

翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

## 23 Monte Carlo inference

23.1 Introduction So far, we discussed various deterministic algorithms for posterior inference. These methods enjoy many of the benefits of the Bayesian approach, while still being about as fast as optimization-based point-estimation methods. The trouble with these methods is that they can be rather complicated to derive, and they are somewhat limited in their domain of applicability (e.g., they usually assume conjugate priors and exponential family likelihoods, although see (Wand et al. 2011) for some recent extensions of mean field to more complex distributions). Furthermore, although they are fast, their accuracy is often limited by the form of the approximation which we choose.

In this chapter, we discuss an alternative class of algorithms based on the idea of Monte Carlo approximation, which we first introduced in Section 2.7. The idea is very simple: generate some (unweighted) samples from the posterior,  $\mathbf{x}_S \sim p(\mathbf{x}|D)$ , and then use these to compute any quantity of interest, such as a posterior marginal,  $p(x_1|D)$ , or the posterior of the difference of two quantities,  $p(x_1 - x_2|D)$ , or the posterior predictive,  $p(y|D)$ , etc. All of these quantities can be approximated by  $E[f|D] \approx \frac{1}{S} \sum_{s=1}^S f(\mathbf{x}_s)$  for some suitable function  $f$ .

By generating enough samples, we can achieve any desired level of accuracy we like. The main issue is: how do we efficiently generate samples from a probability distribution, particularly in high dimensions? In this chapter, we discuss non-iterative methods for generating independent samples. In the next chapter, we discuss an iterative method known as Markov Chain Monte Carlo, or MCMC for short, which produces dependent samples but which works well in high dimensions. Note that sampling is a large topic. The reader should consult other books, such as (Liu 2001; Robert and Casella 2004), for more information.

已思考 42s

## 原文翻译

## 23 Monte Carlo 推断

**23.1 引言** 到目前为止，我们讨论了用于后验推断的各种**确定性**算法。这些方法享有贝叶斯方法的许多优点，同时速度仍然与基于优化的点估计方法差不多。麻烦在于，这些方法的推导可能相当复杂，而且它们的适用范围也有一定限制（例如，它们通常假设共轭先验和指数族似然，不过参见 (Wand et al., 2011)，其中对将均值场扩展到更复杂分布作了近期的一些拓展）。此外，尽管这些方法很快，但其精度往往受制于我们所选择的近似形式。

在本章中，我们讨论另一类基于**蒙特卡罗近似**思想的算法，我们已在第 2.7 节首次介绍过这一思想。思路非常简单：从后验分布中生成一些（**无权重的**）样本， $x^{(s)} \sim p(x | D)$ ，然后用这些样本来计算任何感兴趣的量，例如后验边缘分布  $p(x_1 | D)$ ，或两个量之差的后验分布  $p(x_1 - x_2 | D)$ ，或后验预测分布  $p(y | D)$  等。所有这些量都可以用

$$\mathbb{E}[f | D] \approx \frac{1}{S} \sum_{s=1}^S f(x^{(s)})$$

来近似，其中  $f$  是某个合适的函数。

只要生成足够多的样本，我们就能达到任意想要的精度。主要问题是：**如何高效地**从一个概率分布中生成样本，尤其是在高维情形下？在本章中，我们讨论用于生成**相互独立样本**的**非迭代**方法。在下一章中，我们将讨论一种称为**马尔可夫链蒙特卡罗**（简称 MCMC）的**迭代**方法，它生成**相关**样本，但在高维情况下工作良好。需要注意，采样是一个很大的主题。读者可参考其他书籍，例如 (Liu, 2001; Robert and Casella, 2004)，以获取更多信息。

## 通俗解释（一步步来）

### 1) “确定性”方法是什么？为何有局限？

- 确定性后验近似（如变分推断/均值场）会把“难算的真实后验”替换为“容易算的近似家族”，通过优化找到最接近的那个。
- 优点：快，常常接近“做一次优化”的速度；还能给出贝叶斯式的概率答案。
- 局限：
  - **推导难**：每换个模型，常要重新推公式。
  - **适用窄**：常依赖“共轭先验 + 指数族似然”等结构；超出这个范围就麻烦（虽然后来有扩展）。
  - **近似偏差**：你选的近似家族限制了结果上限——再快也可能**不够准**。

### 2) 蒙特卡罗 (MC) 思想：用样本代表分布

- 如果你能从后验  $p(x | D)$  随机抽样，很多量都能用**样本平均**来估：
  - 例如要某个函数的后验期望  $\mathbb{E}[f | D]$ ，就对样本算  $f(x^{(s)})$  并取平均： $\frac{1}{S} \sum f(x^{(s)})$
  -

- “无权重的样本”= 每个样本一票，同等重要。
- 直观类比：想知道一个巨大人群的平均身高，不必量所有人，随机抽  $S$  个人，算平均， $S$  越大越接近真实平均。

### 3) 能算哪些量？

- **边缘后验**  $p(x_1 | D)$ : 只关心高维参数里的一个分量时，可对样本的第1维做统计（如直方图/密度估计）。
- **差的后验**  $p(x_1 - x_2 | D)$ : 对每个样本算  $x_1^{(s)} - x_2^{(s)}$ ，再统计。
- **后验预测**  $p(y | D)$ : 对每个参数样本  $x^{(s)}$ ，从  $p(y | x^{(s)})$  产生或评估，再平均。
- 统一观点：只要能把“想要的量”写成某个函数  $f(x)$  的期望，就能用样本平均来近似。

### 4) 精度与样本量 ( $S$ )

- **法则**：样本量  $S$  越大，样本平均的随机误差越小，结果越稳。
- **好处**：不需要复杂推导；只要能抽样，就能“任意逼近”真实答案。
- **挑战：从哪里来抽样？** 真实后验常常没有现成的抽样器。

### 5) 难点：如何高效抽样，尤其是高维？

- 在高维空间，分布可能非常复杂（多峰、强相关），**直接抽样**不容易。
- 本章：讲**非迭代方法**，目标是得到**独立样本**（样本之间互不相关）。
- 下一章：讲 MCMC（迭代法），它让样本**相关**，但在高维常更可行、实用。

### 6) 非迭代 vs 迭代的对比

- **非迭代独立抽样**（本章）：理想但往往难构造（除非分布简单或能做变换/重要性抽样等）。
- **MCMC 迭代抽样**（下章）：更通用，适配复杂高维，但样本相关，需要调参与诊断（如链混合、相关性、有效样本量）。

### 7) 一个简单小例子（直觉版）

- 你想知道后验下“参数  $x$  是否大于 0 的概率”。
- 若能从  $p(x | D)$  抽  $S$  个样本，只需数一数有多少个样本  $x^{(s)} > 0$ 。
- 比例  $\approx \Pr(x > 0 | D)$ 。 $S$  越大，估计越稳。

## 关键术语慢解释

- **后验分布**  $p(x | D)$ : 给定数据  $D$  后，参数  $x$  的不确定性分布，是贝叶斯推断的核心。
- **边缘分布**  $p(x_1 | D)$ : 只看参数向量里某一维（或子集）的分布，其它维度“边缘化掉”（积分/求和掉）。
- **后验预测**  $p(y | D)$ : 用后验不确定性去预测新观测  $y$  的分布。

- **无权重样本**: 每个样本同等权重; 相对地, **有权重样本** (如重要性采样) 会给不同样本不同权重。
  - **高维**: 参数个数多; 维度越高, 直接抽样和数值积分越难。
  - **MCMC**: 通过构造马尔可夫链让其平稳分布等于目标后验, 从链上取样; 样本相关, 但普适。
- 

## 本段落想传达的核心观念

1. **确定性近似快但受限**: 推导难、适用性有限、近似偏差可能较大。
  2. **蒙特卡罗 = 用随机样本做期望**: 从后验抽样, 用样本平均近似任何感兴趣的后验量或预测量。
  3. **精度靠样本数**: 样本越多, 越接近真实值。
  4. **难点在抽样**: 如何高效地从复杂 (尤其高维) 的后验分布抽样。
  5. **两大路线**:
    - 本章: 非迭代、独立样本的方法;
    - 下一章: MCMC, 迭代、相关样本, 但高维更实用。
  6. **采样是大主题**: 想深挖可看 Liu (2001)、Robert & Casella (2004)。
- 

## 要点总结

- 目标: 计算后验中的各种量 (边缘、差、预测等)。
- 方法对比:
  - **确定性近似**: 快但可能不准/不通用;
  - **蒙特卡罗**: 若能抽样, 就能用样本平均近似任意期望, 精度随样本数提升。
- 难点: **获得后验样本**, 尤其在高维。
- 结构安排:
  - 本章: 讲能给出**独立样本的非迭代抽样**;
  - 下章: 讲 MCMC (**迭代, 相关样本**), 适合高维复杂后验。
- 参考: Liu (2001); Robert & Casella (2004); 关于变分扩展参见 Wand et al. (2011)。

这一节主要在讲什么? 我不太懂

已思考 28s

没关系，这一节的核心其实就一句话：

**用“随机抽样”来代替复杂推导：从后验分布里抽很多样本，用样本的平均值近似你想要的任何量。难点只是——怎么高效地抽到这些样本，尤其在高维。**

下面把它拆开说清楚：

- **为什么要换思路？**

之前的“确定性近似”（如变分推断）虽然快，但推起来麻烦、对模型有要求（常要共轭/指数族），而且会有近似偏差，可能不够准。

- **蒙特卡罗 (MC) 在干嘛？**

想像你做一个民意调查：不问所有人，只抽样一部分人，算平均。

在这里也是：如果能从后验  $p(x | D)$  抽到样本  $x^{(1)}, \dots, x^{(S)}$ ，那任何“感兴趣的量”只要能写成一个函数  $f(x)$  的期望，就用

$$\frac{1}{S} \sum_{s=1}^S f(x^{(s)})$$

来近似。样本是“无权重”的，意思是每个样本同等重要。

- **能算哪些东西？**

几乎你关心的都能算：

- 某一维的后验（做直方图/密度估计）
- 两个参数差的后验（对每个样本做差，再统计）
- 后验预测（对每个参数样本去预测，再平均）

- **精度怎么保证？**

样本越多，越接近真实答案（直觉上误差大约随样本数增加而下降）。不需要复杂推导，靠抽样就能逼近。

- **真正的难点：怎么抽样？**

从复杂、高维的后验里直接抽样并不容易。

本章谈的是**非迭代的方法**（目标是得到**彼此独立**的样本）。

下一章会讲 MCMC：一种**迭代方法**，样本**彼此相关**，但它在高维更好用。

- **超简例子（直觉版）**

你想要  $\Pr(x_1 > x_2 | D)$ 。

从后验抽  $S$  个样本对  $(x_1^{(s)}, x_2^{(s)})$ ，数一数有多少次  $x_1^{(s)} > x_2^{(s)}$ 。

这个比例就是你要的概率的近似。

## 一句话总结

这节主要告诉你：**别再费劲推复杂公式了，只要能从后验抽样，就用“样本平均”近似任何后验量；问题的关键变成了“如何高效抽样”，本章讲独立抽样法，下一章讲更通用的MCMC。**

## 要点清单

- 确定性近似：快，但推导难、适用性有限、会有近似偏差。
- 蒙特卡罗：从后验抽样，用样本平均近似期望。
- 可处理的对象很广：边缘、差、预测等都行。
- 精度靠样本数；样本越多越稳。
- 难点：高维下如何高效抽样。
- 结构：本章 = 非迭代、独立样本；下章 = MCMC、相关样本、但更适合高维。

翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

### 23.2 Sampling from standard distributions

We briefly discuss some ways to sample from 1 or 2 dimensional distributions of standard form. These methods are often used as subroutines by more complex methods.

#### 23.2.1 Using the cdf

The simplest method for sampling from a univariate distribution is based on the inverse probability transform. Let  $F$  be a cdf of some distribution we want to sample from, and let  $F^{-1}$  be its inverse. Then we have the following result.

Theorem 23.2.1. If  $U \sim U(0, 1)$  is a uniform rv, then  $F^{-1}(U) \sim F$ .

Proof.

$$\Pr(F^{-1}(U) \leq x) = \Pr(U \leq F(x)) \text{ (applying } F \text{ to both sides)}$$

(23.1)

$$= F(x) \text{ (because } \Pr(U \leq y) = y \text{ (23.2)})$$

where the first line follows since  $F$  is a monotonic function, and the second line follows since  $U$  is uniform on the unit interval.

Hence we can sample from any univariate distribution, for which we can evaluate its inverse cdf, as follows: generate a random number  $u \sim U(0, 1)$  using a pseudo random number generator (see e.g., (Press et al. 1988) for details).

Let  $u$  represent the height up the  $y$  axis. Then “slide along” the  $x$  axis until you intersect the  $F$  curve, and then “drop down” and return the corresponding  $x$  value. This corresponds to computing  $x = F^{-1}(u)$ . See Figure 23.1 for an illustration.

For example, consider the exponential distribution

$$\text{Expon}(x|\lambda) = \lambda e^{-\lambda x} I(x \geq 0) \quad (23.3)$$

The cdf is

$$F(x) = 1 - e^{-\lambda x} I(x \geq 0) \quad (23.4)$$

whose inverse is the quantile function

$$F^{-1}(p) = -\ln(1 - p) / \lambda \quad (23.5)$$

By the above theorem, if  $U \sim \text{Unif}(0, 1)$ , we know that

$F^{-1}(U) \sim \text{Expon}(\lambda)$ . Furthermore, since  $1 - U \sim \text{Unif}(0, 1)$  as well, we can sample from the exponential distribution by first sampling from the uniform and then transforming the results using  $-\ln(u)/\lambda$ .

### 23.2.2 Sampling from a Gaussian (Box-Muller method)

We now describe a method to sample from a Gaussian. The idea is we sample uniformly from a unit radius circle, and then use the change of variables formula to derive samples from a spherical 2d Gaussian. This can be thought of as two samples from a 1d Gaussian.

In more detail, sample  $z_1, z_2 \in (-1, 1)$  uniformly, and then discard pairs that do not satisfy  $z_1^2 + z_2^2 \leq 1$ . The result will be points uniformly distributed inside the unit circle, so  $p(z) = 1/\pi I(z \text{ inside circle})$ . Now define

$$x_i = z_i - 2 \ln r_i / r_i \quad (23.6)$$

for  $i = 1, 2$ , where  $r_i = \sqrt{z_1^2 + z_2^2}$ . Using the multivariate change of variables formula, we have

$$p(x_1, x_2) = p(z_1, z_2) | \partial(z_1, z_2) / \partial(x_1, x_2) | = 1 / (\sqrt{2\pi})^2 \exp(-x_1^2/2 - x_2^2/2) \quad (23.7)$$

Hence  $x_1$  and  $x_2$  are two independent samples from a univariate Gaussian. This is known as the Box-Muller method.

To sample from a multivariate Gaussian, we first compute the Cholesky decomposition of its covariance matrix,  $\Sigma = LL^T$ , where  $L$  is lower triangular. Next we sample  $x \sim N(0, I)$  using the Box-Muller method. Finally we set  $y = Lx + \mu$ .

This is valid since

$$\text{cov}[y] = L \text{cov}[x] L^T = L L^T = \Sigma \quad (23.8)$$

已思考 27s

下面先给出逐字不漏的中文翻译；随后用通俗、详细的方式解释；最后列要点小结。

(注：原文中的式 (23.6) 在排版上略显紧凑，我在翻译处写成常见的等价形式，便于阅读

与理解。)

---

## 原文翻译

### 23.2 从标准分布采样

我们将简要讨论一些从 1 维或 2 维的标准形式分布进行采样的方法。这些方法经常被更复杂的方法用作子程序。

#### 23.2.1 使用 cdf

从单变量分布采样的最简单方法基于**逆概率变换**。设  $F$  是我们想要采样的某个分布的 cdf (累计分布函数)，设  $F^{-1}$  是其逆函数。那么我们有如下结果。

**定理 23.2.1.** 若  $U \sim U(0, 1)$  是一个均匀随机变量，则  $F^{-1}(U) \sim F$ 。

**证明.**

$$\Pr(F^{-1}(U) \leq x) = \Pr(U \leq F(x)) \quad (\text{对两边应用 } F) \quad (23.1)$$

$$= F(x) \quad (\text{因为 } \Pr(U \leq y) = y) \quad (23.2)$$

其中第一行是因为  $F$  是单调函数，第二行是因为  $U$  在单位区间上服从均匀分布。

因此，对于任何其逆 cdf 可求的单变量分布，我们都可以如下采样：使用伪随机数发生器生成一个随机数  $u \sim U(0, 1)$  (细节参见例如 (Press et al. 1988))。让  $u$  表示沿  $y$  轴向上的高度。然后“沿着  $x$  轴滑动”，直到与你的  $F$  曲线相交，再“垂直下落”，返回相应的  $x$  值。这对应于计算  $x = F^{-1}(u)$ 。见图 23.1 的示意。

例如，考虑指数分布

$$\text{Expon}(x | \lambda) = \lambda e^{-\lambda x} \mathbf{I}(x \geq 0) \quad (23.3)$$

其 cdf 为

$$F(x) = 1 - e^{-\lambda x} \mathbf{I}(x \geq 0) \quad (23.4)$$

其逆函数是分位数函数

$$F^{-1}(p) = \frac{-\ln(1-p)}{\lambda} \quad (23.5)$$

根据上面的定理，如果  $U \sim \text{Unif}(0, 1)$ ，则  $F^{-1}(U) \sim \text{Expon}(\lambda)$ 。此外，因为  $1 - U \sim \text{Unif}(0, 1)$ ，我们可以先从均匀分布采样，再用  $-\ln(u)/\lambda$  来变换样本，从而从指数分布采样。

### 23.2.2 从高斯分布采样 (Box–Muller 方法)

现在我们描述一种从高斯分布采样的方法。其思想是：我们先从单位半径圆内均匀采样，然后利用变量变换公式推导出来自球面对称二维高斯的样本。可以把这理解为两份来自一维高斯的样本。

更具体地，均匀地采样  $z_1, z_2 \in (-1, 1)$ ，然后丢弃那些不满足  $z_1^2 + z_2^2 \leq 1$  的样本对。这样得到的点将在单位圆内均匀分布，所以  $p(z) = \frac{1}{\pi} \mathbf{I}(z \text{ 在圆内})$ 。现在定义

$$x_i = z_i \left( \frac{-2 \ln r^2}{r^2} \right)^{1/2} \quad (i = 1, 2), \text{ 其中 } r^2 = z_1^2 + z_2^2 \quad (23.6)$$

利用多元变量变换公式，我们有

$$p(x_1, x_2) = p(z_1, z_2) \left| \frac{\partial(z_1, z_2)}{\partial(x_1, x_2)} \right| = \frac{1}{\sqrt{2\pi}} \exp(-\frac{x_1^2}{2}) \frac{1}{\sqrt{2\pi}} \exp(-\frac{x_2^2}{2}) \quad (23.7)$$

