

2 近似参数估计方法

本章探讨基于优化的参数估计方法，其优点是可以应用于高维模型和大数据场合。其中极大似然估计法可以估计参数的一个位置信息，而黑箱变分推断还能够估计参数的大致范围。与马尔科夫链蒙特卡洛方法相比，这类方法的缺陷是无法获得参数的全部信息。

2.1 极大似然估计

为了衡量参数估计的效果，一般可以将频率方法估计的参数作为基准。概率模型参数的位置可以使用极大似然(或最大后验)方法进行估计。如式(2.1)所示，对数函数为单调递增函数，如果将负的对数概率定义为势能（或者目标函数），则极大概率对应于最小势能。因此可以通过寻找最小势能来获得极大似然估计。如果势能函数属于凸函数，则可以采用牛顿法快速寻找到全局最优解。对于非凸目标函数，一般只能找到局部最优解。如果目标函数的维数较高，则无法使用基于海森矩阵的优化算法，而只能使用基于梯度的算法，导致收敛速度较慢。

$$U(\theta) = -\log P(\theta) \quad (2.1)$$

极大似然估计获得的仅是参数的位置信息，为了得到更全面的分布信息，还可以计算参数的置信区间。获得置信区间的一种方式是利用势能的海森矩阵。对于多元正态分布，其势能的海森矩阵是协方差矩阵的逆。因此海森矩阵的逆可以作为协方差矩阵的估计。

梯度下降法是一种应用广泛的优化算法，其特点是实现简单和计算量小。由于不使用海森矩阵，这类算法能够应用于高维模型。该算法的缺点是收敛慢，因此中低维模型的最优化问题一般使用牛顿法求解。由于梯度方向与最后主成分方向近似，梯度下降算法对于高维模型容易产生往复移动，导致优化效果不理想。随机梯度下降法每次根据一个数据的梯度朝着势能下降的方向移动。ADAM 算法是一种改进的随机梯度下降算法。它根据梯度和梯度的平方的历史平滑值决定下降方向^[70]。

2.2 黑箱变分推断

黑箱变分推断使用一个近似分布来拟合目标分布。经典方法采用无相关的正态分布作为近似分布。这种方法能够应用于目标分布为无相关的多元正态分布。一种改进方案是采用拉普拉斯分布或者相关正态分布作为近似分布。另一种方案是将参数变换成正态分布。

2.2.1 无相关正态分布

一些已有的文献基于隐变量和显变量进行公式推导，难以揭示黑箱变分推断的本质思路。对概率模型进行参数估计时，已经观察到的数据可以看做概率密度函数的一部分而不必加以考虑。实际上，在抽样过程中，概率模型中唯一可变的是函数的变量。因此黑箱变分推断的推导应当仅依据函数变量进行。

假设我们需要研究概率密度函数为 $P(x)$ 的概率分布，其中 x 是变量。如果 $P(x)$ 的形式过于复杂而无法直接研究，我们可以用一个概率分布 $Q(x)$ 来近似表示。根据信息论^[71]，用 $Q(x)$ 来近似 $P(x)$ 所产生的误差可以用相对熵（即 KL-散度）来衡量，即式(2.2)。

$$\text{KL}(P \parallel Q) = \int_{-\infty}^{\infty} P(x) \times \log \frac{P(x)}{Q(x)} dx = H(P, Q) - H(P) \quad (2.2)$$

其中 $H(P)$ 为 P 目标概率的熵，而 $H(P, Q)$ 为 P 和 Q 的交叉熵。由于目标概率 P 为未知分布，其熵未知，同时也无法计算交叉熵项。因此上述 KL 散度无法直接进行计算。

如果换一种思路，交换 P 和 Q 的位置，计算用目标分布 $P(x)$ 来模拟已知 $Q(x)$ 所产生的误差，则得到式(2.3)来衡量二者的差别。

$$\text{KL}(Q \parallel P) = \int_{-\infty}^{\infty} Q(x) \times \log \frac{Q(x)}{P(x)} dx = H(Q, P) - H(Q) \quad (2.3)$$

由于 $Q(x)$ 为已知分布，因此熵 $H(Q)$ 已知。此外，由于已知 $Q(x)$ ， $H(Q, P)$ 可以用式(2.4)予以近似，其中 x_i 是从 $Q(x)$ 中抽样的样本，而 n 是抽样个数。

$$\int_{-\infty}^{\infty} Q(x) \times \log \frac{1}{P(x)} dx \approx -\frac{1}{n} \sum_i^n \log(P(x_i)) \quad (2.4)$$

如果拟合分布 Q 为无相关的多维正态分布，则 Q 的熵可以按式(2.5)计算，其中 σ_i 为第 i 维的标准差参数。

$$H(\sigma) = \sum_{i=1}^D \frac{1}{2} \times (\log(2\pi\sigma_i^2) + 1) \quad (2.5)$$

根据 KL 散度的性质，如果 $Q(x)$ 和 $P(x)$ 相等，则 $\text{KL}(Q \parallel P)$ 为零，否则其为正。因此可以用优化算法求最优的参数值。最常用的拟合分布为无相关的多维正态分布，优化函数的变量 x 为无相关多维正态分布的均值 μ 和标准差 σ 的函数。黑箱变分推断算法与极大似然估计类似，二者都使用优化算法求最优值，区别仅仅是采用了不同的目标函数。算法 2.1 第 3 行显示生成标准正态随机数， x 为 μ 和 σ 的符号函数，从而可以利用自动求导算法^[72]。生成随机数的时候不能通过直接指定参数的方式，因为这种方式生成的不再是 μ 和 σ 的符号函数，而是一个数值。算法第 5 行 f 是 x 的符号函数，因此也是 μ 和 σ 的符号函数。黑箱变分的目

标函数 $KL(Q \parallel P)$ 是 μ 和 σ 的符号函数，因此可以对变量求导，从而可以应用基于导数的优化算法。

算法 2.1： 黑箱变分推断的损失函数

输入： μ 和 σ

```

1:  $f=0$ 
2: for  $i = 1$  to  $n$ 
3:    $\varepsilon \sim N(0,1)$ 
4:    $x = \varepsilon \times \sigma + \mu$ 
5:    $f = f - \log p(x)$ 
6: end
7:  $f = f/n - H(\sigma)$ 
8: return  $f$ 

```

输出： $KL(Q \parallel P)$

黑箱变分推断的结果是否准确与优化算法的参数，优化步数，以及抽样次数 n 有关。优化步数和抽样次数越大，则估计越准确。使用 ADAM 优化算法可以降低步长的影响。

根据损失函数，利用自动求导工具获得一阶导数，随后使用基于梯度下降的算法求得最优的 μ 和 σ 。如图 2.1 所示，使用 ADAM 优化算法，用无相关的二维正态分布去拟合无相关的二维正态分布（式 2.6）。坐标轴为空间维度，红色虚线椭圆为初始的拟合分布对应的椭圆，经过 2^{12} 次优化，蓝色虚线表示的最终拟合分布与黑色实线显示的目标分布近似重合，其中黄色虚线椭圆为第 2^i 次的中间步骤。由于拟合分布与目标分布近似，因此当目标分布为无相关的正态分布时，

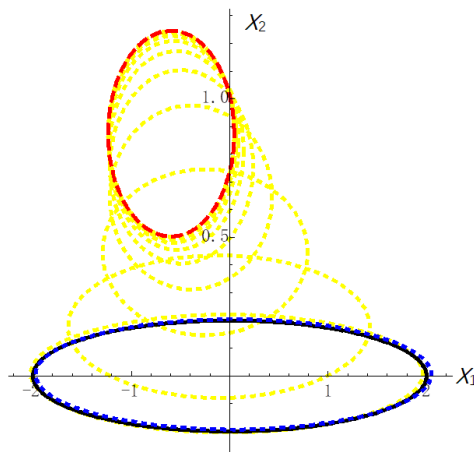


图 2.1 无相关的二维正态分布的变分推断过程，红色虚线椭圆为初始的拟合正态分布，黄色虚线为中间运行过程的拟合分布，黑色实线椭圆为目标分布，蓝色虚线为最终位置

这种方法可以取得较好的效果。

$$P(x_1, x_2) \propto e^{-\left(\frac{x_1^2}{2^2} + \frac{x_2^2}{0.2^2}\right)} \quad (2.6)$$

黑箱变分推断对于其它类型的目标分布的估计误差较大。图 2.2 显示目标分布为相关的二维正态分布（式 2.7）的运行结果。由于目标分布的变量之间具有一定的相关程度，但拟合分布是无相关的二维正态分布，用拟合分布去近似目标分布会产生一些误差。

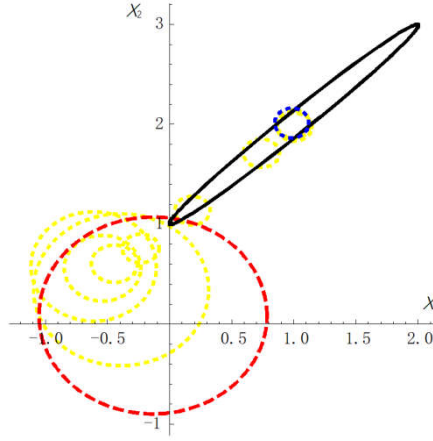


图 2.2 相关的二维正态分布的黑箱变分推断，红色虚线椭圆为初始的拟合正态分布，黄色虚线为中间运行过程的拟合分布，黑色实线椭圆为目标分布，蓝色虚线为最终位置

$$P(x_1, x_2) \propto e^{-\frac{1}{2} \begin{pmatrix} x_1-1 & x_2-2 \end{pmatrix} \begin{pmatrix} 1 & .99 \\ .99 & 1 \end{pmatrix}^{-1} \begin{pmatrix} x_1-1 \\ x_2-2 \end{pmatrix}} \quad (2.7)$$

在这个例子中，拟合分布的标准差远小于目标分布的标准差，定性地说对于高相关模型参数的范围存在较为严重的误差。可以预见，这种方法不适合相关程度较高的场合。一般对于中低维模型，存在较多的可选择参数估计方法，黑箱变分推断的优势不明显，其主要应用场合为高维模型和大数据场合。黑箱变分原则上也可以使用牛顿法进行优化，目的是加快收敛速度，但无法改变目标分布和拟合分布之间存在的差别。

2.2.2 拉普拉斯分布

一种简单的改进方案是将正态分布替换成拉普拉斯分布。设一维拉普拉斯分布的概率密度函数具有式(2.8)的形式。则 D 维可分离的拉普拉斯分布的熵可以用式(2.9)计算，其中 σ_i 为每一维上的尺度参数。

$$\text{laplace}(x; \mu, \sigma) \propto \frac{e^{-\frac{|x-\mu|}{\sigma}}}{2\sigma} \quad (2.8)$$

$$H(\sigma) = D \times (1 + \log(2)) + \sum_{i=1}^D \log(\sigma_i) \quad (2.9)$$

于是，仿照经典黑箱变分的算法，就得到基于拉普拉斯分布的改进算法。在这个算法中，随机数用拉普拉斯分布生成，而 H 为拉普拉斯分布的熵。

算法 2.2： 拉普拉斯黑箱变分推断的损失函数

输入： μ 和 σ ,

```

1:   $f=0$ 
2:  for  $i = 1$  to  $n$ 
3:       $\varepsilon \sim L(0,1)$ 
4:       $f = f - \log p(\varepsilon \times \sigma + \mu)$ 
5:  end
6:   $f = f/n - H(\sigma)$ 
7:  return  $f$ 

```

输出： $KL(Q||P)$

由于拟合分布为拉普拉斯分布，因此当目标分布也服从拉普拉斯分布时，二者可以完全匹配，即 KL 散度最小。与以往讨论类似，定性实验采用式(2.10)所示的概率密度函数。

$$P(x_1, x_2) = \text{laplace}(x_1; 0, 2) \times \text{laplace}(x_2; 0, 0.2) \quad (2.10)$$

由于 x_1 和 x_2 可分离，因此总的概率密度函数分离成两个密度的乘积。图 2.3 中，红色椭圆代表初始位置和尺度，黑色实线椭圆表示目标位置和尺度，黄色椭圆为优化的中间结果，而蓝色虚线椭圆为最终估计结果。实验显示，经过一系列优化，拟合分布与目标分布趋向一致，说明这种改进方案是可行的。

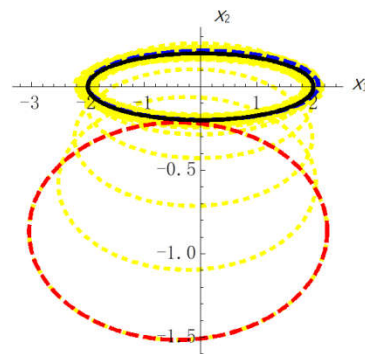


图 2.3 拉普拉斯黑箱变分估计过程，红色虚线：开始位置，黑色实线：目标位置，蓝色虚线：最终位置

拉普拉斯分布属于长尾分布，改进算法可能比基于正态分布的算法更为稳定，但拟合分布仍属于无相关分布，因此仍存在相应的局限性。

2.2.3 相关正态分布

根据上文可知，由于采用了无相关的多维正态分布作为拟合分布，导致算法对有相关的多维正态分布的估计不准确。针对于这点不足，本人用相关多维正态分布作为拟合分布。假设拟合分布为式(2.11)。

$$P(x; \mu, \Sigma) \propto e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \quad (2.11)$$

如果对协方差矩阵进行分解（式 2.12），并作变量变换（式 2.13），则拟合分布变为标准多维正态分布的形式(式 2.14)。

$$\Sigma = M^2 \quad (2.12)$$

$$z = M^{-1}(x - \mu) \quad (2.13)$$

$$e^{-\frac{1}{2}(x-\mu)^T M^{-T} \cdot M^{-1}(x-\mu)} = e^{-\frac{1}{2}z^T \cdot z} \quad (2.14)$$

根据上述讨论可知，对于无相关的正态分布，其熵具有已知的形式，其中 σ_i^2 为第 i 维的方差。类似地，对于相关的多维正态分布可采用同样的熵，此时 σ_i^2 为协方差矩阵（即 M^2 ）的第 i 个特征值，表示拟合分布在某个主成分维度上的方差。

如算法 2.3 所示，首先根据 $D \times (D+1)/2$ 维向量 m 构造 $D \times D$ 维矩阵 M ，然后求协方差矩阵特征值的平方根。在第 6 行，根据均值和 M 矩阵变换得到 x ， x 近似服从目标概率分布。

算法 2.3：相关正态黑箱变分推断的损失函数

输入： μ 和 m

```

1: 用向量  $m$  构造矩阵  $M$ 
2:  $\sigma = \text{eigenvalues}(M^2)^{1/2}$ 
3:  $f = 0$ 
4: for  $i = 1$  to  $n$ 
5:    $z \sim N(0,1)$ 
6:    $x = M \cdot z + \mu$ 
7:    $f = f + \log P(x)$ 
8: end
9:  $f = f/n - H(\sigma)$ 
10: return  $f$ 

```

输出： $KL(Q||P)$

由于拟合分布为相关的正态分布，因此可以使用相关的正态分布检验改进算

法。采用公式(2.7)作为目标分布，此时目标概率的相关程度较高。如图 2.4 所示，红色椭圆为初始位置，黄色椭圆对应于优化的中间阶段，黑色实线椭圆为目标分

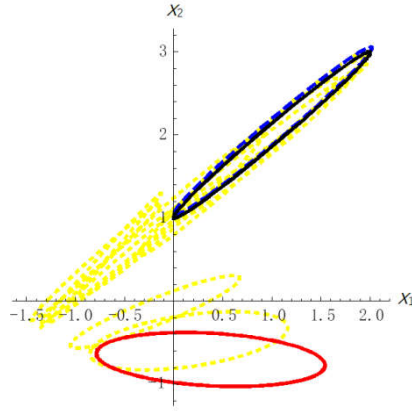


图 2.4 多维正态黑箱变分，红色：开始位置，蓝色虚线：最终位

布，蓝色实虚椭圆为最终估计的拟合分布。由于拟合分布与目标分布比较接近，说明改进方案达到了预期目的。

这种基于协方差矩阵的改进方案的局限在于，需要计算矩阵的特征值和特征向量的导数，计算量比较大。目标概率的协方差信息存在于海森矩阵中，但算法却没有利用这一信息。另一方面，这个算法的参数维数为变量 q 维数的平方级别，难以应用于高维模型，从而失去了黑箱变分的主要优点。

2.2.4 参数变换

在以前的讨论中，假设 P 和 Q 的参数相同。如果 Q 为多维正态分布，那么 P 也应当近似服从多维正态分布。如果 P 分布与多维正态分布差距较远，则无法应用黑箱变分。但假若 P 的参数变换后服从正态分布，那么就可以利用变换后的参数进行黑箱变分推断，对变换后的参数进行估计，然后再变换到原来的参数，就得到 P 分布原来参数的估计。本文用一个环形分布（式 2.15）演示这种参数变换方法。

$$P(x_1, x_2; \mu, \sigma) \propto e^{-\frac{(\sqrt{x_1^2 + x_2^2} - \mu)^2}{2\sigma^2}} = e^{-\frac{(r - \mu)^2}{2\sigma^2}} \propto P(r; \mu, \sigma) \quad (2.15)$$

该环形分布的变量 x_1 和 x_2 的散点形成了环状分布，与拟合分布（如正态分布）差别较大。如果将欧氏坐标变换成极坐标，那么半径 r 服从正态分布，而角度服从均匀分布，因此半径可以用拟合分布进行近似。算法 2.4 显示，由于角度服从固定区域的均匀分布，因此只需考虑半径即可，即半径的均值和标准差为黑箱变分待估计的参数。

算法 2.4：参数变换的损失函数

输入： μ 和 σ

```
1:  $f=0$ 
2: for  $i = 1$  to  $n$ 
3:    $\varepsilon \sim N(0,1)$ 
4:    $r = \varepsilon \times \sigma + \mu$ 
5:    $\theta \sim U(0,2\pi)$ 
6:    $x_1 = r \times \cos(\theta)$ 
7:    $x_2 = r \times \sin(\theta)$ 
8:    $f = f + \log\text{prob}(x_1, x_2)$ 
9: end
10:  $f = f/n - H(\sigma)$ 
11: return  $f$ 
```

输出： $KL(Q\|P)$

图 2.5 显示从任意位置开始，优化逐渐接近目标分布的参数 μ 和 σ 真值(10, 0.01)。说明算法正确估计出环形分布的参数。受到初始值的影响，半径 r 的均值参数也可能为负值，因为当 θ 转向 π 角度时， r 的符号会改变。当半径 r 的标准差远小于其均值时，采用其它方法（如 MCMC）估计圆环的参数较为困难，因为在环上随机游走容易移出高概率区域。在这种情况下，即坐标变换后服从正态分布（或均匀分布），那么进行黑箱变分推断具有一定的优势。参数变换方法的主要难点在于寻找到合适的坐标变换。对于较为复杂的多维模型，一般不存在这种变换，因此参数变化的方式不具有普遍性。

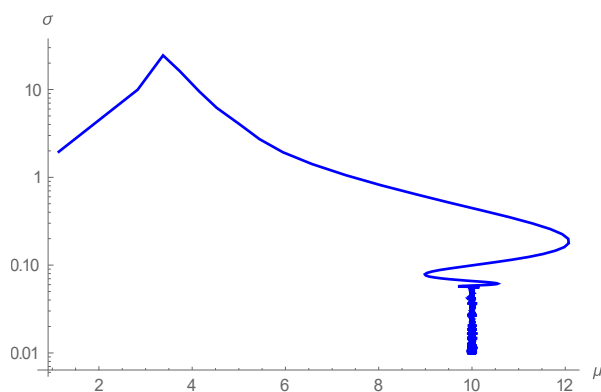


图 2.5 优化过程，左上为初始位置，右下为最终位置

2.3 比较分析

表 2.1 显示，经典的黑箱变分推断基于标准正态分布，因此也只能准确估计无相关（正态）分布的参数。这种方法不仅不能估计一般分布的参数，而且也无法估计相关（正态）分布的参数。基于拉普拉斯分布的黑箱变分的应用范围和经典方法类似，也无法用于相关分布和一般分布。这两种方法唯一的优点是可以应用于高维模型。基于相关正态分布的黑箱变分可以估计相关的（正态）分布，但无法用于一般分布和高维模型。基于参数变换的方法适用范围与经典方法类似，也无法应用于一般分布。根据目前的研究，黑箱变分推断方法从原则上无法准确估计一般多维概率模型的参数。

