Randomized Algorithms assignment 3

Rasmus Stavenuiter, Jacob Harder

Summary

Chernoff bounds

Let $X_1, X_2, \cdots, X_n$ be Bernoulli random variables (Poisson trials) with $P(X_i = 1) = p_i$. Set $\mu = \sum_{i=1}^{n} p_i$ Then

$$P\left(\sum_{i=1}^{n} X_i > (1+\delta)\mu\right) < \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu} =: F^{+}(\mu, \delta)$$

and

$$P\left(\sum_{i=1}^{n} X_i < (1-\delta)\mu\right) < e^{\mu\delta^2/2} =: F^{-}(\mu, \delta)$$

Permutation routing problem

N processors connected by *wires* (seen as a graph: nodes and edges). Each processor p_i sends one *packet* to a *destination* processor $p_{\pi(i)}$ where $\pi : N \to N$ is a permutation. Only one packet may follow the same edge at each timestep. An algorithm must specify 1: a *route* for every packet (that is a path from source to destination) 2: a *queueing discipline* for which packet goes first when multiple want to travel along the same edge. The algorithm is called *oblivious* if the choice of every route only depend on its own destination.

Theorem 1. *A deterministic oblivious permutation routing algorithm on a network of N nodes each of out-degree d there is an instance requiring $\Omega(\sqrt{N/d})$ steps.*

In a special case where the network is the graph called the *boolean hypercube* we can find a randomized algorithm that satisfies

Theorem 2. *With probability at least $1 - (1/N)$ every packet reaches its destination in $14n$ or fewer steps.*

The boolean hypercube has $N = 2^n$ nodes each connected to exactly n neighbor nodes. Thus the randomized algorithm requires only $14\log_2(N)$ steps with high probability, superior to the deterministic worst case of $\Omega(\sqrt{N/n})$.

Exercise 4.3

If two routes separate that means that at a common kth bit they chose different ways. This again means that they must have different goals, differing on the kth bit. Since bit correction is going from left to right, these routes will never rejoin, since they will differ on this kth bit until their goals are reached.

Exercise 4.7

Given a solution $\hat{x}$ to the linear program relaxation we use the following simple rounding procedure.

$$\overline{x_{i0}} = 1 \;\&\; \overline{x_{i1}} = 0 \iff \widehat{x_{i0}} > \frac{1}{2}$$

There are two cases:

If $\overline{x_{i0}} = 1$ and $\overline{x_{i1}} = 0$ we have $\widehat{x_{i0}} > \frac{1}{2}$ and hence $\overline{x_{i0}} < 2\widehat{x_{i0}}$. Also we trivially have $\overline{x_{i1}} \leq 2\widehat{x_{i1}}$.

If $\overline{x_{i0}} = 0$ and $\overline{x_{i1}} = 1$ then $\widehat{x_{i0}} \leq 1/2$ and hence $\widehat{x_{i1}} \geq 1/2$ and hence $\overline{x_{i1}} \leq 2\widehat{x_{i1}}$ and $\overline{x_{i0}} \leq 2\widehat{x_{i0}}$ trivially holds.

In any case we have $\overline{x_{ij}} \leq 2\widehat{x_{ij}}$.
We then have for any boundary b

$$\begin{aligned}\sum_{i\in T_{b0}} \overline{x_{i0}} + \sum_{i\in T_{b1}} \overline{x_{i1}} &\leq \sum_{i\in T_{b0}} 2\widehat{x_{i0}} + \sum_{i\in T_{b1}} 2\widehat{x_{i1}} \\ &\leq 2\left(\sum_{i\in T_{b0}} \widehat{x_{i0}} + \sum_{i\in T_{b1}} \widehat{x_{i1}}\right) \leq 2\widehat{w} \leq w_o\end{aligned}$$

by definition of w_S this yields $w_S \leq w_o$ as desired.

Problem 4.13

The $0-1$ linear program solving this problem is the following:

- Minimize $||c||_1$
- $c_i \in \{0,1\}$ for $0 \leq i \leq m = \#U$
- $Mc \geq 1$

We will consider the linear program relaxation where $c_i \in [0,1]$, which can be solved in time polynomial in the size of M.

Now given a solution to the linear program relaxation, $\widehat{c}$, we sample a $0-1$ vector c such that $P(c_i = 1) = \min\{1, 8\widehat{c_i}\log n\}$.

For $1 \leq j \leq n$ let M_j denote the jth row of M, and for $1 \leq i \leq m$ define

$$X_i = \min\{M_{ji}, c_i\}$$

Then the X_i are poison trials and $M_j \cdot c = \sum_{i=1}^m X_i$. Furthermore

$$E\left(M_j \cdot c\right) = \sum_{i=1}^m EX_i = 8\log n M_j \cdot \widehat{c} \geq 8\log n$$

since the fact that $\widehat{c}$ is a solution to the linear program relaxation yields $M_j \cdot \widehat{c} \geq 1$.

