大数据算法	March 14, 2025

Lecture 5: 数据流：Data Stream Algorithms

Lecturer: 彭攀 | *Scribe:* 申冉冉，徐航宇

在数据流模型中，数据以流的形式连续到达，算法拥有有限的内存，无法存储完整的数据，这导致一些朴素的算法无法实现。因此在数据流算法中，我们通常对过去的数据保留有限的摘要信息，以较高的概率输出问题的正确结果。

本节课我们将学习概率计数（Probabilistic Counting）和蓄水池抽样（Reservoir Sampling）算法。

1 概率计数

1.1 问题定义

考虑一个有限的样本空间 Ω，考虑时刻 $1,2,3,4,\ldots$。有一个机器能源源不断地产生事件 $x_i \in \Omega$（在时刻 i 产生事件 $x_{\pi[i]} \in \Omega$），要求在任意给定时刻，输出机器产生的事件的数量，使用的空间越少越好。

一个 naive 的算法是维护一个计数器 n，初始 $n = 0$，

- 更新：机器每产生一个事件，计数器 n 的值加一；
- 查询：在任意时刻查询时，输出计数器 n 的值即可。

这种 naive 的算法能输出**精确解**，需要 $\Theta(\log n)$ 个比特存储计数器 n 的值。并且 $\Theta(\log n)$ 比特的空间对于输出精确解是必要的。

我们想要在更少的空间内解决概率计数问题。具体来说，我们希望算法输出一个**近似解** $\tilde{n}$，满足对于任意给定的 $\varepsilon, \delta \in (0,1)$，有

$$\Pr[|\tilde{n} - n| > \varepsilon n] < \delta, \tag{1}$$

即

$$\Pr[|\tilde{n} - n| \leq \varepsilon n] > 1 - \delta.$$

接下来我们依次介绍针对概率计数问题的 Morris 算法、Morris+ 算法和 Morris++ 算法。

1.2 Morris 算法

1.2.1 算法描述

Morris 算法的描述如下：

1. 在时刻 0，初始化 $X_0 = 0$；

2. 用 X_i 表示 i 时刻 X 的值，在 $i+1$ 时刻（每个时刻来一个事件）：

$$X_{i+1} = \begin{cases} X_i + 1, & \text{w.p. } \frac{1}{2^{X_i}} \\ X_i, & \text{w.p. } 1 - \frac{1}{2^{X_i}} \end{cases}.$$

   - 直观上看，X 的值越大，下一时刻 X 的值被增加一的概率就越小

3. 在时刻 n 查询时，输出 $\tilde{n} = 2^{X_n} - 1$。

空间　粗略地看，X_n 的最大值为 $\log n$，因此 X_n 可以用 $\log\log n$ 比特存储（在 Morris++ 算法的空间分析中会对这一点进行证明）。

1.2.2　算法分析

接下来我们对 Morris 算法进行分析，看 Morris 算法是否满足式 (1)。

引理 1. $\mathbb{E}[\tilde{n}] = n$.

Proof. 因为 $\mathbb{E}[\tilde{n}] = \mathbb{E}\left[2^{X_n}\right] - 1$，因此只需要证明 $\mathbb{E}\left[2^{X_n}\right] = n+1$。接下来，我们用数学归纳法对此进行证明：

- 在 $n=0$ 时，有 $X_n = X_0 = 0$，因此 $\mathbb{E}\left[2^{X_n}\right] = 1$，$\mathbb{E}\left[2^{X_n}\right] = n+1$ 成立；
- 假设在 n 时刻（$n \geq 0$），$\mathbb{E}\left[2^{X_n}\right] = n+1$ 成立；
- 在 $n+1$ 时刻，

$$\begin{aligned}
\mathbb{E}\left[2^{X_{n+1}}\right] &= \sum_{i=0}^{\infty} \Pr[X_n = i] \cdot \mathbb{E}\left[2^{X_{n+1}} | X_n = i\right] \\
&= \sum_{i=0}^{\infty} \Pr[X_n = i] \cdot \left[2^{i+1} \cdot \frac{1}{2^i} + 2^i \cdot \left(1 - \frac{1}{2^i}\right)\right] \\
&= \sum_{i=0}^{\infty} \Pr[X_n = i] \cdot \left(2 + 2^i - 1\right) \\
&= \sum_{i=0}^{\infty} \Pr[X_n = i] \cdot \left(2^i + 1\right) \\
&= \sum_{i=0}^{\infty} \Pr[X_n = i] \cdot 2^i + \sum_{i=0}^{\infty} \Pr[X_n = i] \\
&= \mathbb{E}\left[2^{X_n}\right] + 1 \\
&= n + 2.
\end{aligned}$$

- 综上，$\mathbb{E}\left[2^{X_n}\right] = n+1$ 成立，证毕。

■

引理 2. $\mathrm{Var}[\tilde{n}] = \frac{n^2}{2} - \frac{n}{2}$.