表 2.1 黑箱变分推断及其改进方法的适用范围

	标准正态分布（BBVI）	拉普拉斯分布	相关正态分布	参数变换
无相关分布	是	是	是	-
相关分布	否	否	是	-
一般分布	否	否	否	否
高维	是	是	否	-

2.4 本章小结

本章从信息论的角度讨论了黑箱变分推断的涵义。将拟合分布替换成拉普拉斯分布和相关的多维正态分布，定性地演示了改进方案的参数估计效果。研究参数可转换为正态类型的黑箱变分推断。黑箱变分推断法可以作为在无法应用MCMC 算法之后的备选方案，优点是能够适用于高维大数据模型，缺点是获得的参数信息较为有限。

3 总能量不恒定的抽样算法

由于黑箱变分推断无法估计概率模型的全部信息,因此后续研究基于 MCMC 的估计算法。目前大部分抽样算法是在空间或相空间上随机游走,这些算法未考虑总能量守恒。本章将回顾这些已有算法,并分析其局限性。第四章和第五章将介绍解决这些问题的一种方法。

3.1 马尔科夫链蒙特卡洛

MCMC 算法源于物理领域研究多粒子系统的宏观性质的需要。此时系统的状态由所有粒子的位置 (x_1, x_2, \dots, x_n) 决定,其中 x_i 为第 i 个粒子的位置。研究的目的是宏观属性,而这些属性可以通过状态函数的加权平均(即期望)进行计算。权重 $e^{-U(x)/kT}$ 中的 k 为玻尔兹曼常数,而 T 是温度。例如,一个简单的液体模型具有式(3.1)所示的能量形式,其中 V 是与粒子之间的距离有关的势能。

$$U(x) = \frac{1}{2} \sum_{i \neq j} V(|x_i - x_j|) \quad (3.1)$$

Metropolis 等人提出一种算法,每次按均匀分布随机移动粒子,然后按一个概率决定接受还是拒绝该移动。按照此描述,该算法有两个隐含的特点:(1)在两个点之间的相互移动概率相同;(2)移动不保持系统总能量守恒。等移动概率的依据为详细平衡方程,使得该算法在理论上确保能够抽样出表示概率的样本。然而,一个潜在的问题是,不保持系统能量守恒可能引起能量发散或者能量消失,从而导致粒子发散或陷入局部空间。

算法的目的是依据从物理模型中抽象出的势能 $U(x)$ 抽样出代表 $e^{-U(x)}$ 的样本。其中的 x 在原来的物理模型中表示所有粒子。在应用到概率模型时, x 更像是一个单独的粒子,即概率模型实际上基于的是单粒子系统。在基于随机游走的模型中,粒子能进行局部运动,但高维模型中的粒子进行有效移动则需要借助于能量导数。

3.2 Metropolis调整的郎之万算法

Metropolis 调整的郎之万算法(Metropolis-Adjusted Langevin Algorithm, MALA)以及混合蒙特卡洛是两种基于导数的抽样算法,其理论基础为郎之万动力学。首先简要回顾一下其理论来源,随后介绍相关的抽样算法。

郎之万动力学描述单个粒子的运动^[63]。一个相对较大的粒子(如花粉)受到周围大量微观粒子(分子或原子)的持续撞击会发生移动。根据牛顿第二定律,粒子所受合力等于动量对时间的变化率。根据斯托克斯公式,粒子受到的摩擦力

与速度成正比。考虑粒子受到的势能力以及周围微观粒子碰撞作用产生的力，于是得到郎之万公式(3.2)，其中的 ξ 是摩擦系数， v 是速度， U 是势能， F 为周围粒子碰撞产生的作用力， m 为粒子质量。这个公式的形式与哈密顿公式的区别在于存在碰撞和损耗，因此粒子的总能量不恒定。总能量不恒定的（单粒子）系统的优点是探索能力强，缺点是稳定性差。

$$\begin{aligned}\dot{x} &= v \\ m\dot{v} &= -U_x(x) - \xi v + F\end{aligned}\quad (3.2)$$

由于碰撞力 F 没有确定和连续的形式，导致上述微分方程没有解析解，也无法通过积分的方式获得 $x(t)$ 。根据时间和尺度的关系可得到式(3.3)，其中 η 表示碰撞强度。

$$Fdt \sim \eta \times N(0, \sqrt{dt}) \quad (3.3)$$

于是得到式(3.4)所示的欧拉积分的实现形式，其中 ε 为标准正态分布随机变量， δ 为积分的时间步长，即时间区间除以离散点数目。

$$\begin{aligned}x_1 &= x_0 + \delta \times v_0 \\ v_1 &= v_0 + \frac{\delta}{m} \times \left(-U_x(x_0) - \xi \times v_0 + \eta \sqrt{\delta} \varepsilon \right)\end{aligned}\quad (3.4)$$

从 x_0 和 v_0 开始，经过一系列迭代，到达 x_T 和 v_T ，因此需要设置初始速度 v_0 。为了应用 Metropolis-Hastings 算法，我们希望计算 x_0 和 x_T 之间的相互跳转概率，即 $P(x_0 \rightarrow x_T)$ 和 $P(x_T \rightarrow x_0)$ ，或者证明二者相等。然而，在式(3.4)的每一步都存在能量损耗和随机游走，尚不知道跳转概率的计算方式，因而难以实现抽样算法。此外，这里采用的物理动能在原则上也无法实现准确抽样。

经过若干简化，MALA 算法采用类似于式(3.4)所示的位置更新公式。该公式说明，从当前位置开始，以梯度下降后的位置为中心，进行一步随机游走，到达新位置。

$$x_1 \sim N(x_0 - \kappa U_x(x_0), \sigma) \quad (3.5)$$

由于上述的建议分布非对称，因此需要使用 Metropolis-Hastings 算法的接受概率（式 3.6），其中 $P(x_0 \rightarrow x_1)$ 为式(3.5)对应的概率密度函数。因此 MALA 算法相当于每次只执行了一步梯度下降，随之进行抽样。

$$\min \left(1, e^{U(x_0) - U(x_1)} \frac{P(x_1 \rightarrow x_0)}{P(x_0 \rightarrow x_1)} \right) \quad (3.6)$$

MALA 算法存在与哈密顿蒙特卡洛同样的问题，即运行参数、稳定性和准确性问题。算法没有保持能量守恒，存在发散的可能。根据本文随后论述可知，抽

样准确性主要取决于动能函数，而物理动能产生的梯度下降方向一般无法实现准确抽样。

3.3 基于哈密顿动力学的算法

图 3.1 显示从 q_0 位置开始，以 p_0 为动量，经过哈密顿仿真到达 q_1 位置，以及相对应的动量 p_1 ，目的是获得概率分布 $e^{-U(q)}$ 的抽样样本，即一组样本值 q_i 。



图 3.1 哈密顿仿真示意图

式(3.7)显示，总能量 H 为动能 K 和势能 U 之和。势能仅为位置 q 的函数。物理动能仅是动量 p 的二次函数，而本文讨论的算法使用的动能也同时是位置的函数。因此，本文所讨论的动能推广了真实的物理动能。

$$H(q, p) = K(p, q) + U(q) \quad (3.7)$$

式(3.8)显示哈密顿动力学公式，下标 p 或者 q 表示能量相对于 p 或者 q 求偏导。

$$\begin{aligned} \dot{p} &= -H_q \\ \dot{q} &= H_p \end{aligned} \quad (3.8)$$

式(3.9)显示代入总能量 H 后的形式。

$$\begin{aligned} \dot{p} &= -U_q - K_q \\ \dot{q} &= K_p \end{aligned} \quad (3.9)$$

因此实现哈密顿仿真算法需要计算三个能量导数。根据随后的讨论可知， K_q 需要计算三阶导数，计算量最大。一个容易忽略的事实是，现有研究经常忽略了动能也是位置的函数，从而忽略了 K_q 。有些学者虽然注意到这一项^[65]，但却没有直接在仿真算法中实现。

式(3.10)显示，如果对总能量相对于时间求导，并带入哈密顿公式，可以得到常数零。

$$\dot{H} = H_q \cdot \dot{q} + H_p \cdot \dot{p} = H_q \cdot H_p - H_p \cdot H_q = 0 \quad (3.10)$$

这说明如果粒子根据哈密顿动力学进行运动，则其在运动轨迹上的总能量恒定，也就是说仿真轨迹两端点的总能量相同。

“Hybrid Monte Carlo” 是一篇与这个问题有关的早期文献^[12]。这篇文献提出从系统的微观状态 (q_0, p_0) 到 (q_1, p_1) 的接受概率，如式(3.11)所示，其中 H 为哈密顿量，即总能量。一种称为哈密顿蒙特卡洛方法与上述方法类似，采用同样的

接受概率。

$$P(q_1, p_1) = \min(1, e^{H(q_0, p_0) - H(q_1, p_1)}) \quad (3.11)$$

算法 3.1 为经典的哈密顿蒙特卡洛算法。第 2 行初始化动量。第 3 行计算初始总能量。从第 5 行到第 8 行进行哈密顿仿真。第 9 行计算仿真后的总能量。第 10 行为接受概率。第 11 行到第 14 行的意图是采用 Metropolis-Hastings 算法。由于仿真误差的原因， H_0 一般不等于 H 。由于初始化动量的时候没有考虑总能量恒定，该算法在运行中可能出现动能为零或者很大的现象，分别意味着仿真停止和发散。

算法 3.1: 经典的哈密顿蒙特卡洛

输入：初始位置 q

```
1:  for  $i=1$  to  $n$ 
2:       $p \sim N(0,1)$ 
3:       $H_0 = U(q) + 0.5p^T p$ 
4:       $q_0 = q$ 
5:      for  $s = 1$  to  $S$ 
6:           $p = p - \delta U_q$ 
7:           $q = q + \delta p$ 
8:      end
9:       $H = U(q) + 0.5p^T p$ 
10:      $\alpha = e^{H_0 - H}$ 
11:      $u \sim U(0, 1)$ 
12:     if  $\alpha < u$ 
13:          $q = q_0$ 
14:     end
15:     save  $q$ 
16: end
```

输出：一系列保存的位置 q

简单起见，只考虑理想的哈密顿轨迹，即仿真过程中没有能量损耗。如果两个相空间端点位于同一条哈密顿轨迹上，则这两个位置的总能量在理论上相同，因此这个接受概率为常数 1，即总是接受新样本。此时仿真结果与初始总能量有关。初始总能量大，则样本覆盖范围就大。如果两个相空间端点的总能量不相同，则二者不可能位于同一条哈密顿轨迹上，能量差只能来源于外部。如果考虑到仿真存在数值误差，则仿真轨迹端点的总能量可以不相同。但此时抽样分布的来源是数值误差，而不是物理系统本身。

顺便需要指出的是，该方法默认假设相空间的两个空间状态之间的相互跳转概率相同（式 3.12）。但是如果系统总能量不恒定，似乎不应再假定微观状态等

概率出现。

$$P(q_0, p_0 \rightarrow q_1, p_1) = P(q_1, p_1 \rightarrow q_0, p_0) \quad (3.12)$$

混合哈密顿蒙特卡洛研究的是整个系统所有粒子的状态。根据接受概率，该方法假设系统的总能量是可变的。然而根据前文的讨论可知，哈密顿仿真保持总能量不变，因此系统总能量变化的原因只可能来自与外部系统之间的能量交换。因此这种方法的模型应当是具有恒定总能量的多个系统，但算法并没有明确涉及多个系统。

这种方法主要用来抽样概率模型的参数 q ，因此忽略了协变量 p 。接受概率在理论上为常数 1，相当于将仿真的终点位置直接作为新样本。哈密顿蒙特卡洛没有明确提及使用多粒子系统，而认为粒子能量变化来源于仿真误差。如果算法没有产生仿真误差，则抽样算法会接受所有候选样本。因此该算法在理论上可能存在问题。

RMHMC 采取的方案是在仿真过程中使用势能的海森矩阵修改动量的方向，因此形成了另一种仿真轨迹。它与哈密顿蒙特卡洛的区别在于动能函数，以及不同类型的仿真轨迹。仿真轨迹能够快速遍历第一主成分方向，因此修正了 HMC 的缺陷。然而，此时又产生了隐含的缺陷，即在其它主成分（尤其是最后几个主成分）的抽样误差较大。RMHMC 采用式(3.14)形式的总动能。该动能形式显然受到多维正态分布的启发。其中第一项与多维正态分布的归一化项有关。由于动能函数不需要进行归一化或正定化，因此动能的第一项实际上是不需要的。其实我们的目的是从标准多维正态分布中抽样 p ，然后设置 p 的幅值，使其动能符合预期的值。因此可以仅使用第二项。

$$K(p, q) = \frac{1}{2} \log(2\pi)^D |M| + \frac{1}{2} p^T M^{-1} p \quad (3.13)$$

RMHMC 的第二个值得注意的问题是矩阵 M 可以是 q 的函数，因此在算法中需要计算动能相对于 q 的导数。如果 M 是势能相对于 q 的二阶导数（即海森常矩阵），且为常数，则 K_q 为零。如果 M 是海森矩阵的函数，则计算 K_q 时需要推导相应的公式。不考虑 K_q 的算法会引入仿真和抽样误差。

HMC 和 RMHMC 需要一个仿真停止准则，从而获得候选样本点。NUTS 为 HMC 和 RMHMC 提供了一个自动调节仿真步数的方案。其基本思想是将空间回转作为停止准则。图 3.2 中橘色点所连接的曲线为仿真在 x_1 - x_2 空间的轨迹。NUTS 试图在仿真的开始初始化阶段寻找一个合适的步长，然后在随后的每次仿真都检测空间回转点，如图中黑色点所示。由于（单个粒子）的仿真对移动的路径长度非常敏感，为了获得更精确的检测，可以采用较小的步长和较多的步数来实现，但这又以增加计算量作为代价。

本人提出使用仿真轨迹上势能的最大值和最小值作为停止准则。图 3.2 中蓝色点为势能-空间联合轨迹。红色点为势能最大或者最小值点。这个轨迹只有一个空间回转点，但三个势能极值。根据势能的最大值和最小值可以区分过大和过小的步长参数。空间回转只有一个标准，无法实现上述区分。由于能量准则提供的信息多于空间回转准则，因此空间回转准则可能要弱于势能极值准则。此外，如果算法能采用遵守能量守恒的多粒子系统，根据所有粒子的状态统计量来调节参数，调节的稳定性会更高。

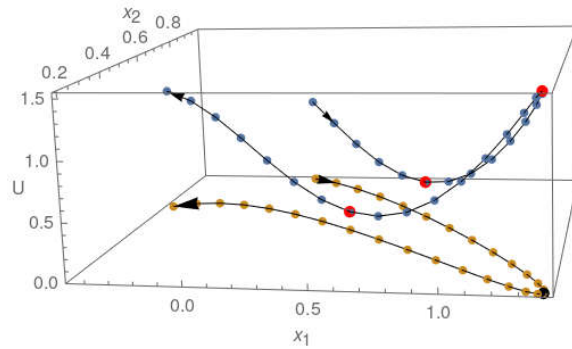


图 3.2 橘色点曲线：空间轨迹，蓝色点曲线：空间-势能联合轨迹，黑点：空间回转点，红色：势能极值，箭头：仿真方向

3.4 本章小结

本章回顾了一些经典的抽样算法。在一定程度上，这些算法都可以看作是总能量不恒定的单粒子系统。经典的马尔科夫链蒙特卡洛对整个系统的位置状态进行抽样，因此也可看作单粒子算法。这些算法存在的问题在于，固定单粒子系统的总能量会严重制约粒子的探索能力。虽然无需遵守能量守恒使得算法无需设置总能量从而简化实现方案，然而这种简化会产生稳定性和准确性问题。这些存在的问题难以在现有的系统框架内得到完全的解决。

4 总能量恒定的系统

第3章分析了一些经典算法（MCMC、MALA、HMC、RMHMC 和 NUTS）存在的缺陷。为了解决这些方法存在的问题，有必要采用多粒子系统和改进的动能。

对于基于哈密顿动力学的参数估计方法，算法的稳定性取决于系统总能量是否恒定，而根据本文提出的总能量恒定的多粒子系统模型可以实现稳定的抽样算法。本章首先研究具有恒定总能量的单粒子系统，然后推广到多粒子系统，并提出其实现方式。哈密顿仿真轨迹的形态对抽样的准确性有重要的影响，而轨迹主要决定于动能。通过比较仿真和优化算法，揭示出轨迹形态和动能函数之间的对应关系。

4.1 单粒子系统

虽然哈密顿动力学适用于一般的总能量，但不是任意的势能和动能的组合都能构成稳定的哈密顿动力系统。例如正的二次动能和负的二次势能组合有可能导致发散。为了获得稳定的哈密顿仿真，有必要确定动能和势能之间的对应关系。参数估计算法需要根据势能函数进行抽样，无法变更势能函数，只能改变动能函数。因此可以根据势能寻找合适形式的动能，使得仿真轨迹有利于抽样。本节先以正的动能和势能为例进行定性分析，随后再揭示动能、势能和仿真轨迹之间的对应关系，探究有利于抽样的仿真轨迹。

考虑单个粒子组成的总能量恒定的系统，粒子在空间中随机游走，定期改变运动方向。可以假定仿真采用的是二次能量，即式(4.1)及式(4.2)。

$$U(x_1, x_2) = \frac{1}{2} (x_1, x_2) \begin{pmatrix} 1 & 0.7 \\ 0.7 & 1 \end{pmatrix}^{-1} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad (4.1)$$

$$K_0(p, q) = \frac{1}{2} p^\top p \quad (4.2)$$

图 4.1 显示从同一个位置开始，具有相同总能量的多条可选择的仿真轨迹，其中 x_1 和 x_2 为两个空间维度。由于确定了初始位置，初始势能也就确定了，从而确定了粒子的初始动能。这些轨迹的初始动量的幅值相同，动量方向为等间隔角度。仿真终点连成一个封闭曲面。对于二维空间，终点位于封闭曲线之上。这种恒定总能量的哈密顿仿真相当于在初始动量方位角上以均匀分布进行随机游走，类似于马尔科夫链蒙特卡洛采用的在空间中按均匀分布进行的随机游走。这里讨论的是总能量恒定的单粒子系统。目前绝大多数算法都是基于总能量不恒定的单粒子系统，与这里讨论的不同。虽然原则上可以在方位角上进行随机游走，

并应用 Metropolis 算法，从而实现马尔科夫链蒙特卡洛算法，但其实际抽样效果

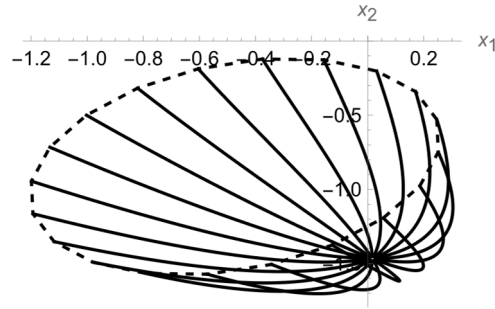


图 4.1 实线：从同一个初始位置开始的具有相同总能量的多条仿真轨迹，
虚线：仿真终点连成的封闭曲线

并不理想。总能量恒定的单粒子系统虽然无法实现有效的抽样算法，但这种模型可以作为理解多粒子模型的基础。

图 4.2 显示两个具有相同总能量的位置之间的相互仿真轨迹。从 q_0 出发，取任意一个具有指定动能的动量，在相同仿真时间内，所有可能仿真轨迹终点形成一个空间曲面（蓝色曲线），该曲面上的任意位置（如 q_1 ）都具有与 q_0 相同的总能量。因为 q_1 具有相同总能量，根据其势能可以计算出动能。从 q_1 出发，取任意一个具有指定动能的动量开始仿真，所有可能的新的终点也形成一个曲面（橘色曲线），而 q_0 正位于此曲面上。此外，另一个显著的特点是 q_0 和 q_1 之间的相互仿真轨迹重合。这说明，在相同的仿真时间内，如果从 q_0 开始以 p_0 作为动量进行仿真到达 q_1 ，终点动量为 p_1 ，那么存在一个反向的仿真从 q_1 开始，以 $-p_1$ 为初始动量，到达 q_0 ，并且终点动量为 $-p_0$ 。

为了应用 Metropolis 算法，需要确保 q_0 和 q_1 之间的相互跳转概率相同。以二维空间为例，如果从 q_0 位置以 p_0 为初始动量进行仿真，到达终点 q_1 ，则 q_1 可以根据算法 4.1 获得。

