582650 Information-Theoretic Modeling (Autumn 2012)
Homework 4 (due 2 October)

You should turn in your solutions on paper at the start of the exercise session.

Except for Problem 3, when considering a code based on a distribution, you may ignore the issue of integer code lengths and just take the code length of \( x \) as \(-\log p(x)\) even if that’s not an integer.

1. Mid-course questionnaire: Please give us feedback about the course. How do you find the pace: too fast or too slow? What do you find most interesting and least interesting: proofs of theorems, algorithms, whatever? How about the exercises? Feel free to give us any other comments about the course and how we could improve it in the future.

2. Consider a parametric model class \( M = \{ p_\theta \mid \theta \in \Theta \} \) in the simple finite case where \( \Theta \) (and thus \( M \)) and \( D \) are finite. Fix some prior \( q(\theta) \) over \( \Theta \), so that \( \sum_{\theta \in \Theta} q(\theta) = 1 \).
   We consider the basic two-part encoding where the parameters are encoded using \( q \), so the codelength for data \( D \in D \) is
   \[
   \ell(D) = \log_2 \frac{1}{q(\hat{\theta}(D))} + \log_2 \frac{1}{p_{\hat{\theta}(D)}(D)}
   \]
   where \( \hat{\theta}(D) = \arg \max_{\theta \in \Theta} p_\theta(D) \) is the maximum likelihood parameter for \( D \).

   Assuming that there is at least one pair \( (\theta, D) \) such that \( q(\theta) > 0, p_\theta(D) > 0 \) but \( \theta \neq \hat{\theta}(D) \) (as there is for any reasonable model), show that the code \( \ell \) is not Kraft-tight, i.e.,
   \[
   \sum_{D \in D} 2^{-\ell(D)} < 1.
   \]
   Hint: consider the sum \( \sum_{(\theta, D) \in L} q(\theta)p_\theta(D) \) where \( L = \{ (\theta, D) \mid \theta = \hat{\theta}(D) \} \).

3. Consider alphabet \( X = \{ a, b, c, ! \} \) with probabilities \( p(a) = 0.4, p(b) = 0.2, p(c) = 0.3 \) and \( p(!) = 0.1 \).

   (a) Find out the interval \( I(cab!) \) that is used for encoding the message \( cab! \) in the arithmetic coding construction described in the lectures. (Use normal decimal arithmetics, don’t worry about accuracy issues in binary representation.)

   (b) Now consider picking a number from \( I(cab!) \) as a codeword for \( cab! \). For ease of calculations, we consider codewords that are decimal numbers, not binary (i.e., the encoding alphabet is \( \{ 0, \ldots, 9 \} \) instead of \( \{ 0, 1 \} \)).

      i. What is the shortest codeword (decimal number with the least number of decimals) you can find within the interval?

      ii. What is the shortest codeword \( C = 0.d_1 \ldots d_k \) such that also all its continuations (numbers of the form \( 0.d_1 \ldots d_k d_{k+1} \ldots d_m \) where \( m > k \) and \( d_i \) can be arbitrary for \( i = k + 1, \ldots, m \)) are also within the interval? (This is the property we need for a prefix code.)

      iii. What is the codeword picked by the method described in lecture notes? Notice that since we are using decimal encoding, we need to use base 10 also in logarithms.

   (c) (Optional) Same as above, but use binary encoding. To make calculations more reasonable, you may round the probabilities to \( p(a) = 3/8, p(b) = 2/8, p(c) = 2/8 \) and \( p(!) = 1/8 \).

Continues on the next page!
4-5. Consider the independent Bernoulli model for sequences of $n$ bits. That is, for a sequence $x^n = x_1 \ldots x_n$ we have $p_\theta(x^n) = (1 - \theta)^{n-s(x^n)} \theta^{s(x^n)}$ where $s(x^n) = \sum_{i=1}^{n} x_i$. It is well known (and easy to see) that the maximum likelihood parameter is $\hat{\theta}(x^n) = s(x^n)/n$.

To use two-part encoding, we quantize the parameter space $\Theta = [0, 1]$ into $\Theta_m = \{0, 1/m, 2/m, \ldots, 1\}$. That is, $\Theta_m$ consists of $m + 1$ uniformly spaced values. This gives us a finite model class $\mathcal{M}_m = \{p_\theta \mid \theta \in \Theta_m\}$. We use the uniform code length of $\log_2(m + 1)$ bits to encode a model from $\mathcal{M}_m$.

Write a program that allows you to draw a sequence of $n$ random bits from an independent Bernoulli($\theta^*$) source, for some the “true” parameter $\theta^*$ you choose. Your program then finds among the models $\mathcal{M}_m$, for $m = 1, 2, 3, \ldots$, the one that gives the smallest two-part code length for the sequence.

Try out a wide range of sequence lengths $n$, and for each of them draw several sequences $x^n$ and determine the $m$ which gives shortest code. Can you make any interesting observations about dependence between $n$ and $m$? In particular, theoretical considerations suggest that having $m$ roughly proportional to $\sqrt{n}$ might be a good idea; can you see something like this empirically? Does the choice of $\theta^*$ have any effect?

(This has been counted as two problems to keep the total work load more manageable. For purposes of grading, and perhaps also for planning your work, you may think that Problem 4 is to implement and test the programs you need, and Problem 5 is to then make the experiments and draw conclusions.)

**Bonus Problem** Same as Problems 4–5, except that instead of two-part encoding you should use a mixture code based on $\mathcal{M}_m$. Use uniform mixture weights $w(\theta) = 1/(m + 1)$. 
