程序设计实习（实验班-2025春）
典型随机算法选讲

授课教师：姜少峰

助教：管晏如、孔启皓、楼家宁、杨卓凡

Email: shaofeng.jiang@pku.edu.cn

快速测试矩阵相乘结果是否正确

- 给出三个n x n实矩阵A B C，测试是否AB = C
- 确定性/暴力算法：$O(n^{\omega})$时间，现在$\omega \approx 2.37188$

达到$\omega = 2$是重大open question

- $O(n^2)$时间随机算法：如果AB = C那么一定返回Yes；但是$AB \neq C$时可能答错
  - 随机选取一个n维随机{0, 1}向量x
  - 测试是否ABx = Cx：是则输出YES否则NO

可以在n^2时间计算ABx：先算y = Bx，再算Ay

正确性

- 如果AB = C：ABx一定等于Cx
- 如果$AB \neq C$
  - 考虑简单情况n = 1，也就是A B C都是实数（而不是矩阵）
  - 比如AB = 1， C = 2，那么不超过0.5概率会判断为相等

尝试分析n = 2?

如何提高成功率?

- 只运行一次的话，相等时一定成功，不相等时成功率只有0.5
- 可以考虑多次运行；但如何汇总结果呢?

- 看到这个序列，应该得出何种结果：YES YES YES YES NO?

典型情况

- 如果相等（Yes）：那么只会看到Yes Yes Yes…
- 如果不相等（No）：至多一半是Yes, 至少一半是No，但是实际上可以略微浮动
- 判据：重复T次，如果有No就回答No，否则回答Yes
    - 分析：不相等时，有$p \leq 0.5$概率输出Yes，那么没有No的概率是p^T
    - 要想达到$1-\delta$成功率，只需要$T = \log 1/\delta$
- 事实上：使用[0, 1]向量可以得到更好的结果（甚至不需要重复实验）

作业三：测试矩阵相乘是否正确

分值：1分；截止日期：3月19日

- 实现课上讲的随机检测方法
- 在openjudge评测，只有所有测试用例都通过才算通过
- 标程使用固定重复次数在设定时间内可以通过
- http://cssyb.openjudge.cn/25hw3/

Rejection Sampling

一个抛硬币问题

- 假设我们有一个均匀硬币，利用该硬币生成一个1/4概率的{0, 1}两点分布
  - 算法是什么？需要抛多少次?
- 如果要求是1/3呢?
  - 注意到：抛掷有限次然后根据抛掷结果输出{0, 1}是不可能得到1/3的！

Rejection Sampling

1/3的解法

- 考虑抛两次的情况，HH TH HT TT四种可能，以不出现TT为条件，则剩下每种出现的概率就是1/3的

TT被reject掉了

- 算法：尝试连续抛掷两次硬币，若TT则重新抛，否则当HH时返回1其他返回0
- 需要抛掷多少次？（需要分析数学期望；最坏情况可以抛任意多次）

本质上：rejection sampling实现了条件概率，但如果reject太多那么会影响性能

比如只有1%的概率不reject，那就大约采样100次才能停下来

Rejection Sampling

一般p

考虑p的二进制表示$p = (p_1, p_2, \ldots)$

for i = 1, 2,…

get random sample $a \in \{0,1\}$

if a = p_i return a

如何解释成rejection sampling?

条件概率视角：返回值对应所有前缀，返回值为1时对应以1结尾的前缀

作业四：Rejection Sampling

分值：2分；截止日期：3月19日

- 题目：推广刚刚的rejection sampling，对于一般的一个目标概率p生成两点分布
- 本题为代码填空/交互题，需要使用我们提供的随机性（即{0, 1}均匀分布）
    - 请不要自己在函数中利用其他随机数发生器
- http://cssyb.openjudge.cn/25hw4/

Rejection Sampling另一个例子

亚线性空间回答“前缀采样”查询

为简化问题，假设n是2的整数次幂

- 输入n个正整数$a_1, \ldots, a_n$；用$a_{\leq i}$表示前缀$a_1, \ldots, a_i$
- 要求支持“前缀采样”查询：给定$i \in [n]$，从$a_{\leq i}$中返回一个均匀随机样本

