6.897 Algorithmic Introduction to Coding Theory September 11, 2002

Lecture 3

Lecturer: Madhu Sudan *Scribe: Emanuele Viola*

1 Introduction

The plan for today is:

- Final discussion about Shannon's theory.
- High level view and constrast of Shannon's theory with Hamming's theory.
- Linear Codes.

2 Shannon Theory

Last time we mentioned the *converse coding theorem*:

Theorem 1 *Let k and n satisfy the relation*

$$\frac{k}{n} > 1 - H_2(p) + \epsilon.$$

Then

$$\lim_{k,n\to\infty} \Pr_{\eta}[D(E(x)+\eta) = x] = 0$$

for any pair of functions $E : \{0,1\}^k \to \{0,1\}^n$, $D : \{0,1\}^n \to \{0,1\}^k$. Where the probability is taken over vectors η such that each bit is 1 with probability p and 0 otherwise, i.e. $\eta \leftarrow Binom(n,p)$.

We now give a full proof of this result.

Proof For every $x \in \{0,1\}^k$ define

$$S_x := \{y \in \{0,1\}^n : D(y) = x\}.$$

We note that $\cup_x S_x = \{0,1\}^n$ and $\sum_x |S_x| = 2^n$.

For a binary string η, let $wt(\eta)$ be its *weight*, that is the number of bits which are 1.

We say that η is *easy* if $wt(\eta) \leq (p-\epsilon)n$. We say that η is *hard* if it is not easy. Notice that $\Pr[\eta \text{ is easy}] \leq exp(-n)$.

Now consider any vector $b \in \{0,1\}^n$ such that $wt(b) \geq (p-\epsilon)n$. Then

$$\Pr_{\eta}[\eta = b] = \frac{1}{\binom{n}{wt(b)}} \approx 2^{-(H(p)+o(1))n}.$$

Now, we have:

$$\begin{aligned}
\Pr[\text{deconding correctly}] &\leq exp(-n) + \Pr[\text{decoding correctly}|\eta \text{ is hard}] \\
&= exp(-n) + 2^{-k} \sum_x \sum_{y \in S_x} \Pr_\eta[y = E(x) + \eta | \eta \text{ is hard}] \\
&= exp(-n) + 2^{-k} \sum_x \sum_{y \in S_x} 2^{-(H(p)+o(1))n} \\
&= exp(-n) + 2^{-(H(p)+o(1))n-k+n}.
\end{aligned}$$

The result follows.

■

3 High Level View and Hamming vs Shannon

In the *noiseless* theorem (from the previous lecture) we have considered a special source which gives a 1 with probability p and 0 otherwise.

In the *noisy* theorem we have considered a simple source of unbiased bits. The noise model was a channel that would flip a bit with probability p.

In the Shannon model, the source is represented by a directed graph. The nodes are labeled with 0 or 1, and the edges are given a weight. The output bits are obtained through a random walk on the graph.

Shannon gave a closed formula of the rate at which such a source is producing information.

We can think of a channel as a stochastic process mapping an input alphabet Σ to an output alphabet Γ. Shannon defined the *capacity* of a graph, and proved that reliable transmission is possible if and only if the rate of the source is less than the capacity of the channel.

However, Shannon's results are completely non constructive.

We now compare Hamming Theory with Shannon Theory

- Objects:

  Hamming: codes, minimum distance.

  Shannon: encoding and decoding functions.

- Goal:

  Hamming: maximizing the rate and the minimum distance.

  Shannon: Maximizing the rate and minimizing the probability of wrong decoding.

- Error model:

  Hamming: Adversarial (worst case).

  Shannon: Random.

We will focus on Hamming Theory.

We describe a few more results which can be found in Hamming's original paper.

Hamming gives constructions of codes with minimum distance $d = 1, 2, 3, 4$. The construction for $d = 1$ is trivial, while the construction for $d = 3$ was given in the last lecture. To go from a code C with minimum distance $2t-1$ to one with minimum distance $2t$, Hamming noticed that we can just append the parity check bits to the elements of C. Indeed, if two elements of C differed in $\leq 2t$ position then clearly the parity bits will not decrease their distance. While if they differed in exactly $2t-1$ then they differered in an *odd* number of positions, and therefore their parity check bit will be different, giving distance $2t$.

Hamming also gives an upper bound on the rate of a code C with minimum distance $d = 2t + 1$. Such a code is a t-error correcting code, and therefore for every two codewords, the corresponding Hamming balls of radius t are disjoint. Since the union of these Hamming balls is contained in $\{0, 1\}^n$ we get

$$|C| \sum_{i \leq t} \binom{n}{i} \leq 2^n.$$

Where we use the fact that a Hamming ball in $\{0, 1\}^n$ of radius t has size $\sum_{i \leq t} \binom{n}{i}$.

To decode a Hamming code (of minimum distance 3), we note that multiplying the received codeword by the parity check matrix H associated to the code will give 0 if no error occurred, while if 1 error occurred it will give the binary representation of the index of the bit where this error occurred. This is true because suppose we receive the codeword c with an error e_i (where e_i is the 0/1 vector which is 1 only in the i-th coordinate). Then

$$(c + e_i)H = cH + e_iH = e_iH.$$

Now note that e_iH is simply the i-th row of H which by construction is the binary representation of i.

4 Linear Codes

In this section we define linear codes.

If the alphabet Σ is a finite field [1], then we say that a code C is linear if it is a linear subspace of Σ^n. That is, $x, y \in C$ implies $x + y \in C$ and $x \in C, a \in \Sigma$ implies $ax \in C$.

Notationally, we represent linear codes with square brackets: $[n, k, d]_q$. All the codes we will see in this class are linear.

Linear codes are interesting for many reasons. For example, the encoding function is simple, just matrix multiplication. It is also easy to detect errors: since the code is linear there is a parity check matrix H such that $C = \{y : Hy = 0\}$, and therefore we can detect errors again by simple matrix multiplication.

Another interesting feature of a linear code is its *dual.* Let C be a linear code generated by the matrix G. C has a parity check matrix H. We define the dual code of C as the code

[1] See the on line lecture notes on Algebra for background on fields.

generated by H^T, the transpose of the matrix H. It can be shown that the dual of the dual of C is C itself.

For example, consider the Hamming code with block length $n = 2^l - 1$. Its dual is the code generated by a $l \times n$ matrix whose columns are all the non zero binary strings of length l. It is easy to see that the encoding of a message b is

$$< b, x >_{x \in \{0,1\}^l - 0},$$

where $< .,. >$ denotes inner product modulo 2. In other words, the encoding of b is the parity check of all the non-empty subsets of the bits of b.

It can be shown that if $b \neq 0$ then $< b, x >= 1$ for at least 2^{l-1} of the x's in $\{0,1\}^l - 0$. This implies that the dual code is a $[2^l - 1, l, 2^{l-1}]_2$ code. It can be shown that this is the best possible, in the sense that there is no $(2^l - 1, l + \epsilon, 2^{l-1})_2$ code. This code is called the *simplex* code or the *Hadamard* code. The second name comes from the french mathematician Jacques Hadamard who studied the $n \times n$ matrices M such that $MM^T = nI$, where I is the $n \times n$ identity matrix. It can be shown that if we form a matrix with the codewords from the previous code, we replace 0 with -1, and we pad the last bit with 0, then we obtain a Hadamard matrix.