دانشکده علوم ریاضی
دانشگاه صنعتی شریف

الگوریتم‌های خلاصه‌سازی برای مه‌داده

محمدهادی فروغمنداعرابی

پاییز ۱۳۹۹

شمارنده با حافظه کم

جلسه ی سوم

نگارنده: محمد جواد سجادی

۱ مروری بر مباحث گذشته

مسئله ی شمارنده : شمارهنده ای داریم که سه عمل انجام میدهد :

- `init()`؛ مقدار عدد n را ۰ می‌گذارد.
- `update()`؛ عدد n را یک واحد افزایش می‌دهد، یعنی $n \leftarrow n + 1$.
- `query()`؛ عدد n و یا تخمینی از آن را خروجی می‌دهد.

که میزان حافظه ی دقیق مورد نیاز برای ان $(\log \mathrm{n})$ برسی شد در این جلسه به راه حل تقریبی با حافظه ی کمتر برای این مسئله می‌پردازیم

۲ راه حل تقریبی

برای تعریف دقیق اینکه منظورمان از تقریب چیست باید گفت که ϵ را اندازه ی خطا و δ را احتمال خطا در نظر می‌گیریم . می‌خواهیم تمامی ϵ و δ را پیدا کنیم که در رابطه ی رو به رو صدق می‌کنند.

$$\mathbb{P}(|\tilde{n} - n| > \epsilon n) < \delta$$

[Mor78]

۱.۲ الگوریتم موریس

اگر بجای اضافه شدن X در هر مرحله اینگونه الگوریتم را در نظر بگیریم :

- `init()`؛ مقدار عدد X را ۰ می‌گذارد.
- `update()`؛ عدد X را با احتمال ۲^{-X} افضایش می‌دهیم. .
- `query()`؛ عدد $۲^X - ۱$ رابه عنوان خروجی ارایه می‌دهیم.

اگر X_n را مقدار X پس از n بار به روز رسانی در نظر بگیریم

لم ۱ ($\mathbb{E}۲^{X_n} = n + ۱$).

پایه: حکم به ازای n=1 بدهی است. چون پس از ۱ بار به روز رسانی با احتمال $۱ = ۲^۰$ مقدار $X_۱ = ۱$ می‌شود

$$\mathbb{E}۲^{X_۱} = ۲^۱ = ۲ = ۱ + ۱$$

گام اثبات : فرض کنید حکم برای n=k درست باشد حکم را برای n+1 اثبات می‌کنیم.

$$\mathbb{E}[۲^{X_{n+۱}}] = \sum_{j=۰}^{\infty} \mathbb{E}[۲^{X_{n+۱}} | X_n = j] * \mathbb{P}(X_n = j)$$
$$= \sum_{j=۰}^{\infty} (۲^{-j}۲^{j+۱} + (۱ - ۲^{-j})۲^j) * \mathbb{P}(X_n = j)$$
$$= \sum_{j=۰}^{\infty} (۲ + ۲^j - ۱) * \mathbb{P}(X_n = j)$$
$$= \sum_{j=۰}^{\infty} (۲^j + ۱) * \mathbb{P}(X_n = j)$$
$$= \sum_{j=۰}^{\infty} ۲^j * \mathbb{P}(X_n = j) + \sum_{j=۰}^{\infty} \mathbb{P}(X_n = j)$$

و باید گفت که $\sum_{j=۰}^{\infty} \mathbb{P}(X_n = j) = ۱$ چون تمامی حالات ممکنه برای X_n را در نظر گرفتیم و از طرفی

$$\sum_{j=۰}^{\infty} ۲^j * \mathbb{P}(X_n = j) = \sum_{j=۰}^{\infty} ۲^{X_n} * \mathbb{P}(X_n = j)$$
$$= \sum_{j=۰}^{\infty} ۲^{X_n} * \mathbb{P}(X_n = j) = \mathbb{E}[۲^{X_n}]$$

طبق فرض استقرا $\mathbb{E}[2^{X_n}] = n + 1$ پس :

$$\mathbb{E}[۲^{X_n}] + ۱ = n + ۱ + ۱ = n + ۲$$

حکم اثبات شد.

لم ۲ ($\mathbb{E}۲^{۲*X_n} = \frac{۳}{۲}n^۲ + \frac{۳}{۲}n + ۱$).

پایه: حکم به ازای n=1 بدهی است. چون پس از ۱ بار به روز رسانی با احتمال $۱ = ۲^۰$ مقدار $X_۱ = ۱$ می‌شود

$$\mathbb{E}۲^{۲*X_۱} = ۲^{۲*۱} = ۴ = \frac{۳}{۲} + \frac{۳}{۲} + ۱ = ۴$$

گام اثبات : فرض کنید حکم برای $n{=}k$ درست باشد حکم را برای $n{+}1$ اثبات می‌کنیم.

$$\begin{aligned}
\mathbb{E}[۲^{۲*X_{n+۱}}] &= \sum_{j=۰}^{\infty} \mathbb{E}[۲^{۲*X_{n+۱}}|X_n = j] * \mathbb{P}(X_n = j) \\
&= \sum_{j=۰}^{\infty} (۲^{-j} ۲^{۲(j+۱)} + (۱ - ۲^{-j}) ۲^{۲j}) * \mathbb{P}(X_n = j) \\
&= \sum_{j=۰}^{\infty} (۲^{j+۲} + ۲^{۲j} - ۲^{j}) * \mathbb{P}(X_n = j) \\
&= \sum_{j=۰}^{\infty} (۳ * ۲^{j} + ۲^{۲j}) * \mathbb{P}(X_n = j) \\
&= ۳ * \sum_{j=۰}^{\infty} ۲^{j} * \mathbb{P}(X_n = j) + \sum_{j=۰}^{\infty} (۲^{۲j}) * \mathbb{P}(X_n = j)
\end{aligned}$$

و باید گفت که $\sum_{j=۰}^{\infty} ۲^j * \mathbb{P}(X_n = j)$ همان $\mathbb{E}[2^{X_n}]$ است و عبارت $\sum_{j=0}^{\infty} (2^{2*j}) * \mathbb{P}(X_n = j)$ همان $\mathbb{E}[2^{2*X_n}]$ است. پس :

$$\begin{aligned}
&\mathbb{E}[۲^{۲*X_n}] + ۳ * \mathbb{E}[۲^{X_n}] = \frac{۳}{۲} n^۲ + \frac{۳}{۲} n + ۱ + ۳ * (n + ۱) \\
&= \frac{۳}{۲}(n^۲ + ۲n + ۱) + \frac{۳}{۲}(n + ۱) + ۱ = \frac{۳}{۲}(n + ۱)^۲ + \frac{۳}{۲}(n + ۱) + ۱
\end{aligned}$$

حکم اثبات شد.

می‌خواهیم این مقدار را محاسبه کنیم و یا کرانی برای ان تعیین کنیم

$$\mathbb{P}(|\tilde{n} - n| > \epsilon n)$$

طبق قضیه ی چبیشف خواهیم داشت :

$$\begin{aligned}
&\mathbb{P}(|\tilde{n} - n| > \epsilon n) < \frac{۱}{\epsilon^۲ n^۲} Var[\tilde{n}] \\
&= \frac{\mathbb{E}[\tilde{n}^۲] - \mathbb{E}[\tilde{n}]^۲}{\epsilon^۲ n^۲} = \frac{\frac{۳}{۲} n^۲ + \frac{۳}{۲} n + ۱ - (n + ۱)^۲}{\epsilon^۲ n^۲} \\
&= \frac{\frac{n^۲ - n}{۲}}{\epsilon^۲ n^۲} = \frac{n - ۱}{۲n\epsilon^۲} < \frac{۱}{۲\epsilon^۲}
\end{aligned}$$

اما اینجا اشکال بزرگی وجود دارد که مثلا برای $\epsilon = \frac{1}{2}$ به کران $\mathbb{P}(|\tilde{n} - n| > \epsilon n) < \frac{۱}{۲(\frac{۱}{۲})^۲} = ۲$ دست پیدا کردیم که اصلا کران خوبی نیست.