Now the generalized Chernoff bound yields

$$P\left(\sum_{i=1}^m X_i < (1-\delta)8\log n\right) < \exp\left(-8\log n\frac{\delta^2}{2}\right) = \frac{1}{n^{4\delta^2}}$$

for $0 < \delta \leq 1$.
Now choosing δ sufficiently close to 1 we can ensure that $(1-\delta)\,8\log n \leq 1$ and that $\delta^2 \geq \frac{1}{2}$, and then the above yields

$$P\left(\sum_{i=1}^m X_i = 0\right) = P\left(\sum_{i=1}^m X_i < (1-\delta)8\log n\right) < \frac{1}{n^{4\delta^2}} \leq \frac{1}{n^2}$$

By the above observations we have proven that $P(M_j \cdot c = 0) \leq \frac{1}{n^2}$.
Hence we can now bound the probability that the vector c obtained by the randomized rounding procedure described above is not a set-cover by

$$\begin{aligned}
&P\left(c \text{ is not a set-cover}\right)\\
=&P\left(\cup_{j=1}^{n}\left(M_j \cdot c = 0\right)\right)\\
\leq&\sum_{j=1}^{n} P\left(M_j \cdot c = 0\right)\\
\leq&\sum_{j=1}^{n} \frac{1}{n^2} = \frac{1}{n}
\end{aligned}$$

To check whether c is in fact a set-cover we just have to compute the matrix product Mc and check whether any entry is 0, which is certainly polynomial in the size of M.

Thus we have deviced a Monte Carlo algorithm with expected polynomial running time such that a given output can be verified in polynomial time and the probability of a correct output is at least $1 - \frac{1}{n}$.
Then by exercise 1.3 this can be used to construct a Las Vegas algorithm whose expected running time is certainly polynomial.

To compute bounds on the expected size of the set-cover observe that since $\widehat{c}$ is a solution to the linear program relaxation we certainly have $||\widehat{c}||_1 \leq C(M)$ and hence

$$E||c||_1 \leq \sum_{i=1}^{m} \widehat{c}_i 8 \log n = ||\widehat{c}||_1 8 \log n \leq C(M) 8 \log n$$

as desired.

Problem 4.14

Let us look at the sorting tree of some execution of the *RandQS* algorithm on some set S. Say that at some node the algorithm chooses a *good pivot* provided that it selects an element from its given subset $S' \subset S$ that is results in two subsets (elements of higher resp. lower order) with size no larger than $3/4|S'|$. We claim that the chance of this selection by the algorithm is $\geq 1/2$. To see this note that the number of elements in the *lower* subset L follows a uniform distribution on $\{0, \cdots, n-1\}$, where $n = |S'|$. The number of elements in the *higher* subset H depends entirely on $|L|$ in that $|L| + |H| = n - 1$. The chance that a bad pivot is chosen is therefore

$$\begin{aligned} P(|L| > 3/4n \vee |H| > 3/4n) &= 2P(|L| > 3/4n) \\ &= 2P(|L| \geq \lceil 3/4n \rceil) \\ &= 2\left(\frac{n - 1 - \lceil 3/4n \rceil + 1}{n}\right) \\ &\leq 2\frac{1}{4} = 1/2 \end{aligned}$$

Thus the chance of a good pivot chosen is $\geq 1/2$. Fix an element $x \in S$ and let h denote the length of the path from the root of the tree to x. For each step on this path if a good pivot is chosen the number of remaining elements in the batch containing x is reduced by at least $3/4$. Solving

$$(3/4)^k n \leq 1 \iff k + \log_{3/4}(n) \leq 0 \iff k \geq \log_{4/3}(n)$$

we see that if more than $m = \log_{4/3}(n) = \ln(n)/\ln(4/3)$ good pivots is chosen the height of x has been attained. Now we calculate the chance that less than m good pivots are chosen in $8m$ steps. The choices of pivots are independent and each is good with probability at least $1/2$. By the lower Chernoff bound with $\delta = 3/4$ and $\mu = 1/2m$ the probability of having too few good pivots is less than

$$e^{-\frac{m}{2}\left(\frac{3}{4}\right)^2/2} \leq e^{-\frac{m}{8}} = e^{-\frac{\ln(n)}{\ln(4/3)}} = n^{-\frac{1}{\ln(4/3)}} \leq n^{-3}$$

Thus for any $x \in S$ the probability that x has height greater than $\frac{8}{\ln(4/3)}\ln(n) = h_m$ is less than n^{-3}. By union bounding the chance that the whole tree has height more than h_m is less than n^{-2}. The sum of the heights of all elements equal the number of comparisons of the algorithm. Therefore the number of comparisons is at most $\frac{8}{\ln(4/3)}n\ln(n) < 32n\ln(n)$ with probability $1 - n^{-2}$ proving that *RandQS* is $O(n\ln(n))$ with high probability.