6.440 Essential Coding Theory February 13, 2013

Lecture 3

Lecturer: Madhu Sudan *Scribe: Adam Hesterberg*

1 Today

1.1 Administrivia

Pset 1 out today, due on 2013-02-27 (in two weeks).

Sign up to scribe if you haven't yet.

1.2 Converse to Shannon's Coding Theorem

Theorem 1.1. $\forall p \in (0, \frac{1}{2})$ *(channel error probability),* $\forall \epsilon > 0$ *(amount above the channel capacity),* $\exists \delta > 0$ *(describing the exponential decay of the success rate w.r.t. message length),* $\exists n_0$, $\forall n \geq n_0$ *(message length),* $\forall E : \{0,1\}^k \to \{0,1\}^n$ *(encryption function),* $\forall D : \{0,1\}^n \to \{0,1\}^k$ *(decryption function), where* $k := \lfloor (1 - H(p) + \epsilon)n \rfloor$*, the probability over* $m \in \{0,1\}^k$ *and error* η *from a binary symmetric channel of error probability* p *that* $D(E(m) + \eta) = m$ *is at most* $2^{-\delta n}$*.*

Proof. We'll ignore the case where the number of errors in η is less than $(p - \frac{\epsilon}{2})n$ or more than $(p + \frac{\epsilon}{2})n$, since a Chernoff bound shows that that happens with probability at most $2^{-\delta_1 n}$. So the error rate is roughly p.

The probability over m and η that you decode correctly is

$$\sum_{x \in \{0,1\}^k, y \in \{0,1\}^n} Pr[x = m, y = E(m) + \eta, D(y) = x].$$

Without loss of generality, the encoding and decoding are deterministic functions. (If some probabilistic functions violated the theorem, then some hardcoding of random inputs would make a deterministic function violating the theorem.)

So, that probability is

$$\begin{aligned}\sum_{x \in \{0,1\}^k, y \in \{0,1\}^n} Pr[x = m, y = E(m) + \eta, D(y) = x] &= \sum_{y \in \{0,1\}^n} Pr[D(y) = m] Pr[y = E(m) + \eta | m = D(y)] \\ &= \sum_{y \in \{0,1\}^n} Pr[D(y) = m] Pr_\eta[y - E(D(y)) = \eta]\end{aligned}$$

Let $wf(w)$ be the number of 1s in $w \in \{0,1\}^n$. Then $Pr[\eta = w] \leq \frac{1}{\binom{n}{wf(w)}}$, since all errors with equal weight have equal probability and sum to at most 1. That's at most $\frac{1}{\binom{n}{(p - \frac{\epsilon}{2})n}}$, which is roughly $2^{-H(p)n}$ as $\epsilon \to 0$.

By assumption, $y - E(D(y))$ has weight between $(p - \frac{\epsilon}{2})n$ and $(p + \frac{\epsilon}{2})n$ (or the decryption was incorrect), and it's deterministic, so the probability that it's η is at most $Pr[\eta = w] \leq 2^{-H(p)n}$, so the chance you decode correctly (given that the weight of the errors is within $\frac{\epsilon}{2}$ of p) is at most

$$\sum_{y \in \{0,1\}^n} Pr[D(y) = m] 2^{-H(p)n} \leq 2^n 2^{-k} 2^{-H(p)n} \leq 2^{-\epsilon n},$$

as desired. □

Shannon's 1948 work mentioned all this stuff about existence of optimal codes (by choosing a random one), but mentioned Hamming's work as a potential way to do it practically. Shannon's has both error and message probabilistics; Hamming's has both of them worst-case.

1.3 Linear Codes: Some Existence Results

Reminder: $(n, k, d)_q$ is a not necessarily linear error correcting code, and $[n, k, d]_q$ is a linear one. q is the alphabet size—the alphabet is an arbitrary set Σ, but is assumed to be the finite field $\mathbb{F}_q$ if it exists, n is the length of the codewords (shorter is better) (the set of codewords C has $C \subseteq \Sigma^n$), k is the length of the message (longer is better; $n \geq k$) ($|C| = q^k$), and $d = \Delta(C) = \min_{x,y \in C, x \neq y}\{\Delta(x, y)\}$ is the minimum distance between codewords (larger is better).

For simplicity, we'll work instead with the (message) rate $R := \frac{k}{n}$ and the error rate $\delta := \frac{d}{n}$.

