22 马尔科夫链蒙特卡罗

Fundamentals of Markov Chain Monte Carlo

使用 pymc3 产生满足特定后验分布的随机数

我们必须谦虚地承认，数字纯粹是人类思想的产物，但宇宙存却是颠扑不破的真理，它超然于人类思想。因此我们不能管宇宙的属性叫先验。

We must admit with humility that, while number is purely a product of our minds, space has a reality outside our minds, so that we cannot completely prescribe its properties a priori.

—— 卡尔·弗里德里希·高斯 (Carl Friedrich Gauss) | 德国数学家、物理学家、天文学家 | 1777 ~ 1855

- `numpy.arange()` 根据指定的范围以及设定的步长，生成一个等差数组
- `numpy.concatenate()` 将多个数组进行连接
- `numpy.linalg.eig()` 特征值分解
- `numpy.random.uniform()` 产生满足连续均匀分布的随机数
- `numpy.zeros_like()` 用来生成和输入矩阵形状相同的零矩阵
- `pymc3.Dirichlet()` 定义 Dirichlet 先验分布
- `pymc3.Multinomial()` 定义多项分布似然函数
- `pymc3.plot_posterior()` 绘制后验分布
- `pymc3.sample()` 产生随机数
- `pymc3.traceplot()` 绘制后验分布随机数轨迹图
- `scipy.stats.beta()` Beta 分布
- `scipy.stats.beta.pdf()` Beta 分布概率密度函数
- `scipy.stats.binom()` 二项分布
- `scipy.stats.binom.pmf()` 二项分布概率质量函数
- `scipy.stats.binom.rsv()` 二项分布随机数
- `scipy.stats.dirichlet()` Dirichlet 分布
- `scipy.stats.dirichlet.pdf()` Dirichlet 分布概率密度函数
- `scipy.stats.norm.pdf()` 正态分布概率分布 PDF
- `scipy.stats.norm.ppf()` 高斯分布百分点函数 PPF
- `scipy.stats.norm.rvs()` 生成正态分布分布随机数

22.1 归一化因子没有闭式解？

贝叶斯推断

回忆前两章贝叶斯推断中用到的贝叶斯定理：

$$\overbrace{f_{\Theta|X}(\theta|x)}^{\text{Posterior}} = \frac{\overbrace{f_{X|\Theta}(x|\theta)}^{\text{Likelihood}}\overbrace{f_{\Theta}(\theta)}^{\text{Prior}}}{\underbrace{f_X(x)}_{\text{Evidence}}} = \frac{\overbrace{f_{X|\Theta}(x|\theta)}^{\text{Likelihood}}\overbrace{f_{\Theta}(\theta)}^{\text{Prior}}}{\int_{\vartheta}\underbrace{f_{X|\Theta}(x|\vartheta)}_{\text{Likelihood}}\underbrace{f_{\Theta}(\vartheta)}_{\text{Prior}}\mathrm{d}\vartheta} \tag{1}$$

其中：

$f_{\Theta|X}(\theta|x)$ 为后验概率 (posterior)；

$f_{X|\Theta}(x|\theta)$ 为似然概率 (likelihood)；

$f_{\Theta}(\theta)$ 为先验概率 (prior)；

$f_X(x)$ 为证据因子 (evidence)，起到归一化作用。

如图 1 所示，贝叶斯推断中最重要的比例关系就是，后验 ∝ 先验 × 似然：

$$\overbrace{f_{\Theta|X}(\theta|x)}^{\text{Posterior}} \propto \overbrace{f_{\Theta}(\theta)}^{\text{Prior}}\overbrace{f_{X|\Theta}(x|\theta)}^{\text{Likelihood}} \tag{2}$$

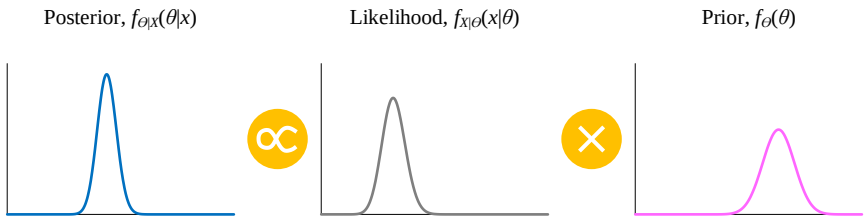


图 1. 后验 ∝ 先验 × 似然

共轭分布

前两章中，如图 2 所示，我们足够“幸运”，成功地避开了 $\int_{\vartheta} f_{X|\Theta}(x|\vartheta) f_{\Theta}(\vartheta)\mathrm{d}\vartheta$ 这个积分。这是因为我们选择的先验分布是似然函数的共轭先验 (conjugate prior)，这样我们便可以得到后验概率 $f_{\Theta|X}(\theta|x)$ 的闭式解。