因此  $x_1$  和  $x_2$  是来自单变量高斯的两个相互独立的样本。这被称为 **Box–Muller 方法**。

要从多元高斯分布采样，我们首先计算其协方差矩阵的 Cholesky 分解， $\Sigma = LL^\top$ ，其中  $L$  为下三角。接着用 Box–Muller 方法采样  $x \sim N(0, I)$ 。最后令  $y = Lx + \mu$ 。这是正确的，因为

$$\text{cov}[y] = L \text{cov}[x] L^\top = LIL^\top = \Sigma \quad (23.8)$$

## 通俗解释（一步一步来）

### 1) 这一节在干嘛？

- 讲两类“从标准分布直接取样”的小工具：
  1. 用 cdf 的反函数来取样（逆变换采样）——适合单变量分布；
  2. Box–Muller（极坐标）法——高效地生成**标准正态**（高斯）样本，再扩展到多元高斯。
- 这些都是**更复杂采样算法的“基础部件”**（子程序）。

### 2) 用 cdf 的反函数取样（逆变换采样）

#### 核心想法

- 随机数发生器能给你很多**均匀**分布的数  $U \sim U(0, 1)$ 。
- 如果想从某个分布  $F$  取样，**把均匀数“喂给”**  $F^{-1}$ ：

$$X = F^{-1}(U) \Rightarrow X \text{ 就服从 } F.$$

- 直观：拿一根“高度尺” $u$ （0 到 1 的均匀数），在 $y$  轴上定好高度，找到 cdf 曲线 $F$  与这根水平线的交点，然后垂直落到 $x$  轴，那个 $x$  就是样本。

## 为什么对？

- 因为  $\Pr(F^{-1}(U) \leq x) = \Pr(U \leq F(x)) = F(x)$  (均匀变量在  $[0, 1]$  上,  $\Pr(U \leq y) = y$ )。

## 一个具体例子：指数分布

- 密度:  $\lambda e^{-\lambda x}$  (只在  $x \geq 0$  上有值)。
- cdf:  $F(x) = 1 - e^{-\lambda x}$ 。
- 反函数:  $F^{-1}(p) = -\ln(1 - p)/\lambda$ 。
- 采样法：
  - 生成  $u \sim U(0, 1)$ ;
  - 输出  $x = -\ln(1 - u)/\lambda$ 。

因为  $1 - U$  还是均匀，常直接写  $x = -\ln(U)/\lambda$ ，更省一步。

## 适用与注意

- 优点：**思路简单透明，只要能算出  $F^{-1}$ 。
  - 限制：**有些分布  $F^{-1}$  **没有解析式**（比如对数正态的分位数没有简单闭式），就需要数值法或换别的采样法。
- 

## 3) 从正态分布采样：Box–Muller (极坐标) 法

### 目标

- 快速得到两份**独立的**标准正态样本  $x_1, x_2 \sim N(0, 1)$ 。

### 步骤 (“极坐标”/Marsaglia 极坐标变体)

- 在正方形  $(-1, 1) \times (-1, 1)$  中**均匀采样**一对  $(z_1, z_2)$ 。
- 若  $r^2 = z_1^2 + z_2^2 > 1$ , **丢弃重来** (拒绝采样)；若  $\leq 1$ , 保留。
  - 保留下来的点在**单位圆内**均匀分布；圆的面积是  $\pi$ , 所以圆内密度是  $1/\pi$ 。
- 计算

$$x_i = z_i \sqrt{\frac{-2 \ln r^2}{r^2}}, \quad i = 1, 2.$$

- 得到的  $(x_1, x_2)$  是两份**相互独立**的  $N(0, 1)$  样本。

### 直觉

- 在极坐标下，标准二维正态的**角度**是均匀的，**半径**的分布可以通过 $-2 \ln(\cdot)$ 的变换得到。上述变换正是把“圆内均匀”映到“二维高斯”的正规手法。

## 多元高斯怎么做？

- 想要  $Y \sim N(\mu, \Sigma)$ ：
  - 先用 Box–Muller 生成  $X \sim N(0, I)$ ；
  - 对  $\Sigma$  做 Cholesky 分解  $\Sigma = LL^\top$  ( $L$  为下三角)；
  - 设  $Y = LX + \mu$ 。
- 为什么对？因为  $\text{cov}[Y] = L \text{cov}[X] L^\top = LIL^\top = \Sigma$ ，且均值从 0 平移到  $\mu$ 。

## 小提示

- Box–Muller 还有“正弦余弦”版本：用两颗独立均匀数  $U_1, U_2$ ，取

$$X_1 = \sqrt{-2 \ln U_1} \cos(2\pi U_2), \quad X_2 = \sqrt{-2 \ln U_1} \sin(2\pi U_2),$$

本质等价。现代库里也常有更快的“Ziggurat”算法，但原理上 Box–Muller 已够用。

## 4) 符号与概念小抄

- $U(0, 1)$ : 0 到 1 区间上的均匀分布。
- $F$ : cdf (累计分布函数)； $F^{-1}$ : 其反函数 (分位数函数)。
- $I(\cdot)$ : 指示函数，条件为真取 1，否则取 0。
- Cholesky 分解：把正定矩阵  $\Sigma$  写成  $LL^\top$  的形式，便于把独立标准正态“拉伸/旋转”成目标协方差。

## 要点总结

- 逆变换采样**：从任何能算出分位数函数  $F^{-1}$  的单变量分布采样，方法是  $X = F^{-1}(U)$ ，其中  $U \sim U(0, 1)$ 。
  - 指数分布示例**： $X = -\ln(U)/\lambda$ 。
- Box–Muller 法**：先在单位圆内均匀取点，再做

$$x_i = z_i \sqrt{\frac{-2 \ln r^2}{r^2}}$$

得到两份独立的  $N(0, 1)$  样本。

- 多元高斯**：先得  $X \sim N(0, I)$ ，做  $\Sigma = LL^\top$ ，再设  $Y = LX + \mu$ ，即可得到  $N(\mu, \Sigma)$ 。
- 这些方法常作为**更复杂采样算法的底层子程序**，简单、高效、实用。

这一节主要在讲什么？我不太懂

已思考 12s

不急～这节其实在讲三件很实用的小工具，用来“直接从常见分布里取随机样本”，给后面更复杂的算法当积木：

## 一句话总结

教你怎么从标准分布抽样：单变量用“逆 CDF（分位数）法”，正态用“Box–Muller”生两条独立的标准正态，再用 Cholesky 把它们变成任意多元高斯。

## 核心内容（通俗版）

### 1. 逆 CDF（分位数）采样：单变量通用招

- 先从  $U(0, 1)$  抽一个均匀数  $u$ 。
- 把它丢进目标分布的**分位数函数**  $x = F^{-1}(u)$ ，这个  $x$  就是该分布的样本。
- 直觉：在 cdf 曲线上找“高度 =  $u$ ”那一点对应的横坐标。
- **例子**（指数分布）： $x = -\ln(u)/\lambda$ 。

### 2. Box–Muller：一次生成两个标准正态

- 在  $(-1, 1)^2$  里随便取点  $(z_1, z_2)$ ，只保留落在**单位圆**里的点。
- 用一个固定公式把  $(z_1, z_2)$  变成  $(x_1, x_2)$ ，就得到**两份独立**的  $N(0, 1)$  样本。
- 直觉：把“圆里均匀分布的点”用变换挤压/拉伸成“钟形的正态”。

### 3. 从标准正态到任意多元高斯

- 想要  $Y \sim N(\mu, \Sigma)$ ：
 

先用 Box–Muller 得到  $X \sim N(0, I)$ ，  
对  $\Sigma$  做 Cholesky： $\Sigma = LL^\top$ ，  
然后设  $Y = LX + \mu$ 。
- 直觉：先有“标准球形”噪声，再用  $L$  做“旋转+拉伸”，最后平移到均值  $\mu$ 。

## 为什么重要

- 这些方法**简单、通用、快**，常被当作更复杂采样算法（比如重要性采样、MCMC）的基础组件或对照基线。

## 小例子（一眼能用）

- **抽指数( $\lambda$ )**：生成  $u \sim U(0, 1)$ ，令  $x = -\ln(u)/\lambda$ 。
- **抽二维标准正态**：用 Box–Muller 一次就给你  $(x_1, x_2)$ 。
- **抽  $N(\mu, \Sigma)$** ：先 Box–Muller 得到  $X \sim N(0, 1)$  向量，再  $Y = LX + \mu$ 。

## 易混点

- 逆 CDF 只要能算出  $F^{-1}$  就很方便；有些分布没简洁的  $F^{-1}$ ，就要数值解或换别的方法。
- Box-Muller 的两个输出是**独立**的标准正态；别忘了这是它的优势之一。

## 要点小结

- 单变量**：逆 CDF (分位数) 法 =  $X = F^{-1}(U)$ 。
- 标准正态**：Box-Muller 一次产两样本。
- 多元高斯**：Cholesky 把标准正态“变形”成目标协方差  $\Sigma$ 。
- 这些都是**标准分布采样的基础积木**，后续算法常会重复用到。

翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

### 23.3 Rejection sampling

When the inverse cdf method cannot be used, one simple alternative is to use rejection sampling, which we now explain.

#### 23.3.1 Basic idea

In rejection sampling, we create a proposal distribution  $q(x)$  which satisfies  $Mq(x) \geq \tilde{p}(x)$ , for some constant  $M$ , where  $\tilde{p}(x)$  is an unnormalized version of  $p(x)$  (i.e.,  $p(x) = \tilde{p}(x)/Z_p$  for some possibly unknown constant  $Z_p$ ). The function  $Mq(x)$  provides an upper envelope for  $\tilde{p}(x)$ . We then sample  $x \sim q(x)$ , which corresponds to picking a random  $x$  location, and then we sample  $u \sim U(0, 1)$ , which corresponds to picking a random height ( $y$  location) under the envelope. If  $u > \tilde{p}(x)/Mq(x)$ , we reject the sample, otherwise we accept it. See Figure 23.2(a), where the acceptance region is shown shaded, and the rejection region is the white region between the shaded zone and the upper envelope.

We now prove that this procedure is correct. Let

$$S = \{(x, u): u \leq \tilde{p}(x)/Mq(x)\}, S_0 = \{(x, u): x \leq x_0, u \leq \tilde{p}(x)/Mq(x)\} \quad (23.9)$$

Then the cdf of the accepted points is given by

$$P(x \leq x_0 | x \text{ accepted}) = P(x \leq x_0, x \text{ accepted}) / P(x \text{ accepted}) \quad (23.10)$$

$$= \int_{-\infty}^{\infty} I((x, u) \in S) q(x) du dx \int_{-\infty}^{\infty} I((x, u) \in S) q(x) du dx = x_0 - \infty$$

$$\sim p(x) dx \approx -\infty \sim p(x) dx \quad (23.11)$$

which is the cdf of  $p(x)$ , as desired.

How efficient is this method? Since we generate with probability  $q(x)$  and accept with probability  $\tilde{p}(x) M q(x)$ , the probability of acceptance is

$$p(\text{accept}) = \tilde{p}(x) M q(x) q(x) dx = 1/M \tilde{p}(x) dx \quad (23.12)$$

Hence we want to choose  $M$  as small as possible while still satisfying  $M q(x) \geq \tilde{p}(x)$ .

### 23.3.2 Example

For example, suppose we want to sample from a Gamma distribution:

$$Ga(x|\alpha, \lambda) = \frac{1}{\Gamma(\alpha)} x^{\alpha-1} \lambda^\alpha \exp(-\lambda x) \quad (23.13)$$

One can show that if  $X_i$  iid  $\sim \text{Expon}(\lambda)$ , and  $Y = X_1 + \dots + X_k$ , then  $Y \sim Ga(k, \lambda)$ . For non-integer shape parameters, we cannot use this trick. However, we can use rejection sampling using a  $Ga(k, \lambda - 1)$  distribution as a proposal, where  $k = \alpha$ . The ratio has the form

$$p(x) q(x) = Ga(x|\alpha, \lambda) / Ga(x|k, \lambda - 1) = x^{\alpha-1} \lambda^\alpha \exp(-\lambda x) /$$

$$\Gamma(\alpha) x^{k-1} (\lambda - 1)^k \exp(-(\lambda - 1)x) / \Gamma(k) \quad (23.14)$$

$$= \Gamma(k) \lambda^\alpha \Gamma(\alpha) (\lambda - 1)^k x^{\alpha-k} \exp(-x) \quad (23.15)$$

This ratio attains its maximum when  $x = \alpha - k$ . Hence

$$M = Ga(\alpha - k|\alpha, \lambda) / Ga(\alpha - k|k, \lambda - 1) \quad (23.16)$$

See Figure 23.2(b) for a plot. (Exercise 23.2 asks you to devise a better proposal distribution based on the Cauchy distribution.)

### 23.3.3 Application to Bayesian statistics

Suppose we want to draw (unweighted) samples from the posterior,  $p(\theta|D) = p(D|\theta)p(\theta)/p(D)$ . We can use rejection sampling with  $\tilde{p}(\theta) = p(D|\theta)p(\theta)$  as the target distribution,  $q(\theta) = p(\theta)$  as our proposal, and  $M = p(D|\hat{\theta})$ , where  $\hat{\theta} = \arg \max p(D|\theta)$  is the MLE; this was first suggested in (Smith and Gelfand 1992). We accept points with probability

$$\tilde{p}(\theta) M q(\theta) = p(D|\theta) p(D|\hat{\theta}) \quad (23.17)$$

Thus samples from the prior that have high likelihood are more likely to be retained in the posterior. Of course, if there is a big mismatch between prior and posterior (which will be the case if the prior is vague and the likelihood is informative), this procedure is very inefficient. We discuss better algorithms later.

### 23.3.4 Adaptive rejection sampling

We now describe a method that can automatically come up with a tight upper envelope  $q(x)$  to any log concave density  $p(x)$ . The idea is to upper bound the log density

with a piecewise linear function, as illustrated in Figure 23.3(a). We choose the initial locations for the pieces based on a fixed grid over the support of the distribution. We then evaluate the gradient of the log density at these locations, and make the lines be tangent at these points. Since the log of the envelope is piecewise linear, the envelope itself is piecewise exponential:

$$q(x) = \sum_i M_i \exp(-\lambda_i(x - x_{i-1})), x_{i-1} < x \leq x_i \quad (23.18)$$

where  $x_i$  are the grid points. It is relatively straightforward to sample from this distribution. If the sample  $x$  is rejected, we create a new grid point at  $x$ , and thereby refine the envelope. As the number of grid points is increased, the tightness of the envelope improves, and the rejection rate goes down. This is known as adaptive rejection sampling (ARS) (Gilks and Wild 1992). Figure 23.3(b-c) gives an example of the method in action. As with standard rejection sampling, it can be applied to unnormalized distributions.

### 23.3.5 Rejection sampling in high dimensions

It is clear that we want to make our proposal  $q(x)$  as close as possible to the target distribution  $p(x)$ , while still being an upper bound. But this is quite hard to achieve, especially in high dimensions. To see this, consider sampling from  $p(x) = N(0, \sigma^2 p)$  using as a proposal  $q(x) = N(0, \sigma^2 q)$ . Obviously we must have  $\sigma^2 q \geq \sigma^2 p$  in order to be an upper bound. In  $D$  dimensions, the optimum value is given by  $M = (\sigma q / \sigma p)^D$ . The acceptance rate is  $1/M$  (since both  $p$  and  $q$  are normalized), which decreases exponentially fast with dimension. For example, if  $\sigma q$  exceeds  $\sigma p$  by just 1%, then in 1000 dimensions the acceptance ratio will be about 1/20,000. This is a fundamental weakness of rejection sampling.

In Chapter 24, we will describe MCMC sampling, which is a more efficient way to sample from high dimensional distributions. Sometimes this uses (adaptive) rejection sampling as a subroutine, which is known as adaptive rejection Metropolis sampling (Gilks et al. 1995).

已思考 1m 11s

下面先给出**不遗漏的中文翻译**；然后用**通俗、详细**的方式解释；最后给出**要点总结**。

(注：原文中“where  $k = \alpha$ ”处疑似排版缺失，按上下文应为“取  $\alpha$  的整数部分/就近整数”之类；我在翻译里保留原文，并在解释里提示这一点。)

# 原文翻译

## 23.3 拒绝采样 (Rejection sampling)

当无法使用逆 cdf 方法时，一个简单的替代方法是**拒绝采样**，我们现在来解释它。

### 23.3.1 基本思想

在拒绝采样中，我们构造一个提议 (proposal) 分布  $q(x)$ ，使得对某个常数  $M$  有  $Mq(x) \geq \tilde{p}(x)$ ，其中  $\tilde{p}(x)$  是  $p(x)$  的一个未归一化版本（即  $p(x) = \tilde{p}(x)/Z_p$ ，其中  $Z_p$  可能未知）。函数  $Mq(x)$  为  $\tilde{p}$  提供了一个上包络。接着我们先从  $q(x)$  抽样  $x \sim q(x)$ （对应于在  $x$  轴上随机取一个位置），然后再从  $U(0, 1)$  抽样  $u \sim U(0, 1)$ （对应于在包络下方随机取一个“高度”（ $y$  轴位置））。如果

$$u > \frac{\tilde{p}(x)}{Mq(x)},$$

我们就**拒绝**该样本；否则就**接受**。见图 23.2(a)，其中阴影部分为接受区域，阴影与上包络之间的白色区域为拒绝区域。

我们现在证明该过程是正确的。令

$$S = \{(x, u) : u \leq \tilde{p}(x)/(Mq(x))\}, \quad S_0 = \{(x, u) : x \leq x_0, u \leq \tilde{p}(x)/(Mq(x))\} \quad (23.9)$$

则被接受点的 cdf 为

$$\begin{aligned} P(x \leq x_0 \mid x \text{ accepted}) &= \frac{P(x \leq x_0, x \text{ accepted})}{P(x \text{ accepted})} \\ &\stackrel{(23.10)}{=} \frac{\iint_{\{(x, u) \in S_0\}} q(x) du dx}{\iint_{\{(x, u) \in S\}} q(x) du dx} \\ &= \frac{\int_{-\infty}^{x_0} \tilde{p}(x) dx}{\int_{-\infty}^{\infty} \tilde{p}(x) dx} \stackrel{(23.11)}{=} \end{aligned}$$

这正是  $p(x)$  的 cdf，如所希望的那样。

这种方法的效率如何？由于我们以概率  $q(x)$  生成（样本），并以概率  $\tilde{p}(x)/(Mq(x))$  接受，因此**接受概率**为

$$p(\text{accept}) = \int \frac{\tilde{p}(x)}{Mq(x)} q(x) dx = \frac{1}{M} \int \tilde{p}(x) dx. \quad (23.12)$$

因此我们希望在满足  $Mq(x) \geq \tilde{p}(x)$  的同时，让  $M$  尽可能小。

### 23.3.2 示例

例如，假设我们希望从 Gamma 分布采样：

$$\text{Ga}(x \mid \alpha, \lambda) = \frac{1}{\Gamma(\alpha)} x^{\alpha-1} \lambda^\alpha \exp(-\lambda x) \quad (23.13)$$

