582650 Information-Theoretic Modeling (Fall 2014)

Homework 3 (due September 25)

Please see the instructions for exercises at
`www.cs.helsinki.fi/group/cosco/Teaching/Information/2014/ex/exercise_instructions.pdf` .

1. **Symbol codes.** Let the sets `SET1`, `SET2`, `SET3`, and `SET4` correspond to

   - the set of all possible symbol codes,
   - the set of prefix(-free) codes,
   - the set of codes that satisfy the Kraft inequality,
   - the set of decodable codes,

   in some order.

   (a) How should the sets be chosen (ordered) so that we have

   $$\mathtt{SET1} \subseteq \mathtt{SET2} \subseteq \mathtt{SET3} \subseteq \mathtt{SET4}?$$

   (b) For each of the subset relations, give an example of a code that belongs to the superset but not the subset.

2. **Shannon-Fano code.** Consider a source alphabet $\mathcal{X} = \{ x_1, \ldots, x_m \}$ of size m. Assume we are given propabilities $p_1, \ldots, p_m$ for the source symbols, so $\Pr[X = x_i] = p_i$. Recall that the Shannon-Fano code works as follows:

   (a) Sort the symbols according to decreasing probability so that we can assume $p_1 \geq p_2 \geq \ldots \geq p_m$.

   (b) Initialize all codewords $w_1, \ldots, w_m$ as the empty string.

   (c) Split the symbols in two sets, $(x_1, \ldots, x_k)$ and $(x_{k+1}, \ldots, x_m)$, so that the total probabilities of the two sets are as equal as possible, i.e., minimize the difference $|(p_1 + \ldots + p_k) - (p_{k+1} + \ldots + p_m)|$.

   (d) Add the bit '0' to all codewords in the first set, $w_i \mapsto w_i 0$, for all $1 \leq i \leq k$, and '1' to all codewords in the second set, $w_i \mapsto w_i 1$ for all $k < i \leq m$.

   (e) Keep splitting both sets recursively (Step (c)) until each set contains only a single symbol.

   Simulate the Shannon-Fano code, either on paper or by computer, for a source with symbols $A : 0.9$, $B : 0.02$, $C : 0.04$, $D : 0.01$, $E : 0.015$, $F : 0.015$, where the numbers indicate the probabilities p_i. Evaluate the expected code-length and compare it to the entropy as well as the expected code-length of the Shannon code with $\ell_i = \lceil \log_2 1/p_i \rceil$.

3. **Shannon-Fano code.** Take a piece of text, estimate the symbol occurrence probabilities from it. Then use them to encode the text using the Shannon-Fano code (on a computer). Compare the code-length to the entropy as well as the expected code-length of the Shannon code with $\ell_i = \lceil \log_2 1/p_i \rceil$.

Continues on the next page!

4. **Huffman code.**

(a) Construct a Huffman code for the source in Exercise 2.

(b) Can you come up with a case where the Huffman codeword for a symbol is much shorter than $\lceil \log_2 1/p_i \rceil$? (*Hint:* Consider the binary source alphabet, which would normally make no sense at all!)

(c) This one may take a bit of thinking but the solution is actually quite elegant, so keep trying even if it takes a while.

Suppose you encounter a file where the number of occurrences of symbols a, b, c and d are 1,1,1, and 2, respectively. If you use the frequencies to obtain probabilities, you get $p(a) = p(b) = p(c) = 1/5, p(d) = 2/5$.

When building a Huffman code, you will encounter a tie where you may combine either the pair (a, b) with c, or c with d. Suppose that in such a case you always make the former choice. Note that this way the Huffman tree becomes maximally unbalanced.

Suppose now that there are also other symbols than a, b, c, d in the file (but no more of the aforementioned symbols).

i. What is fewest number of occurrences of symbol e such that the Huffman tree is still maximally unbalanced? How about symbols f, g, h, etc? Give the sequence of counts for m symbols. Do you recognize this sequence?

ii. What is the probability of symbol a if the number of distinct source symbols is m and if the counts are as in the previous item?

iii. What is the depth of the Huffman tree in this case? Note that this is the codeword length for symbol a.

iv. What is the Shannon codeword length $\lceil \log 1/p(a) \rceil$? Compare this the codeword length in the Huffman code.

5. **Randomness.** (*Cover & Thomas, Ex. 5.35*) Your task is to generate a binary random variable $X \in \{0, 1\}$ with $\Pr[X = 0] = p$ with a given parameter $0 < p < 1$. You have access to a potentially infinite sequence of fair coin flips $Z_1, Z_2, \ldots$. Find a procedure for generating X by using as few of the fair coin flips as possible; in particular, show that the expected number of flips that you need is at most two, $E[N] \leq 2$.