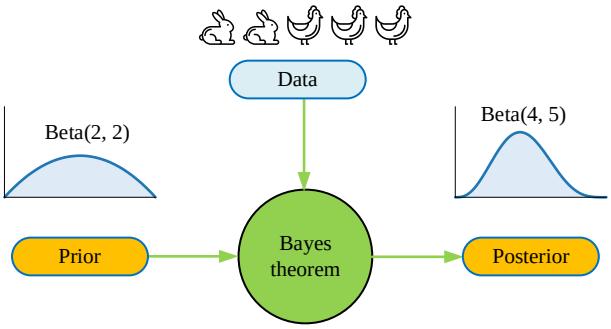


图 2. 先验 Beta(2, 2) + 样本 (2, 3) → 后验 Beta(4, 5)

维数灾难

《数学要素》第 18 章介绍过数值积分。如图 3 所示，利用相同的思路，我们可以通过合理划分区间，获得后验分布的大致形状，以及对应的面积或体积，并且完成归一化。但是，这种思路仅仅适用于模型参数较小的情况。因为当模型参数很多时便会导致维数灾难 (curse of dimensionality)。

所谓的维数灾难是指在涉及到向量的计算的问题中，随着维数的增加，计算量呈指数倍增长的一种现象。举个例子，如果模型有 3 个参数，每个参数在各自区间上均匀选取 20 个点，这个参数空间中共有 8000 个点 (= 20 × 20 × 20 = 20^3)。试想，模型如果有 20 个参数，每个维度上同样选取 20 个点，这样参数空间的点数达到惊人的 1.048×10^{26} (= 20^{20})。

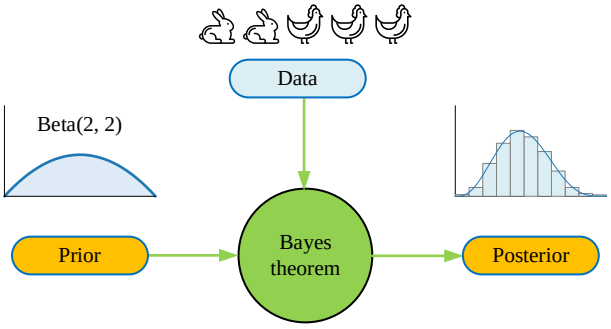


图 3. 先验 Beta(2, 2) + 样本 (2, 3) → 后验分布，数值积分

马尔科夫链蒙特卡洛模拟 MCMC

但是，如果我们想绕过复杂的推导过程，或者想避免数值积分带来的维数灾难，有没有其他办法获得后验分布？如图 4 所示，我们可以用马尔科夫链蒙特卡罗模拟 (Markov Chain Monte Carlo, MCMC)。马尔科夫链蒙特卡罗模拟允许我们估计后验分布的形状，以防我们无法直接获得后验分布的闭式解。此外，蒙特卡洛方法成功地绕开了维数灾难。

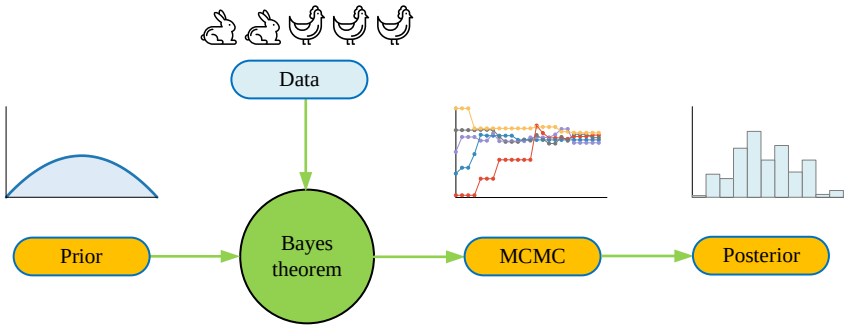


图 4. 先验 Beta(2, 2) + 样本 (2, 3) → 后验分布，马尔科夫链蒙特卡罗模拟

相信大家已经发现马尔科夫链蒙特卡罗模拟有两部分——马尔科夫链、蒙特卡罗模拟。本书第 15 章专门介绍过蒙特卡洛模拟，大家对此应该很熟悉。本系列丛书的读者对“马尔科夫”这个词应该不陌生，我们在《数学要素》第 25 章“鸡兔互变”的例子中介绍过“马尔科夫”。

马尔可夫链 (Markov chain) 因俄国数学家安德烈·马尔可夫 (Andrey Andreyevich Markov) 得名，为状态空间中经过从一个状态到另一个状态的转换的随机过程。限于篇幅，本章不展开讲解马尔科夫链。