等价于先均匀随机选$j \in [i]$然后返回a_j

- 额外要求：只在预处理时允许访问$\{a_i\}_i$，并要求使用$\mathrm{poly} \log n$空间
    - 该额外要求在大数据处理中很自然

进阶版本为“后缀采样”/“滑窗采样”，是重要的数据流primitive

    - 注意到没有该额外要求则问题是平凡的

算法

- 预处理：
  - 对$j = 0,1,\ldots,\log n$均匀采样T个$p_1, \ldots, p_T \in [2^j]$，存储$S_j := \{a_{p_k}\}_{k\in[T]}$

T是参数，直接控制失败概率

- 查询：
  - 找到最小的 j 使得$2^j \geq i$，返回S_j中第一个下标 $\leq i$的元素

可解释成rejection sampling：不满足“下标 $\leq i$”就reject
因此返回的是$a_{\leq i}$上的均匀采样

- 失败概率：仅当全部T个样本的下标都 $> i$才会失败
  - 因为$2^{j-1} < i$，所以S_j每个元素有 ≤ 0.5概率被reject，则都reject概率 $\leq 0.5^T$

其他算法：改进与其他tradeoff

算法一：无失败概率，但预处理空间复杂性是期望 $O(\log n)$

- 预处理：

$S \leftarrow \varnothing, i \leftarrow 1, j \leftarrow n+1$
repeat
    sample $p_i \in [1, j-1]$ u.a.r., $S \leftarrow S \cup \{p_i\}$
    $j \leftarrow p_i, i \leftarrow i+1$
until $p_i = 1$
return S

- 查询：找到最大的落在$[1,i]$的S中的元素并返回

其他算法：改进与其他tradeoff

算法二：保证与算法一类似，仅预处理不同

与reservoir sampling想法类似
参考Wikipedia

- 预处理：

$S \leftarrow \varnothing$
for $i = 1, \ldots, n$
    with probability $1/i$, let $S \leftarrow S \cup \{i\}$
return S

- 查询：与算法一相同

- 问题：是否能做到无失败概率、空间复杂性最坏$O(\log n)$?

一个亚线性算法问题：求中位数

- 中位数问题：输入n个整数$A = \{a_1, \ldots, a_n\}$，求这n个数的中位数
  - 中位数：将这些数排序后，居于最中间的，也就是第$\lceil n/2 \rceil$位的数
- 传统上：
  - 可以排序然后直接统计（但需要$O(n \log n)$复杂度）
  - 另外存在一个线性$O(n)$时间的分治算法
  - 不论哪种方法，至少都需要存储、访问整个数据集

亚线性设定

- 然而，如果不允许访问所有数据，仅可以随机访问某些元素呢?
  - 例如，这些数分布式存储在非常多的机器、硬盘上，无法载入内存
  - 但可以用较小的代价去访问指定的某第i个元素
  - 即，算法可以使用一种查询，该查询每次提交一个i，并得到a_i的值
- 是否能用显著小于$O(n)$次的访问来得到对中位数的估计呢？例如$O(\log n)$?

一个基于采样的亚线性算法

只需要$O(1/\epsilon^2)$次采样就能得到$\pm\epsilon n$位误差的估计

- 给定一个很小的误差限ϵ（例如0.01）
- 算法：均匀独立采样$T := O(1/\epsilon^2)$次，找这个采样上的中位数并返回
- Claim：以大常数概率，算法返回的数排在$(0.5 \pm 2\epsilon)n$位上

注意到$0.5n$位是“精确解”，这里有一个$2\epsilon n$位次的误差

简要分析

Chernoff bound. 设$X_1, \ldots, X_n \in [0,1]$是独立、同期望$\mu \geq t$随机变量

令$X := (X_1 + \ldots + X_n)/n$。对任何失败概率$\delta \in (0,1)$，有

$$\Pr\left[|X - \mu| \geq \sqrt{\frac{\log(1/\delta)}{nt}}\mu \right] \leq \delta$$

- 设$\alpha = 0.5 - 2\epsilon$，$\beta = 0.5 + 2\epsilon$
- 考虑三个子集：A_1为排序后在$[1,\alpha n]$位次的数，$A_2 = [\alpha n, \beta n]$，$A_3 = [\beta n, n]$
- 现在考虑一个均匀采样i
  - 各有$0.5 - 2\epsilon$概率$i \in A_1$和$i \in A_3$

典型情况下，T次采样各有$(0.5 - 2\epsilon)T$个点落在A_1和A_3

- Chernoff bound说明大概率A_1和A_3各被采到至多$(0.5 - \epsilon)T$个点

问题：T应该取多少?

