Algorithm Design
ShanghaiTech University
Spring 2023
Rui Fan

Homework 5

Zhijie Liu (2022233365)

Problem 1

Solution:
We first denote an indicator variable X_i representing whether or not the i-th selected student skateboarding. Where 1 indicates the student skateboard and 0 indicates the student does not. Thus, we can get $Pr[X_i = 1] = f$ and $Pr[X_i = 0] = 1 - f$. X_i is a random variable with a Bernoulli distribution.
Assume that we sample n students, then the sum $X = \sum_{i=1}^{n} X_i$ represents the total number of students skateboarding in the n students. Thus, we have $\hat{f} = \frac{X}{n}$. It is notable that all X_i are independent. $\mu = E[X] = nf$.
We need to estimate f by some $\hat{f}$ such that $Pr[|f - \hat{f}| > \varepsilon f] < \delta$, for any choice of constants $0 < a, \varepsilon, \delta < 1$.
So using Chernoff bound, we can get,
$Pr(|X - \mu| \geq \gamma\mu) \leq 2e^{-\mu\frac{\gamma^2}{3}}$, $0 < \gamma < 1$
we can also get,
$Pr(|X - nf| \geq \gamma nf) \leq 2e^{-nf\frac{\gamma^2}{3}}$, $0 < \gamma < 1$
$\equiv Pr(|\hat{f} - f| \geq \gamma f) \leq 2e^{-nf\frac{\gamma^2}{3}}$, $0 < \gamma < 1$
$\equiv Pr(|f - \hat{f}| \geq \gamma f) \leq 2e^{-nf\frac{\gamma^2}{3}}$, $0 < \gamma < 1$
We let $\varepsilon = \gamma$ and get,
$Pr(|f - \hat{f}| \geq \varepsilon f) \leq 2e^{-nf\frac{\varepsilon^2}{3}}$, $0 < \varepsilon < 1$
Thus, we must ensure that $2e^{-nf\frac{\varepsilon^2}{3}} \leq \delta$.
So, solving n, we can get,
$2e^{-nf\frac{\varepsilon^2}{3}} \leq \delta$
$\equiv \ln(2e^{-nf\frac{\varepsilon^2}{3}}) \leq \ln\delta$
$\equiv \ln 2 + \ln e^{-nf\frac{\varepsilon^2}{3}} \leq \ln\delta$
$\equiv \ln e^{-nf\frac{\varepsilon^2}{3}} \leq \ln\delta - \ln 2$
$\equiv -nf\frac{\varepsilon^2}{3} \leq \ln\delta - \ln 2$
$\equiv nf\frac{\varepsilon^2}{3} \geq \ln 2 - \ln\delta$
$\equiv nf \geq \frac{3}{\varepsilon^2}\ln\frac{2}{\delta}$
$\equiv n \geq \frac{3}{\varepsilon^2 f}\ln\frac{2}{\delta}$
The above inequality means that the condition of the number n of students we need to sample to estimate f by some $\hat{f}$ such that $Pr[|f - \hat{f}| > \varepsilon f] < \delta$.
However, f is unknown, we need something to replace it. According to the problem description, we can know that we know a lower bound $0 < a < f$. Thus, we can take a

as a conservative estimate of f to let us sample enough students. Replacing f with a, we can get,
$n \geq \frac{3}{\varepsilon^2 a} \ln \frac{2}{\delta} > \frac{3}{\varepsilon^2 f} \ln \frac{2}{\delta}$
Thus, under the conditions given, the smallest number of residents we must query is $\lceil \frac{3}{\varepsilon^2 a} \ln \frac{2}{\delta} \rceil$.
Procedure:
1. Set up a, ε and δ.
2. Calculate and determine the number n of students we need to query by using the above inequality $n \geq \frac{3}{\varepsilon^2 f} \ln \frac{2}{\delta}$.
3. Sample n students randomly and get corresponding indicator variables X_i.
4. Calculate $\hat{f}$ by using $X = \sum_{i=1}^{n} X_i$ and $\hat{f} = \frac{X}{n}$.
5. Repeat steps 3-4 r times to achieve r estimates $\hat{f}$. Take the average of all of these $\hat{f}$ as the final estimate of f.
Smallest Number:
Under the conditions given, the smallest number of residents we must query is $\lceil \frac{3}{\varepsilon^2 a} \ln \frac{2}{\delta} \rceil$.