Proof.

$$\begin{aligned}\mathrm{Var}[\tilde{n}] &= \mathbb{E}\left[(\tilde{n} - \mathbb{E}\left[\tilde{n}\right])^2\right] \\ &= \mathbb{E}\left[\left(2^{X_n} - 1 - n\right)^2\right] \\ &= \mathbb{E}\left[\left(2^{X_n} - (1+n)\right)^2\right] \\ &= \mathbb{E}\left[2^{2X_n}\right] - 2(1+n)\cdot\mathbb{E}\left[2^{X_n}\right] + (1+n)^2 \\ &= \mathbb{E}\left[2^{2X_n}\right] - (1+n)^2.\end{aligned}$$

接下来，我们需要计算 $\mathbb{E}\left[2^{2X_n}\right]$。假设 $\mathbb{E}\left[2^{2X_n}\right] = \frac{3}{2}n^2 + \frac{3}{2}n + 1$，我们同样用数学归纳法对此进行证明：

- 在 $n = 0$ 时，有 $X_n = X_0 = 0$，因此 $\mathbb{E}\left[2^{2X_n}\right] = 1$，$\mathbb{E}\left[2^{2X_n}\right] = \frac{3}{2}n^2 + \frac{3}{2}n + 1$ 成立；
- 假设在 n 时刻（$n \geq 0$），$\mathbb{E}\left[2^{2X_n}\right] = \frac{3}{2}n^2 + \frac{3}{2}n + 1$ 成立；
- 在 $n+1$ 时刻，

$$\begin{aligned}\mathbb{E}\left[2^{2X_{n+1}}\right] &= \sum_{i=0}^{\infty}\Pr[X_n = i]\cdot\mathbb{E}\left[2^{2X_{n+1}} | X_n = i\right] \\ &= \sum_{i=0}^{\infty}\Pr[X_n = i]\cdot\left[2^{2(i+1)}\cdot\frac{1}{2^i} + 2^{2i}\cdot\left(1 - \frac{1}{2^i}\right)\right] \\ &= \sum_{i=0}^{\infty}\Pr[X_n = i]\cdot\left(2^{i+2} + 2^{2i} - 2^i\right) \\ &= \sum_{i=0}^{\infty}\Pr[X_n = i]\cdot\left(2^{2i} + 3\cdot 2^i\right) \\ &= \sum_{i=0}^{\infty}\Pr[X_n = i]\cdot 2^{2i} + 3\sum_{i=0}^{\infty}\Pr[X_n = i]\cdot 2^i \\ &= \mathbb{E}\left[2^{2X_n}\right] + 3\,\mathbb{E}\left[2^{X_n}\right] \\ &= \frac{3}{2}n^2 + \frac{3}{2}n + 1 + 3(n+1) \\ &= \frac{3}{2}(n+1)^2 + \frac{3}{2}(n+1) + 1.\end{aligned}$$

- 综上，对于任意 $n \geq 0$ 有 $\mathbb{E}\left[2^{2X_n}\right] = \frac{3}{2}n^2 + \frac{3}{2}n + 1$。

因此，$\mathrm{Var}\left[\tilde{n}\right] = \mathbb{E}\left[2^{2X_n}\right] - (1+n)^2 = \frac{n^2}{2} - \frac{n}{2}$。 ∎

根据切比雪夫不等式，有

$$\Pr\left[|\tilde{n} - n| > \varepsilon n\right] = \Pr\left[|\tilde{n} - \mathbb{E}[\tilde{n}]| > \varepsilon n\right] \leq \frac{\mathrm{Var}\left[\tilde{n}\right]}{\varepsilon^2 n^2} \leq \frac{\frac{n^2}{2}}{\varepsilon^2 n^2} = \frac{1}{2\varepsilon^2}.$$

然而在 ε 很小时，上面不等式右侧 $\frac{1}{2\varepsilon^2}$ 的值会很大。特别地，在 $\varepsilon^2 < \frac{1}{2}$ 时，不等式右侧的值

> 1，不等式没有任何意义，不满足式 (1)。因此 Morris 算法无法解决概率计数问题，接下来我们介绍 Morris+ 算法修正这个问题。

1.3 Morris+ 算法

1.3.1 算法描述

1. 独立地运行 s 个 Morris 算法，获得 $\tilde{n}_1, \ldots, \tilde{n}_s$；
2. 在时刻 n 查询时，输出 $\tilde{n} = \frac{1}{s}\sum_{i=1}^{s} \tilde{n}_i$。

