6.897 Algorithmic Introduction to Coding Theory	September 10, 2001

Lecture 2

Lecturer: Madhu Sudan *Scribe: Omprakash Gnawali*

Today we will cover the following topics:

- Converse to Shannon's coding theorem.
- Some remarks on Shannon's coding theorem.
- Error correcting codes.
- Linear codes.

1 Converse to Shannon's coding theorem

Recall that the Binary Symmetric Channel (BSC) with parameter p is the channel that transmits bits, flipping each transmitted bit with probability p independent of all other events. Lets start by recalling Shannon's coding theorem informally.

> Over the BSC with parameter p, it is possible to transmit information at any rate less than $1 - H(p)$.

(To give a sense of how to make the above formal, here is the formal version in all its quantified glory: "For every $p < \frac{1}{2}$ and $\epsilon, \delta > 0$, there exists an $n_0 < \infty$ such that for every $n \geq n_0$ and $k \leq (1 - H(p) - \epsilon)n$, there exist functions $E : \{0,1\}^k \to \{0,1\}^n$ and $D : \{0,1\}^n \to \{0,1\}^k$ such that

$$\Pr_{\eta \leftarrow D_{p,n}, m \leftarrow U_k}[D(E(m) + \eta) = m] \geq 1 - \delta,$$

where U_k is the uniform distribution on $\{0,1\}^k$ and $D_{p,n}$ is the distribution on n bits chosen independently with each bit being 1 with probability p.")

We will now proof a converse to this theorem.

Theorem 1 *For every $p < \frac{1}{2}$ and $\epsilon, \delta > 0$, there exists an $n_0 < \infty$ such that for every $n \geq n_0$, $k \geq (1 - H(p) + \epsilon)n$, and functions $E : \{0,1\}^k \to \{0,1\}^n$ and $D : \{0,1\}^n \to \{0,1\}^k$, it is the case that*

$$\Pr_{\eta \leftarrow D_{p,n}, m \leftarrow U_k}[D(E(m) + \eta) \neq m] \geq 1 - \delta,$$

where U_k is the uniform distribution on $\{0,1\}^k$ and $D_{p,n}$ is the distribution on n bits chosen independently with each bit being 1 with probability p.

Ignoring all quantifiers above, the essense is that if we are trying to send information at a rate of $1 - H(p) + \epsilon$, then the decoding is erroneous with probability almost 1, *no matter* which encoding and decoding function we use.

Proof The hard part of this proof is deciding how to deal with the encoding and decoding functions which are completely arbitrary! Turns out, we will ignore the encoding function entirely, and ignore the decoding function almost entirely! The main focus is on the error, and the fact that the error distributes the transmitted word over a large space of possibilities (and so any decoding function should be helpless). Specifically, we note the following:

- The number of errors is unlikely to be too small or too large: Formally, for every m and E,

$$\Pr_{\eta}\left[E(m) + \eta \in B(E(m), (p-\epsilon)n)\right] = \Pr_{\eta}\left[\eta \in B(\vec{0}, (p-\epsilon)n)\right] \leq e^{-\epsilon^2 n}. \tag{1}$$

  The equality above is obtained by simply translating the center of the ball from $E(m)$ to the origin $\vec{0}$, the string consisting of all 0's. The inequality above is a straightforward application of Chernoff bounds - however, this time we are using the fact that it proves that a random variable is not likely to take on a value much less than its expectation. Similarly, we get that

$$\Pr_{\eta}\left[E(m) + \eta \notin B(E(m), (p+\epsilon)n)\right] = \Pr_{\eta}\left[\eta \in B(\vec{0}, (p-\epsilon)n)\right] \leq e^{-\epsilon^2 n}.$$

- Given that the error is large (but not too large), no single point in the space has a high probability of being the received word. Specifically, for every m, E and $y \in B(E(m), (p+\epsilon)n) - B(E(m), (p-\epsilon)n)$,

$$\Pr_{\eta}\left[E(m) + \eta = y\right] \leq \frac{n}{2^{H(p-\epsilon)n}}. \tag{2}$$