如果 p_0 的方向是从区间 $[0, 2\pi]$ 中均匀选取，考虑到 p_0 和 q_1 之间存在一一映

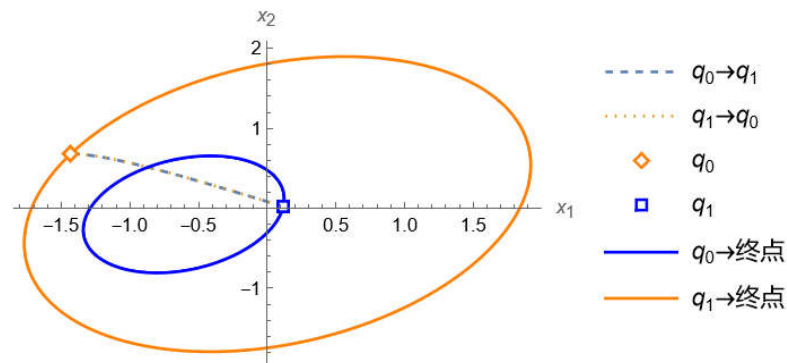


图 4.2 两个位置之间的 K_0 仿真轨迹以及由终点位置，蓝色曲线为从 q_0 开始仿真的终点组成，
而橘色曲线为从 q_1 开始仿真的终点组成，虚线为两个位置之间的相互仿真轨迹

射，则从 q_0 到 q_1 的跳转概率服从均匀分布。类似的，从 q_1 到 q_0 的跳转概率也服从相同区间的均匀分布。综上所述， q_0 和 q_1 之间的相互跳转概率相同，从而可以应用 Metropolis 算法进行抽样。

算法 4.1： 获得 q_1 的步骤

输入： 初始位置 q_0 。

- 1: 为初始动量 p_0 任选一个角度；
- 2: 根据期望的动能设置 p_0 的幅度；
- 3: 从初始位置 q_0 以初始动量 p_0 开始仿真；
- 4: 过了一定时间，仿真在 q_1 位置结束。

输出： q_1 。

上述讨论没有限制动能的形式，原则上可以适用于任意的动能函数，例如与黎曼流形哈密顿蒙特卡洛采用的动能近似的函数，如式(4.3)所示。

$$K_1(p, q) = \frac{1}{2} p^T U_{qq}^{-1} p \quad (4.3)$$

图 4.3 显示采用 K_1 动能的仿真示意图。图 4.2 与图 4.3 的区别主要来自于动能函数：基于 K_0 动能的仿真轨迹主要沿着概率分布的最后主成分方向移动，而 K_1 动能生成的仿真轨迹可以沿着第一主成分方向移动，仿真轨迹的不同特点会影响算法在各个主成分上的抽样准确度。

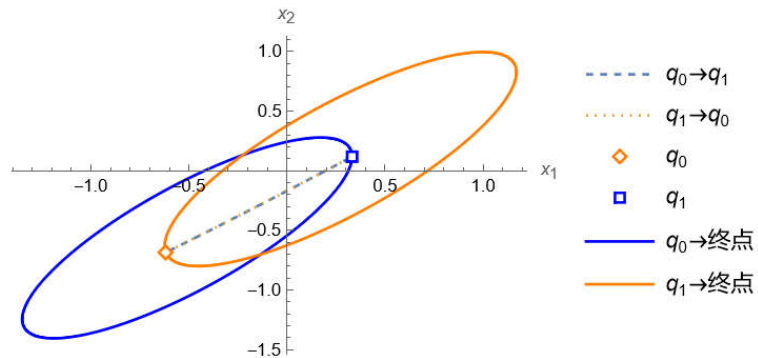


图 4.3 两个位置之间的 K_1 仿真轨迹以及终点，蓝色曲线为从 q_0 开始仿真的终点，而橘色曲线为从 q_1 开始仿真的终点，虚线为两个位置之间的仿真轨迹。

对于单粒子系统，遵守能量守恒，一方面使得仿真保持稳定，另一方面也会

限制粒子的探索能力。如果质点的总能量过高，则其在低势能（即高概率）区域停留的时间较短。反之，如果总能量过低，则可能无法访问到高势能（即低概率）区域。此外，算法需要自动设置总能量和时间步长参数，但是依赖于单一粒子的运行历史进行设置容易导致不稳定的结果。对于单粒子系统，探索能力和稳定性之间存在着难以调和的矛盾。保持恒定的总能量会严重制约探索能力，但是不恒定的总能量又会导致稳定性较差。经典算法采用的是总能量可变的方式，虽然算法实现比较简单，但会给稳定性和准确性带来一系列后续问题。

经典的哈密顿蒙特卡洛和黎曼流形哈密顿蒙特卡洛算法采取的方案是总能量不守恒的单粒子仿真，因此仿真轨迹可以遍历高势能区域和低势能区域。如果固定粒子的总能量，则难以兼顾对不同区域的访问。此外，单粒子仿真相当于单一马氏链，据此调节运行参数稳定性较差。

4.2 多粒子系统

多粒子系统是统计力学中的一个理想模型。从这种模型得出的一些结论可以推广到一般的多粒子系统，从而作为抽样算法的基础。假设三维空间中的一个封闭容器中有 N 个高速运动的粒子，粒子之间可以通过相互碰撞而交换能量。先暂时不考虑重力势能的作用，因为其结论可以推广到具有势能的一般系统。 N 个粒子的动量总共组成一个 $3 \times N$ 维向量。由于碰撞后粒子等概率向各个方向运动，因此 $3 \times N$ 维初始动量的方向等概率，而动量幅度由总动能决定，因此 $3 \times N$ 维动量均匀分布于一个超球面上，在物理学中一般称为 $3 \times N$ 动量超球。由于零均值同方差的正态分布任意旋转后仍然保持形式不变，因此动量的每一维都服从同方差零均值的正态分布^[66]。

式(4.4)所示的麦克斯韦-玻尔兹曼分布是上述思想应用于物理系统的一个实例。球面坐标系和欧氏坐标系微元之间有如下关系，其中左边的 p_x , p_y 和 p_z 为动量的空间分量，右边的 p 为动量幅值。

$$dp_x dp_y dp_z = p^2 \sin(\phi) dp d\theta d\phi \quad (4.4)$$

式(4.4)左边为动量的概率微分，将其代入式(4.5)，可以得到右边的公式。

$$\frac{e^{-\frac{p_x^2}{2\sigma^2} - \frac{p_y^2}{2\sigma^2} - \frac{p_z^2}{2\sigma^2}}}{2\sqrt{2\pi}^{3/2} \sigma^3} dp_x dp_y dp_z = \frac{p^2 \sin(\phi) e^{-\frac{p^2}{2\sigma^2}}}{2\sqrt{2\pi}^{3/2} \sigma^3} dp d\phi d\theta \quad (4.5)$$

对 ϕ 和 θ 分别在 $[0, \pi]$ 和 $[0, 2\pi]$ 范围内积分，可以得到式(4.6)所示的动量幅值分布的概率密度函数，即麦克斯韦-玻尔兹曼分布。

$$\frac{\sqrt{\frac{2}{\pi}} p^2 e^{-\frac{p^2}{2\sigma^2}}}{\sigma^3} dp \quad (4.6)$$

麦克斯韦-玻尔兹曼分布的核心思想是初始动量的方向等概率。初始动量方向等概率的原则可以推广到具有物理势能的多粒子系统。单个粒子的微观状态由位置和动量组成，并假设粒子可以因为碰撞而改变运动方向和速度大小。粒子的初始动量的幅值决定于初始动能，无法任意改变，从而只能改变初始动量的方向。由于每个粒子的初始动量的方向都是等可能的，所有粒子的总动量的方向也是等可能的。因为总能量恒定，在系统中各个粒子的初始空间位置确定后，系统总势能也就确定了，从而系统初始总动能也就随着确定。系统所有可能的 $3 \times N$ 维初始动量都位于一个球面上，球面的半径由总动能决定，从而动量超球的半径也就确定了。动量超球球面上所有点都对应于具有相同的系统总能量的空间状态，即球面上的每个点对应系统的空间位置状态。统计力学假设等能量的状态等概率，相当于这些球面上的点被选择的概率相等。因此系统从一个 $3 \times N$ 维空间状态转向任意后续 $3 \times N$ 空间状态的概率相同，从而系统在任意两个空间状态之间的相互跳转概率相同。

上述物理空间的结论可以应用于一般系统。多粒子系统中的粒子之间可以相互碰撞并交换能量，也就是说每个粒子的总能量可以变化。因为幅值由动能决定，可以暂时不考虑动量幅值的影响。动量的方向等概率，因此每维动量都相当于服从标准正态分布。一个 D 维粒子的动量服从 D 维标准正态分布， N 个粒子的动量服从 $N \times D$ 维标准正态分布。因此 N 个粒子的初始动量从一个 $N \times D - 1$ 维超球面上均匀选取。由于初始动量和仿真结束位置具有一一映射关系，因此 N 个粒子的各个仿真结束位置($N \times D$ 维随机变量)被选取的概率相等，即根据初始位置得到的仿真结束位置服从均匀分布。因此将 N 个粒子作为整体的仿真初始和结束位置相互跳转概率相同，即式(4.7)。

$$P(q_{1,0}, \dots, q_{N,0} \rightarrow q_{1,1}, \dots, q_{N,1}) = P(q_{1,1}, \dots, q_{N,1} \rightarrow q_{1,0}, \dots, q_{N,0}) \quad (4.7)$$

由于粒子之间仅仅通过系统总能量进行关联，而系统总能量仅为一个标量，因此可以假设各个粒子之间几乎没有关联，于是得到式(4.8)。

$$\prod_{i=1}^N P(q_{i,0} \rightarrow q_{i,1}) \approx \prod_{i=1}^N P(q_{i,1} \rightarrow q_{i,0}) \quad (4.8)$$

使上述等式成立的一组最简单的条件是每个粒子在仿真轨迹两端的相互跳转概率相同，即式(4.9)。

$$P(q_{i,0} \rightarrow q_{i,1}) \approx P(q_{i,1} \rightarrow q_{i,0}) \quad (4.9)$$

上述推论的直观物理意义是，总能量恒定的具有碰撞作用的多粒子系统中的任一个粒子在空间轨迹上的两个端点在跳转概率上没有区别。如果粒子能够从甲位置按某概率移动到乙位置，那么粒子也可以以相同的概率从乙位置移动到甲位置。这个性质与能量恒定的单粒子系统相同，区别在于此时粒子的总能量可变，从而大幅增强了粒子的探索能力。

因此整个系统在仿真前后的两个 $N \times D$ 维空间位置之间相互跳转概率相同，最可能的原因（或结果）是每个粒子在仿真前后的两个空间位置之间具有相同的相互跳转概率。这说明，如果系统具有碰撞作用，单个粒子在不同位置的总能量一般不同，但在两个位置之间的相互跳转概率相同。因此每个粒子都可以应用 Metropolis 算法（式 4.10），即按照概率决定是否接受新样本。

$$\alpha = \min(1, e^{U(q_{i,0}) - U(q_{i,1})}) \quad (4.10)$$

动能与动量的幅值的平方呈线性关系，因此可以在算法中通过设置动量幅值来改变动能，从而使总能量保持恒定。式(4.11)显示，对于物理动能 K_0 ，粒子初始动量幅值采用欧氏范数度量。

$$\|p\| = \sqrt{p^\top p} = \sqrt{2K_0} \quad (4.11)$$

此时具有指定动能的动量均匀分布于一个超球面上。初始动量与仿真轨迹末端空间位置一一对应相当于在给定初始空间位置的前提下，均匀选取仿真轨迹的末端空间位置。因此，在保持能量守恒的前提下，新空间位置等概率。

哈密顿蒙特卡洛和黎曼流形哈密顿蒙特卡洛的主要区别在于采用了不同的动能函数。二者的动能函数可以统一定义为式(4.12)，其中 r 为一个参数，而不是求导变量。为简洁起见，本文随后以 K_r 表示此动能。因此，在本文中，只有当能量的下标为 p 和 q 时才表示对能量求导。

$$K(p, q; r) = \frac{1}{2} p^\top U_{qq}^{-r} p \quad (4.12)$$

对于一般的动能 K_r ，动量幅值采用马氏范数，即式(4.13)。

$$\|p\| = \sqrt{p^\top U_{qq}^{-r} p} = \sqrt{2K_r} \quad (4.13)$$

此时具有指定动能的动量分布于一个超椭球面上，动量的方向仍然服从均匀分布。由于椭球面上任意动量的幅值相等，方向等概率，因此动量仍然相当于均匀分布在一个超球面上，从而仿真轨迹的末端空间位置等概率。实质上与 K_0 的情况相同。引入一般动能不会改变上述基于物理动能 K_0 讨论得到的结论，改变的仅仅是仿真轨迹的形态。

多粒子系统中的粒子由于存在能量交换，粒子动能分布在一个较大的范围内。粒子的初始动能的分布范围较广，最小值接近于零，最大值可以为一个很大的数

值。粒子可以移动到附近或者较远的位置，从而有效遍历空间区域。

为了平衡稳定性和探索能力，本人提出的方案是总能量守恒的多粒子系统，粒子之间通过完全弹性碰撞交换动能^[67]。引入多粒子系统的主要目的是使单个粒子的总能量在一定范围内可变。假设封闭系统由若干个初始能量各不相同的粒子组成。这些粒子之间可能会发生完全弹性碰撞而相互交换能量。系统的总能量守恒，即所有粒子总能量之和保持不变，能够保证系统整体稳定。每个粒子的总能量在一定范围内变化，从而具有较强的探索能力。

实现一个多粒子系统的直接方法是仿照实际的物理系统，在仿真过程中实现碰撞检测与反弹算法。实际的碰撞算法存在的一个问题是其运算量与粒子数的平方成正比，因此难以应用于具有大量粒子的系统。另一个问题是，为了实现准确的碰撞算法，需要使用较小的时间步长，从而增加计算量。真实物理系统的粒子碰撞的频率很高，直接实现碰撞需要较多的计算量。最后一个问题是，在高维空间能否实现准确的碰撞检测？由于高维空间比三维空间稀疏得多，碰撞检测算法有可能具有较大的误差。

实际上，对于多粒子系统，碰撞的本质是交换能量，因此可以无需考虑碰撞的实际物理过程，而直接交换粒子的能量。其核心思想是确保初始动量方位角等概率，这样就可以确保仿真前后两个位置之间的相互跳转概率相同。

算法 4.2 显示多粒子系统的初始动量生成过程。第 4 到第 6 行计算系统总势能和总动能。第 4 行初始化动量，保证初始动量方向等概率，意味着系统中的所有粒子开始交换能量。第 5 行累计总动能。第 6 行累计总势能。算法没有指定具体的动能形式，因此原则上适用于任意类型的动能。第 10 行设置每个粒子的动量幅值，以保证系统总能量守恒。绝对值运算考虑到势能函数有可能具有负定海森矩阵，从而导致负的总动能。

算法 4.2：系统能量守恒及生成初始动量

输入：M 个粒子的初始位置 q_j 和系统总能量 H 。

```
1:   $U_{total}=0$ 
2:   $K_{total}=0$ 
   /* M 个粒子 */
3:  for  $j=1$  to M
   |   /* 动量方向等概率 */
4:   |    $p_j \sim N(0, 1)$ 
5:   |    $K_{total} = K_{total} + K(p_j, q_j)$ 
6:   |    $U_{total} = U_{total} + U(q_j)$ 
```

```

7:  end
8:   $K_d = H - U_{\text{total}}$ 
   /* 系统总能量守恒 */
9:  for  $j=1$  to M
10: |   $p_i = p_i \sqrt{K_d / K_{\text{total}}}$ 
11: end
输出: M 个粒子的初始动量  $p_j$ 

```

4.3 主成分方向

根据初始值和仿真算法，可以得到哈密顿仿真轨迹。哈密顿仿真轨迹可以用二次势能函数予以说明。图 4.4 中的势能函数对应于相关系数为 0.7 的二维正态分布，其中 x_1 和 x_2 为两个空间维度。图中蓝色散点显示样本分布。蓝色曲线显示哈密顿仿真轨迹优先在最后主成分方向运动。如果各个主成分差别较大，轨迹将主要遍历最后主成分方向。这个特点将会对高维、高相关和非正态分布的抽样

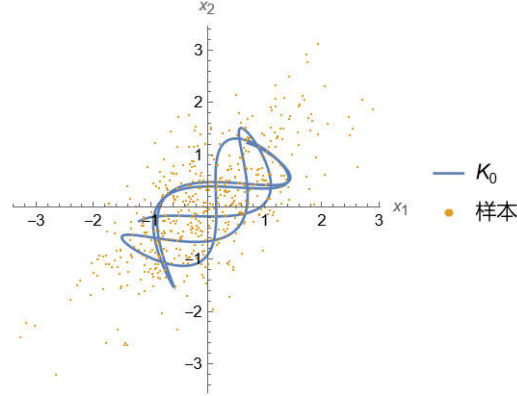


图 4.4 哈密顿蒙特卡洛的仿真轨迹

带来较为不利的影响。

一般的概率模型可以用多维概率密度函数或者与之对应的多维势能函数进行描述。势能函数在每个位置的海森矩阵的逆的特征值和特征向量定义了空间覆盖范围及其方向。对于多维正态分布，海森矩阵的逆为协方差矩阵，其特征值为主成分的方差，特征向量为该主成分的方向。可以用一个二维正态分布的例子对这些性质进行说明。一个二维正态分布对应的势能函数可以定义为式(4.14)。

$$U(x_1, x_2) = \frac{1}{2} (x_1, x_2) \begin{pmatrix} 1 & 0.7 \\ 0.7 & 1 \end{pmatrix}^{-1} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad (4.14)$$

其中 x_1 和 x_2 为两个空间维度。对势能的海森矩阵（即二阶导函数矩阵）求逆，可以得到协方差矩阵（式 4.15），其中 x 为 x_1 和 x_2 组成的向量。

$$U_{xx}^{-1} = \begin{pmatrix} 1 & 0.7 \\ 0.7 & 1 \end{pmatrix} \quad (4.15)$$

多维正态分布的协方差矩阵为正定，即特征值为正。正定的海森矩阵会产生正势能。但一般概率密度函数对应的势能函数的海森矩阵可能为非正定或者随位置发生改变。

主成分的尺度用海森矩阵的逆的特征值的平方根表示，相当于标准差。标准差最大的称为第一主成分， D 维正态分布有 D 个主成分分量。多维正态分布的海森矩阵为常矩阵，在任意位置具有相同的主成分方向和大小，因此主成分可以

描述正态分布的整体性质。

对于一般的多维概率密度函数，对应的势能函数的海森矩阵不再是常量，会随着位置的改变而变化。每一空间位置都可以定义一个局部坐标系，可以描述势能在当前位置的空间覆盖范围。式(4.16)显示一个环形分布对应的势能函数，其海森矩阵随位置变化，可以呈现正定或者负定状态。

$$U(x_1, x_2) = \frac{\left(\sqrt{x_1^2 + x_2^2} - r\right)^2}{2\sigma^2} \quad (4.16)$$

环形分布在不同位置具有不同的局部坐标系及单位长度。由于海森矩阵可能为负定，因此需要使用特征值的绝对值的平方根作为坐标轴的尺度。尽管严格来讲，概率分布的主成分这一术语无法准确描述一般的多维概率分布，但采用主成分的术语并不影响随后进行的讨论。根据多维正态分布进行论述得到的结论也适用于一般的多维概率分布。因此不妨将局部坐标系中单位长度最大的类比于“第一主成分”，其它以此类推。

4.4 优化与仿真

仿真算法模拟的是总能量恒定的物理过程，优化则是与势能降低过程类似，二者有一定的近似关系。具体来说，就是哈密顿仿真轨迹类似于梯度下降优化轨迹，而黎曼流形哈密顿仿真轨迹与牛顿优化轨迹近似。揭示这些轨迹之间的联系有助于寻找适合抽样的仿真轨迹。

