and A.Vardy, "Algebraic List-decoding of Subspace Codes with Multiplicities", *Proceedings of the 49th annual Allerton Conference on Communications, Control and Computing*, September 2011. Chapter 4 and 5 are in part a reprint of the material that is going to be presented at ISIT 2012 in Boston and is also available on arxiv as: H. Mahdavifar and A. Vardy, "List-decoding of Subspace Codes and Rank-Metric Codes up to Singleton Bound," available at http://arxiv.org/pdf/1202.0866.pdf.

VITA

2007	B.Sc. in Electrical Engineering, Sharif University of Technology
2009	M.Sc. in Electrical Engineering (Communication Theory and Systems), University of California, San Diego
2012	Ph.D. in Electrical Engineering (Communication Theory and Systems), University of California, San Diego

PUBLICATIONS

H. Mahdavifar and A. Vardy, "Algebraic List-decoding of Subspace Codes", *IEEE Transactions on Information Theory*, Accepted for publication, available at

http://arxiv.org/pdf/1202.0338.pdf

H. Mahdavifar and A. Vardy, "List-decoding of Subspace Codes and Rank-Metric Codes up to Singleton Bound," to be submitted to *IEEE Transactions on Information Theory*, available at http://arxiv.org/pdf/1202.0866.pdf

E. Yaakobi, H. Mahdavifar, P.H. Siegel, A. Vardy, J.K. Wolf, "Rewriting Codes for Flash Memories", Submitted to *IEEE Transactions on Information Theory*

H. Mahdavifar and A. Vardy, "List-decoding of Subspace Codes and Rank-Metric Codes up to Singleton Bound," to be presented at *2012 IEEE International Symposium on Information Theory*. Full version available at http://arxiv.org/pdf/1202.0866.pdf

H. Mahdavifar and A. Vardy, "Achieving the Secrecy Capacity of Wiretap Channels Using Polar Codes", *IEEE Transactions on Information Theory*, vol. 57, pp. 6428-6443, October 2011

H. Mahdavifar and A.Vardy, "Algebraic List-decoding of Subspace Codes with Multiplicities", *Proceedings of the 49th annual Allerton Conference on Communications, Control and Computing*, September 2011

H. Mahdavifar and A.Vardy, "Algebraic List-decoding on the Operator Channel", *Proceedings of the 2010 IEEE International Symposium on Information Theory*, pp. 1193-1197, Austin, Texas, June 2010

H. Mahdavifar and A.Vardy, "Achieving the Secrecy Capacity of Wiretap Channels Using Polar Codes", *Proceedings of the 2010 IEEE International Symposium on Information Theory*, pp. 913 - 917, Austin, Texas, June 2010

H. Mahdavifar and A.Vardy, "Optimal Interleaving Algorithms for Generalized Concatenated Codes", *Proceedings of the 47th annual Allerton Conference on Communications, Control and Computing (invited paper)*, October 2009

H. Mahdavifar, P.H. Siegel, A. Vardy, J.K. Wolf and E. Yaakobi, "A Nearly Optimal Construction of Flash Codes", *Proceedings of the 2009 IEEE International Symposium on Information Theory*, pp. 1239-1243, Seoul, Korea, June 2009

ABSTRACT OF THE DISSERTATION

List Decoding of Subspace Codes and Rank-Metric Codes

by

Hessam Mahdavifar

Doctor of Philosophy in Electrical Engineering
(Communication Theory and Systems)

University of California, San Diego, 2012

Professor Alexander Vardy, Chair

Subspace codes and rank-metric codes can be used to correct errors and erasures in networks with linear network coding. Both types of codes have been extensively studied in the past five years. We develop in this document list-decoding algorithms for subspace codes and rank-metric codes, thereby providing a better tradeoff between rate and error-correction capability than existing constructions.

Randomized linear network coding, considered as the most practical approach to network coding, is a powerful tool for disseminating information in networks. Yet it is highly susceptible to transmission errors caused by noise or intentional jamming. Subspace codes were introduced by Koetter and Kschischang to correct errors and erasures in networks with a randomized protocol where the topology is unknown (the non-

coherent case). The codewords of a subspace code are vector subspaces of a fixed ambient space; thus the codes are collections of such subspaces.

We first develop a family of subspace codes, based upon the Koetter-Kschichang construction, which are efficiently list decodable. We show that, for a certain range of code rates, our list-decoding algorithm provides a better tradeoff between rate and decoding radius than the Koetter-Kschischang codes. We further improve these results by introducing multiple roots in the interpolation step of our list-decoding algorithm. To this end, we establish the notion of derivative and multiplicity in the ring of linearized polynomials. In order to achieve a better decoding radius, we take advantage of enforcing multiple roots for the interpolation polynomial. We are also able to list decode for a wider range of rates. Furthermore, we propose an alternative approach which leads to a linear-algebraic list-decoding algorithm.

Rank-metric codes are suitable for error correction in the case where the network topology and the underlying network code are known (the coherent case). Gabidulin codes are a well-known class of algebraic rank-metric codes that meet the Singleton bound on the minimum rank-distance of a code. In this dissertation, we introduce a folded version of Gabidulin codes along with a list-decoding algorithm for such codes. Our list-decoding algorithm makes it possible to achieve the information theoretic bound on the decoding radius of a rank-metric code.

Chapter 1

Introduction

1.1 Randomized Network Coding

Network coding has been a very exciting and fast growing area of research in the past decade since it was introduced in 2000 [1]. The literature on network coding encompasses, by now, hundreds of papers contributed by people from various disciplines including coding theory, information theory, networks and wireless communications, security and secrecy etc. Ahlswede, Cai, Li, and Yeung, in their seminal paper [1], say that they "refer to coding at a node in a network as *network coding*". Network coding is basically a technique where, the nodes of a network take several packets and combine them together for transmission instead of simply relaying the packets of information they receive. In fact, network coding generalizes network operation beyond traditional routing approaches.

In a multicast setting, one source communicates simultaneously with several receivers in the network [1,12]. The most famous example of the network coding benefit was given by Ahlswede et al. [1] in a multicast setting. In their example, depicted in Figure 1.1 which is commonly referred to as the *butterfly network*, one source communicates with two receivers in the network. Both receivers r_1 and r_2 wish to know, in full, the message at the source node s. In this network, each edge represents a link capable of carrying one bit in one time unit reliably. There are two information bits b_1 and b_2 available at the source which we wish to transmit to both receivers reliably. This can be done separately for each of the receivers using routing, or store-and-forward, algo-

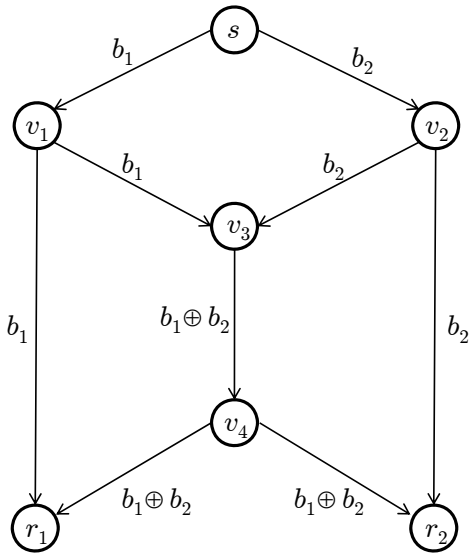


Figure 1.1: In the butterfly network, the source s communicates with two receivers r_1 and r_2 simultaneously.

rithm. It is impossible to transmit both b_1 and b_2 to both receivers, however, in one round of communication by a routing algorithm. The desired multicast operation can be established only if one of the intermediate nodes breaks the rule of traditional routing algorithm, where intermediates nodes are only allowed to transmit copies of what they receive, and performs a simple form of coding operation. The node v_3 in the network takes two received bits and computes the XOR of these two bits and outputs the result on its output link. The receiver r_1 receives b_1 and $b_1 \oplus b_2$ and can easily recover b_2 by XORing them. The receiver r_2 receives b_2 and $b_1 \oplus b_2$ and recovers b_1 as well by performing an XOR operation. In the butterfly network, the rate of information from the source to each receiver is 2 which is equal to the min-cut bound by max-flow min-cut theorem. Network coding enables us to simultaneously achieve the max-flow min-cut bound from the source to each receiver that is, 2 bits of information is transmitted to both of the receivers during each round of communication.

The network coding depicted in the butterfly network is also an example of *linear network coding*. In linear network coding, the underlying network coding operations performed at intermediate nodes are linear; that is, each intermediate node computes a

linear combination of the packets that are available at its input links and transmits the result through its output link. In a multicast setting, linear network coding is sufficient to achieve simultaneously the individual max-flow min-cut bound on the rate of communication between the source and each of the receivers [12].

Randomized linear network coding, first proposed in [9,10], is a powerful tool for disseminating information in networks. Since randomized network coding is completely distributed and decentralized, it is the most promising *practical* approach to network coding to date [2]. Suppose that the source injects into the network a set of packets $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$, all of the same length (the same number of symbols). These packets can be regarded as vectors of length m over a finite field $\mathbb{F}_q$. In the randomized setting, each intermediate node in the network generates random $\mathbb{F}_q$-linear combinations of the packets available at its input links and sends them out on its output links. Finally, receivers collect the packets on their input links and use this information in an attempt to recover $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$. It is proved in [10] that, with high probability, this randomized network coding protocol achieves the max-flow min-cut bound (cf. [1,12]), simultaneously for each receiver, provided the size q of the underlying field $\mathbb{F}_q$ is sufficiently large.

Randomized linear network coding is highly susceptible to transmission errors caused by noise or intentional jamming. Even a single packet error injected in the network could potentially render the entire transmission useless. Packets can also become lost (erased), so that the problem of deducing the transmitted message at the receiver cannot be completed. Errors in this model correspond to the injection of erroneous packets into the network, either by malicious nodes or through link mis-connections, that do not belong to the linear space spanned by the source vectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$; erasures (lost packets) correspond to the projection of $\langle \boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n \rangle$ onto a lower-dimensional subspace. The problem of error-control for randomized network coding was first addressed in the pioneering work of Koetter and Kschischang [11]. They introduced the *operator channel* to capture the essence of randomized network coding. Furthermore, motivated by the fact that randomized network coding is vector-space preserving, Koetter and Kschischang [11] introduced error-correcting codes in projective space [7,11], also known as *subspace codes*. We review the basics of the operator channel model and subspace codes in the next two sections.

1.2 Operator Channel

Koetter and Kschischang [11] introduced the operator channel model in order to capture the essence of randomized linear network coding for multicast, in the case where the network topology is unknown (the non-coherent case). This problem is formulated for the case of a single unicast; that is, communication between a single source and a single receiver. Generalizations to multicasts and sets of disjoint unicasts are relatively straightforward.

Recall [2] that communication between a source and receiver is done in a series of rounds or *generations*. During each generation, the transmitter injects a number of packets into the network; each of them is regarded as a vector of length N over a finite field $\mathbb{F}_q$. These packets pass through the intermediate nodes of the network to reach the targeted receiver node. Each intermediate node in the network creates random $\mathbb{F}_q$-linear combinations of the packets available at its incoming edges and sends them out on its outgoing edges. The receiver collects a number of such network generated packets and tries to infer the set of packets injected into the network.

Let $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$, $\boldsymbol{v}_i \in {\mathbb{F}_q}^N$, be the vectors injected into the network. Suppose that the receiver collects r vectors $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_r$, where the $\boldsymbol{u}_i$'s are also vectors in the vector space ${\mathbb{F}_q}^N$. If there is no error injected into the network, then each $\boldsymbol{u}_i$ is a linear combination of $\boldsymbol{v}_j$'s; that is, $\boldsymbol{u}_i = \sum_{j=1}^{n} h_{i,j}\boldsymbol{v}_j$, where $h_{i,j} \in \mathbb{F}_q$ are unknown coefficients. In fact, the $h_{i,j}$'s are determined by the particular linear combinations created at intermediate nodes. These linear combinations are assumed to be generated completely at random, hence becoming unknown at the receiver.

Consider the scenario that some erroneous packets are injected into the network, for instance as a result of corruption in some of the network links or by some malicious nodes trying to disturb the communication by injecting erroneous packets into the network. Suppose that t erroneous packets $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_t$ are injected into the network. Then each received packet $\boldsymbol{u}_i$ can be expressed as

$$\boldsymbol{u}_i = \sum_{j=1}^{n} h_{i,j}\boldsymbol{v}_j + \sum_{j=1}^{t} g_{i,j}\boldsymbol{e}_j$$

where $\boldsymbol{e}_j$'s and $g_{i,j}$'s are unknown to the receiver. The set of all such equations for

$i = 1, 2, \ldots, r$ can be expressed as a single matrix equation that is,

$$\boldsymbol{U} = H\boldsymbol{V} + G\boldsymbol{E} \tag{1.1}$$

where $\boldsymbol{U}$ is an $r \times N$ matrix whose rows represent the received packets $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_r$, $\boldsymbol{V}$ is an $n \times N$ matrix whose rows are the transmitted packets by the source node, H and G are random matrices of dimensions $r \times n$ and $r \times t$ respectively, and $\boldsymbol{E}$ is a $t \times N$ matrix whose rows are the error vectors $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_t$.

Since the matrix H is random and is not known to the receiver, the natural question is: How can we convey information to the receiver, even in the absence of noise? Suppose that H is full column-rank. Then the only thing that is preserved when we multiply $\boldsymbol{V}$ by H, is the vector space spanned by the rows of $\boldsymbol{V}$ that is, the linear span of the set of packets injected into the network. Indeed, as far as the receiver is concerned, any basis for this vector space is the same as any other one. Therefore, Koetter and Kschischang are led to consider the information transmission by the choice of the vector space spanned by the set of packets injected into the network. This observation is the main motivation for the *operator channel* model and codes constructed for this model called *subspace codes*.

Let $\mathcal{W}$ be a fixed vector space over $\mathbb{F}_q$, called the ***ambient space***. Let $N = \dim \mathcal{W}$. All the packets in the network are viewed as elements of $\mathcal{W}$. Let $\mathscr{P}_q(\mathcal{W})$ denotes the set of all subspaces of $\mathcal{W}$. Let also $\mathcal{G}_q(\mathcal{W}, n)$ denotes the set of all subspaces of $\mathcal{W}$ of dimension n.

Definition 1.2.1. *[11] An **operator channel** $\mathscr{C}$ associated with the ambient space $\mathcal{W}$ is a channel whose input and output alphabets are $\mathscr{P}_q(\mathcal{W})$. If a vector space V is the input to $\mathscr{C}$, the corresponding output vector space U is given by:*

$$U \;=\; \mathcal{H}_k(V) \oplus E \tag{1.2}$$

*where E is an **error vector space** such that $E \cap V = \{\mathbf{0}\}$ and $\mathcal{H}_k$ is the **erasure operator**. The erasure operator $\mathcal{H}_k$ projects V onto a k-dimensional subspace of V chosen uniformly at random, provided $\dim V > k$; otherwise, $\mathcal{H}_k$ leaves V unchanged. The number of errors and erasures that occurred during the transmission over the operator channel $\mathscr{C}$ are defined as $t = \dim E$ and $\rho = \dim V - \dim \mathcal{H}_k(V)$, respectively.*

$\mathcal{H}_k(V) \oplus E$ is the direct sum of $\mathcal{H}_k(V)$ and E which is by definition the set $\{\boldsymbol{v} + \boldsymbol{e} \ : \ \boldsymbol{v} \in \mathcal{H}_k(V), \boldsymbol{e} \in E\}$. In general, $\mathcal{H}_k(V)$ and E may have a non-trivial intersection. However, E can be always decomposed as $E = (E \cap V) \oplus E'$ for some vector space E' such that $E' \cap V = \{\mathbf{0}\}$. Then E' can be regarded as the actual error vector space, while $E \cap V$ may be only helpful by possibly recovering some part of transmitted space V lost due to erasure. Therefore, we may always assume that $E \cap V = \{\mathbf{0}\}$.

If the matrix H in (1.1) is full column-rank, then there will be no erasures. Indeed $\boldsymbol{V}$ and $H\boldsymbol{V}$ will have the same row space in that case. In general, matrix H depends on the random coefficients picked at intermediate nodes as well as the network topology. For instance, if the min-cut between the source and the receiver is less than n, then H can not be full column-rank, no matter how large r is and how the intermediate nodes pick their coefficients. If the min-cut between the source and the receiver is at least n, then H may or may not be full rank depending on the random linear combinations performed at intermediate nodes. It can be proved that if the size of the field q is large enough and the coefficients at each intermediate node are picked completely at random, then with very high probability, H is full-column rank.

1.3 Subspace Codes

In the previous section, we reviewed the operator channel model whose input and output alphabet is the set of all subspaces of an ambient space $\mathcal{W}$ denoted by $\mathscr{P}_q(\mathcal{W})$. In order to define codes for the operator channel, first we need to define a metric on $\mathscr{P}_q(\mathcal{W})$. A distance function on $\mathscr{P}_q(\mathcal{W})$ is defined in [11] as follows:

Definition 1.3.1. *Let $\mathbb{N}$ be the set of non-negative integers. The function $d \ : \ \mathscr{P}_q(\mathcal{W}) \times \mathscr{P}_q(\mathcal{W}) \rightarrow \mathbb{N}$ is defined as*

$$d(A, B) \stackrel{\text{def}}{=} \dim(A + B) \ - \ \dim(A \cap B) \tag{1.3}$$

where $A + B = \{a + b \ : \ a \in A, b \in B\}$ denotes the sum of spaces A and B.

Notice that

$$\dim(A + B) = \dim(A) + \dim(B) - \dim(A \cap B)$$

Therefore, the distance between A and B can be also written as

$$d(A,B) = \dim(A) + \dim(B) - 2\dim(A \cap B)$$

The following lemma proves that the distance function $d(\cdot,\cdot)$ makes the set $\mathscr{P}_q(\mathcal{W})$ a metric space.

Lemma 1.3.2. *[11] The distance function*

$$d(A,B) = \dim(A) + \dim(B) - 2\dim(A \cap B)$$

is a metric for the set $\mathscr{P}_q(\mathcal{W})$.

Proof. We need to verify the following three conditions:

1. $d(A,B) \geqslant 0$ and equality happens if and only if A and B are identical.
2. $d(A,B) = d(B,A)$.
3. $d(A,C) \leqslant d(A,B) + d(B,C)$ for any $A, B, C \in \mathscr{P}_q(\mathcal{W})$

Notice that $A \cap B$ is always a subspace of $A+B$. Also, these two become identical if and only if A and B are equal. Therefore, the first condition is always satisfied. The second condition is trivial. For the third condition, we plug in the formula for the distance function. Then it becomes equivalent to

$$\dim(A \cap B) + \dim(B \cap C) \leqslant \dim(B) + \dim(A \cap C) \tag{1.4}$$

Notice that $A \cap B$ and $B \cap C$ are both subspaces of B and hence, $(A \cap B) + (B \cap C)$ is a subspace of B. Therefore,

$$\dim\big((A \cap B) + (B \cap C)\big) = \dim(A \cap B) + \dim(B \cap C) - \dim(A \cap B \cap C) \leqslant \dim(B) \tag{1.5}$$

Also, we have

$$\dim(A \cap B \cap C) \leqslant \dim(A \cap C) \tag{1.6}$$

Adding (1.5) and (1.6) results in (1.4). ■

Definition 1.3.3. *A subspace code $\mathbb{C}$ for an operator channel with ambient space $\mathcal{W}$ is a non-empty subset of $\mathscr{P}_q(\mathcal{W})$. Thus codewords of $\mathbb{C}$ are subspaces of $\mathcal{W}$. Also, The minimum distance of $\mathbb{C}$ is given by*

$$d(\mathbb{C}) \stackrel{\text{def}}{=} \min_{\substack{X,Y\in\mathbb{C}\\ X\neq Y}} d(X,Y)$$

A minimum-distance decoder for the code $\mathbb{C}$ takes the output of the operator channel U and produces the closest codeword $V \in \mathbb{C}$ to U that is, for any other $V' \in \mathbb{C}$

$$d(U,V) \leqslant d(U,V')$$

Similar to traditional error correction, there is a necessary and sufficient condition on the total number of errors and erasures happening during the transmission through the operator channel which guarantees the minimum-distance decoder to be successful. This is provided in the following theorem.

Theorem 1.3.4. *[11] Consider a subspace code $\mathbb{C}$ with minimum distance d. Suppose that $V \in \mathbb{C}$ is transmitted through an operator channel and*

$$U \;=\; \mathcal{H}_k(V) \oplus E$$

is received. Let t and ρ be the number of errors and erasures, respectively. If

$$2(t+\rho) < d, \tag{1.7}$$

then a minimum distance decoder for $\mathbb{C}$ recovers the transmitted codeword V from the received subspace U.

Proof. Let $n = \dim(V)$. Then $\dim(U) = n - \rho + t$ and $\dim(U \cap V) = n - \rho$. The distance between U and V is given, by definition, as

$$d(U,V) = \dim(U) + \dim(V) - 2\dim(U \cap V) = \rho + t$$

For any other codeword $V' \neq V$, by the triangle inequality we have

$$d(V',U) \geqslant d(V',V) - d(U,V) \geqslant d - d(U,V) = d - (\rho + t) > \rho + t = d(U,V)$$

Therefore, the output of the minimum-distance decoder is V. ■

Next, the *rate* of a subspace code is defined

Definition 1.3.5. *[11] Let $\mathbb{C}$ be a code associated with the ambient space $\mathcal{W}$ of dimension N over $\mathbb{F}_q$. Suppose that the dimension of any $V \in \mathbb{C}$ is at most n. Then the rate R of $\mathbb{C}$ is defined as follows:*

$$R \stackrel{\text{def}}{=} \frac{\log_q |\mathbb{C}|}{nN} \tag{1.8}$$

If the dimension of all codewords in $\mathbb{C}$ is n, then the rate R of $\mathbb{C}$ can be thought of as the *symbol rate* of the code. In fact, $\log_q |\mathbb{C}|$ is the total number of information symbols. We are injecting n packets, each with length N, into the network. Hence, nN is the total number of symbols injected into the network.

At the end of this section, we present a very simple example of subspace codes.

Example 1.3.1. Let n and N be two positive integers with $n \leqslant N$ and let $\mathbb{F}_q$ be a finite field. Consider the set M of all $n \times N$ matrices over $\mathbb{F}_q$ of the form $[I_{n\times n}|A_{n\times(N-n)}]$, where A is an arbitrary $n \times (N-n)$ matrix and I is the $n \times n$ identity matrix. This set contains $q^{n(N-n)}$ different matrices. Observe that each matrix in M generates a different row space. Let the subspace code $\mathbb{C}$ be the set of all n-dimensional subspaces corresponding to the row spaces of elements of M. Then the size of $\mathbb{C}$ is equal to the size of M. The distance between any two elements of $\mathbb{C}$ is at least 2. Also, two matrices in M that differs only in one row generate row spaces with distance 2. Hence, the minimum distance of $\mathbb{C}$ is 2. The rate of the codes is equal to

$$R = \frac{\log_q |\mathbb{C}|}{nN} = \frac{n(N-n)}{nN} = 1 - \frac{n}{N}$$

1.4 Linearized Polynomials and Applications to Subspace Codes

Let $\mathbb{F}_q$ be a finite field and let $\mathbb{F} = \mathbb{F}_{q^m}$ be an extension field for some integer m. Recall from [13, Ch. 4.9] that a polynomial $f(X)$ over $\mathbb{F}$ is called an $\mathbb{F}_q$-linearized polynomial if it has the form

$$f(X) = \sum_{i=0}^{s} a_i X^{q^i}$$

where $a_i \in \mathbb{F}$, for $i = 0, 1, \ldots, s$. When q is fixed under discussion, we will let $X^{[i]}$ denotes X^{q^i}. We use the term q-degree instead of degree for linearized polynomials. For

instance, assuming that $a_s \neq 0$, the linearized polynomial $f(X)$ has q-degree s which means that its actual degree is q^s.

The main property of linearized polynomials, from which they receive their name, is the following. Let $f(X)$ be an $\mathbb{F}_q$-linearized polynomial over $\mathbb{F}$ and let $\mathbb{K}$ be an extension of $\mathbb{F}$. Then the map taking $\alpha \in \mathbb{K}$ to $f(\alpha) \in \mathbb{K}$ is linear with respect to $\mathbb{F}_q$ that is, for all $\alpha_1, \alpha_2 \in \mathbb{K}$ and all $\lambda_1, \lambda_2 \in \mathbb{F}_q$,

$$f(\lambda_1\alpha_1 + \lambda_2\alpha_2) = \lambda_1 f(\alpha_1) + \lambda_2 f(\alpha_2)$$

It is well-known that if two linearized polynomials of q-degree at most $k-1$ agree on at least k linearly independent points, then the two polynomials are identical. This is proved in the following lemma.

Lemma 1.4.1. *Let $f(X)$ and $g(X)$ be two $\mathbb{F}_q$-linearized polynomials with q-degree at most $k-1$, for some $k \in \mathbb{N}$. Suppose that there exist k linearly independent elements $\alpha_1, \alpha_2, \ldots, \alpha_k \in \mathbb{K}$, where $\mathbb{K}$ is an extension of $\mathbb{F}_q$, such that $f(\alpha_i) = g(\alpha_i)$, for $i = 1, 2, \ldots, k$. Then $f(X)$ and $g(X)$ must be identical.*

Proof. Let $h(X) = f(X) - g(X)$. Then $h(X)$ is also an $\mathbb{F}_q$-linearized polynomial with q-degree at most $k-1$. Furthermore, $h(X)$ has at least k linearly independent roots $\alpha_1, \alpha_2, \ldots, \alpha_k \in \mathbb{K}$. All $\mathbb{F}_q$-linear combinations of these elements are also roots of $h(X)$. Hence, $h(X)$ has at least q^k distinct roots while its degree is at most q^{k-1}. Therefore, $h(X)$ must be identically zero which implies that $f(X)$ and $g(X)$ are indeed equal. ■

As proved in the lemma, an $\mathbb{F}_q$-linearized polynomial $f(X)$ with q-degree s has at most s linearly independent roots. In fact, the set of all roots of $f(X)$ spans a vector space of dimension at most s, known as the the *root space* of $f(X)$. Let $\mathbb{K}$ be an extension field of $\mathbb{F}_q$ that contains all the roots of $f(X)$. The root space of $f(X)$ is indeed the kernel of $f(X)$ acting as a linear function on $\mathbb{K}$.

The sum of two linearized polynomials, $f_1(X)$ and $f_2(X)$, is also a linearized polynomial. However, the product $f_1(X)f_2(X)$ is not necessarily a linearized polynomial. Therefore, in order to have a ring structure, the operation $f_1(X) \otimes f_2(X)$ is

defined to be the composition $f_1(f_2(X))$ which is always a linearized polynomial. In fact, if $f_1(X) = \sum_{i \geqslant 0} a_i X^{[i]}$ and $f_2(X) = \sum_{j \geqslant 0} b_j X^{[j]}$, then

$$f_1(X) \otimes f_2(X) = f_1(f_2(X)) = \sum_{k \geqslant 0} c_k X^{[k]} \tag{1.9}$$

where $c_k = \sum_{i=0}^{k} a_i b_{k-i}^{[i]}$. It should be noted that this operation is not commutative. It is easy to construct examples for $f_1(X)$ and $f_2(X)$ such that

$$f_1(X) \otimes f_2(X) \neq f_2(X) \otimes f_1(X)$$

In summary, the set of $\mathbb{F}_q$-linearized polynomials over $\mathbb{F}_{q^m}$ forms a ring that is non-commutative under addition $+$ and composition $\otimes$. Let us denote this ring by $\mathscr{L}_{q^m}[X]$. Though not commutative, the ring of linearized polynomials has many of the properties of a Euclidean domain. In fact, there are two division algorithms: a left division and a right division; that is, given any two linearized polynomials $f_1(X)$ and $f_2(X)$, there exist unique linearized polynomials $q_L(X)$, $q_R(X)$, $r_L(X)$ and $r_R(X)$ such that

$$f_1(X) = q_L(X) \otimes f_2(X) + r_L(X) = f_2(X) \otimes q_R(X) + r_R(X)$$

where $r_L(X) = 0$ or $\deg(r_L(X)) < \deg(f_2(X))$ and similarly where $r_R(X) = 0$ or $\deg(r_R(X)) < \deg(f_2(X))$. A straightforward modification of polynomial division algorithm can be invoked in order to do left division and right division for linearized polynomials.

Koetter and Kschischang used linearized polynomials to construct a remarkable family of subspace codes analogous to Reed-Solomon codes in classical block codes [11]. Recall that Reed-Solomon codes are constructed by evaluating a certain message polynomial over a fixed set of points. They are one of the most famous families of block codes in use today. We briefly review Reed-Solomon codes with the aim of clarifying the analogy between them and Koetter-Kschischang subspace codes.

The construction of generalized Reed-Solomon codes is as follows. Let k be the number of information symbols and n be the length of the code. Let $\mathbb{F}_q$ be a finite field from which the message symbols are chosen. The message is a vector $\boldsymbol{u} = (u_0, u_1, \ldots, u_{k-1}) \in {\mathbb{F}_q}^k$. Then the corresponding codeword is constructed as $(f_{\boldsymbol{u}}(\alpha_1), f_{\boldsymbol{u}}(\alpha_2), \ldots, f_{\boldsymbol{u}}(\alpha_n))$, where $f_{\boldsymbol{u}}(X) = \sum_{i=0}^{k-1} u_i X^i$ is the message polynomial

and $\alpha_1, \alpha_2, \ldots, \alpha_n$ are n distinct and fixed elements of $\mathbb{F}_q$. Observe that generalized Reed-Solomon code is an (n, k) linear block code. The minimum distance of this code is equal to $n-k+1$, hence achieving the Singleton bound on the minimum distance of a block code. In fact, Reed-Solomon codes are in the family of MDS (maximum distance separable) codes.

Koetter-Kschischang algebraic subspace codes, originally called Reed-Solomon-like codes in [11], is analogous to Reed-Solomon codes in classical coding theory wherein symbols are replaced by vectors, regular polynomials with *linearized polynomials*, and sequences of symbols with $\mathbb{F}_q$-linear span of the corresponding vectors.

Next, we review the construction of Koetter-Kschischang subspace codes in more details. Let $\mathbb{F}_q$ be a finite field and $\mathbb{F} = \mathbb{F}_{q^m}$ be an extension of $\mathbb{F}_q$. $\mathbb{F}$ can be actually regarded as a vector space of dimension m over $\mathbb{F}_q$. Let $n \leqslant m$ be a positive integer and let $A = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ be a set of n elements of $\mathbb{F}$ that are linearly independent over $\mathbb{F}_q$. The set A will be the evaluation set and n will be the dimension of the constructed code.

Let $\boldsymbol{u} = (u_0, u_1, \ldots, u_{k-1}) \in \mathbb{F}^k$ be the vector of k information symbols over $\mathbb{F}$. Then we construct the corresponding linearized message polynomial $f_{\boldsymbol{u}}(X) \in \mathscr{L}_{q^m}[X]$ as follows:

$$f_{\boldsymbol{u}}(X) = u_0 X + u_1 X^q + \cdots + u_{k-1} X^{q^{k-1}}$$

We evaluate the message polynomial $f_{\boldsymbol{u}}(X)$ over the elements of A. For each i, $1 \leqslant i \leqslant n$, we append $f_{\boldsymbol{u}}(\alpha_i)$ to α_i to form the vector $\boldsymbol{v}_i = \big(\alpha_i, f_{\boldsymbol{u}}(\alpha_i)\big)$. This is necessary in this model as opposed to Reed-Solomon codes where our codewords are vectors whose entries have a certain order. In Reed-Solomon codes, there is no need to transmit the evaluation points themselves whereas in subspace codes, there is no ordering on the vectors spanning the codeword as a subspace. Therefore, we have to transmit α_i along with $f_{\boldsymbol{u}}(\alpha_i)$ for each evaluation point α_i. We define the ambient space $\mathcal{W}$ as

$$\mathcal{W} = \langle A \rangle \oplus \mathbb{F} = \{(\alpha, \beta) : \alpha \in \langle A \rangle, \beta \in \mathbb{F}\}$$

Observe that $\mathcal{W}$ is an $n+m$-dimensional vector space over $\mathbb{F}_q$. Finally the codeword V is the linear span of $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$ which is a subspace of the ambient space $\mathcal{W}$. Notice that α_i's are linearly independent which consequently make $\boldsymbol{v}_i$'s linearly independent.

In fact, V is an n-dimensional subspace of $\mathcal{W}$. Notice that any vector in V is of the form $\big(\alpha, f_{\boldsymbol{u}}(\alpha)\big)$ for some $\alpha \in \langle A \rangle$ by linearity of $f_{\boldsymbol{u}}(X)$. Let $\mathcal{E} : {\mathbb{F}_q}^k \to \mathcal{G}_q(\mathcal{W}, n)$ be the encoding map that takes elements of $\mathbb{F}^k$ as input and produces the corresponding n-dimensional subspace of $\mathcal{W}$ as explained above.

Lemma 1.4.2. *[11] Suppose that the size of the evaluation set A is at least k. Then the corresponding encoding map $\mathcal{E} : {\mathbb{F}_q}^k \to \mathcal{G}_q(\mathcal{W}, n)$ is injective.*

Proof. Suppose that the encoder $\mathcal{E}$ maps two massages in $\mathbb{F}^k$ with corresponding linearized polynomials $f(X)$ and $g(X)$ to the same subspace. It implies that $f(\alpha_i) = g(\alpha_i)$ for $i = 1, 2, \ldots, n$. Since $n \geqslant k$, $f(X)$ and $g(X)$ must be identical by Lemma 1.4.1. ■

We always assume $n \geqslant k$ in the construction of Koetter-Kschischang codes. Then by Lemma 1.4.2, the image of $\mathbb{F}^k$ is a code $\mathbb{C}$ with size q^{mk}. Therefore, the rate of $\mathbb{C}$ is given as

$$R = \frac{\log_q |\mathbb{C}|}{n \dim(\mathcal{W})} = \frac{mk}{n(n+m)} \tag{1.10}$$

We discuss the minimum distance of Koetter-Kschischang codes in the next theorem.

Theorem 1.4.3. *The minimum distance of the code $\mathbb{C}$ constructed by the encoding map $\mathcal{E} : {\mathbb{F}_q}^k \to \mathcal{G}_q(\mathcal{W}, n)$ is $2(n - k + 1)$.*

Proof. Let $f_1(X)$ and $f_2(X)$ be two distinct linearized polynomials in $\mathscr{L}_{q^m}[X]$ with q-degree at most $k-1$, corresponding to two different massages in $\mathbb{F}^k$. Let V_1 and V_2 be the corresponding codewords constructed by the encoder $\mathcal{E}$. Suppose that the dimension of $V_1 \cap V_2$ is l. Let

$$\{(\beta_1, \gamma_1), (\beta_2, \gamma_2), \ldots, (\beta_l, \gamma_l)\}$$

be a basis for $V_1 \cap V_2$. Then $\gamma_i = f_1(\beta_i) = f_2(\beta_i)$. This implies that $f_1(X)$ and $f_2(X)$ are equal on at least l linearly independent points. Therefore, by Lemma 1.4.1, $l \leqslant k-1$. Hence

$$d(V_1, V_2) = \dim(V_1) + \dim(V_2) - 2\dim(V_1 \cap V_2) = 2n - 2l \geqslant 2(n - k + 1)$$

■

1.5 List Decoding

In general the decoding problem can be formulated as follows. A code $\mathbb{C}$ as a subset of a metric space $\mathcal{M}$ is given and a codeword $c \in \mathbb{C}$, corresponding to the message, is being transmitted. An element of $r \in \mathcal{M}$ as a distorted version of c is received and the problem is how to recover the transmitted codeword c by observing its distorted version r.