  To see why the above is true, let $R = \Delta(y, E(m))$ be the Hamming distance between y and $E(m)$. Note we have $R \in [(p-\epsilon)n, (p+\epsilon)n]$. Let N_R be the number of binary vectors with R ones and $n - R$ zeroes. Since all error patterns η with the same number of errors are equally likely, we note we the probability of having any fixed vector containing exactly R ones as the error vector is at most $\frac{1}{N_R}$. Thus to upper bound the probability of the event in question it suffices to lower bound N_R for $R \in [(p-\epsilon)n, (p+\epsilon)n]$. Using the fact that $N_R = \mathrm{Vol}(R, n) - \mathrm{Vol}(R-1, n)$, and the fact that N_R is increasing for R in the range $[0, \frac{n}{2}]$, we get than $N_R \geq \frac{\mathrm{Vol}(R,n)}{n} \geq \frac{\mathrm{Vol}((p-\epsilon)n,n)}{n}$. Now using the fact that $\mathrm{Vol}((p-\epsilon)n, n) \approx 2^{-H(p-\epsilon)n}$, we get that the probability $\eta = y - E(m)$ is at most $\frac{n}{2^{H(p-\epsilon)n}}$.

To continue the proof, we finally look at the decoding function (though even this look will be very superficial). Let $K = 2^k$ and let $\{m_1, \ldots, m_K\}$ denote the K possible messages. Let $S_i = \{y | D(y) = m_i\}$ be the set of received words that are decoded to the ith message. The only property we use about the decoding is that $\sum_{i=1}^{K} |S_i| = 2^n$, i.e., the decoding is a *function*! (On the other hand, there is little else to use!) We now use the observations in the previous paragraph to prove that the decoding function is not very likely to succeed.

Let ρ be the probability of decoding successfully. In order for the decoding to succeed, we must pick some message m_i to encode and transmit, and the error vector η must be such that the received vector $y = E(m_i) + \eta$ must lie in S_i. This gives:

$$\rho = \sum_{i=1}^{K} \sum_{y \in S_i} \Pr_{m,\eta}\left[m = m_i \text{ and } \eta = y - E(m_i)\right] = \Pr_{m}[m = m_i] \Pr_{\eta}[\eta = y - E(m_i)],$$

where the second equality follows from the fact that the events considered are independent. Fixed m_i and let us bound the inner summation above. The probability that m equals m_i is exactly $1/K = 2^{-k}$. The event that $\eta = y - E(m_i)$ is independent of m and so we can estimate this quantity separately. Fix m_i. Let $U = B(E(m_i), (p-\epsilon)n)$ and $V = \{0,1\}^n - B(E(m_i), (p+\epsilon)n)$ (i.e., U is the points too close to $E(m)$ and V the points too far from $E(m)$. Then

$$\begin{aligned}
&\sum_{y \in S_i} \Pr_\eta[\eta = y - E(m_i)] \\
&\quad\leq \sum_{y \in U} \Pr_\eta[\eta = y - E(m_i)] + \sum_{y \in V} \Pr_\eta[\eta = y - E(m_i)] + \sum_{y \in S_i - U - V} \Pr_\eta[\eta = y - E(m_i)] \\
&\quad\leq 2e^{-\epsilon^2 n} + |S_i| \frac{n}{2^{H(p-\epsilon)n}},
\end{aligned}$$

where the second inequality above follows from Equations (1) and (2). Combining the above, we have:

$$\begin{aligned}
\rho &\leq \sum_{i=1}^{K} 2^{-k} \left(2e^{-\epsilon^2 n} + \frac{n|S_i|}{2^{H(p-\epsilon)n}} \right) \\
&= 2e^{-\epsilon^2 n} + \frac{n2^{-k}}{2^{H(p-\epsilon)n}} \left(\sum_{i=1}^{K} |S_i| \right) \\
&= 2e^{-\epsilon^2 n} + n2^{-k-H(p-\epsilon)n+n}
\end{aligned}$$

