6.440 Essential Coding Theory March 20, 2013

Lecture 13

Lecturer: Madhu Sudan *Scribe: Badih Ghazi*

This lecture is about list-decoding folded Reed-Solomon codes. Folded Reed-Solomon codes will be list-decodable codes from a $1 - R - \epsilon$ fraction of errors where R is the rate and $\epsilon > 0$. This class of codes was introduced by Guruswami and Rudra [GR06] and was inspired by the work of Parvaresh and Vardy [PV05]. The particular algorithm that we will describe is due to Guruswami [Gur11].

1 History

This line of work has an interesting history that we now briefly describe. Fix a prime power q and let $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$. The message will consist of 2 polynomials $p_1(x), p_2(x) \in \mathbb{F}_q[x]$ of degree less than k. The codeword will consist of the evaluations of p_1 and p_2 on the set $\{\alpha_i \mid i \in [n]\}$. Instead of viewing this code as having length $2n$ over $\mathbb{F}_q$, we will view it as having length n over $\mathbb{F}_q^2$. Note that $\mathbb{F}_q^2$ is not a finite field. Such codes are usually referred to as "Interleaved Reed-Solomon Codes".

1.1 Coppermisth-Sudan

Coppermisth and Sudan gave an algorithm for list-decoding interleaved codes but they needed to assume that the error pattern was "random" [CS03]. More precisely, the assumption was that some symbols are received uncorrupted whereas other symbols are completely random. The given algorithm recovers from a $1 - O((k/n)^{2/3})$ fraction of "random" errors.

1.2 Parvaresh-Vardy

Instead of letting $p_1(x)$ and $p_2(x)$ be unrelated, Parvaresh and Vardy related them in the following way. Let $\mathbb{F}_q^{<k}[x]$ be the ring of all polynomials of degree $< k$ and with coefficients in $\mathbb{F}_q$. We define an operator $\Phi : \mathbb{F}_q^{<k}[x] \to \mathbb{F}_q^{<k}[x]$ and let $p_2(x) = \Phi(p_1(x))$. They were able to show that if the operator Φ is "nice", then there is an algorithm that can recover from a $1 - (k/n)^{2/3}$ fraction of errors in the adversarial setting. The drawback of this scheme is that the rate got divided by 2. In fact, since $p_2(x)$ is determined by $p_1(x)$, k/n is now twice the rate. Thus, this code is list-decodable for a $1 - (2R)^{2/3}$ fraction of errors. This code can be generalized for any constant $c \in \mathbb{N}$ where we can recover from a $1 - (cR)^{c/(c+1)}$ fraction of errors. It looked as if the factor c next to the rate R was inevitable until Guruswami and Rudra came up with the following idea.

1.3 Guruswami-Rudra

Guruswami and Rudra [GR06] introduced the following "nice" operator ϕ. We fix a primitive element ω of $\mathbb{F}_q^*$ and we let ϕ map $p_1(x)$ to $p_1(\omega x)$. Note that the choice of the operator in the Pavaresh-Vardy construction does not change the rate of the code. The first row of the interleaved codeword looks like $p(\omega), p(\omega^2), \ldots, p(\omega^{q-1})$ whereas the second row looks like $p(\omega^2), p(\omega^3), \ldots, p(\omega^q)$. All those symbols can be read from $p(\omega)$,...,$p(\omega^q)$. Thus, the rate went up from $\frac{1}{2}\frac{k}{n}$ to $\frac{4}{5}\frac{k}{n}$. Thus, this code can recover from a $1 - (\frac{5}{4}R)^{2/3}$ fraction of errors, which is an improvement over the Parvaresh-Vardy codes. The Guruswami-Rudra codes can be further generalized so that we can recover from a $1 - (\frac{(m+s)R}{m})^{\frac{s}{s+1}}$ fraction of errors where s is the number of blocks that we are interleaving and the alphabet is $\mathbb{F}_q^m$. We will next present a simplified version of the Guruswami-Rudra codes due to Guruswami [Gur11] that allows us to recover

from an $\frac{s}{s+1}(1-\frac{(m+s)}{m}R)$ fraction of errors. Note that this is slightly worse than the performance of the Guruswami-Rudra codes.

