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

Lecture 6: The Elias-Bassalygo Bound

October 18, 2006

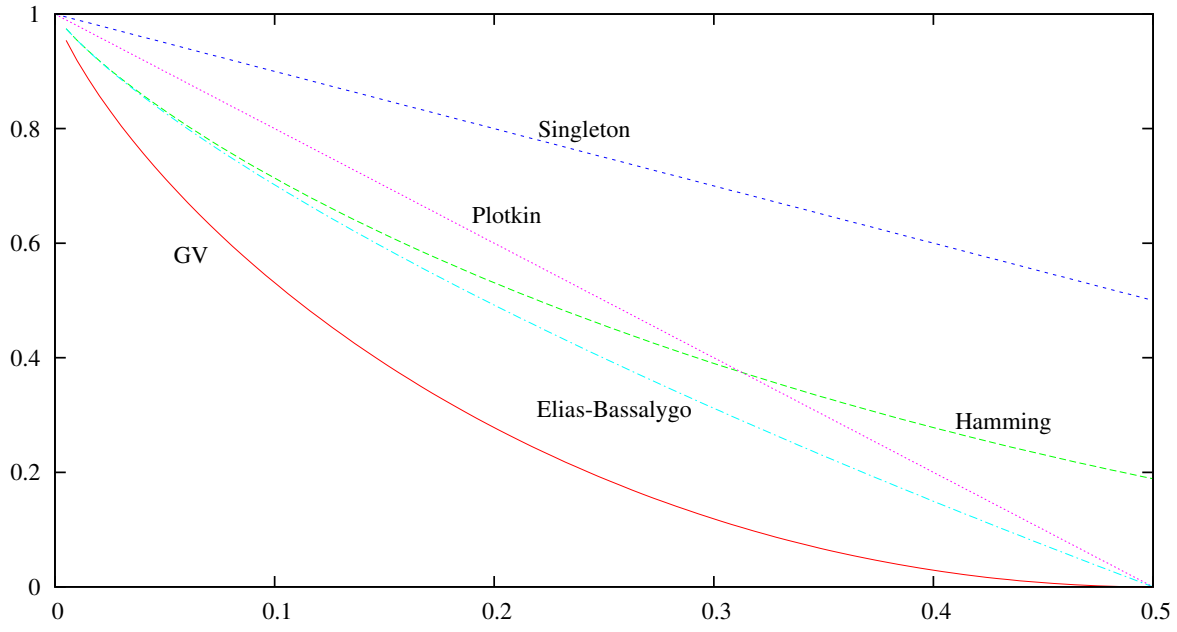
Lecturer: Venkatesan Guruswami *Scribe: Matt Cary*

1 Recapitulation

We currently know the following four bounds on rate as a function of relative distance, three of them upper bounds (which tell us what rate-distance combinations are impossible), and one lower bound (which tells us what rate-distance combinations we can achieve). In the following, R is the rate and δ the relative distance of a code. For example, a (n, k, d) code has $R = k/n$ and $\delta = d/n$. $H_q(x)$ is the q-ary entropy function, $H_q(x) = x \log_q\big((q-1)/x\big) + (1-x)\log_q\big(1/(1-x)\big)$.

Gilbert-Varshamov:	$R \geq 1 - H_q(\delta)$	(1)
Volume (Hamming):	$R \leq 1 - H_q(\delta/2)$	(2)
Singleton:	$R \leq 1 - \delta$	(3)
(Binary) Plotkin:	$R \leq 1 - 2\delta$	(4)

Below we plot these bounds, along with the Elias-Bassalygo bound we will prove today, for the binary case ($q = 2$).



Recall that the Shannon capacity can be achieved via decoding up to half the distance only if we have codes whose rate meets the Hamming upper bound (2). So before today's bound, any hope of achieving the Shannon bound using combinatorial codes, according to the above graph, is only possible with δ less than about 0.3. Today we will extinguish this hope by showing the Elias-Bassalygo upper bound that gives a strict improvement on all our current upper bounds, as the graph shows.

This Elias-Bassalygo bound is the best known bound that can be shown by elementary methods. The current world-record upper bounds extend this proof technique with bounds derived from linear programming duality.

2 The Johnson Bound

We begin by defining a parametrized bound known as the *Johnson bound*. Let n be a block length, d be a distance, and e be a radius. Then $J(n, d, e)$ is the maximum number of codewords contained in any ball of radius e, of any code with blocklength n and distance d. Hence $J(n, d, n)$ is our usual upper bound, usually denoted $A(n, d)$: the most number of codewords in any code of distance d and blocklength n. Looking $J(\cdot)$ from the other end, we have that $J(n, d, \lfloor (d-1)/2 \rfloor) = 1$, as in a distance d code, the balls of radius $\lfloor (d-1)/2 \rfloor$ do not intersect. As a side note, Johnson actually studied the maximum number of codewords of the specified distance all lying on the *surface* of the specified sphere, that is, $\mathrm{wt}(c) = e$ rather than $\mathrm{wt}(c) \leq e$ as we are using.

