AIM Qualifying Review Exam: Probability and Discrete Mathematics

August 16, 2024

There are five (5) problems in this examination.

There should be sufficient room in this booklet for all your work. But if you use other sheets of paper, be sure to mark them clearly and staple them to the booklet.

Problem 1

The complete undirected graph, K_n, on n vertices has edges between all unordered pairs of distinct vertices.

(a) Suppose each edge of K_n is colored either maize or blue with equal probability. In terms of j and n, what is the probability that some j-clique is monochromatic (all edges are the same color)?

(b) Explain why this shows that, for even somewhat large n as a function of j, there is a coloring of K_n in which no j-clique is monochromatic.

(c) Estimate the threshold value of such n to within a constant factor when $1 \ll j \ll n$, using $\binom{n}{r} \approx \frac{n^r}{r!}$ and $r! \approx \sqrt{2\pi r}(r/e)^r$, that hold in this regime of j and n.

Solution

Ross Example 4m, pp. 91–92.

(a) For j given and n yet undetermined, each j clique is monochromatic maize with probability $2^{-\binom{j}{2}}$ and monochromatic blue with equal and exclusive probability, so the probability of a monotonic j-clique is $2^{1-\binom{j}{2}}$. The number of such j-cliques is $\binom{n}{j}$. The probability that some j-clique is monotonic is at most $2^{1-\binom{j}{2}}\binom{n}{j}$.

(b) Continuing, that probability is less than 1 unless n is too large. For not-too-large n, some coloring results in no monochromatic j-clique.

(c) We need

$$2^{-j^2/2}\frac{n^j}{\sqrt{2\pi j}(j/e)^j} \approx 2^{-j^2/2}\frac{n^j}{j!} \approx 2^{1-\binom{j}{2}}\binom{n}{j} < 1.$$

The $\sqrt{2\pi j}$ will go away and the factor e is constant. Isolating n, we get the threshold for n proportional to $j2^{j/2}$.

which

Mathematical concepts: discrete probability; probabilistic method; combinatorial interpretation of binomial symbols

Problem 2

Suppose that if you are s minutes early for an appointment you incur cost cs and if you are s minutes late then you incur ks. Suppose the travel time is a continuous random variable with density f (so $\int_{-\infty}^{+\infty} f(t)\,dt = 1$). Determine (in terms of f and/or $F(s) = \int_{-\infty}^{s} f(t)dt$ and their inverses) the time to leave to minimize your expected cost.

Solution

Ross example 2d, p. 193.

If you leave t minutes before the appointment, the cost is

$$C_t(x) = \begin{cases} c(t-X), & X \leq t; \\ k(X-t), & t \leq X \end{cases}$$

(We are ignoring the zero-probability event that $X = t$.)

So

$$\begin{aligned} E[C_t(X) &= \int_0^\infty C_t(x)f(x)dx \\ &= \int_0^t c(t-x)f(x)dx + \int_t^\infty k(x-t)f(x)dx \\ &= ct\int_0^t f(x)dx - c\int_0^t xf(x)dx + k\int_t^\infty xf(x)dx - kt\int_t^\infty f(x)dx. \end{aligned}$$

We want to minimize this over t. Differentiating with respect to t, we get

$$\begin{aligned} \frac{d}{dt}E[C_t(X)] &= ctf(t) + cF(t) - ctf(t) - ktf(t) + ktf(t) - k[1-F(1)] \\ &= (k+c)F(t) - k. \end{aligned}$$

Setting this to zero, we want t^* making $F(t^*) = \frac{k}{k+c}$. For a sanity check, note that $\frac{k}{k+c}$ is between 0 and 1, behaves properly if $c = 0$ or $k = 0$, and has the expected symmetries under exchange of c with k and of "late" with "early" (i.e., exchange $\int_0^t$ with $\int_t^\infty$ and $(t-x)$ with $(x-t)$—or just use $|x-t|$).

Mathematical concepts: Continuous probability; expectation

Problem 3 A collection of subsets of $\{1, 2, \ldots, n\}$ has the property that each pair of subsets has at least one element in common. Prove that there are at most 2^{n-1} subsets.

Solution

Brualdi 3.4.27, p. 85.

Pair each set with its complement, getting 2^{n-1} pairs. If there are more than 2^{n-1} subsets, then the collection must contain two sets from some pair, i.e., some set along with its complement. Contradiction.

(Note that the situation is realized by taking all sets that contain some fixed element, say, 1.)

Mathematical concepts: pigeonhole principle

Problem 4

Solve the recurrence relation

$$h_n = \begin{cases} 5h_{n-1} - 6h_{n-2} + 1, & n \geq 2 \\ 1, & n = 0, 1. \end{cases}$$

Partial credit awarded for a solution that simplifies the first line to $h_n = 5h_{n-1} - 6h_{n-2}$.

Solution

Brualdi, chap. 7

For the homogeneous part, $h_n - 5h_{n-1} + 6h_{n-2}$ has characteristic equation $x^2 - 5x + 6 = 0$, which has roots $x = 2, 3$. So $c_2 2^n + c_3 3^n$ is a solution, as we can check.

