CS 229r Essential Coding Theory Jan 26, 2017

Lecture 2

Instructor: Madhu Sudan *Scribes: Oxana Poburinnaya*

This lecture covers a paper by Shannon [Sha48] from 1948. Shannon studied the possibility of efficient transmission of information over a noisy channel. For instance, can we communicate reliably, if each bit of the message is flipped with 10% probability? What about 50%? 49%? What rate can we achieve in this setting?

1 Compression and error-correcting.

Besides error-correcting, Shannon was also concerned about *compressing* the message. For instance, if we need to send a stream of pictures which are very similar (e.g. pictures of the same part of the sky from a satellite), it makes sense to send the picture only once, and then transmit *changes* rather than the whole picture. Thus, Shannon modeled the process as follows:

1. The message m is processed by an encoder, which compressed it and adds redundancy for error-correcting;
2. The resulting codeword x is sent over a noisy channel, resulting in a possibly different $\hat{x}$;
3. The receiver applies the decoding procedure (which decompresses the message and corrects errors) and obtains some $\hat{m}$; the hope is to design encoding and decoding such that $m = \hat{m}$ almost always.

Given that we often compress messages before sending them, why does it make sense to design stand-alone error-correcting codes? Maybe if we design a code which compresses and does error-correcting at the same time, we can achieve more? For instance, error-correction could possibly use the knowledge of a compression procedure to be able to correct more errors. It turns out that such knowledge doesn't give us anything; therefore, it is reasonable to split the encoding (resp., decoding) into compression and encoding of ECC (resp, decoding of ECC and decompression). This can be modeled as follows:

1. Original message M is given to compressor to produce a shorter m;
2. m is given to encoding algorithm of ECC to produce a codeword x;
3. x is sent over a noisy channel, resulting in a possibly different $\hat{x}$;
4. $\hat{x}$ is given to decoding algorithm of ECC which outputs $\hat{m}$;
5. $\hat{m}$ is given to decompression algorithm which outputs $\hat{M}$. Again, the hope is that $M = \hat{M}$ almost always.

Here the compression/decompression procedure doesn't know anything about error-correcting; from its point of view, M is compressed, sent over a *noiseless* channel, and then decompressed. Essentially, ECC allows to emulate a noiseless channel.

2 Modeling noisy channels

In the previous lecture we saw one way to model noise in a channel: we assumed that no more than t errors happen per codeword. Shannon instead considered a model where each bit of the codeword can be modified independently of other bits. We describe several examples:

Binary Symmetric Channel (BSC). Each bit is flipped with probability $p \in (0, \frac{1}{2})$. Denoted as BSC_p.

Binary Erasure Channel. Each bit is erased (i.e. replaced with a special symbol "?") with probability $p \in (0, \frac{1}{2})$. This model is more benign, since positions of errors are known.

General case. Assume the codeword is a word in alphabet Σ, and the channel transforms each symbol from Σ to some other symbol (in a possibly different alphabet Γ). To describe such a channel, it is enough to define a matrix P with dimensions $|\Sigma| \times |\Gamma|$, where p_{ij} is the probability that i-th symbol in Σ transforms into j-th symbol in Γ (for a matrix to represent a noisy channel, it should be the case that $\Sigma_j p_{ij} = 1$ for all i).

Note that a noisy channel can be viewed as a function which takes codewords as inputs and outputs words of a possibly different alphabet.

3 Shannon's coding theorem

Shannon's theorem answer the following question: when is it possible to communicate reliably over a BSC$_p$, and how high the rate could be? Intuitively, when probability of error p is fairly small (say, .001), communication should be possible, and rate should be pretty high. When $p = .5$, any received codeword $\hat{x}$ could be the result of *any* sent codeword x, and therefore recovery is impossible. However, is recovery possible when $p = .499$, even if this means that the rate has to be tiny? The answer to this question is not obvious.

Shannon Entropy. Shannon entropy $H(p)$ is defined as $p \log \frac{1}{p} + (1-p) \log \frac{1}{1-p}$ (all logarithms are base 2). In particular, when p is close to 0 or 1, entropy approaches 0; when $p = \frac{1}{2}$ (which corresponds to a uniformly random string), entropy is the highest (1).

Capacity of the channel. Capacity of BSC$_p$ is defined as $1 - H(p)$. Shannon theorem states that reliable communication is possible, as long as capacity of the channel is non-zero (i.e. as long as $p < \frac{1}{2}$):

Theorem 1 (Shannon's Coding Theorem, informal)**.** *Reliable communication over BSC_p is possible with any rate below $1 - H(p)$, and impossible with rate above $1 - H(p)$.*

Now let's formalize this statement:

Theorem 2 (Shannon's Coding Theorem)**.** *Let BSC_p be a binary symmetric channel with error probability p. Then*

- $\forall \epsilon > 0\ \exists \delta > 0$ *such that* $\forall k, n$, *which satisfy* $\frac{k}{n} < 1 - H(p) - \epsilon$, *there exists an encoding function* $E : \{0,1\}^k \to \{0,1\}^n$ *and a decoding function* $D : \{0,1\}^n \to \{0,1\}^k$ *such that*
$$\Pr[D(BSC_p(E(m))) \neq m] \leq 2^{-\delta n}.$$
- $\forall \epsilon > 0\ \exists \delta > 0$ *such that* $\forall k, n$, *which satisfy* $\frac{k}{n} > 1 - H(p) + \epsilon$, *for any encoding function* $E : \{0,1\}^k \to \{0,1\}^n$ *and for any decoding function* $D : \{0,1\}^n \to \{0,1\}^k$,
$$\Pr[D(BSC_p(E(m))) = m] \leq 2^{-\delta n}.$$

Here probability is over the choice of m (uniformly at random from $\{0,1\}^k$) and over noise.

