

# Numerical Analysis

Zhiping Li

LMAM and School of Mathematical Sciences  
Peking University



# Metropolis 算法所产生的马氏链所应具有的性质

Metropolis 算法要求所产生的马氏链的转移概率矩阵  $\mathbf{P}$  具有如下性质:

- ① 转移概率矩阵  $\mathbf{P}$  满足: (i)  $p_{ij} \geq 0, \forall i, j = 1, \dots, N_t$ ; (ii)  $\sum_{j=1}^{N_t} p_{ij} = 1, \forall i = 1, \dots, N_t$ .
- ② 转移概率矩阵  $\mathbf{P}$  是本原矩阵. 此时以  $\mathbf{P}$  为转移概率矩阵的马氏链是本原的 (遍历的).
- ③ 微观态  $\sigma^{(i)}$  的出现概率  $\frac{1}{Z_M} \exp\{-\beta H(\sigma^{(i)})\}$  排成的  $N_t$  维行向量  $\pi$  是转移概率矩阵  $\mathbf{P}$  的不变分布.
- ④ 由转移概率矩阵  $\mathbf{P}$  定义的马氏链满足细致平衡条件, 即

$$\pi(\sigma)P(\sigma \rightarrow \sigma') = \pi(\sigma')P(\sigma' \rightarrow \sigma).$$

注: 上式两端对  $\sigma'$  求和得  $\pi = \pi\mathbf{P}$ . 即由 (4) 可得 (3).



# Metropolis 算法的目标 — 给出保证性质 (1)-(4) 的算法

转移概率矩阵  $\mathbf{P}$  共有  $N_t^2$  个分量, 性质 (1) 和 (4) 分别提出  $N_t$  和  $N_t(N_t - 1)/2$  个独立的约束条件. 可见, 还有相当多的自由度来选择转移概率矩阵  $\mathbf{P}$ , 不同的选择方法对应不同的算法.

Metropolis 算法并不直接构造转移概率矩阵  $\mathbf{P}$ , 而是通过数值模拟一个特定的满足细致平衡条件的马氏链的发展过程, 从而产生一个状态序列  $\sigma^{(1)}, \sigma^{(2)}, \dots, \sigma^{(n)}$ . 发展过程的每一次实现分为彼此独立的两步: 第一步, 产生一个预选态  $\sigma'$ ; 第二步, 判断是否接受 (以多大的概率接受) 该新状态, 如果接受, 则  $\sigma^{(n+1)} = \sigma'$ , 否则,  $\sigma^{(n+1)} = \sigma^{(n)}$ .

不同的预选状态的产生方法给出不同的算法, 这些算法计算复杂度可能会大不相同. 我们给出两种简单直观的预选状态产生方法.



## 第一步：两种简单直观的预选状态产生方法 (一维 Ising 模型)

设(一维 Ising 模型)当前状态为  $\sigma = (\sigma_1, \dots, \sigma_M)$ , 按以下方法

- 方法 1: 从其余的  $N_t - 1$  个状态中按均匀分布随机产生;
- 方法 2: 从  $M$  个网格点中按均匀分布随机挑选一个网格点, 并将该格点的当前自旋状态反转 (即  $1 \rightleftharpoons -1$ );

产生一个预选状态  $\sigma'$ .

**注:** 方法 1 产生新状态后, 能量函数  $H(\sigma')$  与  $H(\sigma)$  毫无关系, 因此要完全重新计算, 因此, 计算量较大; 而方法 2 则只改变了一个分量, 因此新状态的能量函数只需在原状态的能量函数的基础上做一个简单修正, 例如, 反转的是第  $i$  个格点上的自旋状态, 则只需更新第  $i-1$  和第  $i+1$  号格点与第  $i$  号格点的作用能量的变化, 从而大大节约了计算量.



## 两种预选状态产生方法所定义的预选马氏链的转移概率矩阵

从任一初始状态出发, 两种预选状态产生方法各产生一个预选状态马氏链, 它们的转移概率矩阵  $G \in \mathbb{R}^{N_t \times N_t}$  分别为:

- 方法 1 (等可能预选<sub>(满足  $G^2 > 0$ )</sub>):

$$G(\sigma, \sigma') = \begin{cases} \frac{1}{N_t - 1}, & \sigma \neq \sigma', \\ 0, & \sigma = \sigma'; \end{cases}$$

- 方法 2 (单点反转预选<sub>(满足  $G^M > 0$ )</sub>):

$$G(\sigma, \sigma') = \begin{cases} \frac{1}{M}, & \sigma \text{ 与 } \sigma' \text{ 恰有一个格点取值不同,} \\ 0, & \text{其它.} \end{cases}$$

注: 这两个预选转移概率矩阵  $G$  都是对称的本原矩阵, 且满足:

(1)  $G(\sigma, \sigma') \geq 0, \forall \sigma, \sigma' \in \Omega$ , (2)  $\sum_{\sigma' \in \Omega} G(\sigma, \sigma') = 1, \forall \sigma \in \Omega$ .



