MDL exercises, second handout

(due February 25)

1. Combinatorics and Fixed Length Codes.

   a) Show that the number of binary strings of length n with k zeroes is $\binom{n}{k}$.

   b) How many bits does it take to code a binary sequence of length n with k zeroes with a uniform code (assuming both n and k are known to the decoder)?

2. Maximum likelihood.

   a) The Bernoulli probability of a sequence with n_0 zeroes and n_1 ones is $\theta^{n_1}(1-\theta)^{n_0}$. Compute the maximum likelihood estimator for the parameter, that is the value of θ that maximizes this probability.

   b) The numbers $x_1, \ldots, x_n$ are sampled from an exponential distribution, which has density function $f(x) = \lambda e^{-\lambda x}$. Compute the maximum likelihood value for λ.

   c) Suppose that we model data with a uniform distribution on the real numbers between a and b. Given outcomes $x_1, \ldots, x_n$, what are the maximum likelihood values for a and b?

3. Context Free Grammars

   (a) Consider the Context Free Grammar (CFG) described in Section 1 of the handout. Consider data D consisting of the single sentence `The statistician avoids the model`. Compute the code lengths of this data given the grammar on top of page (3), as well as given the promiscuous, and the ad-hoc grammars, using the code described in the handout. Use the following grammar for D:

   $$D \rightarrow SD \mid \epsilon$$

   Use the diamond to separate sentences or end the data *only if necessary.*

   (b) Again calculate $L(D|H)$ given the grammar on top of page 3 of the handout, but now for the sentence `The statistician avoids the big complex model`.

   (c) Again calculate $L(D|H)$ given the grammar on top of page 3 of the handout given both sentences, but now with a grammar which is slightly modified: "Adjectives" in the second rule is replaced by "Adjective", and the fourth rule (starting with "Adjectives") is removed.

4. *This question can only give you bonus points. But* do *try to come up with a good answer!* Somebody claims that the code $L(H)$ for encoding hypotheses given in the handout makes no sense: each production rule is encoded as a sequence of bitstrings indicating (non-) terminal symbols, but it is nowhere specified which of these bitstrings corresponds to which word in natural language (e.g. `prefers` might be encoded as 00101, but how can the decoder know this?). Explain why this is not a real problem.