Error Correcting Codes: Combinatorics, Algorithms and Applications **(Fall 2007)**

Lecture 28: Generalized Minimum Distance Decoding

November 5, 2007

Lecturer: Atri Rudra *Scribe: Sandipan Kundu & Atri Rudra*

1 Decoding From Errors and Erasures

So far, we have seen the following result concerning decoding of RS codes:

Theorem 1.1. *An $[n,k]_q$ RS code can be corrected from e errors (or s erasures) as long as $e < \frac{n-k+1}{2}$ (or $s < n-k+1$) in $O(n^3)$ time.*

Next, we show that we can get the best of the errors and erasures worlds simultaneously:

Theorem 1.2. *An $[n,k]_q$ RS code can be corrected from e errors and s erasures in $O(n^3)$ time as long as*

$$2e + s < n - k + 1. \tag{1}$$

Proof. Given a received word $\mathbf{y} \in (\mathbb{F}_q^n \cup \{?\})^n$ with s erasures and e errors, let $\mathbf{y}'$ be the sub-vector with no erasures. This implies $\mathbf{y}' \in \mathbb{F}_q^{n-s}$, which is a valid received word for an $[n-s,k]_q$ RS code. Now run the Berlekamp-Welch algorithm on $\mathbf{y}'$. It can correct $\mathbf{y}'$ as long as

$$e < \frac{(n-s)-k+1}{2}.$$

This condition is implied by (1). Thus, we have proved one can correct e errors under (1). Now we have to prove that one can correct the s erasures under (1). Let $\mathbf{z}'$ be the output after correcting e errors. Now we extend $\mathbf{z}'$ to $\mathbf{z} \in (\mathbb{F} \cup \{?\})^n$ in the natural way. Finally, run the erasure decoding algorithm on $\mathbf{z}$. This works as long as $s < (n-k+1)$, which in turn is true by (1).

The time complexity of the algorithm above is $O(n^3)$ as both the Berlekamp-Welch algorithm and the erasure decoding algorithm can be implemented in cubic time. □

Next, we will use the errors and erasure decoding algorithm above to design decoding algorithms for certain concatenated codes that can be decoded up to half their design distance.

2 Generalized Minimum Distance Decoding

In the last lecture, we studied the natural decoding algorithm for concatenated codes. In particular, we performed MLD on the inner code and then fed the resulting vector to a unique decoding algorithm for the outer code. As was mentioned last time, a drawback of this algorithm is that it does not take into account the information that MLD can offer. E.g., the situations where a given inner code received word has a Hamming distance of one vs. (almost) half the inner code distance are treated the same by the algorithm. It seems natural to try and make use of this information. Forney in 1966 devised such an algorithm, which is called Generalized Minimum Distance (or GMD) decoding [1]. We study this algorithm next and our treatment follows that of [2].

In the sequel, let C_{out} be an $[N, K, D]_{q^k}$ code that can be decoded from e errors and s erasures in polynomial time as long as $2e + s < D$. Let C_{in} be an $[n, k, d]_q$ code with $k = O(\log N)$

We will in fact look at three versions of the GMD decoding algorithm. The first two will be randomized algorithms while the last will be a deterministic algorithm. We begin with the randomized version as it presents most of the ideas in the final algorithm.

2.1 GMD algorithm (Version 1)

Before we state the algorithm we look at a special case of the problem to build up some intuition. Consider the received word $\mathbf{y} = (y_1, \ldots, y_N) \in [q^n]^N$ with the following special property: for every $1 \le i \le N$, either $y_i = C_{in}(y'_i)$ or $\Delta(y_i, C_{in}(y'_i)) \ge d/2$, where for every $1 \le i \le N$, define $y'_i = MLD_{C_{in}}(y_i)$. Now we claim that if $\Delta(\mathbf{y}, C_{out} \circ C_{in}) < dD/2$, then there are $< D$ positions i such that $\Delta(y_i, C_{in}(y'_i)) \ge d/2$ (call such a position *bad*). This is because for every bad position i, by the definition of y'_i, $\Delta(y_i, C_{in}) \ge d/2$. Now if there are at least D bad positions, this will imply that $\Delta(\mathbf{y}, C_{out} \circ C_{in}) \ge dD/2$, which is a contradiction. Now note that we can decode $\mathbf{y}$ by just declaring an erasure at every bad position and running the erasure decoding algorithm for C_{out} on the resulting vector. The GMD algorithm below generalizes these observations to the general case:

Input: $\mathbf{y} = (y_1, \ldots, y_N) \in [q^n]^N$.