可以证明：若  $X_i \stackrel{\text{iid}}{\sim} \text{Expon}(\lambda)$ ，且  $Y = X_1 + \dots + X_k$ ，则  $Y \sim \text{Ga}(k, \lambda)$ 。当形状参数不是整数时，我们不能用这个技巧。然而，我们可以用拒绝采样，并使用  $\text{Ga}(k, \lambda - 1)$  作为

提议分布，其中  $k = \alpha$ 。其比值为

$$\frac{p(x)}{q(x)} = \frac{\text{Ga}(x | \alpha, \lambda)}{\text{Ga}(x | k, \lambda - 1)} = \frac{x^{\alpha-1} \lambda^\alpha \exp(-\lambda x) / \Gamma(\alpha)}{x^{k-1} (\lambda - 1)^k \exp(-(\lambda - 1)x) / \Gamma(k)} \quad (23.14)$$

$$= \frac{\Gamma(k) \lambda^\alpha}{\Gamma(\alpha) (\lambda - 1)^k} x^{\alpha-k} \exp(-x). \quad (23.15)$$

该比值在  $x = \alpha - k$  处取得最大值。因此

$$M = \frac{\text{Ga}(\alpha - k | \alpha, \lambda)}{\text{Ga}(\alpha - k | k, \lambda - 1)}. \quad (23.16)$$

见图 23.2(b) 的作图。(习题 23.2 要求你基于柯西 (Cauchy) 分布设计一个更好的提议分布。)

### 23.3.3 在贝叶斯统计中的应用

假设我们希望从后验  $p(\theta | D) = p(D | \theta)p(\theta)/p(D)$  抽取 (无权重) 样本。我们可以使用拒绝采样，把  $\tilde{p}(\theta) = p(D | \theta)p(\theta)$  作为目标分布，把  $q(\theta) = p(\theta)$  作为提议分布，并取  $M = p(D | \hat{\theta})$ ，其中  $\hat{\theta} = \arg \max p(D | \theta)$  为 MLE；这首先由 (Smith and Gelfand 1992) 提出。我们以概率

$$\frac{\tilde{p}(\theta)}{Mq(\theta)} = \frac{p(D | \theta)}{p(D | \hat{\theta})} \quad (23.17)$$

接受点。于是，似然高的先验样本更可能在后验中被保留。当然，如果先验与后验之间差异很大（当先验很宽泛而似然信息量很强时常常如此），该过程就会很低效。我们稍后会讨论更好的算法。

### 23.3.4 自适应拒绝采样

我们现在描述一种方法，它可以对任何对数凹的密度  $p(x)$  自动构造一个紧的上包络  $q(x)$ 。思想是：用一条分段线性函数来上界住对数密度，如图 23.3(a) 所示。我们根据分布支持集上的一个固定网格来选择这些分段的初始位置。然后在这些位置计算对数密度的梯度，并让这些直线在这些点相切。

由于包络的对数是分段线性的，包络本身就是分段指数的：

$$q(x) = M_i \lambda_i \exp(-\lambda_i(x - x_{i-1})), \quad x_{i-1} < x \leq x_i \quad (23.18)$$

其中  $x_i$  是网格点。从该分布采样相对直观。如果样本  $x$  被拒绝，我们就在  $x$  处创建一个新的网格点，从而细化包络。随着网格点数量的增加，包络的紧密性改善，拒绝率下降。这称为**自适应拒绝采样** (ARS) (Gilks and Wild 1992)。图 23.3(b-c) 给出了该方法的一个示例。与标准拒绝采样一样，它可用于未归一化分布。

### 23.3.5 高维中的拒绝采样

很明显，我们希望让提议  $q(x)$  尽可能接近目标分布  $p(x)$ ，同时仍是一个上界。但这很难做到，尤其在高维中。为说明这一点，考虑用  $q(x) = N(0, \sigma_q^2 I)$  作为提议，从  $p(x) = N(0, \sigma_p^2 I)$  采样。显然我们必须有  $\sigma_q^2 \geq \sigma_p^2$  才能成为上界。在  $D$  维中，最优值是  $M = (\sigma_q/\sigma_p)^D$ 。接受率为  $1/M$ （因为  $p$  与  $q$  都已归一化），而该接受率会随维度指数级下降。举例来说，如果  $\sigma_q$  只比  $\sigma_p$  大 1%，那么在 1000 维中，接受率大约为  $1/20,000$ 。这是拒绝采样的一个根本缺陷。

在第 24 章，我们将介绍 MCMC 采样，它是从高维分布采样的更高效方法。有时它会把（自适应）拒绝采样作为一个子程序，这被称为**自适应拒绝 Metropolis 采样** (Gilks et al. 1995)。

---

## 通俗解释（一步步讲解）

### 1) 拒绝采样在干什么？（“扔飞镖”直觉）

- 目标：从目标分布  $p(x)$  抽样，但  $p(x)$  可能只知道形状 ( $\tilde{p}(x)$ )，不知道归一化常数  $Z_p$ ，也可能没有简洁的  $F^{-1}$ 。
- 做法：找一个**容易抽样**的分布  $q(x)$ ，并放大一个倍数  $M$ ，让  $Mq(x)$  处处高于  $\tilde{p}(x)$  ——就像在曲线  $\tilde{p}(x)$  上方搭一条“棚子”（**上包络**）。
- 算法像“往棚子里扔飞镖”：
  1. 先按  $q(x)$  选一个横坐标  $x$ （在哪儿扔）。
  2. 再从  $U(0, 1)$  选一个纵向高度  $u$ （飞镖落在多高）。
  3. 如果  $u \leq \tilde{p}(x)/(Mq(x))$ ，就算“**飞镖落在目标曲线下**”，**接受**；否则**拒绝**，重来。
- 结论：留下来的  $x$  的分布**恰好就是**  $p(x)$ （哪怕你不知道  $Z_p$ ）。

### 2) 为什么它是对的？（不硬核的证明思路）

- 想象你在“棚子”下（面积与概率对应）均匀打点：
  - 在某个区间  $(-\infty, x_0]$  内，能“打到目标曲线下”的点占比，恰是  $\frac{\int_{-\infty}^{x_0} \tilde{p}(x) dx}{\int_{-\infty}^{\infty} \tilde{p}(x) dx}$ ，这就是“**目标的 cdf**”。
- 所以**接受样本的 cdf = 目标的 cdf**。

### 3) 效率从哪里来？如何提高？

- **接受率** = 目标曲线的“面积” ( $\int \tilde{p}$ ) ÷ 棚子的“面积” ( $M \int q = M \cdot 1$ ) =  $1/M \cdot \int \tilde{p}$ 。
  - 若  $p$  已归一化， $\int \tilde{p} = 1$ ，接受率就是  $1/M$ 。
- 因此：

- $M$  越小越好 (棚子越贴合目标)。
- $q$  越像  $p$  越好 (容易让  $Mq$  贴住  $\tilde{p}$  而不穿过)。

#### 4) Gamma 分布的例子 (非整数形状)

- 知识点: 若把  $k$  个独立的指数( $\lambda$ )相加, 和服从  $\text{Ga}(k, \lambda)$  (当  $k$  是整数)。
- 若  $\alpha$  不是整数, 就不能用“相加指数”的小技巧。于是:
  - 选  $\text{Ga}(k, \lambda - 1)$  当提议 ( $k$  与  $\alpha$  接近, 原文写“ $k = \alpha$ ”疑似缺符号, 通常取“接近  $\alpha$  的整数”以贴近目标)。
  - 推出比值  $\frac{p(x)}{q(x)} \propto x^{\alpha-k} e^{-x}$ , 其最大在  $x = \alpha - k$ 。
  - 由此得到最小可行的  $M$  (式 23.16)。
- 含义: 找一个“形状接近”的  $q$ , 并精确估算“最坏处”的放缩倍数  $M$ 。

#### 5) 在贝叶斯后验里的用法 (Prior→Posterior 的筛选)

- 目标: 从后验  $p(\theta | D)$  取样。
- 做法:
  - 取  $q(\theta) = p(\theta)$  (先验容易抽)。
  - 目标未归一化密度  $\tilde{p}(\theta) = p(D | \theta)p(\theta)$ 。
  - 最大似然  $M = p(D | \hat{\theta})$  给出一个全局上界 ( $\forall \theta, p(D | \theta) \leq p(D | \hat{\theta})$ )。
  - 接受概率 =  $p(D | \theta)/p(D | \hat{\theta})$ 。
- 直觉: 先从先验抓一把参数, 再用“模型对数据的解释力”(似然)做打分, 分高者留下。
- 局限: 若先验很宽而后验很窄 (数据很“挑”), 大多数先验样本解释力低, 都会被拒——效率很差。

#### 6) 自适应拒绝采样 (ARS) : 让“棚子”自动越搭越贴

- 适用前提: \*\*log-concave (对数凹) \*\*的密度 (很多常见单峰分布满足, 如正态、Gamma 的某些参数范围等)。
- 思路:
  1. 在  $x$  轴上放一组网格点。
  2. 计算  $\log p(x)$  及其斜率 (导数), 用这些斜率作切线, 把  $\log p$  上界住 (分段直线)。
  3. 指数回去, 就得到一个分段指型的上包络  $q(x)$  (式 23.18)。这类分布好采样。
  4. 采到的  $x$  若被拒, 就把它加入网格, 细化包络。反复迭代, 包络越来越贴, 接受率越来越高。
- 优点: 不需要知道归一化常数; 会自动“贴合”目标, 减少浪费。

## 7) 高维为什么不行?

- 在  $D$  维, 哪怕你把提议选成同心的高斯  $q = N(0, \sigma_q^2 I)$ , 目标是  $N(0, \sigma_p^2 I)$ , 只要  $\sigma_q > \sigma_p$ , 最优的  $M = (\sigma_q/\sigma_p)^D$ 。
  - 接受率  $\approx (\sigma_p/\sigma_q)^D$  指数级变小!
    - 例如  $\sigma_q$  仅比  $\sigma_p$  大 1% (1.01 倍), 在  $D = 1000$  时接受率  $\approx 1/20000$ 。
  - 这就是拒绝采样的致命缺陷: 高维效率灾难。
  - 解决方向: 第 24 章的 MCMC 更适合高维; 有时把 (自适应) 拒绝作为它的子步骤, 称 ARMS (Adaptive Rejection Metropolis Sampling)。
- 

## 要点总结

- **拒绝采样:** 用易采的  $q(x)$  搭“上包络” $Mq(x)$  覆盖目标  $\tilde{p}(x)$ ; 先抽  $x \sim q$ , 再抽  $u \sim U(0, 1)$ , 以  $\tilde{p}(x)/(Mq(x))$  的阈值接受/拒绝。
- **正确性:** 被接受的  $x$  的分布恰为目标  $p(x)$ , 即使  $Z_p$  未知。
- **效率:** 接受率  $\propto 1/M$ ; 让  $q$  像  $p$ 、 $M$  尽量小。
- **Gamma 示例:** 用形状相近的 Gamma 当提议, 最大比值位置  $x = \alpha - k$  给出紧的  $M$ 。
- **贝叶斯应用:** 以先验为提议, 按似然比  $p(D | \theta)/p(D | \hat{\theta})$  过滤; 先验宽、后验窄时效率差。
- **ARS:** 对对数凹密度, 用分段线性上界  $\log p$ , 得到分段指数的包络并**自适应细化**, 接受率逐步提高。
- **高维警告:** 接受率随维度**指数级下降** (如 1% 的方差膨胀在 1000 维只剩  $\sim 1/20000$ ), 拒绝采样在高维通常不可行; 转而用 MCMC/ARMS。

这一节主要在讲什么? 我不太懂

已思考 18s

不急~这一节 (23.3) 讲的是**拒绝采样** (rejection sampling) : 当“逆 CDF 法”用不了时, 怎样用一个**容易取样**的分布来“筛”出目标分布的样本。再顺带讲了它在贝叶斯里的用法、一个自适应改进 (ARS), 以及为什么在高维会很差。

## 一句话版

**用好抽的提议分布  $q(x)$  搭一个“上包络” $Mq(x)$  盖住目标的未归一化密度  $\tilde{p}(x)$ , 先从  $q$  抽  $x$ , 再扔个均匀数  $u$ , 若  $u \leq \tilde{p}(x)/(Mq(x))$  就收下; 被收下的  $x$  就服从目标  $p(x)$ 。**

## 直觉版（扔飞镖）

- 把  $Mq(x)$  想成顶上的棚子， $\tilde{p}(x)$  是棚子下面那条目标曲线。
- 步骤：
  1. 先按  $q(x)$  选位置  $x$ ；
  2. 再抽  $u \sim U(0, 1)$  选高度；
  3. 如果高度没超过目标曲线占棚子的比例  $\tilde{p}(x)/(Mq(x))$ ，就接受  $x$ ，否则拒绝。
- 结果：留下来的点正好分布成你想要的  $p(x)$ （即便你不知道归一化常数）。

## 好不好用，取决于两件事

- $q$  像不像  $p$ ：越像越好。
- $M$  能不能小：还要处处满足  $Mq(x) \geq \tilde{p}(x)$ 。
- 接受率  $\approx (\int \tilde{p})/M$ ；若  $p$  已归一化就是  $1/M$ 。

## 书里举的几件事

- **Gamma 示例：**非整数形状时，用形状接近的  $Ga(k, \lambda - 1)$  做提议，算出  $\frac{p}{q}$  的最大值来设最小可行的  $M$ 。
- **贝叶斯应用：**想从后验  $p(\theta|D)$  取样时，用先验  $q(\theta) = p(\theta)$  做提议，接受概率是似然比  $p(D|\theta)/p(D|\hat{\theta})$ 。先验很宽、后验很窄时会很低效。
- **自适应拒绝采样 (ARS)：**对对数凹的密度，把  $\log p(x)$  用分段线性切线往上“罩住”，指数回去得到分段指数的包络，采样被拒就把那个点加入网格，越来越贴，拒绝率下降。
- **高维的致命点：**接受率随维度指数组变差。哪怕提议高斯的方差只比目标大 1%，到 1000 维接受率也可能掉到约  $1/20000$ 。所以高维一般改用 MCMC（有时把 ARS 当子步骤，称 ARMS）。

## 什么时候用/不用

- **适用：**一维或低维、能找到和目标形状接近的  $q$ ，或目标是对数凹（可用 ARS）。
- **不适用：**高维、目标尾部比提议更“厚”（找不到有限  $M$ ）、或  $q$  与  $p$  差很大。

## 要点小结

- 核心： $x \sim q$ ,  $u \sim U(0, 1)$ , 以  $\tilde{p}(x)/(Mq(x))$  为阈值接受，得到目标  $p$  的样本。
- 效率：接受率  $\propto 1/M$ ；让  $q$  贴  $p$ 、 $M$  小。
- 变体：贝叶斯里可用先验当提议；ARS 适合对数凹密度并会自我改进包络。
- 警告：**高维严重低效**，通常改用 MCMC/ARMS。

翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

### 23.4 Importance sampling

We now describe a Monte Carlo method known as importance sampling for approximating integrals of the form

$$I = E[f] = \int f(x)p(x)dx \quad (23.19)$$

#### 23.4.1 Basic idea

The idea is to draw samples  $x$  in regions which have high probability,  $p(x)$ , but also where  $|f(x)|$  is large. The result can be super efficient, meaning it needs less samples than if we were to sample from the exact distribution  $p(x)$ . The reason is that the samples are focussed on the important parts of space. For example, suppose we want to estimate the probability of a rare event. Define  $f(x) = I(x \in E)$ , for some set  $E$ . Then it is better to sample from a proposal of the form  $q(x) \propto f(x)p(x)$  than to sample from  $p(x)$  itself.

Importance sampling samples from any proposal,  $q(x)$ . It then uses these samples to estimate the integral as follows:

$$E[f] = \int f(x)p(x)q(x)dx \approx \frac{1}{S} \sum_{s=1}^S w_s f(x_s) = \hat{I} \quad (23.20)$$

where  $w_s = p(x_s) q(x_s)$  are the importance weights. Note that, unlike rejection sampling, we use all the samples.

How should we choose the proposal? A natural criterion is to minimize the variance of the estimate  $\hat{I} = \sum w_s f(x_s)$ . Now

$$\text{var}_q(x)[f(x)w(x)] = \int q(x) [f(x)w(x)]^2 - \hat{I}^2 \quad (23.21)$$

Since the last term is independent of  $q$ , we can ignore it.

By Jensen's inequality, we have the following lower bound:

$$\int q(x) f(x)w(x) \geq (\int q(x) |f(x)w(x)|)^2 = \int |f(x)|p(x)dx \quad (23.22)$$

The lower bound is obtained when we use the optimal importance distribution:

$$q^*(x) = \frac{|f(x)|p(x)}{\int |f(x)|p(x)dx} \quad (23.23)$$

When we don't have a particular target function  $f(x)$  in mind, we often just try to make  $q(x)$  as close as possible to  $p(x)$ . In general, this is difficult, especially in high dimensions, but it is possible to adapt the proposal distribution to improve the approximation. This is known as adaptive importance sampling (Oh and Berger 1992).

#### 23.4.2 Handling unnormalized distributions

It is frequently the case that we can evaluate the

unnormalized target distribution,  $\tilde{p}(x)$ , but not its normalization constant,  $Z_p$ . We may also want to use an unnormalized proposal,  $\tilde{q}(x)$ , with possibly unknown normalization constant  $Z_q$ . We can do this as follows. First we evaluate

$$E[f] = Z_q \int p(x) \tilde{p}(x) q(x) dx \approx Z_q \int f(x) \tilde{p}(x) q(x) dx \approx Z_q \int f(x) \tilde{w}_s ds \quad (23.24)$$

where  $\tilde{w}_s = \tilde{p}(x_s) \tilde{q}(x_s)$  is the unnormalized importance weight. We can use the same set of samples to evaluate the ratio  $Z_p/Z_q$  as follows:

$$\frac{Z_p}{Z_q} = \frac{1}{Z_q} \int p(x) \tilde{p}(x) dx = \frac{1}{Z_q} \int p(x) \tilde{p}(x) q(x) dx \approx \frac{1}{Z_q} \int \tilde{w}_s ds \quad (23.25)$$

Hence

$$\hat{I} = \frac{1}{Z_q} \int f(x) \tilde{p}(x) dx = \frac{1}{Z_q} \int f(x) \tilde{w}_s ds \quad (23.26)$$

where

$$w_s = \tilde{w}_s / \sum \tilde{w}_s \quad (23.27)$$

are the normalized importance weights. The resulting estimate is a ratio of two estimates, and hence is biased. However, as  $S \rightarrow \infty$ , we have that  $\hat{I} \rightarrow I$ , under weak assumptions (see e.g., (Robert and Casella 2004) for details).

### 23.4.3 Importance sampling for a DGM: likelihood weighting

We now describe a way to use importance sampling to generate samples from a distribution which can be represented as a directed graphical model (Chapter 10). If we have no evidence, we can sample from the unconditional joint distribution of a DGM  $p(x)$  as follows: first sample the root nodes, then sample their children, then sample their children, etc. This is known as ancestral sampling. It works because, in a DAG, we can always topologically order the nodes so that parents precede children. (Note that there is no equivalent easy method for sampling from an unconditional undirected graphical model.)

