**CSE 533: Error-Correcting Codes** **(Autumn 2006)**

# Lecture 15 and 16: List Decoding

November 22, 2006

*Lecturer: Atri Rudra and Venkatesan Guruswami* *Scribe: Jacob Nelson*

## 1 List Decoding

Our $(1-R)/2$ bound came from the fact that we can't uniquely decode a codeword that is a distance of $(1-R)/2$ from two valid codewords. But there are many codewords outside the $(1-R)/2$ radius; we could uniquely decode many of them if we simply chose the nearest valid codeword instead.

In higher dimensions, most of the ambient space is outside this radius.

We can do better by outputting a *list* of answers. In most cases, this list will be of size 1. This is called *List Decoding*, and was discovered independently by Elias and Wozencraft in the late 50's.

More formally, we say that given a code $C$, error bound $e$, and received word $y$, we will output all $c \in C$ such that $\Delta(c,y) \leq e$.

In other words, we take a ball of certain radius around the received codeword and output all the codewords that fall inside this ball. Maybe this doesn't let us decode arbitrary points, but for typical probabilistic models, this will give us mostly unique codewords. Furthermore, outputting a small list is better than failing with no information about why.

List decoding introduces a new parameter $\ell$, the maximum number of things to output in the worst case. We would like it to be small. So for all $p \in \Sigma^n$,

$$|B(y, pn) \cap C| \leq \ell.$$

**Definition 1.1.** *A code $C$ is called an $(e,\ell)$-list-decodable if for all received code $y$,*

$$|\{c | \Delta'(c,y) \leq e\}| \leq \ell.$$

For "efficient" list decoding, we need $\ell$ to be polynomial in length in $C$; that is, the worst case list length must be polynomial.

**Definition 1.2.** *The* List Decoding Radius *(LDR) is the largest $e$ such that a code $C$ is $(e,\ell)$-list-decodable for a polynomial $\ell$.*

Recall the Johnson bound over a field $\mathbb{F}_q$:

$$J_q(\delta) = \frac{q-1}{q}\left(1 - \sqrt{1 - \frac{q\delta}{q-1}}\right).$$

In the binary case, this is

$$J_2(\delta) = \frac{1}{2}\left(1 - \sqrt{1-2\delta}\right).$$

If $e < J_q(\delta)$, then the LDR must be $\geq e/n$.

Alphabet-free version:

The number of codewords is less than $nd$ if $e \leq n - \sqrt{n(n-d)}$. Since $d \leq n-k+1$ and $R = k/n$, if the Hamming ball has radius at most $1-\sqrt{R}$, the number of codewords is $\leq nd$.

## 1.1 List decoding and rate

Let $R_q(p)$ be the largest rate for codes with LDR $p$.

**Theorem 1.3.**

$$R_{\ell,q}(p) \geq 1 - H_q(p) - \frac{1}{\ell}.$$

**Note.** $\ell = O(\frac{1}{\varepsilon})$, so the rate is $\geq 1 - H_q(p) - \varepsilon$. (This is the *list decoding capacity*.) As $\ell$ goes to infinity, $R_{\ell,q}$ goes to $1 - H_q(p)$, so for large list sizes, we achieve capacity.

**Proof Sketch.** Pick a random code $C$ of length $n$.

What is the probability that $\ell+1$ codewords lie in one ball of radius $pn$? The probability that *one* codeword lies in some Hamming ball is

$$\frac{\mathrm{Vol}_q(\Sigma, e)}{q^n},$$

so the probability that $\ell+1$ do is

$$\frac{\mathrm{Vol}_q(\Sigma, e)}{q^n} \leq q^{(H_q(p)-1)n(\ell+1)}.$$

Let $N_{\text{bad}}$ be the number of sets of $\ell+1$ codewords that lie in some ball of radius $pn$. Then

$$E\left(N_{\text{bad}}\right) \leq q^n \left( \binom{M}{\ell+1} q^{(H_q(p)-1)(\ell+1)n} \right).$$

If we choose the rate

$$M = q^{\left(1 - H_q(p) - \frac{1}{\ell}\right)n},$$

then

$$E\left(N_{\text{bad}}\right) < 1.$$

So there exists a code with $N_{\text{bad}} < 1$, and since it's an integer quantity, $N_{\text{bad}}$ must be 0. There is no Hamming ball of radius pn with more than polynomially many codewords.

**Theorem 1.4.** *For every a and $p|0 < p < 1 - 1/q$ (noise) and $\ell \geq 1$, there exists a family of q-ary codes of rate R that are $(p, L)$-list-decodable for $R = 1 - H_q(p) - \frac{1}{L+1} - O(1)$.*

**Note.** As $L$ grows bigger and bigger, we are able to achieve a rate of $1 - H(p)$, achieveing Shannon's result.

**Proof Sketch.** We will use random coding.

We have an ambient space of $q^n$, with alphabet $\Sigma = \{0, 1, \ldots, q-1\}$.

