

学校代码: 10200

研究生学号: 2017200223

分 类 号: O212

密 级: 无



# 东北师范大学 博士学位论文

## 复杂响应曲面的抽样与设计

## Sampling and Design of Complex Response Surfaces

作者: 王素敏

指导教师: 孙法省 教授

一级学科: 统计学

研究方向: 试验设计

学位类型: 理学博士

东北师范大学学位评定委员会

2021年5月



# 东北师范大学

## 博士学位论文

### 学位论文评阅专家及答辩委员会人员信息

论 文 题 目		复杂响应曲面的抽样与设计		
作 者		王素敏		
指 导 教 师		孙法省 教授		
论文评阅人		评阅人 1	匿名评阅	
		评阅人 2	匿名评阅	
		评阅人 3	匿名评阅	
学 位 论 文 答 辩 委 员 会	主 席	朱复康	吉林大学	教授
	委 员	高巍	东北师范大学	教授
		郭建华	东北师范大学	教授
		史宁中	东北师范大学	教授
		徐平峰	长春工业大学	教授



## 摘 要

很多科学试验需要探查未知的复杂系统. 一般情况下, 人们感兴趣的是复杂系统中响应与变量之间的关系, 这种关系被称为响应曲面. 响应曲面的研究目标包括鉴别重要的变量、响应曲面的预测、系统的优化等. 这些研究目标均涉及到响应曲面的抽样和设计. 近年来, 响应曲面的抽样和设计方法一般采用空间填充设计, 其本质是同等地看待变量空间的每一个区域, 让设计点分布的尽可能均匀. 但是在寻找最优响应值中, 空间填充设计难免会抓不住响应曲面的趋势特征, 因而造成设计点的浪费. 本文提出探查复杂响应曲面的一种新的抽样方法, 该方法不仅可用于系统优化, 而且还可用于贝叶斯计算和高维数值积分. 此外, 本文还讨论了一类包含复合型变量的响应曲面设计的大样本性质. 具体研究工作如下:

**1. 基于Kullback-Leibler偏差和函数的核估计方法, 提出了一类新的确定性抽样方法称为Kullback-Leibler点.** 但是当知道一些先验信息后, 例如已知响应曲面包含多个局部最大值, 并且研究目的是寻找响应曲面的最大值时, 由于空间填充设计分布的比较均匀, 所以难免会有设计点会落入低响应值的区域, 进而造成设计点的浪费. 因此, 本文将复杂响应曲面转化为非标准化的密度函数, 进而通过最小化一组点的核密度估计与该密度函数的Kullback-Leibler 偏差, 提出一类新的获取复杂响应曲面代表点的抽样方法, 得到的点集称为Kullback-Leibler(KL)点. KL点能够抓住响应曲面的趋势特征并且不同点之间保持一定的距离. 本文通过两步近似来生成 $n$ 个KL点. 首先, 为了避免高维积分, 通过最小化Kullback-Leibler偏差的Monte Carlo近似来生成KL点. 其次, 由于同时生成 $n$ 个KL点比较困难, 为此采用一次生成一个点的策略. 数值模拟表明KL点与最小化能量设计(Joseph et al., Technometrics, 2019, [1])和支撑点(Mak and Joseph, Ann Stat, 2018, [2])相比, 在空间填充性质与数值积分上更具优势(见图2.2-2.4, 表2.1-2.2).

**2. 针对响应曲面没有显式表达或者计算耗时的情形, 基于Kullback-Leibler偏差和函数的核估计方法, 提出了自适应的Kullback-Leibler(AKL)点.** 若响应曲面具有复杂的结构, 例如没有显式表达或者计算耗时. 本文首先基于初始的空间填充设计拟合一个平稳的高斯过程模型来代替该密度函数, 进而通过最小化一组点

的核密度估计与该高斯过程模型的Kullback-Leibler偏差来生成一类推广的KL点. 本文采用自适应的策略, 每生成一个KL点, 均更新一次该高斯过程模型, 进而产生下一个KL点, 得到的KL点称为自适应的KL(AKL)点. AKL点可用于复杂密度函数的抽样, 例如复杂的贝叶斯后验密度函数, 以及探索和优化复杂的黑盒子函数(见图3.1- 图3.3).

3. 针对响应曲面维数较高的情形, 基于Kullback-Leibler偏差和函数的最小邻近距离估计方法, 提出了Kullback-Leibler-nn(KL-nn)点. 当复杂响应曲面的维数较高时, 依然可将其转化为非标准化的密度函数. 本文通过最小化一组点密度的最小邻近距离估计与该密度函数的Kullback-Leibler偏差提出第二类推广的KL点, 称为Kullback-Leibler-nn(KL-nn)点. 本文通过基于Markov Chain Monte Carlo(MCMC)的局部搜索算法来生成KL-nn点. 数值模拟表明, KL-nn点有较好的空间填充性质(见图3.6), 且在高维情形下的数值积分上比支撑点和最小化能量设计有较小的积分误差(见图3.7).

4. 导出了当复杂响应曲面同时包含定性和定量因子时的边际耦合设计的大样本性质. 当复杂响应曲面同时包含定性和定量因子时, 传统的抽样设计方法不可用. 为处理该问题, Qian (JASA, 2012, [3])提出了一种新的设计方法, 称为切片拉丁超立方体设计. 该设计中定性因子的每个水平组合, 对应的定量因子的设计点都分布的很均匀, 这样可以减少因子间的混杂. He and Qian (Stat Sin, 2016, [4])研究了切片拉丁超立方体设计的大样本性质, 从理论上说明了空间填充性的合理性. 当定性因子的个数和水平数增加时, 切片拉丁超立方体设计的试验次数呈指数增长, 因此实际的试验成本难以承受. 为此, Deng et al. (Stat Sin, 2015, [5])提出了边际耦合设计方法, 该设计可以大大降低试验次数. 由于边际耦合设计的结构与切片拉丁超立方体设计的结构不同, 所以He and Qian (Stat Sin, 2016, [4])获得的大样本性质不能直接应用到边际耦合设计中. 鉴于此, 本文研究了边际耦合设计的大样本性质, 并进一步导出了边际耦合设计的中心极限定理.

**关键词:** 计算机试验; 空间填充设计; 确定性抽样; Kullback-Leibler偏差; Kriging 模型; 贝叶斯计算; 边际耦合设计.

## Abstract

Many scientific experiments involve exploration of unknown complex systems. Generally, people are interested in the relationship between response and variables, which is called response surface. The objectives of exploring response surface include identifying the important variables, prediction of the response surface, optimizing the complex system and so on. These objectives need to sample and design of response surface. In recent years, the commonly used sampling and designs of response surfaces are space filling designs, which spread the design points evenly in the region. However, in some applications, such as finding the optimum, the space filling designs will fail to capture the characteristics, which are not useful. This paper presents a new sampling method for exploration complex response surface. This sampling method can be used not only in system optimization, but also in Bayesian computation and high dimensional numerical integration. In addition, some large sample properties of a class of response surface designs with compound variables are discussed in this paper. The main work is shown as follows:

**1. We propose a new kind of deterministic sampling based on the Kullback-Leibler divergence and the kernel function estimator, which is called Kullback-Leibler points.** When we obtain some prior information, for example the complex surfaces contain many local optima or can be expressed explicitly, and the purpose is to find the maximum response surface, due to the traditional space filling designs are well spread in the space, so design points may placed in the zero-yield regions which are not useful. Therefore, in this paper, we regard the complex response surface as the unnormalized density function, and then, by minimizing the Kullback-Leibler divergence between this density and the kernel density estimator of a points sets, we propose a new sampling method to obtain the representative points of complex response surface, which is called Kullback Leibler (KL) points. KL points can capture the characteristics of the surfaces and make the points are separated as far as possible. In this paper, we generate  $n$  KL points by two-step approximation. Firstly, in order

to avoid the high dimensional integral, we generate KL points by minimizing Monte Carlo approximation of the Kullback-Leibler divergence. Secondly, as a result of generating  $n$  KL points at the same time is very difficult, so we adopt a one point at a time greedy algorithm for the generation of  $n$  KL points. Figure 2.2-2.4 and Table 2.1-2.2 show that KL points have apparent advantage in space filling property and numerical integration compared with support point (Mak and Joseph, Ann Stat, 2018,[2]) and minimum energy design (Joseph et al., Technometrics, 2019,[1]).

**2. When the surfaces can't be expressed explicitly or time-consuming, we propose an adaptive Kullback-Leibler(AKL) points set based on the Kullback-Leibler divergence and the kernel function estimator.** When the surfaces have complex structure, for example it can't be expressed explicitly or time-consuming. Then, we fit a stationary Gaussian process model based a initial space filling design to replace this density, and then generate a class of generalized KL points by minimizing the Kullback Leibler divergence between the Gaussian process model and kernel density estimator. We adopt an adaptive strategy to generate KL points, at every KL point is generated, the Gaussian process model is updated sequentially to generate the next KL point. The resulting points are called adaptive KL (AKL) points. AKL points can be used in sampling of complex density functions, such as Bayesian posterior, and optimizing complex black-box functions (see Figure3.1-3.3).

**3. When the dimension of response surface is large, we propose Kullback-Leibler-nn points based on the Kullback-Leibler divergence and the nearest neighbor estimator of function.** When the dimension of complex response surface is large, we can still regard it as as the unnormalized density function. In this paper, the KL points can be obtained by minimizing the Kullback-Leibler divergence between the density function and the nearest neighbor estimator of density among a points set, which are called Kullback Leibler-nn (KL-nn) points. We use a form of local search algorithm based on MCMC to generate KL-nn points. Numerical simulations show that KL-nn points have better space filling property (see Figure 3.6), and have smaller integration error than support point and minimum energy design in the case of high

dimensional numerical integration (see Figure 3.7).

**4. We derive some sampling properties of marginally coupled designs which are used to explore the complex surfaces involving both qualitative and quantitative variables.** The traditional design method can't be used for complex surfaces involving both qualitative and quantitative variables. For constructing designs for these complex surfaces, Qian (JASA, 2012,[3]) suggested using sliced Latin hypercube designs. Such a design for each level of any qualitative factor, the corresponding design points of quantitative are space filling designs. He and Qian (Stat Sin, 2016, [4]) derived some sampling properties of sliced Latin hypercube designs to show that the space-filling properties. However, the run sizes of these designs can be very large, even for a moderate number of qualitative factors. To solve this issue, Deng et al. (Stat Sin, 2015, [5] ) proposed a new type of designs called marginally coupled designs. These designs can greatly reduce the run sizes, and thus save the cost. Due to the structure of marginal coupled designs are different from sliced Latin hypercube designs, hence the sampling properties obtained by He and Qian (Stat Sin, 2016, [4]) can't be directly applied marginal coupled designs. Motivated by these, we derive some sampling properties of marginally coupled designs and develop the central limit theorems.

**Key Words:** Computer experiments; Space filling design; Deterministic sampling; Kullback-Leibler divergence; Kriging model; Bayesian computation; Marginally coupled designs.



## 目 录

中文摘要	I
英文摘要	III
第一章 绪论和基本知识	1
1.1 研究背景及意义	1
1.2 探索复杂响应曲面的研究现状	4
1.2.1 空间填充设计	4
1.2.2 确定性抽样方法	12
1.2.3 同时包含定性和定量因子的设计	19
1.3 本文的主要工作和结构安排	24
第二章 基于Kullback-Leibler散度的复杂响应曲面的抽样方法	27
2.1 背景介绍	27
2.2 基于核密度估计的Kullback-Leibler 点	30
2.3 序贯生成KL点的算法	33
2.4 数值模拟	36
2.4.1 空间填充性质	36
2.4.2 数值积分	39
2.5 本章小结	41
2.6 理论证明	41
第三章 基于Kullback-Leibler散度抽样方法的推广研究	51
3.1 背景介绍	51
3.2 自适应的KL(AKL)点	52
3.2.1 复杂密度函数的AKL点	53

3.2.2 探索和优化昂贵的黑盒子函数 . . . . .	56
3.3 基于最小邻近距离的Kullback-Leibler-nn点 . . . . .	60
3.3.1 Kullback-Leibler-nn点的定义 . . . . .	60
3.3.2 序贯地生成Kullback-Leibler-nn 点的算法 . . . . .	61
3.3.3 数值模拟 . . . . .	61
3.4 本章小结 . . . . .	65
<b>第四章 同时包含定量和定性因子的复杂响应曲面的试验设计</b>	<b>67</b>
4.1 背景介绍 . . . . .	67
4.2 边际耦合设计的构造 . . . . .	68
4.3 边际耦合设计的大样本性质 . . . . .	69
4.4 数值模拟 . . . . .	71
4.5 本章小结 . . . . .	74
4.6 理论证明 . . . . .	75
<b>第五章 总结与展望</b>	<b>81</b>
5.1 本文的主要工作 . . . . .	81
5.2 未来的研究工作 . . . . .	82
<b>参考文献</b> . . . . .	<b>85</b>
<b>在学期间公开发表论文及著作情况</b> . . . . .	<b>90</b>
<b>致谢</b> . . . . .	<b>91</b>
<b>附录</b> . . . . .	<b>93</b>

# 第一章 绪论和基本知识

## §1.1 研究背景及意义

在现代科学试验中,许多自然和社会现象通常可以用一个复杂的响应曲面来描述.有些响应曲面形式已知但没有显式表达且结构复杂,为此可以运行一个程序来实现对复杂响应曲面的探索.探索复杂响应曲面的目标有:1)预测响应曲面;2)寻找响应曲面的最优值或目标值.这些目标的实现依赖两个基本的统计问题:1)设计与抽样问题,也是后续建模和数据分析的基础;2)数据分析问题,即响应曲面的建模与优化.复杂响应曲面的抽样和设计决定着统计建模与参数估计的精度,因此是统计学的一个重要研究方向.

一般的,复杂的系统或者科学现象可以用数学模型表示为

$$y = f(x_1, \cdots, x_d),$$

其中 $\mathbf{x} = (x_1, \cdots, x_d)$ 表示输入,  $y$ 表示输出或者响应.当响应 $f(\mathbf{x})$ 复杂且没有解析表达式时,常常需要建立一个简单的易于计算的且有解析表达式的模型 $g(\mathbf{x})$ 替代 $f(\mathbf{x})$ ,即

$$y \approx g(x_1, \cdots, x_d).$$

近似模型的流程图如图1.1所示

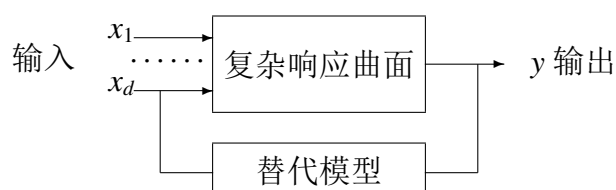


图 1.1 近似模型的流程图

常用的近似模型是高斯随机过程模型(Gaussian Process Models, [6]), 又称为Kriging模型. 高斯随机过程模型首先由法国的地质学家Kriging Danie G.提出并

应用到地质学的研究中. 随后, Sacks et al. (1989)又将高斯随机过程模型应用在计算机试验中, 现已成为计算机试验中最常用的建模方法<sup>[7-11]</sup>. Kriging 模型把响应 $Y(\mathbf{x})$ 处理为下面随机过程的实现:

$$Y(\mathbf{x}) = \sum_{j=1}^p f_j(\mathbf{x})\beta_j + Z(\mathbf{x}) = \mathbf{f}'(\mathbf{x})\boldsymbol{\beta} + Z(\mathbf{x}), \quad (1.1)$$

这里 $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_p(\mathbf{x}))'$ 表示已知的回归函数或者基函数,  $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)'$ 为未知的回归系数向量,  $Z(\mathbf{x})$ 是一个均值为0的平稳高斯过程,  $\mathbf{x}_1$ 与 $\mathbf{x}_2$ 之间的协方差为

$$COV(Z(\mathbf{x}_1), Z(\mathbf{x}_2)) = \sigma_Z^2 R(\mathbf{x}_1, \mathbf{x}_2),$$

其中 $\sigma$ 为随机过程的标准差,  $R(\mathbf{x}_1, \mathbf{x}_2)$ 为两个随机变量 $Z(\mathbf{x}_1), Z(\mathbf{x}_2)$ 的相关函数. 未试验点的预测精度依赖于相关函数 $R(\mathbf{x}_1, \mathbf{x}_2)$ 的选择. 常用的相关函数是高斯函数, 即

$$R(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(-\sum_{i=1}^d \theta_i |x_{1i} - x_{2i}|^2\right),$$

其中 $\mathbf{x}_1 = (x_{11}, \dots, x_{1d}), \mathbf{x}_2 = (x_{21}, \dots, x_{2d}), \boldsymbol{\theta}_1 = (\theta_1, \dots, \theta_d)$ .

