arXiv:2304.02563v2 [stat.ME] 6 May 2024

用于狄利克雷过程混合断棒推断的 转码采样器*

Carlo Vicentini

摘要

狄利克雷过程混合模型的 MCMC 后验链由断棒吉布斯采样器生成，其混合速度较慢；相比之下，基于 Pólya urn 表示的塌缩吉布斯采样器具有更短的积分自相关时间（IAT）。

我们研究了上述两种采样器如何编码簇成员信息，并提出了用于在不同编码之间切换的 转码算法。我们还开发了 转码采样器：它使用任意高效采样器（例如塌缩吉布斯采样器）进行后验划分推断，随后通过转码算法将其转为断棒表示，从而在保留高效采样器较短积分自相关时间的同时，对所有感兴趣的断棍参数进行推断。

转码采样器的实现要比切片采样器简单得多；它既能继承折叠吉布斯采样器较短的积分自相关时间，也能与独立同分布的后验划分采样器（例如序贯重要性采样器）配合，实现零积分自相关时间。

关键词：狄利克雷过程、马尔可夫链蒙特卡罗、大小偏倚、切片采样器、断棒

1. 引言

众所周知，基于断棍表示对狄利克雷过程混合模型参数进行后验推断时，Gibbs 采样器在局部众数之间移动困难，会导致采样速度变慢。这类采样器称为条件采样器，或断棍采样器。相反，基于狄利克雷过程的 Pólya urn 表示的边际Gibbs 采样器（也称折叠采样器）会将狄利克雷过程边际化，因此在较小的参数空间中运行；众所周知，这类采样器受上述问题的影响较小，积分自相关时间（IAT）也较短。此外，与条件采样器相比，它们对簇成员指示变量采用了不同的整数编码

*本文基于我在达勒姆大学博士论文第 7 章的内容 [1]。（Carlo Vicentini）

Email address: carlo.vicentini@mathstat.net

。为促使断棍采样器更频繁地在局部众数之间移动，从而实现更快的混合，人们提出了各种 Metropolis 跳跃；然而，迄今为止，这些方法的性能都未达到边际采样器的水平。关于这一问题的关键参考文献包括 [2]和 [3]。

我们发现，关于 Pólya urn 的坍缩参数空间与断棍表示的更广参数空间之间关系的研究，文献中尚有空白。前者产生的簇标签按出现顺序编号；由于这些标签具有可交换性，因此没有特定含义。后者的标签则具有明确含义，因为它们指明了每个观测值源自断棍构造中的哪一根棍。我们探讨这两种编码之间的关系，并设计了一种双向推断方法，最终构建出一种新的采样器，可在任一编码下获得的分区后验分布基础上进行扩充，并推断所有感兴趣的断棍参数。

本文结构如下。第 2 节讨论并比较这两种编码约定。第 3 节介绍如何在两种编码之间进行转码：既可以采用确定性方法，也可以通过转码算法进行统计推断。第 4 节以转码算法为基础，引入转码采样器，以对所有断棍参数进行完整的后验推断；本节还讨论其性能。第 5 节概述本文与该领域其他研究之间的关系，第 6 节则给出结论。

2. 编码

在统计学中，为了使信息可用，必须按照某种约定，将有关数据和模型参数的信息编码为标准格式。尽管这一过程通常很简单，但狄利克雷过程（下文简称 DP）的众多替代表示方式提供了多种选择，而这些表示之间的关系十分复杂，尤其是在识别划分时更是如此。这在一定程度上源于划分本身作为数学对象存在若干相互竞争的编码约定，另一方面也因为 DP 的许多替代表示方式通常具有可交换性。

本节讨论两种对 DP 簇成员指示变量进行编码的方法，以及它们在所识别的划分方面为何等价；然而，就其隐含携带的信息而言，其中一种方法的信息表达能力更强。

考虑一个由 n 个观测值构成的狄利克雷过程混合模型（以下简称

狄利克雷过程混合模型），其写法如下：

$$
\begin{aligned}
&y_i \mid \theta_i \sim p\left(y_i \mid \theta_i\right), \\
&\theta_i \mid G \sim G, \\
&G \sim \mathrm{DP}\left(\alpha, G_0\right),
\end{aligned} \tag{1}
$$

其中 $\boldsymbol{y}=(y_1,\ldots,y_n)$ 是 n 观测值的向量，而 $\boldsymbol{\theta}=(\theta_1,\ldots,\theta_n)$ 是其潜在参数的向量； G 同时表示狄利克雷过程及其实现， G_0 称为基测度， α 是精度参数。

θ_i 的重复取值标识观测值的随机簇；通常，对这些簇进行推断是首先采用 DPM 模型的主要目标之一。常见做法是在方程 1 的模型设定中增补额外的潜变量，以将簇成员信息与原子位置分离表达，例如以便实现主要基于 DP 诱导的随机划分的算法，或配合原子位置并非主要关注点的应用。将簇成员信息编码到潜在簇归属指示向量中有多种方式和惯例。

2.1. 按出现顺序编码

关于狄利克雷过程 [4] 的波利亚瓮表示的原始文章忽略了狄利克雷过程原子的取值位置，只关注其概率质量以及波利亚瓮序列所诱导的划分。文章只提到波利亚瓮中的球具有某种颜色 x，并未指出对此采用任何特定的标记约定。在这方面， x 可以属于任何集合，只要该集合是可数无限的；原则上，该集合甚至可以由用于标识无限种颜色的词语或描述语组成，不过这样做并不实用，因为据我们所知，没有任何自然语言拥有无限多的描述语来对应无限种颜色。

对从瓮中取出的球的颜色信息进行编码，一种更实用但绝非唯一的方式是使用自然数。例如， [5] 讨论双色波利亚瓮时，就是将 1 指定为白球，0 指定为黑球。同样，在 n色波利亚瓮中，可以用$\mathbb{N}_n$中的整数表示颜色归属，并将其映射到特定颜色。然而，如前所述，列出所有可能的颜色并将其映射到 $\mathbb{N}$ 这一策略并不适用于狄利克雷过程，因为它具有无限维性。

因此，在无限维 Pólya urn 设定中，通常的做法是完全摒弃颜色类比，并通过$\mathbb{N}$对簇成员关系进行编码，同时按照簇在抽样过程中出现的顺序为其编号。给定一个 n维抽样，簇成员关系通过向量 $\boldsymbol{s}=(s_1,\ldots,s_n)$编码，其定义使得 $A_j \equiv \{i: s_i=j,\ i\in\mathbb{N}_n\}$。由此得到一种方案，其中 $s_1=1$始终成立； $s_2=1$ 如果从瓮中抽出的第二个球与第一个球颜色相同，而$\mathbb{N}$则在其他情况下成立，依此类推：

颜色相同，而 $s_2 = 2$ 则在其他情况下成立，依此类推：

$$p(s_1 = 1) = 1,$$
$$p(s_i \mid s_{i-1}, \ldots, s_1) = \frac{\alpha}{\alpha + i - 1}\delta_{k_{i-1}+1}(s_i) + \sum_{l=1}^{k_{i-1}} \frac{n_{i-1,l}}{\alpha + i - 1}\delta_l(s_i), \quad (2)$$

其中，k_{i-1} 和$n_{i-1,\ l}$ 分别表示前 $i-1$ 个观测值中的簇数以及簇 l 的大小。

2.2. 按断棒顺序编码

“断棒 ”一词指的是以下构造，其中 w_1，$w_2, \ldots$可视为从一根长度为1的棍子上折下的片段：

$$\begin{aligned} w_1 &:= v_1, \\ w_h &:= v_h \prod_{l<h} (1 - v_l), \quad h = 2, 3, \ldots \\ v_h &\sim \mathrm{Beta}(1, \alpha), \quad h = 1, 2, \ldots, \\ m_h &\sim G_0. \end{aligned} \quad (3)$$

可以证明 [6]，下列随机测度的分布为狄利克雷过程（α，G_0）：

$$G := \sum_{h=1}^{\infty} w_h \delta_{m_h}. \quad (4)$$

在狄利克雷过程文献中，我们主要看到两种对断棍过程产生的聚类方案进行编码的方法。第一种方法按照观测值所抽取的棍子的出现顺序，为 $\mathbb{N}_k$ 中的簇成员指示向量赋予整数。例如，给定 $\boldsymbol{w}$，如果第一个抽取的观测值来自长度为 w_h的棍子，则将其标记为1；此后从同一根棍子抽取的观测值也标记为1。接下来，从不同于 w_h的任意棍子抽取的观测值，以及此后从同一根棍子抽取的观测值，均标记为2，依此类推。由此得到的簇成员指示变量序列显然与由Pólya urn产生的序列等价，这种方法称为按出现顺序（第2.1节讨论的也是这种方法）。例如，Pitman的许多颇具影响力的文章都采用了这种编码方式，并在其中讨论了断棍过程（例如，参见 [7]的第3.1节）。

另一种方法是按照方程 3 中的定义，按构造顺序为各个棒编号，然后为每个观测分配其抽样所用棒的编号。用符号表示，考虑方程 3，并通过向量 $\boldsymbol{r} = (r_1, \ldots, r_n)$ 表示簇成员关系，其中

$$p(r_i = h \mid \boldsymbol{w}) = w_h, \quad h = 1, 2, \ldots$$

尽管这种编码方法在贝叶斯非参数文献中随处可见，却没有专门的名称来指代它，因此我们将其命名为截断棍顺序。

我们发现，按断棒顺序编码比按出现顺序编码包含更多信息，因为后者无论如何都已隐含在断棒索引的呈现顺序[1]中。按断棒顺序编码的有序向量 $\boldsymbol{r}$ 可以通过检查沿有序序列逐步推进时， $\boldsymbol{r}$ 中的哪些元素是新出现的，从而确定性地还原为其出现顺序（$\boldsymbol{s}$）；但反过来则不成立，需要进行推断，尝试从 $\boldsymbol{s}$还原出 $\boldsymbol{r}$。