Before proving the theorem, we recall several useful lemmas:

Lemma 3. *(Chernoff bound) Let $x_1, \ldots, x_n$ be i.i.d. random variables, such that each $x_i \in [0,1]$. Denote $E[x_i] = \mu$. Then*
$$\Pr[|\frac{\sum x_i}{n} - \mu| \geq \epsilon] \leq \exp(-\epsilon^2 n).$$

Essentially, Chernoff bound says that the average of several random variables is very close to their mean (except with negligible probability).

Lemma 4. *Let $p \in (0, \frac{1}{2})$. Then* $\binom{n}{pn} \approx 2^{H(p)n}(1 + o(1))$.

Exercise 1. Prove lemma 4.

Lemma 5. *Let $p \in (0, \frac{1}{2})$. Then volume V of the ball of radius pn in n-dimensional space is* $\sum_{i=0}^{pn} \binom{n}{i} \approx O(2^{H(p)n})$.

Exercise 2. Prove lemma 5.

Now we are ready to prove Shannon's theorem:

Proof. Set E to be a randomly chosen function from k bits to n bits. Let γ be a parameter, which depends on ϵ and p and which we define later. We define a decoding function as follows: on input $\hat{x}$ it goes over all possible m and computes their encodings $E(m)$. If there exists a unique m such that $E(m)$ lies within a ball with center $\hat{x}$ and radius $(p+\gamma)n$, then $D(\hat{x})$ outputs this m. Else it outputs $\perp$.

To show that this decoding is almost always correct, we need to show two things:

- that $\hat{x}$ falls withing a ball with center x and radius $(p+\gamma)n$ almost always. Intuitively, this holds since corrupted codewords should be concentrated at distance pn from x, and as n grows, probability to be sufficiently far away from x becomes small;

- that the ball with center $\hat{x}$ and radius $(p + \gamma)n$ rarely contains a codeword of another message. Intuitively, this holds since the volume of this ball is small enough compared to the volume of the whole space of codewords.

Now let's give a formal proof. We will show that for our choice of E, D, probability of incorrect decoding is exponentially small in n, where probability is taken over the choice of m, noise, *and encoding function E*. This will imply that for at least one E the probability (over m and noise) is small, as claimed by the theorem.

First let's show that $\hat{x}$ almost always falls into the ball. Let e be an error vector. We need to show that $\Delta(e) \geq (p+\gamma)n$ with negligible probability[1]. By Chernoff bound, for any γ the probability that $|\frac{\sum e_i}{n} - p| > \gamma$ is at most $\exp(-\gamma^2 n)$; therefore $\Delta(e) \geq (p+\gamma)n$ with probability at most $\exp(-\gamma^2 n)$, as required.

Now let's compute the probability that the ball contains a codeword for another $m' \neq m$. Since E is a random function, the probability that for some fixed m' $E(m')$ hits the ball is $\frac{V}{2^n}$ (where V is the volume of the ball), which is approximately $2^{H(p+\gamma)n}2^{-n}$ (lemma 5). Then, by union bound, the probability that there exists $m' \neq m$ such that $E(m')$ hits the ball is at most $2^k 2^{H(p+\gamma)n}2^{-n}$, which can be rewritten as follows:

$$2^k 2^{H(p+\gamma)n}2^{-n} = (2^{\frac{k}{n}+H(p+\gamma)-1})^n \leq (2^{1-H(p)-\epsilon+H(p+\gamma)-1})^n = (2^{-\epsilon+H(p+\gamma)-H(p)})^n;$$

here we used that $\frac{k}{n} \leq 1 - H(p) - \epsilon$. By setting γ sufficiently small, we can make $H(p+\gamma) - H(p)$ be at most, say, $\frac{\epsilon}{2}$, and thus

$$(2^{-\epsilon+H(p+\gamma)-H(p)})^n \leq 2^{(-\epsilon+\frac{\epsilon}{2})n} = 2^{-\frac{\epsilon}{2}n}.$$

Thus, probability of incorrect decryption is at most $2^{-\frac{\epsilon}{2}n} + \exp(-\gamma^2 n)$, which is exponentially small in n, as required.

□

Note that both encoding and decoding algorithms constructed in the proof are quite inefficient (require double exponential and exponential time).

We also give a proof sketch for the converse theorem:

[1]Here $\Delta(e) = \sum e_i$ is a Hamming weight of e.

Proof. Let's consider a bipartite graph with all messages on the left, all n-bit strings on the right, and each message m connected to every n-bit string which is at distance exactly pn from $E(m)$. Intuitively, each n-bit string will be connected (i.e. at the same distance pn) to too many messages, making recovery impossible (note that for any m $E(m)$ could be transformed into any neighbor of m with the same probability, which means that any n-bit string c contains no information about which one of all c's neighbors was initially encoded). Indeed, the degree of each m-node is $\binom{n}{pn} \approx H(p)n(1+o(1))$ (lemma 4), and therefore the number of edges in the graph is $2^k 2^{H(p)n(1+o(1))}$, which is also the amount of all possible decoding attempts. However, the amount of correct decodings is only 2^n, and thus the fraction of correct decoding over all possible ones is

$$2^n 2^{-k} 2^{-H(p)n(1+o(1))} = (2^{1-\frac{k}{n}-H(p)(1+o(1))})^n \leq (2^{1-1+H(p)-\epsilon-H(p)(1+o(1))})^n = (2^{-\epsilon-H(p)o(1)})^n \leq 2^{-\frac{\epsilon}{2}n},$$

for sufficiently large n. □

References

[Sha48] C. Shannon. A mathematical theory of communication. *Bell system technical journal*, 27, 1948.