19 Aug 2024

****

These are unpolished notes I wrote for the first lecture in the ICTS program on Quantum Information, Quantum Field Theory and Gravity.

https://www.icts.res.in/program/qftg

Fuller accounts of the ideas we discuss can be found in standard texts on information theory, e.g.,

(a) Thomas M Cover and Joy A Thomas, Elements of Information Theory
(b) Imre Csiszar and Janos Korner, Information Theory

Some of these topics along with their generalization to the quantum case are also covered in

(c) Edward Witten, A Mini-Introduction To Information Theory, https://arxiv.org/abs/1805.11965

-- Jaikumar
****

We will discuss H[X] and H[Y|X]

Contents

1. The one-shot source coding problem
2. Prefix-free codes
3. Shannon's source coding theorem for one-shot coding
4. Kraft's inequlaity, Gibbs' inequality
5. Types and typical sequences
6. Shannon's source coding theorem for block codes with small errror
7. Conditional entropy

A source or a probabilistic scheme consists of a set of symbols and their corresponding probabilities.

X = [(a_i, p_i): i = 1, 2, ..., n]

The set of symbols {a_1, a_2, ..., a_n} will be called the alphabet of the source. Often we will write p(a) or p_a for the probability of the symbol a. Such source may be used to model an experiment in a lab where there are n distinguishable outcomes, whose occurrence we model using probabilities. The toss of a coin, the roll of a die, the weather, the outcome of an election, indeed, from where the photon would emerge when aimed

at an apparus with two slits, or when aimed at a beam splitter.

Shannon's source coding problem

Imagine two parties, Alice and Bob (for our imagination is so limited that we annot think of other names). Alice observes the outcome of the experiment, Bob would like Alice to inform him by means of a message consisting of a string of bits.

Given: A source X = [(a_i, p_i): i = 1, 2, ..., n]

Task: Assign to each a_i a unique string of bits, w_i, called the codeword for a_i, such that

(i) The codewords are prefix-free (a technical condition that allows Bob to know when the message from Alice has ended.

(ii) sum_i p_i |w_i| is minimum, |w_i| is the number of bits in w_i.

We refer to the above optimum value as the transmission cost for X and denote it by T[X].

The idea is that Alice when she overves a_i will send w_i of length ell_i. The expected cost is the objective value of the optimization problem.

Before we study this constrained optimization problem in a little more detail, let us understand the contstraint (i).

What is prefix free? No sequence should look like the initial part of another.

w_1 = 0011, w_2 = 011, w_3 = 10101 is a prefix-free assignment

by

w_1 = 0011, w_2 = 011, w_3 = 011101 is not prefix-free, because w_2 is a prefix of w_3.

Why prefix-free?

When a string of sybmols is to be communicated, we should be able to tell immediately where the codeword for one symbol has ended. In the above example, on receiving 001, Bob would not be able to tell if the Alice meant to send a_2 or a_3; perhaps some bits are yet to arrive. It turns out that the prefix-free property is not such a central assumption; it just makes some of our deriviations easier.

Kraft's inequality

If w_1, w_2, ..., w_n are prefix-free, the sum_i 2^{-|w_i|} <= 1.

Proof. Draw a binary tree with left edge leaving each node marked 0 and the right edge marked 1. The words w_i appear as leaves in this tree. A random walk from the root passes through w_i with probability 2^{-|w_i|}. Since the w_i are prefix-free, the events of different w_i are disjoint. The inequality follows from this. (end of proof)

Converse of Kraft's inequality.

If numbers ell_1, ell_2, ..., ell_n satisfy sum_i 2^{-ell_i} <= 1, then there are words w_1, w_2, ..., w_n such that (i) the w_i are prefix free and (ii) |w_i| = ell_i.

So, we have the following reformulation of the Source Coding problem.

--- Determining T[X], the transmission cost --------

minimize sum_i p_i ell_i
subject to sum_i 2^{ell_i} <= 1, and ell_i are integers.

Consider the above optimization problem without the constraint that the ell_i be integers.

Claim: The optimum solution is ell_i = log_2 1/p_i. (Today, our logs will be to base 2.)

Observe, that if the claim is true, then the optimum value is

 sum_i p_i log 1/p_i,

(We will have more to say about this later. :)

Proof: First, the clearly ell_i = log 1/p_i satisfies the constraints. Second, consider any solution to the problem: ell_1, ell_2, ..., and compare the value of the objective function for it agains what we have. The difference is

sum_i p_i ell_i - sum_i p_i log 1/p_i
= sum_i p_i log p_i 2^{ell_i}
>= - sum_i p_i log 2^{-ell_i}/p_i
>= log sum_i 2^{-ell_i}
>= 0.

So, no ell_i that satisfies the contraint can do better than what we already have. (end of proof)

It is not hard to see that if we round log 1/p_i up, then we obtain a feasible integral solution to the problem. We thus have

Theorem I (Shannon): H[X] <= T[X] < H[X] + 1

(Shannon's scource coding theorem, with variable length codewords.)

So what is entropy? Why is the entropy of a fair coin toss 1, but the entropy of a (1/4,3/4) coin toss only 0.811, and a (1/3,2/3) coin toss 0.919.

In all cases, it is clear that T[X] = 1.

QUESTIONS: Then, why is H[X] different? Is there a slightly different engineering problem for which perhaps H[X] provides the correct answer?

ANSWER: Shannon's source coding theory with block encoding.

Typical sequences

Imagine sampling from the source repeatedly and independently k times. (The source is memoryless.)