# Metropolis 算法的核心 — 第二步：接受新状态的概率

接受新状态的规则必须使所产生的马氏链满足细致平衡条件，这也将同时保证  $\pi$  是转移概率矩阵  $\mathbf{P}$  的不变分布。

设  $\sigma$  是当前状态,  $\sigma'$  是预选态. 记  $\Delta H = H(\sigma') - H(\sigma)$ , 令

$$R = \frac{\pi(\sigma')}{\pi(\sigma)} = \exp\{-\beta\Delta H\}.$$

接受新状态的规则:

- 如果  $R > 1$ , 则接受  $\sigma'$  作为新状态, 此时有  $\Delta H < 0$ , 即预选状态具有较低的能量值;
- 如果  $R \leq 1$ , 则以概率  $R$  接受  $\sigma'$  作为新状态, 此时有  $\Delta H \geq 0$ , 即预选状态具有较高能量值, 且  $R$  越小能量越高.



## Metropolis 算法的转移概率矩阵 $\mathbf{P}$

将 Metropolis 算法的第一步和第二步结合起来, 并注意到两步之间在概率上是相互独立的, 我们就得到了 Metropolis 算法的转移概率矩阵  $\mathbf{P}$  的定义

$$P(\sigma, \sigma') = \begin{cases} G(\sigma, \sigma') \frac{\pi(\sigma')}{\pi(\sigma)}, & \sigma \neq \sigma' \text{ 且 } \pi(\sigma') < \pi(\sigma), \\ G(\sigma, \sigma'), & \sigma \neq \sigma' \text{ 且 } \pi(\sigma') \geq \pi(\sigma), \\ 1 - \sum_{\tau \neq \sigma} P(\sigma, \tau), & \sigma = \sigma'. \end{cases}$$

显然, 矩阵  $\mathbf{P}$  满足: (i)  $p_{ij} \geq 0, \forall i, j = 1, \dots, N_t$ ; (ii)  $\sum_{j=1}^{N_t} p_{ij} = 1, \forall i = 1, \dots, N_t$ , 且为本原的 (由  $\mathbf{G}$  为本原的).



# Metropolis 算法的转移概率矩阵 $\mathbf{P}$ 满足 (1)–(4)

Metropolis 算法的转移概率矩阵  $\mathbf{P}$  显然满足 (1) 和 (2). 又因为 (3) 是 (4) 的推论, 因此只需验证 (4), 即细致平衡条件.

对任意的  $\sigma \neq \sigma'$ , 由定义和  $G$  的对称性有

$$\begin{aligned}\pi(\sigma)P(\sigma, \sigma') &= G(\sigma, \sigma') \min\{\pi(\sigma), \pi(\sigma')\} \\ &= G(\sigma', \sigma) \min\{\pi(\sigma), \pi(\sigma')\} = \pi(\sigma')P(\sigma', \sigma),\end{aligned}$$

即 Metropolis 算法的转移概率矩阵  $\mathbf{P}$  满足细致平衡条件.



# Metropolis 算法的流程

## Metropolis 算法：

- ① 设定初始态  $\sigma^{(1)}$  和总计算步数  $N$ .
- ② 根据一定的预选规则由  $\sigma^{(n)}$  产生预选态  $\sigma'$ .
- ③ 计算  $\Delta H = H(\sigma') - H(\sigma)$ , 计算

$$A = \min\{1, R\} = \begin{cases} 1, & \Delta H \leq 0, \\ \exp\{-\beta\Delta H\}, & \text{其它}. \end{cases}$$

- ④ 生成一均匀分布随机数  $r \sim \mathcal{U}[0, 1]$ ;
- ⑤ 如果  $r \leq A$ , 则令  $\sigma^{(n+1)} = \sigma'$ ; 否则, 令  $\sigma^{(n+1)} = \sigma^{(n)}$ .
- ⑥ 如果  $n + 1 < N$ , 转步 2; 否则计算  $\frac{1}{N} \sum_{i=1}^N H(\sigma^{(i)})$ .