Metropolis-Hastings 采样

梅特罗波利斯-黑斯廷斯算法 (Metropolis-Hastings algorithm, MH) 是马尔可夫链蒙特卡洛中一种基本的抽样方法。

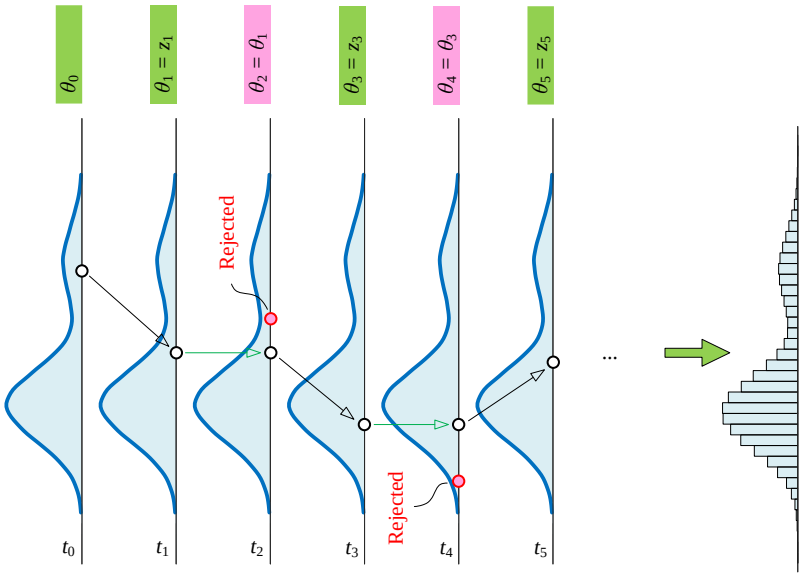


图 5. Metropolis-Hastings 采样算法原理

它通过在取值空间取任意值作为起始点，按照先验分布计算概率密度，计算起始点的概率密度。然后随机移动到下一点时，计算当前点的概率密度。移动的步伐一般从正态分布中抽取。

接着，计算当前点和起始点概率密度的比值 ρ，并产生 (0,1) 之间服从连续均匀的随机数 u。最后，对比 ρ 与产生的随机数 u 的大小来判断是否保留当前点。当前者大于后者，接受当前点，反之则拒绝当前点。这个过程一直循环，直到获得能被接受后验分布。这一步和本书第 15 章介绍的“接受-拒绝抽样”本质上一致。

有关 MH 算法原理和具体流程，请大家参考李航老师的新作《机器学习方法》。

鸡兔比例

下面，我们利用 MH 算法模拟产生“鸡兔比例”中的后验分布。先验分布采用 Beta(α, α)。样本数据为 200 (n)，其中 60 (s) 只兔子。图 6 比较 α 取不同值时先验分布、后验分布的解析解、随机数分布。图中先验分布的随机数服从 Beta 分布，后验分布的随机数则由 MH 算法产生。

