Exercise Sheet 3

Probabilistic models of modern AI

vassilis.papadopoulos@epfl.ch – www.vassi.life/teaching/aiproba

References :

- Cover & Thomas, 'Elements of Information Theory', Chapter 3 [link]

We provide reminder of the different types of convergence in probability theory. You may not need all of them.

Given a sequence of random variables $X_1, X_2, \ldots$, we say that the sequence converges to a random variable X:

- In probability if for every $\epsilon > 0$,
$$\lim_{n\to\infty} P(|X_n - X| > \epsilon) = 0.$$
- Almost surely if
$$P\left(\lim_{n\to\infty} X_n = X\right) = 1.$$

Note that almost sure convergence implies convergence in probability (but not the other way around).

The difference between the two is subtle. When we have convergence almost surely, given $\epsilon > 0$, we can find N such that for all $n \geq N$, $P(|X_n - X| < \epsilon) = 1$. When we have convergence in probability, given $\epsilon > 0$ and $\delta > 0$, we can find N such that for all $n \geq N$, $P(|X_n - X| < \epsilon) > 1 - \delta$. That is, for 'almost surely', we definitely are within ϵ of the limit, while for 'in probability', we are within ϵ of the limit with arbitrarily high probability.

Exercise 1 The typical set

Let $X_1, X_2, \ldots, X_n$ be i.i.d. discrete random variables with distribution $p(x)$. We will be considering sequences $x_1, x_2, \ldots, x_n$ drawn from these random variables.

We will consider the *typical set* of such sequences. Informally, the typical set $A_\epsilon^{(n)}$ is the set of sequences of length n such that most of the probability mass is concentrated on these sequences. In other words, the probability of observing a sequence in the typical set approaches 1 as n grows large.

1. Show the 'Asymptotic Equipartition Property' :[1]
$$-\frac{1}{n}\log p(X_1, X_2, \ldots, X_n) \underset{n\to\infty}{\to} H(X) \text{ in probability} \tag{1}$$
Where $p(X_1, X_2, \ldots, X_n)$ is a random variables that takes value $p(x_1, x_2, \ldots, x_n)$, obtained by sampling the X_i random variables.

This gives us a clear bound on the probability of a sequence, as a function of the entropy of the distribution. We use that to define the typical set.

Definition: The typical set $A_\epsilon^{(n)}$ w.r.t. $p(x)$ is the set of sequences $x_1, x_2, \ldots, x_n$ such that
$$2^{-n(H(X)+\epsilon)} \leq p(x_1, x_2, \ldots, x_n) \leq 2^{-n(H(X)-\epsilon)} \tag{2}$$

2. Show the following properties :

[1]Hint in rot13 : Hfr gur jrnx ynj bs ynetr ahzoref gb pbapyhqr.

a) If $(x_1, x_2, \dots, x_n) \in A_\epsilon^{(n)}$, then $H(X) - \epsilon \leq -\frac{1}{n} \log p(x_1, x_2, \dots, x_n) \leq H(X) + \epsilon$.
b) $P((X_1, X_2, \dots, X_n) \in A_\epsilon^{(n)}) > 1 - \epsilon$ for sufficiently large n.
c) $|A_\epsilon^{(n)}| \leq 2^{n(H(X)+\epsilon)}$ where $|A|$ is the cardinality of the set A.
d) $|A_\epsilon^{(n)}| \geq (1-\epsilon)2^{n(H(X)-\epsilon)}$ for sufficiently large n.

The conclusion from these properties is that the typical set has probability nearly 1, all of its elements are nearly equiprobable, and the number of elements in the set is roughly $2^{nH(X)}$.

Exercise 2 Finding an element in a set

We are given a set S of n elements. Assume elements are drawn uniformly at random from S.