注：若在第三步中取  $A = (1 + \exp\{\beta\Delta H\})^{-1}$ , 则相应的算法称为 Glauber 算法. 该算法也产生满足细致平衡条件的本原马氏链.



# 大规模非凸优化问题及其随机性算法

实际应用中有大量非凸优化问题. 传统的数值方法往往无力解决该类问题, 尤其是当问题的规模很大时. 以下是两个经典的例子.

## 例 1: 旅行经销商问题 (Traveling Salesman Problem, TSP)

假设有  $N$  个不同的城市, 两两之间有唯一路径直通 ( $l_{ij} = l_{ji}$ ), 试寻找一条贯穿所有城市的路径, 使得每一城市必须且仅通过一次, 且总路程最短. 该问题可表述为组合优化问题:

$$\mathbf{x} = \arg \min_{\mathbf{x} \in X} \left\{ H(\mathbf{x}) = \sum_{i=1}^{N-1} l_{x_i x_{i+1}} \right\},$$

其中  $X = \{(x_1, \dots, x_N) : x_1, \dots, x_N \text{ 是 } 1, \dots, N \text{ 的一个排列}\}$ .  $X$  中共有  $\frac{N!}{2}$  条不同的路径, 当  $N$  增加时, 总路径数呈指数性增长, 且路径函数  $H(\mathbf{x})$  无规律可循. 因此, 这是一个典型的 NP hard problem.



# 大规模非凸优化问题及其随机性算法

## 例 2: 图像磨光问题

设一幅图像共有  $J$  个像素点, 每个像素点上有 256 种颜色. 记

$$X = \{(x_1, \dots, x_J) : x_i \in \{0, 1, \dots, 255\}, 1 \leq i \leq J\}.$$

一幅有  $J$  个像素点的图像  $\mathbf{x} \in X$  的光滑度定义为

$$H(\mathbf{x}) \triangleq \alpha \sum_{\langle s, t \rangle} (x_s - x_t)^2, \quad \alpha > 0 \text{ 为常数},$$

这里  $\sum_{\langle s, t \rangle}$  表示对相邻像素点求和. 对给定图像  $\mathbf{y} \in X$  定义

$$H(\mathbf{x}|\mathbf{y}) \triangleq \alpha \sum_{\langle s, t \rangle} (x_s - x_t)^2 + \frac{1}{2\sigma^2} \sum_{s=1}^J (x_s - y_s)^2, \quad \sigma \text{ 为常数}.$$

设  $\mathbf{y} \in X$  是被噪声污染的图像, 为去噪, 可考虑以下优化问题:

$$\mathbf{x} = \arg \min_{\mathbf{x} \in X} H(\mathbf{x}|\mathbf{y}).$$

与 TSP 类似, 这也是一个典型的 NP hard problem.



## 模拟退火算法所依据的基本事实

对于优化问题

$$\mathbf{x} = \arg \min_{\mathbf{x} \in X} H(\mathbf{x}),$$

定义  $H(\mathbf{x})$  的全局极小点集  $M = \{\mathbf{x}_0 : H(\mathbf{x}_0) = \min_{\mathbf{x} \in X} H(\mathbf{x})\}$ ,  
引入参数  $\beta$ , 定义概率密度函数

$$\Pi^\beta(\mathbf{x}) = \frac{1}{Z_\beta} \exp\{-\beta H(\mathbf{x})\}, \quad Z_\beta = \sum_{\mathbf{x} \in X} \exp\{-\beta H(\mathbf{x})\},$$

相应分布称为  $\Pi^\beta(\mathbf{x})$  分布.

**定理:** 记  $|M|$  为集合  $M$  中元素的个数, 则  $\Pi^\beta(\mathbf{x})$  有性质

$$\lim_{\beta \rightarrow +\infty} \Pi^\beta(\mathbf{x}) = \Pi^\infty(\mathbf{x}) \triangleq \begin{cases} \frac{1}{|M|}, & \mathbf{x} \in M, \\ 0, & \mathbf{x} \notin M, \end{cases}$$

且当  $\beta$  充分大时, 对任一  $\mathbf{x} \in M$ ,  $\Pi^\beta(\mathbf{x})$  作为  $\beta$  的函数单调增;  
对任一  $\mathbf{x} \notin M$ ,  $\Pi^\beta(\mathbf{x})$  作为  $\beta$  的函数单调减.