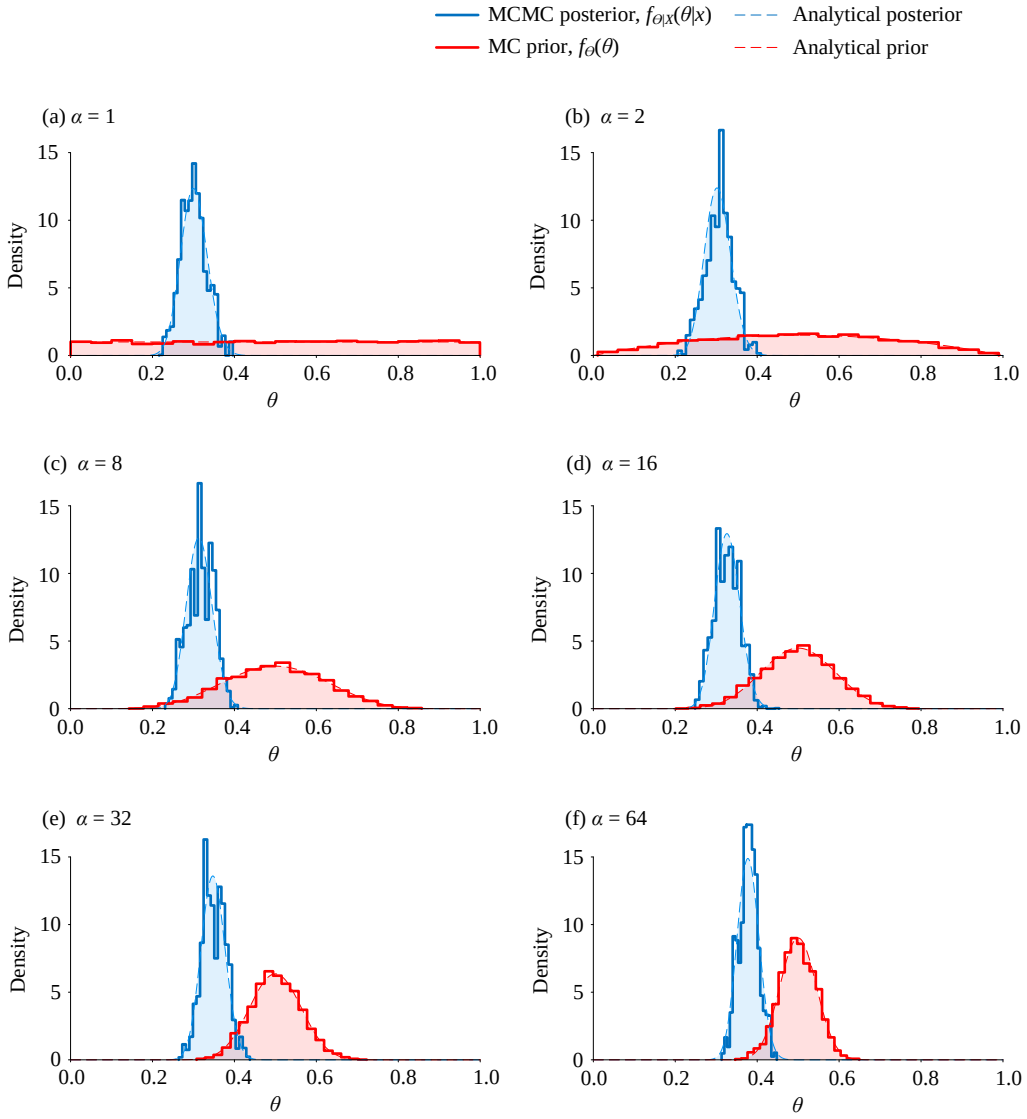


图 6. 对比先验分布、后验分布，α 取不同值时

图 7 所示为马尔科夫链蒙特卡洛模拟的收敛性。图中五条不同的后验分布随机数轨迹路径的初始值完全不同，但是它们对重都收敛于一个稳态分布，这个稳态分布对应我们要求解的后验分布。大家查看本节和本章后文代码时会发现，收敛于稳态分布之前的随机数一般都会被截断去除。

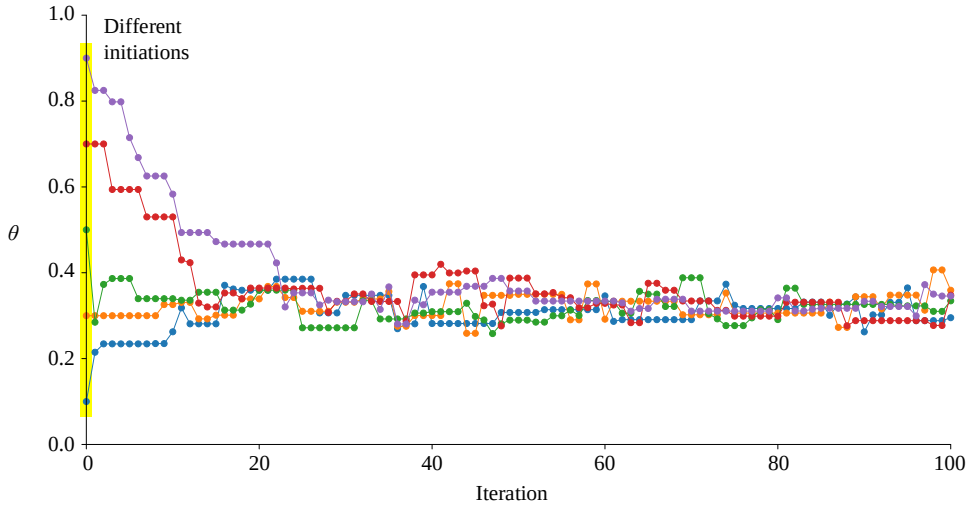


图 7. 马尔可夫链蒙特卡洛的收敛

代码 Bk5_Ch022_01.py 绘制图 6、图 7。

22.2 鸡兔比例：使用 pymc3

本节和下一节利用 pymc3 完成贝叶斯推断中的马尔科夫链蒙特卡罗模拟。

PyMC3 是一种 Python 开源的概率编程库，用于进行概率建模、贝叶斯统计推断和蒙特卡罗马尔可夫链蒙特卡罗（MCMC）采样。PyMC3 允许用户使用 Python 语言定义概率模型，并指定其参数的先验分布；PyMC3 支持多种先验分布，包括连续和离散分布；

PyMC3 支持使用多种 MCMC 算法进行采样，包括 NUTS、Metropolis-Hastings 和 Slice 等。PyMC3 具有丰富的可视化和后处理工具，包括 traceplot、summary、forestplot 等，方便用户对模型进行分析和诊断。

PyMC3 可以用于许多应用领域，包括机器学习、计量经济学、社会科学、物理学、生物学、神经科学等。由于 PyMC3 的简洁易用和高效性，它已经成为了许多学术界和工业界研究者进行概率建模和贝叶斯推断的首选工具之一。

先验 Beta(2,2) + 样本 2 兔 3 鸡

如图 8 所示，根据本书第 20 章内容，对于鸡兔比例问题，我们知道当先验分布为 Beta(2, 2)，引入样本数据 (2 兔、3 鸡)，得到的后验分布为 Beta(4, 5)。先验分布 Beta(2, 2) 的均值、众数都位于 1/2，也就是鸡兔各占 50%，但是确信度不高。请大家自己计算 Beta(4, 5) 均值位置。

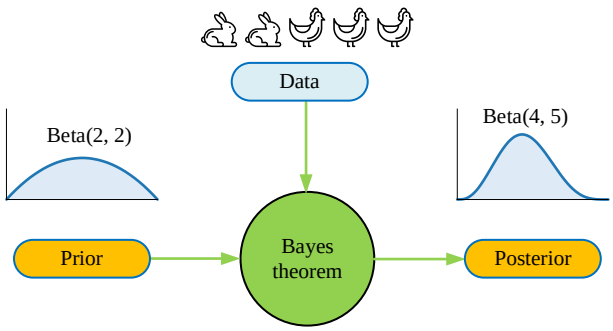


图 8. 先验 Beta(2, 2) + 样本 (2, 3) → 后验 Beta(4, 5)

