LIST LEE-METRIC DECODING ALGORITHM FOR GENERALIZED REED-SOLOMON CODES OVER COMMUTATIVE RINGS WITH IDENTITY

Parampalli Udaya
Dept. of Computer Science
and Software Engg.,
University of Melbourne, Australia
udaya@cs.mun.oz.au

Xin-Wen Wu and Margreta Kuijper
Dept. of Electrical and
Electronic Engg.,
University of Melbourne, Australia
x.wu@ee.mun.oz.au, m.kuijper@ee.mun.oz.au

ABSTRACT

Recently M.A. Armand [1] has generalized the Guruswami-Sudan's list decoding procedure to Generalized Reed-Solomon (GRS) codes over commutative rings with identity. This procedure works with Hamming metric. Lee metric is the preferred metric in many practical situations involving codes over rings. In this paper, we present a list decoding procedure for GRS codes over rings which works for Lee metric using Armand's result.

I. INTRODUCTION

Let $C$ be a block code over an alphabet $A$. Denote by $d$ the minimum distance of the code $C$ (which can be either minimum Hamming distance or minimum Lee distance). If in the transmission at most $\tau \leq \lfloor \frac{d-1}{2} \rfloor$ errors occurred, then by the definition of minimum distance there is a unique codeword (i.e., the transmitted codeword) satisfying $d(y, c) \leq \tau$. Classical decoding methods only consider this case. In other words, a classical decoding algorithm can correct at most $\lfloor \frac{d-1}{2} \rfloor$ errors and return a unique codeword, the transmitted codeword. List decoding is a novel decoding method which tries to find all codewords within distance $\tau$ from the received word. Through list decoding it is possible to recover information from errors beyond the classical error-correction bound for low rates. Later [2] Sudan’s algorithm has been improved by Guruswami and Sudan to achieve better error-correction capabilities for any rate.

Recently, the list-decoding algorithm by Guruswami and Sudan has been generalized along two directions: In [5], [4], we generalized the Guruswami-Sudan algorithm to Reed-Solomon codes over finite fields $GF(p)$, which works with Lee metric. In [1] Armand generalized the Guruswami-Sudan algorithm to Generalized Reed-Solomon (GRS) codes over commutative rings with identity, which works with Hamming metric.

In this paper, extending the ideas developed in [4], [5], we present a Lee-metric list-decoding algorithm for Generalized Reed-Solomon codes over rings $\mathbb{Z}_q$, which works with Lee metric.

Let $\mathbb{Z}_q$ be the ring of integers modulo $q$, where $q$ is a positive integer. Block codes over $\mathbb{Z}_q$ in the Lee metric are very useful in many practical applications. For an element in $\mathbb{Z}_q$, denote by $\bar{\alpha}$ the smallest nonnegative integer such that $\alpha = \bar{\alpha} - 1$, where $1$ is the multiplicative unity of $\mathbb{Z}_q$. The Lee value $|\alpha|$ of $\alpha$ is defined as:

$$|\alpha| = \begin{cases} \bar{\alpha}, & \text{when } 0 \leq \bar{\alpha} \leq (q-1)/2, \\ q - \bar{\alpha}, & \text{when } (q-1)/2 < \bar{\alpha} \leq q-1. \end{cases}$$

For a vector $c = (c_1, c_2, \ldots, c_n) \in \mathbb{Z}_q^n$, the Lee weight is defined as:

$$||c|| = \sum_{i=1}^{n} |c_i|.$$ 

The Lee distance between two vectors is defined as the Lee weight of their difference. The minimum Lee distance of a block code over $\mathbb{Z}_q$ is the minimum Lee distance between any pair of distinct codewords.
II. GENERALIZED REED SOLOMON CODES

Following [1], let \( R \) be a finite commutative ring with identity. By \( N(R) \) we denote the set of ring elements which are not zero divisors. Clearly, \( N(R) \) is a multiplicative subset of \( R \) containing the units of \( R \). Since \( R \) is assumed to be finite, \( N(R) \) is in fact the set of units of \( R \). We say a subset \( S \) of \( N(R) \) is subtractive in \( N(R) \) if for all distinct \( a \) and \( b \) in \( S \), \( a - b \) is contained in \( N(R) \).