Now suppose we have some evidence, so some nodes are “clamped” to observed values, and we want to sample from the posterior  $p(x|D)$ . If all the variables are discrete, we can use the following simple procedure: perform ancestral sampling, but as soon as we sample a value that is inconsistent with an observed value, reject the whole sample and start again. This is known as logic sampling (Henrion 1988).

Needless to say, logic sampling is very inefficient, and it cannot be applied when we have real-valued evidence.

However, it can be modified as follows. Sample unobserved variables as before, conditional on their parents. But don't sample observed variables; instead we just use their observed values. This is equivalent to using a proposal of the form

$$q(x) = t \in E p(x_t | x_{pa}(t)) \quad t \in E \quad \delta x_t \sim p(x_t | x_{pa}(t)) \quad (23.28)$$

where  $E$  is the set of observed nodes, and  $x_t$  is the observed value for node  $t$ . We should therefore give the overall sample an importance weight as follows:

$$w(x) = p(x) q(x) = t \in E p(x_t | x_{pa}(t)) p(x_t | x_{pa}(t)) \quad t \in E \\ p(x_t | x_{pa}(t)) \quad 1 = t \in E p(x_t | x_{pa}(t)) \quad (23.29)$$

This technique is known as likelihood weighting (Fung and Chang 1989; Shachter and Peot 1989).

#### 23.4.4 Sampling importance resampling (SIR)

We can draw unweighted samples from  $p(x)$  by first using importance sampling (with proposal  $q$ ) to generate a distribution of the form

$$p(x) \approx s w_s \delta x_s(x) \quad (23.30)$$

where  $w_s$  are the normalized importance weights. We then sample with replacement from Equation 23.30, where the probability that we pick  $x_s$  is  $w_s$ . Let this procedure induce a distribution denoted by  $\hat{p}$ . To see that this is valid, note that

$$\hat{p}(x \leq x_0) = s \int (x_s \leq x_0) w_s = s \int (x_s \leq x_0) \hat{p}(x_s) / q(x_s) s \\ \hat{p}(x_s) / q(x_s) \quad (23.31)$$

$$\rightarrow \int (x \leq x_0) \hat{p}(x) q(x) q(x) dx \hat{p}(x) q(x) q(x) dx \quad (23.32)$$

$$= \int (x \leq x_0) \hat{p}(x) dx \hat{p}(x) dx = \int (x \leq x_0) p(x) dx = p(x \leq x_0) \quad (23.33)$$

This is known as sampling importance resampling (SIR) (Rubin 1998). The result is an unweighted approximation of the form

$$p(x) \approx 1 S S s=1 \delta x_s(x) \quad (23.34)$$

Note that we typically take  $S$ .

This algorithm can be used to perform Bayesian inference in low-dimensional settings (Smith and Gelfand 1992).

That is, suppose we want to draw (unweighted) samples from the posterior,  $p(\theta|D) = p(D|\theta)p(\theta)/p(D)$ . We can use importance sampling with  $\hat{p}(\theta) = p(D|\theta)p(\theta)$  as the unnormalized posterior, and  $q(\theta) = p(\theta)$  as our proposal.

The normalized weights have the form

$$w_s = \hat{p}(\theta_s) / q(\theta_s) s \hat{p}(\theta_s) / q(\theta_s) = p(D|\theta_s) s p(D|\theta_s) \\ (23.35)$$

We can then use SIR to sample from  $p(\theta|D)$ .

Of course, if there is a big discrepancy between our proposal (the prior) and the target (the posterior), we will

need a huge number of importance samples for this technique to work reliably, since otherwise the variance of the importance weights will be very large, implying that most samples carry no useful information. (This issue will come up again in Section 23.5, when we discuss particle filtering.)

已思考 1m 44s

下面先给出**不遗漏的中文翻译**；然后用**通俗、详细**的方式解释；最后给出**要点总结**。

(注：原文少量排版处可能有小瑕疵，我在“解释”部分会提示常见/标准写法，但**翻译部分严格按原文**。)

---

## 原文翻译

### 23.4 重要性采样 (Importance sampling)

我们现在描述一种称为**重要性采样**的蒙特卡罗方法，用来近似如下形式的积分

$$I = \mathbb{E}[f] = \int f(x)p(x) dx \quad (23.19)$$

#### 23.4.1 基本思想

其思想是在  $p(x)$  概率高的区域，同时在  $|f(x)|$  也大的区域抽取样本  $x$ 。其结果可能是**超级高效的**，意思是它需要的样本数比我们直接从真实分布  $p(x)$  采样还要少。原因在于样本被集中在空间中的**重要部分**。例如，假设我们想估计一个**稀有事件**的概率。令  $f(x) = \mathbf{I}(x \in E)$ ，其中  $E$  是某个集合。那么从形如  $q(x) \propto f(x)p(x)$  的提议分布采样要比从  $p(x)$  自身采样更好。

重要性采样从**任意**提议分布  $q(x)$  采样。随后用这些样本来估计积分如下：

$$\mathbb{E}[f] = \int f(x) \frac{p(x)}{q(x)} q(x) dx \approx \frac{1}{S} \sum_{s=1}^S w_s f(x^{(s)}) = \hat{I} \quad (23.20)$$

其中  $w_s = \frac{p(x^{(s)})}{q(x^{(s)})}$  是**重要性权重**。注意，与拒绝采样不同，我们**使用所有样本**。

我们应当如何选择提议分布？一个自然的准则是**最小化估计量**  $\hat{I} = \frac{1}{S} \sum_s w_s f(x^{(s)})$  的**方差**。现在有

$$\text{var}_{q(x)}[f(x)w(x)] = \mathbb{E}_{q(x)}[f^2(x)w^2(x)] - I^2 \quad (23.21)$$

由于最后一项与  $q$  无关，我们可以忽略它。由**Jensen 不等式**可得如下**下界**：

$$\mathbb{E}_{q(x)}[f^2(x)w^2(x)] \geq (\mathbb{E}_{q(x)}[|f(x)w(x)|])^2 = \left(\int |f(x)|p(x) dx\right)^2 \quad (23.22)$$

当我们使用**最优的重要性分布**时可达到该下界：

$$q^*(x) = \frac{|f(x)|p(x)}{\int |f(x')|p(x') dx'} \quad (23.23)$$

当我们并没有某个特定的目标函数  $f(x)$  时，我们经常只尝试让  $q(x)$  尽可能接近  $p(x)$ 。一般来说，这很困难，尤其是在高维中，但可以**自适应地**调整提议分布以改进近似。这被称为**自适应重要性采样** (Oh and Berger 1992)。

### 23.4.2 处理未归一化分布

很多情况下我们能计算**未归一化**的目标分布  $\tilde{p}(x)$ ，但不能计算其归一化常数  $Z_p$ 。我们也可能想使用一个**未归一化**的提议  $\tilde{q}(x)$ ，其归一化常数  $Z_q$  也可能未知。我们可以如下进行。首先我们计算

$$E[f] = \frac{Z_q}{Z_p} \int f(x) \frac{\tilde{p}(x)}{\tilde{q}(x)} q(x) dx \approx \frac{Z_q}{Z_p} \frac{1}{S} \sum_{s=1}^S \tilde{w}_s f(x^{(s)}) \quad (23.24)$$

其中  $\tilde{w}_s = \frac{\tilde{p}(x^{(s)})}{\tilde{q}(x^{(s)})}$  是**未归一化权重**。我们可以用**同一批样本**来估计**比值**  $Z_p/Z_q$ ，如下：

$$\frac{Z_p}{Z_q} = \frac{1}{Z_q} \int \tilde{p}(x) dx = \int \frac{\tilde{p}(x)}{\tilde{q}(x)} q(x) dx \approx \frac{1}{S} \sum_{s=1}^S \tilde{w}_s \quad (23.25)$$

因此

$$\hat{I} = \frac{\frac{1}{S} \sum_s \tilde{w}_s f(x^{(s)})}{\frac{1}{S} \sum_s \tilde{w}_s} = \sum_{s=1}^S w_s f(x^{(s)}) \quad (23.26)$$

其中

$$w_s = \frac{\tilde{w}_s}{\sum_s \tilde{w}_s} \quad (23.27)$$

**是归一化的重要性权重**。由此得到的估计是**两个估计之比**，因此是有偏的。然而，当  $S \rightarrow \infty$  时，在较弱条件下有  $\hat{I} \rightarrow I$  (细节见例如 Robert and Casella 2004)。

### 23.4.3 用于有向图模型 (DGM) 的重要性采样：似然加权

我们现在描述一种方法，用重要性采样来从一个可以表示为**有向图模型** (第 10 章) 的分布中生成样本。

如果我们没有证据 (观测)，就可以如下从 DGM 的**无条件联合分布**  $p(x)$  采样：先采根节点，再采它们的子节点，再采它们的子节点，等等。这称为**祖先采样** (ancestral sampling)。它之所以有效，是因为在一个 DAG 中，我们总能对节点作**拓扑排序**，使得父节点先于子节点。(注意：对于**无向图模型**的无条件采样并没有等价的简单方法。)

现在假设我们有一些证据，因此一些节点被“夹定”（clamped）在观测值上，我们想从后验  $p(x | D)$  采样。若所有变量都是离散的，我们可以使用以下简单过程：执行祖先采样，但一旦采到的值与某个观测值不一致，就丢弃整个样本并重新开始。这被称为**逻辑采样**（Henrion 1988）。

不用说，逻辑采样**非常低效**，而且在我们有**实值证据**时无法使用。然而，它可以改成如下方式。像之前一样，**对未观测变量**，在给定其父节点的条件下进行采样。**但不要对观测变量进行采样**；我们直接使用它们的观测值。这等价于使用如下形式的提议分布

$$q(x) = \prod_{t \in E} p(x_t | x_{\text{pa}(t)}) \prod_{t \in E} \delta_{x_t^*}(x_t) \quad (23.28)$$

其中  $E$  是观测节点的集合， $x_t^*$  是节点  $t$  的观测值。因此我们应当给整体样本如下**重要性权重**：

$$w(x) = \frac{p(x)}{q(x)} = \frac{\prod_{t \in E} p(x_t | x_{\text{pa}(t)})}{\prod_{t \in E} p(x_t | x_{\text{pa}(t)})} = \prod_{t \in E} p(x_t | x_{\text{pa}(t)}) \quad (23.29)$$

该技术称为**似然加权**（likelihood weighting）（Fung and Chang 1989; Shachter and Peot 1989）。

#### 23.4.4 采样-重要性-再采样 (SIR)

我们可以先用**重要性采样**（提议为  $q$ ）得到如下形式的分布

$$p(x) \approx \sum_s w_s \delta_{x^{(s)}}(x) \quad (23.30)$$

其中  $w_s$  是**归一化的重要性权重**。然后我们**有放回地**从式 (23.30) 再采样，其中选中  $x^{(s)}$  的概率是  $w_s$ 。令该过程所诱导的分布记为  $\hat{p}$ 。要看这为何有效，注意到

$$\hat{p}(x \leq x_0) = \sum_s \mathbf{I}(x^{(s)} \leq x_0) w_s = \frac{\sum_s \mathbf{I}(x^{(s)} \leq x_0) \tilde{p}(x^{(s)}) / q(x^{(s)})}{\sum_s \tilde{p}(x^{(s)}) / q(x^{(s)})} \quad (23.31)$$

$$\rightarrow \frac{\int \mathbf{I}(x \leq x_0) \frac{\tilde{p}(x)}{q(x)} q(x) dx}{\int \frac{\tilde{p}(x)}{q(x)} q(x) dx} \quad (23.32)$$

$$= \frac{\int \mathbf{I}(x \leq x_0) \tilde{p}(x) dx}{\int \tilde{p}(x) dx} = \int \mathbf{I}(x \leq x_0) p(x) dx = p(x \leq x_0) \quad (23.33)$$

这称为**采样-重要性-再采样** (SIR)（Rubin 1998）。其结果是如下形式的**无权重近似**

$$p(x) \approx \frac{1}{S} \sum_{s=1}^S \delta_{x^{(s)}}(x) \quad (23.34)$$

注意我们通常取  $S$  个样本。

该算法可用于低维情形下的贝叶斯推断 (Smith and Gelfand 1992)。也就是说，假设我们要从后验  $p(\theta | D) = p(D | \theta)p(\theta)/p(D)$  抽取 (无权重) 样本。我们可以用重要性采样，令  $\tilde{p}(\theta) = p(D | \theta)p(\theta)$  为未归一化后验， $q(\theta) = p(\theta)$  为提议。归一化权重为

$$w_s = \frac{\tilde{p}(\theta_s)/q(\theta_s)}{\sum_{s'} \tilde{p}(\theta_{s'})/q(\theta_{s'})} = \frac{p(D | \theta_s)}{\sum_{s'} p(D | \theta_{s'})} \quad (23.35)$$

随后我们可用 SIR 从  $p(\theta | D)$  采样。

当然，如果我们的提议 (先验) 与目标 (后验) 之间有很大差异，那么为了让该技术可靠工作，我们需要大量的重要性样本；否则，重要性权重的方差将会很大，意味着大多数样本不携带有用信息。（这一问题会在第 23.5 节再次出现，当我们讨论粒子滤波时。）

---

## 通俗解释（循序渐进）

### 1) 重要性采样想解决什么？

我们要算积分  $I = \int f(x)p(x) dx$  (比如概率、期望、罕见事件概率等)。直接从  $p$  抽样有时浪费巨大：很多样本落在“无关区域” ( $|f|$  很小)，对估计贡献不大。

重要性采样的点子是：从更合适的  $q$  抽样，让样本更多地落在\*\*“重要区域” ( $|f|p$  大的地方)，再用权重\*\*把“抽样偏差”纠回来。

- 公式：从  $q$  抽样  $x^{(s)}$ ，用

$$\hat{I} = \frac{1}{S} \sum_s \underbrace{\frac{p(x^{(s)})}{q(x^{(s)})}}_{\text{权重}} f(x^{(s)})$$

逼近  $I$ 。所有样本都用上 (不像拒绝采样会丢样本)。

- 直觉：想调查“极罕见人群” ( $E$ )，直接从总体抽要抽很久才抽到。倒不如有意多抽  $E$  里的人 (提议  $q \propto fp$ )，再按权重调整，就能高效估计那类人的比例。

### 2) 怎样选一个“好”的 $q$ ？

- 理想目标：让估计的方差最小。
- 推导告诉我们：当

$$q^*(x) \propto |f(x)|p(x)$$

时达到下界，是最优提议 (在知道  $f$  的前提下)。

- 现实里经常不知道特定的  $f$ ，于是通常尽量让  $q \approx p$ ，或自适应调整  $q$  (比如调节均值、协方差、做混合分布等)，这叫自适应重要性采样。
- 小提醒 (工程常识)：

- $q$  的尾巴必须至少和  $p$  一样“厚”，否则权重可能爆炸导致方差无穷。
- 让  $q$  在  $|f|p$  大的地方覆盖充分，避免漏掉关键区域。

### 3) 当 $p, q$ 只有“形状”时（未归一化）

- 只知道  $\tilde{p} \propto p, \tilde{q} \propto q$  也没关系：
  - 先算未归一化权重  $\tilde{w}_s = \tilde{p}/\tilde{q}$ 。
  - 再把它们归一化成  $w_s = \tilde{w}_s / \sum \tilde{w}_s$ 。
  - 用自归一化估计量

$$\hat{I} = \sum_s w_s f(x^{(s)})$$

来近似  $I$ 。

- 这个估计量是有偏的（因为是“两个估计之比”），但只要样本多，它会一致收敛到真实  $I$ 。

### 4) 有向图模型（贝叶斯网络）里的似然加权

- 无证据：按父→子的拓扑顺序采样（祖先采样）。
- 有证据：
  - “逻辑采样”= 先祖先采样，抽到与证据冲突就丢弃（低效）。
  - 似然加权：只对未观测的节点采样；对观测的节点，直接使用其观测值（用“δ 点质量”强制它等于观测值），同时把每个证据节点的条件概率  $p(x_t^* | \text{父母})$  乘到权重里。
- 最终每个样本的权重就是所有证据节点条件概率的乘积。这在离散或连续证据情况下都能用，避免了大量拒绝。

译注：式 (23.28) 在原文中把乘积的指标都写成  $t \in E$ ，易与标准写法混淆。标准理解应是：未观测节点按  $p(x_t | \text{父母})$  采；观测节点用  $\delta$  固定为观测值；权重来自证据节点的  $p(x_t^* | \text{父母})$ 。

### 5) SIR：把“有权重的样本”变成“无权重的样本”

- 第一步 (IS)：得到加权样本集  $\{x^{(s)}, w_s\}$ ，可看成

$$p(x) \approx \sum_s w_s \delta_{x^{(s)}}(x)$$

- 第二步 (R)：按权重有放回重采样，得到一批无权重样本，相当于真的从  $p(x)$  抽的。
- 结果：一个经验分布

$$p(x) \approx \frac{1}{S} \sum_{s=1}^S \delta_{x^{(s)}}(x)$$

- 常见实践：先用很多候选样本算权重，再重采样出较少/同量的无权样本（原文“注意我们通常取  $S$ ”处疑似排版，常见做法是重采样数量  $\leq$  候选数量）。
  - 何时用：在低维贝叶斯推断里，选  $q(\theta) = p(\theta)$ （先验），权重  $\propto p(D | \theta)$ ，再用 SIR 得到近似的后验无权样本。
  - 限制：若先验与后验差很大，权重方差大，大多数样本几乎权重为零——需要巨量样本才稳，这个问题在粒子滤波里还会再见。
- 

## 要点总结

- 目的：**高效近似  $I = \int f(x)p(x) dx$ ，把抽样集中在  $|f|p$  大的“重要区域”。
- 做法：**从  $q$  抽样，用权重  $w = p/q$  纠偏；所有样本都用上。
- 最优提议：**若已知  $f$ ，理论上  $q^* \propto |f|p$  使方差最小；实际常用  $q \approx p$  或自适应调整  $q$ 。
- 未归一化情形：**用未归一化权重  $\tilde{w} = \tilde{p}/\tilde{q}$ ，再归一化得到  $w$ ；自归一化估计有偏但一致。
- 似然加权 (DGM)：**观测节点不采样、直接固定；权重=所有证据节点的条件概率之积；避免逻辑采样的低效。
- SIR：**先用 IS 得加权样本，再按权重重采样得到无权样本，常用于低维后验抽样。
- 风险点：**若  $q$  与  $p$ （或  $|f|p$ ）差很大、尾部不够厚  $\Rightarrow$  权重方差巨大，效果差；在高维尤其要小心，必要时用自适应或转向粒子滤波/MCMC 等方法。

这一节主要在讲什么？我不太懂

已思考 19s

不急~这节 (23.4) 讲的是**重要性采样 (Importance Sampling, IS)**：当直接从目标分布  $p(x)$  抽样不高效时，改从**更好抽、且把样本集中到“重要区域”的提议分布  $q(x)$  抽样，再用权重把偏差补回来**，从而近似积分/期望  $I = \int f(x)p(x)dx$ 。