Step 1: For every $1 \le i \le N$:

(a) Compute $y'_i = MLD_{C_{in}}(y_i)$.

(b) Compute $w_i = \min\left(\Delta(C_{in}(y'_i), y_i), \frac{d}{2}\right)$.

(c) With probability $\frac{2w_i}{d}$, set $y''_i \leftarrow ?$, otherwise set $y''_i = y'_i$.

Step 2 : Run errors and erasure algorithm for C_{out} on $\mathbf{y}'' = (y''_1, \ldots, y''_N)$.

Note that if for every $1 \le i \le N$, either $C_{in}(y'_i) = y_i$ or $\Delta(C_{in}(y'_i), y_i) \ge d/2$, then the GMD algorithm above does exactly the same as in the discussion above.

By our choice of C_{out} and C_{in}, it is easy to see that the algorithm above runs in polynomial time. More importantly, we will show that the final (deterministic) version of the algorithm above can do unique decoding of $C_{out} \circ C_{in}$ up to half its design distance.

Theorem 2.1. *Let* $\mathbf{y}$ *be a received word such that there exists a codeword* $\mathbf{c} = (c_1, \ldots, c_N) \in C_{out} \circ C_{in} \subseteq [q^n]^N$ *such that* $\Delta(\mathbf{c}, \mathbf{y}) < \frac{Dd}{2}$. *Then the deterministic GMD algorithm outputs* $\mathbf{c}$.

As a first step, we will show that in expectation the randomized GMD algorithm above works.

Lemma 2.2. *Let the assumption in Theorem 2.1 hold. Further, if* $\mathbf{y}''$ *has* e' *errors and* s' *erasures (when compared with* $\mathbf{c}$*) after* **Step 1** *, then*

$$\mathbb{E}[2e' + s'] < D.$$

Note that if $2e' + s' < D$, then the algorithm in **Step 2** will output $\mathbf{c}$. The lemma above says that in expectation, this is indeed the case.

Proof of lemma 2.2. For every $1 \le i \le N$, define $e_i = \Delta(y_i, c_i)$. Note that this implies that

$$\sum_{i=1}^{N} e_i < \frac{Dd}{2}. \tag{2}$$

Next for every $1 \le i \le N$, we define two indicator variables:

$$X_i^? = 1 \text{ iff } y_i'' = ?,$$

and

$$X_i^e = 1 \text{ iff } \ C_{in}(y_i'') \neq c_i \text{ and } y_i'' \neq ?.$$

We claim that we are done if we can show that for every $1 \le i \le N$:

$$\mathbb{E}[2X_i^e + X_i^?] \le \frac{2e_i}{d}. \tag{3}$$

Indeed, by definition $e' = \sum_i X_i^e$ and $s' = \sum_i X_i^?$. Further, by the linearity of expectation, we get

$$\mathbb{E}[2e' + s'] \le \frac{2}{d} \sum_i e_i < D,$$

where the second inequality follows from (2).

To complete the proof, we will show (3). Towards this end, fix an arbitrary $1 \le i \le N$. We prove (3) by a case analysis.