We get a sequence x-bar = x_1, x_2, ..., x_k. We call this source X^k and its distribution is the product distribution, that is,

    Pr[X^k = x-bar] = prod_i Pr[X = x_i]

The number of such sequences x-bar is clearly n^k. However, most of the probability resides on the typical sequences. Fix x-bar. For a in A, let

N(a | x-bar) = number of occurrences of a in x-bar
             = |{i: x_i = a}|

Then, (1/k) N (a | x-bar) is the emperical distribution of a in x-bar. We say that x-bar is eps-typical if this emperical distribution is close to the original distibution:

- sum_a | (1/k) N (a | x-bar) - p(a) | <= eps;
- if p(a) = 0, then N(a | x-bar) = 0.

We refer to the the emperical distribution of x-bar as its type. E.g., the type of 0011110000 (6/10, 4/10)

Fact: For all esp in (0,1), Pr[x-bar is not eps-typical] goes to 0 as k goes infinity.

Let T^k(P,eps) be the set of eps-typical sequences.

[Draw a picture on the board.]

* What is the probability of an eps-typical sequences (wrt the produce distribution)?

  prod_a p(a)^{N(a | x-bar)}

which is approximately 2^{-n H[X]} with some correction for eps. More precisely,

lim_{k->infinity} max_{x-bar: eps-typical} log 1/p(x_bar)
= lim_{k->infty} min_{x-bar: eps-typical} log 1/p(x_bar)
= H[X]

* How many typical sequences are there?

We know that as k becomes large, essentially all the probability resides on typical sequences, and each typical sequence has probability about 2^{-n H[X]}.

lim_{k -> infty} (1/k) log |T^k(p,eps)| = H[X].

[Alternatively, write this number as

          n! / prod_a (n p_a)!

and use Stirling's formula.]

Let S(k,delta) be the smallest cardinality set S of sequences x-bar such that

         P^k(S) = Pr[X^k in S] >= 1-delta

Theorem II (Shannon): For all esp in (0,1)

         lim_k (1/k) log S(k,delta) = H[X].

BACK TO THE TRANSMISSION PROBLEM

If we had decided to stick with the original encoding of one symbol at a time, and transmitted the entire block of k symbols by concatenating the individual codewords, we would get a cost close to kT[X], that is, about T[X] per source symbol. Indeed, with high probability, the lenght of the entire message would be close to k T[X]. It is not hard to see that k H[X] is a lower bound, thus this method would be a

       T[X]/H[X] <= 1 + 1/H[X]

worse than the optimum. If H[X] is large, this would perhaps

be acceptable. If H[X] is small (say 0.001), however, the factor 1/H[X] can be significant.

In fact, for (1/4,3/4) or (1/3, 2/3), this method of concatenating the codewords designed for one-shot communication would give us a rate of 1 bit per symbol, whereas the entropy of these distributions is strictly smaller than 1.

Theorem II tells us that if we block together k symbols, and tolerate a vanishingly small amount of error, then we can get by focussing on the set of typical sequences alone. Since almost all the probability resides on typical sequences, the errror in transmission can be made vanishingly small as k increases, e.g.,

probability of error <= exp(-k^{1/3}).

We then transmit only k H[X] bits, that is, we are able to compress the source to H[X] bits per symbol; this is much less than 1 bit per symbol for the examples of biased coin tosses we considered earlier.

(This is what I wanted to say about H[X].)

CONDITIONAL ENTROPY

(X,Y): random variables with some joint distribution.
X takes values in A
Y takes values in B

Notation:

p(a,b) = Pr[X = a and Y = b], the corresponding distribution is P
p(a)= Pr[X = a], the corresponding distribution is P_X
p(b|a) = Pr[Y=b | X=a]
q(b) = Pr[Y=b]

We may define H[(X,Y)] as before. That is,

H[(X,Y)] = sum_{(a,b)} p(a,b) log 1/p(a,b)

Conditional typicality

Suppose all conditional probabilites are given {p(b|a)} are given.

Fix a sequence x-bar, say its type is P_X = (p(a): a in A). Define the
joint distribution P on A x B by

p(a,b) = p(a) p(b|a)

We would like to know how many y-bar are there so that the pair (x-bar, y-bar) is jointly (P, eps) typical. We may count this as follows.

In x-bar there are N(a|x-bar) = p(a) k positions where a appears. Let us ask what appears in these position in y-bar. If (x-bar,y-bar) is to be jointly typical (with some small tolerance eps'), then among these N(a|x-bar) position the symbols b in B should appear about p(b|a) N(a|x-bar) times. By our discussion above, entropy gells us how many possibilities we have, except that we must now replace k by N(a|x-bar) = p(a)k and the source by the conditional sourc Y | X=a (whose distribution is given by number p(b|a)). Thus, rougly speaking, the number of extension of x-bar to (x-bar, y-bar) so that the pair is jointly typical is

$$\prod_{a \in A} 2^{p(a) k H[Y \mid X=a]} = 2^{k \sum_a p(a) H[Y \mid X=a]}$$

So the possibilities grow exponentially with k, but the coefficient k is the quantity

$$\sum_a p(a) H[Y \mid X=a]$$

We call the conditional entropy of Y given X, and denote it by

$$H[Y|X]$$

Now, the number of jointly typical paris is $2^{k H[(X,Y)]}$.

Intuitively, We may think of generating jointly typical pair (x-bar, y-bar) by first generating x-bar for which we have about $2^{k H[X]}$ choices and then extending it to a jointly typical pair, we have another way of counting jointly typical pairs. Thus we have

$$2^{k H[(X,Y)]} = 2^{k H[X]} \times 2^{k H[Y|X]}$$

or

$$H[(X,Y)] = H[X] + H[Y|X]$$

Next, time we will see that this follows more directly from our formulas, but this intuition will be useful for our discussion of Shannon's channel coding theorem. We often write H[XY] when we mean H[(X,Y)], when one is not likely to confuse XY to mean X times Y.

(End of lecture 1.)