6.895 Essential Coding Theory	September 29, 2004
Lecture 7	
Lecturer: Madhu Sudan	*Scribe: Kunal Agrawal*

1 Overview

1. Generalize the Welch Berlekamp decoding algorithm for other linear codes.
2. Impossibility Bound for error correction.
3. Introduction to list decoding.

2 Abstracting the Welch Berlekamp algorithm

In this section, we generalize the conditions for Welch Berlekamp algorithm so we don't necessarily have to use polynomials. The following algorithm can decode linear codes provided they satisfy certain constraints.

Given code C by its generator matrix $G_c \in F^{k*n}$, the algorithm can correct e errors provided we can find

- Error correcting code A given by $G_a \in F^{a*n}$
- Error correcting code B given by $G_b \in F^{b*n}$

such that

- $A * C \subseteq B$ [1]
- $dim(A) \geq e$
- $dist(B) \geq e$
- $dist(A) + dist(C) \geq n$

Goal

Given the received vector $y = (y_1, y_2, \ldots, y_n)$ where $y_i \in F$, we want to find $c = (c_1, c_2, \ldots, c_n) \in C$ such that number of places where $y_i \neq c_i$ is at most e.

Algorithm

1. Find error locater vector $E \in A$ and $N \in B$ s.t.
$$y * E = N$$
2. Now use E_i to separate the coordinates of y into correct and incorrect. Let
$$c_i = \begin{cases} y_i & E_i \neq 0 \\ ? & E_i = 0 \end{cases}$$
3. Do erasure decoding of c.

[1] $*$ operator defines coordinate wise multiplication of two vectors. If $a = (a_1, \ldots a_n)$, $b = (b_i, \ldots, b_n)$ are two vectors, then vector $c = a * b$ if $\forall i, a_i = b_i \cdot c_i$. In this instance, A,B and C are vector spaces. The above notation means that any vector in A coordinate wise multiplied by any vector in C gives a vector in B.

Proof of correctness

Lemma 1 *A pair of vectors (E, N) as described in step 1 of the algorithm exists provided*

1. *y is "close to" some codeword c. That is $\triangle(y, c) \leq e$.*
2. *$dim(A) \geq e$.*
3. *$A * C \subseteq B$.*

Proof Say that $S = \{i | y_i \neq c_i\}$. We know that $|S| \leq e$. Since $dim(A) \geq e$

$$\exists E \in A - \{0\} \text{ s.t.} E_i = 0, \forall i \in S$$

. So we pick this E. We know that $E * c = N$ for some $N \in B$. Thus $N_i = E_i \cdot c_i$. Thus

$$N_i = E_i.c_i = 0 = E_i.y_i, \forall i \in S$$

. But $y_i = c_i$ in all other places. Thus $N_i = E_i \cdot y_i$.
■

Lemma 2 *For any pair (E, N) returned in step 1 of the algorithm, if y is close to c, it is the case that $c * E = N$.*

Proof We know that $y * E = N$. Suppose $c * E = N' \neq N$. We also know that $N', N \in B$. But

$$\forall i \text{ where} y_i = c_i \text{ we have} N_i = N_i'$$

. But $\triangle(y_i, c_i) \leq e$. Thus $\triangle(N, N') \leq e$. But $dist(B) > e$. Thus we have a contradiction.
■

Lemma 3 $\forall i$ *where* $E_i \neq 0$ *we have* $c_i = y_i$

Proof

$$\begin{aligned} N_i &= N_i' \\ y_i \cdot E_i &= c_i \cdot E_i \\ y_i &= c_i \quad \text{where } E_i \neq 0 \end{aligned}$$

■

Lemma 4 *All the erasures in step 2 can be decoded.*

Proof Number of erasures $\leq n - dist(A)$, but $dist(A) + dist(C) \geq n$. Thus number of erasures $\leq dist(A)$. Thus they can all be corrected. ■

Hence the above algorithm works for arbitrary linear codes if we can find the codes A and B with the requisite properties.

3 Chinese Remainder codes

Chinese remainder codes are number theoretic analog of Reed-Solomon codes. They are based on the Chinese remainder theorem.

Theorem 5 Chinese Remainder Theorem: *Given n relatively prime numbers $p_1, p_2, \ldots, p_n$ and numbers $a_1, a_2, \ldots, a_n$, $\exists! a, 0 \leq a \leq p_1 \cdot p_2 \ldots p_n$ s.t. $\forall i a (mod p_i) = a_i$.*

Thus we can use Chinese remainder theorem for coding. The message space is $\{a | 0 \leq a \leq \prod_{i=1} k p_i\}$ where $k < n$. The encoding of a is $< a(mod p_1), a(mod p_2), \ldots, a(mod p_n) >$. Any k of these numbers is enough to reconstruct a.

4 Impossibility Result

The next question we can ask is, can we correct more than $(d-1)/2$ errors using RS decoding? We will prove an impossibility result for it.

Theorem 6 *No code can correct more than $(d-1)/2$ errors.*

Proof Consider two codewords c_1 and c_2 which differ in exactly d places, say the last d. As shown in 1 we can construct a vector that differs in exactly $d/2$ places from both of them. Thus the algorithm cannot know which of the codewords to output as the correct code word.

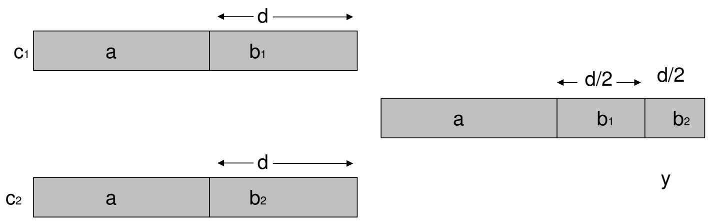

Figure 1: y is at distance d from c_1 and c_2

■

5 List Decoding

We saw above that it is not possible to correct more than $(d-1)/2$ errors. But in the above example, the algorithm could have output both the possible code words and then let the receiver decide which one was correct. This leads us to list decoding.