1. We want to find a coding scheme to be able to communicate to somebody else a sequence of elements that are drawn, sending only bits (0's or 1's). For the coding scheme to be valid, it has to be uniquely decodable: the assignment sequences$\rightarrow$codes should be injective. In other words, to a given code, corresponds only one sequence.
   a) Find a coding scheme for the elements of S that uses $\lceil \log_2(n) \rceil$ bits on average.
   b) Express this value in terms of the entropy of the distribution we use to draw elements from S.
2. (hard) Now instead on drawing a sequence, we draw a single element from S, and we want to send only this element.[2]
   a) Assuming $n = 2^{\ell+1} - 2$. Find a coding scheme that uses $\frac{2^{\ell+1}(\ell-1)+2}{2^{\ell+1}-2}$ bits on average.[3]
   b) Show that it is equivalent to the previous scheme when $n \to \infty$ to first order.

Exercise 3 An example of a simple coding scheme

The goal of this exercise is to give a very rough intuitive idea of how we compress random data by exploiting probabilities. If you want to learn more, checkout 'Huffman coding'.

Consider the random variable X that takes 4 values with probabilities $(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8})$. We try to build a prefix-free (instantaneously decodable) code for this random variable. A prefix-free code is such that no codeword appears as a prefix of another codeword.

1. Explain why a prefix-free code is instantaneously decodable. That is, if you receive a sequence of encoded symbols, you can read the sequence from left to right and decode each symbol as soon as you read its codeword (as opposed to waiting for the entire sequence to be received).

To maximize compression, we should assign codewords that are as short as possible. Since not all codes can be equally short (we have only 2^k codewords of length k, much less with the prefix-free constraint), we should assign the shorter codewords to the most likely symbols.

2. We will proceed in a naive, greedy manner. We will assign codewords in order of decreasing probability. Each time, we assign the shortest available codeword that preserves the prefix-free property, and that allows us to continue assigning codes without problem.
   a) Using this method, find a prefix-free code for X.
   b) What is the expected length of the code you found? Is it optimal?
   c) Repeat the procedure, but now using trits (0,1,2) instead of bits (0,1).
   d) What is the expected length you find now?
   e) In practice, modern computers can usually send only bits. What happens if you encode the trits with bits, $(0 \to 0, 1 \to 1, 10 \to 2)$
3. Consider now the random variable Y that takes 5 values with probabilities $(\frac{1}{2}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8})$.
   a) Using the same greedy method, find a prefix-free code for Y.
   b) What is the expected length of the code you found? Is it optimal?
   c) Can you find an optimal prefix-free code for Y?

[2]Hint in rot13 : Vg zvtug or hfrshy gb pbzchgr $\backslash fhz_{x=1}^{\backslash ryy} xn^x$.
[3]I think this is the absolute best scheme in this case, but I am not certain.

Exercise 4 A good coding scheme (half of Shannon's source coding theorem)

We are given a sequence $x_1, x_2, \ldots, x_n \in \mathcal{X}^n$ drawn i.i.d. from a distribution $p(x)$ over a finite alphabet $\mathcal{X}$. We want to find a short (on average) description of such sequences. Formally, a code will be a function $c : \mathcal{X}^n \to \{0,1\}^*$ that must be injective (so that it is *uniquely decodable*; given a sequence of bits, only one sequence corresponds).

1. Our approach will be to assign shorter description to likely sequences, and longer description to unlikely sequences.
   a) Find a coding scheme that can encode any sequence in the typical set $A_\epsilon^{(n)}$ using no more than $n(H(X)+\epsilon)+1$ bits.
   b) We still need to encode sequences outside the typical set. Find a coding scheme that encodes sequences outside the typical set using $O(n)$ bits.
   c) Combine the two coding schemes to find a code for any sequence in $\mathcal{X}^n$, adding only a constant number of bits to the previous schemes. Make sure the code is uniquely decodable.
2. With the coding scheme you constructed, show that, for any ϵ the expected length of the code L (in bits) satisfies:
$$L \le n(H(X)+\epsilon) \tag{3}$$
   for a sufficiently large n.

You have just shown half of 'Shannon's source coding theorem'. Indeed, we have found a coding schemes that uses $H(X)$ bits per symbol on average, given long enough sequences. The second half of the theorem says that no coding scheme can do better than $H(X)$ bits per symbol, so that our coding scheme is actually optimal!