There are many common decoding methods to recover encoded messages sent over a noisy channel. In the operator channel model, we take the *minimum distance decoding* approach. In general, a minimum distance decoder or *nearest neighbor decoder* takes the received element $r \in \mathcal{M}$ and finds the nearest codeword to r that is, the codeword $c \in \mathbb{C}$ such that $d(c, r) < d(c', r)$ for any other $c' \in \mathbb{C}$, where $d(\cdot, \cdot)$ is the distance defined on the metric space $\mathcal{M}$. It then outputs the message corresponding to the codeword c. Intuitively the minimum distance decoding is the relevant approach in the operator channel model as errors and erasures with more dimensions are less likely to happen. Therefore, loosely speaking, the closest codeword is most-likely the one that is transmitted.

Suppose that a code $\mathbb{C} \subseteq \mathcal{M}$ with minimum distance d_{min} is given that is,

$$d_{\text{min}} = \min_{\substack{X,Y \in \mathbb{C} \\ X \neq Y}} d(X, Y)$$

A codeword $c \in \mathbb{C}$ is transmitted and $r \in \mathcal{M}$ is received such that $d(c, r)$ is at most $(d_{\text{min}} - 1)/2$. Then the minimum distance decoder is guaranteed to successfully recover the transmitted codeword because by triangle inequality

$$d(c', r) \geqslant d(c', c) - d(r, c) \geqslant d_{\text{min}} - d(r, c) \geqslant d_{\text{min}} - \frac{d_{\text{min}} - 1}{2} > \frac{d_{\text{min}} - 1}{2} \geqslant d(c, r)$$

for any other codeword $c' \in \mathbb{C}$. The bound $(d_{\text{min}} - 1)/2$ on the distance between the transmitted codeword and received word is sometimes called the *diameter bound*. In fact spheres of radius $(d-1)/2$ around codewords of $\mathbb{C}$ do not intersect with each other. This is depicted in Figure 1.2. Therefore, the received word is contained in the sphere around exactly one codeword which is indeed the transmitted codeword c.

The minimum distance decoding explained above is a type of *unique decoding*. Generally speaking the unique decoding algorithm decodes only up to a certain bound,

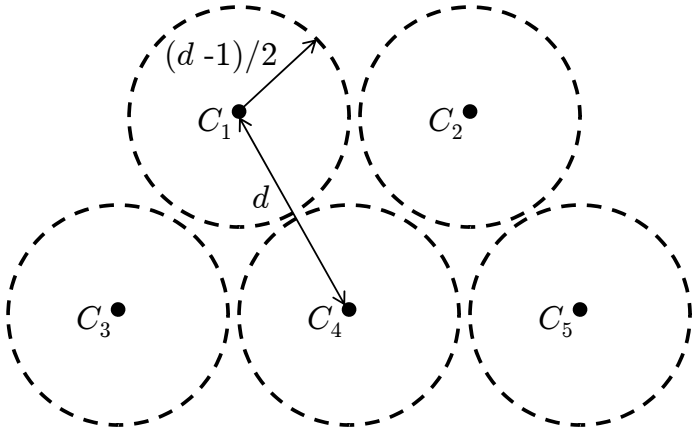


Figure 1.2: Spheres of radius $(d-1)/2$ around the codewords of a code with minimum distance d are all disjoint.

on the number of errors or in general on the distance between the transmitted codeword and received word, which it is guaranteed to find at most one codeword within such a distance of the received word. If no codeword is found within this distance, then the decoder declares decoding failure.

The obvious unique decoding algorithm is to search the whole sphere around the received word in order to find the transmitted codeword. However, this is not efficient and in general requires exponential runtime. Designing efficient decoding algorithms for various families of codes has been in the center of research in coding theory in the past decades since the whole field of coding theory was born. Such results are discussed in detail in any standard coding theory text (e.g. [13]). In the previous section, we discussed Koetter-Kschischang subspace codes. Koetter and Kschischang also provided an efficient decoding algorithm for their proposed codes, in the context of minimum distance decoding, which we will review in the next chapter.

Suppose that we are interested to correct errors beyond the half of d_{min} bound. In this case, the minimum distance decoder may fail to output the transmitted codeword. It either outputs a wrong codeword or a decoding failure is declared. The former case happens if the received word falls within distance $(d_{\text{min}}-1)/2$ of some other codeword. The later case happens if no codeword is found within distance $(d_{\text{min}}-1)/2$ of the

received word. Typically the minimum distance decoder is designed in such a way that it declares decoding failure in such a case. It turns out that there is a meaningful relaxation of unique decoding that allows us to decode beyond half of $d_{\min}$ bound faced by unique decoding. This relaxed notion of decoding, called list decoding, will certainly help to improve traditional bounds on the performance of error-correcting codes.

List decoding was introduced in two independent works by Elias [4] and Wozencraft [20] in the late 50s. List decoding is basically a relaxation of unique decoding that allows a *list* of codewords as the output of the decoder. This offers a potential for recovery from errors beyond the traditional error-correction bound. In this terminology, the decoder is said to be successful if the actual transmitted message is included in the output list. The list decoding problem is the problem of finding all the codewords within a certain distance τ of the received word. The case $\tau = (d_{\min} - 1)/2$ reduces to the unique decoding problem. In fact list decoding is always possible for any τ even when τ is much larger than $(d_{\min} - 1)/2$.

The most important parameter associated with list decoding is the list size that is allowed at the output of the decoder. If we require the list size equal to one, then the list decoding problem reduces to the unique decoding. Also, we want to avoid very large list sizes. For instance, a trivial list decoding algorithm is to just output all the codewords of the code as the output list. This is not certainly desirable. There are two main issues that make large list sizes undesirable. First, we want to make sure that the output list is useful for the decoder to make the final decision about the transmitted message. This may be done using an additional processing on the candidates to pick the best one e.g. the decoder may choose the codeword in the list that is the closest one to the received word. If the list size is exponentially large, this can not be efficiently done in terms of the time complexity. Second, the list decoding algorithm itself should be done in an efficient polynomial time. Notice that the worst-case complexity of the list decoding algorithm is at least as large as the maximum allowed output list size.

1.6 Dissertation Overview

Subspace codes and rank metric codes are two closely related family of codes used for reliable communication of messages in randomized network coding. In Section 1.3, we reviewed the definition of subspace codes that were introduced for error correction in non-coherent randomized network coding, modeled by operator channel. In Chapter 5, we will review the basics of rank-metric codes, introduced for dealing with errors and erasures in *coherent randomized network coding*, with the aim of establishing our list-decoding results in that setting.

The Koetter-Kschischang codes is our starting point in Chapter 2. We modify and generalize these codes in many important respects, thereby producing a family of subspace codes that are efficiently list-decodable. List decoding, in turn, makes it possible to provide a better tradeoff between rate and error-correction capability than Koetter-Kschischang codes, albeit only for low rates. In a sense, we achieve for the Koetter-Kschischang codes a result that is somewhat analogous to Sudan's results for Reed-Solomon codes in [19]. We also extend the Roth-Ruckenstein bivariate factorization algorithm [16] to the domain of linearized polynomials. This result, being also of independent interest, shows that we can perform the factorization step of our list-decoding algorithm, which is essentially solving equations over the ring of linearized polynomials, in an efficient polynomial time.

Given the results of Chapter 2, extending our codes and the list-decoding algorithms to higher rates is set as the next target. Our work in Chapter 3 is motivated by the Guruswami-Sudan list-decoding algorithm of Reed-Solomon codes [8]. Sudan list-decoding algorithm enables list decoding of Reed-Solomon codes beyond the traditional error-correction bound only for a limited range of rates. Then the Guruswami-Sudan idea is to enforce the interpolation polynomial to go through the same set of interpolation points as in Sudan algorithm but with some multiplicity greater than one. At the end, they are able to improve upon the half of minimum distance bound for all rates. In Chapter 3, we consider the problem of list-decoding of subspace codes with multiplicities. To the best of our knowledge, however, no explicit definition of multiplicity for linearized polynomials exists in the literature. Hence, we first establish the notion of multiplicities for linearized polynomials in this context. We take advantage of enforc-

ing multiple roots for the interpolation polynomial in order to enable list-decoding for a wider range of rates. We also achieve a better trade-off between the rate and decoding radius than both the Koetter-Kschischang codes and the results presented in Chapter 2.

In Chapter 4, we introduce a new family of subspace codes based upon a different approach which leads to a linear-algebraic list-decoding algorithm. This new construction can be thought as a folded version of the Koetter-Kschischang codes in which we append evaluations of the message polynomial on certain s elements of the field to each other, where s can be called the folding parameter. This enables a linear-algebraic type of list-decoding in which the output list itself forms a vector space. Comparing to the results of Chapter 3 and Chapter 4 in which we fix a constant L for the list size independent of the field size, the list size with this approach is rather large, yet polynomial in size of the underlying field. On the other hand, there is a significant improvement in the error correction capability of the proposed construction versus rate upon the previously presented results.

In Chapter 5, we turn our attention to rank-metric codes, a well-studied class of codes introduced in independent works by Delsarte [3], Gabidulin [6], and Roth [15]. In a rank-metric code, each codeword is a matrix with fixed dimensions whose entries are taken from a certain finite field. The distance between two matrices is simply the rank of their difference. It turns out that rank-metric codes are a suitable tool to deal with injected errors into the the network in *coherent* linear network coding when *pessimistic* adversarial errors are assumed. In this setting, as opposed to non-coherent case, the network topology and the particular network coding operations done at intermediate nodes are known to both the transmitter and the receiver. Silva et al. show that rank-metric and subspace codes are closely related [18]. Indeed, there is an injective mapping between rank-metric codes and subspace codes through a *lifting* operation. Gabidulin introduced a class of MRD (maximum rank distance) rank-metric codes using linearized polynomials [6]. Also, the Singleton bound is established in the context of rank-metric codes by Gabidulin. In Chapter 5, we define a folded version of Gabidulin codes. Then we propose a list-decoding algorithm that can correct the fraction of errors up to the Singleton bound which is the information theoretic upper bound on the error correction capability of a code.

Bibliography

[1] R. Ahlswede, N. Cai, Sh.Y.R. Li, and R.W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, pp. 1204–1216, July 2000.

[2] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. 41-st Annual Allerton Conference on Communication, Control and Computing,* Monticello, IL, October 2003.

[3] P. Delsarte, "Bilinear forms over a finite field, with applications to coding theory," *J. Comb. Theory. Ser. A*, vol. 25, pp. 226-241, 1978.

[4] P. Elias, "List decoding for noisy channels," Technical Report 335, Research Laboratory of Electronics, MIT, 1957.

[5] P. Elias, "Error-correcting codes for list-decoding," *IEEE Transactions on Information Theory*, vol. 37, pp. 5–12, January 1991.

[6] E. M. Gabidulin, "Theory of codes with maximum rank distance," *Probl. Inf. Trasnm.*, vol. 21, no. 1, pp. 1-12, 1985.

[7] T. Etzion and A. Vardy, "Error-correcting codes in projective space," *IEEE Transactions on Information Theory*, vol. 57, pp. 1165–1173, February 2011.

[8] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometric codes," *IEEE Transactions on Information Theory*, vol. 45, pp. 1757–1767, September 1999.

[9] T. Ho, M. Médard, J. Shi, M. Effros, and D. Karger, "On randomized network coding," in *Proc. 41-st Annual Allerton Conference on Communications, Control, and Computing*, Monticello, IL., October 2003.

[10] T. Ho, M. Medard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, pp. 4413–4430, October 2006.

[11] R. Koetter and F.R. Kschischang, "Coding for errors and erasures in random network coding," *IEEE Transactions on Information Theory*, vol. 54, pp. 3579–3591, August 2008.

[12] Sh.Y.R. Li, R.W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol 49, pp. 371–381, February 2003.

[13] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam: North Holland, 1977.

[14] H. Mahdavifar and A. Vardy "Algebraic list-decoding on the operator channel," in *Proc. IEEE International Symposium on Information Theory*, pp. 1193–1197, Austin, TX., June 2010.

[15] R. M. Roth, "Maximum-rank array codes and their application to criss-cross error correction," *IEEE Transactions on Information Theory*, vol. 37, no. 2, pp. 328-336, Mar. 1991.

[16] R.M. Roth and G. Ruckenstein, "Efficient decoding of Reed-Solomon codes beyond half the minimum distance," *IEEE Transactions on Information Theory*, vol. 46, pp. 246–257, January 2000.

[17] C.E. Shannon, "A mathematical theory of communication," *Bell Systems Tech. J.*, vol. 27, pp. 379–423 and pp. 623–656, 1948.

[18] D. Silva, F. R. Kschischang, R. Koetter, "A rank-metric approach to error control in random network coding," *IEEE Transactions on Information Theory*, vol. 54, no. 9, pp. 3951-3967, Sep. 2008.

[19] M. Sudan, "Decoding of Reed-Solomon codes beyond the error-correction bound," *J. Complexity,* vol. 12, pp. 180–193, March 1997.

[20] John M. Wozencraft, "List decoding," Quarterly Progress Report, Research Laboratory of Electronics, MIT 48 (1958), 90-95.

[21] H. Xie, Z. Yan, and B.W. Suter, "General linearized polynomial interpolation and its applications," in *Proc. International Symposium on Network Coding,* Beijing, China, July 2011.

Chapter 2

Constructions and List-Decoding Algorithms

2.1 Introduction

Randomized linear network coding, first proposed in [4,5], is a powerful tool for disseminating information in networks. Randomized network coding is considered as the most promising *practical* approach to network coding to-date [2], since it is completely distributed and decentralized. Yet it is highly susceptible to transmission errors caused by noise or intentional jamming. Even a single packet error injected into the network could potentially render the entire transmission useless. Packets can also become lost (erased), so that the problem of deducing the transmitted message at the receiver(s) cannot be completed.

In multicast linear network coding [1,7], a set of packets $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$ is injected into the network by the source node. These packets are all of the same length and can be regarded as vectors of length m over a finite field $\mathbb{F}_q$. If the network coding protocol is *randomized*, each intermediate node in the network generates random $\mathbb{F}_q$-linear combinations of the packets available at its incoming edges and sends them out on its outgoing edges. Finally, receiver nodes collect the packets on their incoming edges and use this information in attempt to recover $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$. It is proved in [5] that, with high probability, this randomized network coding protocol achieves the min-

cut max-flow bound (cf. [1,7]), *simultaneously for each receiver*, provided the size q of the underlying field $\mathbb{F}_q$ is sufficiently large.

Errors in this model correspond to injection into the network of extraneous packets that do not belong to the linear space spanned by the source vectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$; erasures (lost packets) correspond to the projection of $\langle \boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n \rangle$ onto a lower-dimensional subspace. The problem of error-control for randomized network coding was first addressed in the pioneering work of Koetter and Kschischang [6]. Motivated by the fact that randomized network coding is vector-space preserving, Koetter and Kschischang [6] introduced error-correcting codes in projective space [3,6], also known as *subspace codes*.

Let $\mathcal{W}$ be a fixed ambient vector space over $\mathbb{F}_q$. The projective space of $\mathcal{W}$, denoted here as $\mathscr{P}_q(\mathcal{W})$, is the set of all subspaces of $\mathcal{W}$. A subspace code $\mathbb{C}$ in $\mathcal{W}$ is any nonempty subset of $\mathscr{P}_q(\mathcal{W})$. Koetter and Kschischang [6] showed that a subspace code $\mathbb{C} \subseteq \mathscr{P}_q(\mathcal{W})$ with minimum distance d (for an appropriately defined distance function) can correct t packet errors and ρ packet erasures introduced anywhere in the network as long as $2t + 2\rho < d$. Koetter and Kschischang also constructed in [6] a remarkable family of subspace codes that are similar to Reed-Solomon codes in that codewords are obtained by evaluating certain polynomials in a set of points. In the case of Koetter-Kschischang codes, however, ordinary polynomials over $\mathbb{F}_q$ are replaced by *linearized polynomials*. Koetter and Kschischang [6] furthermore devised a "list-1" decoding algorithm for their codes, based upon bivariate interpolation in the domain of linearized polynomials. The Koetter-Kschischang algorithm is analogous to the Berlekamp-Welch decoding algorithm for Reed-Solomon codes; it achieves the error-correction radius of $1 - R$, where R is the (symbol) rate of the corresponding subspace code.

The Koetter-Kschischang codes serve as our starting point in this chapter. We modify and generalize these codes in many important respects, thereby producing a family of subspace codes that are efficiently list-decodable. List decoding, in turn, makes it possible to provide a better tradeoff between rate and error-correction capability than Koetter-Kschischang codes, albeit only for low rates. Extending our codes and list-decoding algorithms to higher rates remains an open problem. Nevertheless, in a sense, we have achieved for the Koetter-Kschischang codes a result that is somewhat analogous

to Sudan's results for Reed-Solomon codes in [10]. In order to do so, we had to overcome several obstacles. First, the ring $\mathscr{L}_{q^m}[X]$ of linearized polynomials over $\mathrm{GF}(q^m)$ is not commutative and, therefore, a polynomial over this ring may have more roots than its degree. Consequently, the natural approach to making Koetter-Kschischang codes list-decodable fails — it may lead to lists of exponential size. We overcome this problem by using the *subring* $\mathscr{L}_q[X]$ of $\mathscr{L}_{q^m}[X]$, which turns out to be commutative. Restricting the input symbols to $\mathrm{GF}(q)$ rather than $\mathrm{GF}(q^m)$, however, drastically reduces the rate of the code. In order to overcome this second problem, we make use of certain *normal bases* for $\mathrm{GF}(q^m)$ over $\mathrm{GF}(q)$. However, this restricts the dimension of (all codewords in) our codes to one, since the entire space $\mathrm{GF}(q^m)$ serves as the set of potential roots of the interpolation polynomial, already in the one-dimensional case. Hence, in order to produce list-decodable codes of arbitrary codeword dimension, we further modify our construction. This modification extends the space of potential roots of the interpolation polynomial from $\mathrm{GF}(q^m)$ to $\mathrm{GF}(q^{nm})$, and makes it possible to list-decode subspace codes of arbitrary dimension n.

The rest of this chapter is organized as follows. In Section 2.2, we briefly review some previous work on subspace codes, with the aim of establishing the background necessary for our results. In Section 2.3, we introduce three key modifications of Koetter-Kschischang codes, thereby laying the foundations for list decoding. Then the list-decoding algorithm itself is presented and the correctness of this algorithm is proved. We point out that the introduced list-decodable codes are one-dimensional, meaning that the source injects a single packet $\boldsymbol{v}_1$ into the network. Unfortunately, a straightforward extension of the results of Section 2.3 to dimensions greater than one does not work since the entire space $\mathrm{GF}(q^m)$ serves as the set of potential roots of the interpolation polynomial constructed in our list-decoding algorithm. Consequently, increasing the dimension of the codewords does not yield any new information at the receiver(s). This problem is addressed in Section 2.4, where we show how to extend the space of potential roots of the interpolation polynomial to $\mathrm{GF}(q^{nm})$, thereby constructing list-decodable subspace codes of arbitrary codeword dimension n. The corresponding list-decoding algorithm and a proof of its correctness are also presented in Section 2.4. In Section 2.5, we present a solution for list-decoding of the original

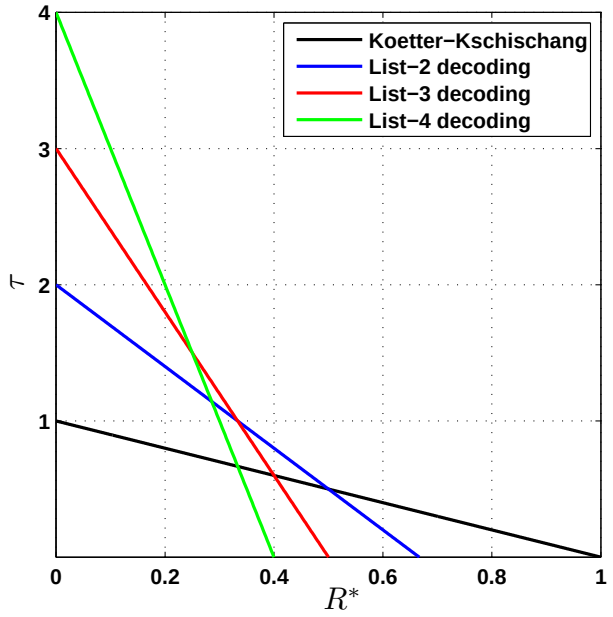


Figure 2.1: The error-correction radius τ versus the packet rate R^* for the Koetter-Kschischang codes and for our list-decoding algorithm with various list sizes

Koetter-Kschischang codes, though only in the one-dimensional case. The general case remains an open problem. In Section 2.6, we extend the Roth-Ruckenstein bivariate factorization algorithm [9] to the domain of linearized polynomials, a result that may be of independent interest. Finally, we conclude the chapter in Section 2.7 with a brief discussion.

The end result of all this effort is most conveniently expressed in terms of the error-correction radius τ and the packet rate R^*. Specifically, we guarantee that the message injected by the source will be recovered at the receivers as long as

$$\tau \;\leqslant\; L \;-\; \frac{L(L+1)}{2}\, R^* \tag{2.1}$$

where L is the list size. Here, as in Koetter and Kschischang [6], the *error-correction radius* is defined as $\tau = t/n$, where t is the dimension of the error and n is the codeword dimension. The *packet rate* R^* is a new parameter introduced in this paper.

Loosely speaking, R^* is the ratio of the number of information packets to the number of encoded packets injected into the network. This is different from the notion

of rate R defined by Koetter and Kschischang [6], which may be thought of as the ratio of the number of information symbols to the number of encoded symbols. For more details on the packet rate R^* and its relationship to the (symbol) rate R, see Section 2.4.

Figure 2.1 depicts the bound (2.1) on the error-correction radius for the first few values of the list size L. For $L = 1$, our results coincide with those of [6], as expected. For higher values of L, we improve upon [6], but only for low rates.

2.2 Prior Work

In this section, we review some of the prior work on subspace codes and corresponding decoding algorithms. In Chapter 2, we reviewed the operator channel model, the ring of linearized polynomials, and the subspace codes. In this section, we briefly recap the necessary background and then we discuss Koetter and Kschischang [6] subspace codes and their list-1 decoding algorithm.

2.2.1 Background

The operator channel model was introduced by Koetter and Kschischang [6] in order to capture the essence of randomized linear network coding for multicast, in the non-coherent case, where network topology and the underlying network coding operations are unknown. Let $\mathcal{W}$ be a fixed vector space over $\mathbb{F}_q$, called the *ambient space*. Let $N = \dim \mathcal{W}$. All the packets in the network are viewed as elements of $\mathcal{W}$. As before, let $\mathscr{P}_q(\mathcal{W})$ denote the set of all subspaces of $\mathcal{W}$. Further, let $\mathcal{G}_q(\mathcal{W}, n)$ denote the set of all subspaces of $\mathcal{W}$ of dimension n. A distance function on $\mathscr{P}_q(\mathcal{W})$ is defined as follows:

$$d(A, B) \stackrel{\text{def}}{=} \dim(A + B) - \dim(A \cap B) \tag{2.2}$$

The input to the operator channel is a subspace $V \in \mathscr{P}_q(\mathcal{W})$. The output of the operator channel is another subspace $U \in \mathscr{P}_q(\mathcal{W})$ with possibly deletion of vectors from the input V, called erasures, or addition of vectors that are linearly independent from the input V, called errors. More precisely,

$$U = \mathcal{H}_k(V) \oplus E \tag{2.3}$$

where E is an *error vector space* such that $E \cap V = \{\mathbf{0}\}$ and $\mathcal{H}_k$ is the *erasure operator*. The erasure operator $\mathcal{H}_k$ projects V onto a k-dimensional subspace of V chosen uniformly at random, provided $\dim V > k$; otherwise, $\mathcal{H}_k$ leaves V unchanged. The number of errors and erasures that occurred during the transmission over this operator channel are defined as $t = \dim E$ and $\rho = \dim V - \dim \mathcal{H}_k(V)$, respectively.

A *subspace code* $\mathbb{C}$ for an operator channel with ambient space $\mathcal{W}$ is a non-empty subset of $\mathscr{P}_q(\mathcal{W})$. Thus codewords of $\mathbb{C}$ are subspaces of $\mathcal{W}$. The minimum distance of $\mathbb{C}$ is given by

$$d(\mathbb{C}) \stackrel{\text{def}}{=} \min_{\substack{X,Y \in \mathbb{C} \\ X \neq Y}} d(X,Y)$$

Koetter and Kschischang proved in [6] that a minimum distance decoder for $\mathbb{C}$ will always recover the transmitted subspace V from the received subspace U in (2.3), provided

$$2(t+\rho) \; < \; d(\mathbb{C}) \tag{2.4}$$

Conversely, if (2.4) is not satisfied, then the minimum distance decoder for $\mathbb{C}$ may fail. Let N denote the dimension of ambient space $\mathcal{W}$ and suppose that the dimension of any $V \in \mathbb{C}$ is at most n. Then the *rate* R of the code is defined as follows:

$$R \stackrel{\text{def}}{=} \frac{\log_q |\mathbb{C}|}{nN} \tag{2.5}$$

The main object in the construction of Koetter-Kschischang subspace codes is the ring of linearized polynomials. Next, we briefly review linearized polynomials, which we discussed in detail in Section 1.4. Let $\mathbb{F}_q$ be a finite field and let $\mathbb{F} = \mathbb{F}_{q^m}$ be an extension field. Then a polynomial $f(X)$ over $\mathbb{F}$ is called an $\mathbb{F}_q$-linearized polynomial if it has the form

$$f(X) = \sum_{i=0}^{s} a_i X^{q^i}$$

where $a_i \in \mathbb{F}$, for $i = 0, 1, \ldots, s$. When q is fixed under discussion, we use $X^{[i]}$ to denote X^{q^i}. We also use the term q-degree instead of degree for linearized polynomials. For instance, assuming that $a_s \neq 0$, the linearized polynomial $f(X)$ has q-degree s which means that its actual degree is q^s. The main property of linearized polynomials, from which they receive their name, is that they act as linear maps over any extension field of $\mathbb{F}$ with respect to the base field $\mathbb{F}_q$. An interesting property, analogous to the

fundamental theorem of algebra for regular polynomials, is that a non-zero linearized polynomial of q-degree k has at most k linearly independent roots.

The sum of two linearized polynomials, $f_1(X)$ and $f_2(X)$, is also a linearized polynomial. In order to have a ring structure the operation $f_1(X) \otimes f_2(X)$ is defined to be the composition $f_1(f_2(X))$ which is always a linearized polynomial. The set of linearized polynomials over $\mathbb{F}_{q^m}$ forms a non-commutative ring with identity under addition $+$ and composition $\otimes$. Let us denote this ring by $\mathscr{L}_{q^m}[X]$. Although $\mathscr{L}_{q^m}[X]$ is not commutative, it has many of the properties of a Euclidean domain. For instance, instead of a devision algorithm there are two division algorithms: a left division and a right division algorithm.

2.2.2 Koetter-Kschischang Codes

In Section 1.4, we reviewed Koetter-Kschischang codes as an application of linearized polynomials in the construction of subspace codes. Here, we briefly recap the construction and then present the list-1 decoding algorithm proposed by Koetter and Kschischang in [6].

As before, let $\mathbb{F}_q$ be a finite field, and let $\mathbb{F} = \mathbb{F}_{q^m}$ be an extension field of $\mathbb{F}_q$. The number of information symbols k and the dimension of code n are also fixed. Notice that $\mathbb{F}$ can be regarded as a vector space of dimension m over $\mathbb{F}_q$. Let $A = \{\alpha_1, \ldots, \alpha_n\}$ be a set of n linearly independent vectors in this vector space.

Koetter-Kschischang Encoding:

The input to the encoder is a vector $\boldsymbol{u} = (u_0, \ldots, u_{k-1})$ which consists of k message symbols in $\mathbb{F}$. The corresponding message polynomial is $f_{\boldsymbol{u}}(X) = \sum_{i=0}^{k-1} u_i X^{[i]}$. Then the corresponding codeword V is the $\mathbb{F}_q$-linear span of the set $\{(\alpha_i, f(\alpha_i)) : 1 \leqslant i \leqslant n\}$.

The code $\mathbb{C}$ is the set of all possible codewords V. The ambient space $\mathcal{W}$ is $\langle A \rangle \oplus \mathbb{F} = \{(\alpha, \beta) : \alpha \in \langle A \rangle, \beta \in \mathbb{F}\}$ which has dimension $n+m$ over $\mathbb{F}_q$. We represent each element of $\mathcal{W}$ as a vector with two coordinates such as (x, y), where $x \in \langle A \rangle$ and $y \in \mathbb{F}$.

Suppose that V is transmitted over the operator channel and a subspace U of $\mathcal{W}$ of dimension d is received.

Koetter-Kschischang Decoding:

Let $(x_i, y_i), i = 1, 2, \ldots, d$ be a basis for U. Construct a non-zero bivariate polynomial $Q(X, Y)$ of the form

$$Q(X, Y) = Q_0(X) + Q_1(Y),$$

where Q_0 and Q_1 are linearized polynomials over $\mathbb{F}$, Q_0 has q-degree at most $\omega - 1$ and Q_1 has q-degree at most $\omega - k$ such that

$$Q(x_i, y_i) = 0 \text{ for } i = 1, 2, \ldots, d \tag{2.6}$$

The parameter ω will be specified later. Then solve the equation $Q(X, f(X)) = 0$ for $f(X)$ to recover the message polynomial.

Notice that $Q(X, Y)$ is constructed to interpolate only a basis for U. Since $Q(X, Y)$ is linearized, this will guarantee that $Q(x, y) = 0$, for any $(x, y) \in \mathcal{W}$.

Observe that (2.6) is indeed a homogeneous system of linear equations with d equations. The total number of potential coefficients of Q_0 and Q_1 is $2\omega - k + 1$. This is in fact the number of variables in this system of equations. Therefore, (2.6) is guaranteed to have a non-zero solution if

$$d < 2\omega - k + 1 \tag{2.7}$$

Suppose that $d = n - \rho + t$, where ρ is the number of erasures and t is the number of errors. Then the dimension of $U \cap V$ is $n - \rho$. Notice that for any $(x, y) \in U \cap V$, $y = f(x)$. Therefore, the equation

$$Q\big(X, f_{\boldsymbol{u}}(X)\big) = 0$$

has at least $n - \rho$ linearly independent roots i.e. basis elements of $U \cap V$. Notice that the q-degree of $Q\big(X, f_{\boldsymbol{u}}(X)\big)$ is at most $\omega - 1$. If the condition

$$n - \rho \geqslant \omega \tag{2.8}$$

is satisfied, then $Q\big(X, f_{\boldsymbol{u}}(X)\big)$ has more linearly independent roots than its q-degree. Therefore, it will be identically zero i.e.

$$Q_0(X) + Q_1(X) \otimes f_{\boldsymbol{u}}(X) = 0$$

and $f_{\boldsymbol{u}}(X)$ can be uniquely recovered by performing right division algorithm on $-Q_0(X)$ and $Q_1(X)$. By combining (2.7) and (2.8), observe that the necessary condition for successful recovery of message is

$$\rho + t < n - k + 1 \tag{2.9}$$

Conversely, suppose that (2.9) is satisfied. Then we can select

$$\omega = \left\lceil \frac{d+k}{2} \right\rceil$$

which in turn guarantees both (2.7) and (2.8). Therefore, (2.9) is the necessary and sufficient condition for successful unique decoding. The minimum distance of Koetter-Kschischang codes is $2(n-k+1)$ as shown in Theorem 1.4.3. Hence, the proposed decoding algorithm indeed achieves half of minimum distance bound for unique decoding of Koetter-Kschischang codes.

The rate of the code is

$$R = \frac{km}{n(n+m)}$$

The result can be expressed in terms of the error-correction radius τ and the rate R. The transmitted message is successfully recovered as long as

$$\tau \leqslant 1 - (1 + \frac{n}{m})R \tag{2.10}$$

where τ is the total dimension of errors and erasures normalized by the dimension n. In the regime where n is much smaller than m, the bound on the error-correction capability of Koetter-Kschischang codes with unique decoding can be approximated as $1 - R$.

2.3 List-decoding of Subspace Codes

We start this section with a brief review of Sudan's list-decoding algorithm of Reed-Solomon codes in Section 2.3.1. Then we justify why it is necessary to modify Koetter-Kschischang codes in order to enable list-decoding. A simple generalization of Koetter-Kschischang codes is proposed in Section 2.3.2. We shall see that a small list size can not be guaranteed as a result of the ring of linearized polynomials being non-commutative. Therefore, we further modify the construction to solve this problem in Section 2.3.3. However, this modification results in a rate reduction by a factor of m. To compensate for this reduction, we exploit the properties of a normal basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ in Section 2.3.4. Having set all that, we explain the encoding and list-decoding of this new construction of subspace codes in Section 2.3.5. We establish the correctness of the proposed algorithm in 2.3.6. Finally the parameters of the code are discussed in Section 2.3.7.

2.3.1 Overview of Sudan's List-Decoding Algorithm

In this subsection, we briefly review Sudan's list-decoding algorithm of Reed-Solomon codes [10] which motivated our approach. Generalized Reed-Solomon codes are constructed as follows. Let $\mathbb{F}_q$ be a finite field. The parameters k, the number of information symbols, and n, the length of the code, are fixed and $k \leqslant n \leqslant q-1$. The message is a vector $\boldsymbol{u} = (u_0, u_1, \ldots, u_{k-1})$ consisting of k information symbols over $\mathbb{F}_q$. The corresponding codeword is $(f_{\boldsymbol{u}}(\alpha_1), f_{\boldsymbol{u}}(\alpha_2), \ldots, f_{\boldsymbol{u}}(\alpha_n))$, where $f_{\boldsymbol{u}}(X) = \sum_{i=0}^{k-1} u_i X^i$ is the message polynomial and $\alpha_1, \alpha_2, \ldots, \alpha_n$ are n distinct and fixed elements of $\mathbb{F}_q$. This codeword is transmitted through the channel. Given the channel output $(y_1, y_2, \ldots, y_n)$, Sudan's list-L decoding algorithm constructs the bivariate interpolation polynomial

$$Q(X, Y) = Q_0(X) + Q_1(X)Y + \cdots + Q_L(X)Y^L$$

such that $Q(\alpha_i, y_i) = 0$ for all i, subject to certain degree constraints. This first step is also called *interpolation step*. Then if not too many errors have occurred, $Q\big(X, f_{\boldsymbol{u}}(X)\big)$ is identically zero, and the message can be recovered by finding all the factors (at most L of them) of $Q(X, Y)$ of the form $Y - F(X)$. This step is called the *factorization step*.