## 一句话版

从  $q$  抽样、用权重  $w = p/q$  修正  $\Rightarrow \hat{I} = \frac{1}{S} \sum w_s f(x^{(s)})$ 。选好  $q$ （让它在  $|f(x)|p(x)$  大的地方给更多样本）就高效。

## 核心思路 (超直白)

- 目标：算  $I = \mathbb{E}_p[f(x)]$  (比如稀有事件的概率、某个期望)。
- 问题：直接从  $p$  抽样，很多样本落在“没用”的地方 ( $|f|$  小)，浪费。
- 招数：从容易抽的  $q$  抽样，把“有用区域”抽得更密，然后乘权重

$$w(x) = \frac{p(x)}{q(x)}$$

来纠偏，**每个样本都用**（不丢样本）。

- 理想提议：若知道要算的  $f$ ，最优是

$$q^*(x) \propto |f(x)| p(x)$$

能把估计方差压到最低；实际常用 " $q \approx p$ " 或**自适应地调  $q$** 。

## 常见扩展

- **未归一化分布**：只知道形状  $\tilde{p} \propto p$ 、 $\tilde{q} \propto q$  也行。先算未归一化权重  $\tilde{w} = \tilde{p}/\tilde{q}$ ，再归一化

$$w_s = \frac{\tilde{w}_s}{\sum_r \tilde{w}_r}, \quad \hat{I} = \sum_s w_s f(x^{(s)})$$

这种“自归一化”估计量有限样本**有偏**，但样本多时**一致**。

- **有向图模型里的“似然加权”**：观测节点不采样，直接设为观测值；权重取所有证据节点的条件概率乘积。比“逻辑采样”（冲突就整条样本丢弃）高效得多。
- **SIR（采样-重要性-再采样）**：先用 IS 得到**加权样本**，再按权重**有放回重采样**，把它们变成**无权重样本**，近似等价于直接从  $p$  抽来的样本。

## 什么时候有效 / 会踩坑

- **有效**： $q$  在  $|f|p$  大的区域覆盖充分、尾部不比  $p$  更“轻”（否则权重爆炸）。
- **会踩坑**： $q$  与  $p$ （或  $|f|p$ ）差很大  $\Rightarrow$  权重方差巨大  $\Rightarrow$  需要海量样本；维度高时尤其明显，通常要用**自适应 IS、粒子滤波或 MCMC** 等。

## 要点小结

- 目的：高效估计  $I = \int fp$ ，把样本投向“重要区域”。
- 配方：从  $q$  抽样， $\hat{I} = \frac{1}{S} \sum(p/q)f$ 。
- 最优提议： $q^* \propto |f|p$ ；实际用  $q \approx p$  或自适应。
- 未归一化：用自归一化权重，偏但一致。
- 应用：似然加权 (DGM)、SIR 把加权样本变成无权样本。
- 风险：提议不当  $\Rightarrow$  权重大起大落、方差大；高维更严重。

翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

## 23.5 Particle filtering

Particle filtering (PF) is a Monte Carlo, or simulation based, algorithm for recursive Bayesian inference. That is, it approximates the predict-update cycle described in Section 18.3.1. It is very widely used in many areas, including tracking, time-series forecasting, online parameter learning, etc. We explain the basic algorithm below. For a book-length treatment, see (Doucet et al. 2001); for a good tutorial, see (Arulampalam et al. 2002), or just read on.

### 23.5.1 Sequential importance sampling

The basic idea is to approximate the belief state (of the entire state trajectory) using a weighted set of particles:

$$p(z_{1:t}|y_{1:t}) \approx \sum_{s=1}^t w_s(t) \delta_{z_s}^{(t)}(z_{1:t}) \quad (23.36)$$

where  $w_s(t)$  is the normalized weight of sample  $s$  at time  $t$ . From this representation, we can easily compute the marginal distribution over the most recent state,  $p(z_t|y_{1:t})$ , by simply ignoring the previous parts of the trajectory,  $z_{1:t-1}$ . (The fact that PF samples in the space of entire trajectories has various implications which we will discuss later.)

We update this belief state using importance sampling. If the proposal has the form  $q(z_s|z_{1:t-1}, y_{1:t})$ , then the importance weights are given by

$$w_s(t) \propto p(z_s|z_{1:t-1}, y_{1:t}) q(z_s|z_{1:t-1}, y_{1:t}) \quad (23.37)$$

which can be normalized as follows:

$$\hat{w}_s(t) = w_s(t) / \sum_s w_s(t) \quad (23.38)$$

We can rewrite the numerator recursively as follows:

$$\begin{aligned} p(z_{1:t}|y_{1:t}) &= p(y_t|z_{1:t}, y_{1:t-1}) p(z_{1:t}|y_{1:t-1}) \\ &\quad (23.39) \end{aligned}$$

$$\begin{aligned} &= p(y_t|z_t) p(z_t|z_{1:t-1}, y_{1:t-1}) p(z_{1:t-1}|y_{1:t-1}) \\ &\quad (23.40) \end{aligned}$$

$$\propto p(y_t|z_t) p(z_t|z_{1:t-1}) p(z_{1:t-1}|y_{1:t-1}) \quad (23.41)$$

where we have made the usual Markov assumptions. We will restrict attention to proposal densities of the following form:

$$q(z_{1:t}|y_{1:t}) = q(z_t|z_{1:t-1}, y_{1:t}) q(z_{1:t-1}|y_{1:t-1}) \quad (23.42)$$

so that we can “grow” the trajectory by adding the new state  $z_t$  to the end. In this case, the importance weights

simplify to

$$\begin{aligned} ws_t &\propto p(yt|zs_t) p(zs_t|zs_{t-1}) p(zs_{1:t-1}|y_{1:t-1}) q(zs_t|zs_{1:t-1}, y_{1:t}) \\ &= ws_{t-1} p(yt|zs_t) p(zs_t|zs_{t-1}) q(zs_t|zs_{1:t-1}, y_{1:t}) \end{aligned} \quad (23.43)$$

If we further assume that  $q(zt|z_{1:t-1}, y_{1:t}) = q(zt|zt-1, yt)$ , then we only need to keep the most recent part of the trajectory and observation sequence, rather than the whole history, in order to compute the new sample. In this case, the weight becomes

$$ws_t \propto ws_{t-1} p(yt|zs_t) p(zs_t|zs_{t-1}) q(zs_t|zs_{t-1}, yt) \quad (23.45)$$

Hence we can approximate the posterior filtered density using

$$p(zt|y_{1:t}) \approx S \sum_{s=1}^S w_s \delta z_{s,t}(zt) \quad (23.46)$$

As  $S \rightarrow \infty$ , one can show that this approaches the true posterior (Crisan et al. 1999).

The basic algorithm is now very simple: for each old sample  $s$ , propose an extension using  $zs_t \sim q(zt|zs_{t-1}, yt)$ , and give this new particle weight  $ws_t$  using Equation 23.45. Unfortunately, this basic algorithm does not work very well, as we discuss below.

### 23.5.2 The degeneracy problem

The basic sequential importance sampling algorithm fails after a few steps because most of the particles will have negligible weight. This is called the degeneracy problem, and occurs because we are sampling in a high-dimensional space (in fact, the space is growing in size over time), using a myopic proposal distribution.

We can quantify the degree of degeneracy using the effective sample size, defined by

$$Seff = S / \text{var}[w_s] \quad (23.47)$$

where  $w_s = p(zs_t|y_{1:t})/q(zs_t|zs_{t-1}, yt)$  is the “true weight” of particle  $s$ . This quantity cannot be computed exactly, since we don’t know the true posterior, but we can approximate it using

$$\hat{S}_{eff} = \frac{1}{S} \sum_{s=1}^S (ws)^2 \quad (23.48)$$

If the variance of the weights is large, then we are wasting our resources updating particles with low weight, which do not contribute much to our posterior estimate.

There are two main solutions to the degeneracy problem: adding a resampling step, and using a good proposal distribution. We discuss both of these in turn.

### 23.5.3 The resampling step

The main improvement to the basic SIS algorithm is to

monitor the effective sampling size, and whenever it drops below a threshold, to eliminate particles with low weight, and then to create replicates of the surviving particles. (Hence PF is sometimes called survival of the fittest (Kanazawa et al. 1995).) In particular, we generate a new set  $\{z_s^* | t\}_{S=1}^S$  by sampling with replacement  $S$  times from the weighted distribution

$$p(z_t^* | y_{1:t}) \approx S \sum_{s=1}^S w_s^t \delta_{z_s^*}(z_t) \quad (23.49)$$

where the probability of choosing particle  $j$  for replication is  $w_j^t$ . (This is sometimes called rejuvenation.) The result is an iid unweighted sample from the discrete density Equation 23.49, so we set the new weights to  $w_s^t = 1/S$ .

This scheme is illustrated in Figure 23.4.

There are a variety of algorithms for performing the resampling step. The simplest is multinomial resampling, which computes

$$(K_1, \dots, K_S) \sim \text{Mu}(S, (w_1^t, \dots, w_S^t)) \quad (23.50)$$

We then make  $K_s$  copies of  $z_s^* | t$ . Various improvements exist, such as systematic resampling residual resampling, and stratified sampling, which can reduce the variance of the weights. All these methods take  $O(S)$  time. See (Doucet et al. 2001) for details.

The overall particle filtering algorithm is summarized in Algorithm 6. (Note that if an estimate of the state is required, it should be computed before the resampling step, since this will result in lower variance.)

Although the resampling step helps with the degeneracy problem, it introduces problems of its own. In particular, since the particles with high weight will be selected many times, there is a loss of diversity amongst the population. This is known as sample impoverishment. In the extreme case of no process noise (e.g., if we have static but unknown parameters as part of the state space), then all the particles will collapse to a single point within a few iterations.

To mitigate this problem, several solutions have been proposed. (1) Only resample when necessary, not at every time step. (The original bootstrap filter (Gordon 1993) resampled at every step, but this is suboptimal.) (2) After replicating old particles, sample new values using an MCMC step which leaves the posterior distribution invariant (see e.g., the resample-move algorithm in (Gilks and Berzuini 2001)). (3) Create a kernel density estimate on top of the particles,

$$p(z_t^* | y_{1:t}) \approx S \sum_{s=1}^S w_s^t \kappa(z_t - z_s^*) \quad (23.51)$$

where  $\kappa$  is some smoothing kernel. We then sample from this smoothed distribution. This is known as a regularized particle filter (Musso et al. 2001). (4) When performing inference on static parameters, add some artificial process noise. (If this is undesirable, other algorithms must be used for online parameter estimation, e.g., (Andrieu et al. 2005)).

#### 23.5.4 The proposal distribution

The simplest and most widely used proposal distribution is to sample from the prior:

$$q(z_t|z_{s,t-1}, y_t) = p(z_t|z_{s,t-1}) \quad (23.52)$$

In this case, the weight update simplifies to

$$w_s t \propto w_{s,t-1} p(y_t|z_{s,t}) \quad (23.53)$$

This can be thought of a “generate and test” approach: we sample values from the dynamic model, and then evaluate how good they are after we see the data (see Figure 23.4). This is the approach used in the condensation algorithm (which stands for “conditional density propagation”) used for visual tracking (Isard and Blake 1998). However, if the likelihood is narrower than the dynamical prior (meaning the sensor is more informative than the motion model, which is often the case), this is a very inefficient approach, since most particles will be assigned very low weight.

It is much better to actually look at the data  $y_t$  when generating a proposal. In fact, the optimal proposal distribution has the following form:

$$q(z_t|z_{s,t-1}, y_t) = p(z_t|z_{s,t-1}, y_t) = p(y_t|z_t) p(z_t|z_{s,t-1}) p(y_t|z_{s,t-1}) \quad (23.54)$$

If we use this proposal, the new weight is given by

$$w_s t \propto w_{s,t-1} p(y_t|z_{s,t-1}) = w_{s,t-1} p(y_t|z_t) p(z_t|z_{s,t-1}) dz_t \quad (23.55)$$

This proposal is optimal since, for any given  $z_{s,t-1}$ , the new weight  $w_s t$  takes the same value regardless of the value drawn for  $z_t$ . Hence, conditional on the old values  $z_{s,t-1}$ , the variance of true weights  $\text{var}[w_s t]$  is zero.

In general, it is intractable to sample from  $p(z_t|z_{s,t-1}, y_t)$  and to evaluate the integral needed to compute the predictive density  $p(y_t|z_{s,t-1})$ . However, there are two cases when the optimal proposal distribution can be used. The first setting is when  $z_t$  is discrete, so the integral becomes a sum. Of course, if the entire state space is discrete, we can use an HMM filter instead, but in some cases, some parts of the state are discrete, and some continuous. The second setting is when  $p(z_t|z_{s,t-1}, y_t)$  is Gaussian. This occurs when the dynamics are nonlinear

but the observations are linear. See Exercise 23.3 for the details.

In cases where the model is not linear-Gaussian, we may still compute a Gaussian approximation to  $p(z_t|z_{t-1}, y_t)$  using the unscented transform (Section 18.5.2) and use this as a proposal. This is known as the unscented particle filter (van der Merwe et al. 2000). In more general settings, we can use other kinds of data-driven proposals, perhaps based on discriminative models. Unlike MCMC, we do not need to worry about the proposals being reversible.

### 23.5.5 Application: robot localization

Consider a mobile robot wandering around an office environment. We will assume that it already has a map of the world, represented in the form of an occupancy grid, which just specifies whether each grid cell is empty space or occupied by something solid like a wall. The goal is for the robot to estimate its location. This can be solved optimally using an HMM filter, since we are assuming the state space is discrete. However, since the number of states,  $K$ , is often very large, the  $O(K^2)$  time complexity per update is prohibitive. We can use a particle filter as a sparse approximation to the belief state. This is known as Monte Carlo localization, and is described in detail in (Thrun et al. 2006).

Figure 23.5 gives an example of the method in action. The robot uses a sonar range finder, so it can only sense distance to obstacles. It starts out with a uniform prior, reflecting the fact that the owner of the robot may have turned it on in an arbitrary location. (Figuring out where you are, starting from a uniform prior, is called global localization.) After the first scan, which indicates two walls on either side, the belief state is shown in (b). The posterior is still fairly broad, since the robot could be in any location where the walls are fairly close by, such as a corridor or any of the narrow rooms. After moving to location 2, the robot is pretty sure it must be in the corridor, as shown in (c). After moving to location 3, the sensor is able to detect the end of the corridor. However, due to symmetry, it is not sure if it is in location I (the true location) or location II. (This is an example of perceptual aliasing, which refers to the fact that different things may look the same.) After moving to locations 4 and 5, it is finally able to figure out precisely where it is. The whole process is analogous to someone getting lost in an office building, and wandering the corridors until they see a sign

they recognize.

In Section 23.6.3, we discuss how to estimate location and the map at the same time.

### 23.5.6 Application: visual object tracking

Our next example is concerned with tracking an object (in this case, a remote-controlled helicopter) in a video sequence. The method uses a simple linear motion model for the centroid of the object, and a color histogram for the likelihood model, using Bhattacharya distance to compare histograms. The proposal distribution is obtained by sampling from the likelihood. See (Nummiaro et al. 2003) for further details.

Figure 23.6 shows some example frames. The system uses  $S = 250$  particles, with an effective sample size of  $\hat{S}_{eff} = 134$ . (a) shows the belief state at frame 1. The system has had to resample 5 times to keep the effective sample size above the threshold of 150; (b) shows the belief state at frame 251; the red lines show the estimated location of the center of the object over the last 250 frames. (c) shows that the system can handle visual clutter, as long as it does not have the same color as the target object. (d) shows that the system is confused between the grey of the helicopter and the grey of the building. The posterior is bimodal. The green ellipse, representing the posterior mean and covariance, is in between the two modes. (e) shows that the probability mass has shifted to the wrong mode: the system has lost track. (f) shows the particles spread out over the gray building; recovery of the object is very unlikely from this state using this proposal.

We see that the method is able to keep track for a fairly long time, despite the presence of clutter. However, eventually it loses track of the object. Note that since the algorithm is stochastic, simply re-running the demo may fix the problem. But in the real world, this is not an option. The simplest way to improve performance is to use more particles. An alternative is to perform tracking by detection, by running an object detector over the image every few frames. See (Forsyth and Ponce 2002; Szeliski 2010; Prince 2012) for details.

### 23.5.7 Application: time series forecasting

In Section 18.2.4, we discussed how to use the Kalman filter to perform time series forecasting. This assumes that the model is a linear-Gaussian state-space model. There are many models which are either non-linear and/or non-Gaussian. For example, stochastic volatility models, which

are widely used in finance, assume that the variance of the system and/or observation noise changes over time. Particle filtering is widely used in such settings. See e.g., (Doucet et al. 2001) and references therein for details.