Problem 2

Denote empty bin as B_{empty} and bin having items as B_{item}.
Algorithm:
1. List the n items in any order.
2. As long as there are unplaced items, take one unplaced item i out. If there exist bins B_{item} that can take in i (i.e., the total volume of bin ≤ 1 after the bin takes in i), choose one of them and put i in; otherwise, take (create) a new B_{empty} to put i in (the B_{empty} becomes B_{item}).
3. Repeat step 2 until all items are iterated and placed.
Algorithm is a 2-approximation:
Now we prove the algorithm is a 2-approximation.
Suppose the algorithm takes M bins. Let the optimal number of bins used be M^*.
We prove that $M \leq 2M^*$.
Case one: $M == M^*$
In this case, $M \leq 2M^*$ holds.
Case two: $M > M^*$
We sort all B_{item} in ascending order based on the total volumes of items of bins.
We denote the sum of volumes of all n items as S.
Since we only take (create) a new B_{empty} when no existing bin B_{item} that can take in item i, the sum of the total volumes of any two bins B_{item} must be greater than 1. We bundle bins in pairs starting with the first bin in the order that B_{item} are sorted (e.g., [bin@1, bin@2, bin@3, bin@4] $=>$ (bin@1, bin@2) and (bin@3, bin@4), or [bin@1, bin@2, bin@3, bin@4, bin@5] $=>$ (bin@1, bin@2), (bin@3, bin@4) and (bin@5,)).
If M is even, then every pair has two bins. Thus, $S > \frac{M}{2}$.
If M is odd ($M > 1$), then every pair has two bins except the final pair has only one bin. The final pair's bin's total volume X of items is greater than 0.5 (i.e., $X > 0.5$). The reason is that suppose $X \leq 0.5$, then "the sum of the total volumes of any two bins B_{item} must be greater than 1" is not hold. Thus, $S - X > \frac{M-1}{2}$,
$S > \frac{M}{2} + X - \frac{1}{2} > \frac{M}{2}$
Moreover, since the optimal solution is M^* and the capacity of each bin is 1, thus $M^* \geq S$.
So, $M^* \geq S > \frac{M}{2}$.
Thus, $2M^* > M$.

Therefore, the algorithm is a 2-approximation algorithm.
QED.

Problem 3

Solution:
According to the problem description, we can know that we need to 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. So we denote $X = \sum_{i=1}^{n} X_i$ as the number of steps for which your net profit is positive, where $X_i = 1$ represents the net profit is positive at step (round) i and $X_i = 0$ otherwise.
Now we denote $Y_i^* = \sum_{j=1}^{i} Y_j$ as the number of profit increasing in i steps, where $Y_j \in \{0, 1\}$. $Y_j = 1$ means that at step (round) j, net profit increases by 1; and $Y_j = 0$ means that at step (round) j, net profit decreases by 1. So $Pr[Y_j = 1] = \frac{1}{3}$ and $Pr[Y_j = 0] = \frac{2}{3}$. It is notable that all Y_j are independent. $\mu = E[Y_i^*] = \frac{i}{3}$.
Using Chernoff bound, we can get,
$Pr[Y_i^* \geq (1+\delta)\mu] \leq e^{-\mu\frac{\delta^2}{3}}$, for $0 < \delta \leq 1$
$\equiv Pr[Y_i^* \geq (1+\delta)\frac{i}{3}] \leq e^{-\frac{i}{3}\frac{\delta^2}{3}}$, for $0 < \delta \leq 1$
We need to know the probability of $X_i = 1$ for calculating the expected number of steps for which your net profit is positive, which means that more than $i/2$ steps achieve profit increasing by 1. Thus, we set $\delta = 0.5$ to let $Y_i^* \geq (1 + 0.2)\frac{i}{3} => Y_i^* \geq \frac{i}{2}$. So, we can get,
$Pr[Y_i^* \geq (1+0.5)\frac{i}{3}] \leq e^{-\frac{i}{3}\frac{(\frac{1}{2})^2}{3}}$
Here, we turn $\geq$ to $>$ to ensure the property of positive.
$Pr[Y_i^* > (1+0.5)\frac{i}{3}] \leq e^{-\frac{i}{3}\frac{(\frac{1}{2})^2}{3}} = e^{-\frac{i}{36}}$
Moreover, $Pr[X_i = 1] = Pr[Y_i^* > (1+0.5)\frac{i}{3}] \leq e^{-\frac{i}{36}}$.
Therefore,
$E[X] = E[\sum_{i=1}^{n} X_i] = \sum_{i=1}^{n} E[X_i] = \sum_{i=1}^{n} Pr[X_i = 1] \leq \sum_{i=1}^{n} e^{-\frac{i}{36}} = \frac{e^{-\frac{1}{36}}(1-e^{-\frac{n}{36}})}{1-e^{-\frac{1}{36}}}$
$< \frac{e^{-\frac{1}{36}}}{1-e^{-\frac{1}{36}}}$
Therefore, 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.
QED.