1.3.2 算法分析

接下来我们对 Morris 算法进行分析。

引理 3. $\mathbb{E}[\tilde{n}] = n.$

Proof.

$$\begin{aligned}
\mathbb{E}[\tilde{n}] &= \mathbb{E}\left[\frac{1}{s}\sum_{i=1}^{s} \tilde{n}_i\right] \\
&= \frac{1}{s}\sum_{i=1}^{s} \mathbb{E}[\tilde{n}_i] \\
&= \frac{1}{s} \cdot s \cdot n && \text{（根据引理 1 有 } \mathbb{E}[\tilde{n}_i] = n\text{）} \\
&= n.
\end{aligned}$$

■

引理 4. $\mathrm{Var}[\tilde{n}] < \frac{n^2}{2s}.$

Proof.

$$\begin{aligned}
\mathrm{Var}[\tilde{n}] &= \mathrm{Var}\left[\frac{1}{s}\sum_{i=1}^{s} \tilde{n}_i\right] \\
&= \frac{1}{s^2}\sum_{i=1}^{s} \mathrm{Var}[\tilde{n}_i] && \text{（因为 } \tilde{n}_1, \ldots, \tilde{n}_s \text{ 独立）} \\
&= \frac{1}{s^2} \cdot s \cdot \left(\frac{n^2}{2} - \frac{n}{2}\right) && \text{（根据引理 2）} \\
&< \frac{n^2}{2s}.
\end{aligned}$$

■

根据切比雪夫不等式，有

$$\Pr\left[|\tilde{n}-n|>\varepsilon n\right]=\Pr\left[|\tilde{n}-\mathbb{E}[\tilde{n}]|>\varepsilon n\right]\leq\frac{\operatorname{Var}\left[\tilde{n}\right]}{\varepsilon^2n^2}<\frac{\frac{n^2}{2s}}{\varepsilon^2n^2}=\frac{1}{2s\varepsilon^2}.$$

为了使 Morris 算法满足式 (1)，我们令 $\frac{1}{2s\varepsilon^2}=\delta$，有 $s=\frac{1}{2\delta\varepsilon^2}$。

因此，在 $s=\frac{1}{2\delta\varepsilon^2}$ 时，Morris+ 算法的输出 $\tilde{n}$ 以 $\geq(1-\delta)$ 的概率满足 $|\tilde{n}-n|\leq\varepsilon n$。

空间 Morris+ 算法独立地运行 s 个 Morris 算法，运行一个 Morris 算法需要 $O(\log\log n)$ 比特，因此 Morris+ 算法需要 $O(s\cdot\log\log n)=O\left(\frac{1}{\delta\varepsilon^2}\cdot\log\log n\right)$ 比特。

接下来我们介绍一种空间更优的算法 Morris++。

1.4 Morris++ 算法

1.4.1 算法描述

首先对于 Morris+ 算法，我们取 $\delta=1/3$，那么 $s=\Theta\left(\frac{1}{\varepsilon^2}\right)$ 且 Morris+ 算法能以 $\geq\frac{2}{3}$ 的概率返回一个值 $\tilde{n}$ 满足 $|\tilde{n}-n|\leq\varepsilon n$，Morris+ 算法需要的空间为 $O\left(s\cdot\log\log n\right)=O\left(\frac{1}{\varepsilon^2}\cdot\log\log n\right)$。

Morris++ 的算法描述如下：

1. 独立地运行 t 个能以 $\geq\frac{2}{3}$ 的概率成功的 Morris+ 算法（即上文取 $\delta=1/3,s=\Theta\left(\frac{1}{\varepsilon^2}\right)$ 的 Morris+ 算法），获得 $\tilde{n}_1,\ldots,\tilde{n}_t$；
2. 在时刻 n 查询时，返回 $\tilde{n}_1,\ldots,\tilde{n}_t$ 的中间值（median），记为 $\tilde{n}$。

1.4.2 算法分析

对 $1\leq i\leq t$，令随机变量 Y_i 表示第 i 个 Morris+ 算法是否成功，即

$$Y_i=\begin{cases}1, & |\tilde{n}_i-n|\leq\varepsilon n\\ 0, & |\tilde{n}_i-n|>\varepsilon n\end{cases}.$$

显然，$\mathbb{E}\left[Y_i\right]=\Pr[Y_i=1]=\Pr[|\tilde{n}_i-n|\leq\varepsilon n]\geq\frac{2}{3}$。
令 $Y=\sum_{i=1}^{t}Y_i$ 表示 t 个 Morris+ 算法里成功的算法个数。

引理 5. $\mathbb{E}\left[Y\right]\geq\frac{2}{3}t$.