- 因此有大概率样本中位数落在A_2上，即某个排位在$(0.5 \pm 2\epsilon)T$的数

作业五：亚线性时间估算分位点

分值：2分；截止日期：3月19日

- 作业题推广到一半的p分位点（中位数是$p = 0.5$的特例）
- 本题为代码填空/交互题，需要用我们提供的查询器来访问数据
- 最后的性能指标不（只）看运行时间，主要看查询次数
- 在标程查询次数的一定范围内都可以通过
- http://cssyb.openjudge.cn/25hw5/

非均匀采样构造数据摘要问题

从均匀采样的局限性说起

- 考虑随机采样的时候，均匀采样是第一个该考虑的
  - 算法设计的一般原则：有简单的就不用复杂的；简单的不够再考虑复杂的

奥卡姆剃刀

- 均匀采样的好处：一般确实可以得到无偏估计，即 期望$\mathbb{E}[X]$ 是正确的
- 不足：未必可以做到 常数概率 落在期望附近！

均匀采样何时不适用?

- 例如：考虑$a_1 = a_2 = \ldots = a_{n-1} = 0, \ a_n = 1$
- 通过均匀采样a_i来估计这列数的和：
- 容易构造无偏估计：例如$Z := na_i$就是无偏估计
- 但需要$\Omega(n)$次采样才能达到常数概率使得误差可控

即使想要以常数概率采到一次非0的a_n
都需要$\Omega(n)$次采样！

一个例题：1-median的数据摘要

- 输入为n个整数$A := \{a_1, \ldots, a_n\}$
- 要求预处理之后高效回答查询：给定整数q，返回

$$\mathrm{cost}(A, q) := \sum_{i=1}^{n} |q - a_i|$$

这个叫做1-median查询，q就是一个候选的median点

- 本题虽然存在基于二分查找的做法，但这里介绍一种采样做法
  - 重点：这种做法可以推广，尤其是可以处理高维

摘要数据结构

我们的算法要找到一个$S \subseteq A$以及对应的权重$w(x)$，使得

$$\forall q, \qquad \sum_{a \in S} w(a) \cdot |q - a| \in (1 \pm \epsilon) \cdot \mathrm{cost}(A, q)$$

即S是A的一个数据摘要，使得任何查询点q上的1-median值都能得到近似

如何构造S?

- 刚刚提到，均匀采样是不行的，会有$a_1 = \ldots = a_{n-1} = 0$，$a_n = 1$的数据
    - 这种数据要求我们要对a_n有更高的采样概率
- 思路：
    - 首先考虑一种均匀采样适用的子集（并在上面做均匀采样构造摘要）
    - 然后将数据集划分成满足上述性质的若干子集，分别构造摘要

特例："环"形子集上的摘要

设这个数据集叫做P

- 考虑一种特殊情况：所有数据点都到某个固定的点c距离在$[r,2r]$之间 $(r > 0)$

几何上这是一个圆环

- 考虑一个均匀采样算法：

m为待定参数

  - 均匀选取P中的m个点，每个点设置权重$|P|/m$，设该随机集合为S

- 因此给一个查询q，我们的估计量就是

$$\sum_{a \in S} w(a) \cdot |q - a|$$

期望分析：无偏估计

结论：$\forall q,\qquad \mathbb{E}\left[\sum_{a\in S} w(a)\cdot|q-a|\right]=\mathrm{cost}(P,q)$

计算过程：

$$\mathbb{E}\left[\sum_{a\in S} w(a)\cdot|q-a|\right]=\frac{|P|}{m}\cdot m\cdot\frac{1}{|P|}\cdot\sum_{a\in P}|q-a|=\mathrm{cost}(P,q)$$

“环”的性质

对$a \in S$设$X_a := |q - a|$，那么我们的估计量就可以写成$X := \frac{|P|}{m} \cdot \sum_{a \in S} X_a$