2.3. 大小偏倚随机排列

出现顺序的概念与大小偏倚的概念密切相关。 $\boldsymbol{w} = (w_1, w_2, \ldots)$的大小偏倚随机排列 $\tilde{\boldsymbol{w}} = (\tilde{w}_1, \tilde{w}_2, \ldots)$定义如下（例如参见 [8]）。首先，从

$$p\left(\tilde{w}_1 = w_h \mid \boldsymbol{w}\right) = w_h, \quad h = 1, 2, \ldots,$$

然后从, $\tilde{w}_3, \ldots$ 中抽取 $\tilde{w}_2$

$$p\left(\tilde{w}_{k+1} = w_h \mid \tilde{w}_1, \ldots, \tilde{w}_k, \boldsymbol{w}\right) = \frac{w_h \ \mathbb{1}\left[w_h \neq \tilde{w}_i, \text{for } 1 \leq i \leq k\right]}{1 - \tilde{w}_1 - \tilde{w}_2 - \ldots - \tilde{w}_k}, \ h = 1, 2, \ldots \tag{5}$$

对于 $k \geq 1$。

一种等价的定义 [9] 是从参数为 $\boldsymbol{w}$的分类分布中抽取指标 $I_1, I_2, \ldots$，并将 $I_1, I_2, \ldots$中的不同值（˜ ˜）按出现顺序记为 $I_1, I_2, \ldots$。于是，$(\tilde{w}_1, \tilde{w}_2, \ldots) := \left(w_{\tilde{I}_1}, w_{\tilde{I}_2}, \ldots\right)$ 是 $\boldsymbol{w}$的大小偏倚排列。

$\boldsymbol{w}$ 的一个重要性质是它对大小偏倚随机排列 [10, 9, 11, 12], 不变，这意味着随机变量 $\boldsymbol{w}$ 及其大小偏倚随机排列 $\tilde{\boldsymbol{w}}$ 在分布上相等:

$$\boldsymbol{w} \stackrel{d}{=} \tilde{\boldsymbol{w}}.$$

因为它与其随机大小偏倚排列具有相同的概率分布，所以序列 $w_1, w_2, \ldots$ 被称为处于大小偏倚顺序。对 $\boldsymbol{w}$ 进行任意次数重复的大小偏倚随机排列所产生的随机变量，仍与 $\boldsymbol{w}$具有相同的概率分布。

3. 转码

本节介绍如何将簇成员指示变量从一种编码转换为另一种编码。可以根据断棒标签确定性地获得出现顺序标签，而反向转换则不那么直接，需要进行统计推断。

[1]这是基于每个数据点的出现顺序已知并反映在数据集中的假设。

3.1. 从断棒到出现顺序

回想一下，$\boldsymbol{r}$ 中的重复值会在$\mathbb{N}_n$上诱导出一个随机划分 $\{A_1, \ldots, A_k\}$。我们将 A_1、.....、A_k 定义为按出现顺序排列，使得 $1 \in A_1$，并且对于每个 $2 \le i \le k$，$\mathbb{N}_n \setminus (A_1 \cup \ldots \cup A_{i-1})$ 中的第一个元素属于 A_i。向量 $\boldsymbol{s}$ 随后可通过由以下关系诱导的映射 $g : \mathbb{N}^n \mapsto \mathbb{N}_k^n$完全由 $\boldsymbol{r}$ 确定：

$$s_i = j \iff i \in A_j.$$

3.2. 从出现顺序到断棒——转码算法

在包含 n 个观测值和 $K = k_n$ 个观测簇的集合中，将 $\boldsymbol{r}$ 的不同取值按照其在向量 $\boldsymbol{r}$ 中出现的顺序记为$\left(r_1^\star、.....、 r_{k_n}^\star\right)$，如下所示：

$$\begin{aligned} r_1^\star &= r_1, \\ r_j^\star &= r_{\min\left\{i \in \mathbb{N}_n :\, r_i \notin \left\{r_1^\star, \ldots, r_{j-1}^\star\right\}\right\}}, \quad j = 2, \ldots, k_n. \end{aligned}$$

随机向量 $\boldsymbol{r}^\star$ 恰好包含了在 $\boldsymbol{s}$ 的基础上恢复 $\boldsymbol{r}$ 所需的额外信息，因为数对 $(\boldsymbol{r}^\star、\ \boldsymbol{s})$ 与 $\boldsymbol{r}$ 在信息上等价。我们进一步用 $\tilde{\boldsymbol{w}}$ 表示在数据抽样过程中按原始断棒被发现的顺序排列的棒长。我们还使用符号 $\tilde{\boldsymbol{n}} = (\tilde{n}_1、.....、\ \tilde{n}_k)$ 来区分按出现顺序排列的簇大小与 $\boldsymbol{n} = (n_1、\ n_2、.....)$，后者是按与 $\boldsymbol{w}$ 相匹配的索引排列的簇大小。

推断 $\boldsymbol{r}$ 自 $\boldsymbol{s}$最简单的方法是使用接受/拒绝算法。我们在附录 C 中介绍了几种效率逐步提高的方法；然而，即使是其中效率最高的方法，仍然十分浪费，以至于在数据集较大时不切实际。这促使我们采用下述方法（转码算法），通过扩展目标空间，实现几乎必然接受的抽样。

我们引入指示变量 $\boldsymbol{t} = (t_1,\ t_2,\ .....)$，以明确描述 $(w_1,\ w_2,\ .....)$ 的出现顺序：

$$t_j = i \Leftrightarrow r_i^\star = j. \tag{6}$$

我们的核心思路是，数对 $(\tilde{\boldsymbol{w}},\ \boldsymbol{t})$ 与 $(\boldsymbol{w},\ \boldsymbol{r}^\star)$ 表达的信息相同；为此，我们借鉴了 [13], 第 3.2 节的思路，并将其反向运用。

现在已定义 $\boldsymbol{t}$，我们可以通过从联合后验分布$p(\boldsymbol{r}^\star,\ \boldsymbol{w},\ \boldsymbol{t},\ \tilde{\boldsymbol{w}} \mid \boldsymbol{s})$ 中抽样来抽取 $\boldsymbol{r}^\star \mid \boldsymbol{s}$：

$$\begin{aligned} p(\boldsymbol{r}^\star, \boldsymbol{w}, \boldsymbol{t}, \tilde{\boldsymbol{w}} \mid \boldsymbol{s}) &= p(\boldsymbol{r}^\star \mid \boldsymbol{w}, \boldsymbol{t}, \tilde{\boldsymbol{w}}, \boldsymbol{s}) \cdot p(\boldsymbol{w}, \boldsymbol{t}, \tilde{\boldsymbol{w}} \mid \boldsymbol{s}) \\ &= p(\boldsymbol{r}^\star \mid \tilde{\boldsymbol{w}}, \boldsymbol{t}) \cdot p(\boldsymbol{w}, \boldsymbol{t} \mid \tilde{\boldsymbol{w}}, \boldsymbol{s}) \cdot p(\tilde{\boldsymbol{w}} \mid \boldsymbol{s}) \\ &= p(\boldsymbol{r}^\star \mid \tilde{\boldsymbol{w}}, \boldsymbol{t}) \cdot p(\boldsymbol{w}, \boldsymbol{t} \mid \tilde{\boldsymbol{w}}) \cdot p(\tilde{\boldsymbol{w}} \mid \boldsymbol{s}), \end{aligned} \tag{7}$$

其中，公式 7 中的第一项是一个指示函数，无需进行任何抽样，因为 $\boldsymbol{r}^\star \mid \tilde{\boldsymbol{w}}, \boldsymbol{t}$ 完全由 $(\tilde{\boldsymbol{w}}, \boldsymbol{t})$ 决定；而第二项和第三项则需要抽样。

该算法包含三个步骤（顺序与公式 7 中三个因子的顺序相反）：

1. 抽取 $\tilde{\boldsymbol{w}} \mid \boldsymbol{s}$，2. 抽取 $\boldsymbol{w},\ \boldsymbol{t} \mid \tilde{\boldsymbol{w}}$，3. 确定性地得到 $p(\boldsymbol{r}^\star \mid \tilde{\boldsymbol{w}},\ \boldsymbol{t})$，因此得到 $\boldsymbol{r} \mid \boldsymbol{w},\ \boldsymbol{t},\ \tilde{\boldsymbol{w}},\ \boldsymbol{s}$。

