6.5240 Midterm Solutions Fall 2024

1. **(8 points.) Testing the monotonicity of a list – the case of bits:** Given a Boolean function $f : [n] \to \{0, 1\}$ and parameter $\varepsilon \in (0, 1)$, present an algorithm that makes $1/\text{poly}(\varepsilon)$ queries to f, and has the following behavior:

   - If f is monotone, then the algorithm always outputs "pass."
   - If f is ε-far from monotone, then the algorithm outputs "fail" with probability at least 3/4.

   Here by "ε-far from monotone" we mean that the value of f need only be changed on at most εn inputs in order to make it monotone.

   For a bonus point, get a query complexity of $O(\frac{1}{\varepsilon} \cdot \log(1/\varepsilon))$.

   Solution

   There are many solutions achieving various $\text{poly}(\varepsilon^{-1})$ query complexities; here we include two that are simple and linear/nearly linear.

   A $O(1/\varepsilon)$-query algorithm:

   - Sample uniformly from $[n]$ $10/\varepsilon$ times. Query the function at the sampled points.
   - Reject if there is any violation of monotonicity among the samples, accept otherwise.

   Query complexity is $10/\varepsilon$.

   If the function is monotone, the test accepts. If the function is ε-far from monotone, there must be matching of size εn in the violation graph, which means there are at least εn pairs of a 1 and a 0 such that the 1 precedes its matched 0. Consider the first $\varepsilon n/2$ 1's in this matching. Since they all precede the last $\varepsilon n/2$ of the matched 1's, they must all precede the last $\varepsilon n/2$ of the matched 0's as well. Thus, if any of the first $\varepsilon n/2$ matched 1's and any of the last $\varepsilon n/2$ matched 0's is sampled, the test finds a violation and rejects. The probability of missing all the 1's is

   $$\leq (1 - \varepsilon/2)^{10/\varepsilon} \leq e^{-5}$$

   and so is the probability of missing all the 0's, so the total failure probability is at most $2e^{-5}$.

   A $\frac{1}{\varepsilon}\log(1/\varepsilon)$-query algorithm:

- Sample uniformly from $[n]$ independently $\frac{10\ln(2/\varepsilon)}{\varepsilon}$ times. Query the function at the sampled points.
- Partition $[n]$ into $\lceil 2/\varepsilon \rceil$ blocks of size $\lfloor \varepsilon n/2 \rfloor$ (except for the last block, which may be smaller).
- Check that there is a unique "transition block" between the 0's and the 1's; i.e. there is at most one block t containing both 0- and 1-samples, every block below t has only 0-samples, and every block above t has only 1-samples. Accept if this is the case and reject otherwise.

Query complexity is $10\ln(2/\varepsilon)/\varepsilon$.

If the function is monotone, then the check for a unique transition block must pass. We will then argue that if the test passes, then the function is ε-close to monotone.

Suppose f is ε-far from all monotone lists with true transition point in block t. There are at most $\varepsilon n/2$ points inside the block, so there must be at least $\varepsilon n/2$ points that differ from this list that are outside of this block (i.e. 1's before block t or 0's after block t). The probability that none of these points occur in the sample is at most

$$(1-\varepsilon/2)^{10\ln(2/\varepsilon)/\varepsilon} = (1-\varepsilon/2)^{2/\varepsilon \cdot 5\ln(2/\varepsilon)} \leq (2/\varepsilon)^{-5}.$$

Union bounding over all $\lceil 2/\varepsilon \rceil$ possible transition blocks, we achieve a failure probability of $O((\varepsilon/2)^4)$ which is certainly smaller than $1/4$ for any $\varepsilon \leq 1$.

