Adaptive Information Processing
Hints to the exercises
for Model complexity and the MDL principle

Bert de Vries and Tjalling Tjalkens
Signal Processing Group

February 22, 2008

Bayes and the Laplace method

1. The two envelope paradox. See `http://www.anc.ed.ac.uk/~amos/doubleswap.html` for an answer.

2. Consider the following integral (similar to the Beta integral)

$$F(\mu_1, \mu_2) = \int_{-\infty}^{\infty} \left(\frac{1}{1+e^{-a}}\right)^{\mu_1} \left(\frac{e^{-a}}{1+e^{-a}}\right)^{\mu_2} da.$$

   (a) Use Laplace's method to approximate this integral.

   Hint:
   Just follow the steps in the lecture notes.

   (b) Use the Beta integral

$$B(\mu_1, \mu_2) = \int_0^1 p^{\mu_1 - 1}(1-p)^{\mu_2 - 1}\, dp = \frac{\Gamma(\mu_1)\Gamma(\mu_2)}{\Gamma(\mu_1+\mu_2)}$$

   with

$$\Gamma(x+1) = x\Gamma(x)$$
$$\Gamma(1) = 1$$
$$\Gamma(0.5) = \sqrt{\pi}$$

   and compare your approximation with the actual values in the cases where $\mu_1 = \mu_2 = 0.5$ resp. $\mu_1 = \mu_2 = 1$.

   Hint:
   Consider the transformation from p to $\frac{1}{1+e^{-a}}$ in the Beta integral.

Kolmogorov complexity

1. $f(x^n)$ is a *Boolean function* of n variables, so all x_i are binary, $x_i \in \{0,1\}$ for $i = 1, 2, \ldots, n$, and also $f(x^n) \in \{0,1\}$.

   Give an upperbound to the *conditional Kolmogorov complexity* of a Boolean function.

   Hint:

   A boolean function of n variables can be described by its 2^n outcomes, one for each value of the input vector. So we consider a binary string of length 2^n.

   Its complexity can be upper bounded by $n + 2^n h(f) + c$ where f is the fraction of ones in the outcome list.

   Most binary functions will have maximal complexity, equal to $2^n + c$ bits.

2. n_1 and n_2 are positive integers. Argue that

   $$K(n_1 + n_2) \leq K(n_1) + K(n_2) + c$$

   Hint:

   p_1 is the program for n_1 and likewise p_2 for n_2. We build a program from p_1 and p_2 that computes the next digit from n_1 and from n_2, least significant digit first. In stead of printing these we print $d_{1i} + d_{2i} + c_i$ where c_i is the carry from the previous digit. This part has constant length so the program p_{1+2} has a length equal to the sum of p_1 and p_2 plus a constant part. If p_1 and p_2 produce the digits in the other order we just store them temporarily at no essential extra cost.

Universal data compression

1. The Shannon-Fano code and Huffman code.

   Consider a binary i.i.d. souce that generates $X_1, X_2, \ldots, X_n$ with the parameter $\theta = \Pr\{X = 1\} = 0.1$.

   Compute, for $n = 1, 2, 3$, the expected code wordlength for the Shannon-Fano code, with lengths

   $$l_C^*(x^n) = \lceil -\log_2 p(x^n) \rceil.$$

   Likewise for the Huffman procedure, see lecture notes Information Theory (5K020/5JJ40).

   Give your comments on this result, (and consider here the source entropy).

   Hint:

   No hint needed, just do it. For the Huffman procedure you can read the lecture notes of Information Theory I or any basic Information Theory textbook.

2. [Hard: See sheets 97–92] Show that

$$\bar{p}(x^n) < \sqrt{\frac{\pi}{2n}} e^{\frac{1}{3n}} \left(\frac{k}{n}\right)^k \left(\frac{n-k}{n}\right)^{n-k},$$

where

$$\bar{p}(x^n) = \int_0^1 (1-\theta)^{N(0|x^n)} \theta^{N(1|x^n)}\, d\theta.$$

Hint:

It's hard and you're all on your own. You can make good use of Stirling's formula, see page 91 of the lecture notes.

ML and MDL

1. Assume x^n are i.i.d. observations from $\mathcal{N}(\theta, 1)$, so the x_i's are independent Gaussians with unit variance but unknown mean $\theta \in \mathbb{R}$. We test two hypothesis, $H_0 : \theta = 0$ versus $H_1 : \theta \neq 0$. Otherwise said, we want to choose between the models

$$\mathcal{M}_0 = \{\mathcal{N}(0,1)\} \text{ and } \mathcal{M}_1 = \{\mathcal{N}(\theta,1)|\theta \neq 0\}$$

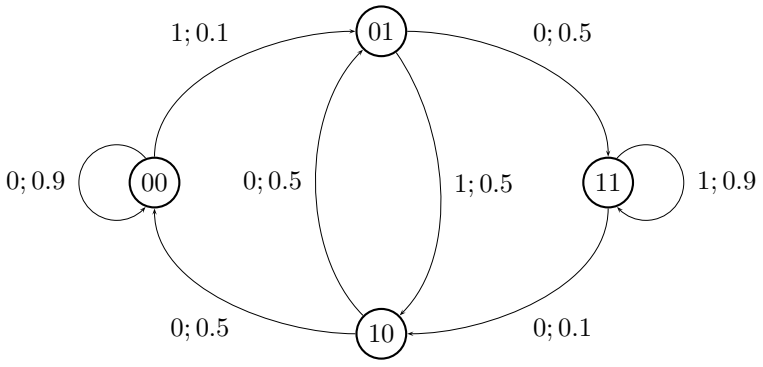
Derive that if we compute the ML probabilities for each model and then choose for the model with the largest ML probability we will never choose for $\mathcal{M}_0$ even if x^n was actually generated by $\mathcal{M}_0$.