已思考 22s

下面先给出**不遗漏的中文翻译**；接着用**通俗、详细**的方式解释；最后给出**要点总结**。

(注：翻译严格按照原文内容与公式编号；个别符号在“解释”部分会再做直白说明。)

---

## 原文翻译

### 23.5 粒子滤波 (Particle filtering)

粒子滤波 (PF) 是一种基于蒙特卡罗 (或基于仿真) 的递归贝叶斯推断算法。也就是说，它近似实现了第 18.3.1 节所描述的预测–更新循环。它在许多领域被广泛使用，包括跟踪、时间序列预测、在线参数学习等。我们在下面解释基本算法。关于书本长度的论述，见 (Doucet et al. 2001)；关于一篇不错的教程，见 (Arulampalam et al. 2002)，当然你也可以继续往下读。

#### 23.5.1 顺序重要性采样 (Sequential importance sampling)

基本思想是使用一组带权粒子来近似**信念状态**（整个状态轨迹的信念）：

$$p(z_{1:t} \mid y_{1:t}) \approx \sum_{s=1}^S \hat{w}_t^s \delta_{z_{1:t}^s}(z_{1:t}) \quad (23.36)$$

其中  $\hat{w}_t^s$  是在时刻  $t$  第  $s$  个样本的**归一化权重**。由这一表示，我们可以很容易地得到**最近时刻状态**的边缘分布  $p(z_t \mid y_{1:t})$ ，方法是直接忽略轨迹的先前部分  $z_{1:t-1}$ 。（PF 在整个轨迹空间中采样这一事实会有若干影响，我们稍后讨论。）

我们使用**重要性采样**来更新这一信念状态。若提议分布具有形式  $q(z_{1:t}^s \mid y_{1:t})$ ，则重要性权重为

$$w_t^s \propto \frac{p(z_{1:t}^s \mid y_{1:t})}{q(z_{1:t}^s \mid y_{1:t})} \quad (23.37)$$

并可按如下方式归一化：

$$\hat{w}_t^s = \frac{w_t^s}{\sum_s w_t^s} \quad (23.38)$$

我们可以递归地改写分子如下：

$$p(z_{1:t} \mid y_{1:t}) = \frac{p(y_t \mid z_{1:t}, y_{1:t-1}) p(z_{1:t} \mid y_{1:t-1})}{p(y_t \mid y_{1:t-1})} \quad (23.39)$$

$$= \frac{p(y_t | z_t) p(z_t | z_{1:t-1}, y_{1:t-1}) p(z_{1:t-1} | y_{1:t-1})}{p(y_t | y_{1:t-1})} \quad (23.40)$$

$$\propto p(y_t | z_t) p(z_t | z_{t-1}) p(z_{1:t-1} | y_{1:t-1}) \quad (23.41)$$

其中我们使用了通常的马尔可夫假设。我们将仅关注如下形式的提议密度：

$$q(z_{1:t} | y_{1:t}) = q(z_t | z_{1:t-1}, y_{1:t}) q(z_{1:t-1} | y_{1:t-1}) \quad (23.42)$$

这样我们就能通过把新状态  $z_t$  接在末尾来“延展”轨迹。在此情形下，重要性权重化简为

$$w_t^s \propto \frac{p(y_t | z_t^s) p(z_t^s | z_{t-1}^s) p(z_{1:t-1}^s | y_{1:t-1})}{q(z_t^s | z_{1:t-1}^s, y_{1:t}) q(z_{1:t-1}^s | y_{1:t-1})} \quad (23.43)$$

$$= w_{t-1}^s \frac{p(y_t | z_t^s) p(z_t^s | z_{t-1}^s)}{q(z_t^s | z_{1:t-1}^s, y_{1:t})} \quad (23.44)$$

如果我们进一步假设  $q(z_t | z_{1:t-1}, y_{1:t}) = q(z_t | z_{t-1}, y_t)$ ，那么为了计算新样本，我们只需要轨迹和观测序列的最近部分，而不是整个历史。在此情形下，权重变为

$$w_t^s \propto w_{t-1}^s \frac{p(y_t | z_t^s) p(z_t^s | z_{t-1}^s)}{q(z_t^s | z_{t-1}^s, y_t)} \quad (23.45)$$

因此我们可以用下式近似滤后后验密度：

$$p(z_t | y_{1:t}) \approx \sum_{s=1}^S \hat{w}_t^s \delta_{z_t^s}(z_t) \quad (23.46)$$

当  $S \rightarrow \infty$  时，可以证明这将收敛到真实后验 (Crisan et al. 1999)。

现在基本算法非常简单：对每个旧样本  $s$ ，根据  $z_t^s \sim q(z_t | z_{t-1}^s, y_t)$  提议一个延展，并使用式 (23.45) 给这个新粒子赋权。遗憾的是，这个基本算法效果并不好，我们在下文讨论。

### 23.5.2 退化 (degeneracy) 问题

基本的顺序重要性采样 (SIS) 算法在若干步之后会失败，因为大多数粒子的权重会变得可以忽略。这被称为退化问题，其发生的原因是我们在一个高维空间中采样（事实上，空间的维度还会随着时间增长），却使用了一个目光短浅的提议分布。

我们可以使用有效样本量来量化退化程度，定义为

$$S_{\text{eff}} = \frac{S}{1 + \text{var}[w_t^{*s}]} \quad (23.47)$$

其中  $w_t^{*s} = p(z_t^s | y_{1:t}) / q(z_t^s | z_{t-1}^s, y_t)$  是粒子  $s$  的“真实权重”。由于我们不知道真实后验，这个量无法精确计算，但可以用

$$\hat{S}_{\text{eff}} = \frac{1}{\sum_{s=1}^S (\hat{w}_t^s)^2} \quad (23.48)$$

进行近似。如果权重的方差很大，那么我们就在浪费资源去更新那些**低权重**的粒子，而它们对我们的后验估计贡献很小。

解决退化问题有两条主要途径：加入**重采样 (resampling) 步骤，以及使用更好的提议分布**。我们依次讨论。

### 23.5.3 重采样步骤

对基本 SIS 算法的主要改进是：**监控有效样本量**，当它降到某个阈值以下时，**消除低权重粒子，并对幸存粒子复制**（因此 PF 有时也被称为“适者生存”（Kanazawa et al. 1995））。具体地，我们通过从加权分布

$$p(z_t | y_{1:t}) \approx \sum_{s=1}^S \hat{w}_t^s \delta_{z_t^s}(z_t) \quad (23.49)$$

**中有放回地抽样**  $S$  次来生成一个新集合  $\{z_t^{s*}\}_{s=1}^S$ ，其中复制粒子  $j$  的概率是  $w_t^j$ 。（这有时被称为**rejuvenation (复壮)**。）结果是来自离散密度 (23.49) 的**独立同分布、无权重样本**，因此我们将新权重设为  $w_t^s = 1/S$ 。该方案如图 23.4 所示。

执行重采样有多种算法。最简单的是**多项式 (multinomial) 重采样**，其计算

$$(K_1, \dots, K_S) \sim \text{Mu}(S, (w_t^1, \dots, w_t^S)) \quad (23.50)$$

然后将  $z_t^s$  复制  $K_s$  次。存在若干改进方法，例如**系统 (systematic) 重采样、剩余 (residual) 重采样**以及**分层 (stratified) 采样**，它们可以降低权重方差。这些方法的时间复杂度均为  $O(S)$ 。细节见 (Doucet et al. 2001)。

总体的粒子滤波算法在**算法 6**中总结。（注意：如果需要状态的估计，应当在重采样步骤之前计算，因为这样会得到更低的方差。）

尽管重采样步骤有助于退化问题，但它也带来自身的问题。特别地，由于高权重粒子会被多次选中，**群体多样性**会下降。这被称为**样本贫化 (sample impoverishment)**。在没有过程噪声的极端情况下（例如，如果我们把**静态但未知的参数**当作状态的一部分），那么所有粒子会在若干次迭代内**塌缩到同一点**。

为缓解这一问题，人们提出了若干方案。(1) 只在**必要时**重采样，而不是每个时间步都重采样。（最初的**自助滤波器 (bootstrap filter)**（Gordon 1993）每步都重采样，但这并非最优。）(2) 在复制旧粒子后，使用一个使后验分布保持不变的**MCMC 步**来采新值（例如 (Gilks and Berzuini 2001) 的**resample-move 算法**）。(3) 在粒子之上创建一个**核密度估计**：

$$p(z_t | y_{1:t}) \approx \sum_{s=1}^S w_t^s \kappa(z_t - z_t^s) \quad (23.51)$$

其中  $\kappa$  是某个平滑核。然后从这个**平滑后的分布采样**。这称为**正则化粒子滤波器** (Musso et al. 2001)。(4) 当对**静态参数**进行推断时，加入一些**人工的过程噪声**。(如果这不可取，那么在线参数估计要使用其他算法，例如 (Andrieu et al. 2005)。)

### 23.5.4 提议分布

最简单且最常用的提议分布是从**先验采样**：

$$q(z_t | z_{t-1}^s, y_t) = p(z_t | z_{t-1}^s) \quad (23.52)$$

在此情况下，权重更新化简为

$$w_t^s \propto w_{t-1}^s p(y_t | z_t^s) \quad (23.53)$$

这可以被看作“**生成-检验**”的方法：我们从动态模型中采值，然后在看到数据之后评估它们的好坏（见图 23.4）。这一方法被用于视觉跟踪中的**condensation 算法**（“条件密度传播”）(Isard and Blake 1998)。然而，如果**似然比动力学先验更窄**（意味着传感器比运动模型更有信息，这种情况很常见），这种方法就**非常低效**，因为大多数粒子会被赋予很低的权重。

更好的做法是在生成提议时**实际利用数据**  $y_t$ 。事实上，**最优提议分布**具有如下形式：

$$q(z_t | z_{t-1}^s, y_t) = p(z_t | z_{t-1}^s, y_t) = \frac{p(y_t | z_t) p(z_t | z_{t-1}^s)}{p(y_t | z_{t-1}^s)} \quad (23.54)$$

若使用该提议，新的权重为

$$w_t^s \propto w_{t-1}^s p(y_t | z_{t-1}^s) = w_{t-1}^s \int p(y_t | z_t) p(z_t | z_{t-1}^s) dz_t \quad (23.55)$$

该提议是**最优的**，因为对给定的  $z_{t-1}^s$ ，新的权重  $w_t^s$  的取值与**所抽取的  $z_t^s$  无关**。因此，在条件于旧值  $z_{t-1}$  的情况下，**真实权重的方差**  $\text{var}[w_t^{*s}]$  为零。

一般而言，从  $p(z_t | z_{t-1}^s, y_t)$  采样以及计算用于求**预测密度**  $p(y_t | z_{t-1}^s)$  的积分都是**不可 tractable** 的。不过，有两种情形可以使用最优提议。其一，当  $z_t$  是**离散的**，此时积分变为求和。当然，如果整个状态空间都是离散的，我们可以改用 HMM 滤波器，但在某些情况下，状态的一部分是离散的，另一部分是连续的。其二，当  $p(z_t | z_{t-1}^s, y_t)$  是**高斯**时，这发生在**动力学非线性但观测是线性**的时候。细节见习题 23.3。

在模型不是**线性-高斯**的情况下，我们仍可用**无迹变换**（第 18.5.2 节）来对  $p(z_t | z_{t-1}^s, y_t)$  做高斯近似，并将其用作提议。这称为**无迹粒子滤波器** (unscented PF) (van der Merwe et al. 2000)。在更一般的设置中，我们可以使用其他种类的**数据驱动提议**，比如基于判别模型的提议。与 MCMC 不同，我们**不需要担心提议的可逆性**。

### 23.5.5 应用：机器人定位

考虑一个在办公室环境中移动的机器人。我们假设它已经有一张世界地图，表示为**占据栅**

**格**（仅指示每个栅格单元是空的还是被墙等固体占据）。目标是让机器人估计自己的位置。这在理论上可以用 HMM 滤波最优化地解决，因为我们假设状态空间是离散的。然而，由于状态数  $K$  往往非常大，每次更新  $O(K^2)$  的时间复杂度令人望而却步。我们可以用粒子滤波作为信念状态的稀疏近似。这称为**蒙特卡罗定位**，详见 (Thrun et al. 2006)。

图 23.5 给出了该方法的一个示例。机器人使用**声纳测距**，因此它只能感知到与障碍物的距离。它从**均匀先验**开始，反映了机器人的主人可能在任意位置把它打开。（从均匀先验出发去判断自己在哪被称为**全局定位**。）第一次扫描后，显示两侧有墙，(b) 展示了此时的信念状态。后验仍然相当宽，因为机器人可能处在任何墙比较近的地方，比如走廊或狭窄的房间。移动到位置 2 后，如 (c) 所示，机器人很确定自己应当在走廊上。移动到位置 3 后，传感器能够探测到走廊尽头。然而由于**对称性**，它还不确定自己在位置 I (真实位置) 还是位置 II。（这是**感知混叠**的一个例子，意思是不同的事物看起来可能一样。）移动到位置 4 和 5 后，它最终能够精确地确定自己位置。整个过程类似一个人在办公楼里迷路，沿着走廊徘徊，直到看到一块熟悉的标识。

在第 23.6.3 节，我们将讨论如何同时估计位置和地图。

### 23.5.6 应用：视频目标跟踪

下一个例子关注在视频序列中跟踪一个目标（这里是一个遥控直升机）。该方法对目标质心使用**简单线性运动模型**，并对似然模型使用**颜色直方图**，用**Bhattacharya 距离**比较直方图。提议分布通过从**似然**中采样获得。更多细节见 (Nummiaro et al. 2003)。

图 23.6 展示了一些示例帧。系统使用  $S = 250$  个粒子，有效样本量  $\hat{S}_{\text{eff}} = 134$ 。(a) 展示第 1 帧时的信念状态。为使有效样本量保持在阈值 150 以上，系统不得不重采样 5 次；(b) 展示第 251 帧时的信念状态；红线表示过去 250 帧中目标中心位置的估计。(c) 展示系统可以处理视觉杂乱，只要其颜色与目标不同。(d) 展示系统在直升机的灰色与建筑物的灰色之间产生混淆。后验是**双峰的**。表示后验均值与协方差的绿色椭圆位于两个峰之间。(e) 展示概率质量已转移到错误的峰：系统**丢失了目标**。(f) 展示粒子分布在灰色建筑上；在使用该提议的情况下，从此状态恢复对象的可能性非常小。

我们看到该方法能够在存在杂乱的情况下**相当长时间地**保持跟踪。然而，最终它丢失了目标。注意，由于算法是**随机的**，简单地重新运行演示也许能修复问题。但在现实世界里，这不是一个选项。最简单的改进方式是**使用更多粒子**。另一种方式是通过**检测来进行跟踪**，即每隔几帧在图像上运行一个**目标检测器**。详见 (Forsyth and Ponce 2002; Szeliski 2010; Prince 2012)。

### 23.5.7 应用：时间序列预测

在第 18.2.4 节中，我们讨论了如何使用**卡尔曼滤波器**进行时间序列预测。那是假设模型是**线性-高斯状态空间模型**。有许多模型是**非线性和/或非高斯的**。例如，**随机波动率模型**（在金融中广泛使用）假设系统和/或观测噪声的方差随时间变化。粒子滤波在这类设置中被广泛使用。详见 (Doucet et al. 2001) 及其中参考文献。

## 通俗解释（循序渐进，图像化理解）

### 1) PF 在做什么？

- 目标：**在线（递归）地估计“状态”随时间的变化：每来一条观测  $y_t$ ，就把对  $z_t$  的信念从“预测”更新为“后验”。
- 做法：**用很多“粒子”  $\{z_t^{(s)}\}$  加上权重  $\{\hat{w}_t^{(s)}\}$  来表示分布。可以把它想成：用一群带分数的小球，整体形状近似目标分布。

### 2) 顺序重要性采样 (SIS)

- 扩展轨迹：**从上一步的粒子  $z_{t-1}^{(s)}$  出发，用提议  $q(z_t \mid z_{t-1}^{(s)}, y_t)$  采一个新位置  $z_t^{(s)}$ 。
- 更新权重：**用

$$w_t^{(s)} \propto w_{t-1}^{(s)} \frac{p(y_t \mid z_t^{(s)}) p(z_t^{(s)} \mid z_{t-1}^{(s)})}{q(z_t^{(s)} \mid z_{t-1}^{(s)}, y_t)}$$

然后归一化得到  $\hat{w}_t^{(s)}$ 。

- 滤后分布：** $\sum_s \hat{w}_t^{(s)} \delta_{z_t^{(s)}}$  近似  $p(z_t \mid y_{1:t})$ 。

直觉：先按“模型 + 数据”生成候选粒子，再按“它解释数据的好坏（似然）/提议的概率”给它打分。

### 3) 退化问题与有效样本量

- 过几步后，**多数权重→几乎为零**，只有极少数粒子“占大头”。这就是**退化**。
- 用

$$\hat{S}_{\text{eff}} = \frac{1}{\sum_s (\hat{w}_t^{(s)})^2}$$

来“体感”还剩下多少“有效”粒子。越小越糟。

### 4) 重采样：适者生存 + 复壮

- 当  $\hat{S}_{\text{eff}}$  太低就**重采样**：
  - 按权重概率抽取并复制粒子（有放回），丢掉低权重粒子；
  - 把新权重设为均匀  $1/S$ 。
- 优点：**把计算力集中到“可能的区域”。
- 副作用：**样本贫化（多样性下降，可能塌缩）。
- 缓解招：**
  - 只在必要时重采样（设阈值）；
  - 重采样后加一小步 MCMC 抖动（resample-move）；

3. 用**核平滑正则化** (regularized PF) ;
4. 静态参数时加少量**人工过程噪声**或用专门的在线参数法。

## 5) 提议分布：从“先验”到“利用数据”

- **最朴素**:  $q = p(z_t | z_{t-1})$  (先验/运动模型)。更新式简化为  $w_t^{(s)} \propto w_{t-1}^{(s)} p(y_t | z_t^{(s)})$ 
  - **问题**: 若**传感器很准** (似然窄)，大部分先验提议会偏离数据  $\Rightarrow$  权重几乎为零  $\Rightarrow$  浪费。
- **更好: 最优提议**

$$q^*(z_t | z_{t-1}, y_t) = p(z_t | z_{t-1}, y_t) \propto p(y_t | z_t) p(z_t | z_{t-1})$$

这样，给定旧粒子时，新权重**不依赖**你抽到的  $z_t$  值，权重方差为 0，效果最好。

- **现实**: 多半算不出/采不出；但在**离散**  $z_t$  或 **线性观测 + 高斯**等特例可以做到；更一般时可用**无迹变换** (Unscented PF) 或其他**数据驱动提议**近似之。

## 6) 三个应用图景 (帮助理解)

- **机器人定位** (离散空间很大) : HMM 理论上最优但太慢；PF 用有限粒子做稀疏近似，能从**全局定位**逐步锁定位置，也会遭遇**感知混叠**。
- **视频目标跟踪** (直升机例) : 用颜色直方图做似然，粒子 250 个，也会因为背景相似颜色而**漂到错误峰**，最终丢失目标；多用粒子/加检测器可改进。
- **时间序列预测**: 非线性/非高斯模型 (如**随机波动率**) 超出卡尔曼滤波范围，PF 很常用。

## 要点总结

- **PF = SIS + 重采样**: 在线近似递归贝叶斯 (预测–更新) 过程，用**带权粒子**表示分布；权重来自**重要性采样**，必要时做**重采样**避免退化。
- **核心更新**: 对每个粒子

$$z_t^{(s)} \sim q(z_t | z_{t-1}^{(s)}, y_t), \quad w_t^{(s)} \propto w_{t-1}^{(s)} \frac{p(y_t | z_t^{(s)}) p(z_t^{(s)} | z_{t-1}^{(s)})}{q(z_t^{(s)} | z_{t-1}^{(s)}, y_t)}.$$

- **退化与  $\hat{S}_{\text{eff}}$** : 权重方差大  $\Rightarrow$  有效样本量小  $\Rightarrow$  算力浪费在低权重粒子上。
- **重采样**: 把资源集中到高权重区域，但会**丧失多样性**；用“只在必要时重采样”“resample-move (加 MCMC)”“核平滑”等缓解。
- **提议选择**:
  - 先验提议简单但在“似然窄”时低效；
  - **最优提议**利用数据  $y_t$ ，理论上权重方差最小；

- 现实中用近似（离散枚举/线性-高斯/无迹 PF/数据驱动）。
- 应用：大状态空间定位、视觉跟踪、非线性/非高斯时间序列（如随机波动率）。
- 一条实践箴言：粒子数、重采样策略、提议质量三者决定性能；观测越“尖锐”，越要把数据融入提议，否则容易退化。

这一节主要在讲什么？我不太懂

已思考 12s

不急～这节（23.5）讲的是**粒子滤波（PF）**：一种用“很多随机粒子 + 权重”来**在线**实现贝叶斯“预测→更新”循环的办法，常用于跟踪、时间序列预测、在线学习等。