Case 1: ($c_i = C_{in}(y_i')$) First, we note that if $y_i'' = ?$ then $X_i^e = 0$. This along with the fact that $Pr[y_i'' = ?] = 2w_i/d$ implies

$$\mathbb{E}[X_i^?] = Pr[X_i^? = 1] = \frac{2w_i}{d},$$

and

$$\mathbb{E}[X_i^e] = Pr[X_i^e = 1] = 0.$$

Further, by definition we have

$$w_i = \min\left(\Delta(C_{in}(y_i'), y_i), \frac{d}{2}\right) \le \Delta(C_{in}(y_i'), y_i) = \Delta(c_i, y_i) = e_i.$$

The three relations above prove (3) for this case.
Case 2: ($c_i \neq C_{in}(y_i')$) As in the previous case, we still have

$$\mathbb{E}[X_i^?] = \frac{2w_i}{d}.$$

Now in this case, if an erasure is not declared at position i, then $X_i^e = 1$. Thus, we have

$$\mathbb{E}[X_i^e] = Pr[X_i^e = 1] = 1 - \frac{2w_i}{d}.$$

Next, we claim that as $c_i \neq C_{in}(y_i')$,

$$e_i + w_i \ge d, \tag{4}$$

which implies

$$\mathbb{E}[2X_i^e + X_i^?] = 2 - \frac{2w_i}{d} \le \frac{2e_i}{d},$$

as desired.

To complete the proof, we show (4) via yet another case analysis.
Case 2.1: ($w_i = \Delta(C_{in}(y_i'), y_i) < d/2$) By definition of e_i, we have

$$e_i + w_i = \Delta(y_i, c_i) + \Delta(C_{in}(y_i'), y_i) \ge \Delta(c_i, C_{in}(y_i')) \ge d,$$

where the first inequality follows from the triangle inequality and the second inequality follows from the fact that C_{in} has distance d.
Case 2.2: ($w_i = \frac{d}{2} \le \Delta(C_{in}(y_i'), y_i)$) As y_i' is obtained from MLD, we have

$$\Delta(C_{in}(y_i'), y_i) \le \Delta(c_i, y_i).$$

This along with the assumption on $\Delta(C_{in}(y_i'), y_i)$, we get

$$e_i = \Delta(c_i, y_i) \ge \Delta(C_{in}(y_i'), y_i) \ge \frac{d}{2}.$$

This in turn implies that

$$e_i + w_i \ge d,$$

as desired. □

2.2 Version 2 of the GMD algorithm

In the first version of the GMD algorithm in **Step 1**(c), we used "fresh" randomness for each i. Next we look at another randomized version of the GMD algorithm that uses the *same* randomness for every i. In particular, consider the following algorithm:

Input: $\mathbf{y} = (y_1, \ldots, y_N) \in [q^n]^N$.

Step 1: Pick $\theta \in [0, 1]$ at random. Then for every $1 \leq i \leq N$:

(a) Compute $y_i' = MLD_{c_{in}}(y_i)$.

(b) Compute $w_i = \min\left(\Delta(C_{in}(y_i'), y_i), \frac{d}{2}\right)$.

(c) If $\theta < \frac{2w_i}{d}$, set $y_i'' \leftarrow ?$, otherwise set $y_i'' = y_i'$.

Step 2 : Run errors and erasure algorithm for C_{out} on $\mathbf{y}'' = (y_1'', \ldots, y_N'')$.

We note that in the proof of Lemma 2.2, we only use the randomness to show that

$$Pr[y_i'' = ?] = \frac{2w_i}{d}.$$

In the current version of the GMD algorithm, we note that

$$Pr[y_i'' = ?] = Pr\left[\theta \in \left[0, \frac{2w_i}{d}\right]\right] = \frac{2w_i}{d},$$

as before (the last equality follows from the our choice of θ). One can verify that the proof of Lemma 2.2 can be used to show that even for version 2 of GMD, $\mathbb{E}[2e' + s'] < D$.

Next lecture, we will see how to derandomize the above version of the GMD algorithm by choosing θ from a polynomially sized set (as opposed to the current infinite set $[0, 1]$.)

References

[1] G. David Forney. Generalized Minimum Distance decoding. *IEEE Transactions on Information Theory*, 12:125–131, 1966.

[2] Venkatesan Guruswami. Error-Correcting Codes: Constructions and Algorithms, Lecture no. 11. Available at `http://www.cs.washington.edu/education/courses/533/06au/`, November 2006.