The theorem follows from the fact that for every $\epsilon, \delta > 0$ we can pick n_0 large enough so that for every $n \geq n_0$, it is the case that $2e^{-\epsilon^2 n} \leq \delta/2$ and $n2^{-k-H(p-\epsilon)n+n} \leq \delta/2$ (assuming $k \geq (1 - H(p) + \epsilon)n$). ■

Recall that at the end of the previous lecture we showed that if we assumed the noiseless coding theorem is tight (i.e, has a converse) then the converse to the noisy coding theorem follows. While the theorem above does not imply a converse to the noiseless coding theorem, the proof technique is general enough to capture the noiseless coding theorem as well. This motivates the following exercise.

Exercise: Prove converse of the noiseless coding theorem (from Lecture 1).

2 Remarks on Shannon's Theorem

2.1 Discrete Memoryless Channels

Shannon's coding theorem is, of course, much more general that what we have presented. We only presented the result for the case of the Binary Symmetric Channel. For starters, the result can be generalized to the case of all "Discrete Memoryless Channels (DMCs)". Such channels are characterized by two finite sets — Σ representing the input alphabet of the channel and Γ representing the output alphabet of the channel — and a transition probability matrix $P = \{p_{\sigma\gamma}\}$, where $p_{\sigma\gamma}$ denotes that probability that the output alphabet is γ given than the input alphabet is σ. We require that $\sum_{\gamma \in \Gamma} p_{\sigma\gamma} = 1$ for every $\sigma \in \Sigma$, so that this definition makes sense. When we attempt to transmit a sequence of symbols from Σ over this channel, it behaves on each element of the channel independently and produces a sequence of elements from Γ, according to the transition probability matrix P.

Given such a channel, characterized by P, Shannon gave a procedure to compute the capacity of the channel. This capacity relates to the mutual information between two random variables.

Given a distribution D_Σ over Σ, let σ be a random variable chosen according to D_Σ. Pick γ at random from Γ with probability $p_{\sigma\gamma}$. Denote by $D_{\Sigma,\Gamma}$ the joint distribution on the pairs (σ, γ) so generated, and by D_Γ the marginal distribution of γ. Since D_Σ, D_Γ and $D_{\Sigma,\Gamma}$ are all distributions on finite sets, their entropy is well defined. Define the "mutual information" between variables σ and γ (or more correctly of the transition matrix P with initial distribution D_Σ) to be $H(D_\Sigma) + H(D_\Gamma) - H(D_{\Sigma,\Gamma})$.

Shannon's theorem showed that the capacity of the channel characterized by P, is the maximum over all distributions D_Σ, of the mutual information between σ and γ. He also gave a linear system whose solution gave the distribution that maximizes this information.

2.2 Markovian Channels

Shannon's theory extends even further. Natural scenarios of error may actually flips bits with some correlation rather than doing so independently. A subclass of such correlations is given by "Markovian channels", where the channel can be in one of several (finitely many) states. Depending on which state the channel is in, the probability with which it makes errors may be different.

Such models are useful in capturing, say, "burst error" scenarios. In this situation the channel makes sporadic sequences of many errors. One can model this source by a channel with two states (noisy/normal) with two different channel characteristics for the two states. (see Figure 1). When in the "noisy" state the channel flips every bit, say, with probability $\frac{1}{2}$ (and so a channel that is perpetually in a noisy state can transmit no information). When in the normal state, however, the channel flips bits with only a small probability, say p. Further, if the channel is in a given state at time t it tends to stay in the same state at time $t+1$ with probability $1-q$ and tends to flip its state with a small probability q. Is it possible to transmit information on such a channel? If so, at what rate? Working this out would be a good exercise!

