CS 37/237
Winter 2025
Information Theory in CS

Homework 3
Due Jan 30, 2025 by 10:00 am EST

Prof. Amit Chakrabarti
Computer Science
Dartmouth

General Instructions: ***You may work on this homework in groups of size at most two.*** Please turn in solutions to the problems marked "graded." The rest are either "quiz" questions to test your own understanding, or "ungraded" problems, which I think are instructive and which you can ask me about during an office hour or X-hour. I do recommend solving every problem—graded or not—to your satisfaction to solidify your understanding of the material.

Sources: I urge you *not to consult* any sources besides what is provided on the course website and the three textbooks mentioned on that website. Here, "sources" includes the internet, large language models (LLMs) and other forms of Generative AI (GenAI). This is for the sake of your own learning. You *are allowed* to refer to books or resources of your choice if you need to brush up on your mathematics background, though keep in mind that as of this writing GenAI is not fully reliable for this purpose. *Whatever sources you do consult must be thoroughly documented and clearly and prominently acknowledged in your submission*, e.g.,

- a detailed citation to any textbooks you used, mentioning specific chapter(s) or page(s);
- the full URLs of any websites you consulted plus a clear description of what you found on those websites;
- the precise prompt(s) you used to interact with an LLM/GenAI tool.

It is okay to ask clarifying questions on Ed Discussion as you work on these problems, but not okay to ask for (or provide) big hints or complete solutions to the problems.

Honor Principle: By submitting a solution to a homework problem, you—i.e., "both of you" if you're doing a group submission—represent that what you are submitting is *your own* work, perhaps produced after consulting certain sources that you have *fully cited*, as explained above.

Submission Instructions: Please submit on Gradescope by the stated deadline. ***If you worked with a partner, please start your submission with a brief statement (at most three sentences) explaining how exactly you collaborated, i.e., who did what.***

---

1. QUIZ Give the LZ78 parsing and subsequent encoding of the string `AAAAAABBABABABAAAAABBABAB`. The alphabet is assumed to be $\{\mathtt{A},\mathtt{B}\}$ and should ultimately be mapped to $\{0,1\}$ using $\mathtt{A} \mapsto 0$, $\mathtt{B} \mapsto 1$.

2. QUIZ The interval update formulas for arithmetic coding look like they should be two lines of code in most programming languages: after all, we're only doing additions, subtractions, and multiplications. In reality, an implementation requires considerably more complicated logic. Why?

3. GRADED A length-12 string over the alphabet $\{\mathtt{A},\mathtt{B},\mathtt{C},\mathtt{D},\mathtt{E}\}$ has been encoded using arithmetic coding under the probability model
$$p(\mathtt{A}) = 0.25\,,\quad p(\mathtt{B}) = 0.4\,,\quad p(\mathtt{C}) = 0.15\,,\quad p(\mathtt{D}) = 0.1\,,\quad p(\mathtt{E}) = 0.1\,,$$
producing the codeword "01010 10110 11101 10111 00101" (the spaces between the bits are for readability only). What was the original string?

   You will probably want to write some code to solve this. Please keep it simple and don't go for a full-fledged arithmetic decoder that handles long codewords.

4. GRADED Interpreting a finite-length bit string as a subinterval of $[0,1)$ as discussed in the lecture—and used in the previous problem—gives us what was shall call a *dyadic interval*. Prove that every interval $[u,v) \subseteq [0,1)$ contains a dyadic interval whose length is at least $(v-u)/4$. Prove that the constant '4' cannot be improved to anything smaller.

   Relate this constant 4 to one of the theoretical guarantees of arithmetic coding.

5. UNGRADED A coin that shows heads with probability $p > 0$ is flipped repeatedly until it lands `HEADS` for the first time, on the Zth flip. Show that $\mathrm{H}(Z) < \infty$ and obtain a formula for $\mathrm{H}(Z)$ in terms of p.

6. GRADED Let X be a positive-integer-valued random variable with $\mathbb{E}\,X = \mu < \infty$. Prove the following fact, which was used in our analysis of the Lempel–Ziv (LZ78) algorithm.
$$\mathrm{H}(X) \le \mu \log \mu - (\mu - 1)\log(\mu - 1)\,.$$
   Since this is an information theory course, I will be most impressed if you prove this using non-negativity of KL-divergence! An alternate method is to consider this as a constrained optimization problem and use Lagrange multipliers.