The degree constraint on $Q(X, Y)$ in the interpolation step is set in such a way that the degree of $Q(X, f_{\boldsymbol{u}}(X))$ is at most $\omega - 1$ for a certain ω that will be specified later. In order to formalize this statement, the weighted degree of bivariate polynomials is defined as follows. For any pair of integers a and b, the (a, b)-weighted degree of a monomial $q_{i,j}X^iY^j$ is defined to be $ai + bj$. The (a, b)-weighted degree of $Q(X, Y)$ is defined to be maximum (a, b)-weighted degree among all its monomials with non-zero coefficients. This definition of weighted degree can be also easily generalized to multivariate polynomials. Given this definition, we simply say that the $\big(1, (k-1)\big)$-weighted degree of the interpolation polynomial $Q(X, Y)$ is at most $\omega - 1$.

The interpolation step is equivalent to solving a homogeneous system of linear equations. All we need in order to guarantee a non-trivial solution for $Q(X, Y)$ is that the number of variables is larger than the number of equations. The number of equations is n. The number of variables can be easily computed in terms of ω and L as $\omega - \binom{L+1}{2}(k-1)$. Therefore, we need to have

$$(L+1)\omega - \binom{L+1}{2}(k-1) > n \tag{2.11}$$

$Q(X, f_{\boldsymbol{u}}(X))$ is a univariate polynomial in $\mathbb{F}_q[X]$ with degree at most $\omega - 1$. Any correct received symbol $y_i = f_{\boldsymbol{u}}(\alpha_i)$ guarantees one particular root α_i for $Q(X, f_{\boldsymbol{u}}(X))$. Hence, $Q(X, f_{\boldsymbol{u}}(X))$ is guaranteed to have at least $n - t$ roots, where t is the maximum number of errors we wish to correct. If the condition

$$n - t \geqslant \omega \tag{2.12}$$

holds, then $Q(X, f_{\boldsymbol{u}}(X))$ is identically zero and $f_{\boldsymbol{u}}(X)$ can be successfully recovered by finding all the possible factors $Y - f(X)$ of $Q(X, Y)$. An efficient polynomial-time factorization algorithm is proposed by Roth and Ruckenstein in [9].

We can combine (2.11) and (2.12) in order to get a bound on the error-correction capability of Sudan's list-L decoder. Substituting ω from (2.12) into (2.11) leads to the condition

$$\frac{t}{n} < \frac{L}{L+1} - \frac{L}{2}\Big(\frac{k-1}{n}\Big)$$

t/n is the normalized error-correction radius τ and $(k-1)/n$ is approximately the rate of the code $R = k/n$. The final condition can be expressed as

$$\tau < \frac{L}{L+1} - \frac{L}{2}R$$

On the other hand, if this is satisfied one can find a suitable value for ω in order to successfully perform the Sudan list-L decoding algorithm.

2.3.2 First Generalization of Koetter-Kschischang Codes

Let $\mathbb{F}_q$ be a finite field and $\mathbb{F} = \mathbb{F}_{q^m}$ be an extension field of $\mathbb{F}_q$. For ease of notation, let $f^{\otimes L}(X)$ denote the composition of $f(X)$ with itself L times for any linearized polynomial $f(X)$. Indeed $f^{\otimes 1}(X) = f(X)$. Also, we define $f^{\otimes 0}(X)$ to be equal to X. $A = \{\alpha_1, \ldots, \alpha_n\} \subset \mathbb{F}_{q^m}$ is a fixed set of n linearly independent vectors over $\mathbb{F}_q$ same as in the Koetter-Kschischang codes.

To express the several steps in the generalization of Koetter-Kschischang codes in a more convenient way, we only consider the case of the list-2 decoding algorithm. We will see that everything can be simply generalized for an arbitrary list size L. The first step in modifying the Koetter-Kschischang codes in order to enable list-2 decoding is the following. We transmit $f_{\boldsymbol{u}}^{\otimes 2}(\alpha_i)$ along with α_i and $f_{\boldsymbol{u}}(\alpha_i)$, where $f_{\boldsymbol{u}}$ is the message

polynomial. This is one of the important differences between this work and Sudan's list-decoding algorithm of RS codes. In Sudan's algorithm, there is no need to modify the Reed-Solomon code. One can compute powers of the received symbols y_i at the decoder. In fact, once y_i is given, all powers of y_i come for free whereas this is not the case in subspace codes. In general, given $f_{\boldsymbol{u}}(\alpha_i)$ one can not compute $f_{\boldsymbol{u}}^{\otimes 2}(\alpha_i)$. This enforces the modification of the Koetter-Kschischang codes which will be elaborated through this section.

Based on the foregoing discussion, the first attempt at a simple generalization of Koetter-Kschischang codes, which enables a list-2 decoding, is explained as follows. The message vector $\boldsymbol{u} = (u_0, u_1, \ldots, u_{k-1})$ consists of k information symbols as elements of $\mathbb{F}_{q^m}$, where $k \leqslant m$. Let $f_{\boldsymbol{u}}(X) = \sum_{i=0}^{k-1} u_i X^{[i]}$ be the corresponding message polynomial. Then the corresponding codeword V is the vector space spanned by the set

$$\left\{ \big(\alpha_1, f_{\boldsymbol{u}}(\alpha_1), f_{\boldsymbol{u}}^{\otimes 2}(\alpha_1)\big), \ldots, \big(\alpha_n, f_{\boldsymbol{u}}(\alpha_n), f_{\boldsymbol{u}}^{\otimes 2}(\alpha_n)\big) \right\}$$

Since α_i's are linearly independent, V has dimension n. V is transmitted through the operator channel and another vector space U of dimension r is received at the receiver. Let $(x_i, y_i, z_i), i = 1, \ldots, r$, be a basis for U. At the decoder, we construct a non-zero trivariate linearized polynomial $Q(X, Y, Z)$ of the form

$$Q(X, Y, Z) = Q_0(X) + Q_1(Y) + Q_2(Z) \tag{2.13}$$

where Q_i's are linearized polynomials over $\mathbb{F}$ subject to certain degree constraints specified later, such that $Q(x_i, y_i, z_i) = 0$ for $i = 1, \ldots, r$. Since Q is linearized, it is zero over the whole vector space U, in particular over the intersection of V and U. Therefore, assuming that not too many errors and erasures happen, the polynomial

$$Q(X, f_{\boldsymbol{u}}(X), f_{\boldsymbol{u}}^{\otimes 2}(X)) = Q_0(X) + Q_1 \otimes f_{\boldsymbol{u}}(X) + Q_2 \otimes f_{\boldsymbol{u}}^{\otimes 2}(X)$$

is guaranteed to have a certain number of linearly independent roots which is more than its q-degree. Thus it is identically zero and the next step is to recover the message polynomial from it. The problem is how many possible solutions for $f_{\boldsymbol{u}}(X)$ there are and how to extract them. Unfortunately, there might be more than two solutions for $f_{\boldsymbol{u}}(X)$. In general, an equation over a non-commutative ring may have more zeros

than its degree. We illustrate this for the ring of linearized polynomials by an example. Consider the equation:

$$f^{\otimes 2}(X) - X^{q^2} = 0$$

This can be regarded as an equation of degree 2 over the ring of linearized polynomials. Then $f(X) = uX^q$ is a solution for this equation for any u which satisfies $u^{q+1} = 1$. If m is even, then $q+1$ divides $q^m - 1$. Therefore there are $q+1$ distinct possible values for u each of which gives a distinct solution for $f(X)$.

2.3.3 Solving the List-Size Problem

As discussed in the foregoing subsection, an equation over the ring of linearized polynomials may have more zeros than its root. This is a a consequence of the fact that the ring of linearized polynomials is not commutative. To solve this problem, the idea is to restrict the set of message polynomials to a commutative subring of this ring. Lemma 2.3.1 shows that linearized polynomials over the base field $\mathbb{F}_q$, $\mathscr{L}_q[X]$, form a commutative subring of $\mathscr{L}_{q^m}[X]$. Theorem 2.3.2 proves that an equation of degree L over the ring of linearized polynomials has at most L roots in $\mathscr{L}_q[X]$, as expected. This suggests the following solution for the problem of having more than two roots. We only consider message polynomials that are over $\mathbb{F}_q$ rather than $\mathbb{F}_{q^m}$ that is, we assume that the message is a vector $\boldsymbol{u} = (u_0, u_1, \ldots, u_{k-1})$ of length k over $\mathbb{F}_q$.

Lemma 2.3.1. *Let $f(X)$ and $g(X)$ be $\mathbb{F}_q$-linearized polynomials over $\mathbb{F}_q$. Then they commute i.e.*

$$f(X) \otimes g(X) = g(X) \otimes f(X)$$

Proof. Let $f(X) = \sum_{i \geqslant 0} f_i X^{[i]}$ and $g(X) = \sum_{j \geqslant 0} g_j X^{[j]}$. Then by (1.9),

$$f(X) \otimes g(X) = \sum_{k \geqslant 0} c_k X^{[k]},$$

where $c_k = \sum_{i=0}^{k} f_i g_{k-i}^{[i]}$ and

$$g(X) \otimes f(X) = \sum_{k \geqslant 0} c'_k X^{[k]},$$

where $c'_k = \sum_{i=0}^{k} f_i^{[k-i]} g_{k-i}$. Since $f_i, g_j \in \mathbb{F}_q$, $f_i^{[k-i]} = f_i^{q^{k-i}} = f_i$ and $g_{k-i}^{[i]} = g_{k-i}^{q^i} = g_{k-i}$, for any i and k. It implies that for any k,

$$c_k = \sum_{i=0}^{k} f_i g_{k-i} = c'_k$$

Therefore, $f(X) \otimes g(X) = g(X) \otimes f(X)$. ■

Theorem 2.3.2. *Let $Q_i(X)$, $i = 0, 1, \ldots, L$, be linearized polynomials in $\mathscr{L}_{q^m}[X]$ such that at least one of them is non-zero. Then the equation*

$$\sum_{i=0}^{L} Q_i \otimes f^{\otimes i}(X) = 0 \tag{2.14}$$

has at most L solutions for $f(x) \in \mathscr{L}_q[X]$.

Proof. We do induction on L for $L \geqslant 0$. For $L = 0$, Q_0 has to be non-zero. Thus there is no solution for (2.14). Now, suppose that it is true for $L - 1$ and we want to prove it for L. If (2.14) does not have any solution for $f(X)$, then we are done. Otherwise, let $f_0(X)$ be a solution for (2.14) that is,

$$\sum_{i=0}^{L} Q_i \otimes f_0^{\otimes i}(X) = 0 \tag{2.15}$$

We show that there are at most $L - 1$ other solutions excluding f_0. Subtracting (2.15) from (2.14) we get

$$\sum_{i=1}^{L} Q_i \otimes (f^{\otimes i} - f_0^{\otimes i}) = 0 \tag{2.16}$$

Since f and f_0 are both over $\mathbb{F}_q$, by Lemma 2.3.1 they commute. As a result,

$$f^{\otimes i} - f_0^{\otimes i} = (\sum_{j=0}^{i-1} f_0^{\otimes(i-j-1)} \otimes f^{\otimes j}) \otimes (f - f_0)$$

for any $i \geqslant 1$. Plugging in this into (2.16) we get

$$\sum_{i=1}^{L} Q_i \otimes \big(\sum_{j=0}^{i-1} (f_0^{\otimes(i-j-1)} \otimes f^{\otimes j}) \otimes (f - f_0)\big) = 0 \Rightarrow$$
$$\big(\sum_{i=1}^{L} Q_i \otimes \sum_{j=0}^{i-1} f_0^{\otimes(i-j-1)} \otimes f^{\otimes j}\big) \otimes (f - f_0) = 0$$

Since $f - f_0 \neq 0$, we can divide both sides by $f - f_0$ to get

$$\sum_{i=1}^{L} \big(Q_i \otimes \sum_{j=0}^{i-1} f_0^{\otimes(i-j-1)} \otimes f^{\otimes j}\big) = 0 \Rightarrow$$
$$\sum_{j=0}^{L-1} \big(\sum_{i=j+1}^{L} Q_i \otimes f_0^{\otimes(i-j-1)}\big) \otimes f^{\otimes j} = 0$$

which has at most $L - 1$ solutions for $f(X)$ by induction hypothesis. This completes the proof. ■

In summary, we consider the message $\boldsymbol{u} = (u_0, \ldots, u_{k-1})$ as a vector of k information symbols over $\mathbb{F}_q$ rather than $\mathbb{F}_{q^m}$ in order to solve the list-size problem. This leads to a rate reduction by a factor of m.

2.3.4 Solving the Rate Penalty Problem

In this subsection, we propose a solution for the rate penalty problem. Indeed, we take advantage of the fact that the message polynomial is over the base field $\mathbb{F}_q$ in order to compensate the rate reduction at the decoder.

Recall from [8, Ch. 4.9] that any finite extension $\mathbb{F}_{q^m}$ of $\mathbb{F}_q$ as a vector space over $\mathbb{F}_q$ has a basis of the form $\alpha, \alpha^q, \ldots, \alpha^{q^{m-1}}$, where α is a primitive element of $\mathbb{F}_{q^m}$. This is called a normal basis for $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Suppose that $f(X)$ is a linearized polynomial over $\mathbb{F}_q$. Then for any j, $f(\alpha^{q^j}) = f(\alpha)^{q^j}$. This implies that given $f(\alpha)$ one can determine $f(\alpha^{q^j})$, for $j = 1, 2, \ldots, m-1$. Therefore, $f(\alpha^q), f(\alpha^{q^2}), \ldots, f(\alpha^{q^{m-1}})$ do not need to be transmitted. The idea is to manufacture them at the receiver while only $f(\alpha)$ is transmitted. We elaborate on this idea by specifying an encoding and list-decoding algorithm.

The input to the encoder is a message $\boldsymbol{u} = (u_0, u_1, \ldots, u_{k-1})$ in ${\mathbb{F}_q}^k$. The corresponding message polynomial is $f_{\boldsymbol{u}}(X) = \sum_{i=0}^{k-1} u_i X^{[i]}$. The output of the encoder is the one dimensional subspace V as follows:

$$V = \Big\langle \big(\alpha, f_{\boldsymbol{u}}(\alpha), f_{\boldsymbol{u}}^{\otimes 2}(\alpha)\big) \Big\rangle$$

The decoder takes the received vector space, denoted by U, as the input. Suppose that the dimension of U is r. The first step of the list-decoding algorithm is to compute the

set of interpolation points. The decoder first finds a basis $(x_i, y_i, z_i), i = 1, 2, \ldots, r$ for U. Then the set of interpolation points $\mathcal{P}$ is as follows:

$$\mathcal{P} = \left\{ \left(x_i^{q^j}, y_i^{q^j}, z_i^{q^j} \right) : 1 \leqslant i \leqslant r, 0 \leqslant j \leqslant m-1 \right\}$$

The next step is to construct the interpolation polynomial. The decoder constructs a non-zero trivariate linearized polynomial

$$Q(X, Y, Z) = Q_0(X) + Q_1(Y) + Q_2(Z)$$

such that Q passes through all the interpolation points that is,

$$Q(x, y, z) = 0$$

for any $(x, y, z) \in \mathcal{P}$. Q_0, Q_1 and Q_2 are linearized polynomials over $\mathbb{F}$ and Q_0 has q-degree at most $m-1$, Q_1 has q-degree at most $m-k$ and Q_2 has q-degree at most $m-2k+1$. If the dimension of error is less than a certain threshold, the following equality holds:

$$Q\big(X, f_{\boldsymbol{u}}(X), f_{\boldsymbol{u}}^{\otimes 2}(X)\big) = 0$$

Thus the last step of the list-decoding algorithm is to find all the roots $f(X) \in \mathscr{L}_q[X]$, with degree at most $k-1$, of the equation:

$$Q\big(X, f(X), f^{\otimes 2}(X)\big) = 0$$

using the LRR algorithm, that will be discussed in Section 2.6. The decoder outputs coefficients of each root $f(X)$ as a vector of length k. Theorem 2.3.2 guarantees that the size of the output list is at most 2.

2.3.5 General List-Size

In this subsection, we generalize the encoding and list-2 decoding algorithm explained in the foregoing subsection to general list size yet the construction is one dimensional. To this end, we transmit all powers of $f_{\boldsymbol{u}}(X)$ up to $f_{\boldsymbol{u}}^{\otimes L}(X)$, where $f_{\boldsymbol{u}}$ is the message polynomial, in order to do list-L decoding at the receiver.

The following parameters of the code are fixed: finite field $\mathbb{F}_q$, an extension $\mathbb{F} = \mathbb{F}_{q^m}$, number of information symbols k, list size L and $\alpha \in \mathbb{F}$ which generates a

normal basis for $\mathbb{F}$. The required condition is that $k \leqslant m$. The ambient space $\mathcal{W}$ is an $Lm+1$-dimensional vector space over $\mathbb{F}_q$.

Encoding Algorithm:

Formally, the encoder is a function $\mathcal{E} : {\mathbb{F}_q}^k \to \mathcal{G}_q(\mathcal{W}, n)$. It accepts as input a message vector $\boldsymbol{u} = (u_0, u_1, \ldots, u_{k-1}) \in {\mathbb{F}_q}^k$. The message polynomial is constructed as $f_{\boldsymbol{u}}(X) = \sum_{i=0}^{k-1} u_i X^{[i]}$. Then the encoder outputs the following one dimensional subspace V:

$$V = \left\langle \left(\alpha, f_{\boldsymbol{u}}(\alpha), f_{\boldsymbol{u}}^{\otimes 2}(\alpha), \ldots, f_{\boldsymbol{u}}^{\otimes L}(\alpha)\right) \right\rangle$$

Definition 2.3.3. *The code $\mathbb{C}_q(k, 1, m, L)$ is the collection of all possible codewords V generated by this encoding algorithm. The second parameter stands for the dimension of the code which is equal to 1 for this code.*

Remark. Each element of the ambient space $\mathcal{W}$ is indicated as a vector with $L+1$ coordinates such as $(x, y_1, y_2, \ldots, y_L)$, where $x \in \langle\alpha\rangle$ and all other coordinates are elements of $\mathbb{F}_{q^m}$. □

Suppose that V is transmitted through the operator channel and another subspace U of $\mathcal{W}$ of dimension $1+t$ is received, where t is the dimension of error. We assume that no erasure happens as only one erasure may destroy all the information. The decoder first checks if the following condition on t is satisfied:

$$t < L - \frac{L(L+1)}{2}\frac{(k-1)}{m} \tag{2.17}$$

If not, then the decoder declares decoding failure. Otherwise, the decoder runs the list-decoding algorithm.

List-decoding Algorithm:

The decoder accepts as input the received vector space U. The output is a list of size at most L of vectors in ${\mathbb{F}_q}^k$ after executing these three steps:

1. *Computing the interpolation points:*

   Find a basis $(x_i, y_{i,1}, \ldots, y_{i,L}), i = 1, 2, \ldots, t+1$ for U. Then the the set of interpolation points is:

$$\mathcal{P} = \left\{ \left(x_i^{q^j}, y_{i,1}^{q^j}, \ldots, y_{i,L}^{q^j}\right) : 1 \leqslant i \leqslant t+1, 0 \leqslant j \leqslant m-1 \right\}$$

2. *Interpolation:*

   Construct a non-zero multivariate linearized polynomial $Q(X, Y_1, \ldots, Y_L)$ of the form

   $$Q_0(X) + Q_1(Y_1) + \cdots + Q_L(Y_L)$$

   with each Q_i having q-degree at most $m-(k-1)i-1$, for $i = 0, 1, \ldots, L$, subject to the constraint that

   $$Q(x, y_1, \ldots, y_L) = 0 \tag{2.18}$$

   for any $(x, y_1, \ldots, y_L) \in \mathcal{P}$.

3. *Factorization:*

   Find all the roots $f(X) \in \mathscr{L}_q[X]$, with q-degree at most $k-1$, of the equation:

   $$Q\big(X, f(X), \ldots, f^{\otimes L(X)}\big) = 0 \tag{2.19}$$

   using the LRR algorithm. The decoder outputs coefficients of each root $f(X)$ as a vector of length k.

2.3.6 Correctness of the List-Decoding Algorithm

In this subsection, we establish correctness of the list-decoding algorithm that we proposed in the foregoing subsection. To this end, we first establish a certain threshold on the dimension of error which in fact guarantees the existence of the interpolation polynomial. Then we prove that the message polynomial is included in the list generated by the list-decoding algorithm. The threshold on the dimension of error leads to the error-correction radius of the list-decoding algorithm which will be discussed in the next subsection.

Lemma 2.3.4. *There is a non-zero solution for multivariate linearized polynomial Q which satisfies (*2.18*) provided that*

$$t < L - \frac{L(L+1)}{2}\frac{(k-1)}{m}$$

Proof. The set of interpolation points $\mathcal{P}$ contains $m(1+t)$ points. Therefore, (2.18) defines a homogeneous system of $m(1+t)$ linear equations. The number of unknown

coefficients is

$$\sum_{i=0}^{L} m - (k-1)i = (L+1)m - (k-1)\frac{L(L+1)}{2}$$

It is known that if the number of variables in a homogeneous system of linear equations is strictly smaller than the number of equations, then there is a non-trivial solution:

$$m(1+t) < (L+1)m - (k-1)\frac{L(L+1)}{2} \tag{2.20}$$

that guarantees a non-zero solution for Q. (2.20) is equivalent to

$$t < L - \frac{L(L+1)}{2}\frac{(k-1)}{m}$$

■

Let $f_{\boldsymbol{u}}(X)$ be the message polynomial and $Q(X, Y_1, \ldots, Y_L)$ be the interpolation polynomial provided by the list-decoding algorithm. Then we form the linearized polynomial $E(X)$ as follows:

$$E(X) = Q\big(X, f_{\boldsymbol{u}}(X), \ldots, f_{\boldsymbol{u}}^{\otimes L}(X)\big) = \sum_{i=0}^{L} Q_i \otimes f_{\boldsymbol{u}}^{\otimes i}(X)$$

Lemma 2.3.5. *For $j = 0, 1, \ldots, m-1$, α^{q^j} is a root of $E(X)$.*

Proof. Since we assume that no erasure occurs, the transmitted codeword V is contained in the received subspace U. In particular, U includes the vector $\big(\alpha, f_{\boldsymbol{u}}(\alpha), \ldots, f_{\boldsymbol{u}}^{\otimes L}(\alpha)\big)$. Notice that raising to the power q^j is a linear operation. Therefore, $\big(x^{q^j}, y_1^{q^j}, \ldots, y_L^{q^j}\big)$ is a linear combination of some elements of the set of interpolation points $\mathcal{P}$, for any $(x, y_1, \ldots, y_L) \in U$, as $\mathcal{P}$ contains all the q^j-powers of the basis elements of U. Furthermore, Q is a linearized polynomial. Therefore,

$$Q\big(x^{q^j}, y_1^{q^j}, \ldots, y_L^{q^j}\big) = 0$$

In particular,

$$Q\big(\alpha^{q^j}, f_{\boldsymbol{u}}(\alpha)^{q^j}, \ldots, f_{\boldsymbol{u}}^{\otimes L}(\alpha)^{q^j}\big) = 0 \tag{2.21}$$

Note that for any polynomial $f(X) \in \mathbb{F}_q[X]$, $f(X^{q^j}) = f(X)^{q^j}$. Since all the coefficients of $f_{\boldsymbol{u}}^{\otimes i}(X)$ are elements of $\mathbb{F}_q$, (2.21) implies that

$$E(\alpha^{q^j}) = Q\big(\alpha^{q^j}, f_{\boldsymbol{u}}(\alpha^{q^j}), \ldots, f_{\boldsymbol{u}}^{\otimes L}(\alpha^{q^j})\big) = 0$$

■

Corollary 2.3.6. *$E(X)$ is the all zero polynomial.*

Proof. Since the q-degree of $f_{\boldsymbol{u}}(X)$ is at most $k-1$, the q-degree of $Q_i \otimes f_{\boldsymbol{u}}^{\otimes i}(X)$ is at most

$$m-(k-1)i-1+(k-1)i = m-1,$$

for $i = 0, 1, \ldots, L$. This implies that q-degree of $E(X)$ is at most $m-1$. On the other hand, $E(X)$ has at least m linearly independent roots $\alpha, \alpha^q, \ldots, \alpha^{q^{m-1}}$ by Lemma 2.3.5. Therefore, $E(X)$ must be the all zero polynomial. ■

Theorem 2.3.7. *The list-decoding algorithm produces a list of size at most L which includes the transmitted message $\boldsymbol{u}$ provided that*

$$t < L - \frac{L(L+1)}{2}\frac{(k-1)}{m}$$

Proof. By Lemma 2.3.4 the non-trivial interpolation polynomial Q exists. Then by Corollary 2.3.6, $E(X)$ is identically zero which means that the message polynomial $f_{\boldsymbol{u}}(X)$ is a solution to (2.19). Also, as Q is non-zero, (2.19) has at most L solutions by Theorem 2.3.2. Therefore, the list size is at most L. ■

2.3.7 Error-Correction Radius

In this subsection, we derive the bound on the error-correction radius of the proposed list-decoding algorithm. The ambient space $\mathcal{W}$ has dimension $Lm+1$. Each codeword is a one dimensional subspace of $\mathcal{W}$. Therefore, $n = 1$ and the rate R of the code $\mathbb{C}_q(k, 1, m, L)$ is given as follows:

$$R = \frac{\log_q(|\mathbb{C}_q(k,1,m,L)|)}{nN} = \frac{k}{Lm+1}$$

The q-degree of Q_L, the one with the smallest degree among Q_i's, has to be non-negative which leads to the following series of inequalities:

$$\begin{aligned} m-(k-1)L-1 \geqslant 0 \Rightarrow & \\ L \leqslant \frac{m-1}{k-1} \approx \frac{1}{LR} \Rightarrow & \\ R \leqslant \frac{1}{L^2} & \end{aligned}$$

By Theorem 2.3.7, the message is recovered as long as

$$t < L - \frac{L(L+1)}{2}\frac{(k-1)}{m}$$

Since $n = 1$, the error-correction radius τ is actually equal to t in this case. Observe that

$$R = \frac{k}{Lm+1} > \frac{k-1}{Lm}$$

Therefore, we guarantee that the message is successfully recovered provided that

$$\tau \leqslant L - \frac{L^2(L+1)}{2}R$$

2.4 List-decodable Codes of Arbitrary Dimension

In the foregoing section, we proposed list-decodable subspace codes along with a corresponding list-L decoding algorithm. A significant weakness of the construction is that the codewords are one dimensional. One dimensional codes seem somewhat unnatural. Besides, as the normalized dimension of error τ has to be an integer in this case, we are not able to take advantage of the whole achievable region for τ. In this section, we generalize our construction to an arbitrary dimension.

One simple way to generalize this construction to dimension 2 is the following. In the construction of $\mathbb{C}_q(k,1,m,L)$, the linear span of $(\alpha, f_{\boldsymbol{u}}(\alpha), \ldots, f_{\boldsymbol{u}}^{\otimes L}(\alpha))$ is the codeword corresponding to message polynomial $f_{\boldsymbol{u}}$, where α is the generator of a normal basis for $\mathbb{F}_{q^m}$. Suppose that β is another primitive element of $\mathbb{F}_{q^m}$ which generates a normal basis for $\mathbb{F}_{q^m}$. Then the corresponding codeword is the $\mathbb{F}_q$-linear span of $\big(\alpha, f_{\boldsymbol{u}}(\alpha), \ldots, f_{\boldsymbol{u}}^{\otimes L}(\alpha)\big)$ and $\big(\beta, f_{\boldsymbol{u}}(\beta), \ldots, f_{\boldsymbol{u}}^{\otimes L}(\beta)\big)$. When we inject more vectors into the network, we in fact add *redundancy* to the code, and we should get something

in return. Adding redundancy means more interpolation points at the receiver which in fact enforces more constraints. In return, we should get more zeros in order to maintain the same performance in terms of decoding radius versus rate. As we shall see in Lemma 2.3.5, however, the root space already covers the whole space $\mathbb{F}_{q^m}$. Therefore, this simple generalization does not lead to good performance. This becomes even worse as the dimension increases.

The idea is to evaluate the interpolation polynomial in a larger field that is, an extension $GF(q^{nm})$ of $GF(q^m)$. Message polynomials over $GF(q)$ are evaluated in n special bases, in such a way that the resulting interpolation polynomial is forced to have many zeros in $GF(q^{nm})$. We elaborate on this idea in this section.

Fix a finite field $\mathbb{F}_q$ and let n divide $q-1$. Then the equation $x^n - 1 = 0$ has n distinct solutions in $\mathbb{F}_q$. Let $e_1 = 1, e_2, e_3, \ldots, e_n$ be these solutions. Let $\mathbb{F} = GF(q^{nm})$ and γ be a generator of a normal basis for $\mathbb{F}$. Then define

$$\alpha_i = \gamma + e_i^{-1}\gamma^{q^m} + e_i^{-2}\gamma^{q^{2m}} + \cdots + e_i^{-(n-1)}\gamma^{q^{(n-1)m}} \tag{2.22}$$

for $i = 1, 2, \ldots, n$.

Next, we discuss the properties of the parameters α_i's.

Lemma 2.4.1. *$\alpha_1 \in \mathbb{F}_{q^m}$ and for $i = 2, 3, \ldots, n$, $\alpha_i^n \in \mathbb{F}_{q^m}$.*

Proof. For $i = 1, 2, \ldots, q-1$, $\alpha_i^{q^m} = e_i^{-1}\alpha_i$ by the following series of equalities:

$$\begin{aligned}
\alpha_i^{q^m} &= \Big(\sum_{j=0}^{n-1} e_i^j \gamma^{q^{mj}}\Big)^{q^m} = \sum_{j=0}^{n-1} (e_i^{q^m})^j \gamma^{q^{m(j+1)}} \\
&= \sum_{j=0}^{n-1} e_i^j \gamma^{q^{m(j+1)}} = e_i^{n-1}\gamma^{q^{nm}} + \sum_{j=1}^{n-1} e_i^{j-1}\gamma^{q^{mj}} \\
&= e_i^{-1}\gamma + \sum_{j=1}^{n-1} e_i^{j-1}\gamma^{q^{mj}} = \sum_{j=0}^{n-1} e_i^{j-1}\gamma^{q^{mj}} = e_i^{-1}\alpha_i
\end{aligned}$$

Then for $i = 1$, $\alpha_1^{q^m} = \alpha_1$ and therefore, $\alpha_1 \in \mathbb{F}_{q^m}$. For $i = 2, 3, \ldots, n$, $(\alpha_i^n)^{q^m} = e_i^{-n}\alpha_i^n = \alpha_i^n$ which implies that $\alpha_i^n \in \mathbb{F}_{q^m}$. ■

Lemma 2.4.2. *The set*

$$Z = \Big\{\alpha_i^{q^j} : 1 \leqslant i \leqslant n, 0 \leqslant j \leqslant m-1\Big\}$$

is a basis for $\mathbb{F}$.

Proof. Let A and Γ be $1 \times n$ vectors as follows:

$$A = (\alpha_1, \alpha_2, \ldots, \alpha_n)$$
$$\Gamma = \left(\gamma, \gamma^{q^m}, \ldots, \gamma^{q^{(n-1)m}}\right)$$

Also, let E be the $n \times n$ matrix whose (i, j) entry is e_i^{j-1}. Then by definition,

$$A = \Gamma E^t$$

Notice that E is a Vandermonde matrix and that e_i's are distinct. Therefore, the determinant of E is non-zero. Then we can write

$$\Gamma = A(E^{-1})^t$$

It implies that for any j, $\gamma^{q^{mj}}$ is a linear combination of α_i's. We can raise this to power q^r, for any $0 \leqslant r \leqslant m-1$, and say that $\gamma^{q^{mj+r}}$ is a linear combination of $\alpha_i^{q^r}$'s. Thus γ^{q^l} is a linear combination of elements of the set Z, for $0 \leqslant l \leqslant nm-1$. Therefore, elements of Z span the whole space $\mathbb{F}$. But $|Z| = nm$. Hence Z is a basis for $\mathbb{F}$. ■

2.4.1 Encoding and Decoding

The following parameters of the construction are fixed: the finite field $\mathbb{F}_q$ and an extension field $\mathbb{F}_{q^m}$, the number of information symbols k, the dimension of code n and the list size L. We require that $k \leqslant nm$ and $n \leqslant q-1$.

We let $[s]$ denote the set of positive integers less than or equal to s, for any positive integer s.

Extended Encoding Algorithm:

A message vector $\boldsymbol{u} = (u_0, u_1, \ldots, u_{k-1}) \in {\mathbb{F}_q}^k$ is the input to the encoder. The corresponding message polynomial is $f_{\boldsymbol{u}}(X) = \sum_{i=0}^{k-1} u_i X^{[i]}$. For $i = 1, 2, \ldots, n$, consider α_i defined in (2.22). The encoder constructs vectors $v_i \in W$ as follows. For $i = 1, 2, \ldots, n$,

$$v_i = \left(\alpha_i, f_{\boldsymbol{u}}(\alpha_i), f_{\boldsymbol{u}}^{\otimes 2}(\alpha_i), \ldots, f_{\boldsymbol{u}}^{\otimes L}(\alpha_i)\right)$$

The encoder then outputs the n-dimensional vector space V spanned by $v_1, v_2, \ldots, v_n$.

In this construction, the ambient space $\mathcal{W}$ is

$$\langle \alpha_1, \alpha_2, \ldots, \alpha_n \rangle \oplus \underbrace{\mathbb{F}_{q^{nm}} \oplus \cdots \oplus \mathbb{F}_{q^{nm}}}_{L \text{ times}} \tag{2.23}$$

which has dimension equal to $n + nmL$.

Remark. Each element in $\mathcal{W}$ is represented by a vector with $L+1$ coordinates such as $(x, y_1, y_2, \ldots, y_L)$, where x belongs to the vector space spanned by $\alpha_1, \alpha_2, \ldots, \alpha_n$ and $y_i \in \mathbb{F}_{q^{nm}}$, for $i = 1, 2, \ldots, L$. □

Definition 2.4.3. *The code $\mathbb{C}_q(k, n, m, L)$ is the collection of all possible codewords V generated by the extended encoding algorithm.*

Suppose that a codeword $V \in \mathbb{C}_q(k, n, m, L)$ is transmitted through the operator channel and the decoder receives a vector space $U \in \mathscr{P}_q(\mathcal{W})$ with dimension d. At the decoder we need a parameter ω which corresponds to the degree of the interpolation polynomial. ω is computed as follows:

$$\omega = \left\lceil \frac{md+1}{L+1} + \frac{1}{2}L(k-1) \right\rceil \tag{2.24}$$

This will guarantee existence of the interpolation polynomial Q in the extended list-decoding algorithm.

Extended List-decoding Algorithm:

1. *Computing the interpolation points:*

   Find a basis for U as follows:

   $$\{(x_i, y_{i,1}, y_{i,2}, \ldots, y_{i,L}) : i = 1, 2, \ldots, d\}$$

   Then for $h = 0, 1, 2, \ldots, m-1$, the set $\mathcal{P}_h$ is defined as follows:

   $$\mathcal{P}_h = \left\{ (x_i^{q^h}, y_{i,1}^{q^h}, \ldots, y_{i,L}^{q^h}) : i \in [d] \right\}$$

   The set of interpolation points $\mathcal{P}$ is equal to:

   $$\mathcal{P} = \mathcal{P}_0 \cup \mathcal{P}_1 \cup \cdots \cup \mathcal{P}_{m-1}$$

2. *Interpolation:*

   Construct a non-zero multivariate linearized polynomial $Q(X, Y_1, Y_2, \dots, Y_L)$ of the form

   $$Q_0(X) + Q_1(Y_1) + Q_2(Y_2) + \cdots + Q_L(Y_L)$$

   with each Q_i having q-degree at most $\omega - (k-1)i - 1$, for $i = 0, 1, \dots, L$ subject to the constraint that

   $$Q(x, y_1, y_2, \dots, y_L) \;=\; 0 \tag{2.25}$$

   for any $(x, y_1, y_2, \dots, y_L) \in \mathcal{P}$.