Now let the alphabet \( A = R \). Let \( d \geq 2 \). Suppose \( \{\alpha_1, \ldots, \alpha_n\} \) is subtractive in \( N(R) \) and \( v_i \in N(R) \) for \( i = 1, \ldots, n \). A generalized Reed-Solomon code \( C \) over \( R \) is defined as

\[
C := \{c \in R^n \mid c \cdot H(C)^T = 0\}
\]

where

\[
H(C) = \begin{pmatrix}
    v_1 & v_2 & \cdots & v_n \\
    v_1 \alpha_1 & v_2 \alpha_2 & \cdots & v_n \alpha_n \\
    \vdots & \vdots & \ddots & \vdots \\
    v_1 \alpha_1^{d-2} & v_2 \alpha_2^{d-2} & \cdots & v_n \alpha_n^{d-2}
\end{pmatrix}
\]

is a parity-check matrix of the code. It is easy to see that this is a \([n, n-d+1, d]\) code over \( R \). We sometimes denote this code by \( GRS_{n-d+1}(\alpha, v) \) where \( \alpha = (\alpha_1, \ldots, \alpha_n) \) and \( v = (v_1, \ldots, v_n) \). Its dual code is also a generalized Reed-Solomon code \( GRS_{d-1}(\alpha, v') \) for appropriately chosen \( v' \in N(R)^n \). From the equality

\[
H(GRS_{d-1}(\alpha, v')) \cdot H(GRS_{n-d+1}(\alpha, v))^T = 0,
\]

we see that \( v' \) may be found by solving the \( n-1 \) equations

\[
\sum_{i=1}^{n} v_i v_i' \alpha_i^j = 0, \quad j = 0, 1, \ldots, n - 2.
\]

In particular, since \( H(GRS_{d-1}(\alpha, v')) \) is a generator matrix of \( GRS_{n-d+1}(\alpha, v) \), it is easily seen that the codewords of this code have the form

\[
(v_1' f(\alpha_1), v_2' f(\alpha_2), \ldots, v_n' f(\alpha_n))
\]

where \( f \) is a polynomial in \( R[X] \) of degree less than \( k := n - d + 1 \). Let \( R[X]_{<k} \) stand for the set of polynomials in the indeterminate \( X \) with coefficients in \( R \) and degrees less than \( k \). \( C = GRS_{n-d+1}(\alpha, v) \) can be also defined as

\[
C := \{(v_1' f(\alpha_1), \ldots, v_n' f(\alpha_n)) \mid f(X) \in R[X]_{<k}\}.
\]

Example II.1: Let \( R = \mathbb{Z}_{p^2}, n = 6, d = 5 \). Choose for some \( b \geq 0 \), \( \alpha_1 = 5^i - 1 \) and \( v_i = 5^i(i-1) \) for \( 1 \leq i \leq n \). Then,

\[
H(C) = \begin{pmatrix}
    1 & 5^b & 5^{2b} & \cdots & 5^{5b} \\
    1 & 5^{b+1} & 5^{2(b+1)} & \cdots & 5^{5(b+1)} \\
    1 & 5^{b+2} & 5^{2(b+2)} & \cdots & 5^{5(b+2)} \\
    1 & 5^{b+3} & 5^{2(b+3)} & \cdots & 5^{5(b+3)}
\end{pmatrix}
\]

is a parity check matrix for a \((6, 2, 5)\) Reed-Solomon code over \( R \). Taking \( b = 1 \), we can obtain the solution to (1) for the parameters \( v_i' s \) and \( \alpha_i's \) of this example as

\[
v' = (1, 22, 15, 43, 8, 43).
\]

Hence the generator matrix for the Reed-Solomon code is given by

\[
G(C) = \begin{pmatrix}
    1 & 22 & 15 & 43 & 8 & 43 \\
    1 & 12 & 32 & 34 & 2 & 17
\end{pmatrix}
\]

Correspondingly, the codewords have the form

\[
(v_1' f(\alpha_1), \ldots, v_n' f(\alpha_n)) = (f(1), 22 f(5), 15 f(25), 43 f(27), 8 f(37), 43 f(38)),
\]

where \( f \) is any polynomial of degree at most one.