While there is a geometric proof of the Johnson bound, we will prove it using a useful combinatorial technique of counting two ways. Let $c_1, \ldots, c_M$ be M codewords, with $\Delta(c_i, c_j) \geq d$ for all $i \neq j$, and $\mathrm{wt}(c_i) \leq e$ for all i. Note that by translation we can assume without loss of generality that our sphere is centered at zero. We will now bound the sum all distances $S = \sum_{1 \leq i < j \leq M} \Delta(c_i, c_j)$ in two ways.

First, as $\Delta(c_i, c_j) \geq d$, we have that

$$S \geq \binom{M}{2} d. \tag{5}$$

Now, consider the codewords arranged in an $n \times M$ matrix, and look at the i-th column. Suppose this contains m_i 1's, and $M - m_i$ zeros. Then each pair of different bits contributes one to S, for a total of $m_i(M - m_i)$ per column. Define $\sum m_i/M = e'$ as the average weight per codeword, and note that $e' \leq e$ as $\mathrm{wt}(c_i) \leq e$ for all codewords. Hence

$$S = \sum m_i(M - m_i) = M^2 e' - \sum m_i^2.$$

As Cauchy-Schwartz tells us $\sum m_i^2 \geq \left(\sum m_i\right)^2 / n$, we have that

$$\leq M^2 e' - M^2 e'^2 / n.$$

Thus by combining this with (5), we have that

$$M(M-1)d \leq 2S \leq 2M^2(e' - e'^2/n).$$

Rearranging,

$$M(d - 2e' + 2e'^2/n) \leq d.$$

Provided the left-hand side is positive, we can divide to bound M by

$$\begin{aligned} M &\leq \frac{nd}{nd - 2e'n + 2e'^2} \\ &= \frac{2nd}{(n-2e')^2 - n(n-2d)} \\ &\leq \frac{2nd}{(n-2e)^2 - n(n-2d)}, \end{aligned}$$

as $e' \leq e$. Finally, if the denominator is positive, it must be at least one as it is an integer. Rearranging the denominator shows that it is positive when $e < (n - \sqrt{n(n-2d)})/2$. Hence we have that

$$J(n, d, e) \leq 2nd \quad \text{if } \frac{e}{n} < \frac{1}{2}\left(1 - \sqrt{1 - 2d/n}\right).$$

In other words, any ball with radius smaller than e as above, contains only polynomially many codewords. In fact, the quantity we are really interested in is *when* the Johnson bound holds, and not what the Johnson bound in fact is, just as long as it is polynomial.

Hence if we define $J(\delta) = (1 - \sqrt{1 - 2\delta})/2$, then in any binary code of relative distance δ, every Hamming ball of fractional radius less than $J(\delta)$ has only polynomially many codewords (as a function of the block length). We call $J(\delta)$ the *Johnson radius*.

as a bound on what the maximum size ball containing a polynomial number of codewords of relative distance δ can be. Though we will not prove it here, the Johnson bound can be extended to q-ary alphabets, and the q-ary Johnson radius is given by

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

There is also an alphabet-independent version that holds for all q,

$$J(n, d, e) \leq nd \quad \text{if } e < n - \sqrt{n(n-d)}.$$

3 Using the Johnson Bound

We know show how the Johnson bound can be used to give an upper bound for the coding problem. The technique we will use has been called the *fishnet* method, and is useful to know about. The outline is that we have a bound on how many codewords can fit into a ball of a certain size. We'll then show that there exists a ball containing relatively large fraction of the code, which when combined with our bound constrains the size of the code.

Lemma 3.1. *Given a code C of blocklength n, for any e, n there exists a Hamming ball of radius e containing at least $|C| \cdot \mathrm{Vol}(n, e)/2^n$ codewords.*

Proof. Consider the event of picking a Hamming ball B of radius e around a random center. For each $c \in C$, let X_c be an indicator variable equal to 1 if $c \in B$ and 0 otherwise. Then for all c, $\mathbf{E}(X_c) = \Pr(X_c = 1) = \mathrm{Vol}(n, e)/2^n$. The total number of codewords in B is equal to $\sum X_c$, so by linearity of expectation, $\mathbf{E}(\#\text{ codewords in } B) = \sum \mathbf{E}(X_c) = |C| \mathrm{Vol}(n, e)/2^n$. As there must exist at least one ball achieving the expectation, the lemma is proved. □