尽管该算法确实涉及无限长度的向量，但从实际角度来看，只需抽取有限个元素，因此该算法仍可精确执行。

3.2.1. 抽取 $\tilde{\boldsymbol{w}} \mid \boldsymbol{s}$

$\tilde{\boldsymbol{w}} \mid \boldsymbol{s}$ 的后验分布可表示为

$$\begin{aligned} p(\tilde{\boldsymbol{w}} \mid \boldsymbol{s}) &= \frac{p(\boldsymbol{s} \mid \tilde{\boldsymbol{w}})\, p(\tilde{\boldsymbol{w}})}{p(\boldsymbol{s})} \\ &= \frac{1}{p(\boldsymbol{s})} \left(\prod_{i=1}^{k} \tilde{w}_i^{\tilde{n}_i - 1} \right) \left(\prod_{i=1}^{k-1} \left(1 - \sum_{j=1}^{i} \tilde{w}_j \right) \right) \\ &\qquad \frac{\alpha^k \left(1 - \tilde{w}_1 - \ldots - \tilde{w}_k\right)^{\alpha - 1}}{\left(1 - \tilde{w}_1\right) \ldots \left(1 - \tilde{w}_1 - \ldots - \tilde{w}_{k-1}\right)} \\ &= \frac{\alpha^k}{p(\boldsymbol{s})} \left(\prod_{i=1}^{k} \tilde{w}_i^{\tilde{n}_i - 1} \right) \left(1 - \tilde{w}_1 - \ldots - \tilde{w}_k\right)^{\alpha - 1}, \end{aligned} \tag{8}$$

其中，上述第二步用到了 [14] 中的推论 7 以及以下事实：

$$p(\tilde{n}_1, \ldots, \tilde{n}_k \mid \tilde{\boldsymbol{w}}) = \left(\prod_{i=1}^{k} \tilde{w}_i^{\tilde{n}_i - 1} \right) \left(\prod_{i=1}^{k-1} \left(1 - \sum_{j=1}^{i} \tilde{w}_j \right) \right). \tag{9}$$

我们回顾一下，$\boldsymbol{w}$ 和 $\tilde{\boldsymbol{w}}$ 具有相同的分布。前者可用 $\boldsymbol{v}$ 表示，其中每个元素先验服从贝塔分布；因此后者也可以这样表示，我们将这些等价项记为 $\tilde{\boldsymbol{v}}$。

回忆一下，该变换的雅可比矩阵行列式 $|J|$

$$v_i = \frac{w_i}{1 - \sum_{l<i} w_l}$$

is

$$|J| = \prod_{i=1}^{k-1} \left(1 - v_i\right)^{k-i},$$

我们应用它，从 $p(\tilde{\boldsymbol{w}} \mid \boldsymbol{s})$ 推导出 $p(\tilde{\boldsymbol{v}} \mid \boldsymbol{s})$：

$$\begin{aligned} p(\tilde{\boldsymbol{v}} \mid \boldsymbol{s}) &= \frac{\alpha^k}{p(\boldsymbol{s})} \left(\prod_{i=1}^{k} \tilde{v}_i^{\tilde{n}_i - 1} \prod_{l<i} (1-\tilde{v}_l)^{\tilde{n}_i - 1} \right) \left(\prod_{i=1}^{k} 1 - \tilde{v}_i \right)^{\alpha-1} \prod_{i=1}^{k-1} (1-\tilde{v}_i)^{k-i} \\ &= \frac{\alpha^k}{p(\boldsymbol{s})} \left(\prod_{i=1}^{k} \tilde{v}_i^{\tilde{n}_i - 1} \prod_{l<i} (1-\tilde{v}_l)^{\tilde{n}_i - 1} \right) (1-\tilde{v}_k)^{\alpha-1} \prod_{i=1}^{k-1} (1-\tilde{v}_i)^{\alpha+k-i-1} \\ &= \frac{\alpha^k}{p(\boldsymbol{s})} \left(\prod_{i=1}^{k} \tilde{v}_i^{\tilde{n}_i - 1} \right) \prod_{i=1}^{k} (1-\tilde{v}_i)^{\sum_{l>i} \tilde{n}_l + \alpha - 1}, \end{aligned}$$

从而得到：

$$\tilde{v}_i \mid \boldsymbol{s} \sim \mathrm{Beta}\left(\tilde{n}_i, \alpha + \sum_{l>i} \tilde{n}_l \right).$$

这给出了 $\tilde{\boldsymbol{w}} \mid \boldsymbol{s}$ 和 $\tilde{\boldsymbol{v}} \mid \boldsymbol{s}$ 的联合分布。它们的边缘分布与 [15], 的结果一致（正如应当如此），根据该结果，$\tilde{w}_1 \mid \boldsymbol{s} \sim \mathrm{Beta}(\tilde{n}_1, \alpha + n - \tilde{n}_1)$，并且根据可交换性，

$$\tilde{w}_i \mid \boldsymbol{s} \sim \mathrm{Beta}\left(\tilde{n}_i, \alpha + n - \tilde{n}_i\right),$$

这也可以在我们的设定中通过对干扰项积分消去，由公式 8 得到。这也与 [16] 的定理 1 相吻合（ [17], 的定理 1 和推论 20 中也对此作了解释，并结合 $\tilde{\boldsymbol{w}}$ 对其进行了更好的背景说明），根据该定理，

$$\left(\tilde{w}_1, \ldots, \tilde{w}_k, 1 - \sum_{j=1}^{k} \tilde{w}_j \mid \tilde{n_1}, \ldots, \tilde{n_k} \right) \sim \mathrm{Dirichlet}\left(\tilde{n}_1, \ldots, \tilde{n}_k, \alpha\right), \tag{10}$$

根据狄利克雷过程的定义。

3.2.2. 抽样 $\boldsymbol{w},\ \boldsymbol{t} \mid \tilde{\boldsymbol{w}}$

根据定义（见公式 6），我们有

$$w_h = \tilde{w}_{t_h},$$

因此， $\boldsymbol{w}$ 与 $\tilde{\boldsymbol{w}}$ 的区别仅在于其元素的顺序，该顺序由$\boldsymbol{t}$给出。因此，此步骤只需对 $\boldsymbol{t} \mid \tilde{\boldsymbol{w}}$ 进行抽样，这也就隐式确定了 $\boldsymbol{w} \mid \tilde{\boldsymbol{w}}$ 。

由于 $\boldsymbol{w}$ 在大小偏倚排列下保持不变（见 [9] 及本文第 2.3 节），且 $\tilde{\boldsymbol{w}}$ 是它的大小偏倚排列，因此我们有

$$(\boldsymbol{w} \mid \tilde{\boldsymbol{w}} = \boldsymbol{\omega}) \overset{d}{=} (\tilde{\boldsymbol{w}} \mid \boldsymbol{w} = \sigma(\boldsymbol{\omega})),$$

对于任意可行的数值向量 $\boldsymbol{\omega}$ 以及任意置换 σ （包括恒等置换）。直观而言，$\boldsymbol{w} \mid \tilde{\boldsymbol{w}} = \boldsymbol{\omega}$ 和 $\tilde{\boldsymbol{w}} \mid \boldsymbol{w} = \sigma(\boldsymbol{\omega})$ 都是对同一组无序权重进行大小偏倚后得到的随机排列。抽样

$\boldsymbol{t} \mid \tilde{\boldsymbol{w}}$ 因此涉及与抽取 $\tilde{\boldsymbol{w}} \mid \boldsymbol{w}$相同的过程，即通过公式 5 进行大小偏倚：

$$\begin{aligned} &p\left(t_1 = \tau \mid \tilde{\boldsymbol{w}}\right) = \tilde{w}_\tau, \quad \tau = 1, 2, \ldots, \\ &p\left(t_j = \tau \mid t_1, \ldots, t_{j-1}, \tilde{\boldsymbol{w}}\right) \\ &\quad = \frac{\tilde{w}_\tau \ \mathbb{1}\left[\tau \neq t_i, \text{ for } 1 \leq i \leq j-1\right]}{1 - \tilde{w}_{t_1} - \ldots - \tilde{w}_{t_{j-1}}}, \quad j > 1, \tau = 1, 2, \ldots \end{aligned}$$

3.2.3. 获取 $\boldsymbol{r} \mid \boldsymbol{w}, \boldsymbol{t}, \tilde{\boldsymbol{w}}, \boldsymbol{s}$

如上所述，可以从以下内容中获取 $\boldsymbol{r}^\star \mid \ldots$：

$$r_j^\star = \min\left\{i : t_i = j\right\}, \quad j = 1, 2, \ldots \tag{11}$$

然后，可以由 $\boldsymbol{s}$， $\boldsymbol{r}^\star$ 推导出 $\boldsymbol{r} \mid \ldots$。

3.3. 测试转码算法

为了测试后验增广方法的能力，我们进行了以下实验：

- 我们从参数为 $\alpha = 1$ 的断棍过程中抽取长度为 5 的向量 $\boldsymbol{r}$，共进行 100 万次；
- 我们按照第3.1节所述，将 $\boldsymbol{r}$ 的所有样本路径重新编码为 $\boldsymbol{s}$，并从 $\boldsymbol{s}$ 和 $\boldsymbol{r}$ 中丢弃所有满足 $\boldsymbol{s} \neq$(1，1，1，1，2) 的样本路径。这样，我们从概率分布 $p(\boldsymbol{r} \mid \boldsymbol{s} =$(1，1，1，1，2)) 的一次抽样中得到 50，090 个数据点；
- 我们对 $\boldsymbol{s} =$(1，1，1，1，2) 运行后验增广算法，以反推出 $\boldsymbol{r} \mid \boldsymbol{s} =$(1，1，1，1，2)，并与上述得到的频率进行比较。显然，两者应当一致。我们比较边际频率和联合频率。