对于上述高斯随机过程模型, 如果选择一组适当的输入也称训练集 $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ , 其对应的响应值为 $\mathbf{Y}^n = (Y(\mathbf{x}_1), \dots, Y(\mathbf{x}_n))'$ , 那么未试验点 $\mathbf{x}_0$ 处的预测为

$$\hat{Y}(\mathbf{x}_0) = \mathbf{f}'(\mathbf{x}_0)\hat{\boldsymbol{\beta}} + \mathbf{r}_0' R^{-1}(\mathbf{Y}^n - \mathbf{F}\hat{\boldsymbol{\beta}}),$$

这里 $\mathbf{F} = (f_j(\mathbf{x}_i))$ 为 $n \times p$ 的关于训练集的回归函数矩阵,  $\mathbf{r}_0 = (R(\mathbf{x}_0 - \mathbf{x}_1), \dots, R(\mathbf{x}_0 - \mathbf{x}_n))'$ , 且 $\mathbf{R} = (R(\mathbf{x}_i - \mathbf{x}_j))_{n \times n}$ 是关于 $\mathbf{Y}^n$ 的 $n \times n$ 的相关函数矩阵.  $\hat{\boldsymbol{\beta}} = (\mathbf{F}'\mathbf{R}^{-1}\mathbf{F})^{-1}\mathbf{F}'\mathbf{R}^{-1}\mathbf{Y}^n$ 是 $\boldsymbol{\beta}$ 的广义最小二乘估计.

特别的, 一个普通的Kriging 模型(ordinary kriging model) 表示为

$$Y(\mathbf{x}) = \boldsymbol{\mu} + Z(\mathbf{x}),$$

这里 $\boldsymbol{\mu}$ 表示均值向量. 在此普通的Kriging 模型下, 给定一组输入 $\mathbf{x}_1, \dots, \mathbf{x}_n$ , 则未试

验点 $\mathbf{x}_0$ 处的预测为

$$\hat{Y}(\mathbf{x}_0) = \hat{\mu} + r_0' R^{-1} (\mathbf{Y}^n - \hat{\mu} \mathbf{1}_n),$$

这里 $r_0'$ 是一个长度为 $n$ 的向量, 其第 $i$ 个元素为 $R(\mathbf{x}_0 - \mathbf{x}_i)$ ,  $R$ 是 $n \times n$ 的相关矩阵, 其第 $(i, j)$ 个元素为 $R(\mathbf{x}_i, \mathbf{x}_j)$ ,  $\mathbf{1}_n$ 是分量全为1的 $n$ 维列向量, 且 $\hat{\mu} = (\mathbf{1}_n' R^{-1} \mathbf{1}_n)^{-1} \mathbf{1}_n' R^{-1} \mathbf{Y}^n$ .

在探索复杂响应曲面时, 存在若干重要的问题:

1. 当对复杂响应曲面没有任何先验信息时, 通常使用高斯随机过程模型来近似复杂响应曲面, 进而预测未试验点的响应. 在这种情形下, 由于对复杂的响应曲面没有任何先验的信息, 故设计点的选取通常是在整个试验空间上选取, 常用的设计是空间填充设计<sup>[12, 13]</sup>, 即在试验空间上均匀地选取设计点. 但是当知道一些先验信息后, 仍然选择一个空间填充设计显然不合适. 因为难免会有许多设计点落入我们不感兴趣的区域内. 例如图1.2-1.3是两个复杂响应曲面的等高线图, 从图中可以看出这两个复杂响应曲面包含多个局部最大值, 因此如果依然使用空间填充设计来探索复杂的响应曲面, 则不可避免的会使许多设计点落入低响应值的区域, 为此会造成设计点的浪费. 如何获取响应曲面的代表点使得这些点均落入感兴趣的区域且不同的点之间保持一定的距离, 是探究复杂响应曲面所面临的重要问题. 这对响应曲面的建模和寻找响应曲面的最优值意义非凡.
2. 大多数响应曲面研究中都假设变量是定量的, 这样的假设下许多设计和建模方法均可借鉴. 但是在实际应用中, 响应曲面同时包含定性和定量因子也很常见, 如Schmidt et al. (2005) 研究了一个数据中心的例子, 该例中既包含定性因子如扩散器的位置和热风回流排气口的位置, 又包含定量因子如电源的功率和扩散器的流量. 但是该文献并没有对设计进行研究. 如何设计试验探索此类复杂响应曲面的最优值或者预测未试验点的响应值也是近年来热门的研究课题. 为此, Deng et al. (2015)提出了一类经济高效的设计称为边际耦合设计. 然而文献中没有对其理论性质进行研究和讨论. 因此, 导出边际耦合设计的大样本性质也是值得研究的重要问题.

下节详细阐述空间填充设计, 代表点方法以及探索复杂响应曲面同时包含定性和定量因子时已有的研究成果.

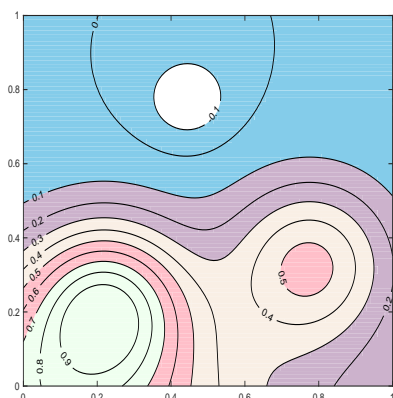


图 1.2

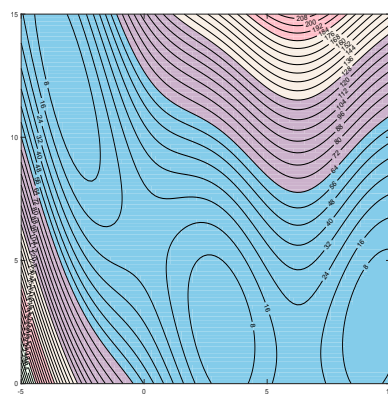


图 1.3

## §1.2 探索复杂响应曲面的研究现状

本节首先介绍当复杂响应曲面没有任何先验信息时的空间填充设计, 然后介绍当获得复杂响应曲面的若干先验信息后好的确定性抽样方法, 最后介绍当复杂响应曲面既包含定性因子又包含定量因子时常用的试验设计.

### §1.2.1 空间填充设计

在探索复杂响应曲面的过程中, 当对复杂响应曲面没有任何先验信息时, 如响应曲面没有显示表达或者计算机运行一次比较耗时时, 通常希望设计点或者抽样点充满整个试验空间. 一个好的设计应该满足以下几个原则:

- (1) 没有重复的设计点;
- (2) 允许拟合多种类型的模型;
- (3) 能够提供试验区域各部分的信息;
- (4) 具有良好的投影性质.

基于上述几个设计原则, 常用的空间填充设计有拉丁超立方体设计<sup>[12]</sup>和均匀设计<sup>[13]</sup>以及这两类设计的相关推广. 下面介绍几类常用的拉丁超立方体设计和均匀设计.

### §1.2.1.1 拉丁超立方体设计

在空间 $[0, 1]^d$ 上的拉丁超立方体设计的定义如下:

**定义1.2.1.** 具有 $d$ 个因子 $n$ 次试验的设计 $X$ 是一个 $n \times d$ 的矩阵, 如果该矩阵的每列均满足在任一区间 $[0, 1/n), [1/n, 2/n), \dots, [(n-1)/n, 1]$ 中有且仅有一个点, 则称 $X$ 为拉丁超立方体设计, 并表示为 $LHD(n, d)$ .

Mckay et al. (1979)首先给出了拉丁超立方体设计的构造方法:

$$x_{ik} = \frac{l_{ik} - u_{ik}}{n}, i = 1, \dots, n, k = 1, \dots, d. \quad (1.2)$$

其中矩阵 $L = (l_{ik})_{n \times d}$ 的每列均为 $\{1, \dots, n\}$ 上的一个均匀排列且列与列之间是相互独立的.  $u_{ik}$ 是相互独立的 $U(0, 1]$ 上的随机变量且与 $l_{ik}$ 相互独立, 这样的矩阵 $L$ 称为拉丁超立方体. 这类设计很受欢迎, 因为当 $X$ 投影到任何一维空间上时, 若将区间 $[0, 1]$ 分成 $n$ 个格子 $[0, 1/n), \dots, [(n-1)/n, 1]$ , 则每个格子中恰好均有一个点, 从而可以实现一维投影均匀性. 例如, 图1.4给出了 $5 \times 2$ 的拉丁超立方体设计, 显然该设计投影到每一维上, 均能达到一维投影均匀性. 因此基于拉丁超立方体设计的抽样要比随机的抽样即Monte Carlo抽样的均匀性要好. 进一步, Mckay et al. (1979)表明基于拉丁超立方体设计抽样的参数估计的方差要比随机抽样下参数估计的方差小. 由于随机生成的拉丁超立方体设计的高维投影均匀性较差以及列于列之间可能有较强的相关性等. 因此近年来, 许多学者提出了优化的拉丁超立方体的优化设计, 其中正交性和最大最小距离是两个常用的设计准则<sup>[17-19]</sup>. 为了提高设计的投影均匀性, Tang (1993)提出了基于正交表的拉丁超立方体设计, 如果正交表的强度为 $t$ , 该设计能达到最高 $t$ 维投影均匀性. 下面介绍基于正交表的拉丁超立方体设计.

### §1.2.1.2 基于正交表的拉丁超立方体设计

首先介绍正交表的概念<sup>[21]</sup>.

**定义1.2.2.** 一个有 $n$ 行,  $d$ 列, 强度为 $t$ 的正交表 $A$ 是一个 $n \times d$ 的矩阵, 其中第 $j$ 列有 $s_j$ 个不同的水平 $\{1, 2, \dots, s_j\}$ 且对于 $A$ 的任意 $n \times t$ 的子矩阵, 所有水平组合均出

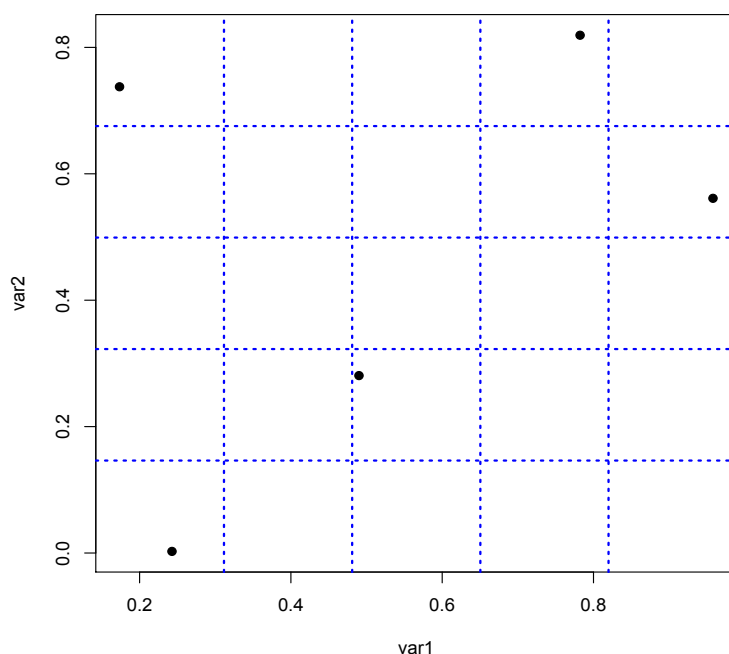


图 1.4 一个  $5 \times 2$  的拉丁超立方体设计.

现且出现的次数相同. 若不是所有的  $s_j$  都相等, 则称此表为非对称的正交表, 表示为  $OA(n, s_1^{d_1} \cdots s_k^{d_k}, t)$ , 其中前  $d_1$  列有  $s_1$  个水平, 接着  $d_2$  列有  $s_2$  个水平, 以此类推,  $d = d_1 + \cdots + d_k$ . 若所有的  $s_j$  个水平都等于  $s$ , 则称此表为对称的正交表, 表示为  $OA(n, s^d, t)$ , 其中  $d$  表示因子的个数.

由于正交表具有好的投影性质, 因此Tang (1993)提出了基于正交表的拉丁超立方体设计, 其构造方法见算法1.1.

**例1.2.1.** 表1.1 给出了基于正交表  $OA(25, 5^6, 2)$  的拉丁超立方体, 从该表中可以看出基于正交表的拉丁超立方体设计的每一列均是  $\{1, \dots, 25\}$  的一个均匀排列. 通过公式(1.3)变换到  $[0, 1]^d$  上后, 该设计投影到每一维上, 每个小区间  $[\frac{i-1}{25}, \frac{i}{25})$ ,  $i = 1, \dots, 25$  均有一个设计点, 说明该设计具有一维投影均匀性. 图1.5 给出了这个基于正交表的拉丁超立方体设计的所有二维投影. 从图中可以看出, 设计点投影到所有二维子空间上的点数相同, 这说明该设计也具有二维投影均匀性.



---

**算法 1.1** 基于正交表的拉丁超立方体设计的构造<sup>[20]</sup>

---

step 1. 基于具有强度为 $t$ ,  $n$ 行、 $d$ 列、 $s$ 水平的正交表 $A = OA(n, s^d, t)$ , 对于 $A$ 的每一列, 水平为 $i$ 的 $r = n/s$ 个位置用 $\{(i-1)r+1, \dots, ir\}$ 上的一个均匀排列来替换,  $i = 1, \dots, s$ , 得到的矩阵记为 $L$ .

step 2. 一个具有强度为 $t$ ,  $n \times d$ 的基于正交表的拉丁超立方体设计 $X$ , 可由 $L$ 通过如下变换得到:

$$x_{ik} = \frac{l_{ik} - u_{ik}}{n}, i = 1, \dots, n, k = 1, \dots, d, \quad (1.3)$$

这里 $u_{ik}$ 是相互独立的 $U(0, 1]$ 上的随机变量.

---

表 1.1 基于正交表 $OA(25, 5^6, 2)$ 的拉丁超立方体

1	1	3	5	2	5
3	7	9	13	17	24
5	11	13	17	23	10
4	19	16	21	6	14
2	25	21	7	15	18
10	5	7	9	9	6
9	6	14	23	4	16
7	15	22	2	20	12
6	17	4	20	14	25
8	23	19	12	25	2
13	3	12	15	12	15
11	9	23	18	10	1
12	14	18	6	3	23
14	16	2	25	16	8
15	22	20	19	18	20
17	4	5	10	24	13
18	8	5	10	24	13
19	12	8	22	13	4
20	18	25	11	5	7
16	24	11	3	7	21
23	2	24	24	21	22
21	10	17	1	11	9
24	13	1	14	8	17
22	20	15	8	19	3
25	21	6	16	1	11

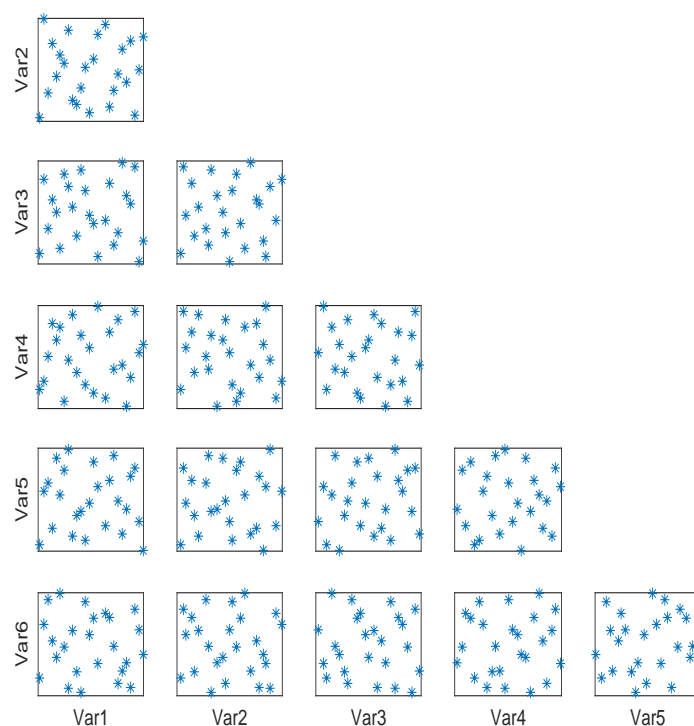


图 1.5 基于正交表的拉丁超立方体设计.

由于基于正交表的拉丁超立方体设计具有好的投影性质, 因此近年来许多学者在此给出具有更好分层性质的拉丁超立方体设计<sup>[22-26]</sup>. 然而, 由于这类基于正交表的拉丁超立方体设计依赖于正交表的构造, 故此类设计对试验次数, 因子个数以及因子的水平数均有限制.

### §1.2.1.3 正交拉丁超立方体设计(OLHD)

随机产生的拉丁超立方体设计, 难免会有高度相关的列, 这样不利于统计建模与参数估计. 另外, 设计的正交性会提高设计点的低维投影均匀性<sup>[16]</sup>. 设计的正交性是指列与列之间的相关系数为0. Ye (1998)给出构造 $n = 2^m + 1$ 行,  $d = 2m - 2$ 列的正交拉丁超立方体设计的方法, 这里 $m > 1$ 是非负整数. 令向量 $\mathbf{e} = (1, 2, \dots, 2^{m-1})^T$ . 令 $(r \ s)\mathbf{e}$ 表示对向量 $\mathbf{e}$ 进行一次排列, 即将向量 $\mathbf{e}$ 中的第 $r$ 与第 $s$ 个元素交换位置,  $(r_1 \ s_1) \cdots (r_k \ s_k)\mathbf{e}$ 表示先后对向量 $\mathbf{e}$ 排列 $k$ 次. 算法1.2给出了具体的构造步骤.

