
Mohammad Sadegh Akbari

msa1376.sa@gmail.com

HW-2-solution

Sublinear Algorithms, Random Walk

Sublinear Algorithms:

To gain an understanding of sublinear time algorithms, you can refer to the following two videos: [video 1](#), [video 2](#)

1.

حالت ضربی نامساوی چرنوف را به صورت زیر در نظر بگیرید.

لم ۱ اگر X_1, \dots, X_n متغیرهای تصادفی دودویی باشند و $X = X_1 + \dots + X_n$ ، آنگاه برای هر $0 \leq \epsilon \leq 1$ داریم

$$\Pr[|X - E[X]| \geq \epsilon E[X]] < 2 \exp\left(\frac{-\epsilon^2 E[X]}{3}\right)$$

با استفاده از این لم، الگوریتمی برای مساله‌ی پیش رو ارائه دهید. آرایه‌ی n عضوی A متشکل از صفر و یک است. می‌خواهیم تقریبی بررسی کنیم که آیا تعداد یک‌ها از k بیشتر است یا خیر. به صورت دقیق‌تر، اگر تعداد یک‌ها بیشتر یا مساوی $(1 + \epsilon)k$ است باید جواب آری برگردانیم، اگر تعداد یک‌ها کمتر مساوی $(1 - \epsilon)k$ است جواب خیر و در غیر این صورت جوابی برنگردانیم. الگوریتمی ارائه دهید که با $O(n/k \cdot \epsilon^{-2} \cdot \log \delta^{-1})$ بار دسترسی، جوابی با احتمال درستی $1 - \delta$ بازگرداند.

راه حل

نمونه‌ی S را از اعضای A می‌گیریم و آنگاه بر اساس اینکه $\hat{S} = \sum_{X \in S} X$ بزرگتر یا کوچکتر از $\frac{k}{n}|S|$ است جواب بلی یا خیر برمی‌گردانیم. احتمال خطا (اینکه به جای آری، خیر برگردانیم یا بالعکس) در بدترین حالت، برای این روش، کمتر است از احتمال دور شدن از میانگین به نسبت ϵ هنگامی که میانگین $\frac{k}{n}|S|$ است و طبق نامساوی چرنوف داریم

$$\Pr(\text{error}) \leq \Pr(|\hat{S} - \frac{k}{n}|S|| \geq \epsilon \frac{k}{n}|S|) < 2 \exp\left(\frac{-\epsilon^2 \frac{k}{n}|S|}{3}\right)$$

پس برای اینکه احتمال خطا کمتر از δ باشد

$$\begin{aligned} 2 \exp\left(\frac{-\epsilon^2 \frac{k}{n}|S|}{3}\right) &\leq \delta \\ |S| &\geq 3 \frac{n}{k} \cdot \epsilon^{-2} \cdot (-\log \delta + \log 2) \\ |S| &\in O(n/k \cdot \epsilon^{-2} \cdot \log \delta^{-1}) \end{aligned}$$

2.

Definition: a list of size n is ϵ -close to sorted if can delete at most ϵn values to make it sorted. Otherwise, ϵ -far.

Given a list y_1, \dots, y_n of integers, our goal is to output whether or not the list is sorted:

- If the list is sorted, that is $y_1 \leq y_2 \leq \dots \leq y_n$ then output PASS.
- If the list is ϵ -far from sorted in Hamming distance, that is $> \epsilon n$ elements need to be changed to make the list sorted, then output FAIL with probability $\geq 3/4$.

(hint: you can use below algorithm)

Input : List y_1, y_2, \dots, y_n , and value ϵ

Output: pass or fail

```
1 for  $O(1/\epsilon)$  random indices  $i$  in  $[1, n]$  do
2   Set  $z \leftarrow y_i$ ;
3   Do binary search on  $y_1, y_2, \dots, y_n$  for  $z$ ;
4   if see inconsistency (left is bigger, right is smaller, or  $z$  not found) then
5     Fail and halt
6 Pass
```

3.

In class, you see an MST approximation algorithm for graphs in which the weights on each edge were integers in the set $\{1, \dots, w\}$.

Show that one can get an approximation algorithm when the weights can be any value in the range $[1, w]$. (It is okay to have a slightly worse running time in terms of $w, 1/\epsilon$).

(Hint: Take the case where $w=2$: consider dividing the range $[1,2]$ into pieces of length ρ with endpoints in $L = \{1, 1+\rho, 1+2\rho, \dots, 2-\rho, 2\}$. Rounding the values of the actual weights to the nearest value in L . Show that using the new weights won't change the value of the output MST by much. Show how to adapt the algorithm from class to work in this "discretized" setting.)

راه حل

برای اینکه از الگوریتم حالت گسسته در این حالت بتوانیم استفاده کنیم. وزن هر یال را ضرب در $1/\epsilon$ کرده و به نزدیک‌ترین عدد صحیح رند می‌کنیم. آنگاه الگوریتم حالت گسسته را برای بازه‌ی $[1, \lceil \frac{w}{\epsilon} \rceil]$ و با ضریب خطای $\epsilon/2$ اجرا می‌کنیم تا خروجی مقدار v را دریافت کنیم. آنگاه مقدار ϵv را به عنوان جواب نهایی باز می‌گردانیم. خطای نسبی ناشی از رند کردن حداکثر به مقدار $\epsilon/2$ در هر یال از درخت پوششی کمینه، و در نتیجه $\epsilon/2$ به ازای تمام درخت است که منجر به خطای نسبی کل ϵ می‌شود. زمان اجرای آن نیز برابر $O(dw\epsilon^3 \log \frac{w}{\epsilon})$ است.

4.

we have a stream of n integers ranging from a to b and a given number k, ϵ .

consider a median-finding algorithm that selects k random samples and returns their median. Determine the value of δ such that with a probability of at least $(1-\delta)$, the median of the sample deviates from the true median by no more than $\lceil \epsilon n/2 \rceil$ elements.

(hint: this problem can be simplified, without loss of generality, to randomly sampling k integers from 1 to n and comparing the sample median with $n/2$)

راه حل:

برای باند زدن k عدد سمپل شده از استریم اصلی را با شماره ی آنها در استریم مرتب (sort) شده جایگزین میکنیم. تعداد اعداد بین میانه اصلی و میانه گفته شده توسط الگوریتم بالا کمتر مساوی اختلاف اندازه شماره این میانه با $n/2$ است.

حال مساله به این صورت تغییر می کند که:

با انتخاب رندم k عدد طبیعی (integer) بین ۱ تا n ، کران بالایی برای احتمال اینکه اختلاف میانه این k عدد تا $n/2$ بیشتر مساوی $\epsilon n/2$ باشد کمتر از δ است. ϵ را بر حسب δ محاسبه کنید؟

برای اینکه جواب مسئله بیشتر از $\epsilon n/2$ با میانه فاصله داشته باشد باید یکی از این دو حالت برقرار باشد:

- یا بیشتر از نصف اعداد بیشتر از $n/2 + \epsilon n/2$ باشد.
- یا بیشتر از نصف اعداد کمتر از $n/2 - \epsilon n/2$ باشد.

حال کران بالای احتمال هر کدام را با هافدینگ باند می‌کنیم و این دو احتمال را با هم جمع می‌کنیم تا کران بالا به ازای $\epsilon n/2$ محاسبه شود.

بیشتر از نصف اعداد بیشتر از $n/2 + \epsilon n/2$ باشد:

متغیر تصادفی x را بدین صورت در نظر می‌گیریم که ۱ است اگر عدد بزرگتر از $n/2 + \epsilon n/2$ باشد و در غیر این صورت ۰ است.

با استفاده از باند هافدینگ داریم:

$$E(\bar{x}) = p(x) = \frac{1-\epsilon}{2}$$

$$P\left(\bar{x} - \frac{1-\epsilon}{2} \geq \frac{\epsilon}{2}\right) \leq e^{-\frac{k\epsilon^2}{2}}$$

متغیر تصادفی x را بدین صورت در نظر می‌گیریم که ۱ است اگر عدد کوچکتر از $n/2 - \epsilon n/2$ باشد و در غیر این صورت ۰ است.

با استفاده از باند هافدینگ داریم:

$$E(\bar{x}) = p(x) = \frac{1-\varepsilon}{2}$$

$$P\left(\bar{x} - \frac{1-\varepsilon}{2} \geq \frac{\varepsilon}{2}\right) \leq e^{-\frac{k\varepsilon^2}{2}}$$

با جمع این دو احتمال داریم:

$$P\left(\bar{x} - \frac{1-\varepsilon}{2} \geq \frac{\varepsilon}{2}\right) + P\left(\bar{x} - \frac{1-\varepsilon}{2} \geq \frac{\varepsilon}{2}\right) \leq 2e^{-\frac{k\varepsilon^2}{2}}$$

با فرض اینکه δ برابر با احتمال سمت راست این نابرابری باشد، داریم:

$$\delta \leq 2e^{-\frac{k\varepsilon^2}{2}}$$

$$\varepsilon^2 \leq \frac{2}{k} \ln\left(\frac{2}{\delta}\right)$$

$$\varepsilon \leq \sqrt{\frac{2}{k} \ln\left(\frac{2}{\delta}\right)}$$

در نهایت داریم:

$$\epsilon = \sqrt{\frac{2}{k} \ln \left(\frac{2}{\delta} \right)}$$

$$\frac{\epsilon n}{2} = \frac{n}{2k} \ln \left(\frac{2}{\delta} \right)$$

Random Walk:

5.

Given any connected graph with $n = \text{number of vertices}$ and $m = \text{number of edges}$, determine a function $f(n, m, y)$ that is linear in n , m , and y , such that a random walk of $f(n, m, y)$ steps ensures each node is visited at least once with a probability of at least $1 - 2^{-y}$, regardless of the starting node.

(hint: start with Markov's inequality, which states that $\Pr(X > w) \leq E(X)/w$)

راه حل:

امید ریاضی تعداد قدم های تصادفی برای دیدن تمام نود ها کمتر از nm است.

پس طبق نامساوی مارکوف احتمال اینکه بعد از $2nm$ قدم تصادفی تمام نود ها را ندیده باشیم را محاسبه میکنیم:

$$P(X > 2nm) \leq \frac{CG}{2nm} \leq \frac{nm}{2nm} \leq \frac{1}{2}$$

حال احتمال اینکه بعد از y بار تکرار $2nm$ قدم تصادفی تمام نود ها را ندیده باشیم حساب میکنیم:

$$P \leq \left(\frac{1}{2}\right)^y = 2^{-y}$$

حال احتمال اینکه بعد از y بار تکرار $2nm$ قدم تصادفی تمام نود ها را دیده باشیم حساب میکنیم:

$$P \geq 1 - 2^{-y}$$

6.

Imagine you're stranded in Monte Carlo with only k dollars to your name. You have one day to earn n dollars to repay a debt—or face death. Your only option is to play a risky casino game with these rules:

1. Gain 1 dollar on winning a round.
2. Lose 1 dollar on losing a round.
3. Each round has a 50% chance of winning or losing.
4. You can play up to m rounds tomorrow.

Can you devise an algorithm to calculate the probability of you seeing the day after tomorrow?

(hint: Construct a probability vector $v_i \in \mathbb{R}^n$, where each element $v_i[j]$ signifies the probability of possessing j dollars at the i -th game. Define a transition matrix $A \in \mathbb{R}^{n \times n}$ that governs the evolution of v_i , satisfying the following relationship: $v_{i+1} = A v_i$. This approach make an algorithm with a time complexity of $O(n^2 * m)$)

راه حل:

بردار v را با m درایه در نظر میگیریم بدین صورت که درایه i احتمال داشتن i دلار را نشان میدهد.

پس در ابتدا v بدین صورت است:

$$v_k = 1, v_{i \neq k} = 0$$

حال ماتریس A را بدین صورت در نظر میگیریم که با داشتن v بتوانیم احتمالات را بعد از انجام یک قمار دیگر محاسبه کرد.

اگر i دلار داشته باشیم بعد از یک قمار دیگر با احتمال $1/2$ $i+1$ دلار داریم و با احتمال $1/2$ $i-1$ دلار داریم.

اگر 0 یا n دلار داشته باشیم در همان خانه می مانیم و دیگر ادامه نمی دهیم.

و بقیه درایه ها 0.

$$A[j, k] = 1 \quad \text{if}(k = j = 0)$$

$$A[j, k] = 1 \quad \text{if}(k = j = n)$$

$$A[j, k] = 1/2 \quad \text{elseif}(k = j + 1)$$

$$A[j, k] = 1/2 \quad \text{else if}(k = j - 1)$$

$$A[j, k] = 0 \quad \text{otherwise}$$

برای فهم بدون بار فکری و با یک نگاه ماتریس با $n=7$ شکل زیر است:

$$\mathbf{A} = \begin{bmatrix} 1 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 1 \end{bmatrix}$$

حال Av معادل محاسبه احتمال داشتن هر حجم از دلار بعد از ۱ قمار است.

حال A^2v معادل محاسبه احتمال داشتن هر حجم از دلار بعد از 2 قمار است.

حال $A^i v$ معادل محاسبه احتمال داشتن هر حجم از دلار بعد از i قمار است.

و

$A^m v$ معادل محاسبه احتمال داشتن هر حجم از دلار بعد از m قمار است.

درایه n ام $A^m v$ معادل احتمال داشتن n دلار با محدودیت m قمار است.