结果见表 1。正如预期，概率一致。

4. 转码采样器

下面我们来开发一个采样器，用于对狄利克雷过程混合的断棒构造中所有感兴趣参数的联合后验分布进行采样： $\boldsymbol{r}$、 $\boldsymbol{w}$ 和 $\boldsymbol{m}$。我们将完整的联合后验分布写出并分解如下：

$$\begin{aligned} &p\left(\boldsymbol{r}, \boldsymbol{w}, \boldsymbol{m}, \boldsymbol{s} \mid \boldsymbol{y}\right) \\ &\quad = p\left(\boldsymbol{s} \mid \boldsymbol{y}\right) \cdot p\left(\boldsymbol{r} \mid \boldsymbol{s}, \boldsymbol{y}\right) \cdot p\left(\boldsymbol{w} \mid \boldsymbol{r}, \boldsymbol{s}, \boldsymbol{y}\right) \cdot p\left(\boldsymbol{m} \mid \boldsymbol{w}, \boldsymbol{r}, \boldsymbol{s}, \boldsymbol{y}\right) \\ &\quad = p\left(\boldsymbol{s} \mid \boldsymbol{y}\right) \cdot p\left(\boldsymbol{r} \mid \boldsymbol{s}, \boldsymbol{y}\right) \cdot p\left(\boldsymbol{w} \mid \boldsymbol{r}, \boldsymbol{y}\right) \cdot p\left(\boldsymbol{m} \mid \boldsymbol{r}, \boldsymbol{y}\right). \end{aligned}$$

$p(r_1 = h \mid \boldsymbol{s} = (1,1,1,1,2))$

h	转码	经验
1	0.6660	0.6670
2	0.2449	0.2432
3	0.0677	0.0682
4	0.0162	0.0164
5	0.0039	0.0040
6	0.0009	0.0010
7	0.0003	0.0001
8	0.0001	0.0000
...	...	...

$p(r_5 = h \mid \boldsymbol{s} = (1,\ 1,\ 1,\ 1,\ 2))$

h	转码	经验
1	0.1659	0.1666
2	0.3592	0.3638
3	0.2281	0.2286
4	0.1219	0.1184
5	0.0635	0.0604
6	0.0304	0.0306
7	0.0156	0.0154
8	0.0080	0.0079
...	...	...

$p(\boldsymbol{r} = \boldsymbol{h} \mid \boldsymbol{s} = (1,\ 1,\ 1,\ 1,\ 2))$

$\boldsymbol{h}$	转码	经验
(1,1,1,1,2)	0.3316	0.3350
(1,1,1,1,3)	0.1671	0.1679
(2,2,2,2,1)	0.1326	0.1336
(1,\ 1,\ 1,\ 1,\ 4)	0.0838	0.0823
(2,\ 2,\ 2,\ 2,\ 3)	0.0561	0.0560
(1,1,1,1,5)	0.0426	0.0401
(2,2,2,2,4)	0.0280	0.0265
(3,3,3,3,1)	0.0266	0.0268
...	...	...

表 1： $\boldsymbol{r} \mid \boldsymbol{s} =$的边缘和联合后验分布(1,1,1,1,2)，通过对 $\boldsymbol{r}$ (“经验”)进行100万次模拟以及通过转码采样器（“转码”）进行100,000次迭代获得，针对 $\alpha = 1$。

原则上，第一个因子可以通过任何采样器获得，包括折叠采样器和序贯重要性采样器（参见第 4.1 节）。第二个因子可以通过转码算法获得，如下所示：

$$p(\boldsymbol{r} \mid \boldsymbol{s}, \boldsymbol{y}) = \frac{p(\boldsymbol{y} \mid \boldsymbol{r}, \boldsymbol{s})\, p(\boldsymbol{r} \mid \boldsymbol{s})}{p(\boldsymbol{y} \mid \boldsymbol{s})} = p(\boldsymbol{r} \mid \boldsymbol{s}),$$

转码算法能够生成的结果。第三个因子也是转码算法的副产品。

因此，转码算法可以作为构建转码采样器的基础模块，其完整联合条件分布为：

$$\begin{aligned} &p(\boldsymbol{r}^\star, \boldsymbol{w}, \boldsymbol{m}, \boldsymbol{t}, \tilde{\boldsymbol{w}}, \boldsymbol{s} \mid \boldsymbol{y}) \\ &= p(\boldsymbol{s} \mid \boldsymbol{y}) \cdot p(\boldsymbol{r}^\star, \boldsymbol{w}, \boldsymbol{t}, \tilde{\boldsymbol{w}} \mid \boldsymbol{s}) \cdot p(\boldsymbol{m} \mid \boldsymbol{r}^\star, \boldsymbol{s}, \boldsymbol{y}), \end{aligned} \tag{12}$$

其中，第一个因子可由任意分区后验采样器生成，第二个因子可由转码算法生成，第三个因子只需将 $\boldsymbol{\theta} \mid \boldsymbol{s}, \boldsymbol{y}$ 重新映射为 $\boldsymbol{m} \mid \boldsymbol{r}, \boldsymbol{y}$；若不存在，则可用标准方法对其进行抽样。下文将针对 $\boldsymbol{s} \mid \boldsymbol{y}$ 的抽样算法称为转码采样器的核心采样器 。我们不仅

可以使用直接生成 $\boldsymbol{s} \mid \boldsymbol{y}$ 的采样器作为核心采样器，也可以使用任何其他生成 $\boldsymbol{r} \mid \boldsymbol{y}$的采样器；通过第3.1节所述的方法，总能将其映射回 $\boldsymbol{s} \mid \boldsymbol{y}$。

公式12中的第三个因子通常一开始就可由核心采样器提供，其形式为 $\boldsymbol{\theta} \mid \boldsymbol{s}, \boldsymbol{y}$（例如，见第 4.1 节中提到的折叠算法 2 和算法8）。对于已占用簇，我们有

$$m_{r_i} = \theta_i, \quad i = 1, \ldots, n, \tag{13}$$

而对于未占用簇，我们有

$$m_j \sim G_0, \quad j \notin \{r_1, \ldots, r_n\}. \tag{14}$$

如果核心采样器不返回 $\boldsymbol{m}$ 的后验分布，我们注意到

$$p(\boldsymbol{m} \mid \boldsymbol{r}^\star, \boldsymbol{s}, \boldsymbol{y}) = p(\boldsymbol{m} \mid \boldsymbol{y}, \boldsymbol{r}) \propto p(\boldsymbol{y} \mid \boldsymbol{m}, \boldsymbol{r})\, p(\boldsymbol{m}),$$

这可以通过标准方法获得（例如，使用 Metropolis-Hastings算法）。

总之，转码采样器由以下步骤组成：

1. 使用核心采样器从 $\boldsymbol{s} \mid \boldsymbol{y}$ 生成后验样本（这可以是任何能够生成划分后验分布的采样器；例如，见第 4.1 节）；2. 使用转码算法从 $\boldsymbol{r}, \boldsymbol{w}, \boldsymbol{t}, \tilde{\boldsymbol{w}} \mid \boldsymbol{s}$ 中抽样；3. 如果核心采样器也提供 $\boldsymbol{\theta}$ 的后验分布，则通过公式13和14获得 $\boldsymbol{m} \mid \boldsymbol{\theta}, \boldsymbol{r}, \ldots$；否则，通过其他标准方法（可能使用 Metropolis-Hastings算法）对 $\boldsymbol{m} \mid \boldsymbol{y}, \boldsymbol{r}$ 进行抽样。

为突出 $\boldsymbol{w} \mid \boldsymbol{y}$ 和 $\tilde{\boldsymbol{w}} \mid \boldsymbol{y}$之间的差异，我们加入了基于第4.1.1节所述测试数据集得出的图1：尽管 $\boldsymbol{w}$ 和 $\tilde{\boldsymbol{w}}$ 的先验概率分布相同，但它们的后验分布并不相同。

4.1. 性能测试

本节进行性能测试，以比较各种采样器。下文将介绍用于测试的数据集、性能度量、参与测试的采样算法以及测试结果。

4.1.1. 数据集

我们采用的测试数据集是 [18], 的图钉数据，其顺序与 [19]中的呈现顺序相同。该数据集包含320个观测值，记录图钉的抛掷结果；每枚图钉抛掷9次，并且“若图钉尖端朝上落地，则记录为1”。我们使用具有二项似然的狄利克雷过程混合对其建模，并将 G_0 设为贝塔分布 (1，1)。为便于评估收敛，我们倾向于固定 $\alpha = 1$，而不是为其指定先验分布；关于 $\alpha = 0.2$ 和 $\alpha = 5$ 的计算（未包含

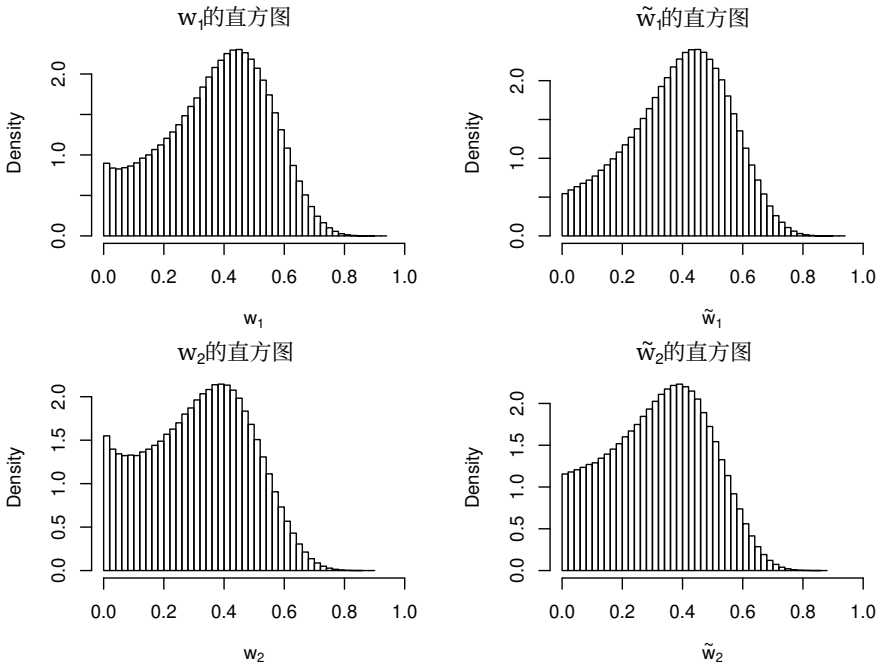

图1：使用折叠算法 2 和转码采样器对图钉数据集进行抽样后得到的 w_1、 $\tilde{w}_1$、 w_2、 $\tilde{w}_2$的后验分布直方图，其中迭代次数为 $\alpha = 1$ ，共 2,000,000 次。