## 一句话版

**粒子滤波 = 顺序重要性采样（给粒子打分） + 必要时重采样（复制强的、淘汰弱的）**，用来实时近似  $p(z_t | y_{1:t})$ 。

## 直观版（走迷宫的比喻）

- 你迷路（不知道位置  $z_t$ ），手里有一堆“猜测”（**粒子**）和各自的**分数**（权重）。
- 新看到线索  $y_t$ （传感器观测）后：
  1. **扩展/预测**：按提议  $q(z_t | z_{t-1}, y_t)$  给每个粒子生成新位置；
  2. **打分/更新**：看新位置与线索是否吻合（似然  $p(y_t | z_t)$ ），乘上动态模型  $p(z_t | z_{t-1})$ ，再除以提议  $q$  得到新权重；
  3. **归一化权重**；
  4. **必要时重采样**：若“有效样本数”  $\hat{S}_{\text{eff}} = 1 / \sum w^2$  太小，就按权重抽样复制强粒子、丢弱粒子（权重重置为均匀）。

## 关键问题与对策

- **退化**：过几步后，大多数粒子权重几乎为零  $\Rightarrow$  浪费算力。  
**对策**：重采样；但会**样本贫化**（多样性下降）。缓解招：
  - 只在需要时重采样（设阈值）；
  - 重采样后加一点“抖动”/MCMC（resample-move）；
  - 用核平滑（regularized PF）；
  - 静态参数可加微小“过程噪声”或用专门的在线参数法。
- **提议分布怎么选？**
  - **先验提议**： $q(z_t | z_{t-1})$  简单，但当**传感器很准**（似然很窄）时效率差。

- **最优提议:**  $q^*(z_t | z_{t-1}, y_t) = p(z_t | z_{t-1}, y_t)$ , 理论上权重方差最小; 在离散或“线性观测+高斯”情形可做, 其他情况可用无迹变换或数据驱动近似。

## 你能把它用在什么场景?

- **机器人定位** (大离散状态空间, HMM 代价太高)
- **视频目标跟踪** (颜色直方图似然, 粒子跟着目标走)
- **非线性/非高斯时间序列** (如金融里的随机波动率)

## 要点小结

- PF 用**带权粒子**在线逼近后验; 核心循环: 提议→加权→归一化→(必要时)重采样。
- **有效样本数**衡量退化; 重采样能救场但会丢多样性, 需要配合抖动/核平滑/按需触发。
- **提议越利用数据越好:** 先验提议易实现但低效; 最优提议最好但常近似实现。
- 适合**跟踪/预测/在线学习**等需要递推更新的不规则模型 (非线性、非高斯)。

翻译并解释以下内容, 翻译不能有遗漏, 解释请通俗易懂且详尽, 因为我的数学基础不太好, 最后总结要点。

### 23.6 Rao-Blackwellised particle filtering (RBPF)

In some models, we can partition the hidden variables into two kinds,  $q_t$  and  $z_t$ , such that we can analytically integrate out  $z_t$  provided we know the values of  $q_{1:t}$ . This means we only have sample  $q_{1:t}$ , and can represent  $p(z_t|q_{1:t})$  parametrically. Thus each particle  $s$  represents a value for  $q_s$  and a distribution of the form  $p(z_t|y_{1:t}, q_s)$ . These hybrid particles are sometimes called distributional particles or collapsed particles (Koller and Friedman 2009, Sec 12.4).

The advantage of this approach is that we reduce the dimensionality of the space in which we are sampling, which reduces the variance of our estimate. Hence this technique is known as Rao-Blackwellised particle filtering or RBPF for short, named after Theorem 24.20. The method is best explained using a specific example.

#### 23.6.1 RBPF for switching LG-SSMs

A canonical example for which RBPF can be applied is the switching linear dynamical system (SLDS) model discussed in Section 18.6 (Chen and Liu 2000; Doucet et al. 2001).

We can represent  $p(z_t|y_{1:t}, q_s)$  using a mean and

covariance matrix for each particle  $s$ , where  $q_t \in \{1, \dots, K\}$ .

If we propose from the prior,  $p(q_t = k | q_{s:t-1}, y_{1:t-1})$ , the weight update becomes

$$w_s t \propto w_{s:t-1} p(y_t | q_t = k, q_{s:t-1}, y_{1:t-1}) = w_{s:t-1} L_{s,t,k} \quad (23.56)$$

where

$$L_{s,t,k} = p(y_t | q_t = k, z_t, q_{s:t-1}, y_{1:t-1}) p(z_t | q_t = k, y_{1:t-1}) \quad (23.57)$$

The quantity  $L_{s,t,k}$  is the predictive density for the new observation  $y_t$  conditioned on  $q_t = k$  and the history  $q_{s:t-1}$ . In the case of SLDS models, this can be computed using the normalization constant of the Kalman filter, Equation 18.41.

We give some pseudo-code in Algorithm 8. (The step marked "KFupdate" refers to the Kalman filter update equations in Section 18.3.1.) This is known as a mixture of Kalman filters.

If  $K$  is small, we can compute the optimal proposal distribution, which is

$$p(q_t = k | y_{1:t}, q_{s:t-1}) = \hat{p}_{s:t-1}(q_t = k | y_t) \quad (23.58)$$

$$\hat{p}_{s:t-1}(y_t | q_t = k) \hat{p}_{s:t-1}(q_t = k) \hat{p}_{s:t-1}(y_t) \quad (23.59)$$

$$= L_{s,t,k} p(q_t = k | q_{s:t-1}) \quad (23.60)$$

where we use the following shorthand:

$$\hat{p}_{s:t-1}(\cdot) = p(\cdot | y_{1:t-1}, q_{s:t-1}) \quad (23.61)$$

We then sample from  $p(q_t | q_{s:t-1}, y_{1:t})$  and give the resulting particle weight

$$w_s t \propto w_{s:t-1} p(y_t | q_{s:t-1}, y_{1:t}) = w_{s:t-1} L_{s,t,k} \quad (23.62)$$

Since the weights of the particles in Equation 23.62 are independent of the new value that is actually sampled for  $q_t$ , we can compute these weights first, and use them to decide which particles to propagate. That is, we choose the fittest particles at time  $t - 1$  using information from time  $t$ . This is called look-ahead RBPF (de Freitas et al. 2004).

In more detail, the idea is this. We pass each sample in the prior through all  $K$  models to get  $K$  posteriors, one per sample. The normalization constants of this process allow us to compute the optimal weights in Equation 23.62. We then resample  $S$  indices. Finally, for each old particle  $s$  that is chosen, we sample one new state  $q_{s,t} = k$ , and use the corresponding posterior from the  $K$  possible alternative that we have already computed. The pseudo-code is shown in Algorithm 7. This method needs  $O(KS)$  storage, but has the advantage that each particle is chosen using

the latest information,  $y_t$ .

A further improvement can be obtained by exploiting the fact that the state space is discrete. Hence we can use the resampling method of (Fearnhead 2004) which avoids duplicating particles.

### 23.6.2 Application: tracking a maneuvering target

One application of SLDS is to track moving objects that have piecewise linear dynamics. For example, suppose we want to track an airplane or missile;  $q_t$  can specify if the object is flying normally or is taking evasive action. This is called maneuvering target tracking.

Figure 23.7 gives an example of an object moving in 2d. The setup is essentially the same as in Section 18.2.1, except that we add a three-state discrete Markov chain which controls the input to the system. We define  $u_t = 1$  and set

$$B_1 = (0, 0, 0)^T, B_2 = (-1.225, -0.35, 1.225, 0.35)^T, B_3 = (1.225, 0.35, -1.225, -0.35)^T$$

so the system will turn in different directions depending on the discrete state.

Figure 23.7(a) shows the true state of the system from a sample run, starting at  $(0, 0)$ : the colored symbols denote the discrete state, and the location of the symbol denotes the  $(x, y)$  location. The small dots represent noisy observations. Figure 23.7(b) shows the estimate of the state computed using particle filtering with 500 particles, where the proposal is to sample from the prior. The colored symbols denote the MAP estimate of the state, and the location of the symbol denotes the MMSE (minimum mean square error) estimate of the location, which is given by the posterior mean. Figure 23.7(c) shows the estimate computing using RBPF with 500 particles, using the optimal proposal distribution. A more quantitative comparison is shown in Table 23.1. We see that RBPF has slightly better performance, although it is also slightly slower.

Figure 23.8 visualizes the belief state of the system. In (a) we show the distribution over the discrete states. We see that the particle filter estimate of the belief state (second column) is not as accurate as the RBPF estimate (third column) in the beginning, although after the first few observations performance is similar for both methods. In (b), we plot the posterior over the  $x$  locations. For simplicity, we use the PF estimate, which is a set of weighted samples, but we could also have used the RBPF

estimate, which is a set of weighted Gaussians.

### 23.6.3 Application: Fast SLAM

In Section 18.2.2, we introduced the problem of simultaneous localization and mapping or SLAM for mobile robotics. The main problem with the Kalman filter implementation is that it is cubic in the number of landmarks. However, by looking at the DGM in Figure 18.2, we see that, conditional on knowing the robot's path,  $q_{1:t}$ , where  $q_t \in \mathcal{R}^2$ , the landmark locations  $z \in \mathcal{R}^{2L}$  are independent. (We assume the landmarks don't move, so we drop the  $t$  subscript). That is,  $p(z|q_{1:t}, y_{1:t}) = \prod_{l=1}^L p(z_l|q_{1:t}, y_{1:t})$ . Consequently we can use RBPF, where we sample the robot's trajectory,  $q_{1:t}$ , and we run  $L$  independent 2d Kalman filters inside each particle. This takes  $O(L)$  time per particle. Fortunately, the number of particles needed for good performance is quite small (this partly depends on the control / exploration policy), so the algorithm is essentially linear in the number of particles. This technique has the additional advantage that it is easy to use sampling to handle the data association ambiguity, and that it allows for other representations of the map, such as occupancy grids. This idea was first suggested in (Murphy 2000), and was subsequently extended and made practical in (Thrun et al. 2004), who christened the technique FastSLAM. See `rbpfSlamDemo` for a simple demo in a discrete grid world.

已思考 28s

下面先给出逐字不漏的中文翻译；接着用通俗、详细的方式解释；最后给出要点总结。

(注：原式中的下标/逗号有些地方在排版上略显紧凑，例如  $L_{t,k}^s$  也写成  $L_{tk}^s$ 。翻译部分尽量保留原貌；在“解释”里会用更易读的记号。)

## 原文翻译

### 23.6 Rao–Blackwell 化的粒子滤波 (RBPF)

在某些模型中，我们可以把隐变量划分成两类， $q_t$  与  $z_t$ ，使得在已知  $q_{1:t}$  的取值时，我们可以解析地边缘化（积分掉） $z_t$ 。这意味着我们只需要对  $q_{1:t}$  进行采样，并且可以参数化地表示  $p(z_t | q_{1:t})$ 。于是，每个粒子  $s$  都表示一个  $q_{1:t}^s$  的取值以及一个形如  $p(z_t | y_{1:t}, q_{1:t}^s)$  的分布。这种混合型粒子有时也称为分布粒子 (distributional particles) 或塌缩粒子 (collapsed particles) (Koller and Friedman 2009, 第 12.4 节)。

这种方法的优点是：我们降低了采样所在空间的维数，从而降低了估计的方差。因此该技术被称为Rao–Blackwell 化粒子滤波（RBPF，简称），其名称来源于定理 24.20。用一个具体例子来解释该方法最为合适。

### 23.6.1 适用于切换 LG-SSM 的 RBPF

一个 RBPF 可用的典型例子是第 18.6 节讨论的切换线性动力系统（SLDS）模型（Chen and Liu 2000; Doucet et al. 2001）。我们可以用均值与协方差矩阵为每个粒子  $s$  表示  $p(z_t | y_{1:t}, q_{1:t}^s)$ ，其中  $q_t \in \{1, \dots, K\}$ 。

如果我们从先验提议  $q(q_t = k | q_{t-1}^s)$ ，则权重更新变为

$$w_t^s \propto w_{t-1}^s p(y_t | q_t = k, q_{1:t-1}^s, y_{1:t-1}) = w_{t-1}^s L_{t,k}^s \quad (23.56)$$

其中

$$L_{t,k}^s = \int p(y_t | q_t = k, z_t, y_{1:t-1}, q_{1:t-1}^s) p(z_t | q_t = k, y_{1:t-1}, q_{1:t-1}^s) dz_t \quad (23.57)$$

量  $L_{t,k}^s$  是对新观测  $y_t$  的预测密度，其条件为  $q_t = k$  以及历史  $q_{1:t-1}^s$ 。在 SLDS 情况下，这可以用卡尔曼滤波的归一化常数来计算，见式 (18.41)。

我们在**算法 8**中给出了一些伪代码（其中标注为“KFupdate”的步骤指的是第 18.3.1 节中的卡尔曼滤波更新公式）。这被称为**卡尔曼滤波器的混合**（mixture of Kalman filters）。

若  $K$  较小，我们可以计算**最优提议分布**：

$$p(q_t = k | y_{1:t}, q_{1:t-1}^s) = \hat{p}_{t-1}^s(q_t = k | y_t) \quad (23.58)$$

$$= \frac{\hat{p}_{t-1}^s(y_t | q_t = k) \hat{p}_{t-1}^s(q_t = k)}{\hat{p}_{t-1}^s(y_t)} \quad (23.59)$$

$$= \frac{L_{t,k}^s p(q_t = k | q_{t-1}^s)}{\sum_k L_{t,k}^s p(q_t = k | q_{t-1}^s)} \quad (23.60)$$

其中我们使用如下速记：

$$\hat{p}_{t-1}^s(\cdot) = p(\cdot | y_{1:t-1}, q_{1:t-1}^s) \quad (23.61)$$

然后我们从  $p(q_t | q_{1:t-1}^s, y_{1:t})$  采样，并给得到的粒子赋予权重

$$w_t^s \propto w_{t-1}^s p(y_t | q_{1:t-1}^s, y_{1:t-1}) = w_{t-1}^s \sum_k L_{t,k}^s p(q_t = k | q_{t-1}^s). \quad (23.62)$$

由于式 (23.62) 中粒子的权重与实际为  $q_t$  抽到的取值无关，我们可以先计算这些权重，再用它们来决定**传播哪些粒子**。也就是说，我们用来自时间  $t$  的信息在时间  $t - 1$  选择“最合适”的粒子。这被称为**前瞻**（look-ahead）RBPF（de Freitas et al. 2004）。

更具体地说，思想是：我们把先验中的每个样本都通过全部  $K$  个模型各跑一遍，得到每个样本的  $K$  个后验。这个过程的归一化常数允许我们计算式 (23.62) 的最优权重。然后我们重采样  $S$  个索引。最后，对每个被选中的旧粒子  $s$ ，我们采一个新状态  $q_t^s = k$ ，并使用我们已经计算好的、对应于  $K$  个备选之一的后验。伪代码见算法 7。该方法需要  $O(KS)$  的存储，但优点是每个粒子都使用了最新观测  $y_t$  来进行选择。

进一步的改进可利用状态空间是离散这一事实。因此我们可以使用 (Fearnhead 2004) 的重采样方法，从而避免粒子复制。

### 23.6.2 应用：机动目标跟踪

SLDS 的一个应用是跟踪具有分段线性动力学的运动目标。举例来说，假设我们要跟踪一架飞机或导弹； $q_t$  可以指示目标是在正常飞行还是在规避机动。这称为机动目标跟踪。

图 23.7 给出一个在二维平面运动的对象的例子。设置本质上与第 18.2.1 节相同，只不过我们额外加入了一个三状态的离散马尔可夫链，它控制系统的输入。我们令  $u_t = 1$ ，并设

$$B_1 = (0, 0, 0, 0)^T, \quad B_2 = (-1.225, -0.35, 1.225, 0.35)^T, \quad B_3 = (1.225, 0.35, -1.225, -0.35)^T$$

因此系统会根据离散状态向不同方向转弯。

图 23.7(a) 展示了从  $(0, 0)$  出发的一次样本运行中的真实状态：彩色符号表示离散状态，符号的位置表示  $(x, y)$  位置。小点代表带噪观测。图 23.7(b) 展示了使用 500 粒子的粒子滤波（提议为从先验采样）计算得到的状态估计。彩色符号表示状态的 MAP 估计，而符号的位置表示位置的 MMSE（最小均方误差）估计，即后验均值。图 23.7(c) 展示了使用 500 粒子、采最优提议分布的 RBPF 计算得到的估计。表 23.1 给出了更定量的比较。我们看到 RBPF 的性能略好一些，不过也略慢。

图 23.8 可视化了系统的信念状态。(a) 图显示了对离散状态的分布。我们看到，开始时粒子滤波的信念估计（第二列）不如 RBPF（第三列）准确，尽管在最初的几次观测之后，两种方法的表现相似。(b) 图展示了对  $x$  位置的后验。为简便起见，我们使用了 PF 的估计（即一组加权样本），但我们同样可以使用 RBPF 的估计（即一组加权高斯）。

### 23.6.3 应用：Fast SLAM

在第 18.2.2 节，我们介绍了移动机器人中的同步定位与建图 (SLAM) 问题。卡尔曼滤波实现的主要问题在于其对地标数的三次复杂度。然而，通过观察图 18.2 的 DGM，我们看到，在已知机器人的路径  $q_{1:t}$ （其中  $q_t \in \mathbb{R}^2$ ）的条件下，地标位置  $z \in \mathbb{R}^{2L}$  相互独立。（我们假设地标不动，因此省略下标  $t$ 。）也就是说，

$$p(z \mid q_{1:t}, y_{1:t}) = \prod_{l=1}^L p(z_l \mid q_{1:t}, y_{1:t}).$$

因此我们可以使用 RBPF：对机器人的轨迹  $q_{1:t}$  进行采样，并在每个粒子内运行  $L$  个相互独立的二维卡尔曼滤波器。这对每个粒子需要  $O(L)$  的时间。幸运的是，为得到较好的性

能所需的粒子数很小（这在一定程度上取决于控制/探索策略），因此该算法对粒子数基本上是**线性的**。该技术还有一个额外的优点：用采样来处理**数据关联的二义性**很容易，并且它还允许地图的其他表示形式，例如**占据栅格**。这个想法最早由（Murphy 2000）提出，随后由（Thrun et al. 2004）扩展并使之实用，他们将该技术命名为 FastSLAM。在离散网格世界中，一个简单演示见 `rbpfSlamDemo`。

---

## 通俗解释（一步步讲清）

### 1) RBPF 是什么，一句话版

把状态拆成两部分：**难的部分**  $q$  用粒子抽样；**容易的部分**  $z$  在条件于  $q$  的情况下**能解析求后验**（比如高斯可用卡尔曼滤波）。

这样**每个粒子不只是一个点**，而是：

- 一条**离散/低维轨迹**  $q_{1:t}^{(s)}$ ；
  - 配套的一个**参数化分布**  $p(z_t \mid y_{1:t}, q_{1:t}^{(s)})$ （如高斯的均值/协方差）。
- 这就叫**Rao–Blackwell 化**：把能积分的那部分**积分掉**，让采样只在**低维难点**上进行  
 $\Rightarrow$  **维度更低、方差更小、更稳更省样本**。

