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

Lecture 40: Folded Reed-Solomon Codes

December 03, 2007

Lecturer: Atri Rudra *Scribe: Michel Kulhandjian*

Recall soft decoding which was defined in the previous lecture.

Input: Given a code $C \in \mathbb{F}_q$ of block length n, $w_{i,\alpha}$ and a threshold $W \geq 0$, where weights $w_{i,\alpha} \geq 0$, $1 \leq i \leq n$, $\alpha \in [q]$.

Output: All codewords $(c_1, \ldots, c_n) \in C$ such that $\sum_{i=1}^{n} w_{i,c_i} > W$.

In GS decoding algorithm the inputs are $(\alpha_i, y_i) \in \mathbb{F}_q^2$ which need not be distinct. Let N be the number of (α_i, y_i)-tuples where α_i can be repeated and $N \leq q^2$. GS decoder can handle:

$$\sum_{i=1}^{n} w_{i,c_i} > \sqrt{k \sum_{i=1}^{N} \binom{w_{i,\alpha_i}+1}{2}} \tag{1}$$

Note that n is the code length of RS codes. Soft decoding can be implemented using inputs of GS decoder.
Input: (α_i, β), $1 \leq i \leq n$, $\beta \in \mathbb{F}_q$. Can do soft decoding for RS codes and by (1) we get,

$$\sum_{i=1}^{n} w_{i,c_i} > \sqrt{k \sum_{i=1}^{n} \sum_{\beta \in \mathbb{F}_q} \binom{w_{i,\alpha_i}+1}{2}} \triangleq W \tag{2}$$

Special case of soft decoding is List recovery where $w_{i,\alpha}$ will satisfy specific property. Given $S_i \subseteq \mathbb{F}_q$ for $1 \leq i \leq n$,

$$w_{i,\alpha} = \begin{cases} 1 & \text{, if } \alpha \in S_i \\ 0 & \text{, if otherwise} \end{cases}$$

The inputs and outputs of the algorithm as follows,
Input: S_i, $1 \leq i \leq n$ where $|S_i| \leq l$.
Output: All codewords $(c_1, \ldots, c_n)$ such that $c_i \in S_i$ for at least t values of i.

List Recovery for RS: Pick

$$w_{i,\alpha} = \begin{cases} r & \text{, if } \alpha \in S_i \\ 0 & \text{, if otherwise} \end{cases}$$

Condition (2) is the same as

$$rt > \sqrt{kln\binom{r+1}{2}}$$

where $t > \sqrt{kln}$, by picking r to be large enough. If $l = 1$, RS can be list decoded upto the Johnson bound. Now we need to ask a question whether RS codes be decoded in poly-time beyond $1-\sqrt{R}$ fraction of errors? It is unknown by 1 list recovery bound of $n-\sqrt{kn}$ is tight for RS codes. There exists $n - \sqrt{kn}$ examples where it can go beyond $l = \lceil \frac{n}{k} \rceil$ implies super-poly output list size[2]. There exist an explicit code with efficient list decoding algorithm that can correct beyond $1 - \sqrt{R}$ fraction of errors(maybe upto $1 - R$) [3, 1]. Note that we can not go beyond $1 - R$, this is the list decoding capacity.

1 Folded Reed-Solomon Codes

We are going to use a simple variant of Reed-Solomon codes called folded Reed-Solomon codes for which we can beat the $1 - \sqrt{R}$ decoding radius possible for RS codes. In fact, by choosing parameters suitably, we can decode close to the optimal fraction $1 - R$ of errors with rate R. This is special case of Parvaresh Vardy codes.

Let $m \geq 1$ be an integer parameter called the "folding parameter" . Start with $[n = q - 1, k]_q$ RS code and assume m divides n. We can remove this restriction if for example $m = 2$. For RS, we define mapping such as:

$$f \mapsto \langle f(\gamma^0), f(\gamma^1), \ldots, f(\gamma^{n-1}) \rangle$$

where $\mathbb{F}_q = \{0, 1, \gamma, \gamma^2, \ldots, \gamma^{n-1}\}$

Definition 1.1. *The m-folded version of the RS code C, denoted $FRS_{\mathbb{F},\gamma,m,k}$ is a code of block length $N = n/m$ over $\mathbb{F}^m$. $FRS_{\mathbb{F}_q,\gamma,m,k}$ are just $[q - 1, k]$ RS codes with m consecutive symbols from RS codewords grouped together, that is:*

$$f \mapsto \langle f(\gamma^i), f(\gamma^{i+1}), \ldots, f(\gamma^{i+m-1}) \rangle$$

where $i \in \{0, m, \ldots, m(\frac{n}{m} - 1)\}$.

Note FRS is over $\mathbb{F}_{q^m}$. Block length, $N = \frac{n}{m}$, Dimension, $K = \frac{k}{m}$ need not be an integer. Note that the folding operation does not change the rate $R = \frac{k}{n} = \frac{K}{N}$ of the original Reed-Solomon code. And the distance $\geq N - K$. This is because if two codewords agree at least in $K + 1$ places which implies two RS codewords has $n + k$ which is a contradiction. Note that FRS no longer linear, it is $\mathbb{F}_q$-linear as RS is $\mathbb{F}_q$-linear. Decoding the folded RS code up to a fraction p errors is certainly not harder than decoding the RS code up the same fraction p of errors. This is because p fraction of errors over $\mathbb{F}_{q^m}$ is equivalent p fraction of errors over $\mathbb{F}_q$ "unfolded" version. We run GS decoder to correct upto $1 - \sqrt{R}$ fraction of errors. Since folding seems like such a simplistic operation, and the resulting code is essentially just a RS code but viewed as a code over a large

alphabet, let us now understand why it can possibly give hope to correct more errors compared to the bound for RS codes. Say we want to correct $\frac{1}{2}$ fraction of errors. Then if we use the RS code, our decoding algorithm should able to correct an error patters that corrupts every other symbols. However, after the folding operation, this error pattern corrupts every one of the symbols over the larger alphabet $\mathbb{F}^2$, and thus need to correct this error pattern.

References

[1] Venkatesan Guruswami and Atri Rudra. Achieving list decoding capacity using folded reed-solomon codes. *Alerton*, 2006.

[2] Venkatesan Guruswami and Atri Rudra. Limits to list decoding reed-solomon codes. *IEEE Transactions on Information Theory*, 52(8):3642–3649, August 2006.

[3] F. Parvaresh and A. Vardy. Correcting errors beyond the guruswami-sudan radius in polynomial time. *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 285–294, 2005.