Pick $M$ codewords $c_1, c_2, \ldots, c_M$, where each $c_i$ is a uniformly-randomly-chosen element of $\Sigma^n$. (This is the same as we did for the proof of Shannon's theorem.) So $M = q^{Rn}$.

A bad event is defined as follows: there exists some $y \in \Sigma^n$ and some subset $S \subset L$ of size $L+1$ such that $B(y, pn) \geq S$.

Fix $y, S$.

If $B(y, pn) \geq S$, then the entire set $S$ lies in close proximity to $y$.

What's the probability of this? By our work on estimating balls,

$$\begin{aligned}\Pr[B(y, pn) \geq S] &= y\left(\frac{|B(y, pn)|}{q^n}\right)^{\ell+1} \\ &\leq \left(\frac{q^{H_q(p)n}}{q^n}\right)^{\ell+1}\end{aligned}$$

for a fixed $y \in S$.

So

$$\begin{aligned}\Pr[\text{Bad event}] &= \Pr[\text{C is not } (p, L) \text{ list decodable}] \\ &\leq q^n \binom{M}{C+1} \left(\frac{q^{H_q(p)n}}{q^n}\right)^{\ell+1} \\ &\leq q^n \left(q^{Rn}\right)^{\ell+1} \left(\frac{q^{H_q(p)n}}{q^n}\right)^{\ell+1}\end{aligned}$$

and collecting terms,

$$\begin{aligned}&= q^{(\ell+1)n(R+H_q(p)-\frac{\ell}{\ell+1})} \\ &= q^{(\ell+1)n(R-(1-H_q(p)-\frac{\ell}{\ell+1}))} \\ &< 1 \quad \text{if} \quad R = 1 - H_q(p) - \frac{1}{\ell+1} - O(1).\end{aligned}$$

If we have a linear code, then the codewords will not all be independent. But the chance of a collision is $\binom{M}{2} \cdot \frac{1}{q^n}$, and we can just include this in the error probability.

So now we have $1 - H_q(p)$ as the "list decoding capacity:" the largest rate we can list decode for error probability $p$.

**Lemma 1.5.** *If C is $(p, \ell)$-list-decodable and $\ell < \text{poly}(n)$ and $C \subset \Sigma^n$, then $R \leq 1 - H_q(p) + O(1)$.*

**Proof Sketch.** This can be proven by contradiction: if the rate were larger than this, C should not be list decodable, so we can pick a random center and show that there are too many code words in the ball around that center.

For the $q = 2$ case, the list decoding capacity is $1 - H(p)$: we can correct a fraction $p$ of worst-case errors and still have rate $1 - H(p) - \varepsilon$. We still get to capactiy even with worst-case errors. In reality, channels don't exactly match our models, so this is more robust.

## 1.2 A toy problem

This problem can be found in a paper by Ar, Lipton, Rubinfeld, and Sudan from 1992.

Take a Reed-Solomon code $[n, k+1, n-k]_{\mathbb{F}}$. Say we have the message $P_1(x)$ with $\deg(P_1) \leq k$. Suppose $n/2$ of the values are corrupted in the following way: the output $y_0 y_1 \dots y_{n-1}$ has $y_i$ either $P_1(\alpha_i)$ or $P_2(\alpha_i)$, and there exist at least $n/2$ values where $y_i = P_1(\alpha_i) = P_2(\alpha_i)$. Clearly in this case we should output both $P_1$ and $P_2$.

We know that at each point, $y_i$ is either from $P_1$ or $P_2$. How do we capture the "or" in this statement? We'll write

$$(y_i - P_1(\alpha_i))(y_i - P_2(\alpha_i)) = 0;$$

one of these must be 0. We write instead

$$Q(x_i, y_i) = (y_i - P_1(x_i))(y_i - P_2(x_i)),$$

so that we know $Q(x_i, y_i) = 0$ for all $i$. Call this Condition 1.

We find this polynomial by expanding $Q$:

$$y_i^2 - (P_1(x_i) + P_2(x_i))y_i + P_1(x_i)P_2(x_i) = 0.$$

Now we ignore the sum and product, stealing a degree:

$$y_i^2 - B(x_i)y_i + C(x_i) = Q(x_i, y_i)$$

where $\deg(B) \leq k$ and $\deg(C) \leq 2k$. Call this Condition 2.

Since the constraints are linear in the coefficients, we can find $B$ and $C$ efficiently. Thus, we define the following two-step algorithm:

1. Find nonzero $Q$ meeting Condition 2 such that $Q(\alpha_i, y_i) = 0$ for $i = 0, 1, \dots, n-1$.
2. Factor $Q$ as $Q(x_i, y_i) = (y_i - P_1(x_i))(y_i - P_2(x_i))$ and output $P_1(x_i)$.

We know that a solution exists to the first part. To prove the second part, let $R(x) = Q(x, P_1(x))$. Note that $\deg(R(x)) \leq 2k$. Since $R(\alpha_i) = 0$ whenever $P_1(\alpha_i) = y_i$, and this happens at least $n/2$ times, if $n/2 > 2k$, then $R(x) = 0$. (This works if $k < n/k$.)