Some review questions for Information Theory

Iain Murray

November 4, 2012

Here is a strategy for revising courses: First, go over all the material. For each part, imagine trying to explain it to someone else. Can you say how it relates to other parts of the course? Try to create a small example, application or extreme case and play with it. If you have trouble, find the material in another reference, or ask a friend. If/when you understand the material, imagine what questions could be asked about it. Ask yourself how you might check the answers to those questions yourself.

Two of the students with the best (very good) exam results last year, were those that reviewed all the material very carefully. How do I know they did this? They asked many good questions on NB.

By request, I have run over the class notes and, in response to what I saw, quickly written a bunch of review questions (less polished and thought out than exam questions). This is something you could do for yourself, and I do encourage you to think beyond these questions. I am *not* going to provide detailed worked answers to these questions. The tutorial questions already come with extensive answers, as do many questions in the textbook.

You are moving towards doing independent research: you need to be able to come up with questions, *and answers*, yourself.

That said, I will answer *specific* questions posted on NB, if you have made some attempt yourself. I will also give feedback on any written answers that you hand to me. There is a deadline: as stated on the website, in the week before the exam I don't guarantee any level of responsiveness, and I don't meet with students.

1. Show that the probability of error in a repetition R_N code applied to a binary symmetric channel with flip probability f is less than $2^N(f(1-f))^{N/2}$.

2. (a) How many bits are required to index K items using a fixed-width binary encoding? If this fixed-width binary encoding is used as a symbol code, is it uniquely decodable? Instantaneously decodable?
   (b) Fred thinks that a fixed-width encoding is wasteful, and decides to index the eight possible outcomes of an experiment with the symbols '1', '10', '11', '100', '101', '110', '111', '000'. Would this code be a uniquely decodable symbol code if used to store the outcomes of multiple experiments?
   (c) What is the limit on the lengths of binary codewords that are uniquely decodable? When using binary symbol codes, why do we always use instantaneously decodable codes?
   (d) Fred now terminates the codes for multiple outcomes with dashes, for example: '11-110-100-000-'.
   By treating this code as a ternary symbol code with output alphabet $\{0, 1, -\}$, is this code uniquely decodable? Is a more efficient symbol code possible? How efficient could a code with ternary outputs be if it need not be a symbol code? How could a near-optimal code be achieved?

3. Consider the ensemble of 20 fair coin toss outcomes in $\{\mathtt{T}, \mathtt{H}\}^{20}$. What is the probability of obtaining 'TTHTTHHTTTHTTHTHHTTT', and what is the probability of twenty heads. Why is the latter outcome likely to be treated with more suspicion than the former?

4. Consider the ensemble of strings of $N = 10^6$ independent Bernoulli outcomes with probability of a 1 equal to $f = 0.01$.
   (a) Which of these strings obtains the shortest encoding under arithmetic coding? What is the length of this shortest encoding (1sf)? What is the length of an average encoding (1sf)? (For reference: $H_2(f) = 0.08$ bits (1sf), $\log_2 1/f = 7$ bits (1sf), $\log_2 1/(1-f) = 0.01$ bits (1sf).)
   (b) Summarize why the proof of the source coding theorem only uses a fixed width encoding, while a near-optimal encoder uses such dramatically different encoding lengths for different files.

5. Your compressor contains a probabilistic model for the next symbol $P(x_{N+1} \mid m)$. Your beliefs about the model given the file so far are summarized by a posterior over models $P(m \mid \mathbf{x})$. Write down an expression for the predictive distribution over x_{N+1}. For numerical reasons, the functions in your program actually report $\log_e P(x_{N+1} \mid m)$ and $\log_e P(m \mid \mathbf{x})$. How would you implement your prediction rule using standard floating point arithmetic, while avoiding underflow?

6. A coin is tossed repeatedly until it comes up heads. The number of coin tosses required to get a head (from 1 to infinity) is recorded. The experiment is then repeated N times, and the sum of the values recorded. Does the Central Limit Theorem (CLT) apply to this situation, for large N? If so, what does it say, and use it to sketch the distribution over results from this procedure. If the CLT does not apply, explain why.

7. Sketch the binary entropy function, measured in bits. Show that this function is concave (covered much later in the course than H_2 itself).

8. A biased coin, with probability of coming up tails equal to f, is spun repeatedly until it comes up heads. The number of coin spins required to get a head (from 1 to infinity) is recorded. What is the entropy of this ensemble? You can use entropy decomposition to find this quickly in terms of the binary entropy function. You could numerically check your answer by truncating the ensemble at some number of spins, and computing the entropy by brute force.

9. Do you understand the source coding theorem? Could you explain individual parts of it and summarize the result? Why is the *log*-probability used as the property to define the typical set? Why not average the probabilities of each outcome in a block of length N to define the typical set?

10. Show that for any ensemble a binary symbol code exists that encodes to within one bit/symbol of the entropy on average. How does one construct an optimal symbol code? Compared to the optimal compressed filesize, what is the worst-case (over ensembles) percentage increase in filesize if forced to use an optimal symbol code? How might this increase in filesize be mitigated a) by representing the ensemble differently; b) by using something other than a symbol code?

11. Give an interpretation of Gibbs inequality: $D_{\mathrm{KL}}(p||q) \geq 0$

12. Explain, with a diagram, Jensen's inequality for the special case of an ensemble with two outcomes.

13. Some machine learning papers report "root mean square error", others report "mean absolute error". Use Jensen's inequality to find out which error measure tends to be bigger. Does this result make sense; can you check it with intuition and/or a special case?

14. Describe how the size and position of intervals corresponding to source strings are defined in arithmetic coding. What sources of information can the probabilistic model use when making predictions, and (depending on your answer) what parts of a model would both the sender and receiver need to have in their arithmetic coding software from the beginning? For example if using PPM to compress English text.

15. Can you explain how a Dirichlet predictor for each possible context would be applied to the lecture notes' "Hi Mom" image with a context window of 3 pixels? You could try computing the information content of the image under that model to see if you agree with the claim in the slides. (You could easily be a few bits out if you make different arbitrary choices to me, for example dealing with the boundaries differently.)

16. Invent a communication channel, and compute its mutual information in more than one way, checking that you get the same answer. For example: a biased coin has probability of heads equal to $1/4$ ($x = 0$) or $3/4$ ($x = 1$). You observe y the outcome of 3 spins. Exam questions have channels painfully constructed so that they can be worked out without a computer or a calculator, but arbitrary examples like this one may be messier. Extreme cases may be simpler though. In the example, what is the mutual information in the limit of observing an infinite number of spins? If you spin only once does this correspond to a standard channel? Write down its mutual information too.

17. Review the proof that the mutual information of the noisy typewriter channel can be maximized for a uniform input distribution. Adapt that to explain why the ternary confusion channel must have optimal input distribution of the form $[p/2, 1-p, p/2]$, (without first proving that $p=1$). I'm not expecting you to remember precisely what I mean by noisy typewriter and ternary confusion... you can find the definitions in the notes.

18. Summarize the result of the noisy channel coding theorem. Explain why large blocks of channel uses be used to obtain vanishingly small error probabilities for channels such as the binary symmetric channel.

19. Explain why there are error patterns with N/3 bit flips (say) that will result in decoding errors if binary code words are chosen at random. Give a channel for which error patterns of $N/3$ bit flips across a block are typical. Does the noisy channel coding theorem say we can have error-less communication for this channel? If so at what rate, or if not, what finite error probability must we tolerate?

20. State why low-density parity check codes are 'low-density', and briefly on what principles decoding operates.