Hint:

The proof is based on the fact that for $\mathcal{M}_1$ we estimate the mean and only if the estimated mean is equal to zero, $\mathcal{M}_0$ will be selected. This is an event of probability zero.

Compare this to the MDL result of pages 124–125. Doesn't that result seem more satisfying?

2. Consider this following 1^{th}-order binary Markov source. Next to the arrow from state a to state b is written $x; \Pr\{X_i = x, S_i = b | S_{i-1} = a\}$.



(a) Determine the probability $\Pr\{X_i = 1\}$.
Hint: Compute the stationary state distribution and then marginalize $\Pr\{X_i = 1, S_{i-1} = s$ to obtain $\Pr\{X_i = 1\}$.

(b) Consider an "ideal" universal datacompression algorithm and we observe a sequence x^n that is typical for the source. How large must i be approximately to select the first order Markov model in stead of the memoryless model.

Hint:

Look at the discussion in the notes " Stochastic Complexity (MDL)". This is similar! Also the lecture notes Information Theory (5K020/5JJ40) will be helpful.

3. [Hard:] can you determine the number of suffix trees with maximal depth not more than D for $D = 0, 1, 2, \ldots, 10$?

Hint:

You should be on your own again but ok.

Let F_i be the number of binary trees of maximum depth at most i. Then

$$F_i = 1 + F_{i-1}^2$$

The 1 comes from the tree containing only a root and otherwise all trees with maximum depth i can be written as the tree of depth 1 having both a tree of depth at most $i-1$ at its zero node and its one node.

D	$\#trees$
0	1
1	2
2	5
3	26
	...
10	$1.437821978 \cdot 10^{181}$

(Unless I made an error in calculating this recursion.)

So the number of models is HUGE for reasonable D. e.g. we use this CTW method for text compression on a byte depth of 12 characters. This is $D = 96$. Still the amount of work is linear in the sequence length times D.

Appendices

1. Using the idea of Elias show that we can find code words for the positive integers with lengths upper bounded as

$$l(n) \leq \log_2 n + 2 \log_2 \log_2 n + c.$$

Hint:

Consider an integer i. We need $m_i = \lfloor \log_2(i+1) \rfloor$ bits to describe i. We first encode m_i using the original scheme in $2\log_2 m_i + c_1 = 2\log_2 \log_2 i + c_2$ bits. Then we transmit i in $m_i = \log_2 i + c_3$ bits.

2. Consider an alphabet $\mathcal{X}$ and sequences of length $n = 50$ of symbols from this alphabet. Let $\mathcal{X} = \{0, 1, 2\}$ and let for the sequence x^n hold:

$$N(0|x^n) = 20; \quad N(1|x^n) = 14; \quad N(0|x^n) = 16.$$

   (a) Compute the size of the type-class $T(\frac{20}{50}, \frac{14}{50}, \frac{16}{50})$.

   (b) Also compute the upper and lower bound according to theorem 6.

Hint:

No hint is needed.

3. Prove Lemma 2 for m is even.

   (a) Consider the requirement $\theta_i - t_{i-1} = t_i - \theta_i$. This condition ensures that the quadratic upperbound to the divergence is the same for both interval endpoints of $[t_{i-1}, t_i]$.
   From this conclude that

$$t_i = 2\sum_{j=1}^{i} (-1)^{i-j}\theta_j$$

   (b) Now take into account the boundary conditions

$$t_0 = 0; \quad t_{\frac{m}{2}} = \frac{1}{2}$$

   and the requirement that θ_i increases quadratically, or

$$\theta_i = \alpha i^2, \text{ for some constant } \alpha$$

   Prove that

$$\sum_{j=1}^{i} (-1)^{i-j} j^2 = \frac{1}{2} i(i+1)$$

   and use this with the conditions above to derive that

$$\alpha = \frac{2}{m(m+2)}$$

   (c) Now finally derive Lemma 2 for m is even.

Hint:

The steps are not easy but simple enough, you should be able to do it! However, if asked during the exam, some hints will be supplied.

Hint for (a): Solve the given equality for t_i and you get a recursion in the t's. Then use the fact that $t_0 = 0$.

Hint for (b): Suppose first that i in $\sum_{j=1}^{i}(-1)^{i-j}j^2$ is even and write the sum as $\sum_{j=1}^{i/2}(2j)^2 - (2j-1)^2$. Then check what happens when i is odd.

Hint for (c): This is straightforward now, just follow the steps in the lecture notes.