---

#### 算法 1.2 正交拉丁超立方体设计的构造<sup>[15]</sup>

---

step 1. 定义 $m - 1$ 个排列

$$A_k = \prod_{j=1}^{2^{m-k-1}} \left\{ \prod_{i=1}^{2^{k-1}} ((j-1)2^k + ij2^k + 1 - i) \right\}, k = 1, \dots, m-1.$$

step 2. 构造 $2^{m-1} \times (2m - 2)$ 的矩阵 $M$

$$M = [\mathbf{e} \ A_1 \mathbf{e} \ \cdots \ A_{m-1} \mathbf{e} \ A_{m-1} A_1 \mathbf{e} \ \cdots \ A_{m-1} A_{m-2} \mathbf{e}].$$

step 3. 构造 $2^{m-1} \times (2m - 2)$ 的矩阵 $S$

$$S = [\mathbf{1} \ \mathbf{a}_1 \ \cdots \ \mathbf{a}_{m-1} \ \mathbf{a}_1 \odot \mathbf{a}_2 \ \cdots \ \mathbf{a}_1 \odot \mathbf{a}_{m-1}].$$

这里

$$\mathbf{a}_k = \mathbf{B}_1 \otimes \mathbf{B}_2 \otimes \cdots \otimes \mathbf{B}_{m-1}, k = 1, \dots, m-1,$$

$$\mathbf{B}_{m-k} = (-1 \ 1)^T, \mathbf{B}_i = (1 \ 1)^T.$$

符号 $\odot$ 表示向量中对应元素的乘积,  $\otimes$ 表示向量之间的Kronecker乘积.

step 4. 构造 $2^{m-1} \times (2m - 2)$ 的矩阵 $T$

$$T = M \odot S.$$

step 5. 一个 $(2^m + 1) \times (2m - 2)$ 的正交拉丁超立方体设计 $O$ 可表示为

$$O = (T^T \ 0_{2^{m-2}} \ -T^T)^T.$$


---

正交拉丁超立方体设计不仅具有好的空间填充性质,而且其列与列还满足正交性,因此可以用来拟合包含交互项的二阶线性模型. 在正交拉丁超立方体设计的基础上Sun et al. (2009)和Sun et al. (2009)给出了具有灵活试验次数的正交拉丁超立方体设计的构造方法.

#### §1.2.1.4 均匀设计

Fang and Wang (1994)利用数论方法提出的均匀设计也是一类广泛用于模型未知的空间填充设计. 不失一般性, 试验区域仍选择为 $[0, 1]^d$ , 均匀设计的思想是使试验点尽可能均匀地分散在试验区域内<sup>[28-30]</sup>. 令 $D = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ 表示试验区域内的 $n$ 个点, 其中 $\mathbf{x}_i = \{x_{i1}, \dots, x_{id}\}, i = 1, \dots, n$ . 如何度量这 $n$ 个点在试验区域中散布的均匀程度是构造均匀性的关键.

考虑非参数回归模型

$$y = g(\mathbf{x}) + \varepsilon,$$

这里 $\varepsilon$ 为随机误差, 且通常假定 $E(\varepsilon) = 0, \text{Var}(\varepsilon) = \sigma^2$ . 假定试验人员希望估计 $y$ 在试验区域上的均值, 即

$$\mu = E(y) = \int g(\mathbf{x})d\mathbf{x}.$$

一个直观的想法是用试验点 $D = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ 上的平均响应值

$$\hat{\mu} = \bar{y} = \sum_{i=1}^n y_i$$

来估计, 这里 $y_i = g(\mathbf{x}_i)$ . 为此, 需要选择一个好的设计使得估计的精度较高. 从伪蒙特卡罗理论中的Koksma-Hlawka不等式可知

$$|E(y) - \bar{y}| \leq V(g)D^*(D),$$

这里 $V(g)$ 表示函数的全变差,  $D^*(D)$ 表示设计 $D$ 的星偏差, 该偏差不依赖函数 $g$ . 因此星偏差 $D^*(D)$ 可以用来度量 $n$ 个点的均匀性程度, 且该偏差越小说明设计 $D$ 在试验区域内分布的越均匀.

Hickernell (1998)提出的中心化 $L_2$ -偏差是一种常用的星偏差, 其定义如下:

**定义1.2.3.** 令 $F(\mathbf{x})$ 表示 $[0, 1]^d$ 上的均匀分布函数,  $F_n(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n I(\mathbf{x}_i \leq \mathbf{x})$ 表示设计 $D = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ 的经验分布函数, 则设计 $D$ 的中心化 $L_2$ -偏差为

$$\begin{aligned} CD(D) &= \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \prod_{k=1}^d \left( 1 + \frac{1}{2} \left| x_{ik} - \frac{1}{2} \right| + \frac{1}{2} \left| x_{jk} - \frac{1}{2} \right| - \frac{1}{2} |x_{ik} - x_{jk}| \right) \\ &\quad - \frac{2}{n} \sum_{i=1}^n \prod_{k=1}^d \left( 1 + \frac{1}{2} \left| x_{ik} - \frac{1}{2} \right| + \frac{1}{2} \left| x_{ik} - \frac{1}{2} \right|^2 \right) + \left( \frac{13}{12} \right)^d. \end{aligned}$$

最小化中心化 $L_2$ -偏差得到的设计称为均匀设计, 记为 $U_n(n^d)$ . 但是直接最小化中心化 $L_2$ -偏差非常困难. 为此, Korobov (1959)提出利用好格子点法来得到近似的均匀设计. 图1.6给出了一个 $5 \times 2$ 的均匀设计 $U_5(5^2)$ , 显然设计点均匀地分布在试验空间内. 由于均匀设计在整个试验空间上的均匀性好, 但是在低维空间尤其是二维空间上的均匀性比较差, 为此, Sun et al. (2019)又提出了均匀投影设计, 该设计不仅在整个试验空间的均匀性比较好, 而且在所有二维投影空间上的均匀性也比较好.

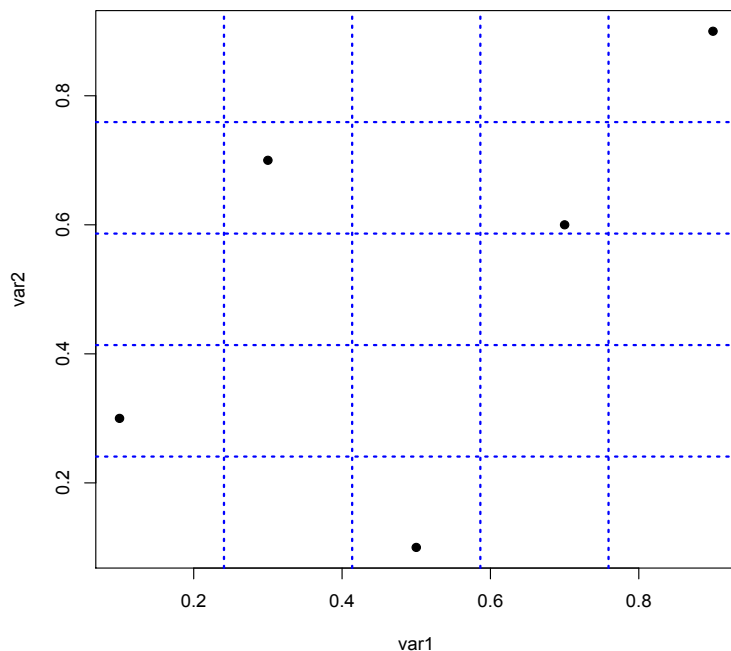


图 1.6 一个 $5 \times 2$ 的均匀设计 $U_5(5^2)$ .

### §1.2.1.5 极大极小和极小极大距离设计

另外一类常用的空间填充设计是Johnson et al. (1990)提出的基于欧氏距离的极大极小和极小极大距离设计. 设计的极大极小距离又称为分离距离(separation distance), 对于任意一个设计  $D = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subseteq [0, 1]^d$ , 设计的分离距离定义为:

$$\rho_s(D) = \min_{i \leq j} d(\mathbf{x}_i, \mathbf{x}_j),$$

这里  $d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_2$  表示两个设计点  $\mathbf{x}_i$  与  $\mathbf{x}_j$  之间的欧氏距离. 试验空间中的一个设计  $D$  称为极大极小距离设计, 如果设计  $D$  的分离距离在所有设计中达到最大. 设计的极小极大距离又称为填充距离(fill distance), 对任意一个设计  $D = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subseteq [0, 1]^d$ , 设计的填充距离定义为:

$$\rho_f(D) = \sup_{\mathbf{x} \in (0,1)^d} \min_{\mathbf{x}_i \in D} d(\mathbf{x}, \mathbf{x}_i),$$

试验空间中的一个设计  $D$  称为极小极大距离设计, 如果设计  $D$  的填充距离在所有设计中达到最小.

结合拉丁超立体设计, Morris and Mitchell (1995) 利用模拟退火算法给出了极大极小拉丁超立体设计的构造. van Dam et al. (2009)给出了极大极小拉丁超立体设计的界. Joseph and Hung (2012)结合正交准则, 给出了正交的极大极小拉丁超立体设计的构造方法.

### §1.2.2 确定性抽样方法

如果试验前已经获得复杂响应曲面的一些信息, 比如已知该曲面的显示表达式  $f(\mathbf{x})$  时, 或者该复杂响应曲面包含多个局部最优值时, 这时再利用空间填充设计探索复杂响应曲面会造成设计点的浪费. 对于此类复杂响应曲面, 设计点应该抓住响应曲面的趋势特征, 并且点与点尽量地隔开. 这样的设计点集称为复杂响应曲面的代表点. 一组好的代表点应该满足:

- (1) 能够抓住响应曲面的趋势特征;
- (2) 没有重复的设计点并且点与点之间的距离尽量地隔开;

(3) 尽量避免大量地计算复杂的响应值.

从复杂响应曲面获取代表点集的方法可以转化为统计学中常见的获取概率密度函数的代表点集, 即把该复杂响应曲面看作是一个密度函数, 然后从该密度函数中获取代表点集. 常用的从复杂的概率密度函数中获取代表点集的方法是Markov Chain Monte Carlo (MCMC). 然而, MCMC产生的样本点会出现点与点的距离很近或者重复点的现象, 距离很近的点或者重复的点, 从计算机试验设计的角度看, 因为这些点不提供额外的信息, 因此会造成点的浪费. 特别地, MCMC抽样方法需要多次迭代且大量的计算密度函数的值, 因此计算成本非常高. 为了克服MCMC抽样方法的缺点, Joseph et al. (2015)、Joseph et al. (2019)和Mak and Joseph (2018)、Mak and Joseph (2017)提出使用“确定性的”抽样方法来代替“随机化”的抽样方法, 得到的确定性抽样点能够抓住密度函数的趋势特征, 即样本点大都集中在密度函数值高的区域且点与点之间尽量地隔开. 接下来详细介绍这几类好的确定性抽样方法.

### §1.2.2.1 最小化能量设计

最小化能量设计(MED)首先由Joseph et al. (2015)提出, 最小化能量设计方法的物理背景是将样本点视为盒子内带有相同电性的带电粒子, 当这些带电粒子在盒子内达到平衡时, 这些粒子的总的电势能达到最小, 这些带电粒子达到平衡状态时的位置即是设计点的位置. 函数 $f(\mathbf{x})$ 的包含 $n$ 个点的最下化能量设计 $\{\mathbf{x}_i\}_{i=1}^n$ 可通过最小化下面的式子得到

$$\sum_{i \neq j} \frac{f(\mathbf{x}_i)^{-k/2d} f(\mathbf{x}_j)^{-k/2d}}{d(\mathbf{x}_i, \mathbf{x}_j)}, k \in [1, \infty]. \quad (1.4)$$

当 $k \rightarrow \infty$ , 最小化能量设计的构造可以等价的通过最小化下式得到

$$\max_{i \neq j} \frac{1}{f(\mathbf{x}_i)^{1/2d} f(\mathbf{x}_j)^{1/2d} d(\mathbf{x}_i, \mathbf{x}_j)}. \quad (1.5)$$

找到复杂响应曲面 $f(\mathbf{x})$ 的最小化能量设计是一个计算比较困难的问题, 为此, Joseph et al. (2015)利用模拟退化算法提出一次生成一个点的贪婪算法来生成最小化能量设计, 具体可以描述为, 假定想要获得具有 $N$ 个点的最小化能量设计, 第

一个设计点 $\mathbf{x}_1$  选择为:

$$\mathbf{x}_1 = \arg \max_{\mathbf{x}} f(\mathbf{x}).$$

然后序贯地生成 $\mathbf{x}_2, \mathbf{x}_3, \dots$ . 假如利用准则(1.4) 已经生成了 $n$ 个点, 那么第 $n+1$ 个点可由下式得到:

$$\begin{aligned} \mathbf{x}_{n+1} &= \arg \min_{\mathbf{x}} \left\{ \sum_{i=1}^{n-1} \sum_{j=i+1}^n \frac{f(\mathbf{x}_i)^{-k/2d} f(\mathbf{x}_j)^{-k/2d}}{d(\mathbf{x}_i, \mathbf{x}_j)^k} + \sum_{i=1}^n \frac{f(\mathbf{x}_i)^{-k/2d} f(\mathbf{x})^{-k/2d}}{d(\mathbf{x}_i, \mathbf{x})^k} \right\} \\ &= \arg \min_{\mathbf{x}} \sum_{i=1}^n \frac{f(\mathbf{x}_i)^{-k/2d} f(\mathbf{x})^{-k/2d}}{d(\mathbf{x}_i, \mathbf{x})^k} \\ &= \arg \min_{\mathbf{x}} \frac{1}{f(\mathbf{x})^{k/2d}} \sum_{i=1}^n \frac{1}{f(\mathbf{x}_i)^{k/2d} d(\mathbf{x}_i, \mathbf{x})^k}. \end{aligned}$$

由于上述模拟退火的算法生成最小化能量设计时需要多次进行全局最优化运算, 因此该方法生成最小化能量设计时十分耗时. 鉴于上述方法的缺点, Joseph et al. (2019) 又提出一种快速生成最小化能量设计的算法, 该算法通过最小化准则(1.5)得到包含 $n$ 个点的最小化能量设计, 最小化准则(1.5) 等价于最大化如下准则:

$$\min_{i \neq j} f(\mathbf{x}_i)^{1/2d} f(\mathbf{x}_j)^{1/2d} d(\mathbf{x}_i, \mathbf{x}_j).$$

该快速算法利用 $\mathbf{K}$ 步迭代更新来生成最小化能量设计, 每一步依然采用一次更新一个点的贪婪算法, 从第一个点 $\mathbf{x}_1$  开始, 依次生成 $\mathbf{x}_2, \mathbf{x}_3, \dots$ . 第 $j$ 个点可由下式得到:

$$\mathbf{x}_j = \arg \max_{\mathbf{x}} \min_{i=1:(j-1)} f(\mathbf{x})^{1/2d} f(\mathbf{x}_i)^{1/2d} d(\mathbf{x}, \mathbf{x}_i).$$

Joseph et al. (2019)提出的快速算法具体可描述为: 该算法需要 $\mathbf{K}$ 步迭代更新来生成最小化能量设计, 首先第一步选择一个空间填充设计 $D_1$ 作为最小化能量设计的初始设计, 然后迭代更新 $\mathbf{K}$ 次得到最终的最小化能量设计. 假定第 $k$ 步迭代后的设计表示为 $D_k$ , 那么 $D_{k+1} = \{\mathbf{x}_1^{k+1}, \dots, \mathbf{x}_n^{k+1}\}$  可由依次更新 $D_k$ 中的每一个点得到, 具体如下



step 1. 第  $j$  个点  $\mathbf{x}_j^k$  在其局部区域内通过下式更新:

$$\mathbf{x}_j^{new} = \arg \max_{\mathbf{x} \in L_{jk}} \min_{i=1:(j-1)} \gamma_k \log \hat{f}^{(jk)}(\mathbf{x}) + \gamma_k \log \hat{f}^{(jk)}(\mathbf{x}_i^{new}) + 2d \log d(\mathbf{x}, \mathbf{x}_i),$$

这里  $\gamma_k = \frac{k-1}{K-1}$ ,  $L_{jk}$  表示第  $k$  次更新后  $\mathbf{x}_j^k$  处的邻域,  $\hat{f}^{(jk)}$  表示  $f(\mathbf{x})$  在  $\mathbf{x}_j$  的邻域  $L_{jk}$  处的估计. 该步得到的设计记为  $D_{k+1}^{new} = \{\mathbf{x}_1^{new}, \dots, \mathbf{x}_n^{new}\}$ .

step 2. 进而在整个试验空间内通过更新下式得到  $\mathbf{x}_j^{k+1}$ :

$$\mathbf{x}_j^{k+1} = \arg \max_{\mathbf{x} \in C_{k+1}} \min_{i=1:(j-1)} \gamma_k \log f(\mathbf{x}) + \gamma_k \log f(\mathbf{x}_i^{new}) + 2d \log d(\mathbf{x}, \mathbf{x}_i), j = 2, \dots, n.$$

这里  $C_{k+1} = C_k \cup D_{k+1}^{new}$ , 且  $C_1 = D_1$  以及  $\mathbf{x}_1 = \arg \max_{\mathbf{x} \in C_{k+1}} \log f(\mathbf{x})$ .