下面，我们利用 PyMC3 模拟产生这个后验分布。注意，由于 Beta 分布是 Dirichlet 分布的特例。本节的先验分布实际上是二元 Dirichlet 分布，所以我们会看到两个后验分布。图 9 (b) 所示为后验分布随机数轨迹图，这些随机数便构成后验分布。

轨迹图中蓝色曲线对应图 9 (a) 中蓝色后验分布，即兔子比例。轨迹图中橙色曲线对应图 9 (a) 中橙色后验分布，即鸡的比例。在代码中，大家会看到随机数轨迹实际上是由两条轨迹合并而成。

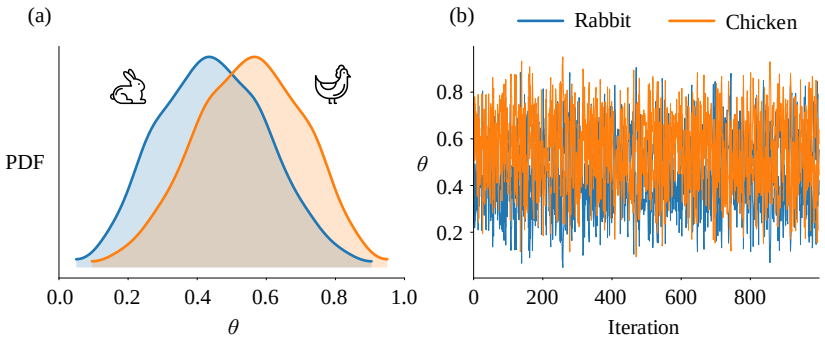


图 9. 后验分布随机数轨迹图，先验 Beta(2,2) + 样本 2 兔 3 鸡

图 10 分别用直方图、KDE 曲线可视化两个后验分布。图 10 给出的均值所在位置就相当于最大后验 MAP 的优化解。

图中 HDI 代表最大密度区间 (highest density interval)。HDI 又叫 HPDI (highest posterior density interval)，本质上是上一章介绍的后验分布可信区间。HDI 的特点是，相同置信度下，HDI 区间宽度最短，HDI 区间两端对应概率密度值相等。但是，HDI 左右尾对应的面积很可能不相等，这一点明显不同于可信区间。

图 10 (a) 告诉我们兔比例的后验分布 94%最大密度区间的宽度为 0.57 (= 0.75 – 0.18)。鸡比例的后验分布 94%最大密度区间的宽度也是 0.57 (= 0.82 – 0.25)。这个宽度可以用来度量确信程度。

再次强调，贝叶斯派认为模型参数本身不确定，也服从某种分布。因此可信区间或 HDI 本身就是模型参数的分布。这一点完全不同于频率派的置信区间。

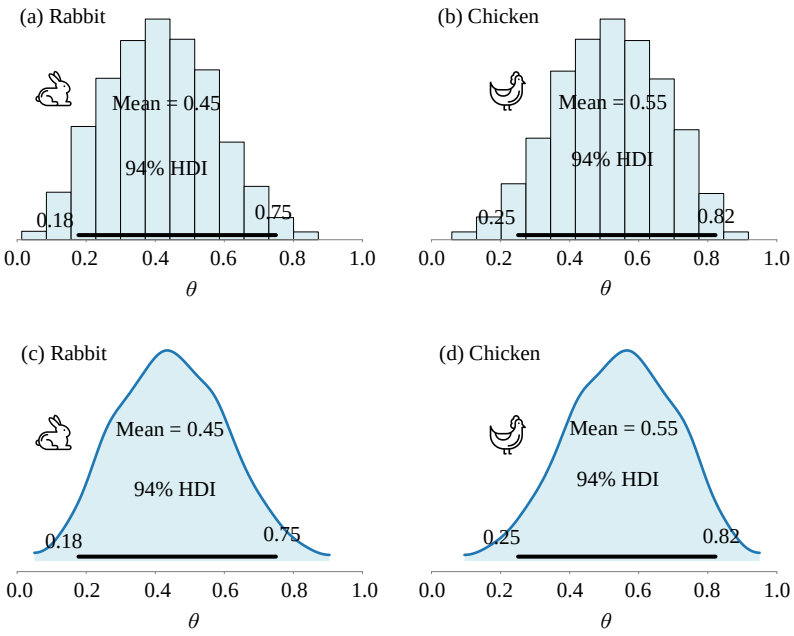


图 10. 后验分布直方图、KDE，先验 Beta(2,2) + 样本 2 兔 3 鸡

先验 Beta(2,2) + 样本 90 兔 110 鸡

再看一个例子。如图 11 所示，先验分布还是 Beta(2, 2)，但是样本数据为 90 只兔、110 只鸡。请大家试着自己推到得到后验分布的解析式。

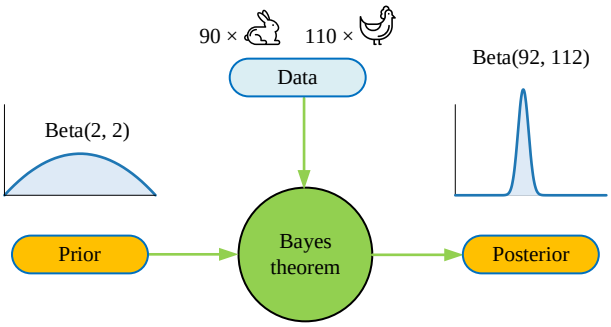


图 11. 先验 Beta(2, 2) + 样本 (90, 110) → 后验 Beta(92, 112)

