Introduction to Information Theory, Fall 2019

Practice problem set #2

You do **not** have to hand in these exercises, they are for your practice only.

1. **Entropy, essential bit content** Let X be a random variable with probability distribution P with five possible outcomes A, B, C, D and E and probabilities $P(A) = 1/2$, $P(B) = 1/8$, $P(C) = 1/4$, $P(D) = 1/16$, $P(E) = 1/16$.

   (a) What is the entropy $H(X)$?

   (b) Sketch $H_\delta(X)$ as a function of δ.

2. **Enumeration of binary sequences** In the lecture a universal compression scheme was discussed. For this week's homework you will have to implement this scheme, and to help you we will work out an algorithm for the compressor and the decoder in this exercise. Let $\mathcal{A}^N$ be the set of all bitstrings of zeros and ones of length N and let $B(N,k) \subset \mathcal{A}^N$ be set of all strings x^N of length N with k ones. We will then order these sets in an appropriate way, and given x^N we compress by sending over k, the number of ones in x^N, and its index in $B(N,k)$. For the decoder, we just read out the appropriate element from $B(N,k)$. In this exercise we will derive a recursive algorithm for enumerating strings in $B(N,k)$ (notice that these sets will be exponentially large in N so we should not just enumerate over them!). We will use the lexicographic order (denoted $\leqslant_{\text{lex}}$), formally defined as follows: Given bitstrings x and y, we have that $x \leqslant_{\text{lex}} y$ if either $x = y$ or $x_i < y_i$ for the smallest i such that $x_i \neq y_i$. For example, $\mathtt{001} \leqslant_{\text{lex}} \mathtt{010} \leqslant_{\text{lex}} \mathtt{110}$.

   (a) To get some intuition, write down $B(4,2)$ in lexicographically increasing order.

   (b) Argue that

   $$B(N,k) = \begin{cases} \{\mathtt{0}\ldots\mathtt{0}\} & \text{if } k = 0, \\ \{\mathtt{1}\ldots\mathtt{1}\} & \text{if } k = N, \\ \{\mathtt{0}x \mid x \in B(N-1,k)\} \cup \{\mathtt{1}x \mid x \in B(N-1,k-1)\} & \text{otherwise.} \end{cases}$$

   (c) We want to find an algorithm that assigns to a bitstring in $B(N,k)$ its index in the lexicographical order on $B(N,k)$. Argue that Algorithm 1 gives the right result (notice that we start counting at 0, and we use the convention that $\binom{N}{k} = 0$ if $k > N$).

   (d) For the decoding, we need an algorithm that finds the bitstring from k and its index in $B(N,k)$. Argue that Algorithm 2 gives the right answer.

Algorithm 1 Calculate index of a bitstring x

procedure INDEX(x)
  $N \leftarrow$ LENGTH(x)
  $k \leftarrow$ NUMBER_OF_ONES(x)
  if $N = 0$ **then**
    return 0
  end if
  if $x[0] = \mathtt{0}$ **then**
    return INDEX($x[1...]$)
  else
    return $\binom{N-1}{k}$+INDEX($x[1...]$)
  end if
end procedure

Algorithm 2 Calculate string from length N, number of ones k and index m

procedure STRING(N, k, m)
  if $N = 0$ **then**
    return Empty string
  end if
  if $m < \binom{N-1}{k}$ **then**
    return APPEND($\mathtt{0}$, STRING($N-1, k, m$))
  else
    return APPEND(1, STRING($N-1, k-1, m - \binom{n-1}{k}$))
  end if
end procedure