2 Description and analysis of the folded Reed-Solomon codes

The number of blocks s that we are interleaving is a parameter of the algorithm. In the following description, we will assume for simplicity that $s = 2$. Moreover, we let $m = 2$ i.e., the code will be over the alphabet $\mathbb{F}_q^2$.

2.1 Encoding

The encoding of the message corresponding to the polynomials $p_1(x), p_2(x) \in \mathbb{F}_q^{<k}[x]$ consists of 2 vectors $(\beta_1, \ldots, \beta_n)$ and $(\gamma_1, \ldots, \gamma_n)$ where $\beta_i = p_1(\alpha_i)$ and $\gamma_i = p_2(\alpha_i)$ for all $i \in [n]$.

2.2 List Decoding

The goal of list-decoding is to find all polynomials $p(x) \in \mathbb{F}_q[x]$ of degree less than k s.t.

$$|\{i \mid \beta_i = p(\alpha_i) \text{ and } \gamma_i = p(\omega\alpha_i)\}| \geq n/3 + (2k)/3 \tag{1}$$

Note that this is better than unique decoding since $n/3$ appears in the equation as opposed to $n/2$. A priori, it is not clear that the number of polynomials that satisfy Equation (1) is small. We start by finding some "algebraic explanation" for the polynomials that satisfy this equation. Namely, we find $Q(x, y, z) = A(x) + B(x)y + C(x)z$ s.t. $Q(x, y, z)$ is not identically equal to 0 but $Q(\alpha_i, \beta_i, \gamma_i) = 0$ for all $i \in [n]$, $\deg(A) < \frac{n+2k}{3}$, $\deg(B) \leq \frac{n-k}{3}$ and $\deg(C) \leq \frac{n-k}{3}$. Note that this is a system of linear homogeneous equations.

Claim 1. *$Q(x, y, z)$ can be found.*

Proof. This can be done by solving the big linear homogeneous system of equations above. □

We can thus find $A(x), B(x), C(x)$ which determine $Q(x, y, z)$.

Claim 2. *$Q(x, p_1(x), p_2(x))$ is identically equal to 0 provided that the number of agreements is at least $\frac{n}{3} + \frac{2k}{3}$.*

Proof. Define $g(x) = Q(x, p_1(x), p_2(x))$. Since $\deg(g) < \frac{n}{3} + \frac{2k}{3}$ and since $g(\alpha_i) = 0$ for all i s.t. $p_1(\alpha_i) = \beta_i$ and $p_2(\alpha_i) = \gamma_i$, we conclude that g is identically equal to 0. □

Thus, the set of all $p_1(x), p_2(x) \in \mathbb{F}_q^{<k}[x]$ s.t. $A(x) + B(x)p_1(x) + C(x)p_2(x) = 0$ and $p_2(x) = p_1(\omega x)$ includes the polynomials that we are interested in. Note that we can write $p_1(x)$ and $p_2(x)$ as follows

$$p_1(x) = \sum_{i=0}^{k-1} c_i x^i$$

$$p_2(x) = \sum_{i=0}^{k-1} \tilde{c}_i x^i$$

where $c_i, \tilde{c}_i \in \mathbb{F}_q$ for all $i \in \{0, 1, \ldots, k-1\}$. We need to find all such $p_1(x)$ and $p_2(x)$ and ensure that there are not too many of them. To do so, we use the following nice idea from the work of Pavaresh and Vardy. We mod out $F_q[x]$ by some irreducible polynomial $h(x) \in F_q[x]$ in order to get a field. Guruswami and Rudra picked $h(x) = x^{q-1} - \omega$. We will use the following fact.