图 12 (a) 所示为鸡兔比例的后验分布。图 12 (b) 所示为产生后验分布的随机数。

图 13 所示为后验分布的直方图和 KDE 曲线。虽然先验分布相同，由于引入更多样本，相比图 10，图 13 的后验分布变得更加“细高”，也就是说确信度变得更强。

图 13 (a) 告诉我们兔比例的后验分布 94% HDI 的宽度为 0.13 (= 0.51 – 0.38)。鸡比例的后验分布 94% HDI 的宽度也是 0.13 (= 0.62 – 0.49)。相比图 10，最大密度区间宽度明显缩小。

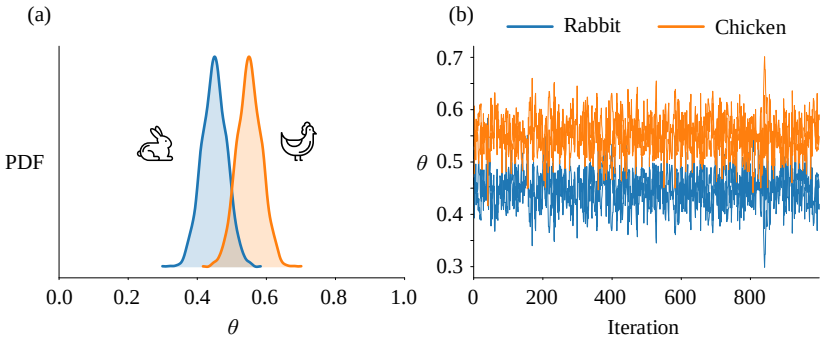


图 12. 后验分布随机数轨迹图，先验 Beta(2,2) + 样本 90 兔 110 鸡

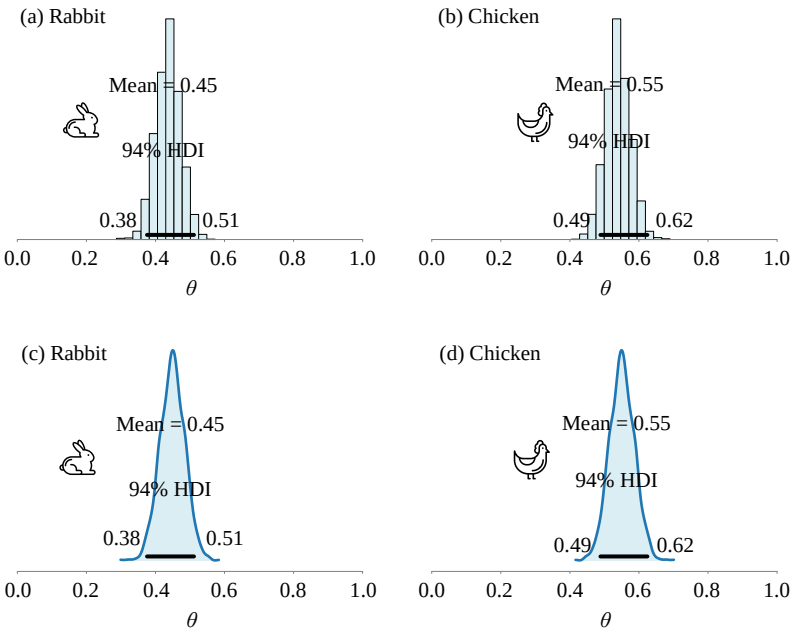


图 13. 后验分布直方图、KDE，先验 Beta(2,2) + 样本 90 兔 110 鸡

代码 Bk5_Ch22_02.ipynb 绘制图 9、图 10、图 11、图 12。请大家用 JupyterLab 打开并运行代码文件。此外，请大家改变先验分布的参数设置，并观察后验分布的变化。

22.3 鸡兔猪比例：使用 pymc3

本节用 PyMC3 求解鸡兔猪比例的贝叶斯推断问题。

先验 Dir(2,2,2) + 样本 3 兔 6 鸡 1 猪

选取 Dir(2, 2, 2) 作为先验分布，这意味着事先主观经验认为鸡兔猪的占比都是 1/3，但是确信度不够强。如图 14 所示，观察到的 10 只动物中有 6 只鸡、3 只兔、1 只猪。利用上一章内容，我们可以推导得到后验分布为 Dir(8, 5, 3)。下面，这一节也用 pymc3 完成 MCMC 模拟并生成后验边缘分布。