## 模拟退火算法所依据的基本事实的证明

证明：记  $m = \min_{\mathbf{x} \in X} H(\mathbf{x})$ , 则由

$$\lim_{\beta \rightarrow +\infty} \exp\{-\beta[H(\mathbf{x}) - m]\} = \begin{cases} 1, & \mathbf{x} \in M, \\ 0, & \mathbf{x} \notin M, \end{cases}$$

和

$$\Pi^\beta(\mathbf{x}) = \frac{\exp\{-\beta[H(\mathbf{x}) - m]\}}{\sum_{\mathbf{z} \in M} \exp\{-\beta[H(\mathbf{z}) - m]\} + \sum_{\mathbf{z} \notin M} \exp\{-\beta[H(\mathbf{z}) - m]\}},$$

得

$$\lim_{\beta \rightarrow +\infty} \Pi^\beta(\mathbf{x}) = \begin{cases} \frac{1}{|M|}, & \mathbf{x} \in M, \\ 0, & \mathbf{x} \notin M. \end{cases}$$



## 模拟退火算法所依据的基本事实的证明 (续)

另一方面, 若  $\mathbf{x} \in M$ , 则有

$$\Pi^\beta(\mathbf{x}) = \frac{1}{|M| + \sum_{\mathbf{z} \notin M} \exp\{-\beta[H(\mathbf{z}) - m]\}},$$

因此,  $\Pi^\beta(\mathbf{x})$  是  $\beta$  的单调增函数.

而若  $\mathbf{x} \notin M$ , 记  $\tilde{Z}_\beta = \sum_{\mathbf{z} \in X} \exp\{-\beta[H(\mathbf{z}) - m]\}$ , 则有

$$\begin{aligned} \frac{\partial \Pi^\beta(\mathbf{x})}{\partial \beta} &= \frac{\exp\{-\beta[H(\mathbf{x}) - m]\}}{\tilde{Z}_\beta^2} \left[ (m - H(\mathbf{x})) \tilde{Z}_\beta \right. \\ &\quad \left. - \sum_{\mathbf{z} \in X} \exp\{-\beta[H(\mathbf{z}) - m]\} (m - H(\mathbf{z})) \right]. \end{aligned}$$

$$\begin{aligned} \text{而} \quad \lim_{\beta \rightarrow +\infty} &\left[ (m - H(\mathbf{x})) \tilde{Z}_\beta - \sum_{\mathbf{z} \in X} \exp\{-\beta[H(\mathbf{z}) - m]\} (m - H(\mathbf{z})) \right] \\ &= |M|[m - H(\mathbf{x})] < 0. \end{aligned}$$



# 模拟退火算法的基本流程

设给定正单调增序列  $\{\beta_n\}_{n=0}^{\infty}$ ,  $\lim_{n \rightarrow \infty} \beta_n = +\infty$ .

模拟退火算法:

- ① 设定初始值  $n = 0$ ,  $\mathbf{x}_0$ ,  $\beta = \beta_0$ , 内循环步数  $K$  和迭代步数  $N$ .
- ② 根据一定的预选规则由  $\mathbf{x}_k$  产生预选态  $\mathbf{x}'$ .
- ③ 计算  $\Delta H = H(\mathbf{x}') - H(\mathbf{x}_k)$ , 计算

$$A = \min\{1, R\} = \begin{cases} 1, & \Delta H \leq 0, \\ \exp\{-\beta \Delta H\}, & \text{其它.} \end{cases}$$

- ④ 生成一均匀分布随机数  $r \sim \mathcal{U}[0, 1]$ ;
- ⑤ 如果  $r \leq A$ , 则令  $\mathbf{x}_{k+1} = \mathbf{x}'$ ; 否则, 令  $\mathbf{x}_{k+1} = \mathbf{x}_k$ .
- ⑥ 如果  $k + 1 < K$ , 转步 2; 否则,  $\mathbf{x}_0 := \mathbf{x}_K$ ,  $k := 0$ .
- ⑦ 如果  $n < N$ ,  $n := n + 1$ ,  $\beta = \beta_n$ , 转步 2; 否则, 输出结果.



# 模拟退火算法的形式收敛性

以上模拟退火算法的核心部分, 即第二到第六步, 实际上就是 Metropolis 算法的核心部分. 因此, 只要  $K$  充分大,  $\mathbf{x}_K$  就会依概率收敛于  $\Pi^\beta(\mathbf{x})$  的不变分布.