例1.2.2. 当密度函数是香蕉形状的函数时<sup>[52]</sup>,

$$f(\mathbf{x}) \propto \exp\{-\frac{1}{2} \frac{x_1^2}{100} - \frac{1}{2}(x_2 - 0.03x_1^2 - 3)^2\}. \quad (1.6)$$

图1.7给出了通过6次迭代得到具有109个设计点的最小化能量设计的过程. 从图中可以看出, 最终得到的最小化能量设计是香蕉状密度函数的一组好的样本点.

### §1.2.2.2 支撑点设计

Mak and Joseph (2018)、Mak and Joseph (2017)提出的支撑点(support points)是另一类确定性抽样方法, 该方法生成的代表点集比最小化能量设计方法生成的代表点集在空间填充性和数值积分上都有优势. 支撑点通过最小化能量距离来生成代表点集, 该能量距离常用于构造拟合优度检验中的检验统计量. 若  $F$  表示密度函数  $f(\mathbf{x})$  对应的分布函数, 则  $F$  的支撑点设计  $\{\mathbf{x}_i\}_{i=1}^n$  可以通过最小化下面的式子得到

$$\arg \min_{\mathbf{x}_1, \dots, \mathbf{x}_n} \frac{2}{n} \sum_{i=1}^n E \|\mathbf{x}_i - \mathbf{Y}\|_2 - \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \|\mathbf{x}_i - \mathbf{x}_j\|_2 - E \|\mathbf{Y} - \mathbf{Y}'\|_2, \quad (1.7)$$

这里  $\mathbf{Y}, \mathbf{Y}' \stackrel{i.i.d.}{\sim} F$ .

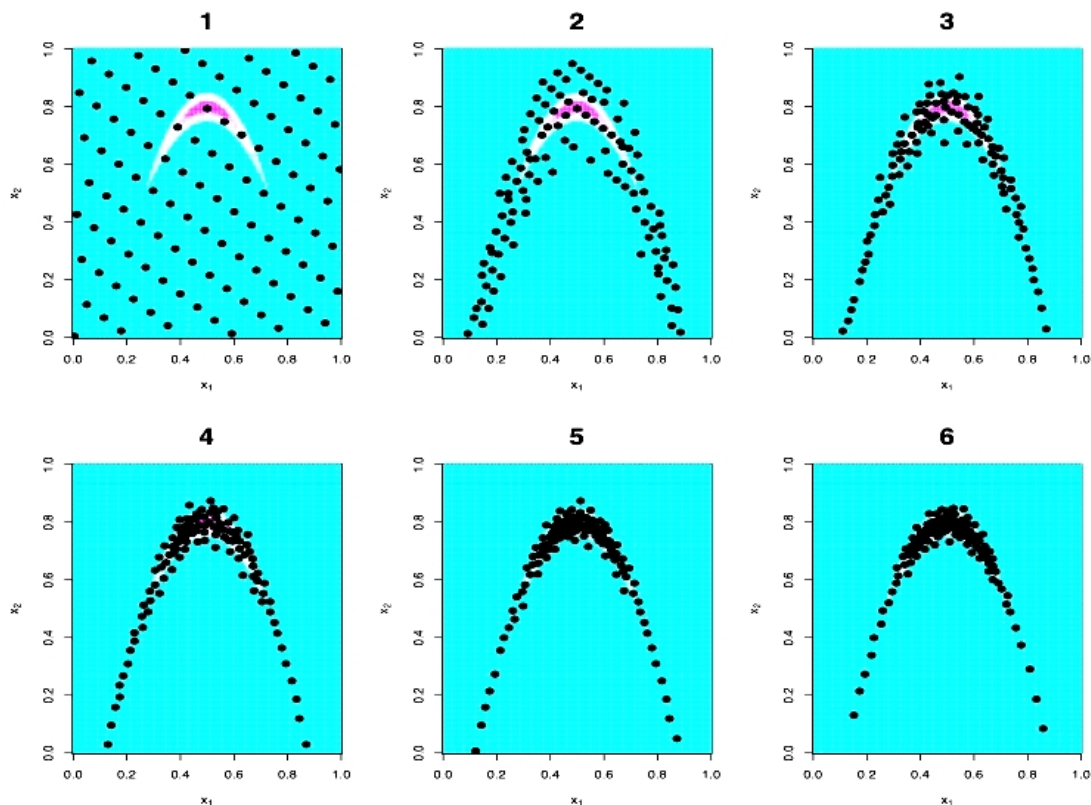


图 1.7 通过6次迭代生成具有109个设计点的最小化能量设计的过程.

直接最小化(1.7)生成支撑点比较困难. 因此, Mak and Joseph (2018)提出近似生成支撑点设计的方法, 首先从分布 $F$ 中获得一个大样本 $\{y_m\}_{m=1}^N$ , 这里 $N \gg n$ , 进而通过最小化(1.7)的Monte Carlo近似来生成 $n$ 个支撑点.

$$\arg \min_{x_1, \dots, x_n} \frac{2}{nN} \sum_{i=1}^n \sum_{m=1}^N \|x_i - y_m\|_2 - \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \|x_i - x_j\|_2, \quad (1.8)$$

Mak and Joseph (2018)利用凸凹优化算法通过最小化(1.8), 提出两种生成支撑点的算法. 算法sp.ccp 和sp.sccp分别用于从分布 $F$ 中获取样本时间成本较高时和较低时的情形见算法1.3-1.4所示, 这里符号 $\leftarrow$ 表示赋值.

**例1.2.3.** 当密度函数分别取2-维*i.i.d* 的指数分布、*Bata* 分布以及正态分布时, 图1.8-1.10 给出了100个支撑点设计的散点图, 从图中可以看出支撑点设计的空间填充性质比较好.

---

**算法 1.3** 支撑点设计的生成算法(sp.ccp)<sup>[2]</sup>

---

**输入:** 从分布 $F$  中获取大样本:  $\{\mathbf{y}_m\}_{m=1}^N$

**输出:**  $\mathbf{x}_1, \dots, \mathbf{x}_n$

- 1: 获取初始设计  $D^0 = \{\mathbf{x}_i^0\}_{i=1}^n$  i.i.d.  $\{\mathbf{y}_m^0\}_{m=1}^N$ .
  - 2: 令  $l = 0$ , 重复下面步骤直到算法收敛得到最终的支撑点设计  $D^l$
  - 3: **for**  $i = 2, \dots, n$  **do**
  - 4:   令  $\mathbf{x}_i^{l+1} \leftarrow (\sum_{m=1}^N \|\mathbf{x}_i^l - \mathbf{y}_m\|_2^{-1})^{-1} \left( \frac{N}{n} \sum_{j=1, j \neq i}^n \frac{\mathbf{x}_i^l - \mathbf{x}_j^l}{\|\mathbf{x}_i^l - \mathbf{x}_j^l\|_2} + \sum_{m=1}^N \frac{\mathbf{y}_m}{\|\mathbf{x}_i^l - \mathbf{y}_m\|_2} \right)$ .
  - 5:   更新  $D^{l+1} \leftarrow \{\mathbf{x}_i^{l+1}\}_{i=1}^n$ , 并且令  $l \leftarrow l + 1$ .
  - 6: **end for**
- 

---

**算法 1.4** 支撑点设计的生成算法(sp.sccp)<sup>[2]</sup>

---

**输入:** 初始设计  $D^0 = \{\mathbf{x}_i^0\}_{i=1}^n$  i.i.d.  $F$

**输出:**  $\mathbf{x}_1, \dots, \mathbf{x}_n$

- 1: 令  $l = 0$ , 重复下面步骤直到算法收敛得到最终的支撑点设计  $D^l$
  - 2: 从分布 $F$  中获取大样本:  $\{\mathbf{y}_m^l\}_{m=1}^N \stackrel{i.i.d.}{F}$
  - 3: **for**  $i = 2, \dots, n$  **do**
  - 4:   令  $\mathbf{x}_i^{l+1} \leftarrow (\sum_{m=1}^N \|\mathbf{x}_i^l - \mathbf{y}_m\|_2^{-1})^{-1} \left( \frac{N}{n} \sum_{j=1, j \neq i}^n \frac{\mathbf{x}_i^l - \mathbf{x}_j^l}{\|\mathbf{x}_i^l - \mathbf{x}_j^l\|_2} + \sum_{m=1}^N \frac{\mathbf{y}_m}{\|\mathbf{x}_i^l - \mathbf{y}_m\|_2} \right)$ .
  - 5:   更新  $D^{l+1} \leftarrow \{\mathbf{x}_i^{l+1}\}_{i=1}^n$ , 并且令  $l \leftarrow l + 1$ .
  - 6: **end for**
-

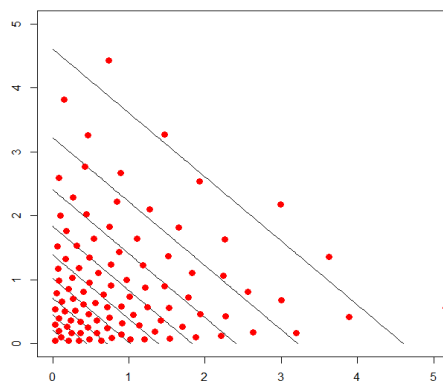


图 1.8 2-维i.i.d 的指数分布 $\text{Exp}(1)$ 的100个支撑点设计.

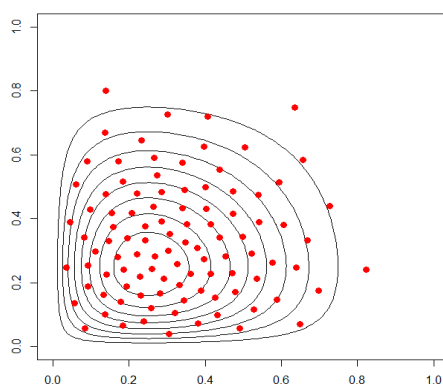


图 1.9 2-维i.i.d 的Beta分布 $\text{Beta}(2,4)$ 的100个支撑点设计.

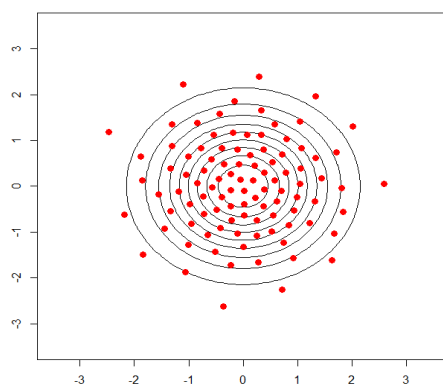


图 1.10 2-维i.i.d 的正态分布 $N(0,1)$ 的100个支撑点设计.

### §1.2.3 同时包含定性和定量因子的设计

计算机试验的标准框架是假设输入因素是定量的. 然而, 在许多计算机试验中, 有一些因素是定性的, 例如, Schmidt et al. (2005)描述的计算机试验涉及了定性因素(如扩散器位置和热风回流排气口的位置)和定量因素(如机架功率和扩散器流量). 许多学者给出了此类计算机试验设计的构造方法. 本节首先介绍常用的切片空间填充设计<sup>[40]</sup>, 如切片拉丁超立方体设计(sliced Latin hypercube design, SLHD), 然后再介绍一类经济高效的边际耦合设计(marginally coupled design, MCD).

#### §1.2.3.1 切片拉丁超立方体设计

第一个系统解决计算机试验中既包含定性因子又包含定量因子的方案是Qian (2012)提出的切片拉丁超立方体设计. 此类设计的核心思想是: 定性因子的每个水平组合对应的定量因子的设计均是一个小的切片拉丁超立方体设计, 并且这些切片合起来之后的设计也是一个拉丁超立方体设计. 算法1.5给出了具有 $d$ 个定量因子和 $t$ 个切片, 每个切片包含 $m$ 个设计点, 设计点总数为 $n = mt$ 的切片拉丁超立方体设计的构造方法.

**例1.2.4.** 依然考虑Schmidt et al. (2005)中的计算机试验既包含定性因子如扩散器的位置( $A$ )和热风回流排气口的位置( $B$ ), 又包含定量因子如电源的功率( $C$ )和扩散器的流量( $D$ ). 定性因子和定量因子的水平设置如下表所示: 定量因子 $C$ 和 $D$

表 1.2 定性因子的取值水平

Factor	Level	
	1	2
$A$ 散器的位置	左	右
$B$ 热风回流排气口的位置	左	右

的取值为 $\{0, 1, \dots, 15\}$ .

针对该试验, 由于定性因子包含4个水平组合, 所以整个设计应包括4个切片, 假定每个切片包含4个设计点, 则该切片拉丁超立方体设计如下表1.3所示.

从这个例子可以看出, 利用切片拉丁超立方体设计在解决既有定性因子又有定量因子的试验时, 即使对于中等数量的定性因子, 此类设计的试验次数都非常

---

**算法 1.5** 切片拉丁超立体设计的构造<sup>[3]</sup>

---

step 1. 将正整数集合  $Z_n = \{1, \dots, n\}$  分为  $m$  个区组, 每个区组记为  $b_i, i = 1, \dots, m$ .  
 $b_i$  可由下式得到:

$$b_i = \{a \in Z_n | [a/t] = i\}, i = 1, \dots, m;$$

step 2. 定义一个  $m \times t$  的矩阵  $H$ ,  $H$  的第  $i$  行为  $b_i$  的一个均匀排列, 并且对  $H$  的每一列再进行一次均匀排列;

step 3. 根据 step 1 和 step 2, 生成  $d$  个独立的矩阵  $H_1, \dots, H_d$ ;

step 4. 对于  $c = 1, \dots, t$ , 定义  $m \times d$  的矩阵  $A^c$ , 其第  $j$  列为  $H_j$  的第  $c$  列,  $j = 1, \dots, d$ .  
 进一步对这  $t$  个矩阵按行合, 得到

$$A = \cup_{c=1}^t A^c;$$

step 5. 若令  $A = (a_{ik})_{n \times d}$ , 那么具有  $d$  列  $n$  行,  $t$  个切片每个切片包含  $m$  个设计点的切片拉丁超立体设计  $D = (d_{ik})_{n \times d}$  可由下式得到:

$$d_{ik} = \frac{a_{ik} - u_{ik}}{n}, i = 1, \dots, n, k = 1, \dots, d,$$

这里  $u_{ik}$  是相互独立的  $U(0, 1]$  上的随机变量.

---

表 1.3 切片空间填充设计

$A$	$B$	$C$	$D$
1	1	0	2
1	1	4	5
1	1	8	9
1	1	12	12
1	2	1	4
1	2	5	0
1	2	9	15
1	2	13	8
2	1	2	10
2	1	6	14
2	1	10	1
2	1	14	7
2	2	3	13
2	2	7	11
2	2	11	6
2	2	15	3

大. 为此, Deng et al. (2015)提出了一种比较经济的策略来设计既有定性因子又有定量因子的试验, 称为边际耦合设计(marginal coupled design, MCD).

### §1.2.3.2 边际耦合设计

Deng et al. (2015)给出了包含 $p$ 个定性因子和 $q$ 个定量因子, 且每个定性因子有 $s$ 个水平的边际耦合设计的定义.

**定义1.2.4.** 令 $D_1 = OA(n, s^p, 2)$ 是一个 $n$ 行,  $p$ 列, 每一列有 $s$ 个水平且强度为2的正交表,  $D_2$ 是一个 $n \times q$ 的拉丁超立体设计, 如果 $D_1$ 中每个因子的每个水平行对应于 $D_2$ 中的行都是一个小的拉丁超立体设计, 则称设计 $D = (D_1, D_2)$ 是一个边际耦合设计, 并表示为 $MCD(D_1, D_2)$ , 其中 $D_1$ 和 $D_2$ 分别是定性和定量因子对应的设计.

**例1.2.5.** 考虑例1.2.4中包含两个定性因子和两个定量因子的试验, 表1.4给出了具有两个2水平的定性因子 $A, B$ 以及两个定量因子 $x_1(C), x_2(D)$ 的 $MCD(D_1, D_2)$ 的示例. 图1.11-1.12给出了定量因子 $A$ 和 $B$ 对应的定量因子 $x_1$ 与 $x_2$ 的散点图. 从图中可以看出, 投影到 $x_1$ 或 $x_2$ 上, 只有一个‘1’或‘2’位于 $(0, 1/4), [1/4, 2/4), [2/4, 3/4), [3/4, 1]$ 四

个区间中的一个. 因此, 对于 $D_1$ 中每个因子的每个水平与之相应的 $D_2$ 中的行可以实现一维投影均匀性.

表 1.4 边际耦合设计MCD( $D_1, D_2$ ).

$D_1$		$D_2$	
$A$	$B$	$x_1$	$x_2$
1	1	0.1129	0.2155
1	2	0.7721	0.2900
1	2	0.4131	0.9181
1	1	0.5854	0.6047
2	2	0.6312	0.8601
2	1	0.3707	0.6877
2	1	0.9452	0.0050
2	2	0.2023	0.4575

其中 $D_1$ 是一个 $OA(8, 2^2, 2)$ ,  $D_2$ 是一个 $LHD(8, 2)$ .

为了提高定量因子设计部分的空间填充性质, He et al. (2017)利用有限域给出了几类新的边际耦合设计的构造方法.



