

Information Theory — Tutorial 3

Iain Murray

October 10, 2014

Remember to use NB for questions and comments on any of the lecture materials.
Email the lecturer if you haven't got an invite to NB.

1. Three questions on Huffman coding:

Do find at least a couple of Huffman codes by hand to check your understanding. However, manually building many large trees can be tedious without a computer. When I did these questions I wrote some Matlab/Octave code (now ported to Python and Haskell), which you are welcome to use (or write your own!):

```
http://www.inf.ed.ac.uk/teaching/courses/it/code/huffman.m  
http://www.inf.ed.ac.uk/teaching/courses/it/code/huffman.py  
http://www.inf.ed.ac.uk/teaching/courses/it/code/huffman.hs
```

MacKay's book Exercise 5.21, p102.

Notation: "codes for X^n " means codes for blocks of n concatenated symbols.
For example, when coding X^2 the *source* alphabet is $\mathcal{A}_{X^2} = \{00, 01, 10, 11\}$.

MacKay Exercise 5.22, p102.

MacKay Exercise 5.29, p103.

2. **The meaning of unbiased:** Dr. Fran Quentin has devised an unbiased estimator, $\hat{\sigma}^2$, for the variance, σ^2 , of a process. The head of the lab, Prof. Natalie Ural is actually interested in the precision $\tau = 1/\sigma^2$, while the programmer on the project only cares about the log variance, $L_{\text{var}} = \log \sigma^2$. At a heated lab meeting Dr. Quentin is accused of incompetence: experience with her 'unbiased' estimator suggests that it tends to underestimate the variance. Why might this have happened? Is it possible that another lab member could simultaneously think that $\hat{\sigma}^2$ tends to be too big?

I won't put silly stories or quite such intentionally vague questions in the exam. But I do think it's important for you to be able to work out how things you've learned might apply when it's not clear what's going on or what the question is.

3. **Arithmetic coding:** Read chapter 6 of MacKay. Answer Ex 6.1: show that arithmetic coding encodes within 2 bits of the information content of a message.

My experience is that many students simply copy down MacKay's answer. Are you sure you understand it? Your goal is to be able to explain it to someone else so that they'll be convinced. If you really get it, you may be able to generalize: What's the maximum overhead of an arithmetic coder that encodes a file into a ternary stream with symbols from $\{\alpha, \beta, \gamma\}$?

I should have finished covering arithmetic coding by the Friday before your tutorial.