（为简洁起见，此处省略）大体上会得出相同的结论；即使 α 是随机变量，本文概述的方法仍然适用。我们使用原生 R [20] 执行每种算法的 2,000,000 次迭代；计算在 AMD Ryzen 7 7700X 中央处理器上运行。

4.1.2. 效率指标

众所周知，马尔可夫链蒙特卡罗链的抽样会受到不同程度的自相关影响，因此其效率低于样本量相同的独立观测值抽样。 [21], 对此有清晰的阐述，其中讨论了观测量 f的积分自相关时间 这一概念，记作 τ:

$$\tau = \frac{1}{2}\sum_{l=-\infty}^{\infty}\rho_l = \frac{1}{2} + \sum_{l=1}^{\infty}\rho_l,$$

其中， ρ_l 是 f 在滞后阶数 l处的自相关。比较马尔可夫链蒙特卡罗算法时，测量 τ 很重要，因为 2τ 衡量了估计量 f 的方差比独立抽样时大多少。与此前的其他研究一样， [22, 23, 2, 24], 我们也采用 τ 来评估本章所讨论算法的效率。

在包含 N 次迭代的马尔可夫链蒙特卡洛链中，计算 τ 的样本估计值 $\hat{\tau}$, 需要在求和中引入截断值 $M \ll N$ ，并要求“M足够大，使得对于 $|l| > M$, ρ_l 可以忽略不计”；这是因为 [21], 中所述的原因，其本质在于：否则，$\hat{\tau}$ 的方差不会在 $N \to \infty$ 时收敛至 0，而这对于估计量而言是不理想的。因此，我们使用以下公式

$$\hat{\tau} = \frac{1}{2} + \sum_{l=1}^{M}\hat{\rho}_l,$$

其中 $M := \min\{l \in \mathbb{N} : l \geq 10 \cdot \hat{\tau}(l)\}$，遵循 [21, 第 145] 页的建议。

我们接下来用于衡量积分自相关时间的函数量为:

- K_n，即簇的数量。我们选择这一指标以便进行比较，因为以往研究广泛采用了该指标。尽管 K_n 不会因成功的标签切换而直接改变，但其积分自相关时间仍应能反映整体收敛速度;

- 如 [22, 23, 2, 24] 中所定义的偏差:

$$D = -2\sum_{i=1}^{n}\log\left\{\sum_j \frac{n_j}{n} p\left(y_i \mid \theta_j\right)\right\}, \tag{15}$$

先前的研究认为，其使用依据在于它被视为“所有模型参数的全局函数” [24];

- θ_1，该方法在 [22] 中首次提出。从理论上讲，由于我们的测试数据集由320个观测值组成，且仅包含9个唯一值，因此可以针对 $\boldsymbol{y}$ 的每个唯一值，监测九种 i 取值下的 θ_i；但为简洁起见，我们只报告有关 θ_1 的信息；

- r_1，以往的研究从未考虑过这一点，但我们认为它很重要，因为我们试图解决的问题涉及标签切换；

为求完整起见，也包括• w_1 和 m_1 。

我们在测试中使用的采样器有两个是序贯重要性采样器（SIS） [1, 25]。积分自相关时间并不是比较序贯重要性采样器效率的合适指标，因为它们生成的是独立同分布观测值。为了比较 SIS 算法，我们使用其归一化权重的方差以及有效样本量（ESS），计算公式如下：

$$ESS \approx \frac{N}{1 + \mathbb{V}\mathrm{ar}\left(\hat{\boldsymbol{W}}\right)},$$

ˆ 其中，权重估计值 W 经过标准化，使其总和为 1，如下所示：

$$\hat{W}_i = \frac{W_i N}{\sum_{i=1}^{N} W_i},$$

其中， N 是模拟样本的大小。有关此性能指标的更多信息，请参阅 [19, 25]。

4.1.3. 测试候选项

本节简要概述三大类狄利克雷过程混合后验采样器，并在测试中对它们进行比较：既考察它们与转码算法结合使用的情况，也考察它们单独使用的情况。

首先介绍边缘化 采样器。它因采用波利亚瓮表示而得名，该表示将 G 积分掉，从而折叠参数空间。参数空间的缩减使边缘化采样器较断棍采样器不易受到混合缓慢的影响；不过，它既不支持对随机测度 G 进行后验推断，也不支持对其他断棍参数进行后验推断（因为它按出现顺序编码簇成员关系）。关于边缘化采样器的全面参考资料见[22]。通常认为，当 G_0 与 $p\left(y_i \mid \theta_i\right)$ 分别共轭或非共轭时，Neal 算法 2 和算法 8 是性能最佳的边缘化采样器。由于我们采用的测试数据集天然适用于贝塔-二项共轭对，因此表格中只列出 Neal 算法 2。

其次，我们将注意力转向断棍采样器，尤其是切片采样器 。它依赖于方程 3 中的狄利克雷过程表示，并采用断棍编码。它确实能够对折叠采样器无法提供后验推断的一些断棍参数进行后验推断，例如

。由于其参数空间较大，已知该采样器可能导致马尔可夫链蒙特卡洛链混合缓慢，因此文献中提出了一些 Metropolis 跳跃来加速采样。在本文的表格中，我们采用了经 [27]，修改的切片采样器 [26]，并使用标签切换 Metropolis 移动 1 2 [2]，以及移动 3 [3] 对其进行加速。

第三，我们介绍序贯重要性抽样（SIS）。在狄利克雷过程混合模型的背景下， [28], 首次探索了这种方法，并提出了一种用于 $\boldsymbol{s}, \boldsymbol{\theta} | \boldsymbol{y}$的采样器。此后不久， [25],提出了效率更高的变体 S2，它通过积分 $\boldsymbol{\theta}$ 将其消去。这两种算法都按出现顺序对簇成员关系进行编码。最近，我们开发了 SIS 算法 R [1], ，它改用断棍编码。序贯重要性抽样的优点在于它是独立同分布的，因此根据定义，其积分自相关时间最短；缺点是其性能取决于观测值在抽样中的出现顺序，而且在运行采样器之前，没有已知的最优准则可用于对数据排序。本文重点介绍算法 S2，并将其与转码算法结合；同时将算法 R 作为对比。算法 R 单独运行，不与转码算法结合，因为它已经采用断棍编码，无需转码。

4.1.4. 测试结果

我们的测试结果报告于 2。该表包括（从上到下）：

- 以 SIS 采样器 S2 作为其核心采样器的转码采样器；
- SIS 算法 R；
- 以折叠采样器 2 作为其核心采样器的转码采样器；
- 四种变体的切片采样器：不带用于加速链的 Metropolis 移动，以及分别带有 Metropolis 移动 1 、 2 或 3。

需要特别指出的是，转码采样器在构造时便保证继承其核心采样器的积分自相关时间；例如，尽管我们没有在 2 中列出独立折叠算法 2 的积分自相关时间，但可以放心地认为，它与以折叠算法 2 为核心采样器的转码采样器的积分自相关时间完全相同。

正如预期，当核心采样器为独立同分布（i.i.d.）时，转码采样器返回的积分自相关时间（IAT）处于整个范围的最低端（即 ≈ 0.50）——它生成的抽样完全独立同分布。若核心采样器采用的是折叠算法，则核心采样器自然生成的参数，其积分自相关时间按构造恰好与核心采样器相同；而通过增广核心采样器得到的棍棒折断参数，其积分自相关时间则显著短于切片采样器的相应值，无论后者采用哪种 Metropolis 标签切换移动。

算法	ESS 积分自相关时间$_K$		IAT_{w_1}	IAT_{r_1}	$\mathrm{IAT}_{w_{r_1}}$	IAT_{m_1}	IAT_{θ_1}	IAT_D
SIS S2+转码	144,211	0.50	0.50	0.50	0.50	0.50	0.50	0.50
SIS R	131,933	0.50	0.50	0.50	0.50	0.50	0.50	0.50
已折叠2+转码。	N.A.	11.86	5.97	2.49	7.73	0.50	0.55	2.15
切片，无移动	N.A.	75.16	126.00	43.70	36.00	388.12	0.87	6.43
切片，移动1	N.A.	72.16	121.15	32.15	34.34	38.69	0.87	6.65
切片，移动2	N.A.	65.39	73.59	32.96	32.44	226.31	0.86	6.39
切片，移动 3	N.A.	57.37	34.38	13.64	29.63	6.15	0.86	6.24

表 2：在图钉数据集上进行 2,000, 000 次迭代时，转码采样器分别以序贯重要性抽样算法 S2（“SIS S2+转码”）和折叠算法 2（“折叠2+转码”）作为核心采样器，并与切片采样器和序贯重要性抽样算法 R 进行比较，其中 $\alpha = 1$。