注意到S是独立采样，所以任意两项X_a, X_b是独立的

该性质与随机性无关

另外由于“环”的性质：对任意$a, b \in P$，X_a, X_b数值上差别不大：

$$X_a - X_b \leq |q - a| - |q - b| \leq |a - b| \leq 2r$$

利用了绝对值不等式（事实上是三角形不等式）
此性质恒成立，与随机性无关

利用相加误差版本的Chernoff Bound

Hoeffding inequality. 设独立随机变量$X_1, \ldots X_m \in [s, t]$。令$X := \sum_i X_i$，则

$$\Pr[X - \mathbb{E}[X] \geq z] \leq 2\exp\left(-\frac{2z^2}{m(t-s)^2}\right)$$

Chernoff bound这里是$t \cdot \mathbb{E}[X]$

经过计算可得$\sum_{a \in S} w(a)|q - a| \in \mathrm{cost}(P, q) \pm \epsilon |P| r$

说明均匀采样大概率产生相加误差$\epsilon |P| r$

需要$1 - \delta$成功率时应当取$m = O(1/\epsilon^2 \cdot \log(1/\delta))$

如何划分成环及处理相加误差$\epsilon|P|r$

总结一下之前的：

- P中的点都到某个c距离在$[r,2r]$之间
- 在P上运行均匀采样得到的S对任何查询q有相加误差$\epsilon|P|r$

结论：c取A的中位数即可

- 解决相加误差：选c为最优的1-median，即c最小化$\text{cost}(A,c)$
- 此时，选$r_0 := \epsilon \cdot \text{cost}(A,c)/n$，$r_i := r_0 \cdot 2^i$，定义$P_i$为A中到c距离$[r_i,2r_i]$的点

把A划分成环！

- 设S_i为在P_i均匀采样的点集

P_0，即$[0,r_0]$一段的处理：任选P_0中一点作为S_0，权重设为$|P_0|$，即"缩点"

如何处理相加误差$\epsilon|P|r$

此时：每个P_i上采样的S_i有相加误差$\epsilon|P_i|r_i$

总误差$\epsilon \sum_i |P_i| r_i \leq \epsilon \cdot \sum_i \mathrm{cost}(P_i, c) = \epsilon \cdot \mathrm{cost}(A, c)$

不等号利用了环的性质：P_i每个点到c距离都至少是r

而因为c最小化了$\mathrm{cost}(A, \cdot\,)$，因此对任何查询$q$有$\mathrm{cost}(A, c) \leq \mathrm{cost}(A, q)$

所以对任何查询q，我们的总误差都是$\epsilon \cdot \mathrm{cost}(A, q)$以内

总结：完整算法、结论和参数选取

问题：这个算法得到的S的大小是多少？

- 算法：
  - 先求使$\mathrm{cost}(A, c)$最小化的c（可以找中位数）
  - 定义环P_i并在每个P_i上依照之前进行m次均匀采样得到S_i，并赋予合适权重
  - 将所有S_i求并集得到S返回
- 保证：对任何q，以概率$1 - \delta$有

可以证明一共有$O(\log n)$个环
要对所有环用union bound保证同时成功，需要取
$m = O(1/\epsilon^2 \log(\log n/\delta))$

$$\mathrm{cost}(S, q) \in (1 \pm \epsilon) \cdot \mathrm{cost}(A, q)$$

推广到高维

- 刚刚讲的算法是对于一维输入的，但容易推广到高维
    - 除了c的选取，其他地方的算法都完全不需要改动
- 如何在高维有效选取c?
    - 首先：常数近似的c只会让S的大小增加相应的常数，因此不必找最优的c
    - 算法：均匀随机选取一个数据点作为c，则$\mathbb{E}[\mathrm{cost}(A,c)] \le 2\min_{c'} \mathrm{cost}(A,c')$

可从刚才的推导得出

即期望上是2-近似；可以利用三角形不等式验证

作业六：d维1-median查询

分值：3分；截止日期：3月26日

- 作业为维针对d = 3维的输入的近似查询
- http://cssyb.openjudge.cn/25hw6/

推广到一般的k-median

- k-median即聚类中心不再是单点$c \in \mathbb{R}$，而是k个点的集合$C \subseteq \mathbb{R}^d$
- 此时推广代价函数为：$\mathrm{cost}(A, C) := \sum_{x \in A} \min_{c \in C} \|x - c\|_2$
- 之前的大框架依然适用：
  - 找一个$O(1)$-近似的C
  - 根据C的最近邻法则将数据划分成k个类，然后每个类划分成$O(\log n)$个环
  - 在环上做均匀采样并设置类似的权重，最后取并集即可