* Lecture 7 概率分析 (opt.)

绳伟光

上海交通大学微纳电子学系

2021-10-25

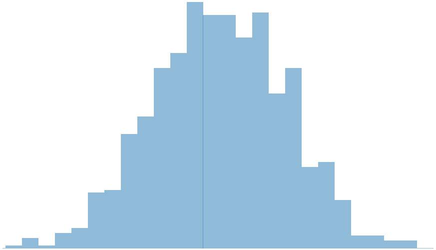
提纲

1. 概率分析简介
2. 雇佣问题
3. 随机排列数组
4. 生日悖论
5. 球与箱子
6. 在线雇佣问题

提纲

概率分析

- 目前为止对算法的分析都是确定性分析，分析的时候并未涉及概率
- 算法的平均情况分析需要考虑概率，即对输入的分布情况进行某种假设，然后以此假设为前提，对一个已知的算法进行此种分布下的复杂性和效率分析
- 概率分析也可以用来进行算法正确性和其他方面的分析



本节仅需简单了解相关方法，非考试内容，主要为后面的随机算法分析进行准备

提纲

雇佣问题

雇佣问题

假设通过中介雇佣一名助理，你会面试中介推荐的应聘者，每面试一个应聘者需要付给中介公司一笔费用；同时，如果当前应聘者比现有的助理要好，则雇佣当前应聘者需要解聘原有的助理并支付一笔赔偿金。试分析该雇佣过程的费用

HIRE-ASSISTANT(n)

```
best ← 0                                ▷ 0 表示一个虚拟应聘者
for i ← 1 to n do
    面试应聘者 i
    if 应聘者 i 好过应聘者 best then
        best = i
        雇佣应聘者 i
```

雇佣问题的初步分析

- 此处 HIRE-ASSISTANT 算法的分析不关注运行时间而是关注面试和雇佣所产生的费用
- 假设面试应聘者 i 的费用为 c_i；雇佣的费用更高，为 c_h
- 假设 m 是雇佣的人数，则 HIRE-ASSISTANT 算法的总费用就是 $O(c_i n + c_h m)$
- 雇佣问题是很多其它问题的良好抽象，比如为球队招募球员的过程等
- 最坏情况出现在所有面试者按照质量的升序进行面试
- 面试所有人的费用 $O(c_i n)$，雇佣所有人的补偿金 $O(c_h n)$
- 通常情况下面试者不是按照严格升序进行，因此具体费用还需深入分析

雇佣问题的概率分析

- 概率分析首先需要对问题的输入分布进行某种假定，未确定输入分布不能应用概率分析技术
- 在已知输入分布的基础上，可以分析算法的平均运行时间以及其它特性
- 对于雇佣问题，可以认为面试者以随机的顺序参加面试，在此基础上，可以为每个面试者 i 分配一个表示优劣的 $rank(i)$，$rank$ 高的优先录用
- 假定有 n 个面试者，则所有的面试者的可能序列共 $n!$ 种，算法运行时的某一特定序列只是 $n!$ 中的一种
- 假定雇佣问题的面试者以随机顺序出现，可以用概率分析技术分析雇佣费用的数学期望

概率分析技术

随机算法

如果一个算法的性能除了受输入数据的影响，还受一个随机数发生器的影响，则称此类算法为随机算法

指示器随机变量

指示器随机变量可以在概率与期望之间建立联系，事件 A 的指示器随机变量定义为

$$I\{A\} = \begin{cases} 1 & \text{if } A \text{ occurs} \\ 0 & \text{if } A \text{ does not occur} \end{cases}$$

概率分析技术 2

引理 1

给定样本空间 S 和随机事件 A，令 $X_A = I\{A\}$，则 $E[X_A] = Pr\{A\}$

证 根据指示器随机变量定义，

$$
\begin{aligned}
E[X_A] &= E[I\{A\}] \\
&= 1 \cdot Pr\{A\} + 0 \cdot Pr\{\overline{A}\} \\
&= Pr\{A\}
\end{aligned}
$$

