Problem Set 1 CS265, Winter 2021-2022

Due: January 12 (Wednesday) at 11am (Pacific Time)

Please follow the homework policies on the course website.

---

1. **(5 pt.) [Perfect Matchings.]**

   Let $G = (V, E)$ be a *bipartite graph* with n vertices on each side. A *perfect matching* in G is a list of edges $M \subset E$ so that every vertex in V is incident to exactly one edge.

   For example, here is a bipartite graph G (on the left), and a perfect matching in G (shown in **bold** on the right):

   Your goal is to determine if the graph G has a perfect matching.

   There are efficient deterministic algorithms for this problem, but in this problem you'll work out a simple randomized one.[1]

   (a) **(2 pt.)** Recall that the *determinant* of an $n \times n$ matrix A is given by

$$\det(A) = \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} A_{i,\sigma(i)},$$

   where the sum is over all permutations $\sigma : \{1, \ldots, n\} \to \{1, \ldots, n\}$, and where $\operatorname{sgn}(\sigma)$ denotes the signature[2] of the permutation σ. (For example, if $n = 3$, then the function $\sigma : \{1, 2, 3\} \to \{1, 2, 3\}$ that maps $1 \mapsto 2$, $2 \mapsto 1$, $3 \mapsto 3$ is a permutation in S_3. The signature of σ happens to be -1, although as noted in the footnote, if you haven't seen this definition before, don't worry about it).

   Let A be the $n \times n$ matrix so that

$$A_{ij} = \begin{cases} x_{ij} & (i, j) \in E \\ 0 & \text{otherwise} \end{cases}$$

   where the x_{ij} are variables, and $(i, j) \in E$ if and only if the i-th vertex on the left and the j-th vertex on the right are connected by an edge in G. Notice that $\det(A)$ is a multivariate polynomial in the variables x_{ij}.

   Explain why $\det(A)$ is not identically zero if and only if G has a perfect matching.

[1] This randomized algorithm has the advantage that (a) it generalizes to all graphs (not necessarily bipartite), and (b) it can be parallelized easily. Moreover, it's possible to generalize it to actually recover the perfect matching (and not just decide if there is one or not).

[2] The *signature* of a permutation is defined as -1 if the permutation can be written as an odd number of transpositions, and $+1$ otherwise. **The exact definition isn't important** to this problem, all you need to know is that it's either ± 1 in a way that depends on σ.

(b) **(3 pt.)** Use the part above to develop a randomized algorithm for deciding whether or not there is a perfect matching. Your algorithm should run in $O(n^3)$ operations. If G has no perfect matching, your algorithm should return "There is no perfect matching" with probability 1. If G has a perfect matching, your algorithm should return "There is a perfect matching" with probability at least 0.9.

You should clearly state your algorithm and explain why it has the desired properties.

[**HINT:** *You may use the fact that one can compute the determinant of a matrix* $A \in \mathbb{R}^{n \times n}$ *in* $O(n^3)$ *operations.*]

(c) **(0 pt.) [Optional: this won't be graded.]** Extend your algorithm to actually *return* a perfect matching. And/or, extend your algorithm to non-bipartite graphs. As a hint, consider the matrix

$$A = \begin{cases} x_{ij} & \{i,j\} \in E \text{ and } i < j \\ -x_{ji} & \{i,j\} \in E \text{ and } i \geq j \\ 0 & \text{else} \end{cases}$$

2. **(8 pt.)** Suppose you are given a fair coin (ie it lands heads/tails with probability 1/2) and want to use it to "simulate" a coin that lands heads with probability exactly 1/3. Specifically, you will design an algorithm whose only access to randomness is by flipping the fair coin (repeatedly, if desired), and your algorithm should return "heads" with probability exactly 1/3 and "tails" with probability exactly 2/3.

   (a) **(3 pt.)** Prove that it is *impossible* to solve the above problem if the algorithm is only allowed to flip the fair coin at most $1,000,000,000$ times. [Hint: read the next two parts of the problem first...]

   (b) **(3 pt.)** Design an algorithm for the above task that flips the fair coin a finite number of times *in expectation.*

   (c) **(2 pt.)** Show that for *any* value v in the interval $[0,1]$, there is an algorithm that flips a fair at most 2 times in expectation, and outputs "heads" with probability v and "tails" with probability $1-v$. [Hint: think about representing the desired probability in its binary representation.] Note: if you do this part, you can leave part (b) blank and still get full credit for that part.