Problem 4

Solution:
Suppose the algorithm takes M bins. Let the optimal number of bins used be M^*.
We prove that $M \leq 2M^*$.
Case one: $M == M^*$
In this case, $M \leq 2M^*$ holds.
Case two: $M > M^*$
We sort all active bins B_{active} in ascending order based on the total sizes of items of bins. We denote the sum of sizes of all n items as S.
Since we only take (open) a new B_{active} when no existing bin B_{active} that can take in item i, the sum of the total sizes of any two bins B_{active} must be greater than V. We bundle bins in pairs starting with the first bin in the order that B_{active} are sorted (e.g., [bin@1, bin@2, bin@3, bin@4] => (bin@1, bin@2) and (bin@3, bin@4), or [bin@1, bin@2, bin@3, bin@4, bin@5] => (bin@1, bin@2), (bin@3, bin@4) and (bin@5,)).
If M is even, then every pair has two bins. Thus, $S > \frac{M}{2}V$.
If M is odd ($M > 1$), then every pair has two bins except the final pair has only one bin. The final pair's bin's total size X of items is greater than $\frac{1}{2}V$ (i.e., $X > \frac{1}{2}V$). The reason is that suppose $X \leq \frac{1}{2}V$, then "the sum of the total sizes of any two bins B_{active} must be greater than V" is not hold. Thus, $S - X > \frac{M-1}{2}V$,
$\equiv S > \frac{M-1}{2}V + X$
$\equiv S > \frac{M}{2}V + X - \frac{1}{2}V > \frac{M}{2}V$
Moreover, since the optimal solution is M^* and the capacity of each bin is V, thus $M^*V \geq S$.
So, $M^*V \geq S > \frac{M}{2}V$.
Thus, $2M^* > M$.

Therefore, the algorithm is a 2-approximation algorithm.
QED.

Problem 5

According to the problem description, we denote $X = \sum_{i=1}^{n} X_i$ as the correct number in n times of running the algorithm, where $X_i \in \{0, 1\}$. $X_i = 1$ means that the i-th run of the algorithm results in the correct answer and $X_i = 0$ otherwise. So $Pr[X_i = 1] = \frac{4}{5}$ and $Pr[X_i = 0] = \frac{1}{5}$. All X_i are independent. $\mu = E[X] = \frac{4}{5}n$.
Since we take the most common result as the final result, if n is even, the respective numbers of correct and incorrect answers may be equal. For convenience, we assume that if the respective numbers of correct and incorrect answers are equal, we get an incorrect answer.
Now we utilize Chernoff bound and get,
$Pr[X \le (1-\delta)\mu] \le e^{-\mu\frac{\delta^2}{2}}$, for $0 \le \delta \le 1$
$\equiv Pr[X \le (1-\delta)\frac{4}{5}n] \le e^{-\frac{4}{5}n\frac{\delta^2}{2}}$, for $0 \le \delta \le 1$
We want to get an upper bound on the probability that this new algorithm produces an incorrect result. So, our target is on $Pr[X \le \frac{n}{2}]$. We let $\delta = \frac{3}{8}$. So,
$Pr[X \le (1-\delta)\frac{4}{5}n] \le e^{-\frac{4}{5}n\frac{\delta^2}{2}}$, for $0 \le \delta \le 1$
$\equiv Pr[X \le (1-\frac{3}{8})\frac{4}{5}n] \le e^{-\frac{4}{5}n\frac{(\frac{3}{8})^2}{2}}$
$\equiv Pr[X \le \frac{1}{2}n] \le e^{-\frac{9}{160}n}$
Therefore, the upper bound on the probability that this new algorithm produces an incorrect result is $e^{-\frac{9}{160}n}$.