

Lecture 2

Basic results and definitions of Shannon theory. Asymptotics \rightarrow law of large numbers \rightarrow typical sequences. Direct coding (achievability) thms and converse (optimality) thms.

Suppose there is a random variable X that can take different values $\{x\}$ w/ prob $p_X(x)$.

How much do we learn — how much info do we gain — when we learn the particular value of x ?

1. If X always takes a particular value x_0 , we learn nothing: $P_X(x_0) = 1$.
 2. We learn more from unusual values than from common ones. E.g., most common letters are etaoinsfhndl.

Dropping the ~~most~~ vowels:

~~We're still digging up most from
the area let's do some more.~~

~~The Above Sections~~

The bvr schm suggests wytmsr nfrmtn

Drop consonants:

c aoe ee uegaſbeave ioaio

The Function

$$i(x) \equiv \log_2 \left(\frac{1}{p(x)} \right) = -\log_2(p(x))$$

②
or "surprise!"

is the "information content" of x . It is 0 for $p=1$; it grows as $p \rightarrow 0$; and it is additive for independent random vars:

$$P_{XY}(x,y) = P_X(x) P_Y(y)$$

$$\Rightarrow i(x,y) = i(x) + i(y).$$

The choice of \log_2 means that the units of $i(x)$ are bits. (Natural logs are measured in "nats".)

The average information content of X is then $\mathbb{E}[i(X)] = \sum_{x \in X} p_X(x) i(x)$

$$= - \sum_x p_X(x) \log_2 p_X(x)$$

$$= H(X)$$

This is the Shannon Entropy of X . For independent random vars H_{xy} is additive:

$$-\sum_{x,y} p_X(x) p_Y(y) \log_2 p_X(x) p_Y(y)$$

$$= - \sum_{x,y} p_X(x) p_Y(y) [\log_2 p_X(x) + \log_2 p_Y(y)]$$

$$= \sum_x p_X(x) (\log_2 p_X(x)) - \sum_y p_Y(y) (\log_2 p_Y(y))$$

$$= H(X) + H(Y)$$

Shannon entropy also has an operational interpretation: it gives the extent (or rate) at which a random source can be compressed asymptotically without loss. ③

E.g., A random source produces ~~four~~ 4 symbols a,b,c,d with probs $\frac{1}{2}, \frac{1}{8}, \frac{1}{4}, \frac{1}{8}$. We would like to encode these as a string of bits using the minimum # of bits on average.

Attempt 1: $a \rightarrow 00, b \rightarrow 01, c \rightarrow 10, d \rightarrow 11$.

Uses 2 bits for every symbol.

Attempt 2: $a \rightarrow 0, b \rightarrow 110, c \rightarrow 10, d \rightarrow 111$

Average length is

$$\frac{1}{2} \cdot 1 + \frac{1}{8} \cdot 3 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 = \frac{7}{4} < 2.$$

In fact, this is the shortest encoding possible. (You can try to do better... it's not hard to see if can't be done.) Note, these bit strings are unambiguous—we can always tell how to divide up the symbols.

0011010111010100010 → 0/0/10/10/111/0/10/
10/0/0/10

The entropy of this r.v. is

$$H = -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{8} \log_2 \frac{1}{8} - \frac{1}{4} \log_2 \frac{1}{4} - \frac{1}{8} \log_2 \frac{1}{8} = 7/4.$$

Most examples do not work out so neatly. But we can get around this problem by lumping together an entire string of symbols from the source, and using an encoding for the whole string. This uses the idea of typical sequences. (4)

Suppose a source produces an i.i.d. string of letters x drawn from an alphabet X . There is a ~~redundant~~ random var \bar{X} w/ prob. mass func.

$$P_{\bar{X}}(x) \Rightarrow H(\bar{X}) = - \sum_{x \in X} P_{\bar{X}}(x) \log_2(P_{\bar{X}}(x)).$$

Now consider the r.v. \bar{X}^n whose values are i.i.d. strings of symbols $x^n = x_1 x_2 \dots x_n$.

The prob. mass func is

$$P_{\bar{X}^n}(x^n) = \prod_{i=1}^n P_{\bar{X}}(x_i)$$

Define $N(a|x^n) \equiv \#$ of occurrences of the letter a in the string x^n . Then

$$P_{\bar{X}^n}(x^n) = \prod_{a \in X} P_{\bar{X}}(a)^{N(a|x^n)}$$

So all strings where all letters occur the same # of times have the same probability.

A typical sequence is a string x^n where the frequencies of the symbols are approximately the same as their a priori probabilities.