Definition 7 *Code C is (e, l) list decodable if for any pattern of e errors, there exists a list of size l that includes the transmitted codeword.*

More formally we can say that $\forall y \in F^n, |\{|Ball(y, e) \cap C\} \leq l$.

Relation ships between the various parameters.

- All (n, k, d) codes are $((d-1)/2, 1)$ list decodable and vice versa. This is pretty obvious from the definition of the distance of the code.

- All (n,k,d) codes are $(n - sqrtn(n-d), poly(n))$ list decodable. That is for any (n,k,d) code, there exists a decoder which can output a relatively short list and correct around $n - sqrtn(n-d)$ errors. This is very useful when d is relatively large compared to n.

We will now prove the second item in the above list.

Claim 8 *Any (n,k,d) code is $(n - sqrtn(n-d), poly(n))$ list decodable.*

Proof Say we received a vector $y = (y_1, y_2, \ldots y_n$ and the output list i.e. the list of codewords within distance e of y are $c_1, c_2, \ldots, c_m$. We can construct an agreement graph with n nodes representing $y_1, \ldots y_n$ on the left and m nodes representing $c_1, \ldots, c_m$ on the right. There is an edge from y_i to c_j of $y_i = c_{j_i}$, as shown in 2.

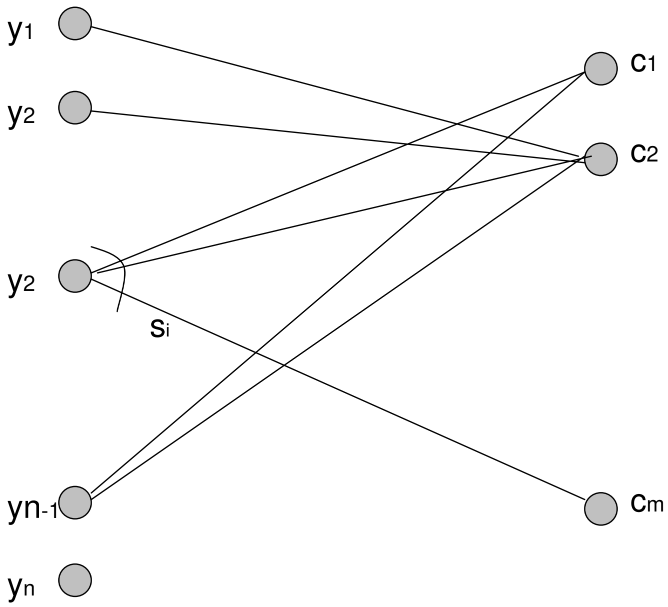

Figure 2: Agreement graph between the received vector and the list

Let $t = n - e$ and $s = n - d$. Then every code c_j there exist at least t indices i such that $(c_j)_i = y_i$. Thus the degree of any right vertex $\geq t$, Furthermore if c_1 and c_2 have y_i as a common neighbor, it means that they agree on that coordinate. But the distance between any two codewords is at least d. Thus number of common neighbors between any two vertices on the right $\leq n - d = s$.

Thus the graph G has no $K_{s+1,2}$ (a complete bipartite graph consisting of $s+1$ vertices on the left and 2 vertices on the right). We now bound the number of right vertices such a graph may have when restricted to having right degree at least t.

Let T denote the total number of edges in the graph and let s_i be the degree of the ith right vertex (representing y_i). Note that $\sum_i s_i = T$ and $T \geq tM$.

To bound the number of right vertices, we pick two random distinct right vertices, say, $j_1 and j_2$. Let X_i be the indicator random variable which is 1 if ith node on the left is a neighbor of both j_1 and j_2. Let X be the random variable denoting the number of common neighbors of j_1 and j_2. Then we have

$$
\begin{aligned}
\mathrm{Exp}[X_i] &= \Pr[X_i = 1] = \frac{\binom{s_i}{2}}{\binom{M}{2}} \\
\mathrm{Exp}[X] &= \sum_{i=1}^{n} \mathrm{Exp}[X_i] \\
&= \frac{\sum_i \binom{s_i}{2}}{\binom{M}{2}} \\
&= \frac{1}{M(M-1)} \left(\sum_{i=1}^{n} s_i^2 - \sum_{i=1}^{n} s_i \right) \\
&= \frac{1}{M(M-1)} \left(\sum_{i=1}^{n} s_i^2 - T \right) \\
&\geq \frac{1}{M(M-1)} \left(T^2/n - T \right) \ \text{(Using the inequality } \textstyle\sum_{i=1}^{n} a_i^2 \geq (\sum_i a_i)^2/n) \\
&= \frac{T(T-n)}{nM(M-1)} \\
&\geq \frac{tM(tM-n)}{nM(M-1)} = \frac{t^2M - nt}{nM - n}
\end{aligned}
$$

On the other hand, we are given that j_1 and j_2 have at most s common neighbors, and thus $\mathrm{Exp}[X] \leq s$. Thus we have $t^2M - nt \leq s(nM - n)$ which gives $M(t^2 - sn) \leq n(t - s)$ which in turn yields $M \leq n(t-s)/(t^2 - sn) \leq n^2$ provided $t^2 > sn$.

Using $s = n - d$ and $t = n - e$, we conclude that there are at most n^2 codewords in any ball of radius e around any vector y, provided $(n-e)^2 > n(n-d)$, or $e < n - \sqrt{n(n-d)}$.

■