Theorem 1.2. *For every alphabet Σ (or $\mathbb{F}_q$), there's a code with $R \geq 1 - H_q(\delta)$, where H_q is the "q-ary entropy" $H_q(\delta) = -\delta \log_q(\delta) - (1 - \delta) \log_q(1 - \delta) + \delta \log_q(q - 1)$, which is maximized at $\delta = \frac{q-1}{q}$.*

The motivation for q-ary entropy is that the volume of a ball of radius δn in Σ^n is $q^{H_q(\delta n}$, which can be proven like the $q = 2$ version (take the log of $\binom{n}{\delta n}(q - 1)^{\delta n}$.)

Proof Techniques:

1. Random code:
   (a) Pick a random code with $2q^{Rn}$ codewords (twice as many as we want).
   (b) Throw out codewords that are too close (throwing out at most half, with high probability).
2. Greedy (Gilbert) code:
   (a) Pick a codeword among the remaining words.
   (b) Throw away every word at distance $d - 1$ from the new word.
   (c) Repeat.
3. Random linear code:
   (a) Pick random basis vectors $b_1, \ldots, b_k \in \mathbb{F}_q^n$.
   (b) Choose $C = span(b_1, \ldots, b_k)$.
4. Greedy parity check matrix (Varshamov): details later; basically, pick codewords greedily, making sure not to create codewords of weight less than d.

5. Wozencraft Ensemble of Codes (details if there's time)

Greedy code:

1. Initially, $C = \emptyset$, $S = \Sigma^n$.
2. While $S \neq \emptyset$:
3. Pick $w \in S$ arbitrarily.
4. Add w to C.
5. Remove the ball of radius $d-1$ around w from S.

After each iteration, $|C| \cdot |B(0, d-1)| + |S| \geq q^n$, since we remove at most $|B(0, d-1)|$ strings from S each time we add a codeword to C, so at the end $|C| \geq \frac{q^n}{|B(0,d-1)|}$, which is asymptotically $q^{n(1-H_q(\delta))}$. So the worst maximal code is at least that good, that is, $k = \log_q(|C|) \geq n(1 - H_q(\delta))$, as claimed.

For sufficiently large values of q (at least, say, 48 or 49), there exist codes which beat that. We don't have any significantly better codes for $q \in \{2, 3\}$, though. We can do slightly better as follows:

Consider the graph $G_{n,d,q}$ on vertices Σ^n with edges between pairs of vertices at distance less than d. Then an error-correcting code is precisely an independent set. Then the previous proof said precisely that there's an independent set of size at least $\frac{|\Sigma^n|}{1+\deg(G)}$ (which might be a theorem of Turán), and that can be tight, for instance, for a graph that's a disjoint union of cliques of size $\deg(G) + 1$.

In general, a graph G has at most $|V(G)|\Delta(G)^2$ triangles, since each vertex is in at most $\Delta(G)^2$ triangles. Ajtai, Komlos, and Szemerédi proved that in random graphs, there's an independent set of size $\log(\Delta(G))\frac{|V(G)|}{\Delta(G)}$. All they needed was that the number of triangles was at most $|V(G)|\Delta(G)^{2-\epsilon}$ (as is true for a random graph), and that happens to be true for our graph $G(n, d, q)$, so there is a code with $H_q(\delta)nq^{n(1-H_q(\delta))}$ codewords, which is an asymptotically trivial improvement.

If we fix $\delta > 0$ and let $q \to \infty$, then $H_q(\delta) = \delta + O(\frac{1}{\log(q)})$. But that's not the best possible, because there are "algebraic geometry" (AG) codes where $R \geq 1 - \delta - \frac{1}{\sqrt{q}-1}$, so for large enough q we can do better.

Greedy Parity Check Code:

Let $m := n - k$. We'll greedily pick an $n \times m$ matrix one row at a time, and choose the code $C = \{x \in \mathbb{F}_q^n : x \cdot H = 0\}$. The code tolerates d errors iff every subset of $d-1$ rows of the matrix is linearly independent. So, never include a row that's a linear combination of up to $d-2$ of the existing rows. If we have l rows already, there are $(q-1)^{d-2}\binom{l}{d-2}$ such linear combinations, which is exactly the volume of a ball of radius $d-2$ in $\mathbb{F}_q^n$, so we can get a code with $\frac{|\mathbb{F}_q^n|}{|B(q,n,d-2)|}$ codewords. That's slightly better than the Gilbert bound, which has $\frac{|\mathbb{F}_q^n|}{|B(q,n,d-1)|}$ codewords.