Query complexity puzzle during a pandemic

Siddhartha Jain

April 2020

1 COVID-19 testing in Germany

This write-up was inspired by the news I read that to scale up testing for COVID-19 (something desperately needed in all countries for suppresing the virus), they were pooling samples and testing them together since they would expect very few to be positive. I assume the fraction of cases they expect to be positive can be estimated. I decided to abstract away the context and think of this as a query complexity problem.

2 Naive problem

Let us formally define the problem as the following. We have a n-bit string to which we have access through an oracle $t : [n] \to \{0, 1\}$. t takes a subset of the indices I and outputs $\vee_{i \in I} x_i$. We want to list all the indices i for which x_i is 1.

This problem clearly has a lower bound of $\Omega(n)$ queries since the input 1^n will always require us to check all indices. To make it interesting, we consider the problem with restricted input. We are given ϵ such that exactly ϵn bits are set to 1 in the input. Now, binary search gives us an algorithm with $O(\epsilon n \log(1/\epsilon))$ queries. We just divide the indices intro groups of size $1/\epsilon$ and on each group we start by querying the entire group and keep reducing the size by half to narrow into positive cases. But the naive lower bound only gives us a lower bound of ϵn. Can the gap be closed? The answer is yes.

Claim 1. *Any algorithm requires $\Omega(\epsilon n \log(1/\epsilon))$ queries.*

Proof. Consider imposing additional structure on the input: every input is equally likely. Since a query only gives us one bit of information, it is enough to lower bound the entropy. Now the entropy of this distribution is $\log \binom{n}{\epsilon n}$. We know $\binom{n}{\epsilon n} \geq (\frac{n}{\epsilon n})^{\epsilon n} = (1/\epsilon)^{\epsilon n}$. Hence the entropy is $\Omega(\epsilon n \log(1/\epsilon))$ which implies we need at least that many queries. □

3 Extensions

In reality, there are two issues not captured by our formalisation.

- The oracle returns the wrong answer with some probability. This error is also two-sided.
- ϵ is not part of the input exactly, but known with some error.

In both cases, we have may to accept error in our output. Here are some suggestions for accepting error. Let I be the true set of indices, and I_{ALG} be the list outputted by our algorithm. Then we can consider two constraints.

- Their symmetric difference is small. Concretely, given δ we want $|I \oplus I_{ALG}| \leq \delta|I| = \delta\epsilon n$.
- Considering we care about false negatives much more than false positives, we may want to ensure that given δ, $|I \setminus I_{ALG}| \leq \delta|I|$.

4 Acknowledgement

Thanks to Parth Mittal for an insightful discussion.