随机投掷硬币问题

- 随机投掷硬币，对于硬币正面向上的事件，可定义如下指示器随机变量：

$$I\{H\} = \begin{cases} 1 & \text{正面向上} \\ 0 & \text{正面向下} \end{cases}$$

- 令 $X_i = I\{\text{第}i\text{次投掷正面向上}\}$，则 n 次投掷正面向上的总次数 $X = \sum_{i=1}^{n} X_i$，其数学期望为

$$E[X] = E\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} E[X_i] = \sum_{i=1}^{n} 1/2 = n/2$$

用指示器随机变量分析雇佣问题

- 令 X 表示实际面试过程中所雇佣的人数的随机变量，则根据数学期望的严格定义 $E[X]=\sum_{x=1}^{n} x Pr\{X=x\}$，然而此式计算非常繁琐
- 定义指示器随机变量 X_i 代表第 i 个面试者被雇佣，

$$X_i = I\{i\text{被雇佣}\} = \begin{cases} 1 & \text{如果}i\text{被雇佣} \\ 0 & \text{如果}i\text{未被雇佣} \end{cases}$$

- 根据前述引理，$E[X_i]=Pr\{i\text{被雇佣}\}$，根据雇佣算法描述，前 i 个面试者中 i 最好的概率为 $1/i$，因此 $E[X_i]=1/i$
- X 的数学期望为

$$E[X] = E\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} E[X_i] = \sum_{i=1}^{n} 1/i = \ln n + O(1)$$

- 面试 n 个人，平均只有 $\ln n$ 个人得到雇佣 (100 个人只雇佣 4.6 人)
- 雇佣代价 $O(c_h \ln n)$，大幅低于最坏情况 $O(c_h n)$

提纲

基于排序的随机排列数组

概率分析一般要求输入满足某种随机分布，对于具体的算法，可将输入数据随机化以便满足概率分析的要求！

PERMUTE-BY-SORTING(A)

$n \leftarrow A.length$
let $P[1..n]$ be a new array
for $i \leftarrow 1$ **to** n **do**
    $P[i] = \text{RANDOM}(1, n^3)$
Sort A, using P as sort keys

PERMUTE-BY-SORTING 算法的有效性

引理 2

假设 *PERMUTE-BY-SORTING* 算法中产生的 P 中元素为唯一的，则 *PERMUTE-BY-SORTING* 将产生输入的均匀随机排列

证 1) 首先考虑每个元素 $A[i]$ 分配到第 i 个最小优先级的特殊排列，令 E_i 代表元素 $A[i]$ 分配到第 i 个最小优先级的事件，其中 $i \in [1, n]$

2) 考虑 $Pr\{E_1 \cap E_2 \cap \cdots \cap E_{n-1} \cap E_n\}$，这个概率可由下式计算：$Pr\{E_1\} \cdot Pr\{E_2|E_1\} \cdots Pr\{E_i|E_{i-1} \cap E_{i-2} \cdots \cap E_1\} \cdots Pr\{E_n|E_{n-1} \cap \cdots \cap E_1\}$，其中 $Pr\{E_1\}$ 相当于从 n 元素集合随机选取到优先级最小元素的概率，为 1/n; 类似，$Pr\{E_2|E_1\}$ 相当于从 n-1 个元素中等可能选取最小优先级的元素的概率，为 1/(n-1); 依此类推，上式概率为 $(1/n)(1/(n-1)) \cdots (1/2)(1) = 1/n!$

3) 上述结论可以推广到任意优先级的情况，因此推广 2) 的证明方法，任何排列形式的概率都是 $1/n!$，因此算法会产生输入的均匀随机排列

在位随机排列数组

PERMUTE-IN-PLACE(A)

1: $n \leftarrow A.length$
2: **for** $i \leftarrow 1$ **to** n **do**
3: swap $A[i]$ with $A[RANDOM(i, n)]$

引理 3

PERMUTE-IN-PLACE 可计算出一个均匀随机排列

证

- **循环不变式：for** 循环第 i 次迭代之前，对每个可能的 $(i-1)$ 排列，子数组 $A[1..i-1]$ 包含这个 $(i-1)$ 排列的概率是 $(n-i+1)!/n!$(全排列 $n!$ 种，固定前 $(i-1)$ 项，后面有 $(n-i+1)!$ 种)
- **初始化：**$i=1$ 时，$(i-1)=0$(0 排列)，$A[1..0]$ 为空，$A[1..0]$ 包含 0 排列的概率为 $(n-i+1)!/n!=1$，第 1 次循环迭代前循环不变式成立

在位随机排列数组 (续)

证 (续)

- **保持：** 假设第 i 次迭代之前，每种可能的 $(i-1)$ 排列出现在子数组 $A[1..i-1]$ 中的概率是 $(n-i+1)!/n!$，考虑第 i 次迭代之后：假设之前 $i-1$ 个元素为 $\langle x_1, x_2, \cdots, x_{i-1}\rangle$，接下来在 $A[i]$ 里放置某特定元素 x_i(因而形成一个特定排列)。令 E_1 表示在前 $i-1$ 次迭代中已经在 $A[1..i-1]$ 中构造了特殊 $(i-1)$ 排列的事件；令 E_2 表示在 $A[i]$ 放置特定元素 x_i 的事件。则特定排列 $\langle x_1, x_2, \cdots, x_i\rangle$ 出现的概率为：

$$Pr\{E_2 \cap E_1\} = Pr\{E_2|E_1\}Pr\{E_1\} = \frac{1}{n-i+1} \cdot \frac{(n-i+1)!}{n!} = \frac{(n-i)!}{n!}$$

  满足归纳假设

- **终止：** 终止时 $i = n+1$，子数组 $A[1..n]$ 是一个给定 n 排列的概率为 $(n-(n+1)+1)!/n! = 1/n!$。因此 PERMUTE-IN-PLACE 产生一个均匀随机排列。

提纲

生日悖论

生日悖论

一个屋子里人数必须达到多少人，才能使其中两人生日相同的机会达到 50%?

解： 1) 假设屋子里包含 k 个人，设一年包含 $n = 365$ 天，对于 $i \in [1..k]$，令 b_i 表示人 i 的生日，还假设生日均匀分布在一年中。由此，对 $r \in [1..n]$，$Pr\{b_i = r\} = 1/n$

2) 假设生日是独立的，则 i 和 j 生日落在同一天 r 的概率为 $Pr\{b_i = r, \ b_j = r\} = Pr\{b_i = r\}Pr\{b_j = r\} = 1/n^2$

3) 上式 r 可取一年中任意一天，因此 i 和 j 生日相同的概率 $Pr\{b_i = b_j\} = \sum_{r=1}^{n} Pr\{b_i = r, \ b_j = r\} = 1/n$。该结果的更直观的分析在于一旦选定 b_i，b_j 被选在同一天的概率确为 $1/n$

4) 令 A_i 表示对所有 $j < i$，i 与 j 生日不同的事件，则 k 个人生日互不相同的事件 $B_k = \prod_{i=1}^{k} A_i$

生日悖论求解 (续)

解 (续): 5) 存在两个人生日相同的概率为 $1 - Pr\{B_k\}$，又有 $B_k = A_k \cap B_{k-1}$，因此 $Pr\{B_k\} = Pr\{A_k \cap B_{k-1}\} = Pr\{B_{k-1}\}Pr\{A_k|B_{k-1}\}$

6) 初始条件 $Pr\{B_1\} = Pr\{A_1\} = 1$；$B_{k-1}$ 对应 $b_1, b_2, \cdots, b_{k-1}$ 两两不同；B_k 的概率等于 $b_1, b_2, \cdots, b_{k-1}$ 两两不同的概率乘以 $b_k \neq b_i, i \in [1, k-1]$ 的概率；$Pr\{A_k|B_{k-1}\} = (n-k+1)/n$，因为 n 天中有 $n-(k-1)$ 天没被占用，递归计算得：

$$
\begin{aligned}
Pr\{B_k\} &= Pr\{B_{k-1}\}Pr\{A_k|B_{k-1}\} \\
&= Pr\{B_{k-2}\}Pr\{A_{k-1}|B_{k-2}\}Pr\{A_k|B_{k-1}\} \\
&\vdots \\
&= Pr\{B_1\}Pr\{A_2|B_1\}Pr\{A_3|B_2\}\cdots Pr\{A_k|B_{k-1}\} \\
&= 1 \cdot (\frac{n-1}{n})(\frac{n-2}{n})\cdots(\frac{n-k+1}{n}) \\
&= 1 \cdot (1 - \frac{1}{n})(1 - \frac{2}{n})\cdots(1 - \frac{k-1}{n})
\end{aligned}
$$

生日悖论求解 (续 2)

解 (续): 由不等式 $1+x\le e^x$ 可得:

$$Pr\{B_k\}\le e^{-1/n}e^{-2/n}\cdots e^{-(k-1)/n}=e^{-\sum_{i=1}^{k-1}i/n}=e^{-k(k-1)/2n}\le 1/2$$

取对数，$-k(k-1)/2n\le\ln(1/2)$，解二次方程得 $k\ge 23$，即只要有 23 个人以上在一个屋子里，则有两人生日相同的概率大于 50%。 □

用指示器随机变量分析生日悖论

解： 1) 对屋子里 k 个人的每一对 (i,j)，定义随机变量 X_{ij}：

$$X_{ij} = I\{i\text{和}j\text{生日相同}\} = \begin{cases} 1 & \text{如果}i\text{和}j\text{生日相同} \\ 0 & \text{其它} \end{cases}$$

2) 根据前述分析，给定两个人 i 和 j，其生日相同的概率为 $1/n$，因此，$E[X_{ij}] = Pr\{i\text{和}j\text{生日相同}\} = 1/n$；令 X 表示计数生日相同的“两人对”的数目的随机变量，则 $X = \sum_{i=1}^{k}\sum_{j=i+1}^{k} X_{ij}$

3) 求 X 的数学期望：

$$E[X] = E[\sum_{i=1}^{k}\sum_{j=i+1}^{k} X_{ij}] = \sum_{i=1}^{k}\sum_{j=i+1}^{k} E[X_{ij}] = \frac{k(k-1)}{2n}$$

当 $k(k-1) \geq 2n$ 时，$E[X]$ 至少为 1，求解方程得 $k = 28$

和上一种解法的区别?

提纲

1. 概率分析简介
2. 雇佣问题
3. 随机排列数组
4. 生日悖论
5. 球与箱子
6. 在线雇佣问题

球与箱子问题

球与箱子问题

把相同的球随机投到 b 个箱子里，箱子编号 $1, 2, \cdots, b$。每次投球都是独立的，每一次投球，球等可能落在每一个箱子中。球落入任一箱子的概率为 $1/b$。

- 上述投球的过程是一组伯努利实验，每次成功 (球落入指定箱子) 概率 $1/b$
- 落入给定箱子里的球数量服从二项分布 $b(k; n, 1/b)$。如果投 n 个球，落在给定箱子里的球数期望值是 n/b
- 要在给定箱子中投中一个球，投球次数服从几何分布，概率为 $1/b$，平均需要 $1/(1/b) = b$ 次投球
- 使得每个箱子至少有一个球所需投球次数是多少？(礼券收集者问题：收集多少张礼券才能集齐 b 种礼券？)

球与箱子 (续)

求使得每个箱子至少有一个球所需投球次数期望 n

1) 将投球分为多个阶段，第 i 阶段表示从第 $i-1$ 次命中到第 i 次命中之间的投球；第 1 阶段包含第 1 次投球

2) 隐含条件：每次可以保证一次命中，对第 i 阶段的投球，有 $i-1$ 个箱子有球，$b-i+1$ 个箱子是空的，因此，第 i 阶段的每次投球，得到一次命中的概率是 $(b-i+1)/b$

3) 设 n_i 表示第 i 阶段的投球次数，为保证每个箱子至少一个球，必须 b 次命中，所需投球次数 $n=\sum_{i=1}^{b} n_i$。每个随机变量 n_i 服从几何分布，成功的概率是 $(b-i+1)/b$，因此 $E[n_i]=b/(b-i+1)$

4) 根据期望的线性性质：$E[n] = E[\sum_{i=1}^{b} n_i] = \sum_{i=1}^{b} E[n_i] = \sum_{i=1}^{b} \dfrac{b}{b-i+1} = b\sum_{i=1}^{b} \dfrac{1}{i} = b(\ln b + O(1))$

提纲

在线雇佣问题

雇佣策略

为降低成本，选择一个正整数 $k < n$，面试并拒绝前 k 个应聘者，再雇佣其后比前面应聘者分数更高的第一个应聘者；如果不存在更好的应聘者，则雇佣最后一个应聘者

ONLINE-HIRE-ASSISTANT(k, n)

```
1: bestscore ← −∞
2: for i ← 1 to k do
3:     if i.score > bestscore then
4:         bestscore = i.score
5: for i ← k + 1 to n do
6:     if i.score > bestscore then
7:         return i
8: return n
```

确定最佳 k 值 (最佳 K 值约为 $0.37n$)

解： 1) 令 S_i 表示最好的应聘者是第 i 个面试者且面试成功的事件；S 表示成功选择最好应聘者的事件；不同 S_i 不相交，因此 $Pr\{S\} = \sum_{i=1}^{n} Pr\{S_i\}$

2) 当最好应聘者是前 k 个中某一个时，其聘用并不成功，$i \in [1, k]$ 时 $Pr\{S_i\} = 0$，因此 $Pr\{S\} = \sum_{i=k+1}^{n} Pr\{S_i\}$

3) 计算 $Pr\{S_i\}$，即第 i 个应聘者分数最高且应聘成功，这隐含两个条件：1) 事件 B_i：分数最高应聘者在 i 位置；2) 事件 O_i：算法不能从位置 $k+1 \sim i-1$ 中选择任一个应聘者 (前 $i-1$ 个人中的最大者需出现在前 k 位中)

4) 上述 B_i 和 O_i 独立，因为 B_i 仅依赖于位置 i 的值是否大于其它值；O_i 仅依赖于 1 到 $i-1$ 中值的相对次序。因此，$Pr\{S_i\} = Pr\{B_i \cap O_i\} = Pr\{B_i\}Pr\{O_i\}$

5) $Pr\{B_i\} = 1/n$，因为最大值等可能为 n 个位置中任一个；事件 O_i 发生表示最大值在 $i-1$ 个位置中处于前 k 个，因此 $Pr\{O_i\} = k/(i-1)$。因此 $Pr\{S_i\} = k/(n(i-1))$

确定最佳 k 值 (续)

解 (续): 6) 求和:$Pr\{S\} = \sum\limits_{i=k+1}^{n} Pr\{S_i\} = \sum\limits_{i=k+1}^{n} \frac{k}{n(i-1)} = \frac{k}{n}\sum\limits_{i=k+1}^{n}\frac{1}{i-1} = \frac{k}{n}\sum\limits_{i=k}^{n-1}\frac{1}{i}$,由于 $\frac{1}{i}$ 为递减函数,因此 $\int\limits_{m}^{n+1} f(x)\,dx \le \int\limits_{m}^{n} f(x)\,dx \le \int\limits_{m-1}^{n} f(x)\,dx$,

所以 $\int\limits_{k}^{n}\frac{1}{x}dx \le \sum\limits_{i=k}^{n-1}\frac{1}{i} \le \int\limits_{k-1}^{n-1}\frac{1}{x}dx$

7) 求解定积分,$\frac{k}{n}(\ln n - \ln k) \le Pr\{S\} \le \frac{k}{n}(\ln(n-1) - \ln(k-1))$,将 $\frac{k}{n}(\ln n - \ln k)$ 对 k 求导得 $\frac{1}{n}(\ln n - \ln k - 1)$,导数为 0 时取极大值,此时 $Pr\{S\}$ 最大,即 $\ln k = \ln n - 1 = \ln(n/e)$,即 $k = n/e$ 时概率下界最大化,此时 $Pr\{S\} \ge 1/e$。

小结

小结：

- 了解概率分析的方法
- 熟悉指示器随机变量的用法