In this paper, we consider generalized Reed-Solomon codes over finite rings \( \mathbb{Z}_q \), for which the Lee-metric is well-defined. In the sequel, we assume \( R = \mathbb{Z}_q \). And for the convenience to state the decoding algorithm we only consider those generalized Reed-Solomon codes \( C = GRS_k(\alpha, v) \) such that \( (v_1', \ldots, v_n') = (1, \ldots, 1) \), i.e.,

\[
C = \{(f(\alpha_1), \ldots, f(\alpha_n)) \mid f(X) \in R[X]_{<k}\}.
\]

III. THE GURUSWAMI-SUDAN ALGORITHM AND ARMAND’S GENERALIZATION

Let \( w_1 \) and \( w_2 \) be two nonnegative integers. For a bivariate polynomial \( Q = \sum_{i,j} q_{i,j} X^i Y^j \in A[X, Y] \), the \((w_1, w_2)\)-weighted degree of \( Q \), which is denoted by \( deg_{w_1, w_2}(Q) \), is the maximal value among \( iw_1 + jw_2 \). Denote by \( <|w_1, w_2| \) as the term order in \( A[X, Y] \) with the degree of \( Y \) as tie-breaker. The leading term \( lt(Q) \) of \( Q \) is the highest order term among all the terms in \( Q \). The leading coefficient \( lc(Q) \) of \( Q \) is the coefficient of leading term of \( Q \).
Conform the literature we define \((0,0)\) to be a zero of \(Q(X,Y) = \sum q_{ij}X^iY^j\) of multiplicity \(\gamma\) if for any \(i,j \geq 0\) such that \(i+j \leq \gamma - 1, q_{ij} = 0\). More generally, a point \((\alpha, \beta)\) is defined to be a zero of \(Q(x,y)\) of multiplicity \(\gamma\) if for any \(i,j \geq 0\) such that \(i+j \leq \gamma - 1, q_{ij} = 0\). Here \(Q(\alpha,\beta) = \sum q_{ij}X^iY^j\) is defined as \(Q(\alpha,\beta)(X,Y) = Q(X+\alpha, Y+\beta)\). By simple calculations, the coefficients of \(Q(X,Y)\) and \(Q(\alpha,\beta)(X,Y)\) satisfy

\[
q_{ij} = \sum_{i' \geq i, j' \geq j} \binom{i'}{i} \binom{j'}{j} q_{i',j'} \alpha^{i'-i} \beta^{j'-j}.
\]

Now let us recall the Guruswami-Sudan algorithm for a \([n,k]\) Reed-Solomon code over a finite field \(A = GF(q)\) in Hamming metric. Given positive integers \(t\) and \(\tau := n - t\) satisfying \(t^2 > nk\), and the received word \(y = (y_1, \cdots, y_n)\), the Guruswami-Sudan algorithm is a polynomial time algorithm which finds all the codewords \(c \in C\) satisfying \(d_H(c,y) \leq \tau\), where \(d_H(c,y)\) denotes the Hamming distance between \(c\) and \(y\).

The algorithm consists of two steps. The first step concerns the computation of a bivariate polynomial \(Q \in A[X,Y] - \{0\}\) satisfying \(\deg_{(1,k)}(Q) \leq \ell\) where \(\gamma\) is an appropriate parameter and \(\ell = \gamma t - 1\), and \(Q\) satisfies that for \(i = 1, \cdots, n\), the coefficients of \(Q(X+\alpha_i, Y+y_i)\) of total degree less than \(\gamma\) are zero. In other words, \(Q\) passes through the points \(\gamma\) time for each \(i\), or \((\alpha_i, y_i)\) is a zero of \(Q(X,Y)\) of multiplicity \(\gamma\). The second step involves finding all polynomials \(f_j \in A[X]\) of degree less than \(k\) such that each \(f_j\) is a \(Y\)-root of \(Q\), i.e., each \(Y - f_j\) is a factor of \(Q\). For each such polynomial \(f_j\), if \((f_j(\alpha_1), \cdots, f_j(\alpha_n))\) and \(y\) differ in no more than \(\tau\) coordinates, then \((f_j(\alpha_1), \cdots, f_j(\alpha_n))\) is included in the output list.

Recently, Armand in [1] proved that the Guruswami-Sudan algorithm remains valid over any commutative ring with identity if the code location \(\{\alpha_1, \cdots, \alpha_n\}\) is subtractive in \(N(R)\). In the sequel we assume that this condition hold.