5. 与其他工作的关系

归根结底，转码采样器由以下三个主要构件组成：

1. 大小偏倚；2. 方程 9；3. $\boldsymbol{w}, \boldsymbol{t} \mid \tilde{\boldsymbol{w}}$ 的推导，
以及方程 11。

我们已确定另外两种围绕构件 1 和 2 的方法，但它们得出的结论与本文所述不同，因为它们没有使用构件 3。

一种方法是 [29], 开发一种算法，其本质上从

$$p\left(s_i \mid \tilde{\boldsymbol{w}}, \boldsymbol{\theta}^{\star} \ldots\right) \propto \tilde{w}_{s_i} p\left(y_i \mid \theta^{\star}_{s_i}\right), \quad i = 1, \ldots, n, \tag{16}$$

然后通过公式 10 对 $\tilde{\boldsymbol{w}} \mid \boldsymbol{s}$、.....进行抽样，接着在吉布斯方案中对它们的位置进行抽样，并在每次迭代时重新编码簇成员指示变量，使其按出现顺序排列。该算法运行的空间介于折叠算法 2 与切片采样器之间：它比前者更宽，因为还包含 $\tilde{\boldsymbol{w}}$；但比后者更窄，因为其簇成员编码按出现顺序排列。发表于 [29] 的积分自相关时间也反映了这一点：就性能而言，该算法介于折叠算法 2 与切片采样器之间。然而，该算法生成的后验分布既不包含 $\boldsymbol{r}$，也不包含 $\boldsymbol{w}$；这些则由切片采样器返回。因此，与转码采样器不同，它不能替代切片采样器。

另一种是 [30][2], , 其本质上[3] 也按前文所述运行。技术上的区别在于，它不是每次迭代都从 $s_i \mid \ldots$ 公式（16）抽样并按出现顺序重新编码标签，而是添加约束

[2]也与 [31] 有关。[3]该算法还有其他特性和用途，例如适用于有限混合模型。

到抽样公式中，从而使采样器生成的簇成员指示变量从一开始就按构造始终满足 admissible 条件，因此无需重新编码。该算法在与 [29], 相同的空间中运行，并且与上述方法一样，其后验分布也不包括 $\boldsymbol{r}$ 或 $\boldsymbol{w}$ ，且不能替代切片采样器。

6. 结论

本文讨论了按照数据中各簇的出现顺序进行的簇成员编码（这正是由 Pólya urn 构造得到的向量 $\boldsymbol{s}$ 所体现的），以及按照各簇所源自的断棍在断棒构造过程中出现的顺序进行的簇成员编码（这正是由断棍构造得到的向量 $\boldsymbol{r}$ 所体现的）；我们确定，后者包含的信息多于前者。我们还提出了一种简单的确定性方法来推导 $\boldsymbol{s} \mid \boldsymbol{r}$，并通过 转码算法 推断 $\boldsymbol{r} \mid \boldsymbol{s}$。

我们还推导出了转码采样器，它可以与任何能够返回数据的可交换后验划分的其他采样器（即其核心采样器）完全整合，从而推断出所有可能感兴趣的断棍参数，包括 $\boldsymbol{r}$，$\boldsymbol{w}$，$\tilde{\boldsymbol{w}}$ 等。由于转码采样器利用的是 i.i.d. 的转码算法，因此它继承了其所依赖的核心采样器的综合自相关时间。因此，转码采样器能够对断棍参数进行完整的后验推断，同时实现最短的自相关时间（当其核心采用序贯重要性采样器时），或达到与边缘化采样器相同的自相关时间（当其核心采用边缘化采样器时）。众所周知，这些自相关时间远短于切片采样器及其他断棍采样器所达到的自相关时间（即使采用 Metropolis 标签切换步骤来加速切片采样器也是如此）。

附录 A. 一个新的先验采样器，用于 $\boldsymbol{s}$

为更好地理解这两种编码之间的联系，本附录将举例说明我们在第 3.2.2 节中使用的 $\boldsymbol{w} \mid \tilde{\boldsymbol{w}} = \boldsymbol{\omega}$ 与 $\tilde{\boldsymbol{w}} \mid \boldsymbol{w} = \sigma(\boldsymbol{\omega})$ 之间的关系，并展示二者在两个方向上互为大小偏倚随机排列。为此，我们介绍一种用于 $\boldsymbol{s}$ 的新先验采样器，该采样器不依赖方程 2 中的经典波利亚坛构造。具体过程如下：

1. 设置 $s_1 = 1$ 并抽样 $\tilde{w}_1 \sim$ 贝塔分布 $(1, \alpha)$ 根据方程 3（这是可能的，因为 $\tilde{\boldsymbol{w}}$ 是 $\boldsymbol{w}$ 的大小偏倚排列，且后者对大小偏倚排列不变；参见 [9] 和 [15]）。然后，从以下抽样 $s_2, s_3, \ldots$：

$$p\left(s_i \mid s_1, \ldots, s_{i-1}, \tilde{\boldsymbol{w}}\right) = \left(\sum_{j=1}^{k_{i-1}} \tilde{w}_j \delta_j\left(s_i\right)\right) + \left(1 - \sum_{l \leq k_{i-1}} \tilde{w}_l\right) \delta_{k_{i-1}+1}\left(s_i\right), \quad \text{(A.1)}$$

其中 k_{i-1} 是 $s_1, \ldots, s_{i-1}$ 中的簇数。我们从 [7] 的方程 3.4 推导出方程 A.1：

$$p\left(\tilde{n}_1, \ldots, \tilde{n}_k \mid \tilde{\boldsymbol{w}}\right) = \left(\prod_{j=1}^{k} \tilde{w}_j^{\tilde{n}_j - 1}\right) \left(\prod_{j=1}^{k-1} 1 - \sum_{l \leq j} \tilde{w}_l\right). \tag{A.2}$$

直观而言，给定 s_1，……，s_{i-1} 和 $\tilde{w}_1$，……，$\tilde{w}_{k_{i-1}}$，开启一个新簇的概率为 $1 - \sum_{l \leq k_{i-1}} \tilde{w}_l$（见方程 A.2 中的第二个因子）；相反，重复某个已观测簇的概率为 $\tilde{w}_j$（见同一方程中的第一个因子）。每当开启一个新簇时，都根据方程 3 抽取其对应的权重，理由与前文相同。

2. 对 $\tilde{\boldsymbol{w}}$进行大小偏倚排列，以获得 $\boldsymbol{w}$，并隐式地获得 $\boldsymbol{r}$；方法是使用方程 5 的一个变体，其中公式里的 $\boldsymbol{w}$ 和 $\tilde{\boldsymbol{w}}$ 互换。

这个采样器值得注意，原因有二：

- 它以不同于通常呈现方式的方法，从波利亚瓮过程中生成数据点。这种方法使权重能够精确确定；相比之下，基于方程 2 的波利亚瓮采样器根据生成簇的大小来估计权重，且只有渐近正确；

- 它表明 $\boldsymbol{w}$ 和 $\tilde{\boldsymbol{w}}$ 互为大小偏倚排列。虽然 $\tilde{\boldsymbol{w}}$ 由于其与无放回抽样的联系而自然地作为 $\boldsymbol{w}$ 的大小偏倚随机排列出现，$\boldsymbol{w}$ 也可以被视为 $\tilde{\boldsymbol{w}}$的大小偏倚随机排列。

附录 B. 可交换性、EPPF 和 Ewens 抽样公式

我们纳入了一些关于可交换性概念的考虑，该概念适用于随机向量和随机划分，这也有助于支持附录 Appendix C 中的一些结论。

附录 *B.1.* 可交换性

通常与狄利克雷过程的 Pólya urn 构造相关联的主要性质之一是它产生“可交换序列”，虽然在某些条件下这是正确的，但我们想澄清不同类型的编码对此关键性质的影响。

已知具有固定颜色数的有限波利亚瓮方案会生成一列随机簇成员指示变量，这些变量具有可交换性（按德菲内蒂的意义），并且在给定狄利克雷概率测度 [32], 的条件下相互独立；此时，它们按照一个有限集合进行编码，该集合中的元素与每种颜色一一对应。类似地，无限维波利亚

瓮也会生成随机簇成员指示变量序列，这些变量具有可交换性，并且在给定狄利克雷过程测度的条件下相互独立，只要每个标签都唯一标识一种特定颜色。然而，我们发现，使用出现顺序编码对波利亚瓮序列进行编码会破坏其德菲内蒂可交换性，因为这种编码施加了顺序约束。例如，假设一个参数为 α的波利亚瓮序列：它所诱导的随机向量$(s_1,\ s_2,\ .....)$几乎必然不具有可交换性 $s_1 = 1$，而 $p(s_2 = 1) < 1$，因此 $p(s_1,\ s_2) \neq p(s_2,\ s_1)$。可以看出，条件 $\boldsymbol{s} \mid \boldsymbol{r}$ 也不具有可交换性。

相反，断棍编码确实会产生一个德菲内蒂可交换的序列 $\boldsymbol{r}$，因为 $r_i,\ r_j$ 在给定 $\boldsymbol{w}$的条件下相互独立。然而，它的条件 $\boldsymbol{r} \mid \boldsymbol{s}$ 仅具有德菲内蒂部分可交换性，并非完全可交换。