对于一个目标函数 $U(q)$ ，希望获得其最小值。由于算法仅使用一阶和二阶导数，因此可以用式(4.17)所示的二次函数来模拟目标函数。

$$U(q) = \frac{1}{2} (q - q_0)^T \Sigma^{-1} (q - q_0) \quad (4.17)$$

目标函数的梯度，即一阶导数为式(4.18)。

$$U_q = \Sigma^{-1} (q - q_0) \quad (4.18)$$

一种获得函数的最小值的方法是梯度下降法。如式(4.19)所示，梯度下降法沿着梯度方向移动，寻找一个标量 δ ，使每次得到的函数值在此方向上最小。

$$\Delta q = -\delta U_q \quad (4.19)$$

δ 决定移动的步长，可以通过精确直线搜索或者回溯直线搜索算法予以动态地确定。精确直线搜索是通过二分法寻找一个 δ ，使得函数 U 在新位置 $q + \Delta q$ 的值最小。回溯直线搜索是一种非精确的方法，其实际效果稍好于精确直线搜索。优化算法一般需要自适应学习步长参数。对于基于仿真的抽样算法，步长也是一个重要的运行参数。由于仿真效果对于步长参数较为敏感，因此也采用非精确的

调整方式。

如果将梯度公式(4.18)带入增量公式(4.19)，可知梯度下降法的移动方向与目标函数的二次型矩阵（或海森）有关，如式(4.20)所示。

$$\Delta q = -\delta \Sigma^{-1} (q - q_0) \quad (4.20)$$

牛顿法是一种使用二阶导数的改进算法，其特点是收敛速度快。假如用式(4.21)的二次函数来近似目标函数。

$$U(q + v) \approx U(q) + v^T U_q + \frac{1}{2} v^T U_{qq} v \quad (4.21)$$

那么针对等式两端相对于 v 求导数，并考虑到 $U(q+v)$ 和 $U(q)$ 为已知数值，导数为 0，于是得到 v 的计算公式(式 4.22)，以及 q 的增量公式(式 4.23)。

$$v = -U_{qq}^{-1} U_q \quad (4.22)$$

$$\Delta q = -\delta \Delta v = -\delta U_{qq}^{-1} U_q \quad (4.23)$$

这个增量公式相当于用海森矩阵动态改变梯度的方向。这个算法与梯度下降算法的唯一区别是使用牛顿公式(4.23)来计算位置的增量。如果采用二次目标函数，求目标函数的二次导数，得到式(4.24)描述的目标函数的海森矩阵。

$$U_{qq} = \Sigma^{-1} \quad (4.24)$$

代入到牛顿公式得到式(4.25)。

$$\Delta q = -\delta (q - q_0) \quad (4.25)$$

因此牛顿法是以直线的方式向最优位置移动，而与目标的海森矩阵无关。这说明牛顿法的收敛速度较快。基于牛顿法的优化算法主要计算量为海森矩阵和求解线性方程组。对于超高维势能函数，其计算量较大。对于中小规模的优化问题，牛顿方法的收敛速度远快于梯度下降算法。

根据以上讨论可知，梯度下降方法偏向于梯度方向，牛顿无偏向。也可以产生一个与梯度下降法对称的方法，即“反梯度下降”，获得另外一种偏向方向。假如采用式(4.26)的更新方向。这种方法在第一主成分方向的更新最快。此外，由于需要计算矩阵的平方，该算法的数值稳定性稍差。这个算法的计算量高于牛顿法，收敛慢于牛顿法，因此不属于实用的优化算法。

$$\Delta q = -\delta U_{qq}^{-2} U_q \quad (4.26)$$

为了比较几种优化算法产生的轨迹，考虑式(4.27)所示的势能函数，二维正态分布的第一主成分为 x_1 ，第二主成分为 x_2 。

$$U(x_1, x_2) = \frac{1}{2} \begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 0.1 \end{pmatrix}^{-1} \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{2} (x_1^2 + 10x_2^2) \quad (4.27)$$

从(0.01, 0.01)位置开始，使用三种更新公式，根据精确直线搜索来找到适合的步长 δ ，来实现优化算法。图 4.5 中显示三种优化轨迹。比较三种轨迹，可以发现轨迹的形态有显著区别。牛顿法相当于同时在所有主成分方向移动，收敛速度最快，相当于是梯度下降法和反梯度下降法的平衡。从初始位置出发，牛顿法以近乎直线的方式移动到最优解。由于一般的非线性函数可以用二次函数进行近似，因此牛顿法也可以应用于非线性函数，从而非常迅速地寻找到最优解。由于 x_2 方向的梯度远远大于 x_1 方向的梯度，因此梯度下降法主要沿着 x_2 方向移动，而在 x_1 方向移动非常缓慢。对于多维函数，如果不同维度的梯度值具有较大差别，则梯度下降法可能无法在有限的时间内寻找到最优解。反梯度下降优先在 x_1 方向上运动，在 x_2 方向反而移动较慢。在优化问题中，希望移动到最低势能位置，因此可以使用牛顿法。如果考虑的是仿真和抽样问题，则可能需要考虑不同

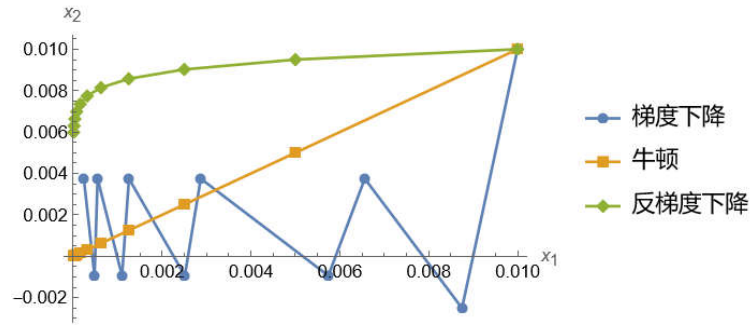


图 4.5 二次势能的优化轨迹

的轨迹以充分遍历空间。

仿照“反梯度下降算法”，也可以用式(4.28)定义“反哈密顿”仿真所使用的动能函数。

$$K_2(p, q) = \frac{1}{2} p^T U_{qq}^{-2} p \quad (4.28)$$

为了展示几种仿真轨迹的区别，使用同样的势能函数产生仿真轨迹，其中坐标轴 x_1 - x_2 为两个空间维度。图 4.6 中蓝色轨迹为哈密顿仿真，橘色轨迹为黎曼哈

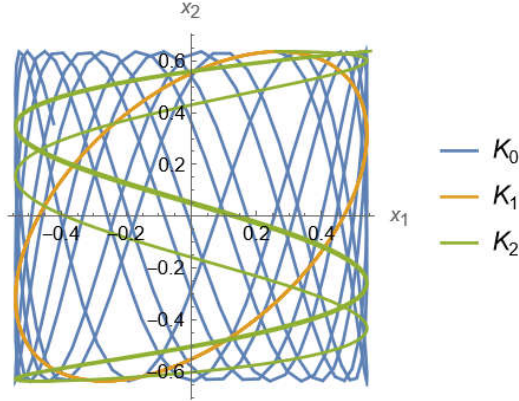


图 4.6 二次势能的仿真轨迹

密顿仿真，而绿色轨迹对应于所谓的“反哈密顿”轨迹。 K_0 优先沿着最后主成分方向移动， K_2 的轨迹优先向着第一主成分方向移动，而 K_1 的轨迹没有偏向任何主成分。 K_0 、 K_1 和 K_2 可以定义为依赖参数 r 的动能。当 r 越小，产生的轨迹越倾向于在最后主成分方向移动， r 越大轨迹越优先在第一主成分方向移动。过小或者过大的 r 所产生的轨迹都不理想。较为合适的 r 为 0 到 1 之间，其轨迹形态介于哈密顿和黎曼轨迹之间。

4.5 动能函数族

式(4.12)动能 K_r 受到参数 r 的控制。当 r 取 0 时对应于哈密顿蒙特卡洛采用的动能函数，动能 K_0 仅为动量 p 的函数。当 r 取 1 时类似于黎曼流形哈密顿蒙特卡洛采用的动能函数，动能同时为动量 p 和位置 q 的函数。当 r 介于 0 和 1 之间时，对应于一族动能函数。作为一种最特殊的动能， K_0 与物理动能类似。当 r 等于其它值不属于物理动能。由于哈密顿公式没有规定动能函数的形式，原则上可以采用任意形式的动能函数。

根据哈密顿公式，利用积分算法可以得到 p 和 q 的仿真轨迹，而我们主要对空间变量 q 感兴趣。图 4.7 显示针对环形势能函数的三种仿真轨迹，其动能函数的参数 r 分别取 0、0.5 和 1。蓝色轨迹对应于 K_0 的仿真轨迹，优先在极坐标的径向维度移动。绿色轨迹为 K_1 的仿真轨迹，优先在角度维移动。当 r 介于 0 和 1 之间时，仿真轨迹的形态也介于上述二者之间。二维概率模型仅有两个"主成分分量"，可以用 K_0 和 K_1 （或 $K_{0.5}$ ）的仿真轨迹进行遍历。如果多维概率模型的各个主成分差别较大时，则也可以考虑使用更多的仿真轨迹。

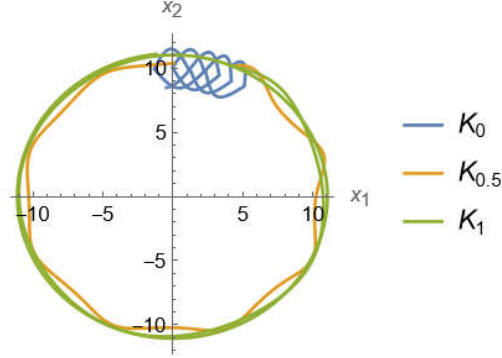


图 4.7 不同动能针对环形势能产生的哈密顿仿真轨迹

如式(4.29)所示，为了计算海森矩阵的小数次幂，对海森矩阵进行特征值分解，其中 V 和 Λ 分别为特征向量矩阵和特征值矩阵。

$$U_{qq} = V \Lambda V^T \quad (4.29)$$

从而将海森矩阵的 r 次幂转化为计算特征值的 r 次幂，如式(4.30)所示。

$$U_{qq}^r = V \Lambda^r V^T \quad (4.30)$$

当 r 位于 0 和 1 之间时，只有正的特征值才能得到有意义的结果，即要求海森矩阵正定。对于一般的概率密度函数对应的势能函数，海森矩阵随位置发生改变，其在某些位置有可能为非正定。因此需要修正动能函数族，使其适用于一般的情况。

由于二次函数的海森矩阵可以有正定、负定和不定之分，因此可以以二次势能函数为例讨论如何构造一般的动能函数。海森矩阵为正定的二次势能是一种简单和理想化的研究对象，如式(4.14)所示。这个势能函数类似于物理势能，仅有唯一一个最小值。可以采用常规的物理动能 K_0 以及动能 K_1 和 K_2 进行仿真。由于势能的海森矩阵为正定矩阵，因此动能 K_1 和 K_2 为正。物理动能 K_0 显然为正。由于动能和势能都是正数，因此总能量也是正数。在哈密顿仿真过程中，动能和势能之间相互转换，在仿真轨迹上总能量恒定。如图 4.8 所示，以这些动能生成的哈密顿仿真轨迹能够遍历势能函数对应的曲面，其中 x_1 和 x_2 是空间变量， U 是势能。 K_2 和 K_0 的轨迹分别优先在两个方向上移动。 K_1 动能的轨迹在两个维度上的访问速度相同。交替使用多种轨迹，可能有利于抽样。

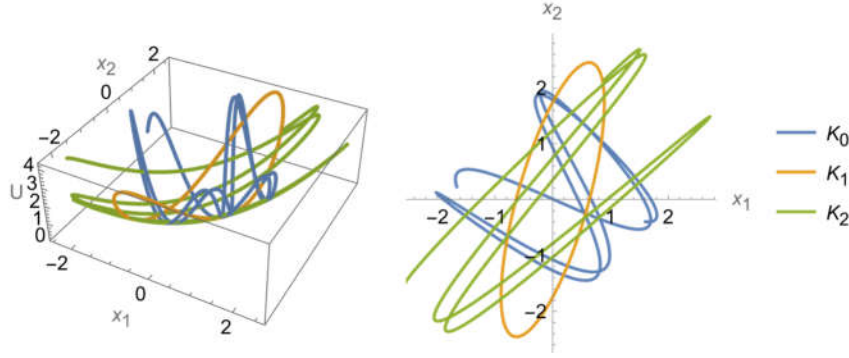


图 4.8 正定二次势能函数的仿真轨迹

具有非正定海森矩阵的二次势能函数也属于研究对象之一。海森矩阵为负定的一个例子如式(4.31)所示。

$$U(x_1, x_2) = -\frac{1}{2} \begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} 1 & 0.7 \\ 0.7 & 1 \end{pmatrix}^{-1} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad (4.31)$$

这个势能函数的值都是负数，为了获得稳定的仿真，需要采用负动能，因为采用正动能 K_0 会导致仿真发散。由于海森矩阵为负定， K_1 动能为负数，因此只需考虑 K_0 （以及 K_2 ）动能。显然，为了防止仿真发散， K_0 采用式(4.32)形式。

$$K_0(p, q) = -\frac{1}{2} p^T p \quad (4.32)$$

由于动能和势能都是负数，因此总能量也是负数，仿真保持总能量不变。如图 4.9 所示，这种情况下也能够生成稳定的仿真轨迹。三种仿真轨迹都能遍历凹曲面，因此轨迹可以用于抽样。此外，由于抽样算法有向低势能位置移动的趋势，因此抽样时应设置边界，使得超出边界的位置的接受概率为零。

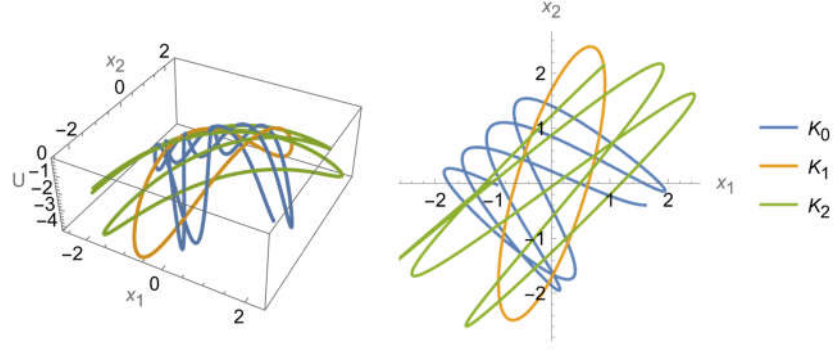


图 4.9 负定二次势能函数的仿真轨迹

除了正定和负定的情况，为了全面起见，还需要讨论海森矩阵为不定的势能，如式(4.33)。

$$U(x_1, x_2) = \frac{1}{2} (x_1, x_2) \begin{pmatrix} 0.7 & 1 \\ 1 & 0.7 \end{pmatrix}^{-1} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad (4.33)$$

此时海森矩阵的特征值包含正数和负数。 K_1 和上述两种情况一样，但采取前面介绍的的正的和负的动能 K_0 都会使仿真发散。使用随后介绍的方法，可以发现 K_0 应当具有式(4.34)形式。

$$K_0(p, q) = p_1 p_2 \quad (4.34)$$

如图 4.10 所示，这种情况下的二次势能函数曲面具有马鞍形状，没有势能极大值或极小值。虽然 K_0 和 K_1 都能遍历该曲面，但 K_1 的轨迹似乎难以访问高势能或者低势能区域。

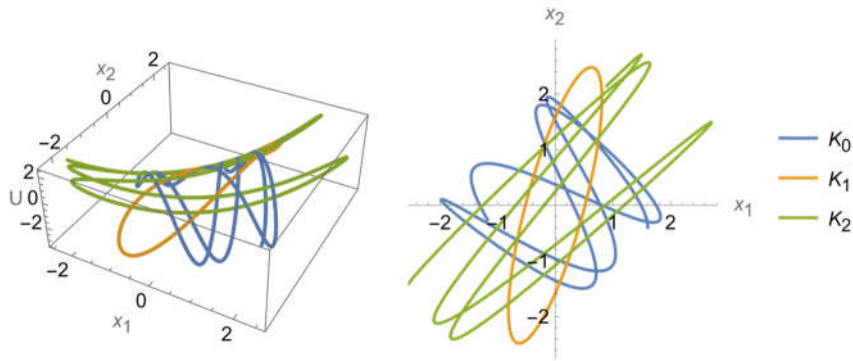


图 4.10 不定二次势能函数的仿真轨迹

总结以上几种情况，可以得出一个结论，即对特征值的绝对值取幂可以保留其符号。因此，如式(4.35)所示，求海森矩阵的 r 次幂的一种实现方式为：对特征值的绝对值求 r 次幂，并按元素乘以其符号。其中符号函数对正和负的特征值分别返回 1、-1， \odot 代表按元素乘法。

$$U_{qq}^r = V \left(|\Lambda|^r \odot \text{sign}(\Lambda) \right) V^T \quad (4.35)$$

因此，一般动能函数定义为式(4.36)，其中的 r 是一个实数参数，而不是求导变量。

$$K_r(p, q) = \frac{1}{2} p^T V \left(|\Lambda|^{-r} \odot \text{sign}(\Lambda) \right) V^T p \quad (4.36)$$

使用这种方式进行定义，可以将改进的动能函数族应用于一般的多维概率模型。因为当 r 为奇数时，按照原来的定义，动能 K_r 总能应用于一般的多维概率模型。但是当 r 为偶数或实数时，为了将动能 K_r 应用到一般的多维模型，则可以按上述方式计算海森矩阵的幂。改进方法除了要求特征值不为零以外，对模型没有其它要求。

除了上述形式的动能，我们也可以定义式(4.37)所示的动能。这两种动能中，哪种更适合抽样还有待进一步研究。

$$K_r(p, q) = \frac{1}{2} p^T V |\Lambda|^{-r} V^T p \quad (4.37)$$

上述的二次势能的例子显示， K_1 动能生成的轨迹没有偏向任何主成分，基于这种动能的方法理应能够实现准确的抽样。然而，定量实验显示，这种方法对于排序靠后的主成分的样本方差偏小，且越靠后的主成分偏小越严重。因此，需要分析产生问题的原因，并提出改进方案。

考虑正定的海森矩阵，对于 K_1 动能，式(4.38)显示将海森矩阵式(4.30)替换为其特征分解。

$$K(p, q) = \frac{1}{2} p^T V \Lambda^{-1} V^T p \quad (4.38)$$

式(4.39)显示，根据哈密顿公式(3.8)，位置的变化率等于动能对动量的偏导数。

$$\dot{q} = H_p = K_p = V \Lambda^{-1} V^T p \quad (4.39)$$

由于 Λ^{-1} 相当于各个主成分的更新幅度，而特征值矩阵倒数的平方根（即 $\Lambda^{-0.5}$ ）对应于主成分的尺度（即标准差），因此主成分的更新幅度和尺度不匹配。为了解决这个问题，我们可以用特征值矩阵的平方根设置各个主成分的更新幅度，即式(4.40)。

$$K_p = V \Lambda^{0.5} \Lambda^{-1} V^T p = V \Lambda^{-0.5} V^T p \quad (4.40)$$

因此动能函数应当具有式(4.41)所示的形式。

$$K(p, q) = \frac{1}{2} p^T V \Lambda^{-0.5} V^T p \quad (4.41)$$

考虑非正定海森矩阵的情况，于是得到式(4.42)所示的动能函数。

$$K(p, q) = \frac{1}{2} p^T V |\Lambda|^{-0.5} V^T p \quad (4.42)$$

这个动能函数是动能函数族 K_r 的一个成员（即 $K_{0.5}$ ）。由于此动能函数族中仅 $K_{0.5}$ 能够以相同的比例访问多维正态分布的各个主成分，因此 $K_{0.5}$ 也是较适宜用来实现抽样算法的动能函数。

4.6 正交仿真轨迹

基于上述动能函数族的多轨迹方法可以有效改善抽样效果。但此时动能函数族的仿真轨迹之间具有冗余作用，每种轨迹都以自己的方式同时访问所有成分。如果各个成分的尺度相差不大，则任何一种单一轨迹都能够充分遍历所有成分。但是，当各个成分的尺度差别非常大时，即使增加更多的动能也未必能够改善抽样效果。

根据前面的讨论可知，对海森矩阵（的逆矩阵）的特征值进行变换可以得到一族动能函数。与此类似，每个特征值都对应一种仿真轨迹，而这些不同类型的轨迹相互正交。本人提出，可以采用基于 $K_{0.5}$ 动能的正交轨迹进行遍历，轨迹之间没有冗余，每种轨迹只遍历局部坐标的一个维度。由于消除了轨迹之间的冗余，所有轨迹的组合能够有效遍历任意类型的空间。