That is, where

$$\frac{N(a|x^n)}{n} \approx p_{\bar{x}}(a)$$

Let's make this quantitative. Define the sample entropy of the random variable \bar{X}^n :

$$\begin{aligned} -\frac{1}{n} \log_2(p_{\bar{x}^n}(\bar{X}^n)) &= -\frac{1}{n} \log_2 \left(\prod_{a \in X} p_{\bar{x}}(a)^{N(a|\bar{X}^n)} \right) \\ &= -\sum_{a \in X} \frac{N(a|\bar{X}^n)}{n} \log(p_{\bar{x}}(a)) \end{aligned}$$

This quantity is itself a random variable. The probability that this quantity will differ from $H(\bar{X})$ appreciably as $n \rightarrow \infty$ goes to zero:

$$\lim_{n \rightarrow \infty} \Pr \left\{ \left| -\frac{1}{n} \log_2(p_{\bar{x}^n}(\bar{X}^n)) + \sum_{a \in X} p_{\bar{x}}(a) \log_2 p_{\bar{x}}(a) \right| \leq \delta \right\} = 1$$

\uparrow
 $H(\bar{X})$

for all $\delta > 0$.

So as $n \rightarrow \infty$, it is highly likely that the sample entropy will be close to the true entropy.

set of
 The typical sequences is the set of all
 sequences whose sample entropy is close
 to the true entropy. (This is not the only
 way to define it, but the different ways
 of defining the set substantially overlap as
 $n \rightarrow \infty$. And for now, we omit details of
 ϵ 's and δ 's, which we will return to
 later.) The typical sequences have these
 properties:

- ① The size of the set is

$$\approx 2^{nH(x)}$$

By contrast, the set of all sequences is
 of size $2^{n\log|x|}$, which is generally
 much larger, unless $p(x) \approx 1/|x| \forall x$.

② Unit probability. As $n \rightarrow \infty$ the probability
 of the typical set $\rightarrow 1$. "Concentration
of measure".

③ The fraction of sequences that are
 typical is $\approx 2^{-n(\log|x|-H(x))}$,
 exponentially small as $n \rightarrow \infty$.

④ The probability of typical sequences
 are roughly equal $\approx 2^{-nH(x)}$.
 "Equipartition".

We can now define a compression algorithm (7)
that is the core of Shannon's noiseless (source)
coding thm:

- ① List all the typical sequences of length n . There are $\approx 2^{nH(x)}$ of them.
- ② To each sequence, assign a binary string of length $\lceil nH(x) \rceil$. There are $\approx 2^{\lceil nH(x) \rceil}$ of them.
- ③ Divide the input from the source into blocks of n symbols. Look each up in the table and transmit the binary string.
- ④ If the input string is not typical, either register an error, or have an escape code that says "not typical" followed by the full sequence. The prob of this $\rightarrow 0$ as $n \rightarrow \infty$.
- ⑤ On receiving the binary string, look it up in the table and output the typical sequence.

Note: this is actually just the direct coding thm. This shows that compression by $\log |X| / H(x)$ is achievable. It doesn't show that this compression is optimal.

For that, one needs a converse thm. This thm states: "Any encoding with prob. of error $\rightarrow 0$ as $n \rightarrow \infty$ must have average length $\geq nH(X)$."

~~and also~~ Many results in I.T. consist of both a direct coding thm and a converse thm. Generally, these are proven by completely different methods. Here, for instance, the converse thm is proven using entropic inequalities. We will not prove this now, but will return to it after ~~passing~~ defining the necessary quantities and proving their properties.

Channel coding.

A channel takes an input x and gives an output y . (For now assume these are both symbols from X .) In a noiseless channel, $y = x$. In a noisy channel we define a conditional probability

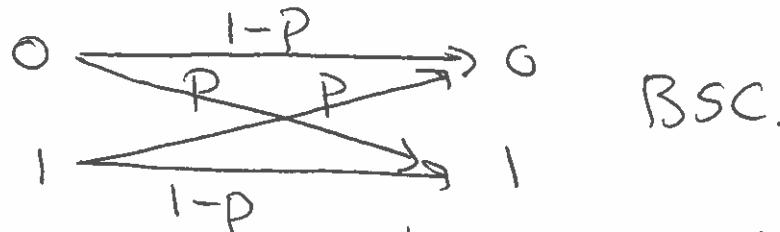
$$N: P_{Y|X}(y|x)$$

← We assume this is independent between channel uses — a "memoryless" channel.

Simple example: the ~~of~~ binary symmetric channel. $X = \{0, 1\}$ ⑨

$$p(0|0) = p(1|1) = 1-p$$

$$p(0|1) = p(1|0) = p$$



We can deal with noisy channels using error-correcting codes^(ECC). E.g., the repetition (majority rule) code:

$$0 \rightarrow \bar{0} = 000$$

$$1 \rightarrow \bar{1} = 111$$

Prob of successfully receiving the bit
is $(1-p)^3 + 3p(1-p)^2$

$$\text{Prob of error} = 3p^2(1-p) + p^3.$$

$$\text{Rate} = 1/3.$$

ECCs work by introducing redundancy.