Shannon's theory actually gives the capacity of such a channel as well. Deciphering what the theory says and unravelling the proofs would be a good topic for a term paper.

2.3 Zero Error Capacity of a channel

An interesting by product of the Shannon theory is the so called "Zero error capacity" of a channel. To motivate this notion, let us consider a "stuck typewriter". Suppose the keys in the typewriter are sticky and likely to produce the wrong symbol when you hit it. Further suppose that the error pattern is very simple. If you hit a letter of the keyboard, you get as output either the correct letter or the next letter of the alphabet. and that such an error happens with probability $\frac{1}{2}$ for every keystroke independently. In other words, typing A results in A or B, typing B results in B or C, etc. and typing Z results in Z or A (so we have a wrap around). (See Figure 2.)

Some analysis of this channel reveals that it has a channel capacity of $\log_2 13$. I.e., each keystroke is capable, on the average of conveying one of 13 possibilities. If the typewriter had not been stuck, it would have had a capacity of $\log_2 26$. Thus the error cuts down the number of possibilities per stroke by a factor of 2.

If we think hard (actually may be not even so hard) we can see a way of achieving this capacity. Simply don't use the even-numbered letters of the alphabet (B,D,F, etc.) and work only with the

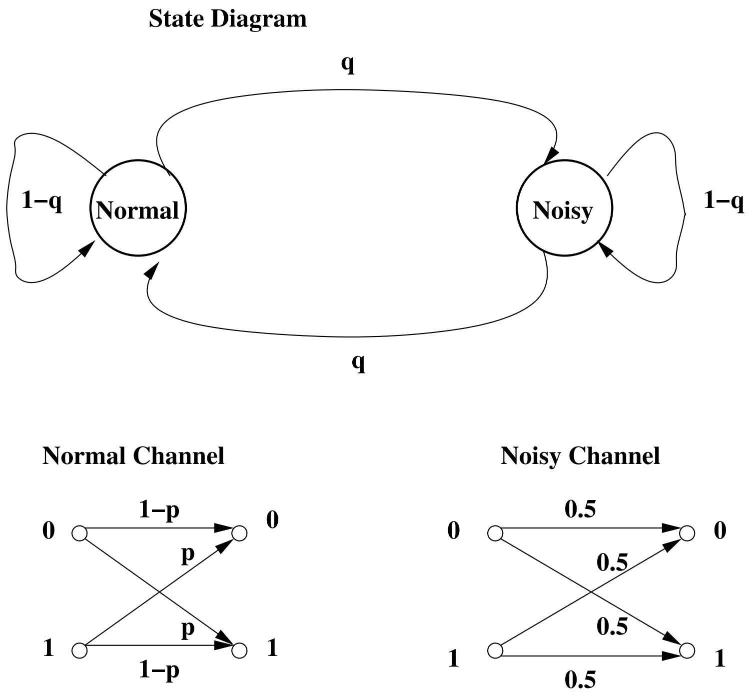

Figure 1: State diagram of a simple burst error channel

odd-numbered ones (A, C, E etc.). Since there is no possibility of confusion within the odd-numbered letters, there is no ambiguity in the message (if an A or B is received, A must have been the keystroke typed, if C or D is received, C is the keystroke etc.). The interesting aspect of this way of using the channel is that we acheive the capacity, *with zero error*! This motivates a general concept: The *Zero Error Capacity* of a channel is informally defined as the optimal rate of transmission that can be achieved while maintaining a zero probability of decoding error.

In the case of the stuck typewriter the zero error capacity equals the Shannon capacity. This is not always the case. For example, the zero error capacity of the binary symmetric channel for any $p \neq 0, 1$ is zero! (Any string has positive probability of being corrupted into to any other string.) It is true that the zero error capacity is less than or equal to the Shannon capacity. Computing the Shannon capacity of even simple channels is non-trivial, and in general this function is not known to be computable.