IV. LIST LEE-METRIC DECODING

It is clear that the Guruswami-Sudan algorithm actually solves the following curve-fitting problem: given a set of \(t\) points in the plane, say \(\{(\alpha_1, y_1), \cdots, (\alpha_n, y_n)\}\) where \(\alpha_1, y_1 \in GF(q)\) and \(\alpha_1, \cdots, \alpha_n\) are distinct elements, for parameters \(k\) and \(t\), find all curves \(Y = f(X)\), where \(f(X) \in GF(q)[X]_{<k}\), that pass through at least \(t\) points in \(\{(\alpha_1, y_1), \cdots, (\alpha_n, y_n)\}\), more precisely, \(f(\alpha_i) = y_i\) for at least \(t\) values of \(i \in \{1, \cdots, n\}\).

How is the curve-fitting problem related to Hamming metric list-decoding? One can see the relationship between the two seemingly different problems as follows: letting \(y = (y_1, \cdots, y_n)\) be the received word, then a curve \(Y = f(X)\) is a solution of the above curve-fitting problem if and only if \(c = (f(\alpha_1), \cdots, f(\alpha_n))\) is a codeword satisfying

\[
d_H(c,y) \leq \tau
\]

where \(\tau = n - t\). In fact, by the definition of Hamming distance, the number of points in \(\{(\alpha_1, y_1), \cdots, (\alpha_n, y_n)\}\) that “fit” (or in other words, lie on) the curve \(Y = f(X)\) is exactly \(n - d_H(y,c)\) for \(c = (f(\alpha_1), \cdots, f(\alpha_n))\).

In the Lee metric case, there is no direct relationship between \(d_L(y,c)\) and the number of points in \(\{(\alpha_1, y_1), \cdots, (\alpha_n, y_n)\}\) that fit the curve \(Y = f(X)\). How can we relate the curve-fitting problem to Lee metric list-decoding? To answer this question we now present the first main idea of this paper. The idea is to establish a relationship between \(d_L(y,c)\) and the number of sets which consist of certain points in the plane defined as follows.

Let \(u\) be an integer and \(0 \leq u \leq \frac{t+1}{2} - 1\). We define \(n\) sets of points as

\[
S_i := \{(\alpha_i, y_i - u), (\alpha_i, y_i - u + 1), \cdots, (\alpha_i, y_i), (\alpha_i, y_i + 1), \cdots, (\alpha_i, y_i + u)\},
\]

for \(i = 1, 2, \cdots, n\). Every \(S_i\) consists of \(2u + 1\) distinct points in the plane. We say that a plane curve \(F(X,Y) = 0\) “passes through” \(S_i\) if at least one point in \(S_i\) lies on the curve. Then we have following result.

Lemma 1 Let \(y = (y_1, \cdots, y_n)\) and \(c = (f(\alpha_1), \cdots, f(\alpha_n))\) where \(f(X) \in \mathbb{Z}_n[X]_{<\gamma}\). Let \(S_i\) be defined as in (2). If

\[
d_L(y,c) \leq \tau
\]

then the curve \(Y = f(X)\) passes through at least \(t := n - \left\lceil \frac{\tau}{u+1} \right\rceil\) distinct sets in \(\{S_1, \cdots, S_n\}\).

Proof: It is clear that the curve \(Y = f(X)\) passes through \(S_i\), i.e., a point in \(S_i\) lies on \(Y = f(X)\) if and only if the Lee value \(|f(\alpha_i) - y_i| \leq u\).
Suppose that the assertion of the lemma is not true. Then there are more than \((n-t)\) values of \(i \in \{1, \cdots, n\}\) such that \(|f(\alpha_i) - y_i| \geq u + 1\). As a result,

\[
d_L(y, c) = \sum_{i=1}^{n} |f(\alpha_i) - y_i| > (n-t) \cdot (u+1) \geq \tau.
\]

This contradicts our assumption and thus proves the lemma. □

Based on the relationship between the Lee-metric list-decoding and the curve-fitting problem, we give a Lee-metric List decoding algorithm for generalized Reed-Solomon codes \(C\) over \(\mathbb{Z}_n\) as follows.