从表面上看，由于在同一马氏链上运行多个相互独立的抽样算法，这种方法似乎稳定性更差。然而，由于多个粒子的整体信息能够实现自动调节运行参数，因此这种多轨迹算法可以与单轨迹算法一样稳定。由于这种方式的多轨迹算法能确保所有成分都能够快速收敛，消除了由于某些成分收敛极慢带来的不利影响，反而有可能降低一些模型的运行时间。

以下以二次势能和环形势能为例，显示正交仿真轨迹。时间步长为 0.01，仿真步数为 2000 步。从两个初始位置开始，各自生成两个正交轨迹。

式(4.14)为二次正定势能，其海森矩阵的特征值的倒数为 1.7 和 0.3，分别为两个主成分方向上的方差。海森矩阵的条件数为两个特征值之比，在这里为 1.7/0.3。如果条件数很高，上文描述的单一轨迹可能会遇到困难。图 4.11 显示在局部坐标系的两种正交的仿真轨迹。左图显示势能-空间联合轨迹，右图显示空间仿真轨迹。灰黑色和蓝色为第一主成分方向的仿真轨迹，其区别在于初始位置不同。红色和橘色轨迹为第二主成分方向的仿真轨迹，与灰色和蓝色轨迹成正交，说明这两种轨迹各自遍历一个局部坐标轴。使用多条正交仿真轨迹的遍历方式不

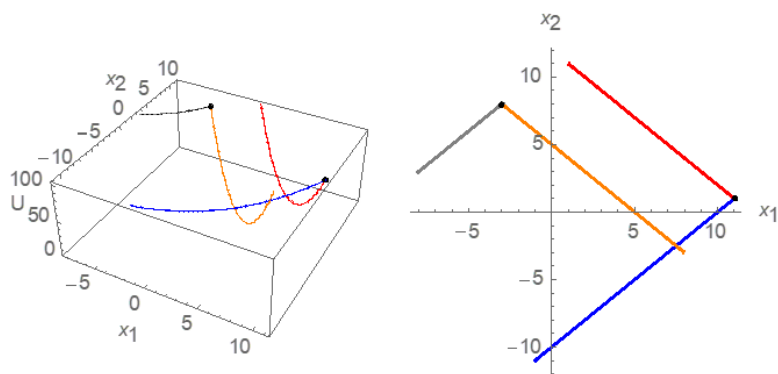


图 4.11 正定势能的正交仿真轨迹，灰色和蓝色：第一主成分方向，橘色和红色：第二主成分方向，黑点：初始位置

再受条件数的限制，能够应用于条件数极高的场合。

图 4.12 显示在相同的条件下，采用式(4.31)负定势能的正交仿真轨迹。如左图所示，负定的势能曲面具有凹的形状，与正定势能的特点相反。右图显示，由于初始条件相同，空间仿真轨迹与正定势能的仿真轨迹相同，说明这两种势能海森矩阵具有相同的特征向量。这个负定势能海森矩阵的特征值的倒数为-1.7 和-0.3，绝对值与正定的情况相同，但符号为负。这个例子说明 $K_{0.5}$ 可以应用于负定势能，并产生稳定的正交仿真轨迹。

全面起见，还可以研究不定势能的情况（式 4.33）。此时，海森矩阵的特征向量与正定和负定的情况相同，特征值的倒数为 1.7 和-0.3。如图 4.13 所示，两种不同的正交轨迹具有不同的凹凸性质，其空间仿真轨迹与正定和不定的情况相同。上述几种情况的特征值的绝对值相同，特征向量也相同，因此如果仿真步数足够多，则仿真轨迹形成的线段也相同。

对于环形势能，局部坐标系就是相当于极坐标系的两个坐标轴，其方向随位置改变。图 4.14 显示，红色和橘色曲线沿着径向移动，蓝色和灰色曲线沿着角度

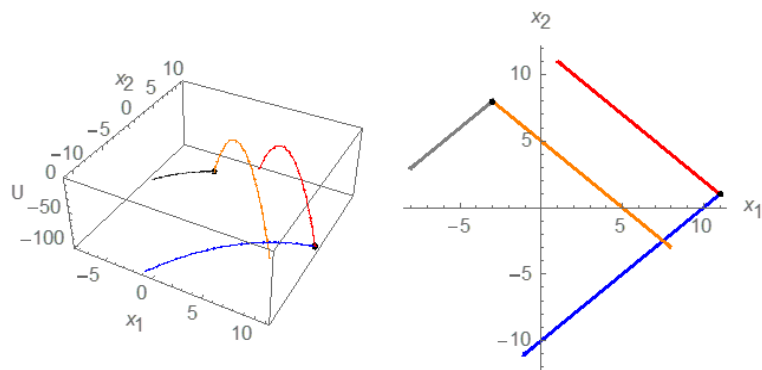


图 4.12 负定势能的正交仿真轨迹

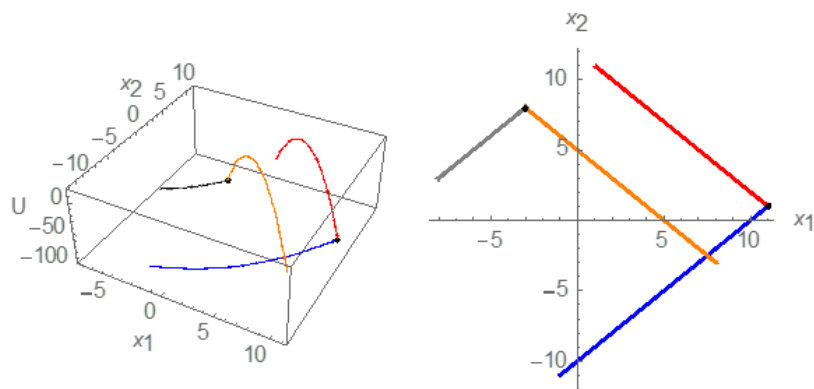


图 4.13 不定势能的正交仿真轨迹

维移动，这两种类型的仿真轨迹相互正交。灰色曲线位于圆内，蓝色曲线位于圆外。沿着角度维的仿真轨迹保持势能不变，能够进行大范围移动。径向维的仿真

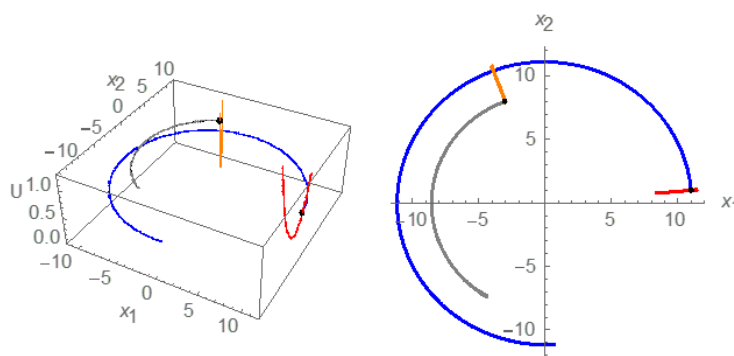


图 4.14 环形势能的正交仿真轨迹，灰色和蓝色：“第一”主成分方向，橘色和红色：“第二”主成分方向，黑点：初始位置

轨迹上的势能变化显著，进行径向局部移动。分别在两个方向上移动能够有效覆盖空间区域。

4.7 本章小结

本章以哈密顿动力学公式和麦克斯韦-玻尔兹曼分布为基础研究了遵守能量守恒的多粒子系统的特点，揭示了动能和仿真轨迹形态之间的联系。多粒子系统主要涉及算法的稳定性，而动能与抽样算法的准确性有关。本人提出，为了同时改进已有算法的稳定性和准确性，需要实现多粒子系统以及使用合适的动能函数。

5 多粒子抽样算法

上一章引入了总能量恒定的多粒子模型及阐释了一些动能函数及其影响。本章首先推导动能函数相对于位置的导数，然后再描述算法，并揭示能量导数的影响，最后展示一些定量的数值实验，以说明经典方法所采用的动能函数存在的缺陷，以及本文算法对此的改进。

5.1 能量导数

使用一般动能函数的哈密顿仿真需要计算能量函数相对于相空间变量的导数。势能仅是 q 的函数，因此需要计算三个导数，即 U_q 、 K_p 和 K_q 。 U_q 由自动求导算法提供，而 K_p 可以直接实现，只需讨论 K_q 。动能函数对特征值进行了变换，而特征值又依赖于势能函数的二阶导数，为了计算动能函数相对于 q 的导数，需要分别计算动能相对于特征值和特征向量的导数，以及特征值和特征向量相对于 q 的导数。因此问题转化为特征值和特征向量相对于单变量 t 的导数，而矩阵每个元素都是 t 的函数。对于 D 维变量 q ，可以直接将计算方法推广到 D 维。附录中列出了特征值和特征向量相对于单变量的导数的计算方法，因此只需利用其结论即可推导出 K_q 。

式(5.1)显示，对势能的海森矩阵进行特征分解，得到 D 个特征向量和特征值，其中 v_i 和 λ_i 为第 i 个特征向量和特征值。

$$U_{qq} v_i = \lambda_i v_i, i = 1, \dots, D \quad (5.1)$$

根据第二章的讨论，动能函数相当于对特征值进行变换，因此使用特征向量和特征值，参数为 r 的动能函数定义如式(5.2)所示，其中 V 为势能的海森矩阵的特征向量矩阵， f 为特征值矩阵的变换函数。

$$K(p, q; r) = \frac{1}{2} p^T V \cdot f(\Lambda; r) V^T p = \frac{1}{2} y^T f(\Lambda; r) y \quad (5.2)$$

为了简化公式，式(5.3)定义向量 y 。

$$y = V^T p \quad (5.3)$$

正交与非正交轨迹的动能函数的区别主要在于特征值变换函数。对于正交轨迹的算法，轨迹种类数等于概率模型的维数。对于非正交轨迹的算法，轨迹种类数需要预先设定。

非正交动能函数族的特征值变换函数由式(5.4)所示，其中 \odot 表示按元素相乘，而 sign 为符号函数。

$$f(\Lambda; r) = |\Lambda|^{-r} \odot \text{sign}(\Lambda) \quad (5.4)$$

其导数定义为式(5.5)。

$$f_{\Lambda} = -r |\Lambda|^{-r-1} \quad (5.5)$$

对于正交轨迹，当生成第 i 个轨迹时，只需使用第 i 个特征值即可。因此正交动能的特征值变换函数定义为式(5.6)。

$$f(\Lambda; i) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & |\lambda_i|^{-0.5} & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (5.6)$$

其导函数定义为式(5.7)。

$$f_{\lambda_i} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -0.5 \operatorname{sign}(\lambda_i) \times |\lambda_i|^{-1.5} & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (5.7)$$

由于需要求 K_q , p 可看作常数，因此特征值和特征向量为位置 q 的函数。动能对特征值进行了变换，但没有对特征向量进行变换。根据分步求导法则，动能函数相对于 q_j 的导数分为两项，如式(5.8)所示。

$$K_{q_j} = p^T V_{q_j} f(\Lambda) V^T p + \frac{1}{2} y^T f_{q_j} y = p^T V_{q_j} f(\Lambda) y + \frac{1}{2} y^T (f_{\Lambda} \odot \Lambda_{q_j}) y \quad (5.8)$$

于是求势能函数相对于 q 的导数转化为求特征值和特征向量对 q 的导数。从附录得知，特征值的导数可以使用海森矩阵相对于 q 的导数计算，即式(5.9)。

$$\lambda_{iq_j} = V_i^T U_{qqq_j} V_i \quad (5.9)$$

类似的，特征向量对 q 的导数为式(5.10)。

$$V_{iq_j} = \sum_{k \neq i} \frac{1}{\lambda_i - \lambda_k} (V_k^T U_{qqq_j} V_i) V_k \quad (5.10)$$

式(5.9)和(5.10)说明，求动能对位置的导数，需要计算势能对位置的三阶导数，其计算量比海森矩阵大一个量级。对于维数较高的模型，可以采用随后介绍的近似方法，即不计算 K_q ，直接在每一步仿真之后重新设置动量幅值，以保证能量守恒。

K_q 也可以用数值差分计算，但抽样效果与微元 δ 有关，因此数值差分可能会引入抽样误差。图 5.1 显示环形分布的半径方向上的样本直方图。对于 σ 参数

等于 0.01 的环形分布，差分中不同的 δ 会影响抽样结果。在这个例子中，较大的 δ 会产生较大的误差，即样本向环外偏移。受到浮点位数的限制， δ 的值不能无限小，因此差分算法一般会引入一定的误差。差分算法的计算量与精确方法的计

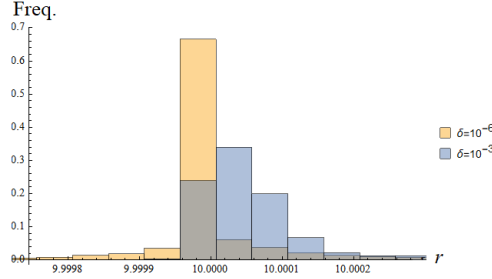


图 5.1 采用差分求导实现的 K_1 抽样算法，环形概率样本在径向样本直方图

算量为同一个数量级，在无法计算函数的三阶导数的环境下可以使用差分算法。

5.2 算法描述

算法 5.1 显示非正交轨迹抽样算法^[68]。算法使用了 n 种轨迹，而 H 和 δ 都是 n 维数组，因为每种轨迹的总能量和步长需要单独进行调节。算法输入 n 维数组 R ，其中每一个元素对应于一种动能的参数 r 。当 R 仅包含 0 时相当于改进的 HMC， R 仅包含 1 时为改进的 RMHMC。当 R 数组包含 0 和 1 时相当于交替双轨迹算法。如果需要抽样更为极端的概率，可以使 R 包含更多的介于 0 和 1 之间的小数。第 4 行从数组中顺序选择一种动能。第 9 行使用这种动能计算所有粒子的总动能。第 12 行，利用总势能对应的总能量来获得期望的总动能。随后重新设置所有粒子的动量幅值，以确保系统总能量守恒。从第 17 行开始，分别对每个粒子进行仿真和抽样。第 18 行执行蛙跳算法。由于只使用变量 q 进行抽样，因此蛙跳算法在最后更新 p 的运算可以忽略。第 20 行和 21 行使用与动能对应的步长。第 32 行调节与该动能对应的总能量和步长。第 33 行开始顺序使用下一种动能。

算法 5.2 定义动能及其相对于 p 的导数，二者都调用一个权矩阵函数。正交和非正交算法的区别在于权矩阵函数，在函数中对特征值进行不同的变换对应不同的算法。

算法 5.3 显示非正交轨迹动能函数的权矩阵函数需要变换特征值， r 为 0 和 1 之间的实数。特征分解得到特征值对角矩阵和特征向量矩阵，随后按前文所述方式变换特征值矩阵。

算法 5.1: 非正交轨迹多粒子抽样算法

输入: M 个粒子的初始位置 q_j , 动能的参数数组 R 。

```
1: initialize  $H$  and  $\delta$ 
2:  $\rho=1$ 
3: for  $i = 1$  to  $n$ 
4:    $r=R[\rho]$ 
5:    $U_{\text{total}}=0$ 
6:    $K_{\text{total}}=0$ 
7:   for  $j=1$  to  $M$ 
8:      $p_j \sim N(0,1)$ 
9:      $K_{\text{total}}=K_{\text{total}} + K(p_j, q_j, r)$ 
10:     $U_{\text{total}}=U_{\text{total}}+U(q_j)$ 
11:   end
12:    $K_d=H[\rho]-U_{\text{total}}$ 
13:   for  $j=1$  to  $M$ 
14:      $p_j = p_j \sqrt{|K_d / K_{\text{total}}|}$ 
15:   end
16:   for  $j=1$  to  $M$ 
17:      $q_0=q_j$ 
18:      $p_j=p_j-0.5\delta[\rho](U_q(q_j)+K_q(p_j, q_j, r))$ 
19:     for  $s = 1$  to  $S$ 
20:        $q_j=q_j+\delta[\rho] K_p(p_j, q_j, r)$ 
21:        $p_j=p_j-\delta[\rho](U_q(q_j)+K_q(p_j, q_j, r))$ 
22:       save  $U(q_j)$ 
23:     end
24:      $\alpha = e^{U(q_0)-U(q)}$ 
25:      $u \sim U(0,1)$ 
26:     if  $\alpha < u$ 
27:        $q_j=q_0$ 
28:     end
29:     save  $q_j, \alpha$ 
30:   end
31: end
32: tune  $H[\rho]$  and  $\delta[\rho]$ 
33: if  $\rho == \text{size of } R$ 
34:    $\rho=1$ 
35: else
36:    $\rho=\rho+1$ 
37: end
38: end
```

输出: 一系列保存的位置 q 。

算法 5.2: 动能及其相对于 p 的导数

```
1: function  $K(p,q,r)$ 
2:    $W = \text{get}W(p,r)$ 
3:   return  $1/2p^T \cdot W \cdot p$ 
4: end
5: function  $K_p(p, q, r)$ 
6:    $W = \text{get}W(p,r)$ 
7:   return  $W \cdot p$ 
8: end
```

算法 5.3: 非正交轨迹的权矩阵函数

```
1: function  $\text{get}W(q,r)$ 
2:    $\Lambda, V = \text{eigen}(U_{qq}(q))$ 
3:   return  $V \left( |\Lambda|^{-r} \odot \text{sign}(\Lambda) \right) V^T$ 
5: end
```

算法 5.4 显示正交轨迹算法。这个算法与之前介绍的非正交算法非常相似，区别在于直接用 ρ 作为调用能量及其导数的参数，其最大值为模型的维数。

算法 5.4: 正交轨迹多粒子抽样算法

输入: M 个粒子的初始位置 q_j 。

```
1: initialize  $H$  and  $\delta$ 
2:  $\rho = 1$ 
3: for  $i = 1$  to  $n$ 
4:   ...
5:   if  $\rho == D$ 
6:      $\rho = 1$ 
7:   else
8:      $\rho = \rho + 1$ 
9:   end
10: end
```

输出: 一系列保存的位置 q 。

算法 5.5 显示计算 K_q 的过程，其中 f 和 f_Λ 为特征值变换函数及其导数。从第 6 行到第 14 行计算特征向量相对于 q 的导数。特征值相对于 q 的导数在第 21 行直接计算。第 10 行变量 u 是 $D \times D \times D$ 维张量，为特征向量相对于 q 的导数。第 16 行累计由于特征向量产生的导数。第 21 行累计由特征值引进的导数。

算法 5.5: 动能相对于 q 的导数

```
1:  function  $K_q(p, q, r)$ 
2:       $D = \text{dimension}(p)$ 
3:       $\Lambda, V = \text{eigen}(U_{aa})$ 
4:       $V = V / \|V\|$ 
5:       $y = V \cdot p$ 
6:      for  $k=1$  to  $D$ 
7:          for  $i=1$  to  $D$ 
8:              for  $j=1$  to  $D$ 
9:                  if  $i \neq j$ 
10:                      $u_{k,i,i} = u_{k,i,i} + v_i / (\lambda_i - \lambda_j) (v_j^T U_{aaa} v_i)$ 
11:                  end
12:              end
13:          end
14:      end
15:      /* due to  $V^*$  */
16:      for  $i=1$  to  $D$ 
17:           $K_{q,i} = p^T \cdot u_i \cdot f \cdot y$ 
18:      end
19:       $g = f_{\Lambda} \odot y^2$ 
20:      /* due to  $\Lambda^*$  */
21:      for  $j=1$  to  $D$ 
22:          for  $i=1$  to  $D$ 
23:               $K_{a,i} = K_{a,i} + (1 / 2 v_i^T \cdot U_{aaa} \cdot v_i) g_i$ 
24:          end
25:      end
26:      return  $K_q$ 
27:  end
```

5.3 能量导数的影响

由于动能也是位置的函数，因此在理论上需要包括哈密顿公式中的 K_q 。一般的动能依赖于势能相对于 q 的二阶导数，因此 K_q 需要计算势能相对于 q 的三阶导数，其计算量较大。因此在实现算法时可以选择忽略 K_q 或者采用近似算法。

如果缺失这一项，则算法可能会产生系统性误差。

图 5.2 显示使用环形势能和 K_1 动能产生的仿真轨迹，坐标轴为空间坐标。在忽略 K_q 的条件下，总能量不守恒，环形势能和 K_1 动能产生的仿真轨迹会逐渐向圆环外侧偏移。使用 K_q 的仿真轨迹比较稳定。物理动能 K_0 仅是 p 的函数， K_q 为零，不受此因素的影响。当交替使用多种轨迹时会降低相应的系统误差。因为即使 K_1 的轨迹向外侧偏移， K_0 也会有一定概率向圆环内测移动，从而消除了部