۲.۲ الگوریتم موریس+

به جای قرار دادن یک ماشین با الگوریتم موریس اگر بتوانیم s ماشین با این الگوریتم قرار بدهیم و برای جواب نهایی میانگین خروجی ماشین ها را قرار بدهیم . پس خواهیم داشت :

$$\tilde{n} = \frac{۱}{s} \sum_{i=۱}^{s} \tilde{n}_i$$

لم ۳ (ماشین های الگوریتم موریس مستقل هستند). اینگونه میتوان استدلال کرد که برای هر n بار به روز رسانی جواب هر یک مسقل از دیگری خواهد بود به این معنا که اضافه شدن هیچ یک مرتبط به دیگری نیست و تاثیری روی هم ندارند پس این s ماشین مستقل از هم هستند چون همگی تعداد ورودی مشخص و یکسانی ورودی می‌گیرند.

به طریق مشابه نامساوی را می‌نویسیم و با استفاده از لم خواهیم داشت:

$$\begin{aligned}
&\mathbb{P}(|\tilde{n}-n|>\epsilon n)<\frac{1}{\epsilon^2 n^2}Var[\tilde{n}]\\
&=\frac{1}{\epsilon^2 n^2}Var[\frac{1}{s}\sum_{i=1}^{s}\tilde{n}_i]=\frac{1}{\epsilon^2 n^2 s^2}Var[\sum_{i=1}^{s}\tilde{n}_i]\\
&=\frac{1}{\epsilon^2 n^2 s^2}\sum_{i=1}^{s}Var[\tilde{n}_i]=\frac{1}{\epsilon^2 n^2 s^2}s*Var[\tilde{n}_i]=\frac{1}{\epsilon^2 n^2 s}\\
&=\frac{1}{\epsilon^2 n*s}*\frac{n-1}{2}<\frac{1}{2*\epsilon^2*s}<\delta
\end{aligned}$$

کران کمی بهتر شد اما همچنان به اندازه ی کافی خوب نیست لازم به ذکر است که :

$$s>\frac{1}{2*\epsilon^2\delta}=\Theta(\frac{1}{\epsilon^2\delta})$$

۳.۲ الگوریتم موریس++

اجرای t تا موریس+ در کنار هم و با $\delta=\frac{1}{3}$ و به عنوان جواب میانه را بر می‌گردانیم

متغیر های Y_i را ۱ در نظر می‌گیریم در صورتی که $\mathbb{P}(|\tilde{n}-n|>\epsilon n)<\frac{1}{3}$ باشد در غیر این صورت ۰ در نظرمی‌گیریم. پس در صورتی الگوریتم ما بخواهد نتیجه ی مطلوب بدهد پس باید بیش از نیمی از از Y_i انها ۱ باشند .

لم ۴ (میانه؟). به صورت معمول میانگین را به عنوان خروجی بر می‌گردانند اما در انجا به علت احتمالاتی بودن سیستم احتمال وجود داده ی پرت ی بسیار زیاد است پس باید تخمین‌گری انتخاب کنیم که نسبت به داده های پرت واکنش زیادی نشان ندهد و یکی از بهترین و ساده ترین این تخمین‌گر ها میانه است.

لم ۵ ($\mathbb{E}(\sum_{i=1}^{t}Y_i)>\frac{2t}{3}$). به علت استقلال :

$$\mathbb{E}(\sum_{i=1}^{t}Y_i)=\sum_{i=1}^{t}\mathbb{E}(Y_i)>\sum_{i=1}^{t}\frac{2}{3}=\frac{2t}{3}$$

حال به برسی اثبات می‌پردازیم اگر داشته باشیم :

$$\begin{aligned}
&\mathbb{P}(\sum_{i=1}^{t}Y_i<\frac{t}{2})=\mathbb{P}(\sum_{i=1}^{t}Y_i-\mathbb{E}[Y_i]<\frac{t}{2}-\mathbb{E}[Y_i])\\
&=\mathbb{P}(\mathbb{E}[Y_i]-\sum_{i=1}^{t}Y_i>\mathbb{E}[Y_i]-\frac{t}{2})=\mathbb{P}(|\mathbb{E}[Y_i]-\sum_{i=1}^{t}Y_i|>\mathbb{E}[Y_i]-\frac{t}{2})
\end{aligned}$$

می‌دانیم طبق لم که عبارت سمت راست احتمال مثبت است پس تغییری روی ان ایجاد نمی‌کنیم:

$$\begin{aligned}
&\mathbb{P}(|\mathbb{E}[Y_i]-\sum_{i=1}^{t}Y_i|>\mathbb{E}[Y_i]-\frac{t}{2})\\
&=\mathbb{P}(|\mathbb{E}[Y_i]-\sum_{i=1}^{t}Y_i|>\frac{1}{4}\mathbb{E}[Y_i]+(\frac{3}{4}\mathbb{E}[Y_i]-\frac{t}{2}))\\
&<\mathbb{P}(|\mathbb{E}[Y_i]-\sum_{i=1}^{t}Y_i|>\frac{1}{4}\mathbb{E}[Y_i])
\end{aligned}$$

لازم به ذکر است که طبق لم داریم که

$$\frac{3}{4}\mathbb{E}[Y_i]-\frac{t}{2}>\frac{3}{4}*\frac{2t}{3}=\frac{t}{2}-\frac{t}{2}=0$$

پس طبق کران چرنوف خواهیم داشت که :

$$\mathbb{P}(|\mathbb{E}[Y_i]-\sum_{i=1}^{t}Y_i|>\frac{1}{4}\mathbb{E}[Y_i])<2e^{\frac{-\epsilon^2*\mathbb{E}[Y_i]}{3}}<2e^{\frac{-\epsilon^2*\frac{2t}{3}}{3}}=2e^{\frac{-\frac{1}{4}^2*\frac{2t}{3}}{3}}=2e^{\frac{-t}{72}}<\delta$$

پس با هم هی این تفاسیر خواهیم داشت که :

$$t > c * \ln(\frac{۱}{\delta}) = \Theta(\ln \frac{۱}{\delta})$$

پس حافظه ی نهایی برابر خواهد بود با :

$$s * t * \ln(\ln(n)) = \Theta(\frac{۱}{\epsilon^۲} * \ln(\frac{۱}{\delta}) * \ln(\ln(n))$$

به عبارت دقیق تر ما به احتمال $1 - \delta$ حافظه ی مورد نیاز ما $O(\frac{1}{\epsilon^2} * \ln(\frac{1}{\delta}) * \ln(\ln(\frac{n}{\epsilon\delta}))$ خواهد بود.

۴.۲ کمی بهتر؟؟

اگر بجای 2^{-X} عبارت $a > 0$, $\frac{1}{1+a}^X$ مناسبی در جای ان قرار دهیم به کران $O(\ln(\frac{۱}{\epsilon}) + \ln(* \ln(\frac{۱}{\delta})) + \ln(\ln(n))$ برای حافظه خواهیم رسید و

$$O(\ln(\ln_{۱+\epsilon}(n)) = O(\ln(\frac{۱}{\epsilon}) + \ln(\ln(n)))$$

کران پایین مسئله ی گفته شده است. [PJN17]

مراجع

[Mor78] Robert Morris. Counting large numbers of events in small registers. *Commun. ACM*, 21(10):840–842, 10 1978.

[PJN17] Vinh-Kha Le Prof. Jelani Nelson. Sketching algorithms for big data. Lecture 01:3–8, Fall 2017.