3. **(14 pt.)** Suppose we are in the midst of a pandemic called COVID, and are running short on diagnostic tests. Let n denote the total number of students who we are hoping to test, and suppose exactly k of these n students do indeed have COVID. For the sake of simplicity, suppose each test is perfect, in that if it is given a sample with any amount of COVID, then it will say "positive", otherwise it will say "negative". [Also, we'll assume that there is no new transmission of COVID–students are either "positive" or "negative" and their status is fixed.] Consider the following two testing options:

   Strategy 1: test all n students, using n tests. It will be expensive, but worth it for the peace-of-mind of the students.

   Group testing: here the idea is to run tests on *mixtures* of samples—given a drop of mucus from several students, mix it up and run the test on the mixture, which will come up positive if any of the students have COVID.

Strategy 2: divide the n students into n/m disjoint groups of size m, at random. (Suppose for convenience that m divides n). Then test each of the groups, using one test per group. If any group tests positive, re-test each student in the group to identify the positive students.

(a) **(4 pt.)** In terms of n, m, k, what is the expected number of groups that will test positive in Strategy 2? It's okay to leave your answer in terms of some binomial coefficients or something like that.

(b) **(2 pt.)** In terms of n, m, k, what is the expected total number of tests that we will need in Strategy 2? Again, it's okay to leave your answer in terms of some binomial coefficients or something like that.

(c) **(2 pt.)** Suppose $n = 10,000$ and $k = 100$. For the optimal choice of m, what is the expected number of tests? Please state the optimal value of m, and the expected number of tests. [To find the optimal value of m, feel free to write a quick a quick script that checks each value of m by brute-force. Full credit will be given for answers that are sufficiently close to optimal so don't worry about off-by-one type of issues.]

(d) **(6 pt.)** Consider the following parameter regimes, where we think of $n \to \infty$. In which of these parameter regimes is Strategy 2 better than Strategy 1, in terms of expected number of tests as $n \to \infty$?

Below, the "little-oh" notation $f(n) = o(g(n))$ means that $f(n)/g(n) \to 0$ as $n \to \infty$; the "little-omega" notation $f(n) = \omega(g(n))$ means that $f(n)/g(n) \to \infty$ as $n \to \infty$; and all asymptotic notation is with respect to the limit $n \to \infty$.

i. $k = \Theta(1)$, $\omega(1) = m = o(\sqrt{n})$.
ii. $k = \Theta(n^{5/6})$, $m = \Theta(n^{1/3})$
iii. $2 \leq m = \Theta(1)$, $k = \Theta(\sqrt{n})$.

[**HINT:** *At this point, you'll probably want to simplify any binomial coefficients you have lying around. There are a number of ways to do this.*[3] *You may want to use the fact (which follows from Stirling's approximation) that for* $\omega(1) = k = o(\sqrt{n})$,

$$\binom{n}{k} = (1 + o(1)) \frac{1}{\sqrt{2\pi k}} \left(\frac{ne}{k}\right)^k.$$

It might also help to remember that $e^{-x} = 1 - x + O(x^2)$ *as* $x \to 0$. *Part of this problem is to get really comfortable with these sorts of approximations!*]

(e) **(0 pt.)[Optional: this won't be graded.]** Consider the following curious strategy that only requires *one* round of testing, and will hopefully identify a large set of "negative" students, and a small (ie $O(k)$) set of "potentially positive" students:

> Strategy 3: Create t groups. Put a sample from student i into group j with probability p, independently for each (i, j). (Note that one student might contribute samples to several groups). For each student, if it participated in at least one group that tested negative, then the student is cleared as "negative". Otherwise, declare the student as "potentially positive".

[3] Here is a delightful—and very useful—note by Shagnik Das for more about various approximations of binomial coefficients: `http://page.mi.fu-berlin.de/shagnik/notes/binomials.pdf`

i. In terms of p, k, n, what is the expected number of false positives (that is, students that are negative but labeled as "potentially positive" in Strategy 3? What is the expected number of false negatives (positive students labeled as "negative")?

ii. Suppose that $p = 1/k$ and $k \geq 2$. Say that we want to designate a set of dorm rooms for only "negative" students, and hope to house at least $n - 2k$ students in these rooms, and under no circumstances do we want to put a positive student in one of these dorms. How many tests does Strategy 3 use? How does this compare to Strategy 1?