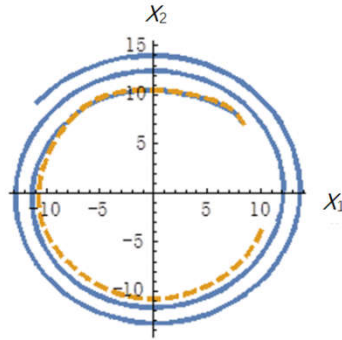


图 5.2 忽略 K_q 对仿真轨迹的影响，蓝色：不使用 K_q
橘色：使用 K_q

分系统误差。

图 5.3 中的浅灰色曲线以(9.9,0)为初始位置的环形势能的仿真轨迹，坐标轴为空间坐标。棕色箭头显示在仿真轨迹上 K_p 的方向，即空间位置的增量方向沿着圆周切线方向。向外的蓝色箭头显示 U_q 的方向，向内的红色箭头显示 K_q 的方向。只考虑 U_q 会导致轨迹向圆外侧移动。黑色箭头显示两个导数之和，即动量的增量方向。由于 U_q 和 K_q 的方向几乎相反而完全抵消，因此动量的增量的幅值很小，而这又会降低位置增量的幅值。因此我们会遇到一个两难问题，使用 K_q 的算法理论上准确，但探索能力差；不使用 K_q 的近似算法反而可能具有较强的探索能力。

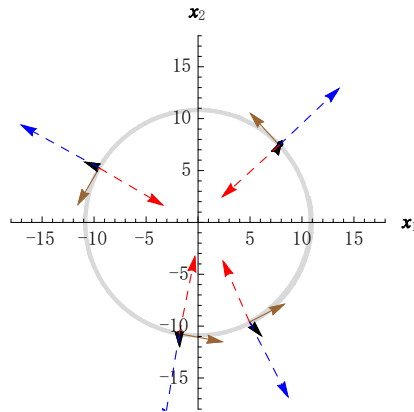


图 5.3 K_1 仿真轨迹上的能量导数，棕色箭头： K_p ，蓝色箭头： U_q
红色箭头： K_q ，黑色箭头： $U_q + K_q$

图 5.4 为使用 K_1 动能得到的环形分布的样本在径向方向上的直方图, σ 参数为 0.1, 半径 ρ 为 10。上文描述的近似方法的样本多数大于 10, 说明存在偏差。如果在算法中使用 K_q , 样本大致平衡分布在 10 两侧, 因此消除了有关的偏差。对于准确方法, 径向样本分布与正态分布也可能有区别。使用 K_q 的算法虽然在径向维比较准确, 但在角度维的误差较大, 说明算法的探索能力较差。右图显示在角度维, 不使用 K_q 的方法更接近于平均分布, 意味其轨迹的探索能力更强。

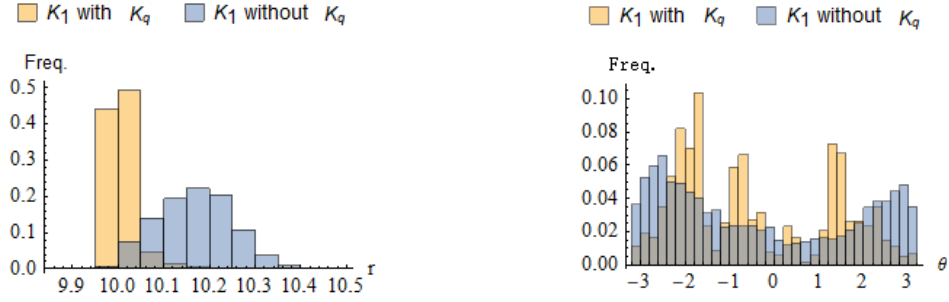


图 5.4 环形概率样本在径向直方图, 左: 径向, 右: 角度

上述导数的计算量大致具有式(5.11)所示的对比关系。

$$U_q \ll K_p \ll K_q \quad (5.11)$$

造成差别的原因是 K_p 需要计算 U_{qq} , 而 K_q 则需要计算 U_{qqq} 。由于 K_q 的计算量较大, 在实际实现算法时可以忽略这一项, 但会带来相应的系统误差。由于误差导致能量不守恒, 因此可以在仿真过程中, 重新设置动量的幅度, 使总能量恒定。假设一个粒子仿真前的总能量为 h_0 , 一步仿真之后的总能量变为 h_1 , 相应的势能为 k_1 。如果仿真使总能量增加了, 则新位置的动能应当减去增加的总能量, 用如式(5.12)所示的比例重新设置动量的幅值。

$$\sqrt{\left| \frac{k_1 - (h_1 - h_0)}{k_1} \right|} \quad (5.12)$$

图 5.5 所示, 采取这种措施消除了仿真轨迹的向外发散的趋势。这种方法虽然无法消除全部误差, 但至少可以保证仿真稳定。既不用计算 K_q , 又消除了部分误差, 在计算量和准确性之间做出了相对的平衡。对于高度非正态的模型, 省略 K_q 可能会导致较大的仿真误差和能量误差, 从而给抽样造成误差。此外, 省略 K_q 带来的误差一般远大于积分算法带来的误差, 因此仿真算法可以采用最简单的欧拉积分。一般需要先消除由于忽略 K_q 带来的误差, 而后才需考虑由积分算法带来的误差。

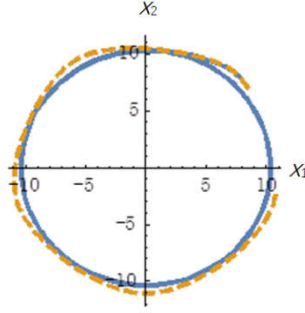


图 5.5 保持能量守恒对仿真轨迹的影响，
蓝色：不使用 K_q ，橘色：使用 K_q

不同动能会产生不同类型的仿真轨迹，当交替使用多种轨迹时能够降低相应的系统误差。如图 5.6 所示， K_0 和 K_1 产生的轨迹呈近似互补角度。如果 K_1 的轨迹向外侧偏移，由于外侧区域势能较高，下一次运行 K_0 仿真时，轨迹会从外侧穿越回圆环内侧，在轨迹上抽样的样本就会分布于圆周内外两侧，从而消除了部分系统误差。因此，对于多轨迹算法，可以使用近似方法。

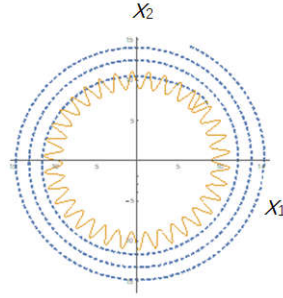


图 5.6 不保持能量守恒的仿真轨迹；橘色： K_0 ，蓝色： K_1

将抽样结果转换为极坐标，得到径向样本和角度维样本。使用 K_1 动能的抽样样本几乎全部大于 10，即位于环外侧区域。而综合使用 K_1 和 K_0 的抽样结果大致对称分布于 10 的两侧。当 σ 参数设置为 0.01，样本应分布在均值两侧 ± 0.03 以内。但 K_1 样本分布较窄，而使用两种势能的样本分布范围更加合理。因此使用多种动能有助于降低样本误差。

使用欧拉积分实现哈密顿仿真，可以得到式(5.13)和式(5.14)所示的更新公式，其中 δ 为步长。

$$q_{n+1} = q_n + \delta K_p \quad (5.13)$$

$$p_{n+1} = p_n - \delta (U_q + K_q) \quad (5.14)$$

欧拉积分会产生一定的数值误差，使得仿真轨迹偏离理想形态。蛙跳算法是一种改进的积分方法，广泛应用于哈密顿蒙特卡洛算法中。蛙跳算法通过稍微增加计算量来降低误差，优势体现在仿真步数较多的时候。

在仿真算法包括 K_q 时，采用蛙跳算法可以取得一定的效果。首先在仿真开始前，动量移动半步，如式(5.15)所示。

$$p_{n+1} = p_n - \frac{\delta}{2} (U_q + K_q) \quad (5.15)$$

然后按式(5.16)和式(5.17)执行若干次哈密顿仿真。

$$q_{n+1} = q_n + \delta K_p \quad (5.16)$$

$$p_{n+1} = p_n - \delta (U_q + K_q) \quad (5.17)$$

蛙跳算法在最后一次仿真，动量应当执行半步。但我们只关心位置变量，因此无需最后调整动量。

为了研究使用蛙跳积分和 K_1 动能算法的抽样准确程度，将抽样样本转换为径向和角度。根据势能公式，径向维应当大致符合标准正态分布。环形分布的径向样本分布如图 5.7 所示，横坐标为径向值，纵坐标为比率。采用蛙跳算法的径向样本平衡分布在 10 两侧，大致符合预期。右图显示欧拉积分的抽样结果分布于 10 的右侧，即样本分布于环外侧，说明欧拉积分导致了较大的抽样误差。

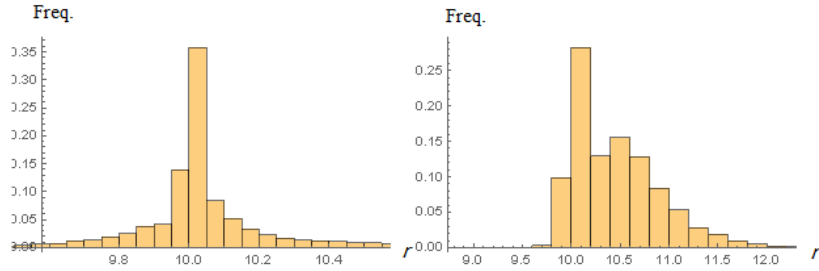


图 5.7 环形分布在径向的样本，左图：蛙跳算法，右图：欧拉积分

5.4 运行参数

影响仿真效果的两个主要参数是步长和仿真步数，二者的乘积大致相当于轨迹路径长度。因此可以固定一个参数，调整另一个参数。如果算法遵守能量守恒，则还需要设置总能量。本论文算法的运行参数为时间步长和总能量。采用固定的仿真步数可以较大程度地降低运算量。由于根据总势能和总能量可以得到总动能，因此算法实际上是通过总动能来调节总能量。增加时间步长和总动能都能增加仿真路径长度，但前者对仿真的影响更大。其原因是时间步长和动量在哈密顿公式中是乘积关系，而动能为动量的平方关系，因此时间步长的影响是动能影响的平方。通过粗调时间步长并微调动能可以将平均接受概率控制在合适范围。

5.4.1 时间步长

时间步长一般被认为是 HMC 算法的主要运行参数,它不仅影响仿真稳定性,还影响抽样的效率。过小的时间步长会减缓仿真进程,但过大的时间步长会导致不稳定。本人提出根据仿真过程中势能最大值和最小值来调节时间步长。当时间步长过小时,仿真过程中势能的最大值和最小值会出现在仿真开始和结束位置。因此如果所有粒子的势能的最大值和最小值都是仅仅出现在仿真的开始时刻和结束时刻,则说明时间步长过小,应当适当增加步长。类似地,如果时间步长过大,则最小势能会出现在仿真开始时刻,而最大势能出现在仿真结束时刻。如果这种现象同时出现在所有粒子的仿真状态中,则说明时间步长过大,应当降低步长。

所有粒子的仿真都同时出现这种状况为小概率事件,因此对于大多数情况,时间步长维持不变。以多个粒子的统计量调节运行参数能够大概率获得较为成功的调节效果。调节步长的方法与“禁止掉头抽样器”的主要区别在于,其基于的是多个粒子在仿真过程中势能的最值而不是单一仿真轨迹的空间回转。基于多粒子的势能的方案不仅在计算上更简单,而且能够更好地描述运行状态。可以根据最大值或是最小值发生在仿真开始或者结束位置来区分步长过大或者过小,而空间回转准则无法区分这两种不同的状况。根据若干个粒子在仿真过程中能量最值的位置信息,我们可以将时间步长调节在一个大致范围。

5.4.2 总能量

为了对算法进行进一步微调,可以通过系统总动能来控制平均接受概率,即利用平均接受概率调节系统总动能。总能量为势能与动能之和,而势能为一个相对值,因此通过调节总动能就可以改变总能量。虽然一般假定动能为正值,但如果采用 RMHMC 的动能形式,当海森矩阵为负定时会产生负动能。因此,总能量可以大于势能,也可以小于势能,即各种能量均可可为正值或者负值。

这里采用根据多个粒子的平均接受概率调节总能量。当平均接受概率过大时,说明总动能过小,需要增大总动能的绝对值。考虑到总能量可以大于或者小于总势能,因此可以使总能量远离总势能。反之,如果接受概率过小,说明总动能过大,应降低总动能的绝对值,使总能量接近总势能。当海森矩阵为非正定时,会产生负的总动能。这种调节方式考虑到了负的总动能的这种情况。

平均接受概率可以设置上限和下限,例如 0.9 和 0.1。平均接受概率的界限较宽,允许接受概率在一定幅度内变化,可以增强探索能力。如果界限过窄,会导致调整较为频繁,容易导致不稳定。合适的调节倍数(如 1.1)可以防止调节速度过快或者过慢。

本论文算法对时间步长和总能量都是以 1.1 的倍数进行调节。由于使用了多个粒子，可以大概率保证正确的调节方向。目前的实验表明，数百次的迭代可以确保调节到合适的参数。因为当 n 为数百时，从 1.1^{-n} 到 1.1^n 能够覆盖相当大的范围，因此时间步长和总能量的具体初始值带来的影响不大。本论文的算法实现将初始步长设置成较小的值（如 10^{-9} ），初始总能量设置为初始总势能的近似值。较小的时间步长会导致初始的平均接受概率接近于常数 1。由于时间步长的调整比能量调整更快，当时间步长调整为合适值之后，调节总能量会使平均接受概率较为均匀地分布于 0 和 1 之间，从而获得较为理想的抽样效果。

5.5 先验概率

先验概率是贝叶斯统计的一个主要特点，不同的先验概率一般会影响后验概率的抽样结果。如果没有明确指定先验，那么实际上使用的是无限区域的非信息先验。在算法中可以添加先验概率的处理。先验概率可以分为一般的连续分布的先验和非信息先验。对于一般的连续分布的先验概率，只需将其与似然函数相乘，即可得到用于抽样的后验概率。因此只需考虑非信息先验概率。

非信息先验相当于在一定区域内的均匀分布。由于区域边界之外的概率为零，有的学者称之为突然变化的先验，并认为这种先验会导致问题^[69]。由于 Metropolis 算法依据的是后验概率的比值，均匀分布被分式约掉，因此可以假设均匀分布的概率密度为常数 1。当仿真后的样本位于先验分布的区域内，后验概率等于似然函数。当检测到候选样本位于先验分布之外，由于先验概率为零，导致后验概率也为零。此时 Metropolis 算法的接受概率为 0，防止了移动到先验区域之外。同样的技术也可以实现一些特殊的先验概率，如半正态分布。此时后验为似然和正态概率密度函数的乘积，但在仿真后需要检测是否进入零概率区域。如果移动到零概率区域，则接受概率也为零。通过这种方式可以实现广泛类型的先验概率，如复杂区域、多区域、非连通区域等类型。

5.6 实验

本节首先使用二维正态分布和环形分布验证改进算法的抽样效果。随后，使用多维正态分布对算法进行了测试。

5.6.1 二维正态分布

本实验使用双轨迹方法对式(5.18)所示的二维正态分布进行抽样。

$$P(x_1, x_2; \sigma) \propto e^{-\frac{1}{2}(x_1 \ x_2) \begin{pmatrix} 1 & 0 \\ 0 & \sigma^2 \end{pmatrix}^{-1} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}} \quad (5.18)$$

为了定量研究抽样的准确程度,本文使用相对熵(即 Kullback-Leibler 散度, KL 散度),式(5.19)来衡量根据样本拟合的分布 $Q(x)$ 和目标分布 $P(x)$ 之间的误差。

$$\int_{-\infty}^{\infty} P(x) \log \frac{P(x)}{Q(x)} dx \quad (5.19)$$

对于二维正态分布,每一个维度都需要单独计算 KL 散度。第一主成分方向的标准差为 1,第二主成分方向的标准差为 σ ,小于 1。通过变动 σ 的大小,可以考察各个维度上的以 KL 散度方式衡量的抽样误差随着协方差矩阵条件值的变化规律。

图 5.8 显示各种方法对于二维正态的第一主成分方向的抽样误差的对比,其中较大的 KL 散度对应于较大的误差。为了研究 σ 的多个数量级的影响,横坐标相当于 σ 参数的负对数。HMC0 和 RMHMC0 分别为经典的方法,而 HMC 和 RMHMC 为遵守能量守恒的改进算法。CHMC(Complementary Hamiltonian Monte Carlo)指的是交替运行两种仿真的改进方法。

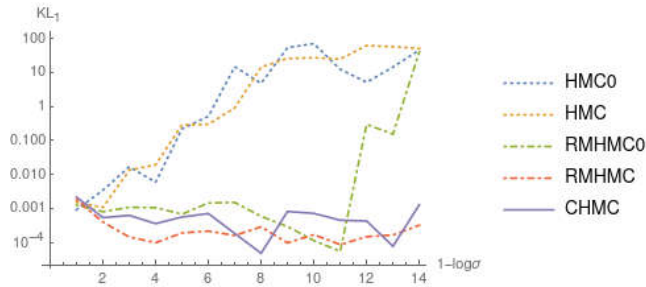


图 5.8 在第一主成分方向上的抽样误差

基于 HMC 的抽样器在第一主成分方向不准确,而且随着 σ 的降低,误差会持续增加。遵守能量守恒的 HMC 具有类似的趋势。基于 RMHMC 的抽样器与此相反,对于广泛的 σ 值都能保持较小的误差。如果 σ 的值非常小,那么经典的 RMHMC 的误差也会迅速增加,但遵守能量守恒的 RMHMC 和 CHMC 仍然保持较小的误差。这说明改进方法在第一主成分方向的稳定性得到改善。

与此类似,图 5.9 显示在第二主成分方向的抽样误差,横轴向右代表 σ 降低

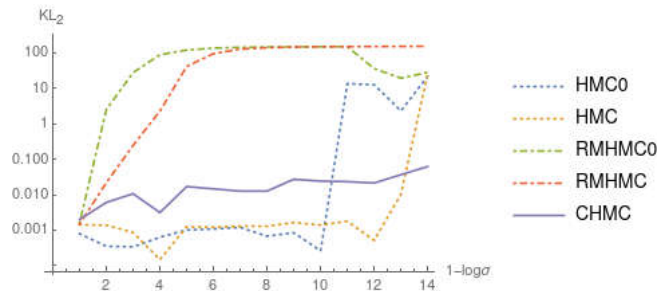


图 5.9 在第二主成分方向上的抽样误差

的方向。此图明显与图 5.8 的趋势几乎相反，即基于 RMHMC 的方法产生了较大的误差，而基于 HMC 方法的误差较小。这种现象的原因是因为 HMC 的轨迹与最后主成分方向一致，而 RMHMC 的轨迹与第一主成分方向一致。遵守能量守恒的方法对于较小的 σ 能够产生较小的误差，因此比经典方法更稳定。经典的 HMC 方法对于较大的 σ 可以产生较小的抽样误差，但当 σ 降低到一定程度，误差会迅速增加。作为对比，在相同的范围内，改进的 HMC 保持了较小的误差，仅当 σ 继续降低，其误差才会增加。另一方面，CHMC 能够一直保持较小的误差。因此本文提出的 CHMC 方法在最后主成分方向更稳定。

比较图 5.8 和图 5.9 可知，HMC 和 RMHMC 都会在某些主成分方向上较为准确，但在另外一些主成分方向上不准确。在相同实验条件下，与经典方法相比，遵守能量守恒的方法可以获得类似的或者更稳定的抽样结果。一般而言，交替运行两种仿真并且保持能量守恒的方法可以同时两个主成分方向上获得较小的抽样误差。