### 2) 为什么更高效？

- 纯 PF 要在整个  $(q, z)$  空间里撒粒子；RBPF 只在  $q$  空间撒， $z$  用解析更新（例如 KF）。
- 相当于把“随机”换成“确定/解析”一部分，**减少随机噪声与方差**。

### 3) SLDS（切换线性系统）里怎么做？

- **设定**： $q_t \in \{1, \dots, K\}$  表示**模式**（直行/左转/右转...）， $z_t$  连续（位置/速度）且**线性-高斯**。
- **每个粒子**携带：离散模式历史  $q_{1:t}^{(s)}$  + 对  $z_t$  的**高斯后验**（均值/协方差）。
- **若从先验提议**  $q(q_t = k \mid q_{t-1}^{(s)})$ ：
  - 权重更新变成：旧权重  $\times$  **预测似然**  $L_{t,k}^s$ ；
  - 这个  $L_{t,k}^s$  正是“KF 预测步 + 观测更新”的**归一化常数**（即观测的预测密度）。
- **若  $K$  小**，可用**最优提议**  $p(q_t \mid y_{1:t}, q_{1:t-1}^{(s)}) \propto L_{t,k}^s p(q_t = k \mid q_{t-1}^{(s)})$ ，权重不依赖抽到的  $q_t$ 。
  - 这允许**先算权重再重采样**（基于  $y_t$  的“前瞻”选择强粒子），然后对选中的粒子再确定  $q_t$ ，并直接拿到对应的 KF 后验。
  - 代价：需存  $K$  份候选后验 ( $O(KS)$  空间)；好处：**用上了最新数据**来挑粒子。
  - 进一步：利用离散结构，可用 Fearnhead (2004) 的重采样避免复制。

### 4) 机动目标跟踪的直观例子

- 模式  $q_t$  控制飞机/导弹的转向（三种  $B$  向量）；观测有噪。
- 比较：
  - PF（先验提议） vs RBPF（最优提议），均用 500 粒子；
  - RBPF 精度略好但稍慢（因为每粒子要跑更“聪明”的更新）；
  - 早期 RBPF 的离散状态后验更准；位置后验可视化：PF 是加权样本，RBPF 是加权高斯。

## 5) FastSLAM：为什么 RBPF 特别合适

- 在 SLAM 里，已知机器人路径  $q_{1:t}$  时，各地标位置  $z_l$  彼此独立。
  - 所以：外层对路径用粒子 (PF/RBPF) 采样；内层对每个地标各跑一个 2D KF。
  - 每粒子代价  $O(L)$ ，而所需粒子数较少  $\Rightarrow$  总体很高效；还能自然处理数据关联、以及采用占据栅格等其他地图表示。
  - 这就是 FastSLAM (Murphy 2000; Thrun et al. 2004)。
- 

## 要点总结

- RBPF 思想：把状态拆成  $(q, z)$ 。对  $q$  用粒子采样；对  $z$  在条件于  $q$  下解析积分/更新（如卡尔曼）。 $\Rightarrow$  降维、降方差、提效率。
- SLDS 实作：每粒子携带模式历史  $q_{1:t} +$  连续态的高斯后验；
  - 先验提议：权重乘预测似然  $L_{t,k}^s$ ；
  - 最优提议（小  $K$  可行）： $p(q_t \mid y_{1:t}, q_{1:t-1}) \propto L_{t,k}^s p(q_t = k \mid q_{t-1})$ ；
  - 前瞻 RBPF：先算权重→重采样→再定  $q_t$  并复用对应 KF 后验；可用 Fearnhead 重采样减少复制。
- 收益：相较纯 PF，RBPF 通常更准/更稳，但每步计算/存储更贵（如需  $O(KS)$  存储）。
- 应用：
  - 机动目标跟踪：RBPF 相比 PF 精度略优、速度略慢；
  - FastSLAM：条件独立把大问题拆成粒子外层 +  $L$  个地标 KF，复杂度对粒子数近线性，易处理数据关联与栅格地图。
- 实践要点：当模型中存在“条件可解析”的连续子结构（高斯/线性），就考虑 RBPF：把能“算”的那部分交给解析（卡尔曼/闭式），把“难”的部分留给粒子。这样常常更省粒子、更稳健。

这一节主要在讲什么？我不太懂

不急~这节 (23.6) 讲的是Rao–Blackwell 化粒子滤波 (RBPF)：当状态能拆成两部分时，把能用公式算的那部分用解析方法（如卡尔曼滤波）直接算掉，只对难抽样的那部分用粒子。这样就降维、降方差、提效率。

## 一句话版

RBPF = 在粒子滤波里“先算能算的（连续高斯部分）”，只对“算不动的（离散/切换部分）”抽样。

每个粒子不再是单一状态点，而是“离散轨迹  $q_{1:t}$  + 连续态  $z_t$  的高斯后验（均值/协方差）”。

## 怎么做（直观）

1. 把隐藏变量拆成  $q_t$  (难) 和  $z_t$  (易)。给定  $q_{1:t}$  时， $z_t$  的后验能解析得到（常用卡尔曼滤波）。
2. 粒子只在  $q_{1:t}$  空间里扩展；对每个粒子，用 KF 更新出  $p(z_t | y_{1:t}, q_{1:t})$  (一个高斯)。
3. 粒子权重用预测似然（来自 KF 的归一化常数）更新；必要时重采样。
4. 若离散模式数  $K$  不大，可用最优提议（看见  $y_t$  再决定  $q_t$ ），甚至做前瞻 RBPF：先用  $y_t$  评估每个粒子在所有模式下的好坏，再挑“最强”的粒子去扩展。

## 典型应用

- 切换线性系统 (SLDS)：模式  $q_t$  (直行/转弯...) + 连续态  $z_t$  (位置/速度，高斯)。  
RBPF ≈ 一堆卡尔曼滤波器的混合，通常比纯 PF 更准但稍慢。
- 机动目标跟踪：目标临时改变运动模式时，RBPF能稳住估计。
- FastSLAM (同步定位与建图)：对机器人路径用粒子，对每个地标用一个小 KF，整体近似线性于地标数，实用高效。

## 为什么更好

- 只在低维的  $q$  上抽样，省粒子； $z$  交给解析计算，方差更小、数值更稳。
- 易引入“看数据再提议”的策略（最优/近似最优提议），进一步减少退化。

## 何时用

- 模型里存在“给定  $q$  时  $z$  可解析（多为线性-高斯）”的结构；
- 需要比纯 PF 更稳定、更省样本的滤波。

## 要点小结

- 核心：粒子采样  $q$ ，解析更新  $z$ （卡尔曼）；每粒子 =  $q_{1:t}$  + 一份高斯后验。

- **权重**: 用 KF 的预测似然更新；可做**最优/前瞻提议**。
- **收益**: 降维+降方差  $\Rightarrow$  通常更准、更高效。
- **应用**: SLDS、机动目标跟踪、FastSLAM (路径粒子 + 地标KF)。

基于本对话里我贴过的整章原文 (19章)，输出：①知识点总结 (分点+关键词+用途) ②子章节关联 (依赖/对比/递进+关系句，附Mermaid可选) ③本章总结。

已思考 20s

下面基于你在对话里贴过的整章内容 (本章“蒙特卡罗推断”，涵盖 23.1–23.6)，给出三部分输出：

---

## ① 知识点总结 (分点 + 关键词 + 用途)

- **蒙特卡罗推断的动机** (23.1)
  - **关键词**: 确定性近似局限、抽样近似期望、样本平均、后验预测
  - **用途**: 当变分/解析法推不动或精度受限时，用随机样本逼近任意后验量与预测量。
- **标准分布采样技巧** (23.2)
  - **逆 CDF (分位数) 法**
    - **关键词**:  $X = F^{-1}(U)$ 、均匀噪声、指数分布示例
    - **用途**: 单变量分布可直接采样，常作更复杂算法的子程序。
  - **Box-Muller 与多元高斯**
    - **关键词**: 极坐标、两份独立  $N(0, 1)$ 、Cholesky 变换
    - **用途**: 高质量正态样本与多元高斯采样的基石。
- **拒绝采样 (RS)** (23.3)
  - **关键词**: 提议  $q$ 、上包络  $Mq$ 、接受率  $1/M$ 、ARS (对数凹)
  - **用途**: 逆 CDF 不可用时的通用备选；在一维/低维或可构造紧上界时有效；贝叶斯中可用先验作提议。
  - **要点**: 高维指~~指~~级退化；选择贴近目标的  $q$  与最小  $M$  以提升效率。
- **重要性采样 (IS)** (23.4)
  - **关键词**: 权重  $w = p/q$ 、最优提议  $q^* \propto |f|p$ 、自归一化、有偏但一致、SIR、似然加权

- **用途**: 高效估计积分/罕见事件概率; 仅知未归一化密度时仍可用; 把加权样本再采样成无权样本 (SIR)。
  - **拓展**: DGM 中的似然加权避免逻辑采样的高拒绝率。
  - **粒子滤波 (PF = SIS + 重采样)** (23.5)
    - **关键词**: 顺序重要性采样、权重递推、有效样本数  $\hat{S}_{\text{eff}} = 1 / \sum w^2$ 、重采样 (多项式/系统/残差/分层)、样本贫化、最优提议  $p(z_t | z_{t-1}, y_t)$ 、无迹 PF
    - **用途**: 在线/递归贝叶斯滤波, 适用于非线性、非高斯的跟踪与时间序列预测。
    - **应用**: 机器人定位 (MCL)、视频目标跟踪、随机波动率等。
  - **Rao–Blackwell 化粒子滤波 (RBPF)** (23.6)
    - **关键词**: 变量拆分  $(q, z)$ 、解析积分 (卡尔曼)、混合粒子、SLDS、前瞻 RBPF、FastSLAM
    - **用途**: 当给定  $q$  时  $z$  可解析 (线性–高斯)  $\rightarrow$  对  $q$  抽样、对  $z$  解析更新, **降维降方差**; 在 SLDS、SLAM 中尤为高效。
- 

## ② 子章节关联 (依赖 / 对比 / 递进)

- 23.1  $\rightarrow$  23.2 (依赖) : 先给出“为何用抽样”的总目标, 再提供**基础采样砖块** (逆 CDF、Box–Muller、Cholesky) 供后续算法复用。
- 23.2  $\leftrightarrow$  23.3 (对比) : 23.2 直接采样“可逆/可构造”的分布; 23.3 用上包络间接采样难分布 (但高维差)。
- 23.3  $\rightarrow$  23.4 (递进) : 从“接受/拒绝”改为“**加权利用所有样本**”, 方差分析引出**提议分布设计**与自归一化技巧。
- 23.4  $\rightarrow$  23.5 (递进) : 把 IS 的思想**时序化**  $\rightarrow$  顺序重要性采样 + **重采样**解决权重退化, 形成在线滤波器。
- 23.5  $\rightarrow$  23.6 (递进+融合) : 在 PF 框架中引入**解析子模块 (卡尔曼)** 对可解子状态进行 Rao–Blackwell 化, **进一步降维降方差**; 在 SLDS/SLAM 中落地。

可选 Mermaid 关系图:

```
graph LR
    A[23.1 动机与总体思路] --> B[23.2 标准分布采样\\n逆CDF/Box-Muller/Cholesky]
    B --> C[23.3 拒绝采样\\n上包络/ARS/高维退化]
    C --> D[23.4 重要性采样\\n最优提议/自归一化/SIR/似然加权]
    D --> E[23.5 粒子滤波 PF\\nNSIS+重采样/提议选择/退化与缓解]
    E --> F[23.6 RBPF\\n变量拆分+卡尔曼/SLDS/FastSLAM]
```

核心关系句:

- “**采样砖块 (23.2) 支撑RS/IS (23.3/23.4)**, IS 的加权思想被**时序化**成 PF (23.5), 而 PF 进一步在可解析子结构上做 \*\*Rao–Blackwell 化 (23.6) \*\*以降方差。”

### ③ 本章总结（实战导向）

- **主线：**用随机样本近似复杂后验与预测。由“能直接采样的基础方法（23.2）”到“间接抽样（RS，23.3）”再到**权重校正（IS，23.4）**，最后扩展到**在线滤波（PF，23.5）**与**结构化降维（RBPF，23.6）**。
- **方法选型速查：**
  - **单变量、分位数可得** → 逆 CDF；**正态/多元高斯** → Box–Muller + Cholesky。
  - **目标难采但能找紧上界、低维** → 拒绝采样/ARS。
  - **估计积分/稀有事件**，有可控提议 → 重要性采样（必要时 SIR）。
  - **在线状态估计（非线性/非高斯）** → 粒子滤波（选好提议、设重采样触发阈）。
  - **模型可拆分：给定离散/慢变量时连续部分可解析** → RBPF（SLDS、FastSLAM 的首选）。
- **关键工程要点：**
  - **提议分布质量决定效率：**让  $q$  覆盖  $|f|p$  大的区域，尾部不轻于目标；在 PF 中尽量用**数据驱动提议**（或其近似）。
  - **控制权重方差：** IS 中关注 ESS；PF 中监控  $\hat{S}_{\text{eff}}$ ，按需重采样并配合**resample-move / 正则化**缓解样本贫化。
  - **高维警告：** RS 接受率随维度指数下降；IS/PF 也会因不当提议导致**权重塌陷**，需自适应/结构化（RBPF）来稳住。
- **一条话术：先能采、再会权、再时序化、再结构化。**先解决“怎么拿样本”，再用权重把偏差校回；来到序列数据时做 PF；若模型允许，把可解部分交给解析更新（RBPF），把难点留给粒子。

如果你想把这些方法落地，我可以按你的任务（比如轨迹跟踪/SLAM/罕见事件估计）给出具体的“提议选择 + 重采样策略 + 样本量与诊断”清单。

请用通俗易懂的话总结本章，使我能够理解

已思考 17s

好的！这章讲的其实就是一条清晰的“抽样路线图”，教你不用复杂推导，也能靠随机样本把贝叶斯推断做出来。下面用最直白的方式把整章捋顺：

## 大图景：用“抽样”代替“推公式”

- 以前的确定性近似（比如变分）又难推又有偏差。

- 蒙特卡罗思想：只要能从目标分布抽样，很多想要的量（期望、概率、预测）都能用样本平均来估。样本越多越准。
- 

## 工具箱一：怎么从常见分布里“拿到样本”

1. 逆 CDF (分位数) 法：先抽均匀数  $u \in [0, 1]$ ，再算  $x = F^{-1}(u)$ 。单变量分布通吃（比如指数分布）。
  2. Box-Muller：一次生成两条标准正态样本；配合 Cholesky，就能拿到多元高斯样本。  
👉 这些是后面更复杂方法的“底层砖块”。
- 

## 工具箱二：目标分布不好抽，怎么办？

### A. 拒绝采样 (RS)

- 思想：找个好抽的分布  $q$ ，把它放大成“棚子”  $Mq$  盖住目标形状，再扔飞镖：落在目标曲线下就收下，否则丢掉。
- 优点：简单、用不上归一化常数。
- 缺点：高维很惨（接受率指数级变差）。
- 变体：自适应拒绝采样 (ARS) —— 对对数凹分布用分段线性上界自动把棚子越搭越贴，拒绝率会下降。

### B. 重要性采样 (IS)

- 思想：还是从好抽的  $q$  抽，但不丢样本，而是给每个样本一个权重  $w = p/q$ ，用加权平均来纠偏。
  - 关键：选好提议  $q$ 。理论最优是让  $q$  在“重要区域”( $|f|p$  大的地方)更密；工程上常自适应调  $q$ 。
  - 只知道未归一化密度也行：先算未归一化权重，再归一化（自归一化估计，有限样本有偏但一致）。
  - 应用：
    - 似然加权（有向图模型里有观测时，比“冲突就丢”的逻辑采样高效）。
    - SIR（采样-重要性-再采样）：把加权样本再采样成无权样本，便于后续使用。
- 

## 工具箱三：有时间序列时，边来数据边更新

### 粒子滤波 (PF = 顺序重要性采样 + 重采样)

- 用一堆“粒子 + 权重”在线逼近  $p(z_t | y_{1:t})$ 。

- **退化问题**: 几个粒子权重独大、其余接近 0  $\Rightarrow$  浪费算力。
    - 诊断: **有效样本数**  $ESS = 1/\sum w^2$  下降。
    - 对策: **重采样** (复制强粒子、淘汰弱粒子), 并用系统/残差/分层等方式降方差; 必要时加一点**MCMC 抖动或核平滑**, 避免样本贫化。
  - **提议分布**:
    - 朴素: 从**先验提议** (容易做, 但若传感器很准就低效)。
    - 更好: 看见数据再提议 (**最优提议**  $p(z_t | z_{t-1}, y_t)$ ) ; 常需近似 (如**无迹粒子滤波**)。
  - 场景: **机器人定位、视频跟踪、非线性/非高斯时间序列** (如随机波动率)。
- 

## 进阶：能算的先算，难的再抽 (RBPF)

### Rao–Blackwell 化粒子滤波 (RBPF)

- 把状态拆成两部分:
    - $q_t$ : 难抽 (常是离散/切换状态)  $\rightarrow$  用粒子抽;
    - $z_t$ : 给定  $q$  可解析 (多为线性-高斯)  $\rightarrow$  用卡尔曼等公式更新。
  - 每个粒子 = “一条  $q$  的轨迹” + “一个关于  $z_t$  的高斯后验 (均值/协方差) ”。
  - 好处: **降维 + 降方差**, 更稳更省粒子。
  - 应用:
    - **切换线性系统 (SLDS)** : 相当于“卡尔曼滤波器的混合”。
    - **FastSLAM**: 对**机器人路径**用粒子; 对**每个地标**跑一个小卡尔曼, 复杂度对粒子数近线性, 还能方便处理数据关联与栅格地图。
- 

## 选型小抄 (不看公式也能用)

- **单变量、能算分位数**  $\rightarrow$  逆 CDF。
  - **正态/多元高斯**  $\rightarrow$  Box–Muller + Cholesky。
  - **低维、能找到紧上界**  $\rightarrow$  拒绝采样/ARS。
  - **要算积分/稀有事件**、能设计提议  $\rightarrow$  重要性采样 (必要时 SIR)。
  - **在线状态估计 (非线性/非高斯)**  $\rightarrow$  粒子滤波 (设 ESS 阈值重采样, 尽量用数据驱动提议)。
  - **模型可拆: 给定离散/慢变量时连续部分可解析**  $\rightarrow$  RBPF (SLDS、FastSLAM 的首选)。
-

## 易踩坑与小诀窍

- **高维慎用拒绝采样**: 接受率会指数级崩。
  - **提议要“覆盖尾部”**: 否则权重爆炸、方差无穷。
  - **监控 ESS**: 太低就重采样；重采样后适当“抖动”，防止样本贫化。
  - **不够稳？** 提高粒子数、改进提议（用上当前观测）、或引入结构化（RBPF）。
- 

## 一句话收尾

这章就是一条从“能抽”到“会权”，再到“能递推”，最后“能结构化降维”的路线：

**先拿到样本 → 用权重把偏差校回 → 把方法做成在线滤波 → 能解析的先算掉，只在难点上抽样。**

按这条路走，你就能把复杂的贝叶斯推断稳稳落地。