Denote \(S = S_1 \cup S_2 \cup \cdots \cup S_n\), where \(S_i\) is a set defined as (2). We re-order the points in \(S\) and denote \(S = \{(x_1, z_1), (x_2, z_2), \cdots, (x_N, z_N)\}\) where every \((x_i, z_i) = (\alpha_i, y_j + v)\) for some \(j\) and \(|v| \leq u\). It is clear that \(S\) consists of \(N := (2u+1)n\) distinct points.

**Algorithm (List Lee-metric Decoding)**

**Inputs:** \(n, k, u, \tau, t = n - \left[\frac{\tau}{u+1}\right]\), and the point set \(S = \{(x_1, z_1), (x_2, z_2), \cdots, (x_N, z_N)\}\).

**Step 0:** Choose the parameters \(\gamma, l\) such that

\[
\gamma t > l \text{ and } (2u+1)n \left(\frac{\gamma + 1}{2}\right) < \frac{l(l+2)}{2k}.
\]

**Step 1:** Find a nonzero polynomial

\[
Q(X, Y) = \sum q_{ij} X^i Y^j
\]

over \(R\) such that \(deg_{(1,k)}(Q) \leq l\), and for any \(i \in \{1, \cdots, N\}\), \((x_i, z_i)\) is a zero of \(Q(X, Y)\) of multiplicity \(\gamma\). In other words, for any \(i \in \{1, \cdots, N\}\), any \(j_1, j_2 \geq 0\), s.t., \(j_1 + j_2 \leq \gamma - 1\),

\[
\sum_{j_1 \geq 1} \sum_{j_2 \geq 2} \left(\begin{array}{c} j_1' \cr j_1 \end{array}\right) \left(\begin{array}{c} j_2' \cr j_2 \end{array}\right) q_{j_1'j_2', j_1j_2} z_i^{j_1'-1} z_i^{j_2'-2} = 0.
\]

**Step 2:** Find all polynomials \(f(X) \in R[X]_{<k}\), such that \(Y - f(X)\) is a factor of the polynomial \(Q(X, Y)\). For each such \(f(X)\) and \(c = (f(\alpha_1), \cdots, f(\alpha_n))\) check if

\[
d_L(y, c) \leq \tau.
\]

If so, include \(c = (f(\alpha_1), \cdots, f(\alpha_n))\) in the output list.

Note that when \(u = 0\) the set \(S\) is exactly \(\{(\alpha_1, y_1), \cdots, (\alpha_n, y_n)\}\) and the algorithm is the same as the Armand’s generalization of the original Guruswami-Sudan algorithm except for the last step that checks the Lee distance rather than the Hamming distance. The parameter \(u\) has two functions. Firstly, as we saw in Lemma 1, the parameter \(u\) helps us to reformulate the Lee distances between the received word and the codewords. Secondly, as we will see in the following, a careful choice of the value of \(u\) helps us to maximize the Lee error-correction capability of the algorithm.

To prove the validity of the algorithm we need to prove:

1. there exist \(\gamma\) and \(l\) such that the conditions in Step 0 hold;
2. a nonzero polynomial \(Q(X, Y)\) sought by Step 1 does exist;
3. for every codeword \(c = (f(\alpha_1), \cdots, f(\alpha_n))\) with \(d_L(y, c) \leq \tau\) we have that \(Y - f(X)\) is a factor of \(Q(X, Y)\). We can prove the following lemmas.

**Lemma 2** If \(t^2 > (2n+1)nk\), then there exist \(\gamma\) and \(l\) such that both \(\gamma t > l\) and \((2u+1)n(\gamma + 1) < \frac{l(l+2)}{2k}\) hold.

**Lemma 3** If \((2u+1)n(\gamma + 1) < \frac{l(l+2)}{2k}\), then a nonzero polynomial \(Q(X, Y)\) as sought in Step 1 does exist.

**Lemma 4** Let \(t = n - \left[\frac{\tau}{u+1}\right]\). Suppose \(\gamma t > l\) and for \(c = (f(\alpha_1), \cdots, f(\alpha_n))\),

\[
d_L(y, c) \leq \tau.
\]

Let \(Q(X, Y)\) be any nonzero polynomial returned from Step 1 of the algorithm. Then \(Y - f(X)\) is a factor of \(Q(X, Y)\), i.e., \(Q(X, f(X))\) is identically zero.