保持系统能量守恒的一个显著优点是可以通过总能量调节接受概率。经典的 HMC 算法几乎没有可以直接控制接受概率的方式，只能依赖于时间步长等间接方式。正如前面所讨论的，经典方法接受概率在理论上为常数 1，因此如果选择合适的时间步长，则接受概率可能会接近于常数 1。一般而言，选择一个合适的时间步长对于经典的 HMC 算法是比较困难的任务，因为算法的运行经常与时间步长有关。图 5.10 显示，当时间步长设置为 10^{-4} 时，经典 HMC 的平均接受概率接近于 0，说明时间步长过大。在相同条件下，如果时间步长设置成一个较小的值（如 10^{-9} ），那么接受概率将有所改善。尽管仍然偏小且不稳定。HMC 算法的一个主要缺陷是对时间步长非常敏感，不同的值可能会导致过大或者过小的接受概率。

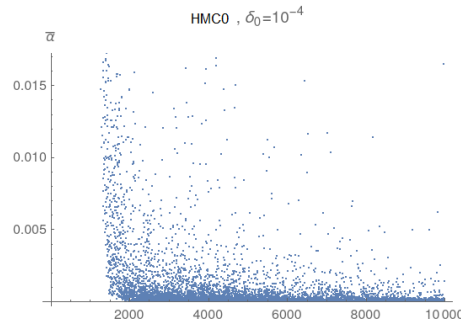


图 5.10 当时间步长设置为 10^{-4} 时经典 HMC 的平均接受概率变化趋势

如图 5.11 所示，本文提出的方法的平均接受概率较为均匀地分散在 0 和 1 之间。其原因在于，根据 Metropolis 算法，接受概率在理论上不等于常数 1。由于初始步长很小，初始接受概率接近于 1。经过几百次迭代，接受概率的值均匀分散开。这个例子中使用了 3 个粒子，预烧阶段为 5000 次迭代，导致在 5000 次

迭代时出现稍许变化。图中位于 0.33 和 0.66 有明显的聚集。其原因是，当其中一个粒子的接受概率接近 1，另外两个粒子的接受概率接近 0 时，平均接受概率为 0.33。本文提出通过调节整个系统的总能量使平均接受概率达到目标值或者目标区间。一个较小的接受概率意味着过大的动能，此时需要降低动能，反之亦然。

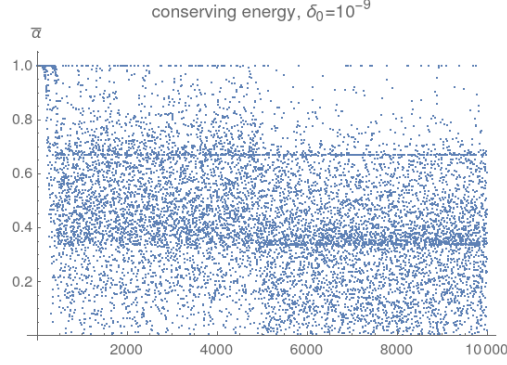


图 5.11 当时间步长设置为 10^{-9} 时 CHMC 方法的平均接受概率变化趋势

为了使能量渐进变化，有必要允许接受概率在一定范围内变化，如允许其分布于区间 $[0.1, 0.9]$ 内，仅当超出此区间才进行调节。

为了定量地研究本文方法的准确程度，可以采用拟合优度检验样本与正态分布的近似程度。由于抽样方法一般难以估计高相关的分布，因此可以用具有式(5.20)协方差矩阵的二元正态分布来研究抽样方法。我们需要从抽样样本获得两个主成分分量的样本，从而对每个主成分进行拟合优度检验。

$$\Sigma = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}, \quad \rho = 0.99999999 \quad (5.20)$$

如果 x 服从目标分布，则按式(5.21)计算得到的 y 服从标准正态分布。这种变换方式相当于对 Σ 求幂运算，以下简称为 **Eigen**。变换后 y 的主成分方向与 x 的主成分方向一致，但与坐标轴方向不一致。

$$y = \Sigma^{-0.5} x \quad (5.21)$$

为了使变换后的主成分方向与坐标轴方向一致，可以对 Σ 进行 Cholesky 分解，如式(5.22)所示。

$$\Sigma = L \cdot L^T \quad (5.22)$$

然后根据式(5.23)得到转换后样本的主成分方向与坐标轴一致，即转换后的样本的维度对应于主成分，因此可以分别对 z 的每一维进行拟合优度检验。

$$z = L^{-T} x \quad (5.23)$$

如果 x 服从目标分布，则 z 的每一维都服从标准正态分布。通过对 z 的每一维进行拟合优度检验，可以定量地衡量 x 在每一主成分上的抽样准确程度。

本文方法的抽样准确程度与动能 K_r 中的 r 有关。图 5.12 显示使用 $K_{0.2}$ 得到的抽样样本，以及使用特征分解和 Cholesky 分解进行变换后的结果。Cholesky 分解将原来蓝色样本的第一主成分（斜线方向）变换为绿色样本的 x_1 方向，将 x 的

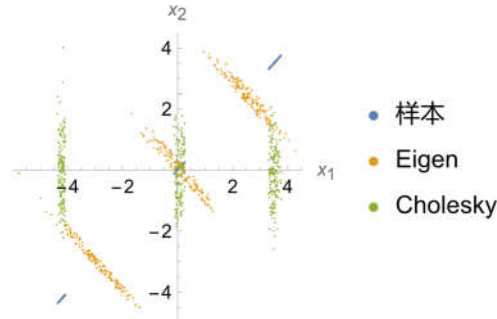


图 5.12 使用 $K_{0.2}$ 获得的二元正态分布样本及变换

第二主成分（反斜线方向）变换为绿色样本的 x_2 方向。

根据转换后的样本 z ，可以得到概率密度函数的平滑直方图，如图 5.13 所示。第二主成分即 x_2 方向大致服从标准正态分布，而第一主成分即 x_1 方向上与正态分布差别较大。

本文方法的抽样准确程度主要受到动能参数 r 的影响。图 5.14 显示使用三个粒子的情况下， $K_{0.5}$ 动能产生的抽样结果与正态分布近似。虽然 $K_{0.5}$ 似乎是最优动能，但通过定性实验，可以发现当粒子数目增加时，需要适当降低 r 的数值。

图 5.15 显示 $K_{0.5}$ 方法产生的两个主成分的平滑直方图与实线所示的正态分布较为接近。正如上述，调整 r 的数值有可能改善抽样效果，使实际分布更接近于目标（标准正态）分布。当粒子数增加时，这种调整所产生的效果将会更为明

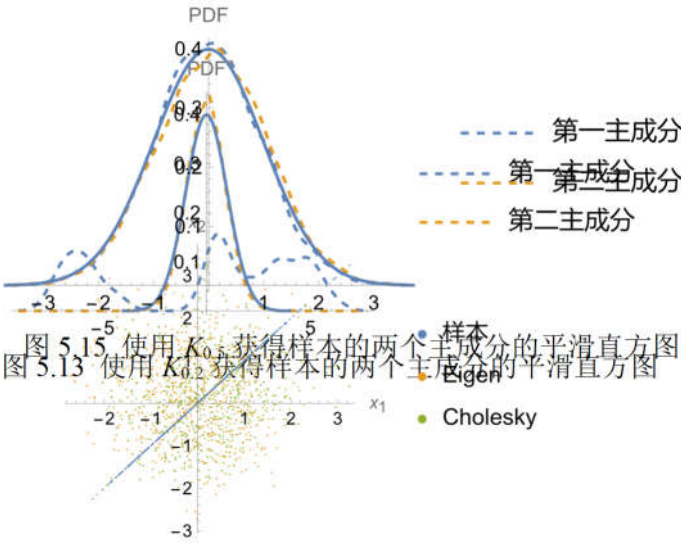


图 5.14 使用 $K_{0.5}$ 获得的样本及转换

显。

当相关程度较低时，现有的工具（PyMC3 和 STAN）能够相对准确地估计正态分布。当相关程度增加时，现有工具往往会失效。这些工具采用了较为复杂的参数调节算法，其抽样算法和采用的动能函数也与本文算法完全不同。此外，这些工具也可能采用单精度，而不是 Mathematica 所采用的双精度。尽管这些潜在问题都可能影响抽样的准确性，难以判断影响准确性的主要因素，但通过实验可以发现这些方法对于高相关的二元正态分布的第一主成分的准确性较低。

图 5.16 所示，PyMC3 在第一主成分上的抽样结果偏窄，这个结果与 STAN 的抽样结果类似。这说明二者可能采取了近似的抽样算法。尽管两种工具都能够相对准确地抽样第二主成分，但第一主成分较低的准确性影响了整体抽样准确性。

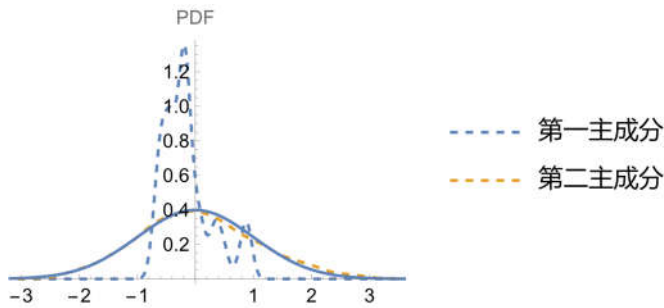


图 5.16 使用 PyMC3 获得的样本的两个主成分上的平滑直方图

为了定量地衡量抽样准确性，表 5.1 和表 5.2 列出了使用 5 种方法的拟合优度结果。这些方法给出的结果差别较大，而且不同的随机种子会产生不同的区别较大的结果，因此这些结果仅仅用来进行定性讨论。表 5.1 显示，本文的 $K_{0.5}$ 方法的拟合优度显著高于 PyMC3 和 STAN，而后两种方法的拟合优度近似为零。

表 5.2 显示，PyMC3 在第二主成分上的拟合优度远远高于 $K_{0.5}$ 和 STAN，而 STAN 的多数值大于 $K_{0.5}$ ，说明 PyMC3 较准确，而 $K_{0.5}$ 方法不准确。根据散点图和平滑直方图可知， $K_{0.5}$ 的抽样结果与目标分布接近，但定量结果显示其拟合优度可能较低。

表 5.1 第一主成分的拟合优度			
	$K_{0.5}$	PyMC3	STAN
Shapiro-Wilk	0.00149993	1.4414×10^{-13}	1.04822×10^{-24}
Anderson-Darling	0.179716	5.85266×10^{-8}	0
Cramer-von Mises	0.581271	0	0
Jarque-Bera ALM	0.120049	0	0
Pearson χ^2	0.00381389	7.23648×10^{-28}	3.74629×10^{-82}
表 5.2 第二主成分的拟合优度			
	$K_{0.5}$	PyMC3	STAN
Shapiro-Wilk	1.65648×10^{-6}	0.44927	0.00200402

Anderson-Darling	0.0000311679	0.342154	0.00643993
Cramer-von Mises	0.000550084	0.462619	0.0108183
Jarque-Bera ALM	0.00165605	0.552885	0.016453
Pearson χ^2	1.72353×10^{-7}	0.289838	0.39424

综上所述，对于高相关的分布，PyMC3 和 STAN 可能无法准确抽样所有的主成分。本文方法能够适用于高相关的分布，但拟合优度可能较低。定性实验显示，抽样准确性（和拟合优度）会受到参数 r 的影响，而 r 似乎与粒子数和相关程度（或主成分的尺度）有关。如何选取动能函数的参数 r 是一个尚未完全解决的问题。

5.6.2 环形分布

式(5.24)定义了一个半径为 10 标准差为 σ 的环形分布势能。该环形分布可以用来研究轨迹的特点和作用。不同的 σ 对应于不同的环形分布，用来研究仿真轨迹的抽样效果。由于环形分布无法再用主成分的方式加以描述，可以将欧氏坐标的抽样样本变换为极坐标，并研究在两个极坐标维度上的抽样准确性。降低 σ 参数会增大抽样难度。本实验采用 20 步仿真，不使用 K_q 。

$$P(x, y; \sigma) \propto e^{-\frac{(\sqrt{x^2+y^2}-10)^2}{2\sigma^2}} \quad (5.24)$$

对于环形分布， K_0 的轨迹主要沿着径向方向，意味着其在这个方向上的抽样更准确。与此形成对比的是 K_1 的轨迹主要沿着环形的周围移动，因此其在角度维上的抽样更为准确。显然 K_0 和 K_1 的仿真轨迹分别各自沿着一个方向，相互之间近似呈互补角度。RMHMC 原来的更新公式没有考虑 K_q ，其仿真轨迹逐渐向外侧移动，会导致抽样产生系统性误差。

计算 K_q 需要在二阶导数的基础上再求一次一阶导数，对于高维模型其计算量较大。因此在某些情况（如近似正态的低维模型）下可以忽略此项，因为对于

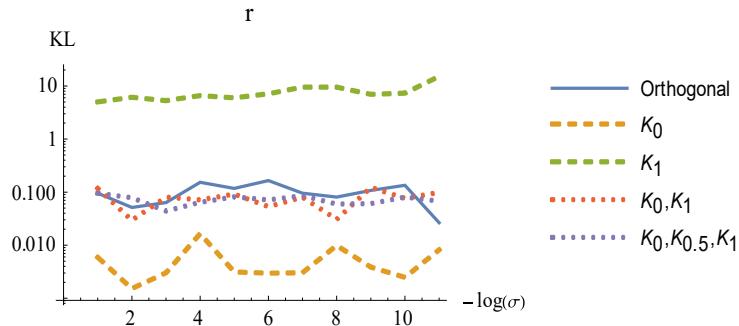


图 5.17 环形概率的径向抽样误差

多维正态分布，海森为常矩阵， K_q 为零向量。

图 5.17 显示径向维上的抽样误差。随着 σ 的降低， K_0 的误差维持在较低水

平，但此时 K_1 保持较大的误差。基于 K_1 的方法在径向维的误差较大。基于 K_0 的方法在径向误差最小，而正交轨迹方法和其它多轨迹方法的误差近似。由于二维模型有两个成分，正交轨迹方法和其它多轨迹方法都能较为完全地遍历空间，说明正交轨迹方法对于低维模型的优势不明显。

图 5.18 显示各种方法在角度维上的误差。 K_0 方法在角度维上随着 σ 的降低误差逐渐增加。本文方法和其它方法的误差随 σ 的变化趋势近似，都优于 K_0 方法。三轨迹方法在某些情况下的误差稍小于其它方法，但也可能是由随机因素造成的。这些方法都是基于同样的实现，因此对于低维模型其结果类似。由于所有方法的实验和实现采用相同基础，正交方法与非正交的多轨迹算法对于低维模型可能会产生类似的抽样效果。正交算法的优势在于多维模型。因为此时两轨迹和三轨迹算法可能无法有效遍历所有成分。

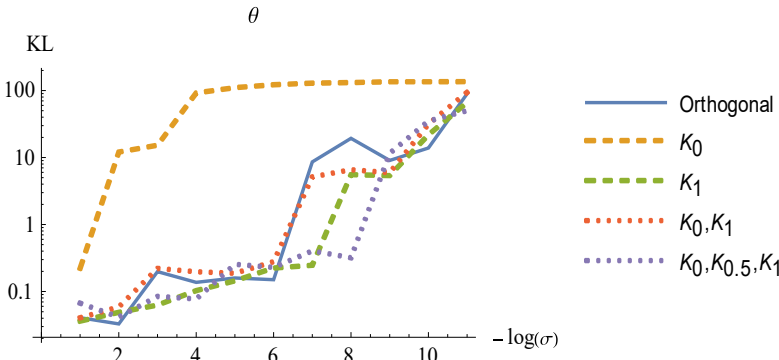


图 5.18 环形概率的角度抽样误差

5.6.3 多维正态分布

由于正态分布的均值和协方差矩阵参数可以根据样本进行估计，因此本文主要比较各个方法对于多维正态分布在各个主成分上的抽样准确性。我们使用一个无相关零均值的 10 维正态分布来测试各种抽样算法，第 i 维即第 i 个主成分的标准差设置为 base^{1-i} ，其中 base 从 1 变化到 12， i 从 1 到 10。 base 相当于指数函数的基数。如表 5.3 所示，当 base 为 1 时对应于标准正态分布，各维度的标准差相同。当 base 为 12 时，从第一维到第 10 维的标准差涵盖了 11 个数量级，因此抽样难度伴随 base 增加而迅速增加。一个正确的算法应当能够准确抽样这 12 种 10 维正态分布。

表 5.3 10 维无相关正态分布的第 i 维的标准差参数

base \ i	1	2	...	9	10
1	1	1	...	1	1
2	1	2^{-1}	...	2^{-8}	2^{-9}

...
12	1	12 ⁻¹	...	12 ⁻⁸	12 ⁻⁹

如式(5.25)所示，如果抽样正确，则转换后的样本应当服从 10 维标准正态分布，即标准差都应当为 1，因此转换后样本的标准差可以检验抽样是否准确。

$$\Sigma^{-\frac{1}{2}}X \sim N(0, 1) \quad (5.25)$$

单粒子系统与多粒子系统的实现区别较大，二者之间难以进行客观比较，因此所有方法都采用多粒子系统的实现方式。所有方法以相同的条件进行实验，即采用 3 个粒子，仿真步数为 3。 K_0 为 HMC 的动能，而 K_1 与 RMHMC 的动能类似，因此基于这两种动能的算法能揭示了 HMC 和 RMHMC 的主要特点。方法" $K_0, K_{0.5}, K_1$ "指的是在同一马氏链上依次运行三种抽样算法。

图 5.19 揭示算法抽样准确性，横坐标为 base，纵坐标为转换后的标准差，每种颜色的曲线对应一个主成分。当 base 小于 3 时，即主成分的尺度差别不大时，基于 K_0 的方法(相当于 HMC)能够相对准确地抽样后几个主成分，因为此时 σ 接近于 1。当 base 进一步增大时，多数主成分的样本标准差偏大。基于 K_1 方法的前几个主成分对于所有情况都比较准确，但排序靠后的主成分的样本标准差偏小，且越靠后的主成分偏小更严重。这说明 K_1 动能生成的轨迹在靠后主成分上更新幅度偏小。交替运行 K_0 和 K_1 的双轨迹算法的效果有所改善。双轨迹算法仅对于 base 小于 6 的情况较为准确。当 base 较大时，靠近前边和后边几个主成分的抽样较为准确，但靠近中间的主成分的抽样误差较大，说明这种方法仍有改进的余地。由于 $K_{0.5}$ 对于所有的 base 值和所有主成分都比较准确，因此这种动能最适于正态分布。如果使用 $K_{0.5}$ 的同时还考虑其它动能，例如 K_0 和 K_1 ，那么准确性将有所降低。这说明使用更多轨迹的算法未必效果更好。正交轨迹算法属

于多轨迹算法，没有正态分布的假设，而且其抽样也比较准确。

潜在比例折减系数（Potential Scale Reduction Factor, PSRF）是一种衡量多条马氏链是否收敛的指标，如果数值接近于 1，说明收敛。图 5.20 显示基于 K_0 的算法（例如 HMC）对于标准正态分布的收敛效果较好。但是，伴随着 base 的