3. *Factorization:*

   Find all the roots $f(X) \in \mathscr{L}_q[X]$, with q-degree at most $k-1$, of the equation:

   $$Q\big(X, f(X), \dots, f^{\otimes L}(X)\big) = 0 \tag{2.26}$$

   using the LRR algorithm. The decoder outputs coefficients of each root $f(X)$ as a vector of length k.

The first step of the extended list-decoding algorithm is done by elementary linear algebraic operations. The interpolation step is in fact solving a system of linear equations. There are several ways to do that. The most straightforward way is the Gaussian elimination method. This method does not take advantage of the certain structure of this system of equations and therefore, it is not efficient. An efficient polynomial-time interpolation algorithm in the ring of linearized polynomials is presented in [11]. The factorization step can be done using the linearized Roth-Ruckenstein algorithm, called the LRR algorithm, which will be presented in detail in Section 2.6. We have modified the Roth-Ruckenstein algorithm [9] in order to solve equations over the ring of linearized polynomials. For instance, an equation of degree L over $\mathscr{L}_{q^m}[X]$ has the following form:

$$Q_0(X) + Q_1(X) \otimes f(X) + \cdots + Q_L(X) \otimes f^{\otimes L}(X) = 0$$

where Q_i's are linearized polynomials over $\mathbb{F}_{q^m}$. The LRR algorithm finds all the roots of this equation in efficient polynomial time.

2.4.2 Correctness of the Extended List-Decoding Algorithm

Lemma 2.4.4. *The choice of ω in (2.24) guarantees existence of a non-zero solution for the interpolation polynomial Q that satisfies (2.25).*

Proof. (2.25) defines a homogeneous system of at most md equations. The number of unknown coefficients is as follows:

$$\sum_{i=0}^{L} \omega - (k-1)i = (L+1)\omega - (k-1)\frac{L(L+1)}{2}$$

A non-zero solution for this homogeneous system of linear equations is guaranteed if and only if the number of equations is strictly less than the number of variables i.e.

$$md < (L+1)\omega - (k-1)\frac{L(L+1)}{2} \Leftrightarrow$$
$$\omega \geqslant \frac{md+1}{L+1} + \frac{1}{2}L(k-1)$$

This is guaranteed by the choice of ω in (2.24). ■

Lemma 2.4.5. *The linear spans of the sets $\mathcal{P}_h$, defined in the first step of the extended list-decoding algorithm, are disjoint for $h = 0, 1, \ldots, m-1$.*

Proof. For any $i \in [d]$, x_i is an element of the span of $\alpha_1, \alpha_2, \ldots, \alpha_n$. Since raising to power q^h is a linear operation, $x_i^{q^h}$ is an element of

$$\left\langle \alpha_1^{q^h}, \alpha_2^{q^h}, \ldots, \alpha_n^{q^h} \right\rangle$$

By Lemma 2.4.2, these are disjoint vector spaces as h varies between 0 and $m-1$. Therefore, linear spans of $\mathcal{P}_h$'s are also disjoint as h varies between 0 and $h-1$. ■

We form the following linearized polynomial $E(X)$ wherein $f_{\boldsymbol{u}}(X)$ is the message polynomial and $Q(X, Y_1, \ldots, Y_L)$ is the interpolation polynomial provided by the extended list-decoding algorithm.

$$E(X) = Q\big(X, f_{\boldsymbol{u}}(X), \ldots, f_{\boldsymbol{u}}^{\otimes L}(X)\big) = \sum_{i=0}^{L} Q_i \otimes f_{\boldsymbol{u}}^{\otimes i}(X)$$

Suppose that the number of errors in the received subspace U is t and the number of erasures is ρ. Thus $d = n - \rho + t$.

Lemma 2.4.6. *The linearized polynomial $E(X)$ has at least $(n-\rho)m$ linearly independent roots.*

Proof. Let U' be the intersection of the transmitted codeword V and the received subspace U. Then U' is a subspace of the received vector space U with dimension $n-\rho$. For any $(x, y_1, \ldots, y_L) \in U'$ and $h = 0, 1, \ldots, m-1$, $\left(x^{q^h}, y_1^{q^h}, \ldots, y_L^{q^h}\right)$ is an element of the linear span of the set $\mathcal{P}_h$, because $\mathcal{P}_h$ contains all the q^h-powers of the basis elements of U and raising to the power q^h is a linear operation. Furthermore, Q is a linearized polynomial. Hence,

$$Q\left(x^{q^h}, y_1^{q^h}, \ldots, y_L^{q^h}\right) = 0 \tag{2.27}$$

On the other hand, $(x, y_1, \ldots, y_L)$ is also an element of the transmitted codeword V. Therefore,

$$(x, y_1, \ldots, y_L) = \left(\beta, f_{\boldsymbol{u}}(\beta), \ldots, f_{\boldsymbol{u}}^{\otimes L}(\beta)\right)$$

for some β in the linear span of $\alpha_1, \alpha_2, \ldots, \alpha_n$. Since coefficients of $f_{\boldsymbol{u}}(X)$ are elements of $\mathbb{F}_q$, for any integer h

$$\left(x^{q^h}, y_1^{q^h}, \ldots, y_L^{q^h}\right) = \left(\beta^{q^h}, f_{\boldsymbol{u}}(\beta^{q^h}), \ldots, f_{\boldsymbol{u}}^{\otimes L}(\beta^{q^h})\right) \tag{2.28}$$

Notice that linear spans of the sets $\mathcal{P}_h$ are disjoint by Lemma 2.4.5. This together with (2.27) and (2.28) implies that there are at least $(n-\rho)m$ linearly independent roots for $E(X)$. ■

Corollary 2.4.7. *If $\omega \leqslant (n-\rho)m$, then the linearized polynomial $E(X)$ is identically zero.*

Proof. The q-degree of $f_{\boldsymbol{u}}(X)$ is at most $k-1$. Therefore, the q-degree of $Q_i(X) \otimes f_{\boldsymbol{u}}^{\otimes i}(X)$ is at most

$$\omega - (k-1)i - 1 + (k-1)i = \omega - 1,$$

for $i = 0, 1, \ldots, L$. Thus the q-degree of $E(X)$ is at most $\omega - 1$. On the other hand, $E(X)$ has at least $(n-\rho)m$ linearly independent roots by Lemma 2.4.6. Therefore, $E(X)$ must be the all zero polynomial. ■

Theorem 2.4.8. *The output of the extended list-decoding algorithm is a list of size at most L which includes the transmitted message $\boldsymbol{u}$ provided that*

$$L\rho + t < nL - \frac{L(L+1)}{2}\frac{(k-1)}{m} \tag{2.29}$$

Proof. The existence of a non-zero interpolation polynomial Q that satisfies (2.25) is guaranteed by Lemma 2.4.4. Then by Corollary 2.4.7, $E(X)$ is the all zero polynomial provided that

$$\frac{md+1}{L+1} + \frac{1}{2}L(k-1) \leqslant (n-\rho)m \tag{2.30}$$

where we have plugged in the expression for ω from (2.24). We plug in $d = n - \rho + t$ into (2.30). Then observe that (2.30) is a consequence of

$$L\rho + t < nL - \frac{L(L+1)}{2}\frac{(k-1)}{m}$$

Thus this condition on the number of errors and erasures implies that $E(X)$ is identically zero. Therefore, the message polynomial $f_{\boldsymbol{u}}(X)$ is a solution to (2.26). Also, since Q is non-zero, (2.26) has at most L solutions by Theorem 2.3.2. Therefore, the list size is at most L. ■

2.4.3 Error-Correction Radius

The ambient space $\mathcal{W}$ in the construction of the code $\mathbb{C}_q(k, n, m, L)$ has dimension $n + nmL$. Each codeword is an n-dimensional subspace of $\mathcal{W}$. The rate R of the code is

$$R = \frac{\log_q(|\mathbb{C}_q(k, n, m, L)|)}{nN} = \frac{k}{n(n+nmL)}$$

The q-degree of each Q_i must be non-negative. Notice that Q_L has the smallest degree among Q_i's. This leads to to the following series of inequalities:

$$\begin{aligned} nm - (k-1)L - 1 \geqslant 0 \Rightarrow \\ L \leqslant \frac{nm-1}{k-1} \approx \frac{1}{nLR} \Rightarrow \\ R \leqslant \frac{1}{nL^2} \end{aligned}$$

We define the *error-correction radius* τ as follows:

$$\tau = \frac{L\rho + t}{n}$$

τ is the total dimension of errors and erasures normalized by the dimension of the code n, where the dimension of erasures is weighted by a factor of L. Then by Theorem 2.4.8, the transmitted message is recovered as long as

$$\tau < L - \frac{L(L+1)}{2}\frac{(k-1)}{nm}$$

This bound can be expressed in terms of the rate R, that is

$$\tau < L - \frac{1}{2}nL(L + \frac{1}{m})(L+1)R$$

guarantees a correct list-decoding. In this expression, $L + \frac{1}{m}$ can be approximated by L. However, we still have the parameter n which prevents us from plotting the bound only in terms of R and L.

We introduce ***packet rate*** as a new parameter in order to express our results in a more convenient way. In fact, the rate R of the code is equal to the number of q-ary information symbols normalized by the number of q-ary symbols injected into the network. This can be interpreted as the *symbol rate* of the code. The packet rate R^* is equal to the number of information packets normalized by the number of encoded packets injected into the network. For Koetter-Kschischang code, the packet rate is

$$R^* = \frac{k}{n}$$

The bound in (2.10) on the error-correction radius of the Koetter-Kschischang code can be expressed in terms of the packet rate R^* that is,

$$\tau \leqslant 1 - R^*$$

guarantees successful recovery of the message.

In our construction, we also have k information packets, indeed of length 1 over $\mathbb{F}_q$, and n encoded packets. In order to make a fair comparison with the Koetter-Kschischang code, however, we assume that there is a common source which generates packets of length m over $\mathbb{F}_q$. In our construction, we actually have to break each packet

into symbols over $\mathbb{F}_q$ which are also regarded as packets of length 1 over $\mathbb{F}_q$. Having set that, the packet rate of our code is

$$R^* = \frac{k}{nm} \tag{2.31}$$

The transmitted message will be successfully recovered provided that

$$\tau \leqslant L - \frac{L(L+1)}{2} R^*$$

This bound is plotted in Figure 2.1.

2.5 Back to Koetter-Kschischang Codes

The codes constructed in this chapter so far depend on the intended list size L at the decoder. We actually need to transmit all the powers of the message polynomial $f_{\boldsymbol{u}}(X)$ up to L evaluated at certain values α_i in order to enable list-L decoding at the decoder. A natural question is then the following: Is there a way to do list-decoding if we only transmit the $f(\alpha_i)$'s the same as in the Koetter-Kschischang scheme? We do not know the answer to this question in general. As we will see, however, the answer is yes for the special case of one-dimensional codes. The idea is that we manufacture some powers of $f_{\boldsymbol{u}}(\alpha)$ from the received $f_{\boldsymbol{u}}(\alpha)$. In fact we are able to manufacture $f_{\boldsymbol{u}}^{q^j}(\alpha)$, for any positive integer j, from the received subspace. We present this idea first for the simpler case of fields of characteristic 2. Then we elaborate it for the general case and prove the correctness of the list-decoding algorithm.

For the sake of simplicity, we first consider the case $q = 2$. Similar to what we discussed for the construction of $\mathbb{C}_2(k, 1, m, L)$ in Section 2.3.5, suppose that α is a primitive element of the field $\mathbb{F}_{2^m}$ that generates a normal basis for $\mathbb{F}_{2^m}$ as a vector space over $\mathbb{F}_2$. Now, suppose that the transmitted codeword V is the one dimensional vector space $\big\langle \big(\alpha, f_{\boldsymbol{u}}(\alpha)\big) \big\rangle$. This actually matches the Koetter-Kschischang code with dimension 1 and the message polynomial evaluated at α. Remember that in construction of Koetter-Kschischang codes, the choice of vectors α_i, on which the message polynomial is evaluated, is arbitrary as long as they are linearly independent.

The ambient space $\mathcal{W}$ is an $(m+1)$-dimensional vector space over $\mathbb{F}_2$. The source transmits the codeword V through the network and another subspace U of $\mathcal{W}$ is

received. At the decoder, we first find a basis for U such as $\{(x_i, y_i) : i = 1, 2, \ldots, d\}$, where $x_i \in \langle\alpha\rangle$, $y_i \in \mathbb{F}_{2^m}$ and d is the dimension of U. Since $\alpha, \alpha^2, \ldots, \alpha^{2^{m-1}}$ is a basis for $\mathbb{F}_{2^m}$, each y_i can be uniquely written as a linear combination of them; that is,

$$y_i = u_{0,i}\alpha + u_{1,i}\alpha^2 + \cdots + u_{m-1,i}\alpha^{2^{m-1}}$$

where the coefficients, the $u_{j,i}$'s, are elements of $\mathbb{F}_2$. Then for $i = 1, 2, \ldots, d$, we define the linearized polynomial $f_i(X)$ as follows:

$$f_i(X) = u_{0,i}X + u_{1,i}X^2 + \cdots + u_{m-1,i}X^{2^{m-1}}$$

Let $z_i = f_i(f_i(\alpha))$. Then the set of interpolation points $\mathcal{P}$ is

$$\mathcal{P} = \left\{ \left(x_i^{q^j}, y_i^{q^j}, z_i^{q^j}\right) : 1 \leqslant i \leqslant d, 0 \leqslant j \leqslant m-1 \right\}$$

With this set of interpolation points, we do the interpolation and factorization steps exactly the same as in the list-decoding algorithm discussed in Section 2.3.5 for $L = 2$.

We have to make sure that

$$\left(\alpha, f_{\boldsymbol{u}}(\alpha), f_{\boldsymbol{u}}^{\otimes 2}(\alpha)\right) \in \left\langle (x_i, y_i, z_i) : i \in [d] \right\rangle \tag{2.32}$$

provided that

$$\left(\alpha, f_{\boldsymbol{u}}(\alpha)\right) \in \left\langle (x_i, y_i) : i \in [d] \right\rangle$$

Then the rest of proof that this list-decoding algorithm is correct, provided that there is a certain bound on the number of errors, is similar to what we had before. We also assume there are no erasures. Hence, $\left(\alpha, f_{\boldsymbol{u}}(\alpha)\right)$ is contained in the received subspace U. Therefore,

$$\left(\alpha, f_{\boldsymbol{u}}(\alpha)\right) = \sum_{i=1}^{d} \lambda_i (x_i, y_i),$$

where $\lambda_i \in \mathbb{F}_2$ for $i \in [d]$. Thus,

$$f_{\boldsymbol{u}}(\alpha) = \sum_{i=1}^{d} \lambda_i f_i(\alpha)$$

This implies that

$$f_{\boldsymbol{u}}(X) = \sum_{i=1}^{d} \lambda_i f_i(X)$$

Because coefficients of $f_{\boldsymbol{u}}(X)$ and all $f_i(X)$'s are elements of $\mathbb{F}_2$ and also that α, α^2,...,$\alpha^{2^{m-1}}$ form a basis for $\mathbb{F}_{2^m}$. Then the following holds for $f_{\boldsymbol{u}}^{\otimes 2}(X)$:

$$f_{\boldsymbol{u}}^{\otimes 2} = (\sum_{i=1}^{d} \lambda_i f_i) \otimes (\sum_{i=1}^{d} \lambda_i f_i) = \sum_{i=1}^{d} \lambda_i^2 f_i^{\otimes 2} = \sum_{i=1}^{d} \lambda_i f_i^{\otimes 2}$$

where we used the fact that f_i's are elements of $\mathscr{L}_q[X]$ and therefore, they commute by Lemma 2.3.1. This proves that (2.32) holds as we required. Notice that we may even gain more by manufacturing $f_{\boldsymbol{u}}^{\otimes 2^i}(\alpha)$ for $i = 1, 2, \ldots, s$, for some positive integer s, at the decoder.

For the rest of this section, we consider the general case where the base field is $\mathbb{F}_q$, yet the construction is one dimensional. We fix the parameters of the code as follows: finite field $\mathbb{F}_q$, finite extension $\mathbb{F} = \mathbb{F}_{q^m}$ and the number of information symbols $= k$. We require the condition that $k \leqslant m$. The ambient space $\mathcal{W}$ is an $(m+1)$-dimensional vector space over $\mathbb{F}_q$. Let α be a primitive element of $\mathbb{F}_{q^m}$ that generates a normal basis for $\mathbb{F}_{q^m}$; that is, $\alpha, \alpha^q, \ldots, \alpha^{q^{m-1}}$ form a basis for $\mathbb{F}_{q^m}$ as a vector space over $\mathbb{F}_q$.

Encoding Algorithm:

Message vector $\boldsymbol{u} = (u_0, u_1, \ldots, u_{k-1}) \in {\mathbb{F}_q}^k$ is the input to the encoder. The message polynomial is $f_{\boldsymbol{u}}(X) = \sum_{i=0}^{k-1} u_i X^{[i]}$. The vector v is constructed as follows:

$$v = \big(\alpha, f_{\boldsymbol{u}}(\alpha)\big)$$

The output of the encoder is the one-dimensional vector space V spanned by v.

In fact, this encoding algorithm matches the Koetter-Kschischang encoding algorithm reviewed in Section 2.2, for dimension $n = 1$, except that the message symbols are restricted to the ground field $\mathbb{F}_q$. Also, we have made a particular choice for the evaluation point α.

Remark. The ambient space $\mathcal{W}$ is

$$\langle \alpha \rangle \oplus \mathbb{F}_{q^m}$$

whose dimension is $m+1$, as said before. Each element of $\mathcal{W}$ is represented as a vector with 2 coordinates such as (x, y), where x belongs to the vector space spanned by α and $y \in \mathbb{F}_{q^m}$.

Suppose that the codeword V is transmitted through the operator channel and subspace U of dimension $d = t + 1$ is received at the receiver, where t is the dimension of the error. We fix a basis for the received subspace U and denote it by

$$\Big\{(x_i, y_{i,0}) : i = 1, 2, \ldots, d\Big\}$$

Each $y_{i,0}$ can be uniquely written as a linear combination of $\alpha, \alpha^q, \ldots, \alpha^{q^{m-1}}$:

$$y_{i,0} = u_{0,i}\alpha + u_{1,i}\alpha^q + \cdots + u_{m-1,i}\alpha^{q^{m-1}}$$

And we define the linearized polynomial $f_i(X)$ as

$$f_i(X) = u_{0,i}X + u_{1,i}X^q + \cdots + u_{m-1,i}X^{q^{m-1}}$$

The decoder fixes a parameter s which is related to the list-size allowed at the output. In fact, the maximum list-size, in terms of the parameter s, is q^s. Let $y_{i,j} = f_i^{\otimes q^j}(\alpha)$, for $j = 1, 2, \ldots, s$. Then the list-decoding algorithm is performed as follows:

List-decoding Algorithm:

1. *Computing the interpolation points:*
   The set of interpolation points $\mathcal{P}$ is

   $$\mathcal{P} = \Big\{\big(x_i^{q^h}, y_{i,0}^{q^h}, \ldots, y_{i,s}^{q^h} : i \in [d], 0 \leqslant h \leqslant m-1\Big\}$$

2. *Interpolation:*
   Construct a non-zero multivariate polynomial $Q(X, Y_1, Y_2, \ldots, Y_{s+1})$ of the form

   $$Q_0(X) + Q_1(Y_1) + Q_2(Y_2) + \cdots + Q_{s+1}(Y_{s+1})$$

   where Q_0 is a linearized polynomial over $\mathbb{F}_{q^m}$ of q-degree at most $m-1$, and Q_j is a linearized polynomial over $\mathbb{F}_{q^m}$ of q-degree at most $m - (k-1)q^{j-1} - 1$, for $j = 1, 2, \ldots, s+1$, subject to the constraint that:

   $$Q(x, y_1, y_2, \ldots, y_{s+1}) \;=\; 0 \tag{2.33}$$

   for any $(x, y_1, y_2, \ldots, y_{s+1}) \in \mathcal{P}$.

3. *Factorization:*

   Find all the roots $f(X) \in \mathscr{L}_q[X]$, with degree of at most $k-1$ of the equation:

$$Q\big(X, f(X), f^{\otimes q}(X), \ldots, f^{\otimes q^s}(X)\big) = 0 \tag{2.34}$$

   using the LRR algorithm. The decoder outputs coefficients of each root $f(X)$ as a vector of length k.

Theorem 2.3.2 shows that number of solutions is at most q^s. Each solution corresponds to an output message $(u_0, u_1, \ldots, u_{k-1})$.

Lemma 2.5.1. *The vector*

$$(\alpha, f_{\boldsymbol{u}}(\alpha), f_{\boldsymbol{u}}^{\otimes q}(\alpha), \ldots, f_{\boldsymbol{u}}^{\otimes q^s}(\alpha))$$

is contained in the linear span of the set of vectors

$$\{(x_i, y_{i,0}, \ldots, y_{i,s}) : i = 1, 2, \ldots, d\}.$$

Proof. Since we assume that no erasure happens, the transmitted codeword V is contained in the received subspace U. Hence, $\big(\alpha, f_{\boldsymbol{u}}(\alpha)\big) \in U$. Therefore, we can write

$$\big(\alpha, f_{\boldsymbol{u}}(\alpha)\big) = \sum_{i=1}^{d} \lambda_i (x_i, y_{i,0})$$

where the coefficients λ_i's are elements of the base field $\mathbb{F}_q$. Then

$$f_{\boldsymbol{u}}(\alpha) = \sum_{i=1}^{d} \lambda_i y_{i,0} = \sum_{i=1}^{d} \lambda_i f_i(\alpha) \tag{2.35}$$

Since $f_{\boldsymbol{u}}$ and f_i's have q-degree of at most $m-1$ and $\alpha, \alpha^q, \ldots, \alpha^{q^{m-1}}$ are linearly independent, (2.35) implies that

$$f_{\boldsymbol{u}}(X) = \sum_{i=1}^{d} \lambda_i f_i(X)$$

Then for any positive integer h,

$$f_{\boldsymbol{u}}^{\otimes q^h} = (\sum_{i=1}^{d} \lambda_i f_i)^{\otimes q^h} = \sum_{i=1}^{d} \lambda_i^{q^h} f_i^{\otimes q^h} = \sum_{i=1}^{d} \lambda_i f_i^{\otimes q^h} \tag{2.36}$$

where the second equality follows from the fact that f_i's are elements of $\mathscr{L}_q[X]$ and therefore, they commute by Lemma 2.3.1. (2.36) imply that

$$\begin{aligned}
&(\alpha, f_{\boldsymbol{u}}(\alpha), f_{\boldsymbol{u}}^{q}(\alpha), \ldots, f_{\boldsymbol{u}}^{\otimes q^s}(\alpha)) \\
&= \sum_{i=1}^{d} \lambda_i (x_i, f_i(\alpha), f_i^{q}(\alpha), \ldots, f_i^{\otimes q^s}(\alpha)) \\
&= \sum_{i=1}^{d} \lambda_i (x_i, y_{i,0}, y_{i,1} \ldots, y_{i,s})
\end{aligned}$$

■

Lemma 2.5.2. *There is a non-zero solution for Q that satisfies (2.33) provided that the number of errors t is bounded as*

$$t < s + 1 - \left(\frac{q^{s+1}-1}{q-1}\right)\left(\frac{k-1}{m}\right)$$

Proof. (2.33) defines a homogeneous system of at most $m(t+1)$ linear equations. The number of unknown coefficients is

$$m + \sum_{h=0}^{s} m - (k-1)q^h = (s+2)m - (k-1)\left(\frac{q^{s+1}-1}{q-1}\right)$$

This system is guaranteed to have a non-zero solution if the number of equations is strictly less than the number of coefficients that is,

$$m(t+1) < (s+2)m - (k-1)\left(\frac{q^{s+1}-1}{q-1}\right)$$

which is equivalent to

$$t < s + 1 - \left(\frac{q^{s+1}-1}{q-1}\right)\left(\frac{k-1}{m}\right)$$

■

The polynomial $E(X)$ is defined as follows:

$$E(X) = Q\big(X, f_{\boldsymbol{u}}(X), f_{\boldsymbol{u}}^{\otimes q}(X), \ldots, f_{\boldsymbol{u}}^{\otimes q^s}(X)\big)$$

where $f_{\boldsymbol{u}}(X)$ is the message polynomial and Q is the interpolation polynomial constructed by the proposed list-decoding algorithm.

Lemma 2.5.3. *Suppose that the number of errors is bounded as in Lemma* 2.5.2. *Then the polynomial* $E(X)$ *is identically zero.*

Proof. Since the q-degree of $f_{\boldsymbol{u}}(X)$ is at most $k-1$, the q-degree of $Q_j\big(f_{\boldsymbol{u}}^{\otimes q^{j-1}}(X)\big)$ is at most

$$m-1-(k-1)q^{j-1}+(k-1)q^{j-1}=m-1$$

for $j=1,2,\ldots,s+1$. Also, $Q_0(X)$ has q-degree at most $m-1$. Therefore, the q-degree of $E(X)$ is at most $m-1$. By Lemma 2.5.1, $(\alpha, f(\alpha), f^q(\alpha), \ldots, f^{\otimes q^s}(\alpha))$ is contained in the linear span of the set of interpolation points $\mathcal{P}$. Also, raising to power q^h is a linear mapping with respect to the base field $\mathbb{F}_q$. Also, by using the fact that $f_{\boldsymbol{u}}(X)$ is over $\mathbb{F}_q$, we conclude that

$$E(\alpha^{q^h})=0,$$

for $h=0,1,\ldots,m-1$. Hence $E(X)$ has at least m roots which are linearly independent by Lemma 2.4.2. But the q-degree of $E(X)$ is at most $m-1$. Therefore, $E(X)\equiv 0$. ■
The next theorem provides a bound on the error-correction capability of our list-decoding algorithm.

Theorem 2.5.4. *If*

$$t<s+1-\left(\frac{q^{s+1}-1}{q-1}\right)\left(\frac{k-1}{m}\right),$$

then our list-decoding algorithm is correct; that is, it outputs a list of size at most q^s *of messages which includes the transmitted message.*

Proof. By Lemma 2.5.2 and Lemma 2.5.3, $f_{\boldsymbol{u}}(X)$ is in the output list. Also, by Theorem 2.3.2, there are at most $L=q^s$ solutions. ■

Now, we turn to compute the rate of the code with the aim of establishing the bound on the error-correction radius in terms of the rate. The ambient space $\mathcal{W}$ has dimension $N=m+1$. The rate R of the code is

$$R=\frac{\log_q(|\mathbb{C}|)}{nN}=\frac{k}{m+1}$$

The polynomials Q_i's must have non-negative q-degrees, i.e. $m-(k-1)q^s>0$. Therefore,

$$q^s<\frac{m}{k-1}\approx\frac{1}{R}$$

The bound on the error-correction capability of our list-decoding algorithm is given by Theorem 2.5.4. Approximating $(k-1)/n$ by R, we can express the bound as follows. Our list-decoding algorithm successfully recovers the transmitted message as long as

$$t < s + 1 - \left(\frac{q^{s+1}-1}{q-1}\right) R$$

2.6 Efficient Factorization in the Ring of Linearized Polynomials

In this section, we present the linearized Roth-Ruckenstein algorithm (LRR algorithm) which is used in the factorization step of all of our list-decoding algorithms. The LRR algorithm essentially solves equations over the ring of linearized polynomials in an efficient polynomial time.

Consider a polynomial $Q(X, Y)$, where Y is a variable in the ring $\mathscr{L}_q[X]$, of the form

$$Q(X, Y) = Q_0(X) + Q_1(X) \otimes Y + \cdots + Q_L(X) \otimes Y^{\otimes L} \tag{2.37}$$

where Q_i's are linearized polynomials over a finite extension of $\mathbb{F}_q$. The LRR algorithm finds all the roots $Y \in \mathscr{L}_q[X]$ with q-degree at most $k-1$, for some fixed $k \in \mathbb{N}$, such that $Q(X, Y)$ is identically zero.

We say that the polynomial $Q(X, Y)$ is divisible by X^{q^s} if all the Q_i's, for $i = 1, 2, \ldots, L$, are divisible by X^{q^s}. In this case, for each i, there is a linearized polynomial Q'_i such that $Q'_i(X)^{q^s} = Q_i(X)$. Then we define

$$Q_{\downarrow s}(X, Y) = Q'_0(X) + Q'_1(X) \otimes Y + \cdots + Q'_L(X) \otimes Y^{\otimes L}$$

Linearized Roth-Ruckenstein (LRR) algorithm

LRR $(Q(X, Y), k \in \mathbb{N}, \lambda \in \mathbb{N} \cup \{0\})$

Global variables:

set $A \subseteq \mathscr{L}_q[X]$,

polynomial $g(X) = \sum_{i=0}^{k-1} u_i X^{q^i} \in \mathscr{L}_q[X]$.

Call procedure initially with $Q(X, Y) \neq 0, k > 0, \lambda = 0$.

if$(\lambda == 0)$

$A \leftarrow \emptyset$;
$s \leftarrow$ largest integer such that $Q(X,Y)$ is divisible by X^{q^s}
$H(X,\gamma) \leftarrow \frac{1}{X} Q_{\downarrow s}(X, \gamma X)$;
$Z \leftarrow$ set of all roots of $H(0,\gamma)$ in $\mathbb{F}_q$;
for each $\gamma \in Z$ do {
$u_\lambda \leftarrow \gamma$;
if $(\lambda < k-1)$
LRR$(Q_{\downarrow s}(X, Y^q + \gamma X), k, \lambda + 1)$;
else
if $(Q(X, u_{k-1}X) == 0)$
$A \leftarrow A \cup \{g(X)\}$;
}

Lemma 2.6.1. *Let $Q(X,Y)$ be as defined in (2.37). Let*

$$f(X) = f_0 X + f_1 X^q + \cdots + f_{k-1} X^{q^{k-1}}$$

and

$$H(X,\gamma) = \frac{1}{X} Q(X, \gamma X)$$

Then the coefficient of X in $Q(X, f(X))$ is equal to $H(0, f_0)$.

Proof. Observe that the coefficient of X in $f^{\otimes i}(X)$ is equal to $f_0^i X$. Therefore, the coefficient of X in $Q(X, f(X))$ is equal to coefficient of X in

$$Q_0(X) + Q_1(f_0 X) + Q_2(f_0^2 X) + \cdots + Q_L(f_0^L X)$$

The latter is equal to $XH(X, f_0)$. Note that coefficient of X in $XH(X, f_0)$ is equal to the constant term in $H(X, f_0)$ which is indeed $H(0, f_0)$. ■
Notice that the level of recursion can not go beyond $k-1$. In fact, each sequence of recursions along a recursion descent is associated with a unique polynomial

$$f(X) = f_0 X + f_1 X^q + \ldots$$

which stands for the contents of the global polynomial $g(X)$ computed by that sequence. For $i = 0, 1, \ldots, k-1$, let $P_i(X,Y)$, $T_i(X,Y)$ and $H_i(X,\gamma)$ be the values of $Q(X,Y)$,

$Q_{\downarrow s}(X,Y)$ and $H(X,\gamma)$, respectively, during recursion level $\lambda = i$. It can be inductively observed that P_i and T_i are non-zero polynomials for $i = 0, 1, \ldots, k-1$. In fact, $P_0 = Q$ is assumed to be non-zero. Since P_i is non-zero, T_i is non-zero which implies that P_{i+1} is non-zero. Therefore, the parameter s is always well-defined.

At each recursion level i, $T_i(X,Y)$ is not divisible by X^q. Therefore, the coefficient of X in $T_i(X,\gamma X)$ is not zero. Then by Lemma 2.6.1, $H(0,\gamma)$ is not the all zero polynomial.

Lemma 2.6.2. *Let A be the set computed by LRR$(Q,k,0)$. Then every element of A is a root of Q.*

Proof. Let

$$f(X) = u_0X + u_1X^q + \cdots + u_{k-1}X^{q^{k-1}}$$

be an element of A. For $0 \leqslant i < k$, define the polynomial $\phi_i(X)$ by

$$\phi_i(X) = u_iX + u_{i+1}X^q + \cdots + u_{k-1}X^{q^{k-i-1}}$$

Since u_i's are elements of $\mathbb{F}_q$, $\phi_i = \phi_{i+1}^q + u_iX$. Let P_i and T_i be the values of Q and T during recursion level $\lambda = i$. We do a backward induction on $i = k-1, k-2, \ldots, 0$ to show that ϕ_i is a root of P_i. The base of induction is $i = k-1$. Note that $\phi_{k-1} = u_{k-1}X$ which is a root of P_{k-1} by the one before the last line of LRR procedure when $\lambda = k-1$. Now, suppose that ϕ_{i+1} is a root of P_{i+1}. Then we have

$$\begin{aligned} P_i(X,\phi_i) = T_i(X,\phi_i)^{q^s} &= T_i(X,\phi_{i+1}^q + u_iX)^{q^s} \\ &= P_{i+1}(X,\phi_{i+1})^{q^s} = 0 \end{aligned}$$

Therefore, ϕ_i is a root of P_i which completes the induction. In particular, for $i = 0$ we see that $f(X) = \phi_0(X)$ is a root of $P_0 = Q$. ■

Lemma 2.6.3. *Let*

$$f(X) = f_0X + f_1X^q + \cdots + f_{k-1}X^{q^{k-1}}$$

be a root of $Q(X,Y)$ in $\mathscr{L}_q[X]$. For $0 \leqslant i \leqslant k-1$, define $P_i(X,Y)$ and $T_i(X,Y)$ inductively by $P_0 = Q$ and

$$T_i(X,Y)^{q^{s_i}} = P_i(X,Y) \text{ and } P_{i+1}(X,Y) = T_i(X,Y^q + f_iX),$$

where s_i is the largest possible integer such that $P_i(X,Y)$ is divisible by $X^{q^{s_i}}$. Also, define

$$H_i(X,\gamma) = \frac{1}{X}T_i(X,\gamma X)$$

Then for $0 \leqslant i \leqslant k-1$,
i)The polynomial ϕ_i defined by

$$\phi_i = f_iX + f_{i+1}X^q + \cdots + f_{k-1}X^{k-1-i}$$

is a root of $P_i(X,Y)$.
ii)$H_i(0,f_i) = 0$

Proof. We prove part i) by induction on i. For $i = 0$, $\phi_0 = f$ is a root of $P_0 = Q$. Now, suppose that ϕ_i is a root of $P_i(X,Y)$. Since $\phi_i = \phi_{i+1}^q + f_iX$, $Y = \phi_{i+1}$ is a root of $P_i(X, Y^q + f_iX)$ and, hence, of $T_i(X, Y^q + f_iX) = P_{i+1}(X,Y)$. This completes the induction which proves part i).