Suppose we have a long stream of bits to send. What is the best rate we can achieve while having the probability of error $\rightarrow 0$ as $n \rightarrow \infty$? This rate is called the channel capacity.

Shannon proved a simple closed form for this capacity in his noisy channel coding theorem. ①

The key idea of his direct coding proof is to use random codes, and again makes use of typical sequences. (We will again put off the converse theorem proof until later in the class.)

Suppose Alice wishes to send one of M possible messages, which we assume are all equally likely. (If they aren't, we first use source coding — the output blocks of source coding are approximately equal in probability.) She will encode each of the messages as a unique string of n symbols $\underline{x}^n = x_1 x_2 \dots x_n$.

Given the input \underline{x}^n , what are the probabilities of the outputs \underline{y}^n ?

$$P_{Y^n|X^n}(y^n|x^n) = \prod_{i=1}^n P_{X_i|X}(y_i|x_i).$$

For this scheme, the code rate is

$$R = \frac{\text{# message bits}}{\text{# channel uses}} = \frac{\log_2(M)}{n}.$$

A code is a map from the messages $m \in [1, \dots, M]$ ⁽¹⁾ to the strings $x^n(m)$: $C = \{x^n(m)\}$. There also has to be a decoding rule: given a received string y^n , what was the most likely message m that was sent? If the wrong m is decoded, that is an error. The average error prob is

$$\bar{P}_e(C) = \frac{1}{M} \sum_{m=1}^M P_e(m, C)$$

"probability of
error for message
m in code C"

and the maximal prob of error is

$$P_e^*(C) = \max_m P_e(m, C)$$

If $\bar{P}_e(C) \leq \epsilon$, then ~~at least~~ at least half the messages m have ~~at~~ $P_e(m, C) \leq 2\epsilon$.

Random coding Choose ~~M~~ strings ~~at~~ $x^n(m)$ at random from the alphabet X according to some distribution $P_X(x_i)$. $x^n_i = x_1, \dots, x_n$.

$$P_{X^n}(x^n) = \prod_{i=1}^n P_X(x_i)$$

We can now prove results based on the expectation of ~~the~~ the average error over all possible codes.

$$\begin{aligned}\mathbb{E}[\bar{p}_e(c)] &= \mathbb{E}\left[\frac{1}{M} \sum_{m=1}^M p_e(m, c)\right] \\ &= \frac{1}{M} \sum_{m=1}^M \mathbb{E}[p_e(m, c)] \\ &= \mathbb{E}[p_e(1, c)]\end{aligned}$$

the expectation
doesn't depend on m.

So, how can we bound the probability of error for a single message $m=1$?

When we send a string x^n , then as $n \rightarrow \infty$ the probability $\rightarrow 1$ that we will receive a string y^n in the conditional typical set. This set contains $\approx 2^{nH(Y|X)}$ strings, where $H(Y|X)$ ~~is the conditional entropy~~ is the conditional entropy. The strings in this set have roughly equal probability.

$$H(Y|X) = -\sum_x \sum_y p_{Y|X}(y|x) \log_2 p_{Y|X}(y|x) \leq H(Y).$$

$$p_Y(y) = \sum_x p_{Y|X}(y|x) p_X(x)$$

Bob decodes by looking at y and determining which conditional typical set it belongs to. If it belongs to more than one set, there is an error. So the code should be chosen such that the overlap between sets is small.

The typical set of all output strings y^n ③ has size $2^{nH(Y)}$. The conditional typical sets have size $2^{nH(Y|X)}$. So we can partition the typical sets into M , mostly nonoverlapping subsets if

$$M \leq \frac{2^{nH(Y)}}{2^{nH(Y|X)}} = 2^{n(H(Y) - H(Y|X))}$$

Define the rate R to be $R = \frac{\log_2 M}{n}$ so $M = 2^{nR}$. Then we have

$$R \leq H(Y) - H(Y|X) \equiv I(X; Y)$$

\nearrow
Mutual information!

The procedure is as follows:

- ① "Generate" roughly 2^M random strings.
- ② Show that the average error is $\leq \epsilon \xrightarrow[n \rightarrow \infty]{(\epsilon \rightarrow 0 \text{ as})}$.
- ③ Find a particular code that satisfies that inequality. (derandomization)
- ④ Identify half the codewords with max error $\leq 2\epsilon$. Discard the rest. (expurgation).

So long as $R \leq I(X; Y)$, $\epsilon \rightarrow 0$ as $n \rightarrow \infty$.

Note one more thing: the probability distribution $p_x(x_i)$ was chosen by Alice and Bob — it is arbitrary! So we can choose it to maximize the rate R . ④

This maximum defines the channel capacity:

$$C(N) = \max_{p_x(x)} I(X; Y)$$