Proof. $\mathbb{E}\left[Y\right]=\sum_{i=1}^{t}\mathbb{E}\left[Y_i\right]\geq\frac{2}{3}t$. ■

Morris++ 算法输出 t 个 Morris+ 算法的中间值 $\tilde{n}$，因此 Morris++ 算法失败时（指 $|\tilde{n}-n|>\varepsilon n$），$t$ 个 Morris+ 算法中必定有 $>\frac{t}{2}$ 个失败，即成功的个数 $\leq\frac{t}{2}$。

记事件 B 表示 Morris++ 算法失败，根据 Chernoff-Hoeffding 不等式，有

$$\Pr[B] = \Pr[|\tilde{n} - n| > \varepsilon n] = \Pr[Y \leq \frac{t}{2}] \leq \Pr[Y - \mathbb{E}[Y] < -\frac{t}{6}] \leq e^{-\frac{2(-t/6)^2}{t}} = e^{-\frac{t}{18}}.$$

为了让 Morris++ 算法满足式 (1)，我们令 $e^{-\frac{t}{18}} \leq \delta$，即 $t \geq 18 \ln \frac{1}{\delta}$。

因此，在 $t \geq 18 \ln \frac{1}{\delta}$ 时，Morris++ 算法的输出 $\tilde{n}$ 以 $\geq (1-\delta)$ 的概率满足 $|\tilde{n} - n| \leq \varepsilon n$。

空间 Morris++ 算法独立运行 t 个 Morris+ 算法，每个 Morris+ 算法独立运行 s 个 Morris 算法，每个 Morris 算法维持一个计数器 X，因此 Morris++ 算法需要维护 $s \cdot t = O\left(\frac{1}{\varepsilon^2} \cdot \ln \frac{1}{\delta}\right)$ 个 X。

接下来我们证明 X 的值不会太大。

假设在时刻 i，有 $X_i \geq \log \frac{stn}{\delta}$，根据 Morris 算法的描述，在 $i+1$ 时刻，X 的值会加一的概率（即 $X_{i+1} = X_i + 1$）的概率为 $\frac{1}{2^{X_i}} \leq \frac{\delta}{stn}$。那么在 X 的值到达 $\log \frac{stn}{\delta}$ 后，根据 union bound，接下来的 n 个时刻内，至少有一个时刻 X 的值被增加一的概率 $\leq n \cdot \frac{\delta}{stn} = \frac{\delta}{st}$。再次根据 union bound，$s \cdot t$ 个 X 里至少有一个 X 的值在 $\log \frac{stn}{\delta}$ 后会增加一的概率 $\leq s \cdot t \cdot \frac{\delta}{st} = \delta$。

因此，以 $\geq (1-\delta)$ 的概率，Morris++ 算法维护的 $s \cdot t$ 个 X 都满足 X 的值最多到达 $\log \frac{stn}{\delta}$ 后将不再增加，即 $X \leq \log \frac{stn}{\delta}$，即存储一个 X 至多需要 $\log \log \frac{stn}{\delta}$ 比特。

注意：以同样的方式，我们可以证明在 Morris+ 算法中，以 $\geq (1-\delta)$ 的概率，Morris+ 算法维护的 s 个 X 都满足 X 的值最多到达 $\log \frac{sn}{\delta}$ 后将不再增加，即 $X \leq \log \frac{sn}{\delta}$，即存储一个 X 至多需要 $\log \log \frac{sn}{\delta}$ 比特。

在 $s = \Theta\left(\frac{1}{\varepsilon^2}\right), t \geq 18 \ln \frac{1}{\delta}$ 时，Morris++ 算法：

- 坏事件 1：算法的输出 $\tilde{n}$ 以 $\leq \delta$ 的概率满足 $|\tilde{n} - n| > \varepsilon n$；
- 坏事件 2：以 $\leq \delta$ 的概率，至少存在一个 X 满足 $X > \log \frac{stn}{\delta}$，即至少存在一个 X 不能用 $\leq \log \log \frac{stn}{\delta}$ 个比特存储；