Also, note that

$$P_i(X,\phi_i(X)) = T_i(X,\phi_i(X))^{q^{s_i}} = 0 \;\Rightarrow\; T_i(X,\phi_i(X)) = 0$$

By Lemma 2.6.1, the coefficient of X in $T_i(X,\phi_i(X))$ is equal to $H_i(0,f_i)$ which has to be zero. This proves part ii). ■

Lemma 2.6.4. *Let A be the set computed by* LRR$(Q,k,0)$. *Then every root of Q in $\mathscr{L}_q[X]$ is contained in A.*

Proof. Let $f(X) = f_0X + f_1X^q + \cdots + f_{k-1}X^{q^{k-1}}$ be a root of $Q(X,Y)$ in $\mathscr{L}_q[X]$. Define P_i, T_i and H_i as in Lemma 2.6.3. We prove by induction on i for $i = 0,1,\ldots,k-1$ that there is a recursion descent in LRR such that recursion level i is called with the parameters (P_i,k,i).

The base of induction is $i = 0$ which is obvious. Suppose that it is true for some i. Then by Lemma 2.6.3, $H_i(0,f_i) = 0$ and therefore, $\gamma = f_i$ is one of the roots. If $i < k-1$, then for $\gamma = f_i$ the recursive call is made with parameters

$$(T_i(X,Y^q + f_iX), k, \lambda+1) = (P_{i+1}(X,Y), k, i+1)$$

If $i = k - 1$, then by Lemma 2.6.3, $P_{k-1}(X, f_{k-1}) = 0$ which means that f is inserted into A. ■

Theorem 2.6.5. *The LRR algorithm is correct that is, for any polynomial Q as defined in (2.37), the call LRR$(Q, k, 0)$ computes a set A which consists of all the roots of Q in $\mathscr{L}_q[X]$.*

The proof follows from Lemma 2.6.2 and Lemma 2.6.4.

2.7 Discussion

In this chapter, we have considered the problem of list-decoding of subspace codes proposed for error correction in random linear network coding. To this end, we modified and generalized the original Koetter-Kschischang codes in various ways. In fact, we constructed a new subspace code and proposed a list-decoding algorithm that enables error-correction beyond the unique decoding bound. Interestingly, for a fixed code dimension, we can actually correct any number of errors provided that the list size is sufficiently large and the rate is small enough. In this case, the worst-case list-size turns out to be proportional to the number of errors.

Nevertheless, we are able to achieve a better error-correction radius than Koetter-Kschischang codes only at low rates. Then one question that arises is how to extend this work in order to enable list-decoding at higher rates as well. We may take advantage of the analogy between this work and the Sudan list-decoding algorithm of RS codes. When Sudan introduced his list-decoding algorithm of Reed-Solomon codes, there was a similar problem. Later Guruswami and Sudan proposed a new method where they enforced multiple roots for the interpolation polynomial which led to a significant improvement upon Sudan's first result. Therefore, it is natural to look for an analogous technique in the ring of linearized polynomials. However, there is no clear notion of multiple roots for linearized polynomials in the literature. In fact, one has to introduce multiplicity in the ring of linearized polynomials in such a way that list-decoding at higher rates is enabled. This will be addressed in the next chapter.

As mentioned, in order to do list-decoding, we modify and generalize the Koetter-Kschischang codes in many ways. Then the natural question is the following: is there a way to list-decode the Koetter-Kschischang codes without any modification at the transmitter side? In Section 2.5, we proposed a solution for this question only when the code is one dimensional. The question remains open in the general case.

Acknowledgments

The results of this chapter was first presented in part at ISIT 2010 in Austin and appeared in the proceedings as: H. Mahdavifar and A.Vardy, "Algebraic List-decoding on the Operator Channel", *Proceedings of the 2010 IEEE International Symposium on Information Theory*, pp. 1193-1197, Austin, Texas, June 2010. A full version of it was submitted to IEEE transactions on information theory and was recently accepted for publication. It is available on arxiv as: H. Mahdavifar and A. Vardy, "Algebraic List-decoding of Subspace Codes", available at http://arxiv.org/pdf/1202.0338.pdf.

Bibliography

[1] R. Ahlswede, N. Cai, Sh.Y.R. Li, and R.W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, pp. 1204–1216, July 2000.

[2] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. 41-st Annual Allerton Conference on Communication, Control and Computing,* Monticello, IL, October 2003.

[3] T. Etzion and A. Vardy, "Error-correcting codes in projective space," *IEEE Transactions on Information Theory*, vol. 57, pp. 1165–1173, February 2011.

[4] T. Ho, M. Médard, J. Shi, M. Effros, and D. Karger, "On randomized network coding," in *Proc. 41-st Annual Allerton Conference on Communications, Control, and Computing*, Monticello, IL., October 2003.

[5] T. Ho, M. Medard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, pp. 4413–4430, October 2006.

[6] R. Koetter and F.R. Kschischang, "Coding for errors and erasures in random network coding," *IEEE Transactions on Information Theory*, vol. 54, pp. 3579–3591, August 2008.

[7] Sh.Y.R. Li, R.W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol 49, pp. 371–381, February 2003.

[8] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam: North Holland, 1977.

[9] R.M. Roth and G. Ruckenstein, "Efficient decoding of Reed-Solomon codes beyond half the minimum distance," *IEEE Transactions on Information Theory*, vol. 46, pp. 246–257, January 2000.

[10] M. Sudan, "Decoding of Reed-Solomon codes beyond the error-correction bound," *J. Complexity,* vol. 12, pp. 180–193, March 1997.

[11] H. Xie, Z. Yan, and B.W. Suter, "General linearized polynomial interpolation and its applications," in *Proc. International Symposium on Network Coding,* Beijing, China, July 2011.

Chapter 3

Algebraic List-Decoding with Multiplicities

3.1 Introduction

In Chapter 2, we reviewed subspace codes that have been recently introduced in order to enable reliable communication of messages in random network coding [2]. In [2] Koetter and Kschischang formulated a theory in the context of a *non-coherent* transmission model for random network coding wherein neither the transmitter nor the receiver are assumed to have any knowledge about the underlying network topology and the particular linear network coding operations performed at each network node. They show that subspace codes capture the effects both of errors, i.e. erroneously received packets, and of erasures, i.e., insufficiently many received packets. Indeed the only thing that is preserved is *the subspace spanned by the set of packets* injected by the transmitter into the network. Information can be conveyed via the choice of that subspace. Koetter-Kschischang construction of subspace codes, originally called Reed-Solomon-like codes in [2], is analogous to Reed-Solomon codes in classical block codes wherein symbols are replaced by vectors, regular polynomials with *linearized polynomials*, and sequences of symbols with a $\mathbb{F}_q$-linear span of the corresponding vectors.

In the previous chapter, we constructed a new family of subspace codes which are list-decodable. Motivated by Koetter-Kschischang subspace codes, we modified and

generalized their construction in many important ways in order to enable list-decoding. Using algebraic list-decoding, we are able to achieve a better tradeoff than Koetter-Kschischang codes between *rate* and *error-correction radius*, at low rates. In a sense, our algorithm is regarded as analogous to the Sudan list-decoding algorithm of Reed-Solomon codes [7]. The first decoding step in Sudan's algorithm, called the interpolation step, computes from the received word a certain bivariate polynomial $Q(X, Y)$ over the ground field, $\mathbb{F}_q$, of the code which passes through certain interpolation points. By looking at $Q(X, Y)$ as a univariate polynomial with indeterminate Y over the ring $\mathbb{F}_q[X]$, the second decoding step, called the factorization step, computes the roots of $Q\big(X, f(X)\big)$ in $\mathbb{F}_q[X]$. The decoder outputs coefficients of each root as a vector. In [1], Guruswami and Sudan improved on this by introducing multiplicities in the interpolation step of Sudan's algorithm. The Guruswami-Sudan idea is basically to force the interpolation polynomial $Q(X, Y)$ to pass through interpolation points multiple times. This will in turn guarantee multiple roots in the factorization step of the decoding algorithm which leads to better error-correction radius. This result showed that list-decoding can be effectively used to go beyond the unique decoding radius for every rate.

In this chapter, we consider the problem of list-decoding of subspace codes with multiplicities. Motivated by the Guruswami-Sudan list-decoding algorithm of Reed-Solomon codes, we aim to achieve a better error-correction capability at higher rates by enforcing multiple roots for the linearized interpolation polynomial. To the best of our knowledge, however, no explicit definition of multiple roots for linearized polynomials exists in the literature. Our result in this chapter is based upon the following key idea. The interpolation polynomial is mapped through an isomorphism to the ring of polynomials, multiple roots are enforced there and then it is lifted back to the ring of linearized polynomials. To this end, we first define a bijective mapping between the ring of linearized polynomials over $\mathbb{F}_q$ and the ring of polynomials over $\mathbb{F}_q$. We prove that this is in fact an isomorphism. The derivative for univariate linearized polynomials is defined based on this isomorphism . We also generalize this definition to multivariate linearized polynomials in such a way that by mapping them to the ring of polynomials, we get the Hasse derivative of the corresponding bivariate polynomial.

The rest of this chapter is organized as follows. In Section 3.2, we give an

overview of the Guruswami-Sudan list-decoding algorithm of Reed-Solomon Codes. We also briefly review subspace codes, linearized polynomials, Koetter-Kschischang codes and our results in the previous chapter with the aim of more firmly establishing the results of this chapter. In Section 3.3, we establish the terminology for derivatives and partial derivatives in the ring of linearized polynomials which leads to a notion of multiplicity in the ring of linearized polynomials. Section 3.4 is devoted to the special case of list-decoding with multiplicity 2. We present a list-L decoding algorithm with multiplicity 2 and prove that it outputs a list of size at most L containing the transmitted message provided that

$$\tau \leqslant \frac{2(L+1)}{3} - 1 - \frac{L(L+1)}{6}R^*$$

where τ is the error-correction radius and R^* is the *packet rate* of the code introduced in Section 2.4.3. Loosely speaking, the packet rate of a subspace code is the ratio of the number of information packets to the number of encoded packets injected into the network. In this chapter, we use this notion of rate in order to express our results in a more convenient way. In Section 3.5, we present a list-decoding algorithm in the general case with arbitrary multiplicity r. We guarantee that the injected message into the network will be recovered at the receiver as long as

$$\tau \leqslant \frac{2(L+1)}{r+1} - 1 - \frac{L(L+1)}{r(r+1)}R^* \tag{3.1}$$

It should be noticed that the choice of r is independent of the code construction. Therefore, r can be chosen at the decoder in such a way that the decoding radius is maximized. The value of r which maximizes the bound in (3.1) is equal to $\lceil LR^* \rceil$. Plugging in this value into (3.1) we get a piecewise linear function for the bound versus the packet rate R^*. This is shown in Figure 3.1 for $L = 5$ for Algorithm B which exploits multiplicities in comparison with the list-decoding algorithm without multiplicity explained in the previous chapter and the Koetter-Kschischang codes.

The limit of decoding radius, as L tends to infinity, is equal to $\frac{1}{R^*} - 1$ for packet rates R^* between 0 and 1.

We conclude the chapter in Section 3.6 with a comparison between results of this chapter and the previous chapter. A list of open problems and some discussions is also compiled in Section 3.6.

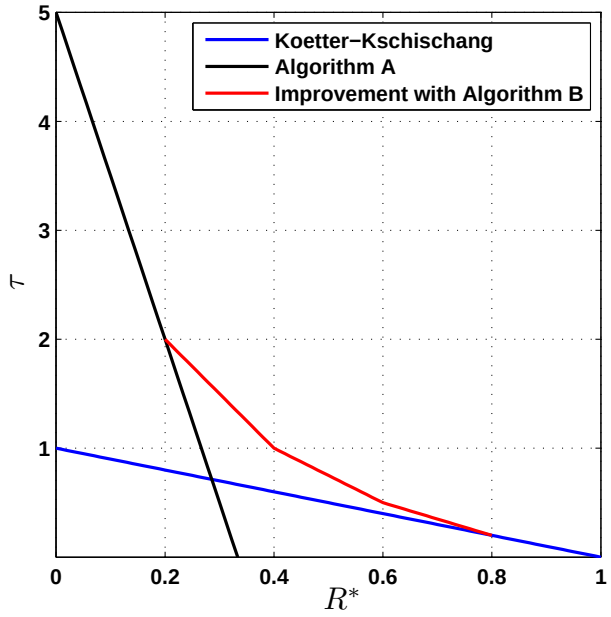


Figure 3.1: Improvement on error-correction radius upon previous works by using multiplicity for list size $L = 3$

3.2 Preliminaries and Prior Work

In this section, we first give an overview of the Guruswami-Sudan list-decoding algorithm of Reed-Solomon codes which provided the motivation for the work presented in this chapter [1]. Following [2] we review the operator channel model, the ring of linearized polynomials and the Koetter-Kshischang codes. Then we briefly recap the results of [4,5] that was presented in details in the previous chapter. In [4,5] we suitably modified and extended the Koetter-Kschischang construction in many important respects in order to facilitate list-decoding.

3.2.1 Guruswami-Sudan List-Decoding Algorithm

Sudan list-decoding algorithm [7], as reviewed in Section 2.3, can be regarded as an algorithm to solve the following curve-fitting problem: Given n pairs of elements $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n))\}$, where x_i's and y_i's are elements of a finite field $\mathbb{F}_q$, a degree parameter k and a an error parameter e, find all polynomials $p(X) \in \mathbb{F}_q[X]$ such

that $p(x_i) = y_i$ for at least $n - e$ values of $i \in \{1, 2, \ldots, n\}$. The Guruswami-Sudan list-decoding algorithm is an extension of the Sudan algorithm in that the properties of the *singularities* of these curves are used.

First, we need to recall the definition of the Hasse derivative and of multiplicity for bivariate polynomials. Let $Q(X, Y) = \sum_{i,j \geqslant 0} q_{i,j} X^i Y^j$ be a polynomial in $\mathbb{F}_q[X, Y]$. Let $\mathbb{N}$ denote the natural numbers including 0. For any $a, b \in \mathbb{N}$, the (a, b)-th *Hasse derivative* of $Q(X, Y)$, denoted by $\mathcal{D}_{(a,b)}\big[Q(X, Y)\big]$, is defined as

$$\mathcal{D}_{(a,b)}\big[Q(X, Y)\big] = \sum_{i \geqslant a, j \geqslant b} \binom{i}{a} \binom{j}{b} X^{i-a} Y^{j-b} \tag{3.2}$$

The polynomial $Q(X, Y)$ is said to have a *zero of multiplicity* m at a point $(x_0, y_0) \in \mathbb{F}_q \times \mathbb{F}_q$ if

$$\mathcal{D}_{a,b}\big[Q(X, Y)\big]\Big|_{(x_0, y_0)} = 0$$

for all $a, b \in \mathbb{N}$ with $a + b < m$.

Given $f(X) \in \mathbb{F}_q[X]$ and $Q(X, Y) \in \mathbb{F}_q[X, Y]$, let x_0 and y_0 be elements of $\mathbb{F}_q$ such that $f(x_0) = y_0$ and $Q(X, Y)$ has a zero of multiplicity m at (x_0, y_0). Then it can be proved that

$$(x - x_0)^m \;\big|\; Q\big(X, f(X)\big)$$

The generalized Reed-Solomon code is constructed as follows. Let k be the number of information symbols and n be the length of the code with $k \leqslant n \leqslant q - 1$. Fix n distinct elements of $\mathbb{F}_q$, such as $\alpha_1, \alpha_2, \ldots, \alpha_n$, as the set of evaluation points. The message is a vector $\boldsymbol{u} = (u_0, u_1, \ldots, u_{k-1})$ which consists of k information symbols over $\mathbb{F}_q$. Then the corresponding codeword is $(f_{\boldsymbol{u}}(\alpha_1), f_{\boldsymbol{u}}(\alpha_2), \ldots, f_{\boldsymbol{u}}(\alpha_n))$, where $f_{\boldsymbol{u}}(X) = \sum_{i=0}^{k-1} u_i X^i$ is the message polynomial. This codeword is transmitted through the channel and the channel output $(y_1, y_2, \ldots, y_n)$ is given. The first step of the Guruswami-Sudan list-decoding algorithm is the interpolation step similar to Sudan list-decoding algorithm except that the interpolation points (α_i, y_i) are forced to be of multiplicity m. The decoder constructs the bivariate interpolation polynomial

$$Q(X, Y) = Q_0(X) + Q_1(X)Y + \cdots + Q_L(X)Y^L$$

with $(1, k-1)$-weighted degree at most $\omega - 1$, such that (α_i, y_i) is a zero of multiplicity m for $Q(X, Y)$ for all i. The parameter ω will be specified later and L is the maximum

list-size at the output of the decoder; that is, it is guaranteed that the size of the output list is at most L. The multiplicity parameter m is completely arbitrary and can be set to any value as long as we can guarantee the existence of a non-zero solution for $Q(X, Y)$.

The only difference between the Guruswami-Sudan list-decoding algorithm and the Sudan list-decoding algorithm is in the interpolation step where we use the notion of multiplicity. As a result, the parameter ω will be set differently. In fact, the Guruswami-Sudan list-decoding algorithm with $m = 1$ reduces to the Sudan list-decoding algorithm.

The next step in this algorithm is the factorization step which is done the same as in the Sudan list-decoding algorithm. The decoder finds all the factors of $Q(X, Y)$ of the form $Y - f(X)$. It can be proved that there are at most L of them. This step can be done efficiently in polynomial time using the Roth-Ruckenstein algorithm [6]. If not too many errors have occurred, $Q(X, f_{\boldsymbol{u}}(X)) \equiv 0$ which guarantees the recovery of the message at the output of the decoder.

By enforcing multiple zeros in the interpolation step, we basically increase the number of constraints by a factor of $\binom{m+1}{2}$. On the other hand, each correct received symbol $f_{\boldsymbol{u}}(\alpha_i)$ leads to a root with multiplicity m for the $Q(X, f_{\boldsymbol{u}}(X)) \equiv 0$ that is,

$$(x - \alpha_i)^m \mid Q\big(X, f_{\boldsymbol{u}}(X)\big)$$

In fact, this trade-off between the number of constraints and the number of guaranteed roots for the polynomial $Q(X, f_{\boldsymbol{u}}(X))$ leads to a higher error-correction radius.

Next, we analyze the Guruswami-Sudan list-decoding algorithm in more detail. Similar to the Sudan's list-decoding algorithm, the weighted degree constraint on the interpolation polynomial $Q(X, Y)$ is such that the degree of $Q\big(X, f_{\boldsymbol{u}}(X)\big)$ is at most $\omega - 1$. The parameter ω is set in such a way that existence of a non-trivial solution for $Q(X, Y)$ is guaranteed; that is, the total number of available monomials is larger than than the number of interpolation equations. For each interpolation point (α_i, y_i), we require $\binom{m+1}{2}$ equations. Hence, the total number of equations is $n\binom{m+1}{2}$. The total number of variables, number of monomials $X^i Y^j$ with $i + (k-1)j < \omega$, is

$$\omega + \omega - (k-1) + \omega - 2(k-1) + \cdots + \omega - L(k-1) = (L+1)\omega - \binom{L+1}{2}(k-1)$$

Therefore, the first condition to be satisfied is

$$(L+1)\omega - \binom{L+1}{2}(k-1) > n\binom{m+1}{2} \tag{3.3}$$

$Q\big(X, f_{\boldsymbol{u}}(X)\big) \in \mathbb{F}_q[X]$ is a univariate polynomial with a degree of at most $(\omega - 1)$. Any correct received symbol $y_i = f_{\boldsymbol{u}}(\alpha_i)$ guarantees one particular root α_i for $Q(X, f_{\boldsymbol{u}}(X))$ with multiplicity m. Let t be the maximum number of errors we wish to correct. Hence, $Q(X, f_{\boldsymbol{u}}(X))$ is guaranteed to have at least $m(n-t)$ roots i.e. $n-t$ distinct roots each with multiplicity at least m. If the condition

$$m(n-t) \geqslant \omega \tag{3.4}$$

holds, then $Q\big(X, f_{\boldsymbol{u}}(X)\big)$ must be the zero polynomial. As a result, $f_{\boldsymbol{u}}(X)$ can be successfully recovered by finding all the possible factors $Y - f(X)$ of $Q(X, Y)$.

(3.3) and (3.4) can be combined in order to get a bound on the error-correction radius of the Guruswami-Sudan list-L decoder. We substitute ω from (3.4) into (3.3) to get the following bound on the error-correction radius:

$$\frac{t}{n} < 1 - \frac{m+1}{2(L+1)} - \frac{L}{2m}\Big(\frac{k-1}{n}\Big)$$

t/n is the normalized error-correction radius τ and $(k-1)/n$ is approximately the rate of the code $R = k/n$. The final condition can be expressed as

$$\tau < 1 - \frac{m+1}{2(L+1)} - \frac{L}{2m}R \tag{3.5}$$

On the other hand, if this is satisfied, one can find a suitable value for ω in order to successfully perform the Sudan list-L decoding algorithm.

Next, we give a rough analysis on how to set the optimum choice for the multiplicity parameter m in terms of list-size L and rate R. It then leads to the famous $1 - \sqrt{R}$ bound on the error-correction radius of Guruswami-Sudan list-decoding algorithm. As mentioned before, the parameter m is completely arbitrary and can be set to be anything at the decoder. However, the best choice for m is to maximize the bound provided in (3.5) given fixed values for L and R. Finally, the bound in (3.5) becomes a piecewise linear function of the rate R for any list-size L. Consider the regime where L is large enough such that we can approximate $(m+1)/(L+1)$ by m/L. Let γ denote the ratio m/L. Then the bound in (3.5) can be expressed as $1 - \gamma/2 - R/2\gamma$. Hence, the whole thing reduces to minimizing $\gamma + R/\gamma$ which happens for $\gamma = \sqrt{R}$. Then the error-correction radius bound for this optimum choice of γ is $1 - \sqrt{R}$. This happens for $\gamma = \sqrt{R}$ or equivalently $m = L\sqrt{R}$. This analysis is for large enough values of L. In fact, the $1 - \sqrt{R}$ bound on error-correction radius can be achieved as L tends to infinity.

3.2.2 Prior Work

Let $\mathcal{W}$ be a fixed N-dimensional vector space over $\mathbb{F}_q$ and $\mathscr{P}_q(\mathcal{W})$ denote the set of all subspaces of $\mathcal{W}$ which is often called the projective geometry of $\mathcal{W}$. For any $V \in \mathcal{G}(W)$, the dimension of V is denoted by $\dim(V)$. For any $A, B \in \mathcal{G}(W)$, the distance between A and B is defined as follows:

$$d(A,B) \stackrel{\text{def}}{=} \dim(A+B) - \dim(A \cap B)$$

$\mathscr{P}_q(\mathcal{W})$ is indeed a metric space under this metric. Furthermore, let $\mathcal{G}_q(\mathcal{W}, n)$ denote the set of all n-dimensional subspaces of $\mathcal{W}$. Following [2], we defined operator channel and subspace codes in Chapter 1 in details. An operator channel $\mathscr{C}$ associated with the ambient space $\mathcal{W}$ is a channel with input and output alphabet $\mathscr{P}_q(\mathcal{W})$. The input to $\mathscr{C}$ is a subspace $V \in \mathscr{P}_q(\mathcal{W})$ and the output of $\mathscr{C}$ is another subspace $U \in \mathscr{P}_q(\mathcal{W})$. Deletion of vectors from V as it is transmitted through $\mathscr{C}$ is called erasures and addition of linearly independent vectors to V is called errors. The output U is the input V which is possibly corrupted by errors and erasures.

A code $\mathbb{C}$ for an operator channel $\mathscr{C}$ with ambient space $\mathcal{W}$ is a non-empty subset of $\mathscr{P}_q(\mathcal{W})$. A codeword is an element of $\mathbb{C}$ which is in fact a subspace of $\mathcal{W}$. The rate of the code $\mathbb{C}$ is defined as follows. Suppose that the dimension of any $V \in \mathbb{C}$ is at most n. Then

$$R \stackrel{\text{def}}{=} \frac{\log_q |\mathcal{C}|}{nN} \tag{3.6}$$

A polynomial over some extension field $\mathbb{F}_{q^m}$ of $\mathbb{F}_q$ is called $\mathbb{F}_q$-linearized if it has the following form:

$$f(X) = \sum_{i=0}^{s} a_i X^{q^i},$$

where $a_i \in \mathbb{F}_{q^m}$, for $i = 0, 1, \ldots, s$. Assuming that $a_s \neq 0$ we say that the polynomial $f(X)$ has q-degree s which means that its actual degree is q^s. When q is fixed under discussion, we will let $X^{[i]}$ denote X^{q^i}. A linearized polynomial with coefficients from $\mathbb{F}_{q^m}$ act as a linear map over any extension of $\mathbb{F}_{q^m}$, with respect to $\mathbb{F}_q$. In other words, for any α_1 and α_2 in any extension field of $\mathbb{F}_{q^m}$ and any $\lambda_1, \lambda_2 \in \mathbb{F}_q$,

$$f(\lambda_1\alpha_1 + \lambda_2\alpha_2) = \lambda_1 f(\alpha_1) + \lambda_2 f(\alpha_2)$$

The set of linearized polynomials over $\mathbb{F}_{q^m}$ forms a non-commutative ring with identity under addition $+$ and composition $\otimes$, where $f_1(X) \otimes f_2(X)$ is defined to be the composition $f_1\big(f_2(X)\big)$ which is always a linearized polynomial.The ring of linearized polynomials over $\mathbb{F}_{q^m}$ is denoted by $\mathscr{L}_{q^m}[X]$.

Koetter-Kschischang subspace codes can be regarded as analogous to Reed-Solomon codes wherein symbols are replaced by vectors, polynomials with linearized polynomials and sequences of symbols with $\mathbb{F}_q$-linear span of the corresponding vectors. A set of n evaluation points $A = \{\alpha_1, \ldots, \alpha_n\} \subset \mathbb{F}_{q^m}$ is selected. α_i's are chosen to be linearly independent. In fact, since we will evaluate a linearized polynomial over the set A, evaluation over a point that is already contained in the linear span of other points is redundant information. Therefore, we require the elements of A to be linearly independent. Given the message symbols $u_0, u_1, \ldots, u_{k-1}$, we construct the linearized message polynomial $f_{\boldsymbol{u}}(X) = \sum_{i=0}^{k-1} u_i X^{[i]}$. The codeword V is the $\mathbb{F}_q$-linear span of the set $\{(\alpha_i, f(\alpha_i)) : 1 \leqslant i \leqslant n\}$. All the codewords are n-dimensional subspaces of the $(n+m)$-dimensional ambient space $\mathcal{W} = \langle A \rangle \oplus \mathbb{F}_{q^m}$.

Suppose that V is transmitted through the operator channel and another subspace U of $\mathcal{W}$ of dimension d is received. At the decoder, a basis $(x_i, y_i), i = 1, 2, \ldots, d$ for U is selected. Then the decoder constructs a non-zero bivariate linearized polynomial $Q(X, Y)$ of the form

$$Q(X, Y) = Q_0(X) + Q_1(Y),$$

where Q_0 and Q_1 are linearized polynomials over $\mathbb{F}_{q^m}$, Q_0 has q-degree of at most $\omega - 1$ and Q_1 has q-degree of at most $\omega - k$ subject to the constraint that

$$Q(x_i, y_i) = 0 \text{ for } i = 1, 2, \ldots, r$$

Then $f_{\boldsymbol{u}}(X)$ is the unique solution to the equation

$$Q\big(X, f(X)\big) = 0$$

Suppose that ρ and t are the number of erasures and errors, respectively. Koetter and Kschischang [2] prove that if $\rho + t < n - k + 1$, then the decoding algorithm successfully recovers the transmitted message polynomial $f_{\boldsymbol{u}}(X)$ by choosing $\omega = \left\lceil \frac{d+k}{2} \right\rceil$. Thus the bound on error-correction capability of Koetter-Kschischang codes is

$$\tau < \frac{n-k+1}{n} = 1 - \frac{k-1}{n} \approx 1 - \frac{1}{n}(1 + \frac{n}{m})R^* \tag{3.7}$$

where $\tau = (\rho + t)/n$ is the normalized error-correction radius.

In Chapter 2, we presented our list-decoding algorithm of subspace codes which requires modification and generalization of the Koetter-Kschischang codes in many important ways. Our work, also presented in [4,5], essentially leads to a new construction of subspace codes which is efficiently list-decodable. We now briefly recap the results discussed in the foregoing chapter.

Recall from [3, Ch. 4.9] that any finite extension $\mathbb{F}_{q^n}$ of $\mathbb{F}_q$ contains a primitive element γ such that $\gamma, \gamma^q, \ldots, \gamma^{q^{n-1}}$ forms a basis for $\mathbb{F}_{q^n}$ as a vector space over $\mathbb{F}_q$. This is called a normal basis for $\mathbb{F}_{q^n}$. For the purpose of this chapter, we consider the construction discussed in Chapter 2, only for the special case where $m = 1$. We assume that $q - 1$ is divisible by n. Then $x^n - 1 = 0$ has n distinct solutions in $\mathbb{F}_q$. Let $e_1 = 1, e_2, e_3, \ldots, e_n$ be these solutions. Let $\mathbb{F} = \mathbb{F}_{q^n}$ and γ be a generator of a normal basis for $\mathbb{F}$. Then define

$$\alpha_i = \gamma + e_i^{-1}\gamma^q + e_i^{-2}\gamma^{q^2} + \cdots + e_i^{-(n-1)}\gamma^{q^{n-1}} \tag{3.8}$$

for $i = 1, 2, \ldots, n$. In fact, this matches the definition of α_i's in (2.22) for the special case $m = 1$.

Next, we briefly review the encoding and decoding of list-decodable subspace codes that we proposed in Chapter 2. We fix the following parameters: the number of information symbols k, the dimension of code n and the list size L. The ambient space $\mathcal{W}$ is an $Ln + n$-dimensional vector space over $\mathbb{F}_q$. We require that $k \leqslant n$ and $q - 1$ is divisible by n.

Encoding Algorithm:

Formally, the encoder is a function $\mathcal{E} : {\mathbb{F}_q}^k \to \mathcal{G}(W, n)$. It accepts as input a message $\boldsymbol{u} = (u_0, u_1, \ldots, u_{k-1}) \in {\mathbb{F}_q}^k$. The corresponding message polynomial is $f_{\boldsymbol{u}}(X) = \sum_{i=0}^{k-1} u_i X^{[i]}$. For $i = 1, 2, \ldots, n$, consider α_i defined in (3.8). The encoder constructs vectors $v_i \in W$ as follows. For $i = 1, 2, \ldots, n$,

$$v_i = \left(\alpha_i, f_{\boldsymbol{u}}(\alpha_i), f_{\boldsymbol{u}}^{\otimes 2}(\alpha_i), \ldots, f_{\boldsymbol{u}}^{\otimes L}(\alpha_i)\right)$$

The encoder then outputs n-dimensional vector space V spanned by $v_1, v_2, \ldots, v_n$.

The first coordinate of each vector of the codeword V belongs to the vector space spanned by $\alpha_1, \alpha_2, \ldots, \alpha_n$. All the other L coordinates are elements of $\mathbb{F}_{q^n}$. Therefore,

in this construction, the ambient space $\mathcal{W}$ is

$$\langle \alpha_1, \alpha_2, \ldots, \alpha_n \rangle \oplus \underbrace{\mathbb{F}_{q^n} \oplus \cdots \oplus \mathbb{F}_{q^n}}_{L \text{ times}} \tag{3.9}$$

Its dimension, as mentioned before, is $Ln + n$. The code $\mathbb{C}_q(k, n, 1, L)$, as given in Definition 2.4.3, is the collection of all possible codewords V generated by this encoding algorithm. All the codewords are n-dimensional subspaces of the $Ln + n$-dimensional ambient space $\mathcal{W}$.

Remark. We represent each element of the ambient space $\mathcal{W}$ as a vector with $L + 1$ coordinates such as $(x, y_1, y_2, \ldots, y_L)$, where $x \in \langle \alpha_1, \alpha_2, \ldots, \alpha_n \rangle$ and $y_i \in \mathbb{F}_{q^n}$ for $i = 1, 2, \ldots, L$. □

Suppose that a codeword $V \in \mathbb{C}_q(k, n, 1, L)$ is transmitted through the operator channel and the decoder receives a vector space $U \in \mathcal{G}(W)$ with dimension d.

List-decoding Algorithm A:

Find a basis

$$\Big\{(x_i, y_{i,1}, y_{i,2}, \ldots, y_{i,L}) : i = 1, 2, \ldots, d\Big\}$$

for U. Then the set of interpolation points $\mathcal{P}$ is the following:

$$\mathcal{P} \;=\; \Big\{(x_i^{q^h}, y_{i,1}^{q^h}, \ldots, y_{i,L}^{q^h}) : i \in [d], h = 0, 1, \ldots, m-1\Big\}$$

We use the notation $[s]$ to denote the set of positive integers less than or equal to s.

Construct a non-zero multivariate linearized polynomial $Q(X, Y_1, Y_2, \ldots\ , Y_L)$ of the form

$$Q_0(X) + Q_1(Y_1) + Q_2(Y_2) + \cdots + Q_L(Y_L)$$

with each Q_i having q-degree of at most $n - (k-1)i - 1$, for $i = 0, 1, \ldots, L$, subject to the constraint that

$$Q(x, y_1, y_2, \ldots, y_L) \;=\; 0$$

for any $(x, y_1, y_2, \ldots, y_L) \in \mathcal{P}$. Then find all the roots of the following equation in $\mathscr{L}_q[X]$ using LRR algorithm explained in Section 2.6:

$$Q\big(X, f(X), f^{\otimes 2}(X), \ldots, f^{\otimes L}(X)\big) = 0$$

By Theorem 2.3.2 there are at most L solutions for $f(X) \in \mathscr{L}_q[X]$. Each solution corresponds to one possible output message.

Suppose that the dimension of received vector space U is equal to $d = n - \rho + t$, where ρ is the dimension of erasure and t is the dimension of error. We proved in Section 2.4 that if

$$L\rho + t < nL - \frac{1}{2}L(L+1)(k-1)$$

then the list-decoding algorithm A is correct that is, it outputs a list of size at most L which includes the transmitted message $\boldsymbol{u}$. Notice that there is a parameter m in the original equation in (2.29). However, we restrict our attention to the case $m = 1$ in this chapter. Then the bound on the error-correction radius $\tau = (L\rho + t)/n$, where the number of erasures is weighted by L, of this list-decoding algorithm in terms of list size L and packet rate R^* is given by

$$\tau < L - \frac{1}{2}L(L+1)R^* \tag{3.10}$$

3.3 Multiplicity in the ring of linearized polynomials

In this section, we establish the notion of multiplicity for linearized polynomials. As mentioned in Section 3.1, this work is motivated by Guruswami-Sudan list-decoding algorithm of Reed-Solomon codes. Their idea is to enforce the interpolation polynomial to go through the same set of roots as in Sudan algorithm but with some multiplicity. In fact, the Guruswami-Sudan algorithm with multiplicity one reduces to the Sudan algorithm. Enforcing multiple roots imposes more constraints on the interpolation polynomial. In return one get multiple zeros corresponding to each correct symbol of the received vector rather than having only one in the Sudan algorithm. The trade off is such that at the end, one can achieve better decoding radius using Guruswami-Sudan list-decoding algorithm compared to the Sudan algorithm.