2. **(8 points) Lower bounds for estimating MST weight:** Give a lower bound on computing a multiplicative estimate on the MST of a graph G in adjacency list representation (i.e. neighbor queries, degree queries, and edge weight queries in one step): Give two distributions over graphs of degree at most d and weights in the range $\{1, \ldots, w\}$ (for $w = o(n)$) such that

   a) graphs in one distribution have an MST weight that is at least twice the MST weight of the graphs in the in other distribution

   b) in order to distinguish the two distributions with constant probability of success, one must make at least $\Omega(w)$ queries

   There is a simple construction that works. For a bonus point, show you can get the lower bound to have some nontrivial dependence on d.

Solution

There are many alternative solutions: Essentially any pair of distributions with MST weight difference of 2, and with the only difference being a $O(1/w)$ fraction of edges/vertices works in the non-adaptive setting. (Note that they do not necessarily work in an adaptive setting where an adversary can perform a walk on the graph; many proposals, such as stars or paths, still work.) The complete graph gives a lower bound of $\Omega(n)$ but its degree is not bounded by d.

For no dependence on d: the two distributions are over paths of length n. In one distribution, all weights are 1, so the MST weight is always $n-1$. In the other, randomly sample $\frac{2(n-1)}{w}$ edges; sampled edges are given weight w while the others have weight 1, so the MST weight is at least $2(n-1)$. To distinguish, you must query an edge of weight w. $2/w$-fraction of the edges are of weight w. By a union bound, the probability that any such edge is sampled in $o(w)$ non-adaptive queries is $o(1)$.

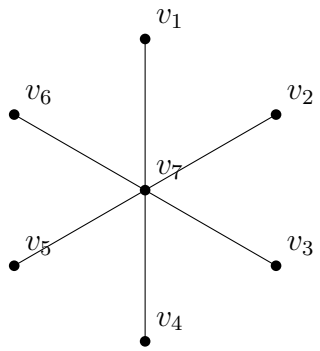
For a linear dependence on d with multigraphs: same as above, but replace each edge with a d-wise multiedge. In the first distribution, give all edges weight w except sample a random member of each superedge to give weight 1, so the MST weight is $n-1$. In the other distribution, sample $\frac{2(n-1)}{w}$ superedges. Sampled superedges have all edges of weight w, while others have one randomly chosen edge to be weight 1. The MST weight is at least $2(n-1)$. To distinguish, you must query a sampled superedge, which requires $\Omega(w)$ queries. But you also must be able to distinguish a sampled superedge from a normal one, which requires observing an edge of weight 1. Edges of weight 1 are a $1/d$-fraction of the members of a superedge and a $\leq 1/d$-fraction of the graph overall, so $\Omega(d)$ queries are required to observe one with constant probability. This gives a lower bound of $\Omega(w+d)$.

To get a dependence of $\Omega(w+d)$ with simple graphs, simply replace each multi-edge by d-cliques of edge weights w. To simulate an edge with weight 1, plant a random Hamiltonian path with edge weights 1 in the clique.

3. **(8 points.) Counting wedges.** In this problem we will create an algorithm for counting the number of *unlabeled wedges* (2-stars, or undirected paths of length 2) in a simple graph $G = (V, E)$ with n vertices and m edges. The access model is that the algorithm can ask for any of the following in one step: a uniform random edge, a uniform random neighbor of a given vertex, and the degree of a given vertex.

   Please see the following figure for an example of how wedges are counted.

Figure 1: In this example, there are $\binom{6}{2} = 15$ wedges, each with v_7 as the center. For instance, (v_6, v_7, v_3) is a wedge (and (v_3, v_7, v_6) is the same wedge), but (v_1, v_7, v_1) is not because it contains repeated vertices.



a) **(1 point.)** Give an expression for the number of *wedges* in G as a function of the degrees.
Solution
Denote d_v as the degree of a vertex $v \in V_G$. $w = \sum_{v \in V_G} \binom{d_v}{2}$.