另一方面, 由定理 7.5.1

$$\lim_{\beta \rightarrow +\infty} \Pi^\beta(\mathbf{x}) = \Pi^\infty(\mathbf{x}) \triangleq \begin{cases} \frac{1}{|M|}, & \mathbf{x} \in M, \\ 0, & \mathbf{x} \notin M, \end{cases}$$

即, 当  $N$  充分大时,  $P_{\Pi^{\beta_N}(\mathbf{x})}\{\mathbf{x} | \mathbf{x} \in M\} \approx 1$ . 因而此时有

$$P\{\mathbf{x}_N \in M\} \approx 1.$$

注: 模拟退火算法形式上模拟了冶炼中的退火工艺, 理论和实践均表明收敛性的关键在于控制  $\beta_n$  趋于无穷的速度.



# 模拟退火算法收敛性的基本定理

为简单起见, 内循环次数  $K$  甚至可以取 1. 此时有收敛性结果:

**定理:** 设  $X$  为一有限集,  $H(\mathbf{x})$  为  $X$  上的非常数函数,  $\mathbf{G}$  是以  $\mathbf{G}(\mathbf{x}, \mathbf{y})$  为元素的对称不可约预选阵,  $\mathbf{P}^\beta$  为由  $\mathbf{G}$  和相应的参数为  $\beta$  的满足细致平衡条件的规则所定义的转移概率矩阵. 如果退火速度  $\beta_n \leq C \ln n$ , 其中  $C = C(\mathbf{G}, H)$  为常数, 则对任意初始分布  $\nu$ , 有

$$\lim_{n \rightarrow \infty} \|\nu \mathbf{P}^{\beta_1} \dots \mathbf{P}^{\beta_n} - \Pi^\infty\|_1 = 0.$$

**注:** 退火速度  $\beta_n \leq C \ln n$ , 意味着若要  $\beta_n$  达到指定的  $N_0 \gg 1$  从而使计算满足指定的精度要求, 则必须取  $n \sim \exp(N_0)$ . 这通常是无法忍受的. 实际计算时, 常选  $\beta_n \sim p^{-n}$  ( $p \lesssim 1$  例如 0.999), 往往也可得到不错的计算效果(不得已而求其次).



# 拟 Monte Carlo 方法涉及的基本概念 — 差异

基于 Koksma-Hlawka 定理和拟随机序列的  $I^d = [0, 1]^d$  上  $d$  次连续可微函数  $f$  的拟 Monte Carlo 积分方法的收敛速度可达到  $O(\ln N)^d N^{-1}$ . 首先, 介绍拟 Monte Carlo 方法涉及到的基本概念.

**差异 (discrepancy)**: 对离散点集在  $I^d = [0, 1]^d$  上分布均匀性的一种度量. 对点列  $\{\mathbf{x}_n\}_{n=1}^N \subset I^d$  和任意可测集  $J \subset I^d$ , 定义

$$R_N(J) \triangleq \frac{1}{N} \# \{\mathbf{x}_n \in J\} - m(J),$$

其中  $\# \{\mathbf{x}_n \in J\}$  表示点列  $\{\mathbf{x}_n\}_{n=1}^N$  中落在  $J$  中的点的个数,  $m(J)$  表示  $J$  的  $d$ -维测度. 对  $\mathbf{x} \leq \mathbf{y}$ , 即  $x_i \leq y_i, i = 1, \dots, d$ , 定义

$$J(\mathbf{x}, \mathbf{y}) = \{\mathbf{z} | x_i \leq z_i \leq y_i, i = 1, \dots, d\},$$

显然, 这是一个分别以  $\mathbf{x}, \mathbf{y}$  为最下角和最上角的正矩形区域.



# 拟 Monte Carlo 方法涉及的基本概念 — 差异

记  $I^d$  中正矩形全体的集合为  $E \triangleq \{J(\mathbf{x}, \mathbf{y}) | \mathbf{x}, \mathbf{y} \in I^d, \mathbf{x} \leq \mathbf{y}\}$ . 记  $E^* \triangleq \{J(\mathbf{0}, \mathbf{y}) | \mathbf{y} \in I^d, \mathbf{0} \leq \mathbf{y}\}$ .