Motivated by the Guruswami-Sudan list-decoding algorithm of Reed-Solomon codes, we aim to enforce multiple roots for the interpolation polynomial in the list-decoding algorithm A discussed in the foregoing section. To the best of our knowledge, however, no explicit definition of multiplicity for linearized polynomials exists in the literature. Therefore, we define an isomorphism between the ring of linearized polynomials over $\mathbb{F}_q$ and the ring of polynomials over $\mathbb{F}_q$. Then the idea is to map the resulted interpolation polynomial into its image in the ring of polynomials, enforce multiple roots

there and then lift it back to the ring of linearized polynomials. Based on this mapping, we define a derivative for univariate linearized polynomials and an analogous one to the Hasse derivative for multivariate linearized polynomials.

The mapping $\mathcal{H}$ from the ring of linearized polynomials to the ring of polynomials is simply defined as follows:

Definition 3.3.1. *For any linearized polynomial*

$$f(X) = f_0X + f_1X^q + \cdots + f_sX^{q^s}$$

We define

$$\mathcal{H}\big(f(X)\big) \stackrel{\text{def}}{=} f_0 + f_1X + \cdots + f_sX^s$$

It is clear that $\mathcal{H}$, restricted to $\mathscr{L}_q[X]$, is a bijective map between $\mathscr{L}_q[X]$ and $\mathbb{F}_q[X]$. Moreover, it is indeed an isomorphism i.e. it preserves the ring structure of $\mathscr{L}_q[X]$. This is shown in the next lemma.

Lemma 3.3.2. *The mapping $\mathcal{H}\colon \mathscr{L}_q[X] \to \mathbb{F}_q[X]$ is a ring isomorphism i.e. for any two linearized polynomials $f, g \in \mathscr{L}_q[X]$*

$$\mathcal{H}\big(f(X) + g(X)\big) = \mathcal{H}\big(f(X)\big) + \mathcal{H}\big(g(X)\big)$$

$$\mathcal{H}\big(f(X) \otimes g(X)\big) = \mathcal{H}\big(f(X)\big)\mathcal{H}\big(g(X)\big)$$

Proof. It is clear that $\mathcal{H}$ preserves the addition. For the second part, Let $f(X) = \sum_{i\geqslant 0} f_iX^{[i]}$ and $g(X) = \sum_{j\geqslant 0} g_jX^{[j]}$. Then coefficients of $f(X)\otimes g(X) = \sum_{k\geqslant 0} c_kX^{[k]}$ can be computed as $c_k = \sum_{i=0}^{k} f_i g_{k-i}^{[i]}$. On the other hand, $\mathcal{H}\big(f(X)\big)\mathcal{H}\big(g(X)\big) = \sum_{k\geqslant 0} c'_kX^k$, where $c'_k = \sum_{i=0}^{k} f_i g_{k-i}$. Since all the coefficients of $g(X)$ are elements of $\mathbb{F}_q$, $g_{k-i}^{[i]} = g_{k-i}^{q^i} = g_{k-i}$, for any i and k. Therefore, for any k,

$$c_k = \sum_{i=0}^{k} f_i g_{k-i} = c'_k$$

As mentioned before, $\mathcal{H}$ is a bijective mapping. Therefore, it is a ring isomorphism. ■

For ease of notation, for any linearized polynomial $f(x)$, we will let $\tilde{f}(X)$ denote $\mathcal{H}\big(f(X)\big)$.

Definition 3.3.3. *For a linearized polynomial $f(X) = \sum_{i \geqslant 0} f_i X^{[i]}$, we define its derivative $f'(X)$ as follows:*

$$f'(X) \stackrel{\text{def}}{=} \sum_{i \geqslant 1} i f_i X^{[i-1]}$$

In general, for any integer $a \in \mathbb{N}$, the a-th derivative of $f(X)$ is the following:

$$f^{(a)}(X) \stackrel{\text{def}}{=} \sum_{i \geqslant a} \binom{i}{a} f_i X^{[i-a]} \tag{3.11}$$

Remark. This definition for derivative of a linearized polynomial does not have any direct interpretation in the ring of linearized polynomials. However, translating things into the ring of polynomials through mapping $\mathcal{H}$ we shall see that it becomes the actual derivative there. In fact, $\mathcal{H}\big(f'(X)\big) = \mathcal{H}\big(f(X)\big)'$. Therefore, with a slight abuse of notation we simply write $\tilde{f}'(X)$ which may refer to both $\mathcal{H}\big(f'(X)\big)$ and $\mathcal{H}\big(f(X)\big)'$. □

Remark. For a polynomial $f(X)$ in the ring of polynomials, we let $f'(X)$ to denote its actual derivative. This is an abuse of notation as a linearized polynomial can be also regarded as an element in the ring of polynomials whose actual derivative is zero. In this chapter, however, it is always clear from the context whether a polynomial is an element in the ring of linearized polynomials or it belongs to the ring of polynomials. □

Throughout this chapter, by a multivariate linearized polynomial we mean a polynomial Q of the following form:

$$Q(X, Y_1, Y_2, \ldots, Y_L) = Q_0(X) + Q_1(Y_1) + Q_2(Y_2) + \cdots + Q_L(Y_L) \tag{3.12}$$

where all Q_i's are linearized polynomials. Then we define the corresponding bivariate polynomial $\widetilde{Q}(X, Y)$ as follows:

$$\widetilde{Q}(X, Y) \stackrel{\text{def}}{=} \tilde{Q}_0(X) + \tilde{Q}_1(X)Y + \tilde{Q}_2(X)Y^2 + \cdots + \tilde{Q}_L(X)Y^L \tag{3.13}$$

Q_X and Q_Y which are analogous to first order partial derivatives are defined as follows:

$$Q_X(X, Y_1, Y_2, \ldots, Y_L) \stackrel{\text{def}}{=} Q_0'(X) + Q_1'(Y_1) + Q_2'(Y_2) + \cdots + Q_L'(Y_L)$$

$$Q_Y(X, Y_1, Y_2, \ldots, Y_L) \stackrel{\text{def}}{=} Q_1(X) + 2Q_2(Y_1) + Q_2(Y_2) + \cdots + LQ_L(Y_{L-1})$$

Observe that $\widetilde{Q}_X = \widetilde{Q_X}$, where $\widetilde{Q}_X$ is the first derivative of $\widetilde{Q}(X, Y)$ with respect to X. Similarly, $\widetilde{Q}_Y = \widetilde{Q_Y}$. This is generalized to an analog of the Hasse derivative for a multivariate linearized polynomial Q of the form given in (3.12).

Definition 3.3.4. *For any $a, b \in \mathbb{N}$, we define $\mathscr{D}_{a,b}(Q)$ as follows. If $b > L$, then*

$$\mathscr{D}_{a,b}(Q) \stackrel{\text{def}}{=} 0$$

Otherwise,

$$\mathscr{D}_{a,b}(Q)(X, Y_1, \ldots, Y_L) \stackrel{\text{def}}{=} Q_b^{(a)}(X) + \sum_{i=b+1}^{L} \binom{i}{b} Q_i^{(a)}(Y_{i-b}) \tag{3.14}$$

*where $Q_i^{(a)}(X)$ is the a-th derivative of $Q_i(X)$ defined in (*3.11*).*

Observe that $\mathscr{D}_{a,b}(Q)$ reduces to Q_X and Q_Y, defined earlier, for $(a, b) = (1, 0)$ and $(a, b) = (0, 1)$ respectively. Also, $\mathcal{H}\big(\mathscr{D}_{a,b}(Q)\big)$ is indeed the (a, b)-th Hasse derivative of $\widetilde{Q}(X, Y)$.

3.4 List-decoding with multiplicity two

We start explaining our list-decoding algorithm with the simpler case of multiplicity two. Then we go to the general case in the next section. We do not change the construction of our code reviewed in Section 3.2. The improvement is accomplished by introducing multiplicity at the decoder side.

We fix the parameters of the code: the finite field $\mathbb{F}_q$, the number of information symbols k, the dimension of code n and the list size L. The ambient space $\mathcal{W}$ is an $L + n$-dimensional vector space over $\mathbb{F}_q$. We require that $k \leqslant n \leqslant q - 1$. Then the code $\mathbb{C}_q(k, n, 1, L)$ is constructed as explained in the encoding algorithm in Section 3.2.

Suppose that a codeword $V \in \mathbb{C}_q(k, n, 1, L)$ is transmitted through the operator channel and a vector space $U \in \mathcal{G}(W)$ with dimension d is received. The decoder first checks the following condition on the dimension of received vector space U:

$$d < \frac{L+1}{3}\Big(2n - \frac{L(k-1)}{n}\Big) \tag{3.15}$$

and if it does not hold, then the decoder declares a decoding failure.

List-decoding with multiplicity 2:

The decoder finds a basis for U:

$$\Big\{(x_i, y_{i,1}, y_{i,2}, \ldots, y_{i,L}) : i = 1, 2, \ldots, d\Big\}$$

This is the set of interpolation points $\mathcal{P}$. Then construct a non-zero multivariate linearized polynomial $Q(X, Y_1, Y_2, \ldots, Y_L)$ of the form in (3.12), with each Q_i having q-degree at most $2n - (k-1)i - 1$, for $i = 0, 1, \ldots, L$, subject to the constraint that

$$\begin{aligned} Q(x, y_1, y_2, \ldots, y_L) &= 0 \\ Q_X(x, y_1, y_2, \ldots, y_L) &= 0 \\ Q_Y(x, y_1, y_2, \ldots, y_L) &= 0 \end{aligned} \tag{3.16}$$

for any $(x, y_1, y_2, \ldots, y_L) \in \mathcal{P}$. Then find all the roots $f(X) \in \mathbb{F}_q[X]$, with degree at most $k-1$, of the following equation using the Roth-Ruckenstein algorithm [6]:

$$\widetilde{Q}\big(X, f(X)\big) = 0$$

Coefficients of each root $f(X)$ correspond to an output vector.

We discuss how the various parts of this algorithm can be done efficiently in the next section. In fact, this is a special case, with multiplicity parameter $r = 2$, of list-decoding algorithm B which will be presented in the next section.

Next, we establish the correctness of this algorithm.

Lemma 3.4.1. *For any linearized polynomial $g(X)$ and $i = 1, 2, \ldots, n$,*

1. $g(\alpha_i) = \alpha_i \tilde{g}(e_i)$
2. $g\big(f(\alpha_i)\big) = \alpha_i \tilde{g}(e_i) \tilde{f}(e_i)$, *for any $f(X) \in \mathscr{L}_q[X]$.*

*where α_i is as defined in (*3.8*).*

Proof. Observe that

$$\alpha_i^q = \Big(\sum_{j=0}^{q-2} e_i^{-j} \gamma^{q^j}\Big)^q = \sum_{j=0}^{q-2} (e_i^q)^{-j} \gamma^{q^{j+1}} = \sum_{j=0}^{q-2} e_i^{-j} \gamma^{q^{j+1}} = \alpha_i e_i$$

Then by induction on j, for any $j \geqslant 0$, $\alpha_i^{q^j} = \alpha_i e_i^j$. Suppose that $g(X) = \sum_{j \geqslant 0} g_j X^{[j]}$. Then

$$g(\alpha_i) = \sum_{j \geqslant 0} g_j \alpha_i^{q^j} = \sum_{j \geqslant 0} \alpha_i e_i^j = \alpha_i \tilde{g}(e_i)$$

This completes the first part. The second part is proved by the following sequence of equalities:

$$g\big(f(\alpha_i)\big) = g\big(\alpha_i \tilde{f}(e_i)\big) = g(\alpha_i) \tilde{f}(e_i) = \alpha_i \tilde{g}(e_i) \tilde{f}(e_i)$$

where the first and the last equalities simply follow by the first part of this lemma. The second equality holds since $\tilde{f}(e_i) \in \mathbb{F}_q$ as all the coefficients of $\tilde{f}$ together with e_i are elements of $\mathbb{F}_q$. ■

Suppose that the dimension of the received subspace U is $d = n + t$, where t is the number of errors. We assume that no erasure occurs. In fact, the transmitted codeword V is a subspace of the received subspace U.

Lemma 3.4.2. *Suppose that the number of errors t is bounded as*

$$t < \frac{2}{3}n(L+1) - n - \frac{1}{6}L(L+1)(k-1) \tag{3.17}$$

then there is a non-trivial solution for the multivariate linearized polynomial Q which satisfies (3.16).

Proof. Notice that (3.16) defines a homogeneous system of $3(n+t)$ equations. The number of unknown coefficients is as follows:

$$\sum_{i=0}^{L} 2n - (k-1)i = 2(L+1)n - (k-1)\frac{L(L+1)}{2}$$

This system has a non-zero solution if the number of equations is strictly less than the number of variables. Also, this is necessary in order to guarantee existence of a non-trivial solution i.e.

$$\begin{aligned} 3(n+t) < 2(L+1)n - (k-1)\frac{L(L+1)}{2} \Leftrightarrow \\ t < \frac{2}{3}n(L+1) - n - \frac{1}{6}L(L+1)(k-1) \end{aligned}$$

■

Corollary 3.4.3. *The bound on the number of errors in (3.17) is necessary in order to guarantee a non-trivial solution for the interpolation polynomial Q.*

Lemma 3.4.4. *For $i = 1, 2, \ldots, n$, e_i is a root of the univariate polynomial $\widetilde{Q}\big(X, \tilde{f}_{\boldsymbol{u}}(X)\big)$ with multiplicity 2, where $f_{\boldsymbol{u}}(X) \in \mathscr{L}_q[X]$ is the message polynomial.*

Proof. Since Q is a linearized polynomial and all the basis elements of U are roots of Q, it is zero over the whole vector space U. Notice that the transmitted subspace V is a subspace of U as we assume that no erasure happens. Therefore,

$$Q\big(\alpha_i, f_{\boldsymbol{u}}(\alpha_i), \ldots, f_{\boldsymbol{u}}^{\otimes L}(\alpha_i)\big) = 0$$

Similarly, Q_X and Q_Y are also zero at $\big(\alpha_i, f_{\boldsymbol{u}}(\alpha_i), \ldots, f_{\boldsymbol{u}}^{\otimes L}(\alpha_i)\big)$. Then $\widetilde{Q}\big(e_i, \tilde{f}_{\boldsymbol{u}}(e_i)\big)$ is 0, by the following sequence of equalities:

$$\begin{aligned}
0 =\ & Q\big(\alpha_i, f_{\boldsymbol{u}}(\alpha_i), \ldots, f_{\boldsymbol{u}}^{\otimes L}(\alpha_i)\big) \\
=\ & Q_0(\alpha_i) + Q_1\big(f_{\boldsymbol{u}}(\alpha_i)\big) + \cdots + Q_L\big(f_{\boldsymbol{u}}^{\otimes L}(\alpha_i)\big) \\
=\ & \alpha_i\Big[\tilde{Q}_0(e_i) + \tilde{Q}_1(e_i)\tilde{f}_{\boldsymbol{u}}(e_i) + \cdots + \tilde{Q}_L(e_i)\tilde{f}_{\boldsymbol{u}}^{L}(e_i)\Big] && (3.18) \\
=\ & \alpha_i\widetilde{Q}\big(e_i, \tilde{f}_{\boldsymbol{u}}(e_i)\big) && (3.19)
\end{aligned}$$

By Lemma 3.3.2, we know that $\mathcal{H}$ preserves the multiplication over $\mathscr{L}_q[X]$. Therefore, $\mathcal{H}\big(f_{\boldsymbol{u}}^{\otimes L}(X)\big) = \mathcal{H}\big(f_{\boldsymbol{u}}(X)\big)^L = \tilde{f}_{\boldsymbol{u}}^{L}(X)$. This fact together with Lemma 3.4.1 imply (3.18). (3.19) holds just by definition of $\widetilde{Q}$ in (3.13).

Next we show that $\widetilde{Q}_X(X, Y)$ and $\widetilde{Q}_Y(X, Y)$ are also zero at $\big(e_i, \tilde{f}_{\boldsymbol{u}}(e_i)\big)$, where $\widetilde{Q}_X$ and $\widetilde{Q}_Y$ are the first derivatives of $\widetilde{Q}$ with respect to X and Y, respectively.

$$\begin{aligned}
0 =\ & Q_X\big(\alpha_i, f_{\boldsymbol{u}}(\alpha_i), \ldots, f_{\boldsymbol{u}}^{\otimes L}(\alpha_i)\big) \\
=\ & Q_0'(\alpha_i) + Q_1'\big(f_{\boldsymbol{u}}(\alpha_i)\big) + \cdots + Q_L'\big(f_{\boldsymbol{u}}^{\otimes L}(\alpha_i)\big) \\
=\ & \alpha_i\Big[\tilde{Q}_0'(e_i) + \tilde{Q}_1'(e_i)\tilde{f}_{\boldsymbol{u}}(e_i) + \cdots + \tilde{Q}_L'(e_i)\tilde{f}_{\boldsymbol{u}}^{L}(e_i)\Big] && (3.20) \\
=\ & \alpha_i\widetilde{Q}_X(X, Y)\Big|_{\left(e_i, \tilde{f}_{\boldsymbol{u}}(e_i)\right)} && (3.21)
\end{aligned}$$

(3.20) holds similar to (3.18) if we replace Q by Q_X and Q_i by Q_i'. (3.21) follows by simply taking the first derivative of $\widetilde{Q}(X, Y)$, given in (3.13), with respect to X. Also,

$$\begin{aligned}
0 =\ & Q_Y\big(\alpha_i, f_{\boldsymbol{u}}(\alpha_i), \ldots, f_{\boldsymbol{u}}^{\otimes L}(\alpha_i)\big) \\
=\ & Q_1(\alpha_i) + 2Q_2\big(f_{\boldsymbol{u}}(\alpha_i)\big) + \cdots + LQ_L\big(f_{\boldsymbol{u}}^{\otimes (L-1)}(\alpha_i)\big) \\
=\ & \alpha_i\Big[\tilde{Q}_1(e_i) + 2\tilde{Q}_2(e_i)\tilde{f}_{\boldsymbol{u}}(e_i) + \cdots + L\tilde{Q}_L(e_i)\tilde{f}_{\boldsymbol{u}}^{L-1}(e_i)\Big] && (3.22) \\
=\ & \alpha_i\widetilde{Q}_Y(X, Y)\Big|_{\left(e_i, \tilde{f}_{\boldsymbol{u}}(e_i)\right)} && (3.23)
\end{aligned}$$

(3.22) holds similar to (3.18) and (3.20). (3.23) follows by simply taking the first derivative of $\widetilde{Q}(X, Y)$, given in (3.13), with respect to Y.

We showed that the bivariate polynomial $\widetilde{Q}(X, Y)$ passes through $\big(e_i, \tilde{f}_{\boldsymbol{u}}(e_i)\big)$ with multiplicity 2. Therefore, e_i is a root of the univariate polynomial $\widetilde{Q}(X, \tilde{f}_{\boldsymbol{u}}(X))$ with multiplicity 2. ■

Corollary 3.4.5. *$\widetilde{Q}\big(X, \tilde{f}_{\boldsymbol{u}}(X)\big)$ is identically zero.*

Proof. Notice that the degree of each polynomial $\tilde{Q}_i$ is equal to the q-degree of Q_i which is $2n - (k-1)i - 1$. Also, the degree of $\tilde{f}(X)$ is at most $k-1$. Therefore, the degree of each term $\tilde{Q}_i(X)\tilde{f}^i(X)$ is at most $2n-1$ which implies that the total degree of $\widetilde{Q}\big(X, \tilde{f}_{\boldsymbol{u}}(X)\big)$ is at most $2n-1$. On the other hand, $\widetilde{Q}\big(X, \tilde{f}_{\boldsymbol{u}}(X)\big)$ has at least n distinct roots $e_1, e_2, \ldots, e_n$, each with multiplicity 2. Therefore, it must be identically zero. ■

Theorem 3.4.6. *Suppose that the number of errors normalized by the dimension n is at most*

$$\frac{2}{3}(L+1) - 1 - \frac{1}{6}L(L+1)\frac{(k-1)}{n}$$

Then the list-decoding algorithm with multiplicity 2 produces a list of size at most L which includes the transmitted message $\boldsymbol{u}$.

Proof. Observe that this condition on the number of errors t is equivalent to (3.15) by plugging $d = n + t$ in (3.15). Therefore, the list-decoding algorithm is performed and by Lemma 3.4.2, there is a non-trivial interpolation polynomial Q that satisfies (3.16). Then by Corollary 3.4.5, $\tilde{f}_{\boldsymbol{u}}(X)$ is a solution to $\widetilde{Q}\big(X, f(X)\big) = 0$. Thus the message u is included in the list generated by the decoder. Notice that $\widetilde{Q}\big(X, f(X)\big)$ can be seen as a univariate polynomial with degree L over $\mathbb{F}[X]$ which is a Euclidean domain. Since Q is a non-zero polynomial, so is $\widetilde{Q}$. Therefore, there are at most L roots for the equation $\widetilde{Q}\big(X, f(X)\big) = 0$. ■

By Theorem 3.4.6 and Corollary 3.4.3, the necessary and sufficient condition for a correct list-decoding with multiplicity 2 is that the normalized number of errors is

bounded as

$$\tau < \frac{2}{3}(L+1) - 1 - \frac{1}{6}L(L+1)\frac{(k-1)}{n}$$

where $\tau = t/n$ is the normalized number of errors. The packet rate R^*, as defined in Section 2.4.3, is given by k/n. Basically the number of information packets is k and the number of encoded packets is n. Then we can express the bound on error-correction radius as

$$\tau < \frac{2}{3}(L+1) - 1 - \frac{1}{6}L(L+1)R^* \tag{3.24}$$

3.5 List-decoding with arbitrary multiplicity

In this section, we present list-decoding algorithm with multiplicity in the general case. We essentially generalize the results in the foregoing section. We first state list-decoding algorithm B which takes multiplicity into account in the general case. Then we prove the correctness of this algorithm and show the achievable error-correction radius. At the end, we discuss the parameters of the proposed algorithm.

3.5.1 List-decoding Algorithm

Consider the code $\mathbb{C}_q(k, n, 1, L)$ constructed by the encoding algorithm discussed in Section 3.2. A codeword $V \in \mathbb{C}_q(k, n, 1, L)$ is transmitted through the operator channel and the decoder receives a vector space $U \in \mathcal{G}(W)$. There is a multiplicity parameter r that is picked by the decoder arbitrarily and is independent of the code construction. We will discuss later how to pick r in order to maximize the error-correction capability. Then the decoder looks at the dimension of received vector space U and if the condition

$$\dim(U) < \frac{L+1}{r+1}\Big(2n - \frac{L(k-1)}{n}\Big) \tag{3.25}$$

is satisfied, then the list-decoding algorithm B is run. Otherwise, the decoder declares a decoding failure.

List-decoding Algorithm B:

The decoder accepts as input a vector space U which is a subspace of $\mathcal{W}$. It then outputs a list of size at most L of vectors in ${\mathbb{F}_q}^k$ in three steps:

1. *Computing the interpolation points:*
   Find a set of basis elements
   $$\Big\{(x_i, y_{i,1}, y_{i,2}, \ldots, y_{i,L}) : i = 1, 2, \ldots, d\Big\}$$
   for U. This is used as the set of interpolation points $\mathcal{P}$.

2. *Interpolation:*
   Construct a non-zero multivariate linearized polynomial $Q(X, Y_1, Y_2, \ldots, Y_L)$ of the form in (3.12) with each Q_i having q-degree at most $nr - (k-1)i - 1$, for $i = 0, 1, \ldots, L$, subject to the constraint that
   $$\mathscr{D}_{a,b}(Q)(x, y_1, y_2, \ldots, y_L) = 0 \tag{3.26}$$
   for any $(x, y_1, y_2, \ldots, y_L) \in \mathcal{P}$ and $a, b \in \mathbb{N}$ such that $a + b < r$.

3. *Factorization:*
   Find all the roots $f(X) \in \mathbb{F}_q[X]$, with degree at most $k - 1$, of the following equation:
   $$\widetilde{Q}\big(X, f(X)\big) = 0$$
   The decoder outputs coefficients of each root $f(X)$ as a vector of length k.

The first step of the list-decoding algorithm B can be done using elementary linear algebraic operations. The second step is basically solving a system of linear equations. There are several ways for doing that. The most straightforward way is the Gaussian elimination method. However, this method does not take advantage of the certain structure of this system of equations and therefore, it is not efficient. An efficient polynomial-time interpolation algorithm in the ring of linearized polynomials is presented in [8]. The factorization step can be done in an efficient polynomial-time using Roth-Ruckenstein algorithm [6].

3.5.2 Correctness of List-decoding Algorithm

We assume that the decoder receives transmitted codeword V corrupted with t errors and no erasures. Therefore, the input to the decoder is an $n+t$ dimensional vector

space U which contains the subspace V. The following lemma provides a bound on t which guarantees correct list-decoding.

Lemma 3.5.1. *There is a non-trivial solution for multivariate linearized polynomial Q which satisfies (3.26) provided that the number of errors t satisfy the following condition:*

$$\frac{t}{n} < \frac{2(L+1)}{r+1} - 1 - \frac{L(L+1)(k-1)}{r(r+1)n} \tag{3.27}$$

Proof. The set $\mathcal{P}$ contains $n+t$ elements. The number of equations in (3.26) corresponding to each interpolation point is $\binom{r+1}{2}$. Therefore, (3.26) defines a system of $\binom{r+1}{2}(n+t)$ linear equations. The number of unknown coefficients is:

$$\sum_{i=0}^{L} rn - (k-1)i = (L+1)rn - (k-1)\frac{L(L+1)}{2}$$

If the number of equations is strictly less than the number of variables, then there is a non-trivial solution to this system of equations. This is also a necessary condition in order to guarantee a non-trivial solution i.e.

$$\begin{aligned}\binom{r+1}{2}(n+t) < (L+1)rn - (k-1)\frac{L(L+1)}{2} \Leftrightarrow \\ \frac{t}{n} < \frac{2(L+1)}{r+1} - 1 - \frac{L(L+1)(k-1)}{r(r+1)n}\end{aligned}$$

which completes the proof of lemma. ■

Corollary 3.5.2. *The condition in (3.27) is necessary to guarantee a non-zero solution for the interpolation polynomial Q.*

Lemma 3.5.3. *Let $f_{\boldsymbol{u}}(X)$ be the message polynomial and for $i = 1, 2, \ldots, n$, let e_i be the element of $\mathbb{F}_q$ corresponding to α_i. Then e_i is a root of the univariate polynomial $\widetilde{Q}\big(X, \tilde{f}_{\boldsymbol{u}}(X)\big)$ with multiplicity r.*

Proof. Notice that for any a and b, $\mathscr{D}_{a,b}(Q)$ is also a linearized polynomial. Since all the basis elements of U are roots of $\mathscr{D}_{a,b}(Q)$, it is zero over the whole vector space U. We assume that there are no erasures which implies that the transmitted codeword V is a

subspace of U. In particular, $\mathscr{D}_{a,b}(Q)\big(\alpha_i, f_{\boldsymbol{u}}(\alpha_i), \ldots, f_{\boldsymbol{u}}^{\otimes L}(\alpha_i)\big) = 0$. Then the following sequence of equalities imply that for any $a, b \in \mathbb{N}$ such that $a + b < r$, $\widetilde{Q}^{(a,b)}(X, Y)$ is zero at $\big(e_i, \tilde{f}(e_i)\big)$, where $\widetilde{Q}^{(a,b)}(X, Y)$ is the (a, b)-th Hasse derivative of $\widetilde{Q}(X, Y)$.

$$
\begin{aligned}
0 = \; & \mathscr{D}_{a,b}(Q)\big(\alpha_i, f_{\boldsymbol{u}}(\alpha_i), \ldots, f_{\boldsymbol{u}}^{\otimes L}(\alpha_i)\big) \\
= \; & Q_b^{(a)}(\alpha_i) + \sum_{j=b+1}^{L} \binom{j}{b} Q_j^{(a)}\big(f_{\boldsymbol{u}}^{\otimes (j-b)}(\alpha_i)\big) \\
= \; & \alpha_i \tilde{Q}_b^{(a)}(e_i) + \sum_{j=b+1}^{L} \binom{j}{b} \alpha_i \tilde{Q}_j^{(a)}(e_i) \mathcal{H}\big(f_{\boldsymbol{u}}^{\otimes (j-b)}(X)\big)\Big|_{e_i} \qquad (3.28) \\
= \; & \alpha_i \sum_{j=b}^{L} \binom{j}{b} \tilde{Q}_j^{(a)}(e_i) \tilde{f}_{\boldsymbol{u}}^{j-b}(e_i) \qquad (3.29) \\
= \; & \alpha_i \widetilde{Q}^{(a,b)}\big(X, Y\big)\Big|_{\left(e_i, \tilde{f}_{\boldsymbol{u}}(e_i)\right)} \qquad (3.30)
\end{aligned}
$$

(3.28) holds by Lemma 3.4.1. By Lemma 3.3.2, $\mathcal{H}$ preserves the multiplication over $\mathscr{L}_q[X]$. Therefore, (3.29) follows. (3.30) holds just by definition of $\widetilde{Q}$ in (3.13) and the Hasse derivative of a bivariate polynomial.

We showed that the bivariate polynomial $\widetilde{Q}(X, Y)$ passes through $\big(e_i, \tilde{f}_{\boldsymbol{u}}(e_i)\big)$ with multiplicity r. Therefore, e_i is a root of the univariate polynomial $\widetilde{Q}(X, \tilde{f}_{\boldsymbol{u}}(X))$ with multiplicity r. ■

Corollary 3.5.4. *$\widetilde{Q}\big(X, \tilde{f}_{\boldsymbol{u}}(X)\big)$ is the all zero polynomial.*

Proof. The degree of each polynomial $\tilde{Q}_i$ is equal to the q-degree of Q_i which is $rn - (k-1)i - 1$. Also, the degree of $\tilde{f}_{\boldsymbol{u}}(X)$ is at most $k - 1$. Therefore, the degree of each term $\tilde{Q}_i(X)\tilde{f}_{\boldsymbol{u}}^i(X)$ is at most $rn - 1$. Thus the total degree of $\widetilde{Q}\big(X, \tilde{f}_{\boldsymbol{u}}(X)\big)$ is at most $rn - 1$. On the other hand, $\widetilde{Q}\big(X, \tilde{f}_{\boldsymbol{u}}(X)\big)$ has at least n distinct roots $e_1, e_2, \ldots, e_n$, each with multiplicity at least r. Therefore, it must be identically zero. ■

Theorem 3.5.5. *List-decoding algorithm B with multiplicity r produces a list of size at most L which includes the transmitted message $\boldsymbol{u}$ as long as*

$$
\frac{t}{n} < \frac{2(L+1)}{r+1} - 1 - \frac{L(L+1)(k-1)}{r(r+1)n} \qquad (3.31)
$$

Proof. Observe that the condition on the dimension of received vector space U in (3.25) is equivalent to this condition on the normalized dimension of errors t/n. Therefore, the list-decoding algorithm B is run, and by Lemma 3.5.1, there is a non-trivial interpolation polynomial Q that satisfies (3.26). Consequently Corollary 3.5.4 implies that $f_{\boldsymbol{u}}(X)$ is a solution to the equation $\widetilde{Q}\big(X, f(X)\big) = 0$. Therefore, the transmitted message $\boldsymbol{u}$ is included in the output list. Notice that $\widetilde{Q}\big(X, f(X)\big)$ can be regarded as a univariate polynomial with degree L over $\mathbb{F}_{q^n}[X]$ which is a Euclidean domain. Since Q is a non-zero polynomial, so is $\widetilde{Q}$. Therefore, there are at most L roots for the equation $\widetilde{Q}\big(X, f(X)\big) = 0$. ■

3.5.3 Error-Correction Radius

In Section 2.4, we introduced the new parameter *packet rate* in order to express the results in a more convenient way. We take the same approach to present the results of this section. The packet rate R^* is equal to the number of information packets normalized by the number of encoded packets injected into the network. The packet rate of the Koetter-Kschischang code is k/n. Notice that we did not change the structure of our list-decodable codes proposed in Chapter 2. We just restricted our attention to the special case of $m = 1$. The packet rate of the code $\mathbb{C}_q(k, n, m, L)$ is given in (2.31) as k/nm. For the special case of $m = 1$, the packet rate of $\mathbb{C}_q(k, n, 1, L)$ is then k/n.

Notice that the q-degree of linearized polynomials Q_i has to be non-negative. Therefore, the following condition is enforced:

$$\begin{aligned} rn - (k-1)L - 1 \geqslant 0 \Rightarrow \\ r > \frac{L(k-1)}{n} \approx LR^* \end{aligned} \tag{3.32}$$

Corollary 3.5.2 and Theorem 3.5.5 together imply that the necessary and sufficient condition for a correct list-decoding with multiplicity r is the following:

$$\frac{t}{n} < \frac{2(L+1)}{r+1} - 1 - \frac{L(L+1)(k-1)}{r(r+1)n}$$

Therefore, this provides a bound on the error-correction radius of list-decoding algo-

rithm B with multiplicity r. Using the approximation

$$\frac{k-1}{n} \approx \frac{k}{n} = R^*$$

we express the final result as follows. The list-decoding algorithm B with multiplicity r successfully recovers the message polynomial provided that

$$\tau < \frac{2(L+1)}{r+1} - 1 - \frac{L(L+1)}{r(r+1)} R^* \tag{3.33}$$

where $\tau = t/n$ is the error-correction radius of the proposed algorithm. Notice that the value of r is independent of the construction of the code $\mathbb{C}_q(k, n, 1, L)$ and we can choose it arbitrarily at the decoder. Therefore, we pick r such that the bound on error-correction radius given in (3.33) is maximized. We call this value r_{max}. This is a simple optimization problem. Given the fact that r has to be an integer, r_{max} in terms of L and R^* is given as $\lceil LR^* \rceil$ which also satisfies (3.32).

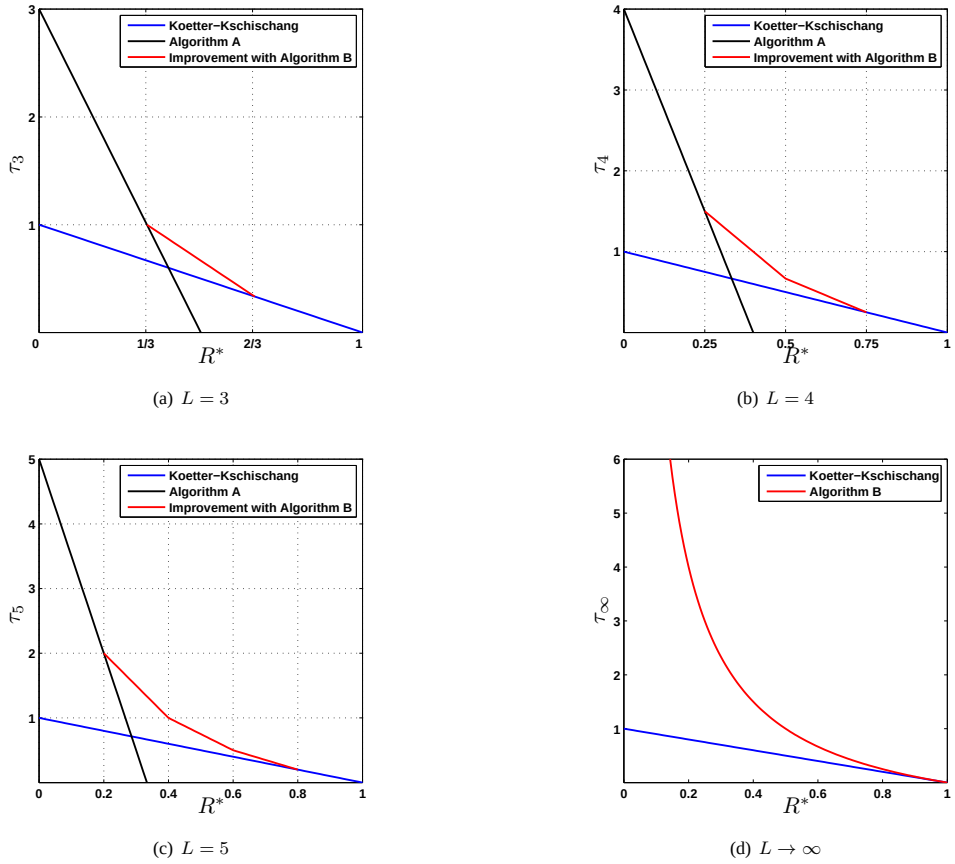
Observe that for multiplicity $r = 1$, as expected, (3.33) reduces to the result of foregoing chapter, on error-correction radius of list-decoding algorithm, given in (3.10).

