ShanghaiTech University

CS240 Algorithm Design and Analysis

Spring 2023

Problem Set 5

Due: 23:59, May 24, 2023

1. Submit your solutions to the course Blackboard.
2. If you want to submit a handwritten version, scan it clearly.
3. Late homeworks submitted within 24 hours of the due date will be marked down 25%. Homeworks submitted more than 24 hours after the due date will not be accepted unless there is a valid reason, such as a medical or family emergency.
4. You are required to follow ShanghaiTech's academic honesty policies. You are allowed to discuss problems with other students, but you must write up your solutions by yourselves. You are not allowed to copy materials from other students or from online or published resources. Violating academic honesty can result in serious penalties.

Problem 1:

Suppose you want to estimate the fraction f of people who skateboard in ShanghaiTech. Assume that you can select a ShanghaiTech student uniformly at random and determine whether they skateboard. Also, assume you know a lower bound $0 < a < f$. Design a procedure for estimating f by some $\hat{f}$ such that $\Pr[|f - \hat{f}| > \varepsilon f] < \delta$, for any choice of constants $0 < a, \varepsilon, \delta < 1$. What is the smallest number of residents you must query?

Solution:

We may think of asking a resident as flipping a coin with bias $p = f$. Flip the coin N times. If you get k heads, set $\hat{p} = \frac{k}{N}$. Note that k has a binomial distribution with mean $\mu = pN$. Thus, using Chernoff bounds:

$$Pr[|p - \hat{p}| > \varepsilon p] = Pr[|\mu - k| > \varepsilon\mu]$$

$$= Pr[k < (1-\varepsilon)\mu] + Pr[k > (1+\varepsilon)\mu]$$
$$\leq e^{-\frac{\varepsilon^2\mu}{2}} + e^{-\frac{\varepsilon^2\mu}{3}} \leq 2e^{-\frac{\varepsilon^2\mu}{3}} = 2e^{-\frac{\varepsilon^2 pN}{3}}.$$

Set this bound equal to δ, and solve for N to find that $N = \frac{3\ln(2/\delta)}{\varepsilon^2 p}$ trials suffice. Since $p \geq a$ by assumption, certainly $N = \frac{3\ln(2/\delta)}{\varepsilon^2 a}$ trials suffice.

Problem 2:

Suppose that there are n items which need to be placed in some bins. The capacity of each bin is 1. The volumes of the items may not be the same and all volumes are smaller than 1. We want to use the fewest number of bins to place all items. Design a 2-approximation algorithm to solve this problem.

Solution:

For each item, puts it in one of partially packed bins if it can be placed into.If all the partially packed bins can not fit the item, then we use a new bin to place the item.The number of the bins is at most 2 times of the optimal solution.
Proof: Let k$*$ denote the optimal solution. Suppose that the algorithm above returns k bins. Then we can know that at least k-1 bins are more than half full. Since every bin contains more than half, we have $\sum_{i=1}^{k} 1/2 \leq \sum_{i=1}^{k} v_i \leq \sum_{i=1}^{k*} 1$, where v_i denotes the volume of items for bin_i. Therefore $\sum_{i=1}^{k} 1/2 \leq \sum_{i=1}^{k*} 1$, which proves the 2-approximation.

Problem 3:

Consider the following simple model of gambling in the presence of bad odds. At the beginning, your net profit is 0. You play for a sequence of n rounds. In each round, your net profit increases by 1 with probability 1/3, and decreases by 1 with probability 2/3. Show that the expected number of steps for which your net profit is positive can be upper-bounded by an absolute constant, independent of the value of n.

Solution:

Let Y denote the number of steps in which your net profit is positive. Then $Y = Y_1 + Y_2 + .Y_n$,where $Y_k = 1$ if your net profit is positive at step k,and 0 otherwise. Now consider a particular step $k.Y_k = 1$ if and only if you have had more than $k/2$ steps in which your profit increased. Since the expected number of steps in which your profit increased is $k/3$, we can apply the Chernoff bound with $\mu = k/3$and $1 + \delta = 3/2$ to conclude that EY_k is bounded by

$$\left[\frac{e^{1/2}}{(3/2)^{(3/2)}}\right]^{(k/3)} < (.97)^k$$

Thus,

$$EY = \sum_{k=1}^{n} EY_k < \sum_{k=1}^{n} (.97)^k < \frac{1}{1-(.97)} < 34$$

Problem 4:

Given a set of items, where the i'th item has size c_i ($c_i > 0$), we want to place them in as few bins as possible. Each bin has a capacity V, where $V \geq \max_i c_i$. To try to minimize the number of bins used, we take the following steps. We start with one active bin, then iterate through the items one by one and try to place the item into any active bin which can accommodate it. If an item cannot fit into any active bin, we open a new active bin. The algorithm is shown below. Prove that this algorithm is a 2-approximation, i.e. the number of bins it uses is at most twice the minimum possible number of bins required.

Algorithm:

Input : Set of items with sizes $c_1, c_2, \ldots, c_n$
Output: Number of bins used
Sort the items in non-decreasing order of their sizes: $c_1 \leq c_2 \leq \ldots \leq c_n$

```
for i ← 1 to n do
    for j ← 1 to k do
        if item i fits into active bin j then
            place item i in bin j
            break
        end
    end
    else
        open a new active bin k + 1
        k ← k + 1
        place item i in bin k + 1;
    end
end
return number of active bins k
```

Solution:

Algorithm:

We can make an improvement of Algorithm ??. When considering a coin, rather than checking just the last piggy bank, we check all previous piggy banks to see if there is enough volume such that the coin can be stored in. If not, we start a new piggy bank.

Proof:

Suppose OPT and N are the number of piggy banks used by the optimal strategy and our algorithm, respectively.

There is an observation for the output of our algorithm that at least $N-1$ piggy banks are more than half full, i.e., at most one piggy bank is more than half empty. The reason is that if there are more than one piggy banks which are more than half empty, our algorithm can transfer all the coins of one piggy bank to another one, which will reduce the number of piggy banks.

Therefore, we have,

$$\frac{V}{2}(N-1) \leq \sum_{i=1}^{n} c_i \leq OPT \cdot V'$$

where $V' \geq \max_{i=1}^{n} c_i$ is the size of the largest coin.

Simplifying the above inequality, we get

$$N-1 \leq \frac{2\sum_{i=1}^{n} c_i}{V} \leq 2\frac{OPT \cdot V'}{V} = 2OPT$$

Therefore, our algorithm is a 2-approximation, i.e., it uses at most twice the minimum possible number of piggy banks.

Problem 5:

Suppose that for some decision problem, we have an algorithm which on any instance computes the correct answer with probability at least 4/5. We wish to reduce the probability of error by running the algorithm n times on the same input using independent randomness between trials and taking the most common result. Using Chernoff bounds, give an upper bound on the probability that this new algorithm produces an incorrect result.

Solution:

Let X_i be the indicator of the i-th iteration. $X_i = 1$ if the i-th iteration computes the correct answer; $X_i = 0$ if the i-th iteration computes the wrong answer. $X = \sum_{i=1}^{n} X_i$, $E(X) = n * E(X_i) = 4n/5$ According to Chernoff bounds, $P(X \leq n/2) = P(E[X] \leq (1 - 3/8))e^{-[-4n/5*(3/8)^2]/2} = e^{9n/160}$.