两种编码方法都保留了狄利克雷过程所诱导的随机划分的可交换性：两种编码方法都会产生可交换的随机划分。回顾一下，随机划分是可交换的，当且仅当对于 $\mathbb{N}_k$的每个置换 σ，其概率分布都是对称的：

$$p(n_1, \ldots, n_k) = p(n_{\sigma_1}, \ldots, n_{\sigma_k}).$$

我们在表 B.3 中总结了这些性质。

性质	$\boldsymbol{s}$	$\boldsymbol{s} \mid \boldsymbol{r}$	$\boldsymbol{r}$	$\boldsymbol{r} \mid \boldsymbol{s}$
可交换性	no	no	yes	no
部分可交换性	no	no	yes	yes
其划分的可交换性	yes	yes	yes	yes

表 B.3：以出现顺序（$\boldsymbol{s}$）和断棒顺序（$\boldsymbol{r}$）编码时，簇成员指示向量及其后验分布的可交换性。

附录 *B.2. EPPF* 和 *Ewens* 抽样公式

可交换随机划分 Π_n 可由以下函数描述，该函数称为 可交换划分概率函数 (EPPF)：

$$p(n_1, \ldots, n_k) = \alpha^k \frac{\Gamma(\alpha)}{\Gamma(\alpha + n)} \prod_{j=1}^{k} \Gamma(n_j). \tag{B.1}$$

EPPF 是一个对称函数，它返回某个特定（无序）集合划分的概率，且仅取决于无序块大小；块大小相同的划分具有相同的概率。

公式 B.1 还与下式有关，该式称为埃文斯抽样公式。定义 $M_j := \#\{i\colon n_i = j,\ i = 1, \ldots, k\},\ j = 1, \ldots, n$。那么

$$p(M_1, \ldots, M_n) = n! \frac{\Gamma(\alpha)}{\Gamma(\alpha + n)} \prod_{i=1}^{n} \frac{\alpha^{M_i}}{i^{M_i} M_i!}, \tag{B.2}$$

因为对于公式 B.2 中的每一种配置，都有

$$\frac{n!}{\prod_{i=1}^{n} i!^{M_i} M_i!}$$

B.1 中的配置（参见 [33] 以全面总结 Ewens 抽样公式的用途）。

最后，按出现顺序的组成 $(n_1, \ldots, n_k)$ 的概率为 [7,方程 2.6]:

$$\begin{aligned} p_{\mathrm{OOA}}(n_1, \ldots, n_k) &= \frac{n!}{n_k (n_k + n_{k-1}) \cdots (n_k + \cdots + n_1)} \frac{p(n_1, \ldots, n_k)}{\prod_{i=1}^{k} (n_i - 1)!} \\ &= \frac{n!}{n_k (n_k + n_{k-1}) \cdots (n_k + \cdots + n_1)} \alpha^k \frac{\Gamma(\alpha)}{\Gamma(\alpha + n)}. \end{aligned} \tag{B.3}$$

附录 C. 接受-拒绝转码方法

从按出现顺序编码的簇成员指示变量获得断棒编码的最简单方法是通过接受/拒绝算法。我们在此按效率递增的顺序介绍其变体；然而，即使是最高效的形式，该算法仍然浪费，因此第 3.2 节中转码算法 的价值在于它通过增广目标空间来操作。

附录 *C.1.* 接受-拒绝方法 *1*

为了从 $\boldsymbol{s}$ 推断 $\boldsymbol{r}$，我们写出

$$p(\boldsymbol{r} \mid \boldsymbol{s}) \propto p(\boldsymbol{s} \mid \boldsymbol{r}) p(\boldsymbol{r}), \tag{C.1}$$

并且我们观察到:

- $p(\boldsymbol{r})$ 很容易通过逐步从 $p(r_1)$ $p(r_2 \mid r_1)$ ...中抽样得到，依据:

  $$p(r_1 = h) = \mathbb{E}[w_h] = \frac{\alpha^{h-1}}{(\alpha + 1)^h},$$

  并且，对于 $i = 2$，3，..，依据:

  $$\begin{aligned} p(r_i = h \mid r_1, \ldots, r_{i-1}) &= \mathbb{E}[w_h \mid r_1, \ldots, r_{i-1}] \\ &= \mathbb{E}[v_h \mid r_1, \ldots, r_{i-1}] \prod_{l<h} \mathbb{E}[1 - v_l \mid r_1, \ldots, r_{i-1}], \end{aligned}$$

  其中，条件期望可以从以下内容中获得

  $$v_h \mid \ldots \sim \mathrm{Beta}\left(1 + n_h, \alpha + \sum_{l=h+1}^{H} n_l\right), \tag{C.2}$$

  其中， n_h 是簇 h中的观测值数量，而这又归因于广义狄利克雷分布 [34]的共轭性;

- 等式 C.1 右侧的第一项充当指示函数：

$$p(\boldsymbol{s} \mid \boldsymbol{r}) = \mathbb{1}_{g(\boldsymbol{r})}(\boldsymbol{s}).$$

因此，我们可以从 $p(\boldsymbol{r})$ 中抽取一个提议 $\hat{\boldsymbol{r}}$，若 $g(\hat{\boldsymbol{r}}) = \boldsymbol{s}$，则接受该提议；否则重复尝试。由此得到的接受率等于公式 B.1（埃文斯分布），它给出了 $\boldsymbol{s}$ 的某一种特定配置的概率，其中簇大小为 n_1、. . .、n_k：

$$\alpha^k \frac{\Gamma(\alpha)}{\Gamma(\alpha+n)} \prod_{j=1}^{k} \Gamma(n_j).$$

然而，随着 n 增大，上述概率会迅速趋近于零，很快便变得无法使用；例如，对于一个序列，其中$n = 30$，$\alpha = 1$ 包含 3 个簇，簇大小分别为 (22、7、1)，根据公式 B.1 推导出的理论接受率约为 1.4×10^{-10}，因此该算法相当低效。

Row	$\boldsymbol{s}$	$p(\boldsymbol{s})$	可接受的 $\boldsymbol{r}$ 模式
1	(1，1，1)	0.3333	(a，a，a)
2	(1,1,2)	0.1666	(a，a，b)
3	(1,2,1)	0.1666	(a，b，a)
4	(1，2，2)	0.1666	(a，b，b)
5	(1，2，3)	0.1666	(a，b，c)

表 C.4：接受-拒绝算法 1：对于狄利克雷过程中的所有可能结果 $\boldsymbol{s}$，$\boldsymbol{r}$ 的可接受配置，其中 $n = 3$，$\alpha = 1$。

附录 *C.2.* 接受-拒绝方法 *2*

考虑 $\mathbb{N}$ 的一个置换 σ，并注意到：

- 虽然 $\boldsymbol{s}$ 不是德菲内蒂可交换的（由于其顺序约束），但只要 σ 不改变簇的出现顺序，我们仍有 $p(s_1, \ldots, s_n) = p(s_{\sigma_1}, \ldots, s_{\sigma_n})$；

- $p(r_1，\ldots\ldots，r_n) = p(r_{\sigma_1}，\ldots\ldots，r_{\sigma_n})$，对于任意 σ，因为随机向量 $\boldsymbol{r}$ 是德菲内蒂可交换的。

因此，可以将附录 C.1 中的接受准则放宽为接受所有簇的有序簇大小[4] 向量

[4]当 $\boldsymbol{r}$ 按断棒顺序编码，并按簇的大小排序，使得第一个出现的簇的大小排在第一位，第二个出现的簇的大小排在第二位，依此类推。

$\hat{\boldsymbol{r}}$ 与 $\boldsymbol{s}$的有序簇大小相匹配的情况；这样做会将接受率提高到

$$\alpha^k \frac{\Gamma(\alpha)}{\Gamma(\alpha+n)} \frac{n!}{n_k(n_k+n_{k-1})\cdots(n_k+\ldots+n_1)},$$

如公式 B.3 所示。

我们借助表 C.4 和表 C.5 举例说明，它们展示了当 $n=3$ 时 $\boldsymbol{s}$ 的所有可能结果集合。例如，当 $\boldsymbol{s}=(1,1,2)$ 时，附录 C.1 中的算法旨在涵盖所有配置与 $\boldsymbol{s}=(1,1,2)$ 兼容的 $\boldsymbol{r}$ 结果；表 C.4 中只有一种此类情况（见行 2）。而本节的算法则基于这样一个观察：一旦将 $\boldsymbol{s}$ 和 $\boldsymbol{r}$ 都通过 $\sigma_1=1$，$\sigma_2=3$，$\sigma_3=2$ 进行转换，表 C.4 的行 3 与同一表中的行 2 就完全等价。在这个具体例子中，这样做会使接受比率加倍；在附录 C.1 的例子中（$n=30$，$\alpha=1$；3 个簇的大小分别为 22，7，1），接受率从 1.4×10^{-10} 提高到 0。00416。表 C.5 总结了经过适当重新排列后可以接受的 $\boldsymbol{r}$ 配置。

Row	$\boldsymbol{s}$	$p(\boldsymbol{s})$	可接受的 $\boldsymbol{r}$ 模式
1	(1，1，1)	0.3333	$(a,\ a,\ a)$
2	(1，1，2)	0.1666	(a,a,b)，(a,b,a)
3	(1,2,1)	0.1666	(a,b,a)，(a,a,b)
4	(1,2,2)	0.1666	$(a,\ b,\ b)$
5	(1，2，3)	0.1666	$(a,\ b,\ c)$

表 C.5：接受-拒绝算法 2：在狄利克雷过程中，对于 $\boldsymbol{r}$，$\boldsymbol{s}$ 的所有可能结果所对应的可接受配置，其中 $n=3$，$\alpha=1$。