3.6 Discussion and Conclusions

We have proposed a new list-L decoding algorithm with error-correction radius given in (3.33). r is the multiplicity which is independent of the code construction and can be decided at the decoder. R^* is the packet rate of the code. As mentioned in the foregoing section, for a given packet rate R^* and list size L, the parameter r which maximizes the bound on error-correction radius is $\lceil LR^* \rceil$. For a fixed L, plugging in this value into the expression (3.33) we get a piecewise linear function for error-correction radius versus the packet rate. In fact, for $i = 1, 2, \ldots, L$, the bound on normalized error-correction radius is linear on the interval $[\frac{i-1}{L}, \frac{i}{L}]$ of packet rates R^* and is given by

$$\begin{cases} \frac{2(L+1)}{i+1} - 1 - \frac{L(L+1)}{i(i+1)} R^* & i \in [L], R^* \in [\frac{i-1}{L}, \frac{i}{L}] \\ 0 & R^* > 1 \end{cases} \tag{3.34}$$

Let τ_L denote the normalized error-correction radius for the list size L. We plot the bound on normalized error-correction radius τ_L, given in (3.34), versus the packet rate



(a) $L = 3$ (b) $L = 4$

(c) $L = 5$ (d) $L \to \infty$

Figure 3.2: Improvement on error-correction radius upon previous works by using multiplicity for several values of list size L

R^* for various amounts of L in Figure 3.2. In general, for any L, $r_{\max} = 1$ for packet rates less than $\frac{1}{L}$ which means that we are back to the list-decoding algorithm A. In this case, we get no improvement upon the previous work. Also, for packet rates between $\frac{L-1}{L}$ and 1, the optimum value for multiplicity r is equal to L. In this case, we get same results as in Koetter-Kschischang construction. For $R^* \in [\frac{1}{L}, \frac{L-1}{L}]$, we get improvements in error-correction radius, upon both list-decoding algorithm A and Koetter-Kschischang codes, using multiplicities in list-decoding algorithm B. This can be seen in Figure 3.2(a)-(c) for $L = 3, 4, 5$. As L tends to infinity, it can be shown that τ_L tends to $\frac{1}{R^*} - 1$. This is plotted in Figure 3.2(d).

A natural question that arises is the following: Is there a direct way of using multiplicities in the ring of linearized polynomials without resorting to the ring of poly-

nomials? This goes back to the very first question in this chapter which is how to define multiple roots in the ring of linearized polynomials. Despite all our efforts, we have not been able to give an explicit answer to this question. In fact, this problem can be of independent interest for someone who wishes to study the ring of linearized polynomials.

In this chapter, we expressed the normalized error-correction radius of the code in terms of the packet rate R^* rather than the symbol rate R. This enables us to present the results in a more convenient way. It is still interesting, however, to compare the results of this chapter with the previous list-decoding work presented in Chapter 2 and the Koetter-Kschischang code [2] in terms of the symbol rate R defined in (1.8). The normalized error-correction radius of Koetter-Kschischang code is bounded as

$$\begin{aligned}\tau < \frac{n-k+1}{n} \approx & 1-\frac{k}{n} \\ = & 1-(1+\frac{n}{m})\frac{km}{n(m+n)} = 1-(1+\frac{n}{m})R\end{aligned}$$

When m is large compared to n, this can be approximated by $1-R$. Therefore, the plots for Koetter-Kschischang code remain the same regardless of whether we express τ in terms of R or R^*.

The bound on the normalized error-correction radius of list-decoding algorithm A, given in (3.10), can be expressed in terms of symbol rate R as

$$\tau < L - \frac{1}{2}nL^2(L+1)R$$

For the error-correction radius of list-decoding algorithm B, (3.33) can be expressed in terms of symbol rate R as

$$\tau < \frac{2(L+1)}{r+1} - 1 - \frac{nL^2(L+1)}{r(r+1)}R$$

The problem is that one has to take n into account in order to make a comprehensive comparison. We can not plot the results but we still improve the error-correction radius over the results of the previous chapter. In order to compare with Koetter-Kschischang codes, however, we need to consider the behavior for a certain value of n. As n grows large, our list-decoding results, with or without multiplicity, loses its advantage over the Koetter-Kschischang results.

A disadvantage of the list-decoding algorithm with multiplicities presented in this Chapter compared to the previous results discussed in Chapter 2 is that we are only

able to correct errors, whereas in the original Koetter-Kschischang codes [2] and also in the list-decoding algorithm without multiplicities both errors and erasures can be recovered at the decoder as long as the total number of errors and erasures satisfies a certain bound. This seems to be an inherent property of our algorithm. An open problem here is how to improve the error-correction radius upon the previous list-decoding algorithm, for a fixed list-size at the output, such that both errors and erasures can be handled at the decoder.

Acknowledgements

The results of this chapter, for the special case of multiplicity two, was presented at Allerton conference 2011 and appeared in proceedings as: H. Mahdavifar and A.Vardy, "Algebraic List-decoding of Subspace Codes with Multiplicities", *Proceedings of the 49th annual Allerton Conference on Communications, Control and Computing*, September 2011.

Bibliography

[1] V. Guruswami and M. Sudan, "Improved Decoding of Reed-Solomon and Algebraic-Geometric codes," *IEEE Transactions on Information Theory*, vol. 45, pp. 1757–1767, Sept. 1999.

[2] R. Koetter and F.R. Kschischang, "Coding for Errors and Erasures in Random Network Coding." *IEEE Transactions on Information Theory*, vol. 54, pp. 3579–3591, August 2008.

[3] F. J. MacWilliams and N. J. A. Sloane, "The Theory of Error-Correcting Codes." New York: North Holland, 1977

[4] H. Mahdavifar and A. Vardy "Algebraic list-decoding on the operator channel," *Proc. IEEE Intern. Symp. Information Theory*, pp. 1193–1197, Austin. TX., June 2010

[5] H. Mahdavifar and A. Vardy "Algebraic list-decoding of subspace codes," *to appear in IEEE Transactions on Information Theory*

[6] R. M. Roth, G. Ruckenstein, "Efficient decoding of Reed-Solomon codes beyond half the minimum distance," *IEEE Transactions on Information Theory*, vol. 46, pp. 246–257, Jan. 2000.

[7] M. Sudan, “Decoding of Reed-Solomon codes beyond the error-correction bound,” *J. Complexity,* 12: 180–193, March 1997

[8] H. Xie, Z. Yan, B. W, Suter, “General linearized polynomial interpolation and its applications,” *2011 International Symp. on Network Coding,* Beijing, China, July 25-27 2011

Chapter 4

An Alternative Approach: Construction and List-Decoding

4.1 Introduction

Subspace codes were introduced by Koetter and Kschischang to correct errors and erasures in networks with a randomized protocol where the topology is unknown (the non-coherent case). In this model, the codewords are vector subspaces of a fixed ambient space; thus codes for this model are collections of such subspaces. In Chapter 1, we reviewed the basics of subspace codes and Koetter-Kschischang algebraic constructions that are regarded as analogous to Reed-Solomon codes. In Chapter 2, we developed a family of subspace codes based upon the Koetter-Kschischang construction which are efficiently list decodable. Using these codes, we achieved a better error-correction radius than low rate Koetter-Kschischiang codes. In Chapter 3, we introduced multiplicity in the ring of linearized polynomials with the aim of enforcing multiple roots for the interpolation polynomial in our list-decoding algorithm in order to further improve the error-correction results of our codes. Basically, we did not change the construction proposed in Chapter 2. All the additional operations with respect to the multiplicity part are done on the decoder's side.

Koetter-Kschischang algebraic subspace codes, originally called Reed-Solomon-like codes in [4], is analogous to the Reed-Solomon codes in classical block codes

wherein symbols are replaced by vectors, regular polynomials by *linearized polynomials*, and sequences of symbols with an $\mathbb{F}_q$-linear span of the corresponding vectors. Our starting point in Chapter 2 was to evaluate all the powers of the linearized message polynomial, up to some power L, in order to list-decode with an output list of size of at most L. In a sense, our algorithm can be regarded as analogous to the Sudan list-decoding algorithm of Reed-Solomon codes [10].

In this chapter, we introduce a new family of subspace codes that allows a simple linear-algebraic list-decoding by using $s + 1$-variate interpolation polynomials, where s is a design parameter. In fact, we append evaluations of the message polynomial over a certain set of s evaluation points in order to construct each of the basis elements of our codeword. This is analogous to doing a folding in the construction of Reed-Solomon codes. The entire list-decoding algorithm is linear-algebraic. A system of linear equations is solved for the interpolation step and another linear system is solved to compute the set of all the possible solutions which is in fact a linear space. This is motivated by the recent work of Vadhan [11, Ch. 5] and Guruswami [2] which suggested a simplified version with no need of the multiplicity of previously proposed for a list-decoding algorithm of folded Reed-Solomon codes by Guruswami and Rudra in [3]. The latter was built on the work of Parvaresh and Vardy on list-decoding of Reed-Solomon codes by proposing multivariate interpolation [8]. The end result on error-correction capability of our new construction can be expressed as follows: for any integer s, our list-decoder using $s+1$-interpolation polynomials guarantees successful recovery of the message subspace provided the normalized dimension of errors is at most $s(1 - sR)$. The same list-decoding algorithm can be used to correct erasures as well as errors. The size of output list is at most $\mathcal{Q}^{s-1}$, where $\mathcal{Q}$ is the size of the field that the message symbols are chosen from.

The rest of this chapter is organized as follows. We start with a brief overview of linearized polynomials, subspace codes and our results in previous chapters in Section 4.2. In Section 4.3, we discuss our new construction of subspace codes, then propose a list-decoding algorithm. Then we establish the correctness of the algorithm and provide the error-correction radius and other parameters of our code.

4.2 Background and Prior Work

In this section, we first briefly review the relevant terminology for subspace codes and Koetter-Kschischang algebraic subspace codes that we discussed in detail in Chapter 1 and Chapter 2. Then we recap our list-decoding results of [6] and [7] presented in Chapter 2, which provide a new family of subspace codes that are efficiently list-decodable. In [6] and [7], we suitably modified and extended Koetter-Kschischang codes in many important respects in order to enable list-decoding.

Linearized polynomials are a family of polynomials which act as linear maps with respect to a certain base field. This fundamental property make them very useful in the construction of codes over subspaces. A polynomial over some extension field $\mathbb{F}_{q^m}$ of $\mathbb{F}_q$ is called $\mathbb{F}_q$-linearized if it has the following form:

$$f(X) = \sum_{i=0}^{s} a_i X^{q^i},$$

where $a_i \in \mathbb{F}_{q^m}$, for $i = 0, 1, \ldots, s$. Suppose that X^{q^s} is the leading monomial with non-zero coefficient. Then we say that the polynomial $f(X)$ has q-degree s. When q is fixed under discussion, we let $X^{[i]}$ denote X^{q^i}. The fundamental property of linearized polynomials from which they receive their name is that they act as linear maps with respect to the base field $\mathbb{F}_q$. The set of linearized polynomials forms a non-commutative ring under addition $+$ and composition operation $\otimes$. For any two linearized polynomials $f_1(X)$ and $f_2(X)$, the sum $f_1(X) + f_2(X)$ is also linearized. Furthermore, the composition operation $f_1(X) \otimes f_2(X)$ is defined to be the composition $f_1\big(f_2(X)\big)$ which is always a linearized polynomial. The ring of linearized polynomials over $\mathbb{F}_{q^m}$ is denoted by $\mathscr{L}_{q^m}[X]$.

Let $\mathcal{W}$ be a fixed N-dimensional vector space over $\mathbb{F}_q$. The set of all subspaces of $\mathcal{W}$, denoted as $\mathscr{P}_q(\mathcal{W})$, forms a metric space under the following metric. For any two subspaces $A, B \in \mathscr{P}_q(\mathcal{W})$, the distance $d(A, B)$ between A and B is defined as

$$d(A, B) \stackrel{\text{def}}{=} \dim(A + B) - \dim(A \cap B)$$

A subspace code $\mathbb{C}$ associated with the ambient space $\mathcal{W}$ is a non-empty subset of $\mathscr{P}_q(\mathcal{W})$. A codeword is an element of $\mathbb{C}$ which is in fact a subspace of $\mathcal{W}$. Suppose

that the dimension of any $V \in \mathbb{C}$ is at most n. Then the rate of the code R is defined as follows:

$$R \stackrel{\text{def}}{=} \frac{\log_q |\mathcal{C}|}{nN} \tag{4.1}$$

Koetter and Kschischang [4] constructed a remarkable family of subspace codes, regarded as an analogous to Reed-Solomon codes, wherein symbols are replaced by vectors, polynomials with linearized polynomials and sequences of symbols with an $\mathbb{F}_q$-linear span of the corresponding vectors. A set $A = \{\alpha_1, \ldots, \alpha_n\}$ of n linearly independent vectors in $\mathbb{F}_{q^m}$ is fixed. In fact, $\mathbb{F}_{q^m}$ can be regarded as a vector space of dimension m over $\mathbb{F}_q$. The set A is used as the set of evaluation points over which we evaluate the message polynomial. Let $\boldsymbol{u} = (u_0, \ldots, u_{k-1})$ be the message vector. We regard u_i's as coefficients of a linearized polynomial that is, $f_{\boldsymbol{u}}(X) = \sum_{i=0}^{k-1} u_i X^{[i]}$ is the linearized message polynomial. For each i, $i = 1, 2, \ldots, n$, we evaluate $f_{\boldsymbol{u}}(X)$ over α_i and then append it to α_i to form the vector $v_i = \big(\alpha_i, f_{\boldsymbol{u}}(\alpha_i)\big)$. Vectors v_i's are elements of an ambient space

$$\mathcal{W} = \langle A \rangle \oplus \mathbb{F}_{q^m}$$

which is an $n+m$-dimensional vector space over $\mathbb{F}_q$. Then the corresponding codeword V is the $\mathbb{F}_q$-linear span of v_i's, for $i = 1, 2, \ldots, n$. In fact, the codeword V is an n-dimensional subspace of the ambient space $\mathcal{W}$. We represent each element of $\mathcal{W}$ as a vector (x, y) where x belongs to the span of α_i's and y is an element of $\mathbb{F}_{q^m}$.

The codeword V is transmitted through the network and another vector space $U \in \mathscr{P}_q(\mathcal{W})$ is received. At the decoder, we aim to construct a non-zero interpolation polynomial that passes through all the elements of the received vector space U. To this end, we find a basis

$$\Big\{(x_i, y_i) \ : \ 1 \leqslant i \leqslant \dim(U)\Big\}$$

Then we construct a non-zero bivariate linearized polynomial $Q(X, Y)$ of the form

$$Q(X, Y) = Q_0(X) + Q_1(Y),$$

such that $Q(x_i, y_i) = 0$ for all the basis elements (x_i, y_i) of the received subspace. Also, Q_0 and Q_1 are subject to certain degree constraints. Then the equation $Q\big(X, f(X)\big) = 0$ is solved to recover the message polynomial. If not too many errors and erasures happen, then a sufficient number of roots, i.e. $\alpha \in \langle A \rangle$ such that $\big(\alpha, f_{\boldsymbol{u}}(\alpha)\big) \in U \cap V$, is

guaranteed for the univariate polynomial $Q\big(X, f_{\boldsymbol{u}}(X)\big)$ that makes it identically zero. Hence, $f_{\boldsymbol{u}}(X)$ is the unique solution to the equation $Q\big(X, f(X)\big) = 0$. This decoding algorithm is guaranteed to recover the transmitted message provided that the normalized error-correction radius $\tau = (t + \rho)/n$, where t and ρ are the dimension of errors and erasures respectively, satisfies

$$\tau < \frac{n-k+1}{n} = 1 - \frac{k-1}{n} \approx 1 - \Big(1 + \frac{n}{m}\Big)R = 1 - R^* \tag{4.2}$$

where R^* is the *packet rate* of the code. The packet rate is defined as the ratio of number of information packets to the number of encoded packets.

The main obstacle in the list-decoding of Koetter-Kschischang codes is that the ring of linearized polynomials is non-commutative. Because of that, an equation of certain degree over the ring of linearized polynomials may have exponentially many roots, while one has to guarantee a bounded list-size at the output of the decoder. In order to enable list-decoding, we modified the Koetter-Kschischang codes in many important ways in Chapter 2. Our work essentially leads to a new construction of subspace codes that is efficiently list-decodable. Next, we briefly review the encoding and decoding of these new subspace codes.

We use the *normal basis* of an extension field $\mathbb{F}_{q^m}$ of $\mathbb{F}_q$ in our construction (see [5, Ch. 4.9]). $\mathbb{F}_{q^m}$ contains a primitive element γ such that $\gamma, \gamma^q, \ldots, \gamma^{q^{m-1}}$ form a basis for $\mathbb{F}_{q^m}$ as a vector space over $\mathbb{F}_q$. This is called a normal basis for $\mathbb{F}_{q^m}$. Fix a finite field $\mathbb{F}_q$ and let n divide $q-1$. Then the equation $x^n - 1 = 0$ has n distinct solutions in $\mathbb{F}_q$. Let $e_1 = 1, e_2, e_3, \ldots, e_n$ be these solutions. Let $\mathbb{F} = GF(q^{nm})$ and γ be a generator of a normal basis for $\mathbb{F}$. Then define

$$\alpha_i = \gamma + e_i^{-1}\gamma^{q^m} + e_i^{-2}\gamma^{q^{2m}} + \cdots + e_i^{-(n-1)}\gamma^{q^{(n-1)m}} \tag{4.3}$$

for $i = 1, 2, \ldots, n$. For a given linearized message polynomial $f_{\boldsymbol{u}}(X)$, our encoder constructs the vector v_i's as follows:

$$v_i = \big(\alpha_i, f_{\boldsymbol{u}}(\alpha_i), f_{\boldsymbol{u}}^{\otimes 2}(\alpha_i), \ldots, f_{\boldsymbol{u}}^{\otimes L}(\alpha_i)\big)$$

for $i = 1, 2, \ldots, n$. Then it outputs the n-dimensional vector space spanned by vectors $v_1, v_2, \ldots, v_n$. In this construction, the ambient space $\mathcal{W}$ has dimension $n + nmL$ and

each element in $\mathcal{W}$ is represented as a vector with $L+1$ coordinates $(x, y_1, y_2, \ldots, y_L)$, where x belongs to the vector space spanned by $\alpha_1, \alpha_2, \ldots, \alpha_n$ and $y_i \in \mathbb{F}_{q^{nm}}$, for $i = 1, 2, \ldots, L$. The decoding algorithm consists of three steps. In the first step, it computes the interpolation points. In the second step, a multivariate linearized polynomial $Q(X, Y_1, Y_2, \ldots, Y_L)$ of the form

$$Q_0(X) + Q_1(Y_1) + Q_2(Y_2) + \cdots + Q_L(Y_L)$$

is constructed, where each Q_i is subject to a certain degree constraint, such that

$$Q(x, y_1, y_2, \ldots, y_L) = 0$$

for all the interpolation points $(x, y_1, y_2, \ldots, y_L)$. Then in the factorization step, we compute all the roots $f(X) \in \mathscr{L}_q[X]$, with degree of at most $k-1$, of the equation:

$$Q\big(X, f(X), \ldots, f^{\otimes L}(X)\big) = 0$$

To solve this equation efficiently, in Section 2.6, we proposed a linearized version of the Roth-Ruckenstein algorithm, which was designed to solve equations over the ring of polynomials [9]. We also prove in Theorem 2.3.2 that there are at most L solutions for $f(X) \in \mathscr{L}_q[X]$. Each solution corresponds to one possible output message.

The final result is expressed in terms of the error-correction radius of this code and the corresponding list-decoding algorithm. We proved in Section 2.4 that our list-decoding algorithm successfully recovers the message polynomial as long as

$$\frac{t}{n} < L - \frac{1}{2}L(L+1)R^* \tag{4.4}$$

where t is the dimension of the error space added to the transmitted codeword. Our list-decoding algorithm can also correct erasures. Each erasure, however, costs equivalent to L errors.

We further improve this result by introducing multiplicities for the interpolation polynomial in Chapter 3. First, we establish the notion of multiplicity for linearized polynomials in this context. Then by enforcing multiple roots for the interpolation polynomial, we manage to achieve a better error-correction radius. We are also able to list-decode at higher rates. For every positive integer L and r, our list-L decoder with

multiplicity r guarantees successful recovery of the message subspace provided that the normalized error-correction radius $\tau = t/n$ satisfies

$$\tau < \frac{2(L+1)}{r+1} - 1 - \frac{L(L+1)}{r(r+1)}R^*$$

This improves the normalized error-correction radius over the previous results, given in (4.2) and (4.4), for a wide range of rates. The multiplicity parameter r is independent of the code construction and can be chosen at the decoder in such a way that the error-correction radius is maximized. As L tends to infinity, the bound on the error-correction radius of our construction with a suitable choice of r approaches $\frac{1}{R^*} - 1$.

4.3 New Subspace Codes and Algebraic List-decoding Thereof

In this section, we present a new construction of subspace codes and a list-decoding algorithm capable of correcting both errors and erasures. Our results in this section are motivated by the recent work of Vadhan [11, Ch. p] and Guruswami [2]. Then we establish the correctness of our algorithm and compute the error correction radius of the proposed construction.

4.3.1 Code Construction and List-decoding Algorithm

The following parameters of the construction are fixed: the finite field $\mathbb{F}_q$ and an extension $\mathbb{F} = \mathbb{F}_{q^m}$, the number of information symbols k, the dimension of code n and the parameter s which is related to the list size. We require that $k \leqslant n \leqslant m$. A set $A = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ of linearly independent elements of $\mathbb{F}_{q^m}$ is also fixed. In this construction, the ambient space $\mathcal{W}$ is an $n+sm$-dimensional vector space over $\mathbb{F}_q$. Let γ be an element of $\mathbb{F}_{q^m}$ which is not contained in any subfield of $\mathbb{F}_{q^m}$ i.e. $\gamma, \gamma^q, \ldots, \gamma^{q^{m-1}}$ are all distinct.

Encoding Algorithm:

Formally, the encoder is a function $\mathcal{E} : \mathbb{F}^k \to \mathcal{G}(W, n)$. It accepts as input a message $\boldsymbol{u} = (u_0, u_1, \ldots, u_{k-1}) \in \mathbb{F}^k$. The corresponding message polynomial is $f_{\boldsymbol{u}}(X) =$

$\sum_{i=0}^{k-1} u_i X^{[i]}$. Then the corresponding codeword V is the $\mathbb{F}_q$-linear span of the set $\big\{\big(\alpha_i, f(\alpha_i), f(\gamma\alpha_i), \ldots, f(\gamma^{s-1}\alpha_i)\big) : i \in [n]\big\}$.

Notice that, Koetter-Kschischang code is a special case of this for $s = 1$. Since the α_i's are linearly independent, each codeword is an n-dimensional vector space which is a subspace of

$$W = \langle \alpha_1, \alpha_2, \ldots, \alpha_n \rangle \oplus \underbrace{\mathbb{F}_{q^m} \oplus \cdots \oplus \mathbb{F}_{q^m}}_{s \text{ times}} \tag{4.5}$$

The dimension of $\mathcal{W}$ is equal to $n + sm$, as mentioned before. Each element in $\mathcal{W}$ is represented as a vector with $s + 1$ coordinates such as $(x, y_1, \ldots, y_s)$, where x is an element of the vector space spanned by $\alpha_1, \alpha_2, \ldots, \alpha_n$ and all y_i's belong to $\mathbb{F}_{q^m}$.

Now, we explain the list-decoding algorithm. Suppose that V is transmitted and a subspace U of $\mathcal{W}$ of dimension d is received. We need another parameter ω at the decoder which is computed as follows:

$$\omega = \left\lceil \frac{d + s(k-1) + 1}{s+1} \right\rceil \tag{4.6}$$

As we will see, ω is chosen in such a way that existence of the interpolation polynomial is guaranteed at the decoder.

List-decoding Algorithm:

The decoder accepts as input a vector space U which is a subspace of $\mathcal{W}$. It then outputs a list of size at most $q^{m(s-1)}$ of vectors in $\mathbb{F}^k$ in three steps:

1. *Computing the interpolation points:*
   Find a basis $(x_i, y_{i,1}, y_{i,2}, \ldots, y_{i,s}), i = 1, 2, \ldots, d$, for U. This is the set of interpolation points.

2. *Interpolation:*
   Construct a non-zero multivariate linearized polynomial

   $$Q(X, Y_1, Y_2, \ldots, Y_s) = Q_0(X) + Q_1(Y_1) + Q_2(Y_2) + \cdots + Q_s(Y_s)$$

   where Q_i's are linearized polynomials over $\mathbb{F}_{q^m}$, Q_0 has q-degree of at most $\omega - 1$ and Q_i has q-degree of at most $\omega - k$, for $i = 1, 2, \ldots, s$, subject to the constraint that

   $$Q(x_i, y_{i,1}, y_{i,2}, \ldots, y_{i,s}) = 0 \text{ for } i = 1, 2, \ldots, d \tag{4.7}$$

3. *Message recovery:*

   Find all polynomials $f(X) \in \mathscr{L}_{q^m}[X]$ of degree at most $k-1$ that satisfy the following equation

$$Q\big(X, f(X), f(\gamma X), \ldots, f(\gamma^{s-1}X)\big) = 0$$

   The decoder outputs coefficients of each solution $f(X)$ as a vector of length k.

The first step of this list-decoding algorithm can be done using elementary linear algebraic operations. The second step is basically solving a system of linear equations. There are several ways for doing that. The most straightforward way is the Gaussian elimination method. However, this method does not take advantage of the structure of this system of equations, and therefore it is not efficient. Efficient interpolation algorithms in the ring of linearized polynomials are presented in [12]. In this case, the complexity of the corresponding interpolation algorithm is given as $O(n^2 s^3)$ field operations over $\mathbb{F}_{q^m}$. The parameter s is in fact a design parameter and can be regarded as a constant. Indeed, the interpolation step is quadratic in terms of n. In the next subsection, we explain how the message recovery step can be carried out using a linear algebraic method. The complexity of the message recovery step is also quadratic. Hence, the total complexity of our algorithm is quadratic in terms of n, the dimension of the code.

4.3.2 Recovering the Message Polynomial

As discussed in the foregoing section, in the last step of the list-decoding algorithm we need to find all polynomials $f(X) \in \mathscr{L}_{q^m}[X]$ of degree at most $k-1$ that satisfy

$$Q_0(X) + Q_1\big(f(X)\big) + Q_2\big(f(\gamma X)\big) + \cdots + Q_s\big(f(\gamma^{s-1}X)\big) = 0 \tag{4.8}$$

Remark. Suppose that $f, g \in \mathscr{L}_{q^m}[X]$ are two solutions to the equation (4.8). Since Q_i's are linearized polynomials, for any $\alpha \in \mathbb{F}_q$, $\alpha f + (1-\alpha)g$ is also a solution to (4.8). Therefore, the set of solutions, which can be regarded as vectors of length k over $\mathbb{F}_{q^m}$, forms an affine subspace of $\mathbb{F}^k$ as a vector space over $\mathbb{F}_q$. □

In the next lemma, we establish an upperbound on the number of solutions to (4.8). The proof of this lemma also clarifies how the affine space of solutions can be computed with quadratic complexity.

Lemma 4.3.1. *The dimension of the affine space of solutions $f(X) \in \mathscr{L}_{q^m}[X]$, of degree at most $k-1$, to (4.8) is at most $m(s-1)$.*

Proof. For $i = 0, 1, 2, \ldots, s$, let

$$Q_i(X) = \sum_{j \geqslant 0} q_{i,j} X^{q^j}$$

If $q_{i,0} = 0$ for $i = 0, 1, 2, \ldots, s$, then we replace Q_i with Q'_i, where $Q_i(X) = Q'_i(X^q)$, in (4.8) and the space of solutions remains unchanged. Therefore, one can assume that at least one $q_{i^*,0}$ is non-zero for some $i^* \in \{0, 1, 2, \ldots, s\}$. Also, if $q_{1,0}, q_{2,0}, \ldots, q_{s,0}$ are all zero, then so is $q_{0,0}$, otherwise there is no solution to (4.8). Thus, we can take i^* from the set $\{1, 2, \ldots, s\}$.

Let us define the linearized polynomial $P(X)$ as

$$P(X) = Q_0(X) + \sum_{i=1}^{s} Q_i\big(f(\gamma^{i-1}X)\big)$$

and the polynomial $A(X)$ as

$$A(X) = q_{1,0} + q_{2,0}X + \cdots + q_{s,0}X^{s-1}$$

Then the coefficient of X^{q^i} in $P(X)$, for $i = 0, 1, \ldots, k-1$, is equal to

$$\begin{aligned} q_{0,i} \;+\; & u_i\big(q_{1,0} + q_{2,0}\gamma^{q^i} + \cdots + q_{s,0}\gamma^{(s-1)q^i}\big) \\ +\; & u_{i-1}^{q}\big(q_{1,1} + q_{2,1}\gamma^{q^i} + \cdots + q_{s,1}\gamma^{(s-1)q^i}\big) \\ +\cdots+\; & u_0^{q^i}\big(q_{1,i} + q_{2,i}\gamma^{q^i} + \cdots + q_{s,i}\gamma^{(s-1)q^i}\big) \end{aligned}$$

which can be simply expressed as

$$q_{0,i} + A(\gamma^{q^i})u_i + \sum_{j=0}^{i-1} a_j^{(i)} u_j^{q^{i-j}} \tag{4.9}$$

for some elements $a_j^{(i)} \in \mathbb{F}_{q^m}$. Now, suppose we want to find all possible solutions for $f(X)$ in (4.8). Then all the coefficients of $P(X)$ have to be equal to zero. In particular, for the coefficient of X in $P(X)$:

$$A(\gamma)u_0 + q_{0,0} = 0$$

If $A(\gamma)$ is non-zero, then $u_0 = -\frac{q_{0,0}}{A(\gamma)}$. If $A(\gamma)$ is zero but $q_{0,0}$ is not zero, then there is no solution for u_0 and consequently for $f(X)$. If both $A(\gamma)$ and $q_{0,0}$ are zero, then we can set u_0 to any element of $\mathbb{F}_{q^m}$. Then we find the solutions to u_i's iteratively. For each i, suppose that $u_0, u_1, \ldots, u_{i-1}$ are already computed. If $A(\gamma^{q^i})$ is non-zero, then u_i can be uniquely determined by (4.9). Otherwise, we take all the elements of $\mathbb{F}_{q^m}$ as possible solutions to u_i and keep going for each of them separately. Notice that $A(X)$ is a non-zero polynomial of degree $s-1$ and $\gamma, \gamma^q, \ldots, \gamma^{q^{k-1}}$ are all distinct elements of $\mathbb{F}_{q^m}$. Therefore, $A(\gamma^{q^i})$ is equal to zero for at most $s-1$ possible values of i. This implies that the total number of solutions for $f(X)$ to (4.8) is at most $q^{m(s-1)}$ which proves the lemma. ■

Corollary 4.3.2. *The affine space of solutions to (*4.8*) can be computed with quadratic complexity in terms of dimension* n*.*

4.3.3 Correctness of the Algorithm and Code Parameters

In this subsection, we first establish the correctness of our list-decoding algorithm. Then we consider the error-correction capability of our scheme.

Lemma 4.3.3. *The particular choice of* ω *in (*4.6*) guarantees the existence of a non-zero solution for the interpolation polynomial* Q *that satisfies (*4.7*).*

Proof. (4.7) defines a homogeneous system of d linear equations. The number of unknown coefficients is equal to

$$\omega + (\omega - k + 1)s = \omega(s+1) - s(k-1)$$

A non-zero solution for this homogeneous system of linear equations is guaranteed if the number of equations is strictly less than the number of variables. i.e.

$$d \leqslant \omega(s+1) - s(k-1) - 1 \Leftrightarrow$$
$$\omega \geqslant \frac{d + s(k-1) + 1}{s+1}$$

This is guaranteed by the choice of ω in (4.6). ■

We form the following linearized polynomial $E(X)$ wherein $f_{\boldsymbol{u}}(X)$ is the message polynomial and $Q(X, Y_1, \ldots, Y_L)$ is the interpolation polynomial provided by the list-decoding algorithm.

$$\begin{aligned} E(X) &= Q\big(X, f_{\boldsymbol{u}}(X), f_{\boldsymbol{u}}(\gamma X), \ldots, f_{\boldsymbol{u}}(\gamma^{s-1} X)\big) \\ &= Q_0(X) + \sum_{i=1}^{s} Q_i \otimes f_{\boldsymbol{u}}(\gamma^{i-1} X) \end{aligned}$$

Let ρ and t denote the number of erasures and errors in the received subspace U, respectively. Hence, the dimension of U is in fact equal to $d = n - \rho + t$.