b) **(1 points.)** Let w be the number of wedges in G. Assume you know some bound $\overline{w}$ that is at least $w/2$ and $\overline{w} \geq 1$. Show that no vertex has degree greater than $d_{ub} := \min(n, 3\sqrt{\overline{w}})$
Solution
By (a), for any particular $u \in V_G$, $w = \sum_{v \in V_G} \binom{d_v}{2} > \binom{d_u}{2}$. We are given that $\overline{w} \geq w/2$, so $2\overline{w} \geq w > d_u(d_u - 1)/2$, which simplifies to $4\overline{w} \geq d_u(d_u - 1)$. When $d_u = 1$, $d_u < 3 \leq 3\sqrt{\overline{w}}$, and if $d_u \geq 2$, $d_u(d_u - 1) \geq d_u^2/2$, so $9\overline{w} > 8\overline{w} \geq d_u^2$.

c) **(3 points.)** Give an algorithm that returns each wedge with probability $1/(m \cdot d_{ub})$ (or else returns $\perp$) and makes $O(1)$ queries.
Hint: Consider first sampling an edge uniformly at random.
Solution
First use a random edge query to sample an edge $e = (u, v)$, then pick u or v uniformly as the center, and rejection-samples one of its neighbors such that any a is returned with probability $1/d_{ub}$ to form the wedge.

The probability that a wedge $u - v - a$ is sampled is indeed $\frac{1}{m} \cdot \frac{1}{2}\frac{2}{d_{ub}}$, depending on whether (u, v) or (v, a) is sampled in the beginning.

d) **(3 points.)** Assuming there is an algorithm satisfying (c), give an algorithm that returns a $(1 \pm \varepsilon)$-approximation of w and makes

$$O\left(\varepsilon^{-2} \cdot \frac{md_{ub}}{\overline{w}}\right)$$

queries, assuming knowledge of $\overline{w}$. Note that because of the bounds on d_{ub}, this complexity is equivalently

$$O\left(\varepsilon^{-2} \min\left(\frac{mn}{\overline{w}}, \frac{m}{\sqrt{\overline{w}}}\right)\right).$$

The failure probability can be a small constant, such as 1/4.

Solution

Note: The assumption that $\bar{w}/w$ is bounded is necessary, otherwise the query complexity could approach 0 as $\bar{w}$ grows! This assumption was missing from the problem statement, so we gave full credit to both the solutions that made the assumption and the solutions that were wrong only because they were trying to evade this assumption. If you think this describes your solution and you didn't get full credit, let us know.

In this solution, we will assume $\bar{w}/w \leq 2$.

Part c) returns some wedge with probability $\frac{w}{2m \cdot d_{ub}}$. Take $k := \frac{20m \cdot d_{ub}}{\overline{w}\varepsilon^2}$ samples and let X be the number of times a wedge is returned. Then

$$E[X] = \frac{10w}{\overline{w}\varepsilon^2}.$$

We'll return $X \cdot \frac{\overline{w}\varepsilon^2}{10}$.
By Chernoff:

$$\Pr[|X - E[X]| \geq \varepsilon E[X]|] \leq 2\exp(-2\varepsilon^2 \cdot E[X]/3) = 2\exp(-6w/\overline{w})$$

Constant failure probability follows from the fact that $w/\bar{w} \geq 2$.

Alternate solution by Chebyshev: Consider $\mathsf{Var}(X)$; X is sum of k Bernoullis of probability p so its variance is $kp(1-p)$. By Chebyshev:

$$\Pr[|X - E[X]| \geq \varepsilon E[X]] \leq \frac{kp(1-p)}{\varepsilon^2 E[X]^2} = \frac{kp(1-p)}{\varepsilon^2 k^2 p^2} \leq \frac{1}{\varepsilon^2 kp}.$$

Constant failure probability is achieved by taking $k = 5/p\varepsilon^2$. Since $p = \frac{w}{md_{ub}}$, it suffices to take $k \geq \frac{5md_{ub}}{\overline{w}\varepsilon^2} \cdot \max(\frac{\overline{w}}{w}) = \frac{10md_{ub}}{\overline{w}\varepsilon^2}$.

Blank page