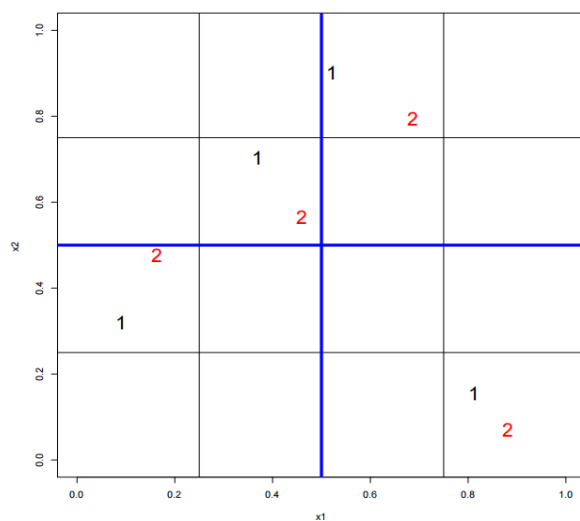


图 1.11 定性因子A对应的定量因子 $x_1$ 与 $x_2$ 的散点图.

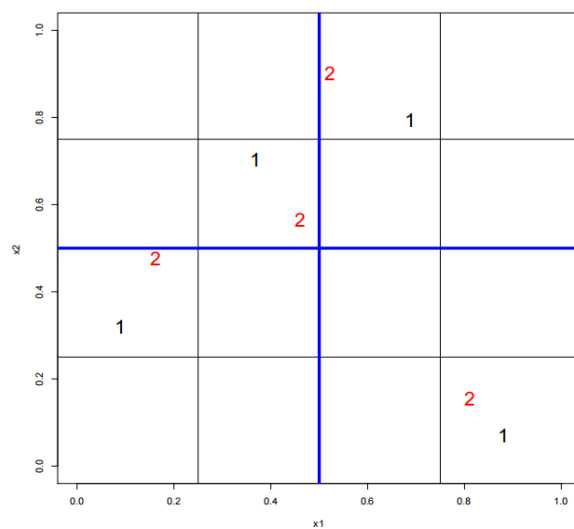


图 1.12 定性因子B对应的定量因子 $x_1$ 与 $x_2$ 的散点图.

### §1.3 本文的主要工作和结构安排

当没有响应曲面的先验信息时,空间填充设计是探索响应曲面比较流行的设计方法,该设计方法同等看待设计空间中的任何区域,使试验点均匀地分散在区域内.但是当获取响应曲面一定的信息后,仍然使用空间填充设计有时会抓不住响应曲面的趋势特征.例如,若试验的目的是寻找响应曲面的最大值,且知道曲面具有多个局部最大值时,空间填充设计难免会使许多设计点落入低响应值区域,进而造成浪费.因此,本文基于Kullback-Leibler 偏差提出三类新的确定性抽样方法来获得复杂响应曲面的代表点.此外,当复杂响应曲面即包含定性因子又包含定量因子时,边际耦合设计是常用的抽样设计方法.边际耦合设计是一类比切片空间填充设计更经济有效的抽样设计方法.为了从理论上说明边际耦合设计的合理性,本文研究了边际耦合设计的大样本性质,进一步导出了边际耦合设计的中心极限定理.这些大样本性质可以广泛的应用于求参数估计的置信区间等.本文的结构安排如下.

第一章详细介绍了Kriging建模方法和几类空间填充设计如拉丁超立方体设计<sup>[12]</sup>、均匀设计<sup>[13]</sup>、以及这两类设计的推广如基于正交表的拉丁超立方体设计<sup>[20]</sup>和正交拉丁超立方体设计<sup>[15, 17, 18]</sup>等.如果复杂响应曲面包含多个局部最优值或者试验前已知响应曲面的解析式时,仍然选择利用空间填充设计探索复杂响应曲面时会造成设计点的浪费.为此本章介绍了已有的确定性抽样方法(代表点),即最小化能量设计<sup>[1, 38]</sup>和支撑点设计<sup>[2, 39]</sup>,这两类设计又称为加权的空间填充设计.另外,本章还介绍了切片空间填充设计和边际耦合设计,它们是处理响应曲面同时包含定性和定量因子的两类常用设计方法.

第二章针对当获取响应曲面一定的信息后,例如已知复杂响应曲面包含多个局部最优值或者已知响应曲面的解析式.为了使设计点均落入感兴趣的区域,本章基于Kullback-Leibler偏差和函数的核估计方法提出了一种新的确定性抽样方法来得到响应曲面的一组代表点.该抽样方法会在感兴趣的区域倾向于布更多的点,且不同的点之间保持一定地距离.得到的点集称为Kullback-Leibler(KL)点,是一类加权的空间填充设计方法.进一步,理论结果表明KL点确实是复杂响应曲面的代表点.数值模拟表明KL点与支撑点和最小化能量设计相比,在空间填充性质

与数值积分上更具优势(见图2.2-2.4, 表2.1-2.2).

第三章根据响应曲面的不同复杂结构提出了两类推广的KL点. 首先, 针对响应曲面没有显式表达或者计算耗时, 基于Kullback-Leibler偏差和函数的核估计方法, 提出了一类推广的Kullback-Leibler点称为自适应的Kullback-Leibler(AKL)点. 在探究此类复杂结构的响应曲面时, AKL点比传统的空间填充设计更高效. 而且数值模拟表明AKL点与其他确定性抽样相比, 如序贯最小化能量设计, 无论是在复杂密度函数的抽样上还是在探索和优化复杂的黑盒子函数上均有明显的优势(见图3.1-图3.3). 其次, 针对响应曲面的维数较高时, 基于Kullback-Leibler偏差和函数的最小邻近距离估计方法, 提出了另一类推广的Kullback-Leibler点称为Kullback-Leibler-nn(KL-nn)点. KL-nn点依然能够抓住响应曲面地趋势特征, 并且不同点之间保持一定的距离. 数值模拟表明, KL-nn点有较好的空间填充性质(见图3.6), 且在高维情形下的数值积分上比支撑点和最小化能量设计有较小的积分误差(见图3.7).

第四章介绍了边际耦合设计的大样本性质. 边际耦合设计是处理同时包含定性和定量因子模型的常用设计方法, 比切片空间填充设计更经济. 虽然有一些文献已经讨论了边际耦合设计的构造方法, 但关于其抽样性质还没有得到研究. 本章导出了边际耦合设计的大样本性质, 并进一步导出了相应的中心极限定理. 这些结果广泛用于建立参数估计的置信区间以及量化估计的不确定性等.

第五章是本文的总结和未来研究的工作.

附录部分给出了部分数值模拟的程序代码.



## 第二章 基于Kullback-Leibler散度的复杂响应曲面的抽样方法

本章基于Kullback-Leibler偏差提出一类复杂响应曲面好的确定性抽样方法. 本章的思想是将复杂响应曲面转化为非标准化的密度函数, 然后通过最小化一组点的核密度估计与目标密度之间的Kullback-Leibler偏差来获取代表点集, 并记为KL点. Kullback-Leibler偏差是信息论中用于度量两个密度函数之间相似性的偏差, 且常用于构造拟合优度检验统计量. 基于全变差与Kullback-Leibler偏差之间的关系, 本章证明了KL点作为响应曲面代表点的合理性. 本章通过两步近似来生成 $n$ 个KL点. 首先, 为了避免高维积分, 通过最小化Kullback-Leibler偏差的Monte Carlo近似来生成KL点. 其次, 由于同时生成 $n$ 个KL点比较困难, 为此采用一次生成一个点的策略. 进一步, 数值模拟表明: KL点不仅比MCMC抽样和其他确定性抽样方法有更好的空间填充性质; 而且在数值积分上, KL点方法下的积分误差比MCMC抽样以及其他确定性抽样方法下的积分误差要小.

### §2.1 背景介绍

许多现代科学试验均涉及到探索复杂响应曲面. 探索复杂响应曲面的目标常常包含以下几个方面: i) 预测复杂响应曲面的响应值; ii) 探索响应值超过期望值的试验区域; 3) 找到复杂响应曲面的全局最优值等等. 当试验前对这个复杂响应曲面没有任何先验信息时, 传统的探索复杂响应曲面的方法是使得样本点均匀地散布在试验区域内, 以此来尽可能多的获取关于复杂响应曲面的信息. 常用的空间填充设计有均匀设计<sup>[13, 27]</sup>, 拉丁超立方体设计<sup>[12]</sup>, 极大极小距离设计<sup>[35]</sup>, 正交拉丁超立方体设计<sup>[17]</sup>, 均匀投影设计<sup>[33]</sup>等等. 如果试验前已经获得复杂响应曲面的一些信息, 比如已知该曲面的显式表达式 $f(\mathbf{x})$ , 或者已知该复杂响应曲面包含多个局部最优值, 这时再用空间填充设计探索复杂响应曲面会造成设计点的浪费.

从复杂响应曲面获取代表点集的方法可以转化为统计学中常见的获取概率密度函数的代表点的方法, 即把该复杂响应曲面转化为非标准化的密度函数, 然后从该密度函数中获取代表点集, 这就是常见的抽样问题. 例如贝叶斯校准<sup>[42]</sup>和

机器学习中一个重要的研究目标是从一个复杂的贝叶斯后验密度函数中抽样, 这个复杂的贝叶斯后验密度函数可以看作是一类复杂的响应曲面. 常用的从复杂的概率密度函数中获取代表点集的方法是Markov Chain Monte Carlo<sup>[43-51]</sup> (MCMC). MCMC 是一种随机的抽样方法. 由于MCMC方法需要大量地计算密度函数的值, 因此利用MCMC方法获取样本时计算成本较高. 当密度函数易于计算时, 这一成本往往可以忽略不计. 但是, 当密度函数比较复杂并且计算一次密度函数值耗时时, 这个成本往往非常巨大且不可忽略. 例如研究人员经常需要等待数小时甚至数天时间使得MCMC链收敛到一个平稳分布, 删除前面不收敛的点, 后面产生的样本点才是最终的样本点. 除此之外, MCMC产生的样本点会出现点与点的距离很近或者重复点的现象, 出现距离很近的点或者重复的点, 从计算机试验设计的角度看, 因为这些点不提供额外的信息, 因此会造成点的浪费. 例如, 当密度函数是香蕉形状的函数时<sup>[52]</sup>,

$$f(\mathbf{x}) \propto \exp\left\{-\frac{1}{2} \frac{x_1^2}{100} - \frac{1}{2}(x_2 - 0.03x_1^2 - 3)^2\right\}. \quad (2.1)$$

图2.1的左边给出了香蕉状密度函数的1000个MCMC样本点, 可以看到尽管大多数点都集中在密度值高的区域, 但是这1000个样本点中有很多重复点并且点与点之间的距离很近, 这些重复的点和距离很近的点不会提供额外的信息, 因此会浪费很多点. 如果能使得这些点之间的距离尽量地隔开, 那么将会获得更多关于密度函数的信息.

为了解决点聚集的现象, 研究人员可采用确定性的抽样方法来替代随机化抽样方法. Quasi-Monte Carlo (QMC)是一种常用的确定性抽样方法<sup>[53]</sup>, 该方法可以避免点的聚集. 然而, 尽管QMC 的方法可以使点很好的隔开, 但是这样的点集很少落在密度函数值高的区域, 那些落入密度函数值为0的区域同样会造成点的浪费. 这是QMC 方法的主要缺点, 造成这一现象的原因是QMC利用均匀分布进行抽样. 尽管QMC的其他文献提出了一种非均匀分布的抽样策略, 即使用分布函数的逆函数将均匀分布上的点集映射到非均匀分布 $F$ 上. 然而, 这一策略实施的要求是各变量是相互独立的, 这在现代科学试验中是很少见的. 为了克服MCMC和QMC的不足之处, 我们希望基于一些准则利用确定性的抽样方法来

获得一组好的代表点, 即样本点大都集中在密度值高的区域, 且不同的样本点之间保持一定的距离.

为此, 一个好的确定性抽样应该满足:

- (1) 使得更多的点集中在密度值高的区域;
- (2) 不同点之间保持一定的距离;
- (3) 尽量避免大量计算复杂的密度函数值.

Joseph et al. (2015, 2019), Mak and Joseph (2017, 2018)提出了几类确定性抽样方法. 然而这两类抽样方法在实际应用中又有各自的局限.

本章基于Kullback-Leibler偏差通过密度函数的核估计方法提出一类新的从连续分布函数 $F$ 中抽取代表点集的方法, 记为KL点. Kullback-Leibler偏差最先由Kullback and Leibler (1951) 提出, 该偏差广泛用于度量两个概率分布的相似性, 且Kullback-Leibler偏差是一类特殊的 $\phi$ -偏差<sup>[55]</sup>. Kullback-Leibler 偏差常应用于构造拟合优度检验统计量, 且在检验功效上比传统的Kolmogorov-Smirnov检验统计量<sup>[56]</sup>更高效. Jourdan and Franco (2009)通过最小化设计点的经验分布函数与均匀分布之间的Kullback-Leibler偏差构造了一类新的空间填充设计. 受此启发, 本章把这一想法从均匀分布推广到一般的分布函数, 进而构造一类加权的空间填充设计. 本章的核心思想是最小化一组点的经验分布函数与目标分布函数之间的Kullback-Leibler偏差, 这样生成的点集之间的距离不是很近, 并且密度值高的区域分布的样本点多, 密度值低的区域分布的样本点少, 即得到目标分布的一组好的代表点集. 图2.1的右边给出了从香蕉形状的密度函数中得到的100个KL点, 可以看出这100个KL点看起来并不比1000个MCMC点少. 换句话说, 从这个例子可以看出, 较少的KL点所提供的分布的信息并不比较多的MCMC点提供的信息少, 这是KL 点的明显优势.

图2.2-2.3分别给出了混合正态分布和二元正态分布在四种代表点方法下的散点图, 从图中可以看出KL点飞出密度曲线的轮廓线的个数要比最小化能量设计和支撑点都要少, 因此在本例中KL 点比最小化能量设计和支撑点设计有更好的空间填充性质见. 进一步, 从图2.4可以看出KL点比最小化能量设计和支撑点在

数值积分上的表现也比较好, 即利用KL点在求复杂函数的积分时误差较小.

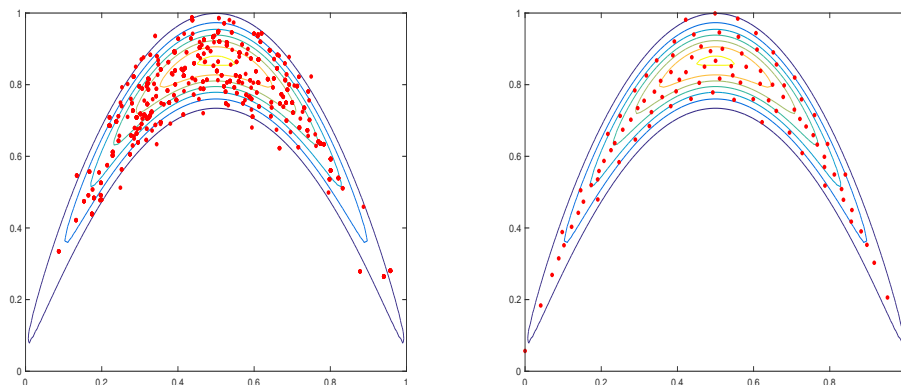


图 2.1 1000 MCMC 样本点v.s. 100 个确定性抽样点.

## §2.2 基于核密度估计的Kullback-Leibler 点

本节介绍了Kullback-Leibler偏差的定义, 以及如何利用Kullback-Leibler偏差产生Kullback-Leibler点. 本节的后半部分给出了KL点几个重要的理论结果.

**定义2.2.1.** <sup>[58]</sup> 假定 $f(\mathbf{x})$  和 $g(\mathbf{x})$  为两个密度函数, 并且 $f(\mathbf{x})$  的支撑集为 $\mathcal{X} \subseteq \mathbb{R}^d$ , 则 $f(\mathbf{x})$  和 $g(\mathbf{x})$  之间的Kullback-Leibler 偏差定义为

$$D(g\|f) = \int_{\mathcal{X}} g(\mathbf{x}) \log \frac{g(\mathbf{x})}{f(\mathbf{x})} d\mathbf{x}. \quad (2.2)$$

**命题2.2.1.** <sup>[58]</sup>  $D(g\|f) \geq 0$ , 且 $D(g\|f) = 0$  当且仅当 $g(\mathbf{x}) = f(\mathbf{x})$ .

为了从给定的目标密度中获取确定性样本, 可以利用一组点的核密度估计与目标密度函数之间的Kullback-Leibler偏差来度量这组点的代表性. 令 $f_n(\mathbf{x})$ 表示 $\{\mathbf{x}_i\}_{i=1}^n \subseteq \mathcal{X}$ 上的核密度估计, 那么 $f(\mathbf{x})$  与 $f_n(\mathbf{x})$ 之间的Kullback-Leibler偏差为

$$D(f_n\|f) = \int_{\mathcal{X}} f_n(\mathbf{x}) \log \frac{f_n(\mathbf{x})}{f(\mathbf{x})} d\mathbf{x}. \quad (2.3)$$



这里 $f_n(\mathbf{x})$ 表示基于 $\{\mathbf{x}_i\}_{i=1}^n$ 的核密度估计, 其定义如下

$$f_n(\mathbf{x}) = \frac{1}{nh_n^d} \sum_{i=1}^n k\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right), \quad (2.4)$$

这里核函数 $k(\mathbf{t})$ 和带宽 $h_n$ 满足下列条件:

(K1)  $k(\mathbf{t})$ 是一个概率密度函数且其支撑集是 $\mathcal{X} \subseteq \mathbb{R}^d$ ;

(K2)  $k(-\mathbf{t}) = k(\mathbf{t}), \forall \mathbf{t} \in \mathcal{X}$ ;

(K3)  $0 < \int_{\mathcal{X}} \mathbf{t}' \mathbf{t} k(\mathbf{t}) d\mathbf{t} < \infty$ ;

(K4)  $\|\mathbf{k}\|_2^2 = \int_{\mathcal{X}} k^2(\mathbf{t}) d\mathbf{t} < \infty$ ;

(K5)  $h_n \rightarrow 0, nh_n^d \rightarrow \infty$  as  $n \rightarrow \infty$ .