A simple illustrative example is the zero error capacity of a stuck keyboard with only five keys (or in general an odd number of keys)! Then it is no longer clear what the zero error capacity of this channel is. In 1979, L. Lovász [3] wrote a brilliant paper that showed how to compute the Shannon capacity of several graphs (and in general give a lower bound on the Shannon capacity). In particular he shows that the Shannon capacity of a stuck typewriter with 5 keys is $\sqrt{5}$ - an irrational number! This paper has played a pioneering role in computer science leading to the notion of semi-definite programming and its consequences on combinatorial optimization.

Term paper topic: Study the zero error capacity of arbitrary channels and survey the work of Lovász and successors.

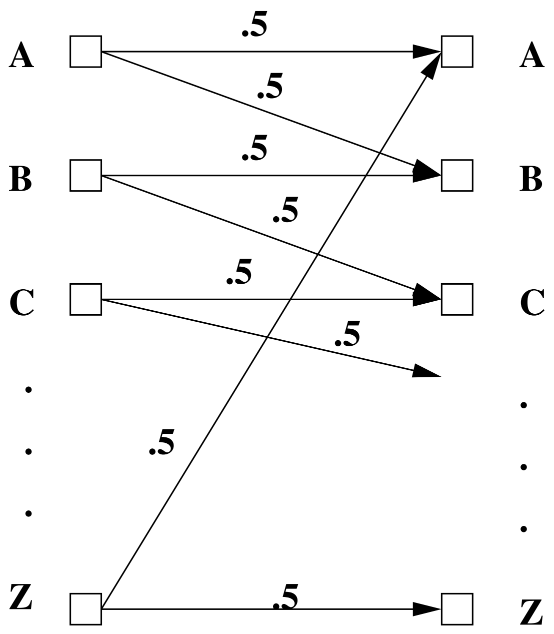

Figure 2: Channel for a stuck typewriter

This will terminate our discussion of the Shannon based theory. As mentioned in the first lecture, Shannon's original paper [4] and the text by Cover and Thomas [1] are excellent sources for further reading.

3 Error correcting codes

Shannon's theory, while providing exact results for the rate at which one can communicate on a noisy channel, are unfortunately highly non-constructive. Specifically the two key ingredients: the *encoding* and *decoding* functions are totally non-constructive. In order to get some sense of how to make these results constructive, one has to examine the encoding function and see what properties about it are useful. Shannon noticed that a result of Hamming indeed does so, and that this may be a step in making his results more constructive. In retrospect it seems this was crucial in making Shannon's results constructive. This is the theory of error-correcting codes, as initiated by the work of Hamming [2]. Hamming focussed on the set of strings in the image of the encoding map, and called them "error-correcting codes". He identified the distance property that would be desirable among the codes and initiated a systematic study. We develop some notation to study these notions.

3.1 Notation

We consider codes over some alphabet Σ and reserve the letter q to denote the cardinality of Σ. It is often helpful to think of $\Sigma = \{0, 1\}$ and then the codes are termed binary codes.

We consider transmissions of sequences of n symbols from Σ from sender to receiver. Recall that for two strings $x, y \in \Sigma^n$, the Hamming distance between x and y, denoted $\Delta(x, y)$, is the number of coordinates where x differs from y. We note that the Hamming distance is indeed a metric: i.e., $\Delta(x, z) = \Delta(z, x) \leq \Delta(x, y) + \Delta(y, z)$ and $\Delta(x, y) = 0$ if and only if $x = y$.

A code C is simply a subset of Σ^n for some positive integer n. The minimum distance of a code C, denoted $\Delta(C)$, is given by $\Delta(C) = \min_{x,y \in C, x \neq y}\{\Delta(x, y)\}$. The Hamming theory focusses on the task of constructing (or showing the existence of) codes with large minimum distance and large cardinality.