附录 *C.3.* 接受-拒绝方法 *3*

通过观察到，对于任意置换 σ，τ，只要它们不改变 $(\tau(s_{\sigma_1})\ \ldots,\tau(s_{\sigma_n}))$ 中簇的出现顺序，$p(s_1,\ldots,s_n)=p(\tau(s_{\sigma_1})\ \ldots,\tau(s_{\sigma_n}))$，因为 $\boldsymbol{s}$ 在满足上述条件的情况下对其标签的置换保持不变，因此可以将附录 C.2 中的方法进一步推广。表 C.6 展示了一个例子。

该算法通过生成一个提议 $\hat{\boldsymbol{r}}$，按簇大小升序（或降序）对其进行排序，并与 $\boldsymbol{s}$ 的簇大小排序结果进行比较来运行；如果两者一致，则接受 $\hat{\boldsymbol{r}}$，并对其元素的位置进行置换，使其与 $\boldsymbol{s}$ 的模式相匹配。

该算法的接受率为（见附录 B.2）：

$$\frac{n!}{\prod_{i=1}^{n} M_i!\,(i!)^{M_i}} \cdot \alpha^k \frac{\Gamma(\alpha)}{\Gamma(\alpha+n)} \prod_{j=1}^{k} \Gamma(n_j). \tag{C.3}$$

这是对附录 C.2 中算法的一项适度改进；例如，在与上文相同的情形下（$n=30$，$\alpha=1$；3 个簇，大小分别为 22，7，1），其接受率为 0。0065，高于之前的 0。00416。然而，正如预期的那样，随着 $n=30$ 增大，即使是这个算法也会变得更加低效；事实上，对于样本量为

（例如， [19] 中使用的图钉数据集），对于簇大小（226，75，13，3，2，1），接受率约为七千万分之一，这说明有必要设计一种更好的算法。

Row	$\boldsymbol{s}$	$p(\boldsymbol{s})$	可接受的 $\boldsymbol{r}$ 模式
1	(1,1,1)	0.3333	$(a,\ a,\ a)$
2	(1,1,2)	0.1666	(a,a,b), (a,b,a), (a,b,b)
3	(1,2,1)	0.1666	(a,b,a), (a,a,b), (a,b,b)
4	(1,2,2)	0.1666	(a,b,b), (a,a,b), (a,b,a)
5	(1,2,3)	0.1666	$(a,\ b,\ c)$

表 C.6：接受-拒绝算法 3：狄利克雷过程中 $\boldsymbol{r}$ 针对 $\boldsymbol{s}$所有可能结果的可接受配置，其中 $n = 3$。

参考文献

[1] C. Vicentini，关于狄利克雷过程混合模型的蒙特卡罗方法及其精度参数先验分布的选择，博士论文，杜伦大学（2023年）。网址：http://etheses.dur.ac.uk/14898[2] O. Papaspiliopoulos，G. O. Roberts，狄利克雷过程层次模型的回顾式马尔可夫链蒙特卡罗方法，Biometrika 95 (1) (2008年) 169–186，软件包见 http://www.econ.upf.edu/~omiros/ fast_conditional_mcmc_local.for。doi：https://doi.org/10.1093/biomet/asm086.[3] D. I. Hastie，S. Liverani，S. Richardson，从未知浓度参数的狄利克雷过程混合模型中抽样：大型数据实现中的混合问题，Statistics and Computing 25 (5) (2015年) 1023–1037，软件包见 https://www.jstatsoft.org/article/view/ v064i07。doi：https://doi.org/10.1007/s11222-014-9471-3.[4]D. Blackwell，J. B. MacQueen，通过波利亚瓮方案构造 Ferguson 分布，The Annals of Statistics 1 (2) (1973年) 353–355。doi：https://doi.org/10.1214/aos/1176342372.[5] B. de Finetti，A. Machi，A. F. M. Smith，《概率论：批判性的入门论述》，约翰·威立父子出版公司，英国，2017年。doi：https://doi.org/10.1002/9781119286387.[6]J. Sethuraman，狄利克雷先验的构造性定义，Statistica Sinica (1994年) 639–650。

[7] J. Pitman，《组合随机过程》，施普林格，荷兰，2002年。doi:https://doi.org/10.1007/b11601500.

[8] E. G. Phadia，《先验过程及其应用》，施普林格，瑞士，2015年。doi:https://doi.org/10.1007/978-3-642-39280-1.[9] J. Pitman，受大小偏倚置换不变的随机离散分布，《应用概率进展》28 (2) (1996年) 525–539。doi:https://doi.org/10.2307/1428070.[10] R. Arratia, A. D. Barbour, S. Tavar´e,《对数组合结构：一种概率方法》，欧洲数学会，德国，2003年。doi:https://doi.org/10.4171/000.[11] S. Engen，关于作为物种频数模型的几何级数的一则注记，Biometrika 62 (3) (1975年) 697–699。doi:https://doi.org/10.1093/ biomet/62.3.697.[12] J. W. McCloskey，环境中个体按物种分布的模型，博士论文，密歇根州立大学（1965年）。[13] J. Lee, F. A.Quintana, P. M¨uller, L. Trippa，物种抽样模型预测概率函数的定义，《统计科学》28 (2) (2013年)。doi:https://doi.org/10.1214/12-sts407.[14] J. Pitman，可交换及部分可交换随机划分，《概率论及相关领域》102 (2) (1995年) 145–158。doi:https: //doi.org/10.1007/bf01213386.[15] F. M. Hoppe，群体遗传学中中性等位基因的抽样理论和一种瓮模型，《数学生物学杂志》25 (2) (1987年) 123–159。doi:https://doi.org/10.1007/bf00276386.[16]T. S. Ferguson，若干非参数问题的贝叶斯分析，The Annals of Statistics 1 (2) (1973年) 209–230。doi:https://doi.org/10.1214/aos/1176342360.[17] J. Pitman，Blackwell-MacQueen urn方案的一些进展，《讲义—专著系列》30 (1996年) 245–267。doi:https://doi.org/10.1214/lnms/1215453576.[18] L. Beckett, P. Diaconis，离散纵向数据的谱分析，《数学进展》103 (1) (1994年) 107–128。doi:https://doi.org/10.1006/aima.1994.1002.[19] J. S. Liu，通过序贯插补实现非参数层次贝叶斯，《The Annals of Statistics》24 (3) (1996年) 911–930。doi:https://doi.org/10.1214/aos/1032526949.[20] R核心团队，R：一种统计计算语言与环境，R统计计算基金会，维也纳，奥地利（2023年）。URL https://www.R-project.org/

[21]A. Sokal，《统计力学中的蒙特卡罗方法：基础与新算法》，载于：C. DeWitt-Morette、P. Cartier、A. Folacci（编），《泛函积分》，第361卷，Springer US，美国，1997年，第131–192页，丛书名称：北约高级研究所系列。doi:https://doi.org/10.1007/ 978-1-4899-0319-8_6.[22] R. Neal，狄利克雷过程混合模型的马尔可夫链抽样方法，《计算与图形统计学杂志》9（2）（2000年）249–265。doi:https://doi.org/10.2307/1390653.[23] P. J. Green、S. Richardson，基于狄利克雷过程与不基于狄利克雷过程的异质性建模，《斯堪的纳维亚统计学杂志》28（2）（2001年）355–375。doi:https://doi.org/10.1111/1467-9469.00242.[24] M. Kalli、J. E. Griffin、S. G. Walker，混合模型的切片抽样，《统计与计算》21（1）（2011年）93–105。doi:https://doi.org/10. 1007/s11222-009-9150-y.[25] S. N. MacEachern、M. Clyde、J. S. Liu，非参数贝叶斯模型的序贯重要性抽样：新一代方法，《加拿大统计学杂志》27（2）（1999年）251–267。doi:https://doi.org/10.2307/ 3315637.[26] S. G. Walker，利用切片对狄利克雷混合模型进行抽样，《统计通讯：模拟与计算》36（1）（2007年）45–54。doi:https://doi.org/10.2139/ssrn.945330.[27] O. Papaspiliopoulos，关于从狄利克雷混合模型中进行后验抽样的说明（2008年）。URL http://wrap.warwick.ac.uk/35493/1/WRAP_papaspliiopoulos_08-20wv2.pdf[28] J. S. Liu，贝叶斯计算中的塌缩吉布斯采样器及其在基因调控问题中的应用，《美国统计学会会刊》89（427）（1994年）958–966。doi:https://doi.org/10.1080/ 01621459.1994.10476829.[29] M. D. Fall，E. Barat，Pitman-Yor混合模型的吉布斯抽样方法（2014年）。URL https://hal.archives-ouvertes.fr/hal-00740770v2[30] P. De Blasi、M. F. Gil-Leyva，按出现顺序排列的混合模型的吉布斯抽样：有序分配采样器，《计算与图形统计学杂志》32（4）（2023年）1416–1424。doi:10.1080/10618600. 2023.2177298.[31] M. F. Gil–Leyva、R. H. Mena，具有可交换长度变量的断棍过程，《美国统计学会会刊》（2021年）。doi:https://doi.org/10.1080/01621459.2021.1941054。

[32] B. M. Hill、D. Lane、W. Sudderth，可交换罐过程，《概率年刊》15（4）（1987年）1586–1592。[33]H. Crane，无处不在的Ewens抽样公式，《统计科学》31（1）（2016年）1–19。[34] R. J. Connor、J. E. Mosimann，比例独立性的概念及狄利克雷分布的推广，《美国统计学会会刊》64（325）（1969年）194–206。