6.895 Essential Coding Theory	November 29, 2004
Lecture 23	
Lecturer: Madhu Sudan	*Scribe: Swastik Kopparty*

1 The rest of the course

- TaShma-Zuckerman-Safra extractor
- Guruswami's List Decodable codes
- Capalbo-Reingold-Vadhan-Wigderson Zig-zag product for expanders with good vertex expansion
- Locally Testable and Decodable Codes

2 T-Z-S Continued

First a quick recap of what is going on:

- We are extracting from an n bit source.
- We are working over $\mathbb{F}_q$, with $q \approx \sqrt{n}$.
- We use a small code $\mathcal{C}_{small} : \mathbb{F}_q \to \{0,1\}^l$, list decodable from $\frac{1}{2} - \delta$ errors with polynomial size lists.
- View the output of the weak random source as giving a polynomial P of degree $\sqrt{n}$ in each variable.
- Our seed consists of the tuple $((a,b),j)$, with $a,b \in \mathbb{F}_q, j \in [l]$.
- The extracting function is $E(P,((a,b),j)) = (\mathcal{C}_{small}(P(a+1,b))_j, \mathcal{C}_{small}(P(a+1,b))_j, \ldots \mathcal{C}_{small}(P(a+m,b))_j)$
- This extractor can, for example, when the source has at least $k = n^{3/4}$ bits of min-entropy, extract $m = n^{1/4}$ output bits.
- This was generalized to work for better parameters in the paper of Shaltiel and Umans.

2.1 Analysis Continued

Another quick recap of what we were trying to do in the proof:

- Suppose $\exists A : \{0,1\}^m \times \mathbb{F}_q^2 \times [l] \to \{0,1\}$, and X, $|X| = 2^k$ with
$$Pr_{x \in_R X, y}[A(E(x,y),y) = 1] > Pr_{z,y}[A(z,y) = 1] + \epsilon$$
- **Step 1 (the usual):** Convert our distinguisher to a predictor:
$$\exists i \leq m, \epsilon' > 0, B : \mathbb{F}_q^{i-1} \times \mathbb{F}_q^2 \to \mathbb{F}_q, X', |X'| > large$$
such that
$$\forall P \in X', Pr_{a,b}[B(P(a+1,b), P(a+2,b), \ldots P(a+i-1,b)) = P(a+i,b)] > \epsilon'$$
for some $\epsilon' = poly(\epsilon, 1/m)$.
- **Step 2 (the interesting part):** Use the predictor to conclude that there is a short description for the elements in X' , forcing $|X| <$ something.

Now, the details for Step 1.

- We focus on those polynomials where the predictor always works well; i.e. we go from $P \in_R X$ to $P \in_R X_{\epsilon/2}$, $|X_{\epsilon/2}| > \epsilon/2|X|$, such that

$$\forall P \in X_{\epsilon/2}, Pr_y[A(E(P,y),y) = 1] > Pr_{z,y}[A(z,y) = 1] + \epsilon/2$$

- By hybridization, we can focus on one predictable bit: $\exists i, b_{i+1}, \dots b_m$ s.t. $\forall P \in X_{\epsilon/2}$,

$$Pr_{y=((a,b),j)}[A(E(P,y)_1, E(P,y)_2, \dots E(P,y)_i, b_{i+1}, \dots b_m) = 1]$$
$$> Pr[A(E(P,y)_1, \dots E(P,y)_{i-1}, b_i, b_{i+1} \dots b_m) = 1] + \epsilon/2m$$

- Now we concentrate only on those (a,b) which allow E to be prone to prediction:

$$S = \{(a,b) : Pr_j[A(E(P,((a,b),j))_1, \dots E(P,((a,b),j)_i, b_{i+1}, \dots b_m) = 1]$$
$$> Pr[A(E(P,((a,b),j))_1, \dots b_i, b_{i+1}, \dots b_m) = 1] + \epsilon/4m$$

  By Markov, $\frac{|S|}{|\mathbb{F}_q|^2} \geq \epsilon/4m$.

- Now we make the predictor B. It first tries to guess (using the above property of E) for every $j \in [l]$, the value of $C_{small}(P(a+i,b))_j$. Given these values, it list decodes C_{small} to get a small list of candidates, and outputs one of them at random. This will get the right answer with probability $\epsilon' = poly(\epsilon/4m)$.

Given this predictor B, we will now reconstruct P by taking only a few bits of non-uniform advice. This will allow us to bound the maximum possible size of X.

Our reconstructor works as follows. First pick a random pair $c, d \in \mathbb{F}_q^2$. Ask for $P|_{L_j}$ for $j = 1, \dots i-1$, where L_j is the line $\{c + j + td : t \in \mathbb{F}_q\}$. This is $2\sqrt{n}m$ elements. Then, use B to predict the possible values of P on L_i. Then, by the list decodability of Reed-Solomon codes, narrow down the possibilities for $P|_{L_i}$. Finally, ask the non-uniform advisor for which one is actually correct (this requires a very small number of bits). This can be repeated $\sqrt{n}$ times when enough values of the polynomial are known to completely reconstruct it.

This procedure will succeed if for every line, we guess enough values correctly for the list-decoding to work. We choose parameters so that given $\epsilon'/2$ correct values on a line, the list of possible codewords is . The following calculation show that this will

Let $S = \{(a,b) : B(P(a+1,b), P(a+2,b), \dots P(a+i-1,b)) = P(a+i,b)$. Call a line L good if $|L \cap S| \geq \frac{\epsilon'}{2}|\mathbb{F}_q|$. By Chebyshev's inequality, probability that L is not good $< 4\sigma^2/\epsilon'^2 < \frac{4}{\epsilon' q}$. Thus the probability that all lines involved are good is at least $1 - O(\frac{\sqrt{n}}{\epsilon' q})$.

Thus with a total of $2m\sqrt{n} + (small)\sqrt{n}$ bits of advice, we can reconstruct any polynomial in X' completely. This limits the size of X' to have at most $2^{O(2m\sqrt{n})}$ polynomials, implying that X has to be small. Thus for large enough X, E is an extractor.

3 Guruswami Codes

Guruswami codes combine 3 of the constructions that we saw in an ingenious way to produce codes with non-trivial list decodability properties. In particular, if one uses the TaShma-Zuckerman-Safra extractor to get list-decodable codes using the canonical TaShma-Zuckerman equivalence, and then plug this in (as the "left hand side" code) the Alon-Edmonds-Luby expander based code construction, we get Guruswami codes. These codes have $O(1)$ alphabet size, rate $O(\epsilon)$ and can be list decoded from $1-\epsilon$ fraction errors with lists of size $2^{\sqrt{n}}$. Further, there is a $O(2^{\sqrt{n}})$ time algorithm that can find this list. Until now, we did not even know about the existence of codes with these parameters.