如果 $f(\mathbf{x})$ 的支撑集包含于 $\mathcal{X}$ ,  $f_n(\mathbf{x})$ 是(2.4)中定义的核密度估计, 则 $D(f_n \| f)$ 可度量了 $f_n$ 与 $f$ 的距离. 因此, 可以通过最小化 $D(f_n \| f)$ 来得到Kullback-Leibler点. 本文把这种基于核密度估计生成的点集称为KL点.

**定义2.2.2.** (KL点). 对于固定的点集大小 $n \in \mathbb{N}$ ,  $f(\mathbf{x})$ 的Kullback-Leibler(KL)点定义如下:

$$\{\xi_i\}_{i=1}^n = \arg \min_{\mathbf{x}_1, \dots, \mathbf{x}_n} D(f_n \| f) = \arg \min_{\mathbf{x}_1, \dots, \mathbf{x}_n} \int_{\mathcal{X}} f_n(\mathbf{x}) \log \frac{f_n(\mathbf{x})}{f(\mathbf{x})} d\mathbf{x}. \quad (2.5)$$

若令 $F(\mathbf{x})$ 表示 $f(\mathbf{x})$ 对应的累积分布函数. 接下来, 本节介绍几个理论结果, 这些结果从理论上保证了KL点作为代表点的合理性. 定理2.2.1 表明基于 $n$ 个KL点的核密度估计函数 $f_n(\mathbf{x})$ 收敛到目标函数 $f(\mathbf{x})$ . 定理2.2.2 表明该核密度估计 $f_n(\mathbf{x})$ 对应的分布收敛到真实的目标分布函数 $F(\mathbf{x})$ . 基于上述两个定理, 定理2.2.3表明KL点的标准经验分布函数也收敛到真实的概率分布函数 $F(\mathbf{x})$ .

**定理2.2.1.** 假定密度函数 $f(\mathbf{x})$ 满足:

(A1)  $f(\mathbf{x})$ 的支撑集包含于 $\mathcal{X}$ , 且 $\mathcal{X} \in \mathbb{R}^d$ ;

(A2)  $f(\mathbf{x})$  满足李普希兹连续, *i.e.*,

$$|f(\mathbf{x}) - f(\mathbf{y})| \leq L \|\mathbf{x} - \mathbf{y}\|_2, \forall \mathbf{x}, \mathbf{y} \in \mathcal{X},$$

这里  $\|\mathbf{z}\|_2 = \sqrt{\mathbf{z}'\mathbf{z}}$  表示  $\mathbf{z}$  的欧式范数, 且  $L > 0$ .

对于满足条件(K1)–(K5)的核函数  $k(t)$  和带宽  $h_n$ , 令  $D = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  表示  $f(\mathbf{x})$  的KL点, 则有

$$\lim_{n \rightarrow \infty} D(f_n \| f) = 0 \quad (2.6)$$

和

$$\lim_{n \rightarrow \infty} \int_{\mathcal{X}} |f_n(\mathbf{x}) - f(\mathbf{x})| d\mathbf{x} = 0. \quad (2.7)$$

定理2.2.1的证明依赖于如下引理2.2.1, 该引理表明, 当  $\{\mathbf{x}_j^*\}_{j=1}^n \stackrel{i.i.d.}{\sim} f(\mathbf{x})$ , 那么当  $n \rightarrow \infty$ ,  $D(f_n^* \| f)$  的期望收敛到0.

**引理2.2.1.** 假定  $f(\mathbf{x})$  满足(A1)–(A2), 且  $\{\mathbf{x}_j^*\}_{j=1}^n \stackrel{i.i.d.}{\sim} f(\mathbf{x})$ .  $f_n^*(\mathbf{x})$  是  $\{\mathbf{x}_j^*\}_{j=1}^n$  上的核密度估计, 核函数  $k(t)$  以及带宽  $h_n$  满足(K1)–(K5), 则有

$$\lim_{n \rightarrow \infty} E[D(f_n^* \| f)] = 0.$$

根据上述准备, 令  $F_n(\mathbf{x})$  表示核密度估计函数  $f_n(\mathbf{x})$  对应的累积分布函数. 下面的定理表明  $F_n(\mathbf{x})$  收敛到目标分布函数  $F(\mathbf{x})$ .

**定理2.2.2.** 假定密度函数  $f(\mathbf{x})$  满足条件(A1)–(A2), 核函数  $k(t)$  和带宽  $h_n$  满足条件(K1)–(K5), 且  $D = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  为  $f(\mathbf{x})$  的KL点. 令  $F_n(A) = \int_A f_n(\mathbf{x}) d\mathbf{x}$ ,  $A \in \mathcal{B}$ , 这里  $\mathcal{B}$  为  $\mathcal{X}$  上的  $\sigma$ -代数, 则

$$\lim_{n \rightarrow \infty} \sup_{A \in \mathcal{B}} |F_n(A) - F(A)| = 0,$$

进而, 如果  $A = (-\infty, \mathbf{x}] \in \mathcal{B}$ , 有

$$\lim_{n \rightarrow \infty} \sup_{\mathbf{x}} |F_n(\mathbf{x}) - F(\mathbf{x})| = 0.$$

进一步, 定理2.2.1和定理2.2.2能导出基于KL点的标准经验分布函数收敛到目标分布函数 $F(\mathbf{x})$ . 这里 $\mathbf{x}_i \leq \mathbf{x}$ 表示向量 $\mathbf{x}_i$ 的每个分量小于等于向量 $\mathbf{x}$ 的每个分量.

**定理2.2.3.** 假定密度函数 $f(\mathbf{x})$ 满足(A1)-(A2), 核函数 $k(\mathbf{t})$ 和带宽 $h_n$ 满足(K1)-(K5), 且 $D = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ 为 $f(\mathbf{x})$ 的KL点. 令 $F_n^{KL}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n I(\mathbf{x}_i \leq \mathbf{x})$ 表示KL点的标准经验分布函数, 则有

$$\lim_{n \rightarrow \infty} \sup_{\mathbf{x}} |F_n^{KL}(\mathbf{x}) - F(\mathbf{x})| = 0.$$

定理2.2.2和定理2.2.3表明KL点确实是分布 $F$ 一组好的代表点.

## §2.3 序贯生成KL点的算法

在实际中直接最小化(2.5)式生成KL点比较困难. 因此本节提出一种序贯算法来生成KL点.

首先, 为了避免高维积分, 选择优化 $D(f_n \| f)$ 的Monte Carlo近似 $\hat{D}(f_n \| f)$ 来生成KL点, 即

$$\arg \min_{\mathbf{x}_1, \dots, \mathbf{x}_n} \hat{D}(f_n \| f) = \frac{1}{n} \sum_{i=1}^n \log \frac{f_n(\mathbf{x}_i)}{f(\mathbf{x}_i)}, \quad (2.8)$$

其中 $f_n(\mathbf{x}) = \frac{1}{nh_n^d} \sum_{i=1}^n k(\frac{\mathbf{x}-\mathbf{x}_i}{h_n})$ , 其次, 核函数 $k(\mathbf{t})$ 取为多元的指数幂分布, 即

$$k(\mathbf{t}) \propto \exp(-\|\mathbf{t}\|_2). \quad (2.9)$$

其次, 带宽 $h_n$ 依据近似的积分均方误差准则<sup>[59]</sup>, 选择为关于 $n$ 的递减函数,

$$h_n = \left( \frac{d \|\mathbf{k}\|_2^2}{n \mu_2^2(\mathbf{k}) \int_{\mathcal{X}} [\text{tr}\{H_f(\mathbf{x})\}]^2 d\mathbf{x}} \right)^{1/(d+4)} = O(n^{-1/(d+4)}).$$

这里 $H_f(\mathbf{x})$ 为关于 $f(\mathbf{x})$ 二阶导数的黑塞矩阵.

下面介绍序贯生成KL点的步骤. 假定已经生成 $m-1$ 个KL点, 则第 $m$ 个点可以通过下面的公式得到:

$$\begin{aligned} \mathbf{x}_m &= \arg \min_{\mathbf{x} \in \mathcal{X}} \hat{D}(f_m \| f) \\ &= \arg \min_{\mathbf{x} \in \mathcal{X}} \frac{1}{m} \left\{ \sum_{i=1}^{m-1} \log \frac{mh_m^d \sum_{j=1}^{m-1} k(\frac{\mathbf{x}_i - \mathbf{x}_j}{h_m})}{f(\mathbf{x}_i)} + \log \frac{mh_m^d \sum_{j=1}^{m-1} k(\frac{\mathbf{x} - \mathbf{x}_j}{h_m})}{f(\mathbf{x})} \right\} \\ &= \arg \min_{\mathbf{x} \in \mathcal{X}} \log \frac{mh_m^d \sum_{j=1}^{m-1} k(\frac{\mathbf{x} - \mathbf{x}_j}{h_m})}{f(\mathbf{x})} \\ &= \arg \min_{\mathbf{x} \in \mathcal{X}} M(\mathbf{x} | \mathbf{x}_1, \dots, \mathbf{x}_{m-1}), \end{aligned}$$

这里  $M(\mathbf{x} | \mathbf{x}_1, \dots, \mathbf{x}_{m-1}) = \log \frac{mh_m^d \sum_{j=1}^{m-1} k(\frac{\mathbf{x} - \mathbf{x}_j}{h_m})}{f(\mathbf{x})}$ .

特别地, 第一个点 $\mathbf{x}_1$ 取为

$$\mathbf{x}_1 = \arg \min_{\mathbf{x} \in \mathcal{X}} \hat{D}(f_1 \| f) = \arg \min_{\mathbf{x} \in \mathcal{X}} \log \frac{1}{f(\mathbf{x})},$$

即第一个点 $\mathbf{x}_1$ 为 $f(\mathbf{x})$ 的全局最大值.

序贯生成每一个KL点均需要在 $\mathcal{X}$ 上进行全局最优化. 常用的优化算法有模拟退火方法, 但是该优化算法当目标函数比较复杂时比较耗费时间, 为此本节提出一种高效的随机Nelder-Mead优化算法. Nelder-Mead优化算法<sup>[60]</sup>是一种单纯形法, 常见于数值优化理论中. 而本文所采用的随机Nelder-Mead优化算法相比传统的Nelder-Mead优化算法能避免算法陷入局部最优值. 随机Nelder-Mead优化算法通过最小化 $M(\mathbf{x} | \mathbf{x}_1, \dots, \mathbf{x}_{m-1})$ 来得到第 $m$ 个KL点. 随机Nelder-Mead优化算法需要定义一个提议分布来避免算法陷入局部区域, 该算法需要的输入有:

- (1) KL点的个数 $n$ ;
- (2) 从提议分布抽取的初始点的个数 $n_{\text{init}}$ ;
- (3) 提议分布变成自适应分布之前生成的KL点的个数 $n_{\text{delay}}$ ;
- (4) 初始提议分布的均值 $\mu_0$ 和协方差 $\Sigma_0$ ;
- (5) 自适应的混合分布的每个部分的方差 $\lambda, \lambda > 0$ ;

(6) 有界区域的下界 $l$ 和上界 $u$ .

假定初始的提议分布选择为限制在空间 $[l, u]^d$ 上的多元正态分布 $\mathcal{N}(\mu_0, \Sigma_0)$ , 自适应的提议分布选为限制在空间 $[l, u]^d$ 上的混合多元正态分布

$$\Pi(\{\mathbf{x}_j\}_{j=1}^{m-1}, \lambda) := \frac{1}{m-1} \sum_{j=1}^{m-1} \mathcal{N}(\mathbf{x}_j, \lambda I).$$

$\text{NelderMead}_x[M(\mathbf{x}|\mathbf{x}_1, \dots, \mathbf{x}_{m-1}), \mathbf{x}_i^{\text{init}}, l, u]$  表示目标函数为 $M(\mathbf{x}|\mathbf{x}_1, \dots, \mathbf{x}_{m-1})$ 的标准随机Nelder-Mead程序,  $\mathbf{x}_i^{\text{init}}$ 是从提议分布抽取初始点集, 支撑集为 $\mathbf{x} \in [l, u]^d$ . 运算符 $\text{trunc}_l^u[\cdot]$ 表示分布的支撑集是超立方体 $[l, u]^d$ . 下面的算法2.1 给出了生成 $n$ 个KL点的详细步骤, 其中第一个点 $\mathbf{x}_1$ 为 $f(\mathbf{x})$ 的全局最大值.

---

**算法 2.1** 通过随机Nelder-Mead 方法生成 $n$ 个KL点

---

输入:  $n, n_{\text{init}}, n_{\text{delay}}, \mu_0, \Sigma_0, \lambda, l, u$

输出:  $\mathbf{x}_2, \dots, \mathbf{x}_n$

```

1: for  $m = 2, \dots, n$  do
2:   基于 $\{\mathbf{x}_i\}_{i=1}^{m-1}$ 计算 $M(\mathbf{x}|\mathbf{x}_1, \dots, \mathbf{x}_{m-1})$ .
3:   for  $i = 1, \dots, n_{\text{init}}$  do
4:     if  $m \leq n_{\text{delay}}$  then
5:        $\mathbf{x}_i^{\text{init}} \leftarrow \text{trunc}_l^u[\mathcal{N}(\mu_0, \Sigma_0)]$ 
6:     else
7:        $\mathbf{x}_i^{\text{init}} \leftarrow \text{trunc}_l^u[\Pi(\{\mathbf{x}_j\}_{j=1}^{n_{\text{curr}}}, \lambda)]$ 
8:     end if
9:      $\mathbf{x}_i^{\text{local}} \leftarrow \text{NelderMead}_x[M(\mathbf{x}|\mathbf{x}_1, \dots, \mathbf{x}_{m-1}), \mathbf{x}_i^{\text{init}}, l, u]$ 
10:  end for
11:   $i^* \leftarrow \text{argmin}_{i \in \{1 \dots n_{\text{init}}\}} M(\mathbf{x}|\mathbf{x}_1, \dots, \mathbf{x}_{m-1})$ 
12:   $\mathbf{x}_m \leftarrow \mathbf{x}_{i^*}^{\text{local}}$ 
13: end for

```

---

## §2.4 数值模拟

从最小化Kullback-Leibler偏差的意义上来说, KL点可被视为目标分布函数的一组最优样本, 即在密度值高的区域分布的点多, 在密度值低的区域分布的点少, 并且点与点尽量地隔开, 故可称为分布函数的一组代表点. 这样好的性质使得KL点在空间填充性和Monte Carlo积分上均表现较好. 本节通过几个例子来展示KL点相比于Monte Carlo (MC) 点、最小化能量设计点(MED points)和支撑点设计(Support points) 的优势.

### §2.4.1 空间填充性质

为了直观体现空间填充性质, 图2.2给出了利用四种方法从2-维混合正态分布中抽取 $n = 100$ 的散点图, 即

$$P = \frac{1}{2}N(\mu_1, \Sigma_1) + \frac{1}{2}N(\mu_2, \Sigma_2),$$

这里 $\mu_1 = (-1.5, 0)'$ ,  $\mu_2 = (1.5, 0)'$ , 且 $\Sigma_1 = \Sigma_2 = I$ . 图中的实线表示密度函数的等高线. 从图中可以看出KL点比最小化能量设计和支撑点设计更具有代表性, 并且KL点的空间填充性质明显比MC点好, 这是因为MC点会出现点与点之间粘连的现象. 显然, KL点分布在密度值高的区域并且点与点之间的距离不是很近, 而且没有出现重复点以及点与点之间粘连的现象.

图2.3给出了利用四种方法从2-维正态分布中抽取 $n = 100$ 的散点图, 即

$$P = N(\mu, \Sigma),$$

这里 $\mu = (0, 0)'$ ,  $\Sigma_{ij} = 0.5^{|i-j|}$ ,  $i, j = 1, 2$ . 图中的实线表示密度函数的等高线. 该图同样表明KL点比最小化能量设计点和支撑点更能代表密度函数 $f(\mathbf{x})$ , 并且KL点的空间填充性质明显比MC点好, 这是因为MC点会出现点与点之间粘连的现象.