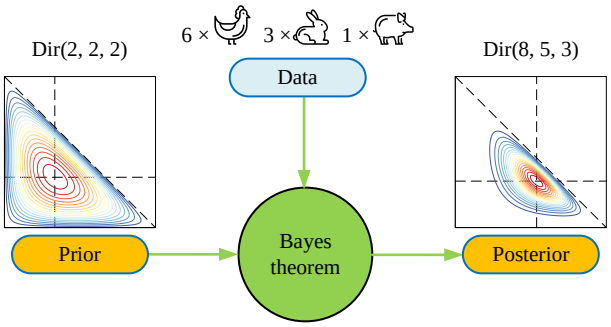


图 14. 先验 Dir(2, 2, 2) + 样本 → 后验 Dir(8, 5, 3)

图 15 (b) 所示为后验分布随机数轨迹图，由此得到图 15 (a) 左图的后验分布。

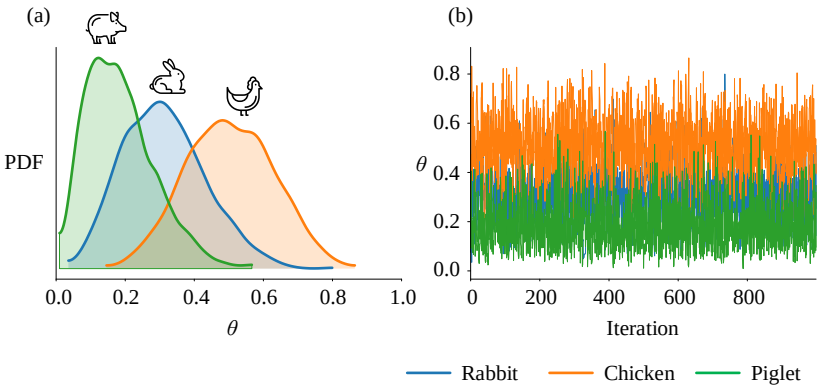


图 15. 后验分布随机数轨迹图，先验 Dir(2,2,2) + 样本 3 兔 6 鸡 1 猪

图 16 所示为三种动物比例的后验分布直方图、KDE 曲线。

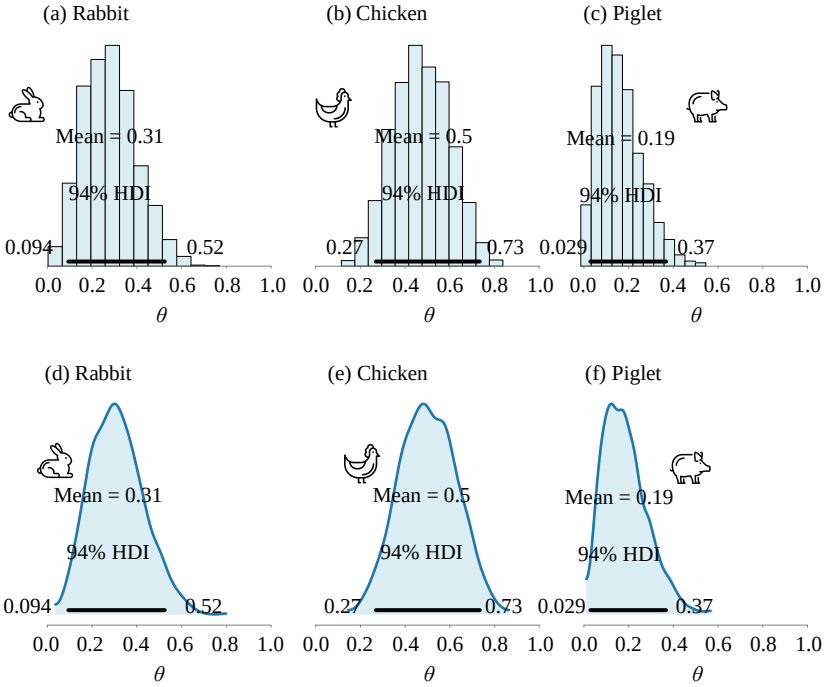


图 16. 后验分布直方图、KDE，先验 Dir(2,2,2) + 样本 3 兔 6 鸡 1 猪

先验 Dir(2,2,2) + 样本 65 兔 115 鸡 20 猪

下面保持先验分布 Dir(2, 2, 2) 不变，增加样本数量 (115 鸡、65 兔、20 猪)，得到的后验分布为 Dir(117, 67, 22)。建议大家自己试着推导后验分布闭式解。

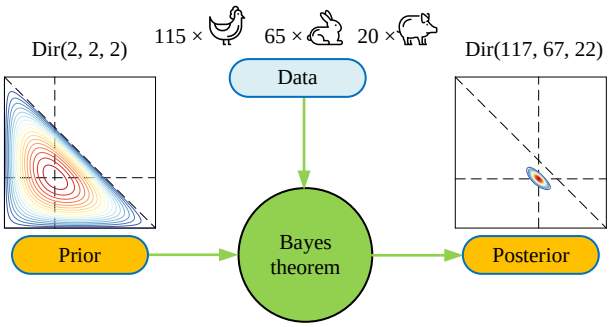


图 17. 先验 Dir(2, 2, 2) + 样本 → 后验 Dir(117, 67, 22)