根据 union bound，Morris++ 算法的输出以$\geq (1-2\delta)$的概率满足：

- 算法的输出 $\tilde{n}$ 满足$|\tilde{n} - n| \leq \varepsilon n$；
- 每个 X 都满足 $X \leq \frac{stn}{\delta}$，因此 Morris++ 算法使用的空间为 $O\left(s \cdot t \cdot \log \log \frac{stn}{\delta}\right) = O\left(\frac{1}{\varepsilon^2} \cdot \log \frac{1}{\delta} \cdot \log \log \frac{stn}{\delta}\right) = O\left(\frac{1}{\varepsilon^2} \cdot \log \frac{1}{\delta} \cdot \log \log \frac{n}{\varepsilon \delta}\right)$ 比特：
  - δ, ε 取常数时，Morris++ 算法需要的空间为 $O\left(\frac{1}{\varepsilon^2} \cdot \log \frac{1}{\delta} \cdot \log \log n\right)$，相比较 Morris+ 算法的 $O\left(\frac{1}{\varepsilon^2} \cdot \frac{1}{\delta} \cdot \log \log n\right)$ 在 δ 上有级别的提升。

注意：想要以 $\geq (1-\delta')$ 的概率满足以上两点，可以令 $\delta = \delta'/2$。

1.5 后续对空间的改进

是否可以进一步对空间进行改进?

考虑每次以 $\frac{1}{(1+\alpha)^X}$ 的概率将 X 增加 1，输出 $\tilde{n} = \frac{(1+\alpha)^X - 1}{\alpha}$：

- $\alpha = 0$（不太严谨），即 naive 的算法，需要 $O(\log n)$ 比特；
- $\alpha = 1$，即 Morris 算法，需要 $O(\log\log n)$ 比特。

后续的改进如下：

1. [Mor78, Fla85]：令 $\alpha = \Theta(\varepsilon^2\delta)$，算法以 $\geq (1-\delta)$ 的概率输出 n 的 $(1+\varepsilon)$-近似，需要的空间为 $O\left(\log\frac{1}{\varepsilon} + \log\log n + \log\frac{1}{\delta}\right)$；
2. [NY22]：空间为 $\Theta\left(\log\frac{1}{\varepsilon} + \log\log n + \log\log\frac{1}{\delta}\right)$，
    - 为上、下界，空间已不能继续改进。

2 蓄水池抽样（Reservoir Sampling）

2.1 问题定义

考虑一串以数据流形式输入的整数 $a_1, \ldots, a_m \in [n] = \{1, \ldots, n\}$，要求在任意给定时刻，从数据流中均匀随机地采样出一个元素，使用空间越少越好。

可以利用条件概率，设计如下的算法：

1. 在时刻 1，维护 $\tilde{a} = a_1$ 并输出；
2. 在时刻 t，以概率 $\frac{1}{t}$ 将 $\tilde{a}$ 更新为 a_t；否则，令 $\tilde{a}$ 的值不变。输出 $\tilde{a}$。

引理 6. *蓄水池抽样算法在时刻 t 输出的 $\tilde{a}$ 是 $a_1, \ldots, a_t$ 的一个均匀采样，即，对于任意的 $i \in [t]$，$\Pr[\tilde{a}_{t_0} = a_i] = \frac{1}{t}$。*

Proof. 当 $t = 1$ 时，算法以概率 1 输出 a_1；

假设当 $t = t_0$ 时，算法输出 $a_1, \ldots, a_{t_0}$ 的一个均匀采样，记输出的值为 $\tilde{a}_{t_0}$，则有对于任意的 $i \in [t_0]$，$\Pr[\tilde{a}_{t_0} = a_i] = \frac{1}{t_0}$。

则当 $t = t_0 + 1$ 时，记输出的值为 $\tilde{a}_{t_0+1}$，由算法描述，

$$\Pr[\tilde{a}_{t_0+1} = a_{t_0+1}] = \frac{1}{t_0 + 1}$$

且对于任意的 $i \in [t_0]$，

$$\Pr[\tilde{a}_{t_0+1} = a_i] = \Pr[\tilde{a}_{t_0+1} = \tilde{a}_{t_0}] \cdot \Pr[\tilde{a}_{t_0} = a_i] = \frac{t_0}{t_0 + 1} \cdot \frac{1}{t_0} = \frac{1}{t_0 + 1}$$

综上，$\tilde{a}_{t_0+1}$ 是 $a_1, \ldots, a_{t_0+1}$ 的一个均匀采样。由数学归纳法，引理得证。 ■

空间： 因为算法需要维护 $\tilde{a}$ 和时间 t，所以需要 $O(\log n + \log m)$ 比特的空间。

References

[Fla85] Philippe Flajolet. Approximate counting: a detailed analysis. *BIT Numerical Mathematics*, 25(1):113–134, 1985.

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

[NY22] Jelani Nelson and Huacheng Yu. Optimal bounds for approximate counting. In *Proceedings of the 41st ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 119–127, 2022.