Fact 3. *If ω is a primitive element of $\mathbb{F}_q^*$, then $x^{q-1} - \omega$ is irreducible over $\mathbb{F}_q$.*

Then, $\mathbb{K} = \mathbb{F}_q[x]/(x^{q-1} - \omega)$ is a field. Note that in $\mathbb{K}$, we have that $\omega = x^{q-1}$ which implies that $\omega x = x^q$. Thus, we have that $p_2(x) = p_1(\omega x) = p_1(x^q) = (p_1(x))^q$. We need to solve the equation $A(x) + B(x)p_1(x) + C(x)(p_1(x))^q = 0$ where $A(x), B(x), C(x) \in \mathbb{K}$ are given and $p_1(x) \in \mathbb{K}$ is the unknown. We can find the set of all solutions $(p_1(x), p_2(x))$ by factoring the polynomial $f(x) = A(x) + B(x)p_1(x) + C(x)(p_1(x))^q$ in $\mathbb{F}_q[x]$. Since $f(x)$ has degree q, the number of solutions is at most q. Thus the size of the list is at most q. For general values of s, this construction yields a list size bound of q^{s-1}. We next consider the question of whether we can somewhat reduce the list size.

3 Reducing the list size

The Guruswami-Rudra construction and the Guruswami construction both have the property that for sufficiently large values of s and m (that depend on ϵ), the list decoding radius exceeds $1 - R - \epsilon$ but the list size bound is $n^{\Omega(\frac{1}{\epsilon})}$ which has a rather poor dependence on the distance parameter ϵ to the optimal tradeoff. Note that existentially a list size as small as $O(1/\epsilon)$ is possible. In order to reduce the list size, Guruswami [Gur11] came up with the proposal which was later implemented of Dvir and Lovett [DL12]. Since the polynomial $p_1(x)$ comes from an $(s-1)$-dimensional affine subspace, we can restrict the space of all possible messages to those that don't have too many polynomials in any particular low-dimensional subspace. Previously, we were thinking of $p_1(x)$ as an element of $\mathbb{F}_q^k$. Now we think of $p_1(x)$ as an element of $S \subseteq \mathbb{F}_q^k$ where S has the following property: For every affine subspace $L \subseteq \mathbb{F}_q^k$ s.t. $\dim(L) = s - 1$, $|L \cap S|$ is "small" i.e. $|L \cap S| \leq \tau = \text{poly}(1/\epsilon)$. Such as set S is said to be (s, τ)-evasive. Guruswami proved existentially that $|S| \geq q^{(1-o(1))k}$ and Dvir and Lovett later gave an explicit construction.

4 Linearity of the codes

Note that the types of codes described above are not linear. Guruswami and Rudra noted that by an "appropriate composition" with an appropriate code, we can get the following statement.

Theorem 4. *([GR06])*
For all $\epsilon, R > 0$, there exists $q, n \in \mathbb{N}$, and $\mathbb{F}_q$-linear codes of length n over $\mathbb{F}_q$, rate R and that are list-decodable from a $1 - R - \epsilon$ fraction of errors, with lists of size $n^{\text{poly}(1/\epsilon)}$.

References

[CS03] Don Coppersmith and Madhu Sudan. Reconstructing curves in three (and higher) dimensional space from noisy data. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 136–142. ACM, 2003.

[DL12] Zeev Dvir and Shachar Lovett. Subspace evasive sets. In *Proceedings of the 44th symposium on Theory of Computing*, pages 351–358. ACM, 2012.

[GR06] Venkatesan Guruswami and Atri Rudra. Explicit capacity-achieving list-decodable codes. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 1–10. ACM, 2006.

[Gur11] Venkatesan Guruswami. Linear-algebraic list decoding of folded reed-solomon codes. In *Computational Complexity (CCC), 2011 IEEE 26th Annual Conference on*, pages 77–85. IEEE, 2011.

[PV05] Farzad Parvaresh and Alexander Vardy. Correcting errors beyond the guruswami-sudan radius in polynomial time. In *Foundations of Computer Science, 2005. FOCS 2005. 46th Annual IEEE Symposium on*, pages 285–294. IEEE, 2005.