图 18 所示为三种动物的后验概率随机数的轨迹图和分布。图 19 所示为后验分布的直方图、KDE 曲线。请大家自己计算并对比图 16 和图 19 中 94% HDI 宽度。

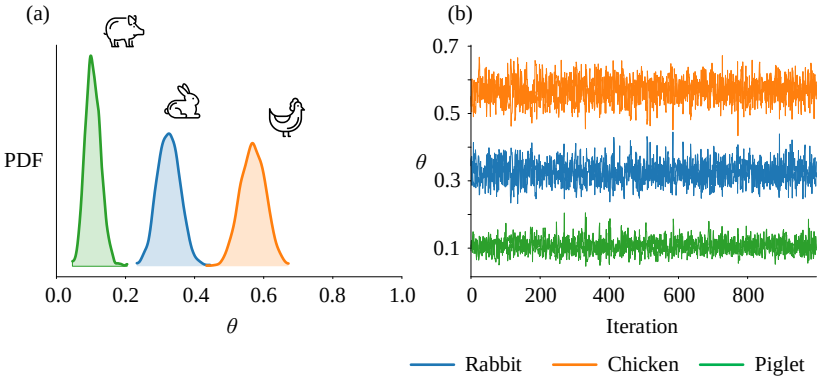


图 18. 后验分布随机数轨迹图，先验 Dir(2,2,2) + 样本 65 兔 115 鸡 20 猪

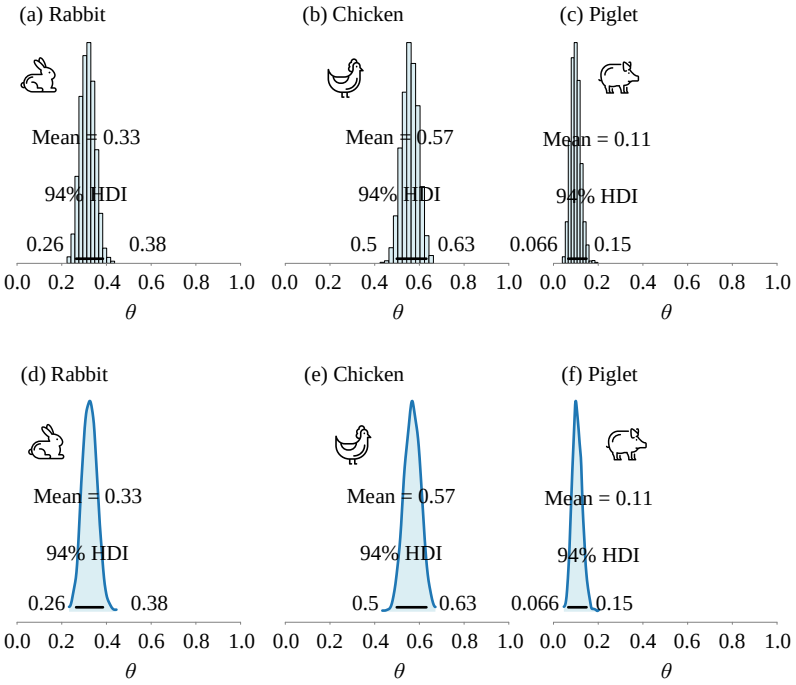
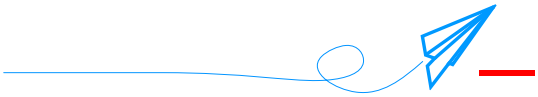


图 19. 后验分布直方图、KDE，先验 Dir(2,2,2) + 样本 65 兔 115 鸡 20 猪

代码 Bk5_Ch22_03.ipynb 绘制图 15、图 16、图 18、图 19。请大家用 JupyterLab 打开并运行代码文件。请大家改变先验分布参数，从而调整置信度，并观察后验分布的变化。



总结来说，贝叶斯推断把总体的模型参数看作随机变量。在得到样本之前，根据主观经验和既有知识给出未知参数的概率分布，称为先验分布。从总体中得到样本数据后，根据贝叶斯定理，基于给定的样本数据，得出模型参数的后验分布。并根据参数的后验分布进行统计推断。贝叶斯推断对应的优化问题为最大化后验概率，即 MAP。

在贝叶斯推断中，我们关注的核心是模型参数的后验分布。而样本数据服从怎样的分布不是贝叶斯推断关注的重点。

贝叶斯推断也并不完美！明显的缺点之一就是分析推导过程十分复杂。先验分布的建立，需要丰富的经验。采用马尔科夫链蒙特卡罗模拟，可以避免复杂推导，避免数值积分可能带来的维度灾难，但是计算成本显然较高。

想深入学习贝叶斯推断的读者可以参考开源图书 *Bayesian Methods for Hackers: Probabilistic Programming and Bayesian Inference*：

https://github.com/CamDavidsonPilon/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers