Lecture Notes for Probability Theory - Class 5

Zhihan Jin

1 Problem (二叉树 *I*) 证明一棵有 n 个叶子的有根二叉哈夫曼树的内部节点有 $n-1$ 个. 二叉哈夫曼树是说每个节点的度数非 0 即 2.

Solution 归纳法是简单而无奈的, 我们将直接证明这个问题.

我们在新建一个节点作为根节点的父亲, 这样只需要构造一个内部节点和叶子的双射即可.

我们把每个内部节点的两个儿子标上左右, 不妨认为根节点只有右儿子. 这样我们也就有了父亲相对于儿子的方向, 例如某个点在其右儿子左边. 考虑每个叶子对应于其向上走第一次向左拐到的点, 也就是把这棵树考虑成二叉排序树之后每个叶子的前驱. 其实也就是中序遍历的前驱和后继. 那么任何两个叶子对应于不同的点, 而每个内部节点又对应于其右儿子向左一直走到的叶子节点, 也就是它的后继. 这样我们就有了一个内部节点和叶子节点的双射, 于是内部节点共有 $n-1$ 个.

之后所有的二叉哈夫曼树的问题, 我们都新建一个点作为新的根节点, 原来的根是其右儿子 (如果区分左右的话).

2 Problem (二叉树 *II*) Counting: n 个叶子节点的二叉哈夫曼树. 每个内部节点的不区分左右儿子, 叶子节点有编号 $1\cdots n$.

Solution 我们把所求设成关于 n 的数列 T_n^l. 显然 $T_2^l = 1$.

考虑一颗 n 个叶子节点的二叉哈夫曼树, 我们在这棵树上新建一个叶子, 则可以接在任意一条边中间. 由于一共有 n 个叶子节点和 n 个内部节点, 所以一共有 $2n-1$ 条边.

另外, 任意两棵 n 个叶子节点的树加一个叶子之后将仍然是不同的, 这一点可以通过递归得到.

于是 $T_{n+1}^l = T_n^l \cdot (2n-1)$. 那么

$$T_n^l = (2n-3)!! \quad n \geq 2$$

3 Problem (二叉树 *III*) Counting: n 个叶子节点的二叉哈夫曼树. 内部节点不区分左右儿子, 叶子节点有编号 $1\cdots n$. 并且给每个内部节点赋予各自不同的 $1, 2, \cdots, n$ 的权值, 使得父亲的权值总是小于儿子.

Solution 我们把所求设成关于 n 的数列 R_n^l. 显然 $R_2^l = 1$.

我们仍然考虑在 n 个叶子节点的树中加入 $n+1$ 号叶子. 设每个节点的权值是 a_i, 叶子节点的 $a_i = n+1$. 考虑一条边 $e = (u, v)$, 其中 u 是 v 的父亲, 那么 $n+1$ 号叶子接在这条边上有 $a_v - a_u$ 种方案. 于是我们希望将每条边的方案加起来. 显然的是每个叶子被加了 2 次, 每个 $2\cdots n$ 的内部节点被加了 1 次, 减了两次.

$$\sum_{e=(u,v)} [u \text{ is father of } v](a_v - a_u) = n(n+1) - \sum_{i=2}^{n} i - 1 = \binom{n+1}{2}$$

$$R^l_{n+1} = R^l_n \cdot \binom{n+1}{2} = \cdots = \prod_{i=2}^{n+1} \binom{i}{2}$$

$$R^l_n = \prod_{i=2}^{n} \binom{i}{2} = \frac{n! \cdot (n-1)!}{2^{n-1}}$$

这种做法可以通过在树上画水平线轻松得到这个公式, 读者可以自行尝试.

另外, 这个问题还可以用更简单的方式加以解释, 那就是每次找两个叶子接在一个新节点上, 再将这个新节点视作另一个叶子, 这样问题的规模就从 n 变为 $n-1$, 也就是

$$R^l_n = R^l_{n-1} \cdot \binom{n}{2}$$

4 Problem (二叉树 *IV*) Calculating: 按照二叉树 III 每步等概率生成树. 现在给定有 n 个叶子的二叉哈夫曼树 T, 叶子有标号, 内部节点尚未赋值, 问其出现的概率 $P(T)$.

Solution 总方案数为 R^l_n, 我们只需要知道有多少方案能够产生 T.

假如先不考虑叶子节点标号, 一个基于递归的想法是从根节点, 按照 BFS 顺序逐个考虑每个节点 u, 由于每个节点都要在所在子树中最小, 我们设 λ_u 表示 u 的子树中有多少内部节点 (包括自己), 那么点 u 赋值合法的概率是 $\frac{1}{\lambda_u}$. 由于树上的两个点的子树要么交集为空, 要么一方被包含, 所以不同的 u 是独立的. 于是总方案数为 $(n-1)! \cdot \frac{1}{\prod\limits_u \lambda_u}$.

$$P(T) = \frac{2^{n-1}}{n! \cdot \prod\limits_u \lambda_u}$$

当然, 这个问题也有一个稍微复杂一下的解决方案. 还是按照 BFS 序从根向下考虑这个问题, 对于每个点 u, 我们会有一个还没有使用的权值集合, 我们做的只是分配这些权值给 u 这个点并将剩下的权值划分给 u 的两个子树 (子树接收一个权值集合), 并确保 u 的权值在子树中最小. 我们假定 u 的左右儿子分别为 a 和 b, 那么方案数为 $\binom{\lambda_a+\lambda_b}{\lambda_a}$, 其中 $\lambda_u = \lambda_a + \lambda_b + 1$. 那么总方案为 $\prod\limits_{u \text{ is a internal node}} \binom{\lambda_a+\lambda_b}{\lambda_a}$. 考虑到每个除了根以外的内部节点都贡献了 $(\lambda_u - 1)!/(\lambda_u)! = 1/\lambda_u$, 我们有

$$\prod_{u \text{ is a internal node}} \binom{\lambda_a + \lambda_b}{\lambda_a} = \frac{(n-1-1)!}{\prod\limits_{u \text{ is a internal node but not root}} \lambda_u} = (n-1)! \cdot \frac{1}{\prod\limits_u \lambda_u}$$

剩余步骤同上.

Homework [概率链]

现在有 17 个点, 分别是 $(1,1,1), (1,1,2), (1,1,3), (1,2,1), (1,2,2), (1,3,1), (1,3,2), (1,4,1), (2,1,1), (2,1,2), (2,2,1), (2,2,2), (2,3,1), (3,1,1), (3,1,2), (3,2,1), (4,1,1)$. 我们

把点按 $1\cdots17$ 编号, 并建立 $\{1\cdots17\}$ 到 $\{1\cdots17\}$ 的双射 f(也就是每个点映射到一个整数), 使得 $\forall u,v\in\{1\cdots17\}\quad u.x\le v.x\quad\&\quad u.y\le v.y\quad\&\quad u.z\le v.z\Rightarrow f_u\le f_v$, 此时我们称 u 控制了 v.

Counting: $I\subseteq\{1\cdots17\}$, 使得 $P(\bigcap\limits_{u\in I}E_u)=\prod\limits_{u\in I}P(E_u)$, 其中 E_u 表示 f_u 是 u 控制的点中最小的事件.

5 Problem (二叉树 *III* 另解) 问题同二叉树 III

Solution 我们尝试证明如下等式,

$$(n-1)!=R_n^l/n!\cdot2^{n-1}$$

等式右边说的是 Counting 有 n 个叶子节点, 叶子节点无标号并且内部节点区分左右儿子有标号 a_i 的二叉哈夫曼树.

我们对内部节点中序遍历定顺序 x_i, 这样每个内部节点左子树右子树就分别是两个区间. 仔细思考 a 的取值, 不难看出 u 的左儿子和右儿子都是所在区间的最大值, 如果没有就直接连叶子. 更准确的说, 这是笛卡尔树的构造, 于是我们对 $n-1$ 个内部节点任意标号, 每次找当前最大节点做根递归左右区间总能完成不同的构造. 这样我们就有了一个树到 $n-1$ 个元素的置换的双射, 那么所求树的个数也就是 $(n-1)!$.

6 Problem (二叉树 *V*) 按照如下规律构造 n 个叶子的二叉哈夫曼树. 一开始只有 1 个点, 每次随机选择一个叶子在下面接两个完全一样的叶子, 最终形成了 T_1 和 T_2 左右两个子树. Calculating: $P(T_1\text{ has }k\text{ leaves})$

当然, 这个答案其实不依赖于 k, 所以只需要证明等概率就可以了.

Solution 把这个问题做成计算题并不十分好玩, 我提供一种组合证明. 我们可以先在根上操作一次, 使得左右子树各自至少有一个叶子.

我们构造这么一个模型, 有一个黑白序列, 一开始有两个元素, 1 后面接 2. 1 代表着左儿子, 黑色, 1 代表右儿子, 白色. 对于第 k 次操作, 我们准备插入 $k+2$ 并着色. 我们随机选择某个白点或者黑点. 如果我们选择了黑点 x, 则 $k+2$ 着黑色并且插在其左侧; 若我们选择了白点, 则着白色并插入其右侧. 这样我们最终能够得到一个长度为 n 的序列, 每个点有颜色, 且分割线是 1-2 中间, 1 及其左侧为黑, 2 及其右侧为白.

每次插入一个点的过程就是在选择一片叶子分裂, 黑色白色点的个数就维护了当前的树左子树的叶子数和右子树的叶子数. 事实上, 给定任意一个 1 恰好在 2 左侧的 n 置换, 我们只需要按照数字从小到大构造就能够构造出不同的树来. 于是形成了双射. 那么 T_1 有 k 个叶子就相当于找有多少 n 元置换使得 1 恰好在 2 左侧一个位置并且 1 极其左侧一共 k 个数字, 这个数字显然是和 k 无关的, 也就是 $\binom{n-2}{k-1}(k-1)!(n-k-1)!=(n-2)!$, 那么

$$P(T_1 \text{ has } k \text{ leaves}) = \frac{1}{n-1}$$

Homework 构造组合模型证明 n 个元素的置换中 1 所在环长为 k 的概率和 k 无关.