Homework 3
Due Jan 30, 2025 by 10:00 am EST

7. GRADED In our analysis of LZ78, one step requires an upper bound on the entropy of the pmf $(c_1/c, \ldots, c_L/c)$, where c_ℓ is the number of phrases of length ℓ, L is the maximum length of a phrase, and $c = \sum_\ell c_\ell$. Clearly, this entropy is at most $\log L$, but that's not the upper bound we used. Why not? This problem explores the issue.

   Let $L_{\min}(n)$ and $L_{\max}(n)$ denote the minimum and maximum (resp.) possible value of L for a given message length n.

   7.1. Find a good $O(\cdot)$-style upper bound on $L_{\min}(n)$.

   7.2. Find a good $\Omega(\cdot)$-style lower bound on $L_{\max}(n)$.

   7.3. Based on your lower bound for $L_{\max}(n)$, explain how using $\log L$ as the upper bound on the entropy of the pmf would affect the results we derived for LZ78's compression guarantees. Stick to the i.i.d. case, for which we saw a detailed proof in class.

8. UNGRADED Alice wants to send Bob an integer N using a prefix code over $\{0,1\}$—presumably, N is part of some longer message. The catch is that Bob doesn't know how big N is, so he doesn't know in advance how many bits Alice will use to describe N. Design a suitable prefix code that ensures Alice will send only $\log N + o(\log N)$ bits to describe N.

   Such a scheme is an essential part of several data compression algorithms, including Lempel–Ziv variants.

9. UNGRADED Read up about Tunstall coding. What relations do you notice between it and Huffman coding? What is one advantage of Tunstall coding over Huffman coding?

10. GRADED We return to symbol codes for this problem. Let the message alphabet be $\mathscr{X} = [m]$. Consider the following construction (due to Fano) of a prefix code for a source with pmf $\boldsymbol{p} \in \Delta^m$. Suppose $p_1 \ge p_2 \ge \cdots \ge p_m$. Choose k to minimize the quantity

$$\left| \sum_{i=1}^{k} p_i - \sum_{i=k+1}^{m} p_i \right|$$

   thereby partitioning the alphabet $[m]$ into subsets $[k]$ and $[m] \smallsetminus [k]$. Recursively design prefix codes for these subsets and combine them by using a leading bit to distinguish between the two subsets.

   Prove that the mean length of the code C thus obtained satisfies $L(C) \le \mathrm{H}(X) + 1$.

   Hint: Here is a suggested proof outline.

   (a) Let T be binary tree representing the above code and let $V^+(T)$ be the set of its internal nodes. Find a natural correspondence between each internal node and a consecutive set of alphabet symbols $\{a, a+1, \ldots, b\} \subseteq [m]$; we can then denote that internal node as $[a:b]$. Similarly, we can denote the leaf corresponding to symbol a as $[a:a]$.

   (b) Prove that

$$L(C) = \sum_{[a:b] \in V^+(T)} p_{[a:b]}\,, \qquad \text{where } p_{[a:b]} := \sum_{i=a}^{b} p_i\,.$$

   (c) Prove that

$$\mathrm{H}(X) = \sum_{[a:b] \in V^+(T)} p_{[a:b]} H_2\!\left(\frac{p_{[a:c(a,b)]}}{p_{[a:b]}}\right),$$

   where $c(a,b)$ is the number c such that the children of $[a:b]$ are $[a:c]$ and $[c+1:b]$, and H_2 is the binary entropy function.

   (d) Prove the inequality $H_2(x) \ge 2x$ for all $x \in [0, \frac{1}{2}]$. Then use it to deduce that

$$L(C) - \mathrm{H}(X) \le \sum_{[a:b] \in V^+(T)} \left| p_{[a:c(a,b)]} - p_{[c(a,b)+1:b]} \right| \le \sum_{[a:b] \in V^+(T)} p_{c(a,b)}\,,$$

   where the second step will make use of the minimizing ("balancing") property of the bifurcation point $c(a,b)$.

   (e) Based on all your work above, prove that $L(C) \le \mathrm{H}(X) + 1$.