定义：一个点列  $\{\mathbf{x}_n\}_{n=1}^N$  的  $L^\infty$  差异定义为

$$D_N \triangleq \sup_{J \in E} |R_N(J)|,$$

记  $\mathbf{0} = (0, 0, \dots, 0)$ ,  $\mathbf{1} = (1, 1, \dots, 1)$ ,  $\{\mathbf{x}_n\}_{n=1}^N$  的  $L^2$  差异定义为

$$T_N \triangleq \left\{ \int_{\mathbf{0} \leq \mathbf{x} \leq \mathbf{y} \leq \mathbf{1}} |R_N(J(\mathbf{x}, \mathbf{y}))|^2 d\mathbf{x} d\mathbf{y} \right\}^{1/2}.$$

特别地, 定义  $D_N^* \triangleq \sup_{J \in E^*} |R_N(J)|$ , 以及

$$T_N^* \triangleq \left\{ \int_{I^d} |R_N(J(\mathbf{0}, \mathbf{x}))|^2 d\mathbf{x} \right\}^{1/2}.$$



# 拟 Monte Carlo 方法涉及的基本概念—变差

对  $[0, 1]$  区间上定义的一维函数  $f(x)$ , 变差定义为

$$V[f] \triangleq \sup_{\tau} \sum_i |f(x_{i+1}) - f(x_i)|,$$

其中  $\tau$  指  $[0, 1]$  区间的所有可能的有限剖分的集合. 如果  $f$  连续可微, 则有  $V[f] = \int_0^1 |df| = \int_0^1 |f'(x)| dx$ .

以下高维函数变差的定义是 Hardy 和 Krause 意义下对一维函数变差定义的推广. 对  $[0, 1]^d$  上定义的  $d$  次连续可微函数  $f(x)$ , 变差 (递归地) 定义为

$$V[f] \triangleq \int_{I^d} \left| \frac{\partial^d f}{\partial x_1 \cdots \partial x_d} \right| dx_1 \cdots dx_d + \sum_{i=1}^d V[f_1^{(i)}],$$

这里  $f_1^{(i)}$  是函数  $f$  在边界  $x_i = 1$  上的限制.



# 拟 Monte Carlo 方法理论基础 — Koksma-Hlawka 定理

**定理：** 对任何序列  $\{\mathbf{x}_n\}_{n=1}^N$  及任何有界变差函数  $f$ , 数值积分误差满足 *Koksma-Hlawka* 不等式：

$$\mathcal{E}[f] \triangleq |I(f) - I_N(f)| = \left| \int_{I^d} f(\mathbf{x}) d\mathbf{x} - \frac{1}{N} \sum_{n=1}^N f(\mathbf{x}_n) \right| \leq V[f] D_N^*,$$

证明从略.

**注 1：** 由 Koksma-Hlawka 不等式知, 拟 Monte Carlo 积分的关键在于生成尽可能小的  $D_N^*$  的拟随机数序列.

**注 2：** 已有若干种方法, 能够产生  $D_N \leq C_d (\ln N)^{k_d} N^{-1}$  的拟随机数序列, 其中  $C_d, k_d$  是与  $d$  有关的常数.



# 能够产生 $D_N \leq C_d(\ln N)^d N^{-1}$ 的拟随机数序列方法的例

- Van der Corput 序列 ( $d = 1$ ):  $x_n$  的产生方法:

- ① 将  $n$  表示为 2 进制数  $n = a_m a_{m-1} \cdots a_1 a_0$ ;
- ② 产生 2 进制数  $x_n = 0.a_0 a_1 \cdots a_{m-1} a_m$ .

- Halton 序列 ( $d > 1$ ):  $\mathbf{x}_n = (x_n^1, \cdots, x_n^d)$  的产生方法:

- ① 将  $n$  表示为  $p_k$  进制数  $n = a_{m_k}^k a_{m_k-1}^k \cdots a_1^k a_0^k$ ,  $1 \leq k \leq d$ , 其中  $p_k$  为第  $k$  个素数;
- ② 产生  $p_k$  进制数  $x_n^k = 0.a_0^k a_1^k \cdots a_{m_k-1}^k a_{m_k}^k$ .

注 1: 可以证明 Halton 序列满足  $D_N \leq C_d(\ln N)^d N^{-1}$ .

注 2 (拟 Monte Carlo 方法的局限性): 一般仅适合于空间维数中等 ( $(\ln N)^d$  不太大)、被积函数光滑 ( $V[f]$  不太大)、盒形区域上的积分.



上机习题七：6.

**Thank You!**