Now we can prove the following theorem.

**Theorem** Consider the GRS code \(C\) defined in (1). Suppose a nonnegative integer \(\tau\) satisfies

\[
\tau \leq u + 1(n - \sqrt{2u+1}) - 1).
\]

Then for any received word \(y = (y_1, \cdots, y_n)\), the algorithm finds all codewords \(c\) satisfying \(d_L(y, c) \leq \tau\), in other words, the algorithm decodes up to \(\tau\) Lee errors.

**Proof:** From \(\tau \leq u + 1(n - \sqrt{2u+1}) - 1)\), we have

\[
t := n - \left[\frac{\tau}{u+1}\right] \geq \sqrt{2u+1}\] + 1.
Therefore, \( t^2 > (2u + 1)nk \). Then by Lemmas 2-4 the algorithm finds all the codewords \( c \) satisfying \( d_L(y, c) \leq \tau \).

The above theorem shows that for any integer \( u \) such that \( 0 \leq u \leq \frac{q-1}{2} - 1 \), the algorithm works for up to

\[
\tau(u) = (u + 1)(n - \sqrt{(2u + 1)nk}) - 1
\]

Lee errors.

**Remark IV.1:** Note that the algorithm given in [1] which holds only for Hamming metric can also be used for Lee errors. Hence the algorithm of [1] can output lists with maximum of \( (n - \sqrt{nk}) - 1 \) Lee errors. The new algorithm can output lists with maximum of \( (u + 1)(n - \sqrt{(2u + 1)nk}) - 1 \) errors which is significantly better than the algorithm in [1] for low rate codes.

Viewing \( \tau(u) \) as a function of \( u \), we next study the problem of maximizing \( \tau(u) \). Denoting

\[
A(u) = (u + 1)(n - \sqrt{(2u + 1)nk}) - 1,
\]

and

\[
B(u) = (u + 1)(n - \sqrt{(2u + 1)nk}),
\]

it is obvious that

\[
A(u) \leq \tau(u) < B(u).
\]

Viewing \( u \) as a real variable for which \( 0 \leq u \leq \frac{q-1}{2} - 1 \), we now consider the functions \( A(u) \) and \( B(u) \). By simple calculations, \( A(u) \) achieves its maximal value for

\[
u = u_0(n, k) := \frac{(n-1)^2 - 6nk + (n-1)\sqrt{(n-1)^2 - 3nk}}{9nk},
\]

and \( B(u) \) achieves its maximal value for

\[
u = u_1(n, k) := \frac{n - 6k + \sqrt{n^2 - 3nk}}{9k}.
\]

Therefore, the integer \( u \) that maximizes \( \tau(u) \) should be in the range \( u_0 - \epsilon \leq u \leq u_1 + \epsilon \) (or \( u_1 - \epsilon \leq u \leq u_0 + \epsilon \)) if \( u_1 \leq u_0 \) with small \( \epsilon \).

It is interesting to note that for fixed rate \( k/n := r \) the functions \( u_0(n, k) \) and \( u_1(n, k) \) (and thus \( u \)) converge to a constant for \( n \) tending to infinity. In fact, both functions converge to the constant \( (1 - 6r + \sqrt{1 - 3r})/9r \). This means that for large \( n \), the integer \( u \) that maximizes \( \tau(u) \) is relatively small. For example, for \( n = 100 \) and \( k = 6 \) its value equals \( u = 3 \). Thus, the complexity of the algorithm is same as that of the Guruswami-Sudan algorithm, which has complexity \( O(n^3) \). Also note that all the fast implementations the Guruswami-Sudan algorithm are applicable to our algorithm.

We next consider the size of the output list. Let \( d_L \) be the true minimum Lee distance of the code. If \( \tau \leq \left\lfloor \frac{d_L - 1}{2} \right\rfloor \), then the output list consists of a single codeword (i.e. the transmitted codeword). If \( \tau > \left\lfloor \frac{d_L - 1}{2} \right\rfloor \), then the output list may have more than one codeword which contains the transmitted codeword. In this case, we can pick out the transmitted codeword using some additional information of the channel.

V. CONCLUSIONS

In this paper, we present a list decoding procedure for GRS codes over rings which works for Lee metric using Armand’s result.

REFERENCES