Lemma 4.3.4. *The linearized polynomial $E(X)$ has at least $n - \rho$ linearly independent roots in $\mathbb{F}_{q^m}$.*

Proof. Let U' denote the intersection of the transmitted codeword V and the received subspace U. Then U' is a subspace of the received vector space U with dimension $n - \rho$. Since Q is a linearized polynomial

$$Q(x, y_1, \ldots, y_s) = 0$$

for any $(x, y_1, \ldots, y_s) \in U'$. On the other hand, $(x, y_1, \ldots, y_s)$ is also an element of the transmitted codeword V. Therefore,

$$(x, y_1, \ldots, y_s) = \big(\beta, f_{\boldsymbol{u}}(\beta), f_{\boldsymbol{u}}(\gamma\beta), \ldots, f_{\boldsymbol{u}}(\gamma^{s-1}\beta)\big)$$

for some β in the linear span of $\alpha_1, \alpha_2, \ldots, \alpha_n$. Therefore, β is a root for the polynomial $E(X)$. Hence, there are at least $n - \rho$ linearly independent roots for $E(X)$. ■

Corollary 4.3.5. *If $\omega \leqslant n - \rho$, then the linearized polynomial $E(X)$ is identically zero.*

Proof. The q-degree of $f_{\boldsymbol{u}}(X)$ is at most $k-1$. Therefore, the q-degree of $Q_i \otimes f_{\boldsymbol{u}}(\gamma^{i-1} X)$ is at most

$$\omega - k + k - 1 = \omega - 1$$

for $i = 1, \ldots, L$. Also, the q-degree of $Q_0(X)$ is at most $\omega - 1$. Thus the q-degree of $E(X)$ is at most $\omega - 1$. On the other hand, $E(X)$ has at least $n - \rho$ linearly independent roots by Lemma 4.3.4. Therefore, $E(X)$ must be the all zero polynomial. ■

Theorem 4.3.6. *The output of our list-decoding algorithm is a list of size at most* $q^{m(s-1)}$ *which includes the transmitted message* $\boldsymbol{u}$ *provided that*

$$s\rho + t < ns - s(k-1) \tag{4.10}$$

Proof. The existence of non-zero interpolation polynomial Q that satisfies (4.7) is guaranteed by Lemma 4.3.3. Then by Corollary 4.3.5, $E(X)$ is the all zero polynomial provided that

$$\left\lceil \frac{d + s(k-1) + 1}{s+1} \right\rceil \leqslant (n - \rho) \tag{4.11}$$

where we have used the expression for ω from (4.6). We plug in $d = n - \rho + t$ into (4.11). Then observe that (4.11) is in fact equivalent to

$$s\rho + t < ns - s(k-1)$$

Thus this condition on the number of errors and erasures implies that $E(X)$ is identically zero. Therefore, the message polynomial $f_{\boldsymbol{u}}(X)$ is a solution to (4.8). There are at most $q^{m(s-1)}$ solutions to (4.8) by Lemma 4.3.1. Therefore, the list size is at most $q^{m(s-1)}$. ■

Now, we turn our attention to the parameters of the proposed construction. The ambient space $\mathcal{W}$ is given in (4.5) which has dimension equal to $n + sm$. The symbol rate R of the code can be computed as defined in (4.1):

$$R = \frac{\log_q(\text{size of the code})}{n(\dim(W))} = \frac{km}{n(n+sm)}$$

We define the error-correction radius τ as

$$\tau = \frac{t + s\rho}{n}$$

In fact, the cost of each erasure is equal to the cost of s errors. By Theorem 4.3.6, our list-decoding algorithm successfully recovers the transmitted message as long as

$$\tau < s - \frac{s(k-1)}{n} \approx s - s^2(1 + \frac{n}{ms})R \tag{4.12}$$

In the regime where n is much smaller than ms, the bound on the error-correction radius can be approximated as $s - s^2R$.

Acknowledgements

This chapter is in part a reprint of the material that is going to be presented at ISIT 2012 in Boston and is also available on arxiv as: H. Mahdavifar and A. Vardy, "List-decoding of Subspace Codes and Rank-Metric Codes up to Singleton Bound," available at http://arxiv.org/pdf/1202.0866.pdf.

Bibliography

[1] R. Ahlswede, N. Cai, Sh. Y. R. Li and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, pp. 1204–1216, Jul. 2000.

[2] V. Guruswami, "Linear-algebraic list decoding of folded Reed-Solomon codes," *Proc. of the 26th IEEE Conference on Computational Complexity (CCC)*, pp. 77-85, 2011.

[3] V. Guruswami and A. Rudra, "Explicit codes achieving list decoding capacity: Error-correction up to the Singleton bound." *IEEE Transactions on Information Theory*, vol. 54, pp.İ35150, Jan. 2008.

[4] R. Koetter and F.R. Kschischang, "Coding for errors and erasures in random network coding." *IEEE Transactions on Information Theory*, vol. 54, pp. 3579–3591, Aug. 2008.

[5] F. J. MacWilliams and N. J. A. Sloane, "The theory of error-correcting codes." New York: North Holland, 1977

[6] H. Mahdavifar and A. Vardy "Algebraic list-decoding on the operator channel," *Proc. IEEE Intern. Symp. Inf. Theory*, pp. 1193–1197, Austin. TX., June 2010

[7] H. Mahdavifar and A. Vardy "Algebraic list-decoding of subspace codes," *IEEE Transactions on Information Theory*, accepted for publication, available at http://arxiv.org/abs/1202.0338.

[8] F. Parvaresh and A. Vardy, "Correcting errors beyond the Guruswami-Sudan radius in poly- nomial time," *Proc. of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pp. 285294, 2005.

[9] R. M. Roth, G. Ruckenstein, "Efficient decoding of Reed-Solomon codes beyond half the minimum distance," *IEEE Transactions on Information Theory*, vol. 46, pp. 246–257, Jan. 2000.

[10] M. Sudan, "Decoding of Reed-Solomon codes beyond the error-correction bound," *J. Complexity,* 12: 180–193, Mar. 1997

[11] S. Vadhan, "Pseudorandomness." Foundations and Trends in Theoretical Computer Science (FnT-TCS). NOW publishers, 2010. To appear. Draft available at http://people.seas.harvard.edu/salil/pseudorandomness.

[12] H. Xie, Z. Yan, B. W, Suter, "General linearized polynomial interpolation and its applications," *2011 Int. Symp. on Network Coding,* Beijing, China, July 25-27 2011

Chapter 5

List-Decoding of Rank-Metric Codes

5.1 Introduction

In rank-metric codes, each codeword is a matrix with fixed dimensions whose entries are taken from a finite field $\mathbb{F}_q$. The distance between any two matrices is defined as the rank of their difference. There is indeed a close relation between rank-metric codes and subspace codes. In [13], Silva et al. show that there is an injective mapping between rank-metric codes and subspace codes through a *lifting* operation. Gabidulin codes were introduced as a class of MRD (maximum rank-distance) codes [6]. They achieve the Singleton bound on the minimum rank distance of a rank-metric code. In Gabidulin codes, the rows of each codeword are evaluations of a *linearized message polynomial* over certain fixed points. In fact, Koetter-Kschischang subspace codes are the image of Gabidulin codes through a lifting operation defined in [13]. In this chapter, we consider the problem of list-decoding of rank-metric codes.

There are various applications of rank-metric codes addressed by Roth in [11]. He refers to the error patterns confined to a particular number t of rows, or columns, that may happen to an array of symbols as *crisscross* errors. Crisscross errors can happen in memory chip arrays, where row or column failures occur because of the mulfunctioning of row drivers, or column amplifiers (for example, see [5] and [9]). In magnetic recording applications, where the errors usually occur along the tracks and information is recorded across the tracks, we can model the errors as crisscross errors and use rank-metric codes to deal with errors. Recently, rank-metric codes have received a lot of

attention as a suitable tool for error-correction in *coherent* network coding [12,13].

In coherent network coding, the topology of the network and the underlying network code is known at the source and receivers of the network. Suppose that there is a source node that transmits n packets through the network. Each packet is regarded as a vector of length m over a finite field $\mathbb{F}_q$. Each intermediate node in the network receives some packets through its incoming edges, linearly combines them, and then sends the result out on its outgoing edges. There are one or more destination nodes that try to obtain the message transmitted by the source node. This is the scenario in a multicast model for linear network coding. Let the rows of $\boldsymbol{X} \in \mathbb{F}_q^{n\times m}$ denote the transmitted packets by the source. In the error-free case, at a particular receiver, the received packets can be represented as rows of an $N \times m$ matrix $\boldsymbol{Y} = \boldsymbol{AX}$, where $\boldsymbol{A}$ is the transfer matrix of the network from the source to that particular receiver. Suppose that we allow up to t error packets to be injected into the network. Then the received matrix $\boldsymbol{Y}$ can be written as

$$\boldsymbol{Y} = \boldsymbol{AX} + \boldsymbol{BZ} \tag{5.1}$$

where $\boldsymbol{Z}$ is a $t \times m$ matrix whose rows represent the error packets, and $\boldsymbol{B}$ is the transform matrix from the error packets to the receiver. The error correcting problem can be viewed in various ways depending on certain assumptions about which of the parameters are known.

Cai and Yeung initiated the error correcting problem for coherent network coding in [1]. They established some fundamental bounds in [2] and [3]. However, a different approach can be taken to the problem as in [12]. We assume a *pessimistic* situation wherein an adversary injects up to t packets into the network while it is also free to choose the transfer matrix $\boldsymbol{B}$. Essentially, we consider the model with input matrix $\boldsymbol{X} \in \mathbb{F}_q^{n\times m}$, output matrix $\boldsymbol{Y} \in {\mathbb{F}_q}^{N\times m}$, fixed transfer matrix $\boldsymbol{A} \in {\mathbb{F}_q}^{N\times n}$, and $\boldsymbol{B} \in {\mathbb{F}_q}^{N\times t}$ and $\boldsymbol{Z} \in {\mathbb{F}_q}^{t\times m}$ are chosen by the adversary. The parameter t is the maximum number of linearly independent error packets injected by the adversary into the network. Silva and Kschischang in [12] show that the pessimistic assumption on the error model actually incurs no penalty since maximum rank-distance codes indeed achieve the Singleton bound derived by Yeung and Cai in [2]. In fact, this approach suggests a universality in the sense that the outer rank-metric code for network error correction and the underlying

network code can be designed independently.

In this chapter, we introduce folded version of Gabidulin codes, a family of maximum rank-distance codes, that allows a simple linear-algebraic list-decoding by using $s+1$-variate interpolation polynomials, where s is a design parameter. All the steps of this list-decoding algorithm are done by linear-algebraic methods. A system of linear equations is solved for the interpolation step and another linear system is solved to compute the set of all the possible solutions which indeed is a linear space. This is motivated by the recent work of Vadhan [14, Ch. 5] and Guruswami [7] which suggested a simplified version, with no need for multiplicity, of the previously proposed list-decoding algorithm of folded Reed-Solomon codes by Guruswami and Rudra in [8]. The later was built upon the work of Parvaresh and Vardy on list-decoding of Reed-Solomon codes by proposing multivariate interpolation [10].

This chapter is organized as follows. We first give an overview of rank-metric codes and Gabidulin codes in 5.2. In Section 5.3, we introduce the folded version of Gabidulin codes and provide the list-decoding algorithm. Then we show that we are able to correct the fraction of errors up to $1-R$, R being the rate of the code, hence achieving the Singleton upper bound on the error-correction radius which is the information theoretic upper bound on the error-correction radius of rank-metric codes.

5.2 Background

5.2.1 Rank-Metric Codes

Let $\mathbb{F}_q^{n\times m}$ denote the set of all $n \times m$ matrices over $\mathbb{F}_q$. The distance between $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{F}_q^{n\times m}$ is defined as rank $(\boldsymbol{X}-\boldsymbol{Y})$. $\mathbb{F}_q^{n\times m}$ is a metric space with this distance metric. It is easy to verify all the conditions. This metric is clearly symmetric and also if $\boldsymbol{X}=\boldsymbol{Y}$, then the distance is 0. Furthermore, for any $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z} \in \mathbb{F}_q^{n\times m}$,

$$\text{rank}(\boldsymbol{X}-\boldsymbol{Y}) + \text{rank}(\boldsymbol{X}-\boldsymbol{Z}) \geqslant \text{rank}(\boldsymbol{Z}-\boldsymbol{Y})$$

That is because rank satisfies subadditivity: $\text{rank}(A+B) \leqslant \text{rank}(A) + \text{rank}(B)$ for any two matrices A and B of the same size.

For any $\boldsymbol{X} \in \mathbb{F}_q^{n\times m}$, let $\langle \boldsymbol{X} \rangle$ denote the row space of the matrix $\boldsymbol{X}$. A rank-metric code $\mathbb{C}$ is just a subset of $\mathbb{F}_q^{n\times m}$ which is called an array code in [11]. We define the rate R of a rank-metric code $\mathbb{C} \subseteq \mathbb{F}_q^{n\times m}$ as follows:

$$R \stackrel{\text{def}}{=} \frac{\log_q(|\mathbb{C}|)}{nm} \tag{5.2}$$

The minimum (rank) distance of $\mathbb{C}$ is the minimum distance between distinct elements of $\mathbb{C}$. A rich coding theory is developed for rank-metric codes analogous to the classical block codes with Hamming metric in [4] and [6]. In particular, we state a Singleton bound that is established in the context of rank-metric codes by Gabidulin in [6]:

Theorem 5.2.1. *[6] A rank-metric code $\mathbb{C} \subseteq \mathbb{F}_q^{n\times m}$ with minimum distance d must satisfy*

$$\log_q\big(|\mathbb{C}|\big) \leqslant \min\{n(m-d+1), m(n-d+1)\}$$

Proof. The proof is very similar to the proof of the Singleton bound for block codes. We map each codeword $\boldsymbol{X} \in \mathbb{C}$ to an element of ${\mathbb{F}_q}^{n\times(m-d+1)}$ by erasing the last $d-1$ columns of $\boldsymbol{X}$. We claim that this is an injective mapping. Assume, to the contrary that, $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{C}$ map to the same element of ${\mathbb{F}_q}^{n\times(m-d+1)}$. This implies that the first $m-d+1$ columns of $\boldsymbol{X}$ and $\boldsymbol{Y}$ are equal. Therefore, $\boldsymbol{X}-\boldsymbol{Y}$ may have non-zero entries only in its last $d-1$ columns. Hence, its rank is at most $d-1$ which contradicts the assumption that the minimum distance of $\mathbb{C}$ is d. Therefore, this mapping is injective which immediately implies that

$$|\mathbb{C}| \geqslant \left|{\mathbb{F}_q}^{n\times(m-d+1)}\right| = q^{n(m-d+1)}$$

By erasing the last $d-1$ rows of elements of $\mathbb{C}$ and using the same argument we can prove that

$$|\mathbb{C}| \geqslant \left|{\mathbb{F}_q}^{(n-d+1)\times m}\right| = q^{(n-d+1)m}$$

which completes the proof of the theorem. ■

A rank-metric code that meets the Singleton bound on the minimum distance is called a maximum rank-distance (MRD) code. Gabidulin codes are a class of MRD codes proposed in [6].

Rank-metric codes in $\mathbb{F}_q^{n\times m}$ can be constructed as block codes of length n over the extension field $\mathbb{F}_{q^m}$. In other words, we fix a basis for $\mathbb{F}_{q^m}$, as a vector space of

dimension m, over the base field $\mathbb{F}_q$. Then we regard each symbol in $\mathbb{F}_{q^m}$ as a row vector of length m over $\mathbb{F}_q$. Hence, any codeword of length n over $\mathbb{F}_{q^m}$ is regarded as an $n \times m$ matrix over $\mathbb{F}_q$.

5.2.2 Gabidulin Codes

Linearized polynomials play an important role in the construction of Gabidulin codes just as in the construction of Koetter-Kschischang subspace codes. We reviewed the ring of linearized polynomials and their properties in Section 1.4. Linearized polynomials are a family of polynomials which act as linear maps with respect to a certain base field $\mathbb{F}_q$. A polynomial over an extension field $\mathbb{F}_{q^m}$ of $\mathbb{F}_q$ is called $\mathbb{F}_q$-linearized if it has the form

$$f(X) = \sum_{i=0}^{s} a_i X^{q^i}$$

We say that the polynomial $f(X)$ has q-degree s assuming that $a_s \neq 0$. X^{q^i} is denoted by $X^{[i]}$, when q is fixed under discussion. The fundamental property of linearized polynomials is that they act as linear maps with respect to the base field $\mathbb{F}_q$. For any two linearized polynomials $f_1(X)$ and $f_2(X)$, the summation $f_1(X) + f_2(X)$ is clearly linearized. However, the product $f_1(X)f_2(X)$ is not necessarily a linearized polynomial. In order to have a ring structure, the composition operation $f_1(X) \otimes f_2(X)$ is defined to be the composition $f_1(f_2(X))$ which is always a linearized polynomial.The set of linearized polynomials forms a non-commutative ring under addition $+$ and composition operation $\otimes$. We denote the ring of linearized polynomials over $\mathbb{F}_{q^m}$ by $\mathscr{L}_{q^m}[X]$.

As we discussed in the previous subsection, rank-metric codes in $\mathbb{F}_q^{n\times m}$ can be regarded as block codes over $\mathbb{F}_{q^m}$. A Gabidulin code in $\mathbb{F}_q^{n\times m}$ is a linear (n, k) block code over $\mathbb{F}_{q^m}$ with the following parity check matrix:

$$\begin{bmatrix} h_1^{[0]} & h_2^{[0]} & \dots & h_n^{[0]} \\ h_1^{[1]} & h_2^{[1]} & \dots & h_n^{[1]} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ h_1^{[n-k-1]} & h_2^{[n-k-1]} & \dots & h_n^{[n-k-1]} \end{bmatrix}$$

where the parameters $h_1, h_2, \ldots, h_n \in \mathbb{F}_{q^m}$ are picked arbitrarily as long as they are linearly independent. Hence, we require that $n \leqslant m$. Furthermore, given the parity check matrix of the Gabidulin codes, it can be shown that the corresponding generator matrix G has the following form:

$$\begin{bmatrix} \alpha_1^{[0]} & \alpha_2^{[0]} & \ldots & \alpha_n^{[0]} \\ \alpha_1^{[1]} & \alpha_2^{[1]} & \ldots & \alpha_n^{[1]} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ \alpha_1^{[k-1]} & \alpha_2^{[k-1]} & \ldots & \alpha_n^{[k-1]} \end{bmatrix}$$

where the elements $\alpha_1, \alpha_2, \ldots, \alpha_n \in \mathbb{F}_{q^m}$ are linearly independent over $\mathbb{F}_q$. The rate of Gabidulin code, as a rank-metric code, can be computed using the definition given in (5.2). The rate is k/n which is in fact equal to the rate of the corresponding (n, k) block code.

Suppose that the input to the Gabidulin encoder is a message vector

$$\boldsymbol{u} = [u_0\, u_1\, \ldots\, u_{k-1}]$$

which consists of k message symbols in $\mathbb{F}_{q^m}$. Let $f_{\boldsymbol{u}}(X)$ denote the corresponding linearized message polynomial $\sum_{i=0}^{k-1} u_i X^{[i]}$. Then the corresponding codeword $\boldsymbol{V} = (\boldsymbol{u}G)^T$ is indeed equal to

$$\left[f_{\boldsymbol{u}}(\alpha_1)\, f_{\boldsymbol{u}}(\alpha_2)\, \ldots\, f_{\boldsymbol{u}}(\alpha_n)\right]^T$$

which can be also regarded as a matrix in $\mathbb{F}_q^{n\times m}$. The following lemma proves that Gabidulin codes are a class of maximum rank distance codes.

Lemma 5.2.2. *The minimum rank distance of the Gabidulin code in $\mathbb{F}_q^{n\times m}$ with rate $R = k/n$, is $n - k + 1$.*

Proof. Suppose that there are two linearized polynomials $f(X), g(X) \in \mathscr{L}_{q^m}[X]$, with q-degree at most $k - 1$, such that the distance between their corresponding codewords is at most $n - k$. Let $h(X) = f(X) - g(X)$. Then the rank of the matrix

$$\boldsymbol{H} = \left[h(\alpha_1)\, h(\alpha_2)\, \ldots\, h(\alpha_n)\right]^T,$$

which is regarded as an element in $\mathbb{F}_q^{n\times m}$, is at most $n-k$. Therefore, the dimension of the null space of $\boldsymbol{H}$ is at least k. Observe that each element in the null space of $\boldsymbol{H}$ corresponds to a root of $h(X)$. Therefore, the dimension of the root space of $h(X)$ is at least k. On the other hand, the q-degree of $h(X)$ is at most $k-1$. Therefore, $h(X)$ must be identically zero which means that $f(X)$ and $g(X)$ must be identically equal. This completes the proof of the lemma. ■

Corollary 5.2.3. *Gabidulin codes are a family of maximum rank-distance codes.*

Proof. The proof follows from Lemma 5.2.2, Theorem 5.2.1 and the fact that $n \leqslant m$ in the construction of Gabidulin codes. ■

Notice that when $n \leqslant m$, which is always the case in the construction of Gabidulin codes, the Singleton bound on the minimum distance d of the rank-metric code $\mathbb{C}$ reduces to

$$\log_q\big(|\mathbb{C}|\big) \leqslant m(n-d+1)$$

which can be normalized to

$$R = \frac{\log_q\big(|\mathbb{C}|\big)}{nm} \leqslant 1 - \frac{d-1}{n}$$

Hence, $1-R$ is the bound on the normalized minimum distance of the code $\mathbb{C}$, when we normalize it by the number of rows n. This is also the information-theoretic bound on the error-correction radius of the code $\mathbb{C}$. Furthermore, the unique decoding radius bound is $(1-R)/2$. A decoding algorithm which can correct errors as long as the rank of error is less than $(d-1)/2$, is proposed in [6], hence achieving the bound $(1-R)/2$ on a unique decoding radius.

5.3 List-decoding of Gabidulin Codes

In this section, we first introduce a folded version of Gabidulin codes. Then, we propose a list-decoding algorithm which provides the decoding radius up to the Singleton bound $1-R$, the best possible trade-off between the rate and the error-correction radius.

Let γ be a primitive element of $\mathbb{F}_{q^m}$. Let $\mathbb{C}$ denote the Gabidulin code constructed with parameters $\alpha_i = \gamma^{[i-1]}$ as discussed in Section 5.2.1. Let also h be a positive integer that divides n. Then let $N = n/h$ and $M = hm$.

Definition 5.3.1. *(Folded Gabidulin Code)*
The h-folded version of the Gabidulin code $\mathbb{C}$ is a rank-metric code whose codewords are elements of ${\mathbb{F}_q}^{N\times M}$. The message polynomial $f_{\boldsymbol{u}}(X)$ of q-degree of at most $k-1$ is encoded into a matrix with the h-tuple $\big(f_{\boldsymbol{u}}(\gamma^{ih}), f_{\boldsymbol{u}}(\gamma^{ih+1}), \ldots, f_{\boldsymbol{u}}(\gamma^{(i+1)h-1})\big)$, which is regarded as an element in ${\mathbb{F}_q}^M$, as its i-th row, for $0 \leqslant i < N$. The rate of the folded version of $\mathbb{C}$ is k/n, equal to the rate of original code $\mathbb{C}$.

Notice that folding does not change the rate. The rate of the folded version of code $\mathbb{C}$ is equal to the rate of $\mathbb{C}$ which is equal to k/n.

Before going into the details of the list-decoding algorithm, we would like to clarify the difference between the notion of "error" in subspace codes and rank-metric codes. Suppose that a codeword $\boldsymbol{X}$ in code $\mathbb{C}$ is transmitted and a word $\boldsymbol{Y}$ with t errors is received i.e. $\text{rank}(\boldsymbol{X} - \boldsymbol{Y}) = t$. Now consider $\langle \boldsymbol{X} \rangle$ and $\langle \boldsymbol{Y} \rangle$ in the context of subspace codes. Then $\langle \boldsymbol{Y} \rangle$ is corrupted with t errors and t erasures with respect to $\langle \boldsymbol{X} \rangle$. In fact, in rank-metric codes, there is no notion of "erasure" and each error corresponds to one error together with one erasure in the context of subspace codes.

For $0 \leqslant i \leqslant N-1$ and $0 \leqslant j \leqslant h-1$, let $y_{i,j} \in \mathbb{F}_{q^m}$ denote the (i,j)-th coordinate of the received word $\boldsymbol{Y}$ regarded as a matrix in ${\mathbb{F}_{q^m}}^{N\times h}$. Let s be a positive integer less than or equal to h. We propose a decoding algorithm based on interpolating an $s+1$-variate linearized polynomial $Q(X, Y_1, \ldots, Y_s)$. The q-degree of Q is characterized in terms of parameter ω which is defined as follows:

$$\omega = \left\lceil \frac{N(h-s+1) + s(k-1) + 1}{s+1} \right\rceil \tag{5.3}$$

This particular choice of ω will guarantee existence of the interpolation polynomial.

List-decoding algorithm of folded Gabidulin codes

1. *Interpolation:* Construct a non-zero multivariate linearized polynomial

$$Q(X, Y_1, Y_2, \ldots, Y_s) = Q_0(X) + Q_1(Y_1) + Q_2(Y_2) + \cdots + Q_s(Y_s)$$

where Q_i's are linearized polynomials over $\mathbb{F}_{q^m}$, Q_0 has q-degree of at most $\omega - 1$ and the q-degree of all other Q_i's is at most $\omega - k$ subject to the constraint that

$$Q(\gamma^{ih+j}, y_{i,j}, y_{i,j+1}, \ldots, y_{i,j+s-1}) = 0 \tag{5.4}$$

for $i = 0, 1, \ldots, N-1$ and $j = 0, 1, \ldots, h - s$.

2. *Message recovery:* Find all the solutions $f(X) \in \mathscr{L}_{q^m}[X]$ to the following equation:

$$Q\big(X, f(X), f(\gamma X), \ldots, f(\gamma^{s-1}X)\big) = 0 \tag{5.5}$$

The decoder outputs coefficients for each solution $f(X)$ as a vector of length k.

The interpolation step is very similar to the interpolation step of the list-decoding algorithm discussed in Section 4.3.1. It can be executed using either the straightforward Gaussian elimination method or an efficient interpolation algorithm in the ring of linearized polynomials as presented in [15], similar to the algorithm presented in Section 4.3.1. The message recovery step is exactly similar to that of the list-decoding algorithm in Section 4.3.1. It also can be executed as discussed in Section 4.3.2. The total complexity of our list-decoding algorithm is then quadratic in terms of the dimension n.

Next, we establish correctness of the proposed list-decoding algorithm and compute the decoding radius of the code.

Lemma 5.3.2. *The particular choice of ω in (5.3) guarantees existence of a non-zero solution for the interpolation polynomial Q that satisfies (5.4).*

Proof. (5.4) is in fact a homogeneous system of $N(h - s + 1)$ linear equations. The number of unknown coefficients is given by

$$\omega + (\omega - k + 1)s = \omega(s + 1) - s(k - 1)$$

If the number of equations is strictly less than the number of variables in a homogeneous system of linear equations, then a non-zero solution is guaranteed to exist . i.e.

$$N(h - s + 1) \leqslant \omega(s + 1) - s(k - 1) - 1 \Leftrightarrow$$
$$\omega \geqslant \frac{N(h - s + 1) + s(k - 1) + 1}{s + 1}$$

This is guaranteed by the choice of ω in (5.3). ■

Let $\boldsymbol{U} \in \mathbb{F}_q^{N\times M}$ denote the codeword corresponding to the message polynomial $f_{\boldsymbol{u}}(X)$. Then $\langle \boldsymbol{U} \rangle \cap \langle \boldsymbol{Y} \rangle$, the intersection of the row spaces of matrices $\boldsymbol{U}$ and $\boldsymbol{Y}$, has dimension $N - t$, where t is the rank of error. We also define the linearized polynomial $E(X)$ as follows:

$$\begin{aligned} E(X) &= Q\big(X, f_{\boldsymbol{u}}(X), f_{\boldsymbol{u}}(\gamma X), \ldots, f_{\boldsymbol{u}}(\gamma^{s-1}X)\big) \\ &= Q_0(X) + \sum_{i=1}^{s} Q_i \otimes f_{\boldsymbol{u}}(\gamma^{i-1}X) \end{aligned}$$

Lemma 5.3.3. *There are at least $(N-t)(h-s+1)$ linearly independent roots in $\mathbb{F}_{q^m}$ for the linearized polynomial $E(X)$.*

Proof. Notice that any element in the row space of $\boldsymbol{U}$ can be represented as

$$\big((f_{\boldsymbol{u}}(\beta), f_{\boldsymbol{u}}(\gamma\beta), \ldots, f_{\boldsymbol{u}}(\gamma^{h-1}\beta)\big)$$

for some $\beta \in \mathbb{F}_{q^m}$. Now consider a basis for $\langle \boldsymbol{U} \rangle \cap \langle \boldsymbol{Y} \rangle$. The basis can be represented as

$$\Big\{ \big((f_{\boldsymbol{u}}(\beta_i), f_{\boldsymbol{u}}(\gamma\beta_i), \ldots, f_{\boldsymbol{u}}(\gamma^{h-1}\beta_i)\big) \ : \ i = 1, 2, \ldots, n-t \Big\}$$

where $\beta_1, \ldots, \beta_{N-t}$ are $N - t$ linearly independent elements of $\mathbb{F}_{q^m}$. In fact, they are taken from the subspace spanned by $1, \gamma^h, \ldots, \gamma^{h(N-1)}$. Then linearity of the interpolation Q and (5.4) together imply that

$$Q\big(\gamma^j\beta_i, f_{\boldsymbol{u}}(\gamma^j\beta_i), f_{\boldsymbol{u}}(\gamma^{j+1})\beta_i, \ldots, f_{\boldsymbol{u}}(\gamma^{j+s-1}\beta_i)\big) = 0$$

for $i = 1, 2, \ldots, N-t$ and $j = 0, 1, \ldots, h-s$. It is indeed equivalent to $\gamma^j\beta_i$ being a root for $E(X)$. We claim that $\gamma^j\beta_i$, for $i = 1, 2, \ldots, N-t$ and $j = 0, 1, \ldots, h-s$ are all linearly independent elements of $\mathbb{F}_{q^m}$. Let $\mathcal{P}_j$ denote the subspace spanned by $\gamma^j, \gamma^{j+h}, \ldots, \gamma^{j+h(N-1)}$. Since $1, \gamma, \ldots, \gamma^{n-1}$ are all linearly independent, the $\mathcal{P}_j$'s are all disjoint. Also, $\gamma^j\beta_i$, for $i = 1, 2, \ldots, N-t$, are $N-t$ linearly independent elements of $\mathcal{P}_j$. This completes the proof of the claim. Therefore, $\gamma^j\beta_i$, for $i = 1, 2, \ldots, N-t$ and $j = 0, 1, \ldots, h-s$, are $(N-t)(h-s+1)$ linearly independent roots for $E(X)$. ■

Corollary 5.3.4. *If $\omega \leqslant (N-t)(h-s+1)$, then $E(X)$ is identically equal to zero.*

Proof. The proof is very similar to the proof of Corollary 4.3.5. The q-degree of $E(X)$ is at most $\omega - 1$ by the same argument. $E(X)$ has at least $(N-t)(h-s+1)$ linearly independent roots by Lemma 5.3.3. Thus, $E(X)$ must be the all zero polynomial. ■

Theorem 5.3.5. *If the number of errors, t, is bounded as*

$$t < \frac{Ns}{s+1}\big(1 - \frac{h}{h-s+1}R\big) \tag{5.6}$$

Then the proposed list-decoding algorithm of folded Gabidulin codes is correct i.e. it outputs a list of size of at most $q^{m(s-1)}$ which includes the transmitted message $\boldsymbol{u}$.

Proof. The interpolation polynomial Q that satisfies (5.4) is guaranteed to exist by Lemma 5.3.2. If

$$\left\lceil \frac{N(h-s+1)+s(k-1)+1}{s+1} \right\rceil \leqslant (N-t)(h-s+1) \tag{5.7}$$

then by Corollary 5.3.4 and using the expression for ω from (5.3), $E(X)$ is the all zero polynomial. (5.7) is equivalent to

$$N(h-s+1)+s(k-1) < (N-t)(h-s+1)(s+1)$$

which can be simplified to (5.6) by using the approximation

$$R \approx \frac{k-1}{n}$$

Therefore, the message polynomial $f_{\boldsymbol{u}}(X)$ is a solution to (5.5). There are at most $q^{m(s-1)}$ solutions to (5.5) by Lemma 4.3.1. Therefore, the list size is at most $q^{m(s-1)}$. ■

Corollary 5.3.6. *The normalized decoding radius of the folded Gabidulin code using the proposed list-decoding algorithm is equal to*

$$\frac{s}{s+1}\big(1 - \frac{h}{h-s+1}R\big)$$

If we let both s and h grow large while s is much smaller that h, we get a decoding radius arbitrarily close to $1 - R$. Notice that $1 - R$ is indeed equal to the normalized minimum rank distance of the code. This means that we are able to achieve the ultimate error-correction radius for rank-metric codes.This result is stated in the following theorem.

Theorem 5.3.7. *For every $\varepsilon > 0$ and $0 < R < 1$, there is a family of folded Gabidulin codes with rate R that can be list-decoded up to a normalized number of errors $1-R-\varepsilon$. The size of output list is at most $\mathcal{Q}^{O(1/\varepsilon)}$, where $\mathcal{Q}$ is the size of the field that the message symbols are chosen from.*

Proof. Given R and ε, we can apply the results of Theorem 5.3.5 and Corollary 5.3.6 with the choice $s = 1/2\varepsilon$ and $h = 1/4\varepsilon^2$. ■

Acknowledgements

This chapter is in part a reprint of the material that is going to be presented at ISIT 2012 in Boston and is also available on arxiv as: H. Mahdavifar and A. Vardy, "List-decoding of Subspace Codes and Rank-Metric Codes up to Singleton Bound," available at http://arxiv.org/pdf/1202.0866.pdf.

Bibliography

[1] N. Cai and R. W. Yeung, "Network coding and error correction," *Proc. of 2002 IEEE Information Theory Workshop,* pp. 119-122, Oct. 2002

[2] N. Cai and R. W. Yeung, "Network error correction, part I: Basic concepts and upper bounds," *Communications in Information and Systems*, vol. 6, no. 1, pp. 19-36, 2006.

[3] N. Cai and R. W. Yeung, "Network error correction, part II: Lower bounds," *Communications in Information and Systems*, vol. 6, no. 1, pp. 37-54, 2006.

[4] P. Delsarte, "Bilinear forms over a finite field, with applications to coding theory," *J. Comb. Theory. Ser. A*, vol. 25, pp. 226-241, 1978.

[5] S. A. Elkind and D. P. Siewiorek, "Reliability and performance of error-correcting memory and register arrays," *IEEE Transactions on Computers*, vol. C-29, pp. 920 - 927, Oct. 1980

[6] E. M. Gabidulin, "Theory of codes with maximum rank distance," *Probl. Inf. Trasnm.*, vol. 21, no. 1, pp. 1-12, 1985.

[7] V. Guruswami, "Linear-algebraic list decoding of folded Reed-Solomon codes," *Proc. of the 26th IEEE Conference on Computational Complexity (CCC)*, pp. 77-85, 2011.

[8] V. Guruswami and A. Rudra, "Explicit codes achieving list decoding capacity: Error-correction up to the Singleton bound." *IEEE Transactions on Information Theory*, vol. 54, pp.İ35150, Jan. 2008.

[9] W. F. Mikhail, R. W. Bartoldus, and R. A. Rutledge, "The reliability of memory with single-error correction," *IEEE Transactions on Computers*, vol. C-31, pp. 56-564, June 1982.

[10] F. Parvaresh and A. Vardy, "Correcting errors beyond the Guruswami-Sudan radius in poly- nomial time," *Proc. of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pp. 285294, 2005.

[11] R. M. Roth, "Maximum-rank array codes and their application to criss-cross error correction," *IEEE Transactions on Information Theory*, vol. 37, no. 2, pp. 328-336, Mar. 1991.

[12] D. Silva, F. R. Kschischang, "On Metrics for Error Correction in Network Coding," *IEEE Trans. Info. Theory*, vol. 55, no. 12, pp. 5479-5490, Dec. 2009.

[13] D. Silva, F. R. Kschischang, R. Koetter, "A rank-metric approach to error control in random network coding," *IEEE Transactions on Information Theory*, vol. 54, no. 9, pp. 3951-3967, Sep. 2008.

[14] S. Vadhan, "Pseudorandomness." Foundations and Trends in Theoretical Computer Science (FnT-TCS). NOW publishers, 2010. To appear. Draft available at http://people.seas.harvard.edu/salil/pseudorandomness.

[15] H. Xie, Z. Yan, B. W, Suter, "General linearized polynomial interpolation and its applications," *2011 Int. Symp. on Network Coding,* Beijing, China, July 25-27 2011.