Theorem 3.2 (The Elias-Bassalygo Bound)**.**

$$R \leq H\left((1 - \sqrt{1 - 2\delta})/2\right)$$

Proof. Set $e = J(\delta)n - 1$. Then by the above lemma there is a ball of radius e containing $|C| \mathrm{Vol}(n, e)/2^n$ codewords. By the Johnson bound, no such ball can contain more than $2nd$ codewords, hence $|C| \mathrm{Vol}(n, e)/2^n < 2nd$. Rearranging and using the entropy approximation for the volume of a Hamming ball, we have that

$$|C| \leq 2^{1 - H(J(\delta))n} \cdot 2^{o(n)}.$$

By taking $n \to \infty$, we get the claimed upper bound on the rate. □

As shown in the graph at the beginning of this lecture, this bound is a strict improvement to both the Hamming bound and the binary Plotkin bound.

4 Revisiting the Gilbert-Varshamov Bound: Linear Codes

The Elias-Bassalygo bound has been slightly improved, but there has been no asymptotic improvement for binary codes to the GV bound. Hence it is still open whether the GV bound of $R = 1 - H(\delta)$ the best asymptotic rate that can be achieved. One way to address this is to study what sort of codes meet the GV bound. The version of the bound that we saw constructed a general code, so it is natural to ask if there is a *linear* code that meets the GV bound. We show below that there is. The theorem below is due to Varshamov, who proved it independently of the theorem of Gilbert that achieved a similar bound for general, nonlinear codes. The two results are usually cited together, giving us the Gilbert-Varshamov bound (or Varshamov-Gilbert bound, depending on if you learned it in Cyrillic or not).

Theorem 4.1. *For every $0 < \delta < 1/2$, and all large eough n, there exists an $[n, k, \geq \delta n]_2$ binary linear code with with $k \geq n(1 - H(\delta)) - 1$.*

Proof. Let $k = \lfloor n(1 - H(\delta)) \rfloor - 1$, so that $k \geq n(1 - H(\delta)) - 2$. Pick a random linear code by forming $(n - k) \times n$ Boolean parity-check matrix H with each entry chosen independently at random. The probability that the resulting code does not have distance at least $d = \delta n$ is equal to the probability that there exists a point x with weight $\leq d - 1$ and $Hx = 0$. Let us call this event $\mathcal{E}$. Then by the union bound, we have that

$$\Pr(\mathcal{E}) \leq \sum_{\substack{\mathrm{wt}(x) \leq d-1 \\ x \neq 0}} \Pr(Hx = 0).$$

Consider computing the product Hx one row at a time. For every $x \neq 0$, for a uniformly random binary vector r, $\Pr(\langle x, r\rangle = 0) = 1/2$, as fixing all random choices except one whose coefficient in x is one, we have that the dot product is 1 with probability $1/2$. Note this use critically the constraint that $x \neq 0$. Now as the rows of H are chosen independently, for any fixed x, $\Pr(Hx = 0) = (1/2)^{n-k}$. Hence

$$\begin{aligned}\sum_{\substack{\mathrm{wt}(x)\leq d-1\\ x\neq 0}} \Pr(Hx = 0) &\leq \sum_{\mathrm{wt}(x)\leq d-1} \left(\frac{1}{2}\right)^{n-k} \\ &\leq 2^{H(\delta)n-n+k} \leq 1/2 < 1\ .\end{aligned}$$

Therefore, there exists a choice of H which defines a code of relative distance at least δ. □

The above proof in fact shows that for $k = (1 - H(\delta) - \varepsilon)n$, all but an exponentially small fraction of parity check matrices yield codes of rate at least $1 - H(\delta) - \varepsilon$. Therefore codes that approach the GV bound are in abundance. As is unfortunately a common situation in combinatorics, despite this good performance of most codes, we do not know an explicit construction of a binary linear code that meets the GV bound. Settling this would be a major breakthrough in coding theory.

Note that we can construct a good meeting the GV bound with high probability simply by choosing a random parity matrix after setting n and k appropriately. The problem is that there is no known efficient algorithm to compute the minimum distance of a linear code—in fact, the problem is NP-hard—so there is now way to ascertain that the code picked in fact has distance meeting the GV bound. this fact does not have much practical impact.

Exercise: Give a deterministic algorithm to construct a binary linear code meeting the GV bound that runs in $2^{O(n)}$ time. The trivial brute-force algorithm that tries all parity-check matrices will take $2^{O(n^2)}$ time, so a faster algorithm, while still exponential, is interesting, and in fact will later be useful in finding "inner codes" for concatenation schemes.