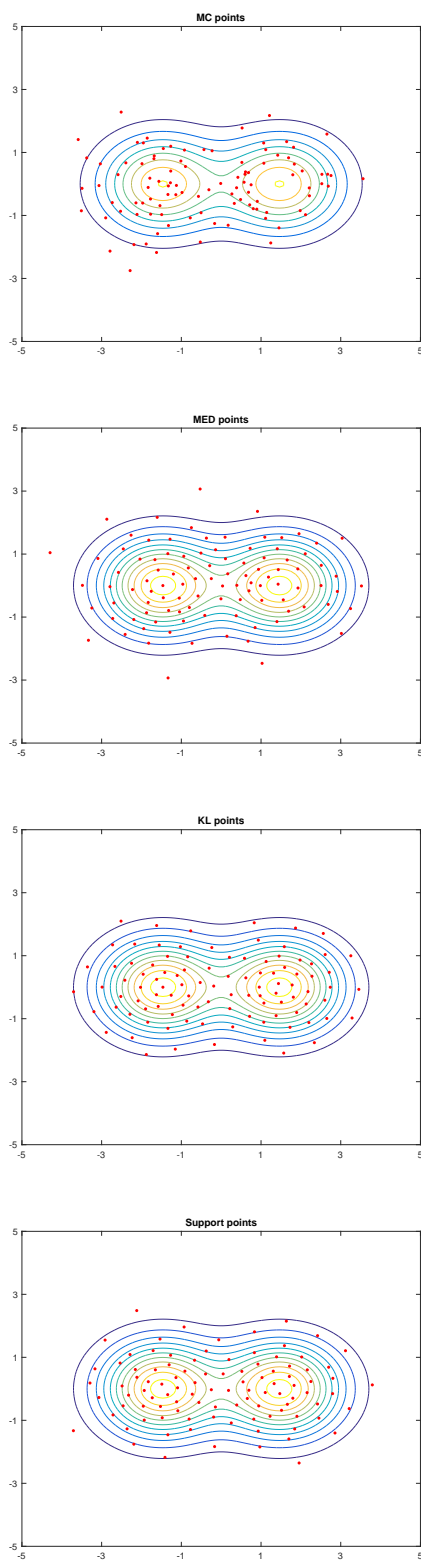


图 2.2 从混合正态分布中获得的100个MC 点, 最小化能量设计点,支撑点以及KL点.

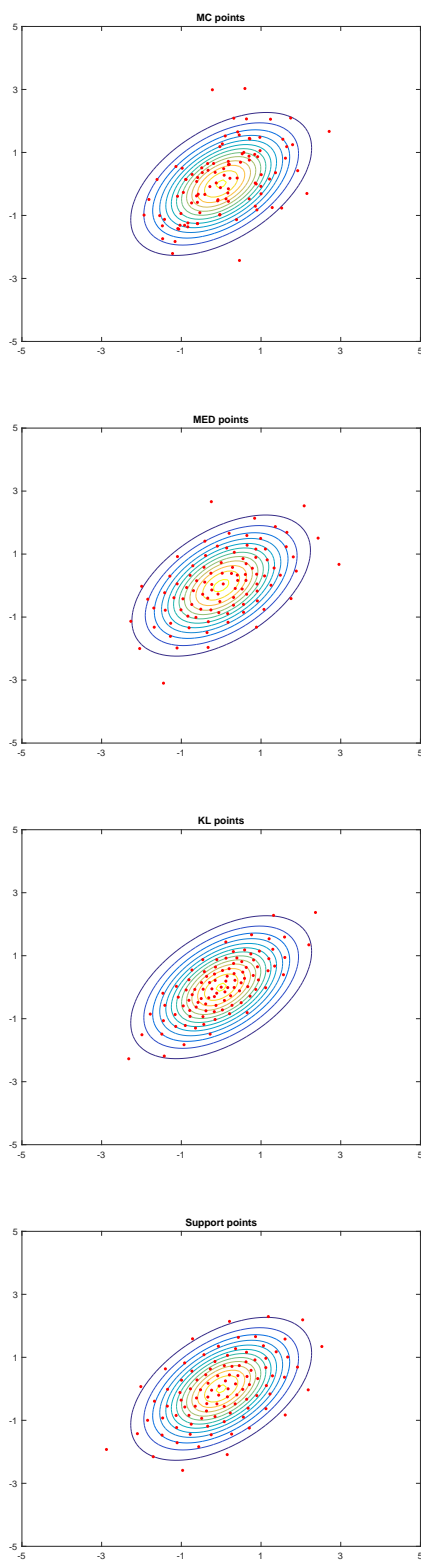


图 2.3 从二维正态分布中获得的100个MC点, 最小化能量设计点,支撑点以及KL点.



表 2.1 GAPK函数在分布 $N(0, \sigma^2 I)$ 下的积分误差的方差, 方差单位取 $10^{-4}$ .

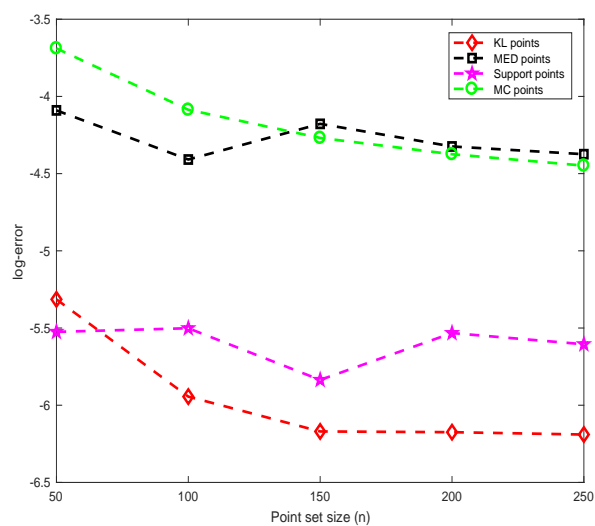
	$N(0, \sigma^2 I)$				
$n$	50	100	150	200	250
MC	3.2738	1.7438	1.1005	0.9316	0.7248
MED	0.9407	0.3558	0.9407	0.2581	0.1989
Support points	0.0415	0.0458	0.0325	0.0488	0.0317
KL points	0.1201	0.0288	0.0177	0.0125	0.0097

## §2.4.2 数值积分

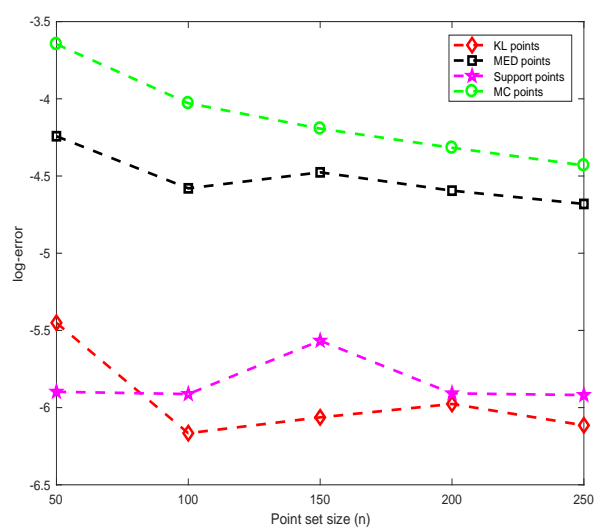
本节介绍了KL点与MC点、最小化能量设计点以及支撑点在数值积分上的表现. 模拟的设置如下, 本节生成的MC点、最小化能量设计点、支撑点和KL点的点集大小 $n$ 分别取为50, 100, 150, 200, 250. 由于算法具有随机性, 因此对最小化能量设计、支撑点和KL点均重复50次进而计算每种抽样方法下的平均误差, 对MC抽样方法重复1000次来计算其平均误差. 本节的模拟实例中采用两种类型的二元正态分布 $F$ , 分别是 $N(0, \sigma^2 I)$ 和 $N(0, \sigma^2 R)$ , 这里 $\sigma = 1$ ,  $I$ 是单位矩阵,  $R_{ij} = 0.5^{|i-j|}$ ,  $i, j = 1, 2$ . 积分函数取作GAPK函数<sup>[61]</sup>:  $g(\mathbf{x}) = \exp\{-\sum_{l=1}^p \alpha_l^2 x_l^2\}$ , 这里 $\mathbf{x} = (x_l)_{l=1}^p$ ,  $p = 2$ , 且尺度参数 $\alpha_l$ 's 取为10.

图2.4给出了GAPK函数在分布函数分别取 $N(0, \sigma^2 I)$  2.4(a)和 $N(0, \sigma^2 R)$ 时不同点集大小下的平均绝对误差. 图2.4表明对于不同的点集大小 $n$ , KL点的平均绝对误差都要比MC点和最小化能量设计点方法下的平均绝对误差小, 并且当 $n$ 大于100时, KL点的平均绝对误差也比支撑点的平均绝对误差小.

表2.1-2.2给出了当点集大小 $n$ 分别取50, 100, 150, 200, 250时最小化能量设计点, 支撑点和KL点分别重复50次, MC点重复1000次时, GAPK函数在分布 $F$ 取 $N(0, \sigma^2 I)$ 和 $N(0, \sigma^2 R)$ 下的积分误差的方差. 从这两个表中可以看出KL点的积分误差的方差要比最小化能量设计点和MC点方法下积分误差的方差都要小, 并且当 $n$ 大于100时, KL点的积分误差的方差也比支撑点的积分误差的方差小.



(a)



(b)

图 2.4 GAPK函数的积分对数绝对误差: (a)  $N(0, \sigma^2 I)$ ; (b)  $N(0, \sigma^2 R)$ .

表 2.2 GAPK函数在分布 $N(0, \sigma^2 R)$ 下的积分误差的方差, 方差单位取 $10^{-4}$ .

	$N(0, \sigma^2 R)$				
$n$	50	100	150	200	250
MC	4.2641	1.7464	1.3165	0.9729	0.7676
MED	0.9277	0.5617	0.2402	0.3397	0.3169
Support points	0.0273	0.0332	0.0428	0.0315	0.0312
KL points	0.1278	0.0249	0.0243	0.0202	0.0107

## §2.5 本章小结

本章基于Kullback-Leibler偏差提出了一类复杂响应曲面好的确定性抽样方法, 得到的代表点集记为KL点. 该方法得到的样本点在响应值高的区域布的点多, 在响应值低的区域布的点少, 并且样本点之间充分地散开. 这类代表点可用于寻找响应曲面 $f(\mathbf{x})$ 的最大值. 如果目标是寻找响应曲面的最小值, 只需将该响应曲面变换为 $-f(\mathbf{x})$ 即可. 本章提出的确定性抽样的核心思想是最小化一组点的核密度估计与目标密度函数的Kullback-Leibler 偏差, 该偏差常用于信息论中度量两个分布之间的相似性. 数值模拟结果表明KL 点是函数一组好的代表点. 并且, KL 点在空间填充性质和数值积分上均要比现有的确定性抽样方法和Monte Carlo抽样方法好. 进一步, 本章还得到了一些理论结果, 证明了KL点作为代表点的合理性.

毋庸置疑, 只要目标分布函数易于计算, MCMC抽样方法是一类简单灵活的抽样方法. KL点的方法相比于MCMC方法的明显优势是KL点利用较少的点提供的信息与较多的MCMC点提供的信息大致相同. 这种优势在贝叶斯贝叶斯推断中至关重要.

## §2.6 理论证明

本节分别给出第二章复杂响应曲面的确定性抽样中引理2.2.1, 定理2.2.1-定理2.2.3的证明.

引理2.2.1 的证明.

为了简化起见, 我们以 $d = 1$ 为例,  $d > 1$ 的证明与之类似. 令 $\{x_j^*\}_{j=1}^n \stackrel{i.i.d.}{\sim} f(x)$ ,  $f_n^*(x) = \frac{1}{nh_n} \sum_{i=1}^n k\left(\frac{x-x_i^*}{h_n}\right)$ 为基于 $\{x_j^*\}_{j=1}^n$ 的核密度估计, 并且核函数 $k(t)$ 与带宽 $h_n$ 满足(K1)-(K5).

由于 $f(x)$ 满足(A1)-(A2), 则 $f(x)$ 在 $X$ 连续. 且由于 $f(x)$ 是一个密度函数, 则存在常数 $f_{\max}$ 使得 $f(x) \leq f_{\max} < \infty$ 成立.  $f_n^*(x)$ 的方差满足:

$$\begin{aligned} \text{var}(f_n^*(x)) &= \text{var}\left(\frac{1}{nh_n} \sum_{i=1}^n k\left(\frac{x-x_i^*}{h_n}\right)\right) \\ &\leq \frac{1}{nh_n^2} E\left[k^2\left(\frac{x-x_1^*}{h_n}\right)\right] \\ &= \frac{1}{nh_n^2} \int_X k^2\left(\frac{z-x}{h_n}\right) f(z) dz \\ &= \frac{1}{nh_n} \int_X k^2(t) f(th_n + x) dt \\ &\leq \frac{f_{\max}}{nh_n} \int_X k^2(t) dt = \frac{C_1}{nh_n}, \end{aligned} \quad (2.10)$$

这里 $C_1 = f_{\max} \|k\|_2^2$ ,  $\|k\|_2^2 = \int_X k^2(t) dt$ .

$f_n^*(x)$ 的偏差有如下的结论.

$$\begin{aligned} E[f_n^*(x)] - f(x) &= \frac{1}{nh_n} \sum_{i=1}^n E\left[k\left(\frac{x-x_i^*}{h_n}\right)\right] - f(x) \\ &= \frac{1}{h_n} \int_X k\left(\frac{z-x}{h_n}\right) f(z) dz - f(x) \\ &= \int_X k(t) f(th_n + x) dt - f(x) \\ &= \int_X k(t) [f(th_n + x) - f(x)] dt. \end{aligned} \quad (2.11)$$

鉴于 $f(x)$ 满足 $\forall x, y \in \mathcal{X}, |f(x) - f(y)| \leq L|x - y|$ . 因此, 可得

$$\begin{aligned}
 |E[f_n^*(x)] - f(x)| &\leq \int_{\mathcal{X}} k(t) |Lth_n| dt \\
 &= Lh_n \int_{\mathcal{X}} |t| k(t) dt \\
 &\leq Lh_n \left\{ \int_{\mathcal{X}} t^2 k(t) dt \right\}^{1/2} \\
 &= C_2 h_n,
 \end{aligned} \tag{2.12}$$

这里 $C_2 = L \left\{ \int_{\mathcal{X}} t^2 k(t) dt \right\}^{1/2}$ .

$f(x)$  与  $f_n^*(x)$  之间的Kullback-Leibler偏差为:

$$D(f_n^* \| f) = \int_{\mathcal{X}} f_n^*(x) \log \frac{f_n^*(x)}{f(x)} dx.$$

根据不等式 $\log x \leq (x - 1)$ , 则 $D(f_n^* \| f)$ 的期望满足:

$$\begin{aligned}
 E[D(f_n^* \| f)] &= E \left[ \int_{\mathcal{X}} f_n^*(x) \log \frac{f_n^*(x)}{f(x)} dx \right] \\
 &\leq E \left[ \int_{\mathcal{X}} f_n^*(x) \left( \frac{f_n^*(x)}{f(x)} - 1 \right) dx \right] \\
 &= E \left[ \int_{\mathcal{X}} f_n^*(x) \frac{f_n^*(x)}{f(x)} dx \right] - 1 \\
 &= \int_{\mathcal{X}} E \left[ f_n^*(x) \frac{f_n^*(x)}{f(x)} \right] dx - 1 \\
 &= \int_{\mathcal{X}} \frac{E[f_n^*(x)]^2 - f^2(x)}{f(x)} dx \\
 &\leq \int_{\mathcal{X}} \frac{|E[f_n^*(x)]^2 - f^2(x)|}{f(x)} dx.
 \end{aligned} \tag{2.13}$$

依据Fubini 定理倒数第二个“=”成立.

根据(2.10) 和(2.12), 有

$$\begin{aligned}
 |E[f_n^*(x)]^2 - f^2(x)| &= |E[f_n^*(x) - f(x) + f(x)]^2 - f^2(x)| \\
 &= |E[f_n^*(x) - f(x)]^2 + 2f(x)E[f_n^*(x) - f(x)]| \\
 &\leq \text{var}(f_n^*(x)) + 2f(x)|E[f_n^*(x)] - f(x)| \\
 &\leq \frac{C_1}{nh_n} + 2C_2h_nf(x) \triangleq G_n(x).
 \end{aligned} \tag{2.14}$$

显然,  $\frac{G_n(x)}{f(x)}$  关于  $n$  单调递减, 并且  $\lim_{n \rightarrow \infty} \frac{G_n(x)}{f(x)} = 0$ . 由单调收敛定理, 下式成立,

$$\lim_{n \rightarrow \infty} \int_X \frac{G_n(x)}{f(x)} dx = \int_X \lim_{n \rightarrow \infty} \frac{G_n(x)}{f(x)} dx = 0. \tag{2.15}$$

注意到

$$\int_X \frac{|E[f_n^*(x)^2] - f^2(x)|}{f(x)} dx \leq \int_X \frac{G_n(x)}{f(x)} dx,$$

因此, 可得

$$\lim_{n \rightarrow \infty} \int_X \frac{|E[f_n^*(x)^2] - f^2(x)|}{f(x)} dx = 0. \tag{2.16}$$

进而结论  $\lim_{n \rightarrow \infty} E[D(f_n^* \| f)] = 0$  成立.  $\square$

为了证明定理2.2.1-定理2.2.2, 首先需要介绍下面的引理.