For a particular solution, we can try a constant, A, because the non-homogeneous part, $+1$, is constant. (See also the solution to problem 5.) We get

n	$h(n)$	$A + c_2 2^n + c_3 3^n$
0	1	$A + c_2 + c_3$
1	1	$A + 2c_2 + 3c_3$
2	0	$A + 4c_2 + 9c_3$

This has solution $c_3 = -1/2, c_2 = 1, A = 1/2$. So the recurrence has solution $h_n = \frac{1}{2} + 2^n - \frac{1}{2}3^n$. For the particular solution with $c_2 = c_3 = 0$, we have $\frac{1}{2} = 5\frac{1}{2} - 6\frac{1}{2} + 1$, which checks out. (This is also a way to find $A = \frac{1}{2}$: We have $A = 5A - 6A + 1$, leaving h_0 and h_1 unexamined to be used separately to determine c_2 and c_3.)

We can check at $n = 3$: the formula gives $A + c_2 2^n + c_3 3^n = \frac{1}{2} + 2^3 - \frac{1}{2}3^3 = -5$ while the recurrence gives $h_3 = 5h_2 - 6h_1 + 1 = -5$.

Mathematical concepts: Recurrence relations

Problem 5

An *inversion* in a sequence $(a_1, a_2, a_3, \ldots)$ is a pair (i, j) such that $i < j$ but $a_i > a_j$. For example, in $(10, 30, 40, 20)$, the pair $(2, 4)$, for entries $(30, \ldots, 20)$, is an inversion. For simplicity, assume all sequence elements are distinct in what follows.

(a) For general n, find a sequence of length n with $\Omega(n^2)$ inversions. (That is, the number of inversions is at least proportional to n^2.)

(b) Sketch an algorithm that counts the number of inversions in a sequence of length n and runs in time $O(n \log n)$. Sketch a proof of correctness and runtime. *(Hint: modify the MERGESORT algorithm, that sorts the left and right halves recursively and then merges the two sorted lists.)*

(c) Why can we assume that the sequence length is a power of 2? (Or easily reduce the problem with general sequence lengths to the problem where sequence lengths are a power of 2 -?)

Solution

Kleinberg and Tardos, pp. 223–225.

(a) $(n, n-1, \ldots, 1)$

(b) Recursively sort and count inversions on the left half and right half of the array while simultaneously getting two sorted lists $A = (a_1 < a_2 < \cdots < a_{n/2})$ and $B = (b_1 < b_2 < \cdots < b_{n/2})$. We also need to account for inversions consisting of $a_i \in A$ and $b_j \in B$ with $a_i > b_j$. We do this as we merge: Compare a_1 to b_1, remove the smaller from its list, and output it. If $a_1 < b_1$, note that $a_1 < B$, so a_1 is not involved in any inversion with B and need not be considered further. If $b_1 < a_1$, then $b_1 < A$, and b_1 is in all $n/2 = |A|$ possible inversions with A. Record this in a running count, and note that we need not consider b_1 again.

For the runtime $T(n)$, note that we have reduced a problem of size n to two problems of size $n/2$ plus $O(n)$ overhead (the cost to merge); $T(1)$ is constant. So by the Master Theorem on divide-and-conquer runtimes, the total cost is $O(n \log n)$. In more detail, at the top-level merge, the cost to merge-and-count two lists of size $n/2$ is cn for some constant c. At the stage below that, we are twice merge-and-counting lists of size $n/4$ into lists of size $n/2$, for cost $2 \cdot c\frac{n}{2} = cn$. Similarly, the total cost is cn at each of $\log_2 n$ levels, one level per division by 2 to reduce n to 1. (We may assume the cost to merge-and count a singleton list is just the same c, so, at the bottom level, we have cost c incurred n times.)

Alternatively, prove by induction that $cn \log n$ solves the following recurrence:

$$T(n) = \begin{cases} 2T(n/2) + cn, & n > 1 \\ c, & n = 1. \end{cases}$$

For the inductive step, we have $T(2n) = 2T(n) + 2cn$ from the recurrence, which is $2cn \log n + 2cn$ by induction, or $2cn(\log n + 1) = 2cn \log(2n) = c(2n)\log(2n)$.

The recurrence could also be solved after rewriting by taking $n = 2^m$ and $t_m = T(2^m)$:

$$t_m = \begin{cases} 2t_{m-1} + c2^m, & m > 0 \\ c, & m = 0, \end{cases}$$

which has solution $t_m = cm2^m$.

Finally, the first line can be rewritten as an inhomogeneous first-order linear difference equation $\Delta t - t = 2c2^m$ (or with m replaced by $m\pm1$ in some formulations). Akin to solving $y' + py = q$ as $y = e^{-\int p} \int q e^{\int p}$, we can solve the above as $t_{m+1} = 2^{-\sum_{i=1}^{m+1} -1} \sum 2c2^m 2^{\sum -1} = 2^{m+1} \sum c = c(m+1)2^{(m+1)}$. (It's tricky to get the indices right in the discrete setting, and the above may be off by 1 somewhere. But this is a legitimate approach.)

(c) Given a sequence of length n with $2^m < n < 2^{m+1}$, we can append $2^{m+1} - 2^m < n$ copies of the symbol $+\infty$ to the end. This gives a new sequence whose length is a power of 2, with no new inversions, and such that the new length is at most twice the old length and so won't matter in a runtime of the form $O(n \log n)$.

Mathematical concepts: divide and conquer, sorting, inversions