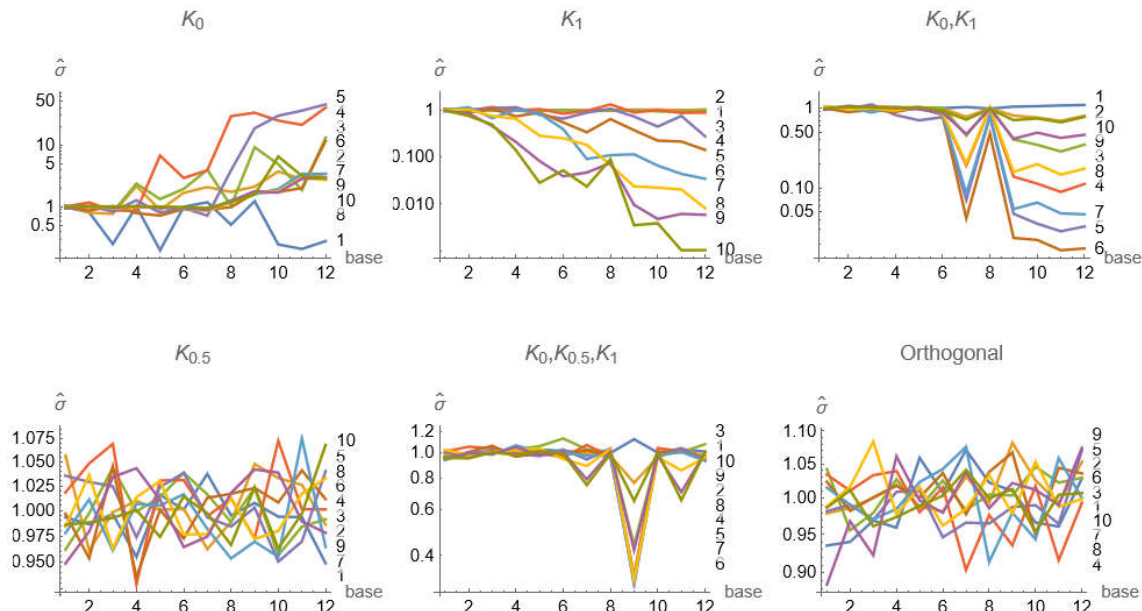


图 5.19 转换后的样本的标准差，侧边为主成分序号

增加，其收敛状况迅速变差。尽管基于 K_1 的方法（如 RMHMC）对于大多数情

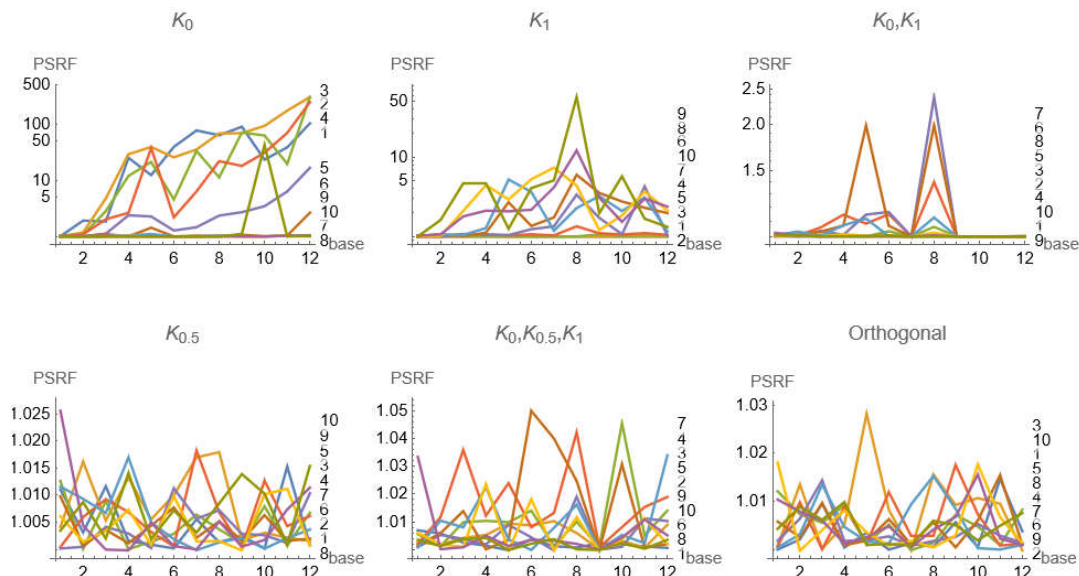


图 5.20 潜在比例折减系数(PSRF)

况仍然未收敛，较之 K_0 具有明显改善。使用 K_0 和 K_1 的双轨迹算法的收敛有了进一步改善，但仍然未满足收敛判据。基于 $K_{0.5}$ 的方法对于所有的 base 值都实现了收敛，说明这种方法在所有主成分上都实现了较为充分的遍历。如果在算法

中使用 K_0 、 $K_{0.5}$ 和 K_1 动能，其收敛状况比 $K_{0.5}$ 的收敛状况稍差。正交轨迹方法也满足收敛判据。

自相关函数能够衡量一个马氏链相邻样本之间的近似程度，如果数值下降较快，说明抽样效率较高。图 5.21 显示，当 base 等于 6 时，基于 K_0 的方法对于排序靠后的主成分的抽样效率较高，这个特点与 K_1 方法几乎正好相反。由于基于 K_0 或者 K_1 动能的方法都存在抽样效率较低的主成分，这两种方法的总体收敛均较慢。在同一马氏链上运行两种方法可以改善抽样效率，此时所有主成分上的自相关函数均能够快速衰减。基于 $K_{0.5}$ 的方法在所有主成分上的抽样效率较高且近似，说明这种方法能够同时充分遍历所有主成分。如果将 $K_{0.5}$ 和其它动能（如 K_0 和 K_1 ）一起使用，那么抽样效率反而会降低。正交轨迹方法的自相关函数与 $K_{0.5}$

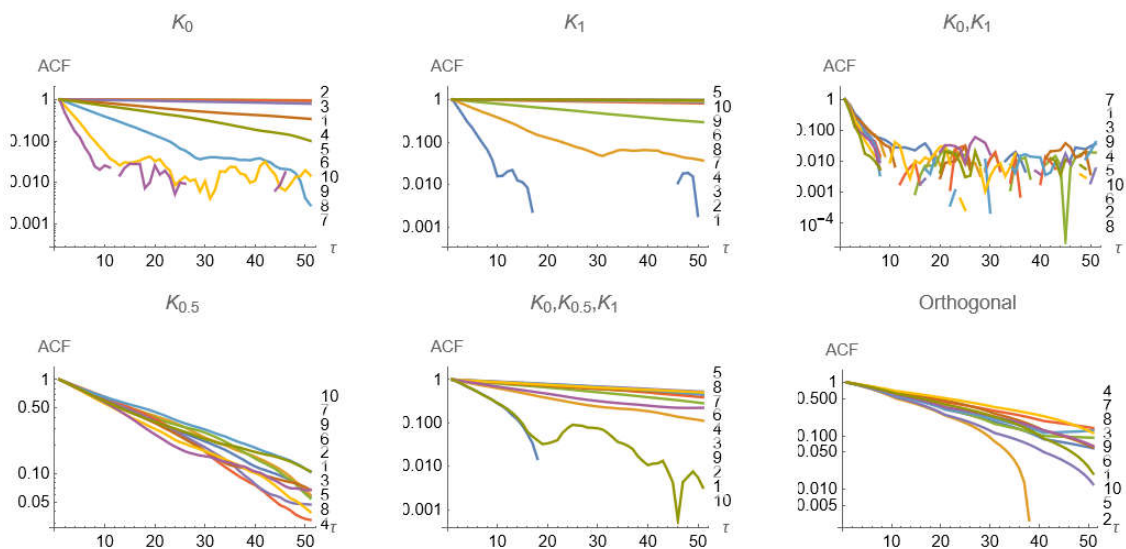


图 5.21 自相关函数

的情况近似，其抽样效率也较高。

基于多维正态的实验显示， K_0 的样本方差偏大。 K_1 的样本方差偏小，而且越靠后的主成分偏小越严重。本人提出了几种多轨迹算法，其中的双轨迹方法虽然有所改善，但其样本方差仍然偏小，说明仍有待改进。基于 $K_{0.5}$ 的方法和正交多轨迹方法在所有主成分上的表现均较好，其准确性、收敛性和抽样效率较 K_0 和 K_1 有较大改善。

表 5.4 几种抽样方法的比较

特点 \ 动能	K_0	$K_{0.5}$	K_1	K_0, K_1	$K_0, K_{0.5}, K_1$	正交
样本方差	偏高	正常	偏低	略偏低	略偏低	正常
准确性	低	高	低	略低	略高	高
收敛	差	好	差	中	好	好
抽样效率	低	高	低	高	中	高

如表 5.4 所示， $K_{0.5}$ 和正交方法的样本方差较为准确，收敛状况较好，并且

抽样效率较高。 K_0 和 K_1 方法的样本方差不准确, 收敛状况较差, 且抽样效率较低, 说明这两种方法在应用中可能会出现问题。多轨迹方法优于 K_0 和 K_1 方法, 并且如果使用了 $K_{0.5}$, 其收敛效果将有所改善。

5.6.4 参数估计实例

除了多元正态分布模型, 基于抽样的参数估计方法的主要优点是可以应用于非正态概率模型。这里介绍一个贝塔分布的应用实例。设两位裁判分别给出 0.7 和 0.125 的评分, 分数范围介于 0 和 1 之间。由于资源的限制, 无法任意增加裁判的数量。裁判之间需要消除相互影响, 以防评分数据违反独立同分布假设。此外, 裁判还可能受到其它因素的影响。这些潜在的问题都会使得数据偏离模型假定的理想状态。参数估计的目的是根据这些理想(或非理想)的数据估计概率模型的参数, 以及进行统计推断。由于多种主客观因素的影响, 评分可能呈现两极化状态、单极化状态、或均匀分布状态。根据以上描述, 可以假设评分服从式(5.26)所示的贝塔分布, 其参数 α 和 β 决定了分布的形状。我们希望使用评分观测值估计 α 和 β 的分布, 进而估计评分的分布。

$$s_i \sim \text{Beta}(\alpha, \beta), i = 1, 2 \quad (5.26)$$

如果我们有一些评分的历史数据, 则可以据此构造参数的先验分布。反之, 如果没有评分的历史数据, 在不知道进一步信息的情况下, 先假设 α 和 β 服从正无限区间上的均匀分布, 即式(5.27)。

$$\alpha, \beta \sim U(0, \infty) \quad (5.27)$$

如式(5.28)所示, 有了数据分布和参数的先验分布, 根据贝叶斯公式, 可以得到参数的后验分布。

$$p(\alpha, \beta | s_1, s_2) \propto p(\alpha, \beta) \prod_{i=1}^2 p(s_i | \alpha, \beta) \quad (5.28)$$

最后, 根据参数的后验分布和评分分布可以估计以观测值为条件的评分的后验分布, 即式(5.29)所示。

$$p(s | s_1, s_2) = \int p(s | \alpha, \beta) p(\alpha, \beta | s_1, s_2) d\alpha d\beta \quad (5.29)$$

由于收集数据可能消耗大量资源, 因此参数估计和统计推断过程就需要尽量准确。参数的真实值或分布属于未知状态, 无法根据"最优"参数进行比较实验。对于这种使用实际数据的参数估计问题, 如果多种方法的样本都收敛到同一分布, 则可以定性地认为所有方法都获得了正确的结果。我们可以运行多种实现方法, 然后比较其样本分布。因此, 我们主要关心本文方法的结果是否与其它已有方法的结果近似。

图 5.22 显示使用几种本文实现的方法（实线）和一些已有方法获得的后验评分分布的平滑直方图。Metropolis、NUTS、HMC 和 Slice 方法由 PyMC3 工具实现，最后一种方法由 STAN 实现。对于这个例子，本文实现的 K_0 方法产生的结果与其它方法较为近似，都呈现单极形式。但本文提出的 $K_{0.5}$ 方法、 K_1 方法和 K_{orth} （即正交轨迹）方法与所有已有方法的区别均较大。而且 $K_{0.5}$ 方法和 K_{orth} 方法的结果较为近似，都呈现两极形态。根据上述针对多元正态分布实验的讨论中可知， $K_{0.5}$ 方法和 K_{orth} 方法应当比较准确。但在这个例子中，这两种方法的结果却与 K_0 方法和已有方法不同。于是产生了疑问： K_0 方法还是 $K_{0.5}$ 方法更准确？

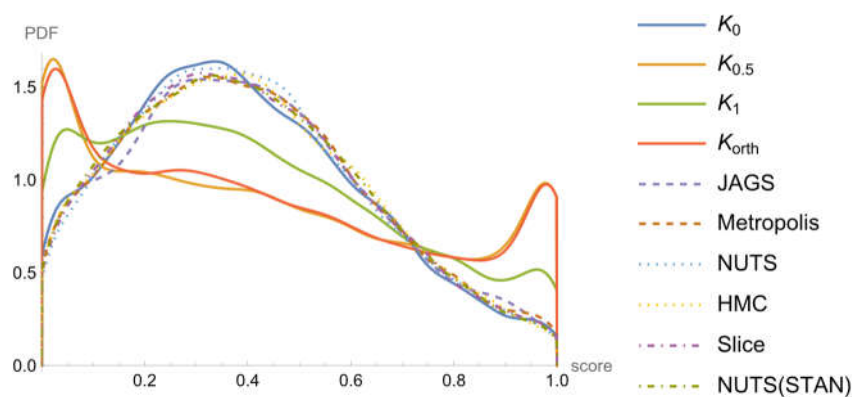


图 5.22 α 和 β 为均匀分布先验时评分的平滑直方图

本文方法的效果受到参数 r 的影响。 $K_{0.5}$ 对于正态分布的效果较好。但是对于这个实例， $K_{0.1}$ 的效果似乎更好。图 5.23 显示，（未归一化的）参数后验概率密度函数和 $K_{0.1}$ 方法产生的参数样本的平滑直方图的匹配程度较好。如果采用其它的动能，如 K_0 或者 $K_{0.5}$ ，则两个分布的匹配程度均会变差，其中 $K_{0.5}$ 的参数样

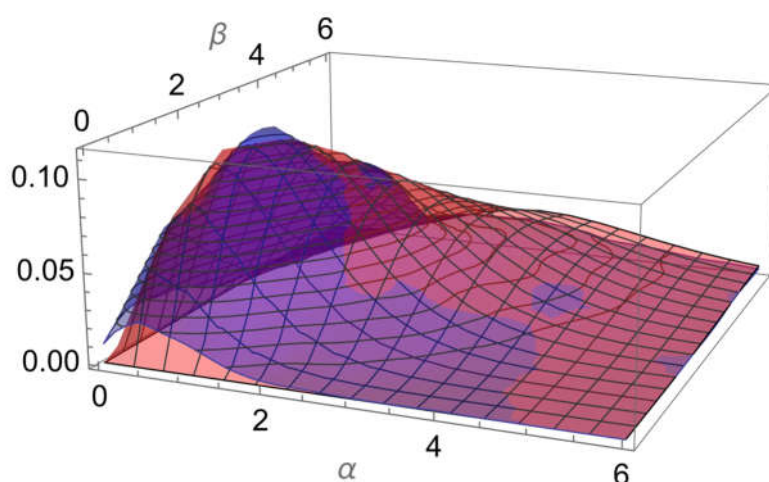


图 5.23 红色：（未归一化的）参数后验概率密度函数
蓝色： $K_{0.1}$ 方法产生的样本的平滑直方图散点图

本偏小。而 K_0 的参数样本偏大。

图 5.24 和图 5.25 显示参数 α 和 β 的后验样本分布，短横线为 K_0 和已有方法， $K_{0.1}$ 为实线，其它 K_r 方法用虚线表示。 $K_{0.5}$ 和已有结果近似。对于 K_r 动能，当 r 增加时，参数偏小，而当 r 减小时，参数偏大。这个实例的概率密度函数为单峰函数，最大概率位置的 α 和 β 分别大致为 1 和 1.6。 K_0 方法和已有方法的样本分布的峰值都偏大，而 $K_{0.5}$ 方法的样本分布的峰值偏小，因此这些方法都不十分准确。正如前文所提到的，如何根据模型和应用选择选择的参数 r 仍然有待解

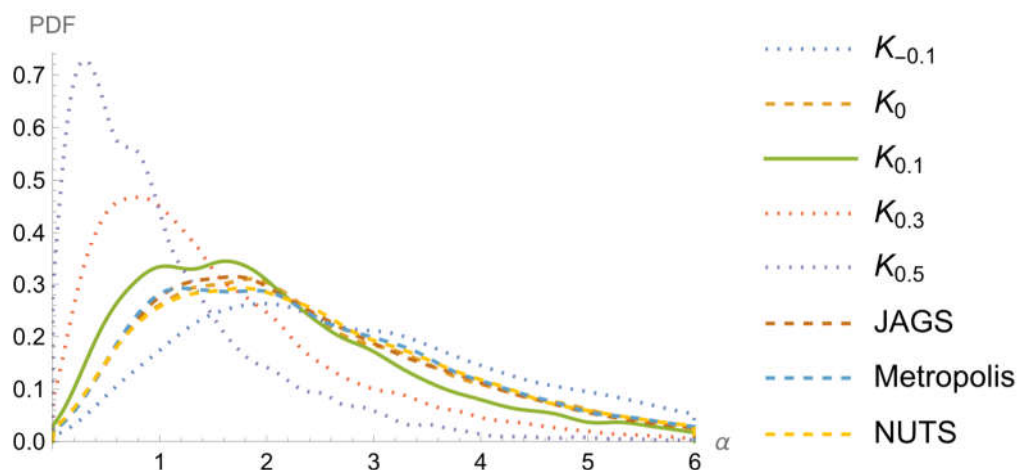


图 5.24 α 的后验样本分布

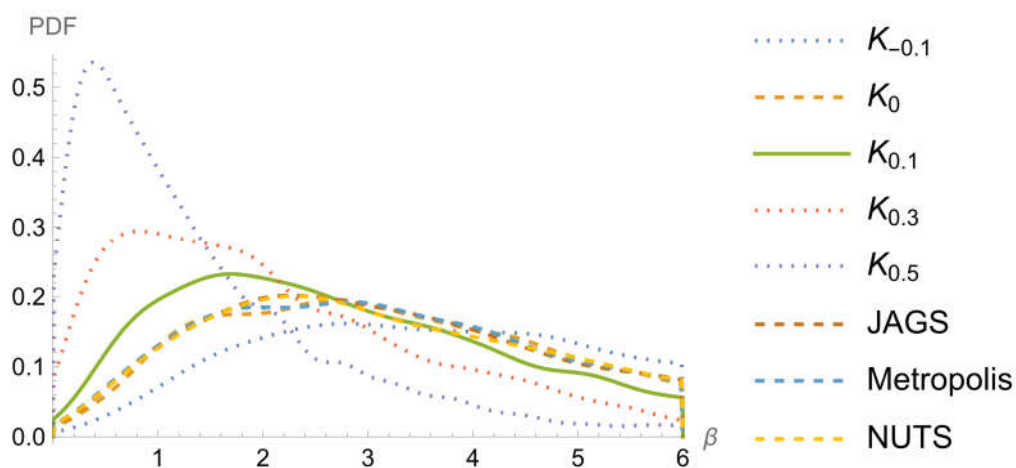


图 5.25 β 的后验样本分布

决。

5.7 本章小结

本章介绍了改进的哈密顿蒙特卡洛抽样算法。改进方案主要体现在能量守恒、

多粒子系统以及改进的动能及仿真轨迹。采用上述改进措施能够改善抽样结果。通过固定仿真步数，可以有效降低运算量。通过多个粒子的势能极值在轨迹中出现的位置可以判断步长过大、过小还是适中，从而实现对步长的调节。通过多粒子的平均接受概率判断总动能过大还是过小，从而调节总能量，进而使平均接受概率分布在适当范围。基于 $K_{0.5}$ 的方法能够较为准确和稳定地抽样多维正态分布。

附 录

以下部分描述了矩阵的特征值和特征向量相对于单变量求导的方法，这部分内容来源于互联网上对该问题的解答^[84]。

假设矩阵 B 依赖于一个单独的变量 t ，那么相对于等式两边求导

$$Bn_i = \lambda_i n_i \quad (\text{a.1})$$

我们得到

$$\dot{B}n_i + B\dot{n}_i = \dot{\lambda}_i n_i + \lambda_i \dot{n}_i \quad (\text{a.2})$$

我们这里假设 $\|n_i\|=1$ 。于是对于任意 i ，有 $\dot{n}_i \perp n_i$ 。用 n_i 与上式进行内积运算，并注意到

$$(B\dot{n}_i, n_i) = (\dot{n}_i, Bn_i) = \lambda_i (\dot{n}_i, n_i) = 0 \quad (\text{a.3})$$

于是得到

$$\dot{\lambda}_i = (\dot{B}n_i, n_i) \quad (\text{a.4})$$

因此得到以 \dot{B} 表示 $\dot{\lambda}_i$ 的方式，即特征值相对于变量 t 的导数。

下一步，我们用 n_j 对式(a.2)进行内积运算，其中 j 不等于 i 。利用 B 的对称性可以得到

$$(\dot{B}n_i, n_j) + (\dot{n}_i, Bn_j) = \lambda_i (\dot{n}_i, n_j) \quad (\text{a.5})$$

于是

$$(\dot{B}n_i, n_j) + \lambda_j (\dot{n}_i, n_j) = \lambda_i (\dot{n}_i, n_j) \quad (\text{a.6})$$

这说明

$$(\dot{n}_i, n_j) = \frac{1}{\lambda_i - \lambda_j} (\dot{B}n_i, n_j) \quad (\text{a.7})$$

因此得到用矩阵 B 相对于 t 的导数来表示的特征向量相对于变量 t 的导数：

$$\dot{n}_i = \sum_{i \neq j} \frac{1}{\lambda_i - \lambda_j} (\dot{B}n_i, n_j) n_j \tag{a.8}$$