There are four fundamental parameters associated with a code C:

- Its *block length*: n, where $C \subseteq \Sigma^n$.
- Its *message length*: $k = \log_q |C|$. (To make sense of this parameter, recall that we are thinking of the code as the image of an encoding map $E : \Sigma^k \to \Sigma^n$ and in this case $\log_q |C| = k$ is the length of the messages.)
- Its *minimum distance* $d = \Delta(C)$.
- Its *alphabet size*: $q = |\Sigma|$.

It is often customary to characterize a code by just the four parameters it achieves and refer to such a code as an $(n, k, d)_q$-code.

3.2 Broad Goals of Coding Theory

In a nutshell the broad goal of coding theory can be stated in one of the four ways below, where we fix three of the four parameters and try to optimize the fourth. The correct optimizations are:

- Given k, d, q find an $(n, k, d)_q$ code that minimizes n.
- Given n, d, q find an $(n, k, d)_q$ code that maximizes k.
- Given n, k, q find an $(n, k, d)_q$ code that maximizes d.
- Given n, k, d find an $(n, k, d)_q$ code that minimizes q.

The first three choices are self-explanatory. It is always desirable to have a small block length, large message length, and large distance. However it is not so immediate that minimizing q is the right thing to do (in particular, we don't have a monotonicity result). However empirically (and almost certainly) it seems to be the case that one can get get values of the parameters for larger values of q and getting good parameters for small values of q is the challenging part. Furthermore, building codes with large q and then trying to reduce q is a very clever way of getting good codes. So we will keep this version in mind explicitly.

3.3 Error Correcting Codes

Why are we interested in codes of large minimum distance? It may be worth our while to revisit Hamming's paper and see what he had to say about this. Hamming actually defined three related properties of a code C.

1. The minimum distance of C, which we have already seen.
2. The error detection capacity of C: A code C is e-error detecting if under the promise that no more than e errors occur during transmission, it is always possible to detect whether errors have occured or not, and e is the largest integer with this property. Hamming notes that a e-error correcting code has minimum distance $e + 1$.
3. The error correction capacity of C: A code C is t-error correcting if under the promise that no more than t errors occur during transmission, it is always possible to determine which locations are in error (information-theoretically, but not necessarily efficiently) and correct them, and if t is the largest integer with this property. Hamming notes that a t-error correcting code has minimum distance $2t + 1$ or $2t + 2$.

Thus the minimum distance of a code is directly relevant to the task of correcting errors and we will focus on this parameter for now. Later (in the second part of the course) we will turn our attention to the question - how can we make these error-detection and error-correction capabilities algorithmic.

3.4 Some simple codes

The famed Hamming codes are codes of minimum distance three. Even though Hamming's name is associated with distance-3 codes, he also gave, in the same paper. codes of distance two and four! Let us start even slower and describe the distance one codes first!

- $d = 1$: This is trivial. All we want is that the encoding function be injective. So the identity function works and gives the best possible $(n, n, 1)_q$ code.
- $d = 2$, This is already interesting. A simple way to achieve distance 2 is to append the parity of all the message bits to message and thus get a code with $n = k + 1$, i.e., an $(n, n-1, 2)_2$ code. For general q, one identifies Σ with $\mathbb{Z}_q$, the additive group of integers modulo q and uses (instead of the parity check bit) the check symbol that is the sum of all message symbols over $\mathbb{Z}_q$. This gives, for every q, a $(n, n-1, 2)_q$ code.
- $d = 3$, non-trivial interesting case.

Interpolating from the first two examples, one may conjecture that a $(n, n-d+1, d)_q$ code is always possible. This turns out not to be the case and in fact $d = 3$ already gives a counterexample to this conjecture. Hamming gave codes for this case and proved their optimality. We will describe the codes in the next lecture.

References

[1] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* Wiley Publishing, New York, 1991.

[2] Richard W. Hamming. Error Detecting and Error Correcting Codes. *Bell System Technical Journal*, 29:147–160, April 1950.

[3] László Lovász. On the Shannon capacity of a graph. *IEEE Transactions on Information Theory*, 25:1–7, January 1979.

[4] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.