**引理2.6.1.** 令  $f(x)$  和  $g(x)$  为两个支撑集包含于  $X \subseteq R^d$  的密度函数, 则

(a)

$$V(f, g) \stackrel{\text{def}}{=} \sup_{A \in \mathcal{B}} \left| \int_A (f(x) - g(x)) dx \right| = \frac{1}{2} \int_X |f(x) - g(x)| dx,$$

这里  $\mathcal{B}$  为  $X$  上的 Borel  $\sigma$ -代数.

(b)

$$2V^2(f, g) \leq D(g \| f).$$

证明. 该引理分别参见于 Tsybakov (2009) 中的 Scheffé's 定理和 Pinsker's 不等式.  $\square$

现在给出定理2.2.1 的证明.

**定理2.2.1的证明.**

首先定义随机变量 $\{\mathbf{x}_j^*\}_{j=1}^\infty \stackrel{i.i.d.}{\sim} f(\mathbf{x})$ , 且令 $f_n^*(\mathbf{x})$  表示 $\{\mathbf{x}_j^*\}_{j=1}^n$ 上的核密度估计. 根据引理2.2.1, 有 $\lim_{n \rightarrow \infty} E[D(f_n^* \| f)] = 0$ .

现在考虑KL点 $\{\mathbf{x}_i\}_{i=1}^n$ 上的核密度估计 $f_n(\mathbf{x})$ . 根据KL点的定义,

$$D(f_n \| f) \leq E[D(f_n^* \| f)],$$

因此 $\lim_{n \rightarrow \infty} D(f_n \| f) = 0$ .

再由引理2.6.1 (b)中的Pinsker's 不等式,

$$\frac{1}{2} \left( \int_X |f_n(\mathbf{x}) - f(\mathbf{x})| d\mathbf{x} \right)^2 \leq D(f_n \| f),$$

则结论 $\lim_{n \rightarrow \infty} \int_X |f_n(\mathbf{x}) - f(\mathbf{x})| d\mathbf{x} = 0$  成立.  $\square$

**定理2.2.2 的证明.**

由于 $f_n(\mathbf{x})$ 为KL点上的核密度估计并且满足条件(K1)- (K5), 则根据引理2.6.1 (a), 有

$$\sup_{A \in \mathcal{B}} \left| \int_A [f_n(\mathbf{x}) - f(\mathbf{x})] d\mathbf{x} \right| = \frac{1}{2} \int_X |f_n(\mathbf{x}) - f(\mathbf{x})| d\mathbf{x}. \quad (2.17)$$

再结合定理2.2.1 和(2.17), 可得

$$\lim_{n \rightarrow \infty} \sup_{A \in \mathcal{B}} \left| \int_A [f_n(\mathbf{x}) - f(\mathbf{x})] d\mathbf{x} \right| = 0. \quad (2.18)$$

令 $A = (-\infty, \mathbf{x}] \in \mathcal{B}$ , 则

$$\lim_{n \rightarrow \infty} \sup_{\mathbf{x}} |F_n(\mathbf{x}) - F(\mathbf{x})| = 0.$$

这里 $F_n$  为密度函数 $f_n(\mathbf{x})$  对应的累积分布函数,  $F$ 为密度函数 $f(\mathbf{x})$ 对应的累积分布函数. 因此该定理得证. $\square$

为了证明定理2.2.3, 首先需要以下两个引理.

**引理2.6.2.** 假定核函数 $k(t)$  满足条件(K1)-(K5). 则,

(a)  $\forall \epsilon > 0$ , 存在 $M > 0$ , 使得 $\int_{[-M, M]^d} k(t) dt \geq 1 - \epsilon$ .

(b) 对于上述 $M > 0$ ,  $\exists y_0$ , 使得 $\varphi(x, y_0) \geq 1/3$ , 这里

$$\varphi(x, y) = \int_{\prod_{i=1}^d [x_i - y_i, x_i + y_i]} k(t) dt,$$

$$x = (x_1, \dots, x_d) \in [-M, M]^d, y = (y_1, \dots, y_d).$$

**证明.** 由 $k(t)$  是一个密度函数, 故(a)成立. 注意到, 对任意 $x = (x_1, \dots, x_d) \in [-M, M]^d$ ,  $\lim_{y \rightarrow +\infty} \varphi(x, y) \geq 1/2$ , 因此(b)成立.  $\square$

**引理2.6.3.** 令 $x_1, \dots, x_n$  为 $f(x)$ 的KL点. 则

$$\lim_{n \rightarrow \infty} \sup_{x \in X} \frac{N_x}{n} = 0,$$

这里

$$N_x = \sum_{i=1}^n I\left(\frac{x - x_i}{h_n} \in [-M, M]^d\right),$$

$h_n$ 是生成KL点时的带宽,  $M$ 是引理2.6.2定义的常数.

**证明.** 为简化期间, 以 $d = 1$ 为例.

由于 $\forall \delta > 0$ ,  $\lim_{n \rightarrow \infty} \frac{\delta}{h_n} = +\infty$ , 因此 $\exists n_0$ , 使得当 $n > n_0$ 时,  $\frac{\delta}{h_n} \geq y_0$ , 这里 $y_0$  满足引理2.6.2定义的 $I(|x| \leq M)\varphi(x, y_0) \geq 1/3$ .

根据反证法, 如果引理2.6.3不成立, 则 $\exists x^*$ , 对于 $\forall N > n_0$ ,  $\exists n_k > N$ , 使得 $\frac{N_{x^*}}{n_k} \geq c_0$ , 即

$$\frac{\sum_{i=1}^{n_k} I(|\frac{x^* - x_i}{h_{n_k}}| \leq M)}{n_k} \geq c_0,$$

这里 $c_0$  是非负的常数.



因此, 有

$$\begin{aligned}
 F_{n_k}(x^* + \delta) - F_{n_k}(x^* - \delta) &= \frac{1}{n_k} \sum_{i=1}^{n_k} \int_{x^* - \delta}^{x^* + \delta} \frac{1}{h_{n_k}} k\left(\frac{t - x_i}{h_{n_k}}\right) dt \\
 &= \frac{1}{n_k} \sum_{i=1}^{n_k} \int_{\frac{x^* - x_i}{h_{n_k}} - \frac{\delta}{h_{n_k}}}^{\frac{x^* - x_i}{h_{n_k}} + \frac{\delta}{h_{n_k}}} k(z) dz \\
 &\geq \frac{1}{n_k} \sum_{i=1}^{n_k} I\left(\left|\frac{x^* - x_i}{h_{n_k}}\right| \leq M\right) \int_{\frac{x^* - x_i}{h_{n_k}} - y_0}^{\frac{x^* - x_i}{h_{n_k}} + y_0} k(z) dz \\
 &\geq \frac{\sum_{i=1}^{n_k} I(|\frac{x^* - x_i}{h_{n_k}}| \leq M)}{3n_k} \\
 &\geq \frac{c_0}{3},
 \end{aligned}$$

这里 $F_{n_k}$ 表示基于KL点的核密度估计 $f_{n_k}$ 对应的累积分布函数. 由引理2.6.2, 倒数第二个“ $\geq$ ”成立. 然而, 这与 $F_{n_k}$ 是连续分布函数相矛盾, 故引理2.6.3得证.  $\square$

**定理2.2.3 的证明.**

令 $F_n^{KL}(\mathbf{x})$ 表示KL点 $\mathbf{x}_1, \dots, \mathbf{x}_n$ 的标准经验分布函数.  $F_n(\mathbf{x})$ 表示核密度估计函数 $f_n(\mathbf{x})$ 对应的累积分布函数. 我们首先证明下式成立

$$\lim_{n \rightarrow \infty} \sup_x |F_n(\mathbf{x}) - F_n^{KL}(\mathbf{x})| = 0.$$

为简化起见, 我们以 $d = 1$ 为例,  $\forall x \in \mathcal{X}$ ,

$$\begin{aligned}
 |F_n(x) - F_n^{KL}(x)| &= \frac{1}{n} \sum_{i=1}^n \left| \int_{-\infty}^x \frac{1}{h_n} k\left(\frac{t - x_i}{h_n}\right) dt - I(x_i \leq x) \right| \\
 &= \frac{1}{n} \sum_{i=1}^n \left| \int_{-\infty}^{\frac{x - x_i}{h_n}} k(z) dz - I(x_i \leq x) \right|.
 \end{aligned}$$

如果  $x_i \leq x$ , 则  $\frac{x-x_i}{h_n} \geq 0$ , 且

$$\begin{aligned} \left| \int_{-\infty}^{\frac{x-x_i}{h_n}} k(z)dz - I(x_i \leq x) \right| &= \left| \int_{-\infty}^{\frac{x-x_i}{h_n}} k(z)dz - 1 \right| \\ &= \int_{\frac{x-x_i}{h_n}}^{+\infty} k(z)dz \\ &\leq I\left(0 \leq \frac{x-x_i}{h_n} \leq M\right) \int_{\frac{x-x_i}{h_n}}^M k(z)dz + \frac{\epsilon}{2}, \end{aligned}$$

这里  $M$  和  $\epsilon$  为引理2.6.2 定义的常数, i.e.  $\forall \epsilon > 0$ , 存在  $M > 0$ , 使得  $\int_{-M}^M k(t)dt \geq 1 - \epsilon$ , 以及  $\int_{-\infty}^{-M} k(t)dt = \int_M^{+\infty} k(t)dt < \frac{\epsilon}{2}$ .

如果  $x_i > x$ , 则  $\frac{x-x_i}{h_n} < 0$ , 且

$$\begin{aligned} \left| \int_{-\infty}^{\frac{x-x_i}{h_n}} k(z)dz - I(x_i \leq x) \right| &= \left| \int_{-\infty}^{\frac{x-x_i}{h_n}} k(z)dz - 0 \right| \\ &\leq I\left(-M \leq \frac{x-x_i}{h_n} < 0\right) \int_{-M}^{\frac{x-x_i}{h_n}} k(z)dz + \frac{\epsilon}{2}. \end{aligned}$$

总之, 对于  $\forall x \in \mathcal{X}$ ,  $F_n(x)$  与  $F_n^{KL}(x)$  的绝对误差为:

$$\begin{aligned} 0 \leq |F_n(x) - F_n^{KL}(x)| &\leq \frac{1}{n} \sum_{i=1}^n I\left(\left|\frac{x-x_i}{h_n}\right| \leq M\right) \int_{\left|\frac{x-x_i}{h_n}\right|}^M k(z)dz + \frac{\epsilon}{2} \\ &\leq \frac{1}{2n} \sum_{i=1}^n I\left(\left|\frac{x-x_i}{h_n}\right| \leq M\right) + \frac{\epsilon}{2}. \end{aligned} \quad (2.19)$$

根据引理2.6.3,

$$\lim_{n \rightarrow \infty} \sup_x \frac{1}{n} \sum_{i=1}^n I\left(\left|\frac{x-x_i}{h_n}\right| \leq M\right) = 0.$$

因此,  $\forall \epsilon > 0$ , 存在  $N$ , 使得当  $n \geq N$  时, 对于  $\forall x \in \mathcal{X}$ , 有

$$0 \leq |F_n(x) - F_n^{KL}(x)| \leq \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon.$$

因此,  $\lim_{n \rightarrow \infty} \sup_x |F_n(x) - F_n^{KL}(x)| = 0$  成立.  $d > 1$  的证明类似, 因此

$$\lim_{n \rightarrow \infty} \sup_x |F_n(\mathbf{x}) - F_n^{KL}(\mathbf{x})| = 0. \quad (2.20)$$

由于

$$\sup_x |F_n^{KL}(\mathbf{x}) - F(\mathbf{x})| \leq \sup_x |F_n^{KL}(\mathbf{x}) - F_n(\mathbf{x})| + \sup_x |F_n(\mathbf{x}) - F(\mathbf{x})|,$$

再结合定理2.2.2, 有

$$\lim_{n \rightarrow \infty} \sup_x |F_n^{KL}(\mathbf{x}) - F(\mathbf{x})| = 0.$$



### 第三章 基于Kullback-Leibler散度抽样方法的推广研究

本章基于Kullback-Leibler偏差提出两类推广的Kullback-Leibler点. 一类是在上一章节中提出的KL点的基础上, 为了避免频繁的计算复杂的响应值提出的自适应的Kullback-Leibler(AKL) 点, 即通过序贯的近似复杂的响应函数, 进而自适应地更新KL点. AKL点可被用于复杂密度函数的抽样与探索复杂响应曲面的最优值. 另一个推广是当响应曲面的维数较高时, 基于一组点密度函数的最小邻近距离估计提出了Kullback-Leibler-nn(KL-nn)点. KL-nn点也能抓住响应曲面的趋势特征, 且不同的点之间保持一定的距离. 数值结果表明, KL-nn点有较好的空间填充性质, 且在响应曲面的维数较高时比最小化能量设计和支撑点有较小的积分误差.

#### §3.1 背景介绍

在探索复杂响应曲面时, 若将其视为非标准化的密度函数, 则可以用一组点密度函数的估计与目标函数之间的Kullback-Leibler偏差来度量这组点的代表性. 进而, 通过最小化样本的经验分布与目标分布之间的Kullback-Leibler偏差便可得到一组好的代表点集. 这类代表点集具有在响应值高的区域分布的点数多, 响应值低的区域分布的点数少, 且点与点之间的距离较大等优点. 上一章通过最小化一组点的核密度估计与目标密度之间的Kullback-Leibler偏差得到的好的样本点称为KL点. 这类KL点不仅具有好的空间填充性质, 而且在数值积分上具有较小的误差.

当复杂响应曲面计算耗时或者没有显示表达式时, 直接生成KL点计算成本往往较高. 为此本章通过最小化一组点的核密度估计与目标密度之间的Kullback-Leibler偏差, 提出一类自适应的KL点, 即通过序贯的近似复杂响应曲面进而自适应的生成KL点. 自适应的KL点依然具有具有在响应值高的区域分布的点数多, 响应值低的区域分布的点数少, 且点与点之间的距离较大等优点. 因此在探索复杂响应曲面时自适应的KL点比传统的空间填充设计更高效. 值得注意的是Joseph et al. (2015)提出的序贯的最下化能量设计(SMED)也是一类好的代表点集, 与之相

比数值模拟表明, AKL点无论是在复杂密度函数的抽样上还是在探索复杂黑盒子函数的最优值上均具有明显的优势.

当复杂响应曲面的维数较高时, 传统的抽样方法是空间填充设计和QMC, 即在空间内均匀地布点. 因此这类抽样方法可能会使许多样本点落入低响应曲面的区域, 进而造成样本的浪费. 本章通过最小化一组样本点密度的最小邻近距离估计与目标函数的Kullback-Leibler偏差, 得到另一类好的代表点记为KL-nn点. 近年来Mak and Joseph (2018)提出的支撑点和Joseph et al. (2019)提出的最小化能量设计也是两类应用于高维响应曲面的代表点集. 数值模拟表明, KL-nn点具有较好的空间填充性质, 且在高维积分上, 比支撑点和最小化能量设计具有较小的误差.

### §3.2 自适应的KL(AKL)点

当 $f(\mathbf{x})$ 没有显示表达或者计算耗时时, 直接应用第三章中随机Nelder-Mead方法最小化一组点的核密度估计与目标密度之间的Kullback-Leibler偏差来生成KL点并不容易. 根据序贯最小化能量设计(SMED)的思想, 一个可行的方法是利用一个简单容易计算的拟模型 $\hat{f}(\mathbf{x})$ 来代替 $f(\mathbf{x})$ . 本节采用自适应的策略, 核心思想是序贯地近似 $f(\mathbf{x})$ , 进而依次产生KL点. 本节依然采用一次产生一个点的贪婪算法. 首先生成一个极大极小距离的拉丁超立方体设计 $D_m = \{\mathbf{x}_1, \dots, \mathbf{x}_m\}$ 作为初始点集, 然后求出响应值向量 $\mathbf{y}^m = (y_1, \dots, y_m)'$ , 这里 $y_i = f(\mathbf{x}_i)$ . 最后利用某些统计模型例如kriging模型来近似 $f(\mathbf{x})$ , 拟模型记为 $\hat{f}^{(m)}(\mathbf{x})$ . 基于初始的集合 $D_m$ , 可以序贯地生成接下来的KL点. 在 $l = m + 1, \dots, n$ 时, 通过 $\{\mathbf{x}_i, y_i\}_{i=1}^{l-1}$ 更新拟模型 $\hat{f}^{(l-1)}(\mathbf{x})$ . 那么, 第 $l$ 个点 $\mathbf{x}_l$ 可通过下式生成

$$\begin{aligned}\mathbf{x}_l &= \arg \min_{\mathbf{x} \in \mathcal{X}} \hat{D}(f_{||} \hat{f}^{(l-1)}) \\ &= \arg \min_{\mathbf{x} \in \mathcal{X}} \log \frac{(lh_l^d)^{-1} \sum_{j=1}^{l-1} k(\frac{\mathbf{x}-\mathbf{x}_j}{h_l})}{\hat{f}^{(l-1)}(\mathbf{x})} \\ &= \arg \min_{\mathbf{x} \in \mathcal{X}} M'(\mathbf{x} | \{\mathbf{x}_j, y_j\}_{j=1}^{l-1}),\end{aligned}$$

这里 $\hat{D}(f_{||} \hat{f}^{(l-1)})$ 的定义同(2.8)式,  $M'(\mathbf{x} | \{\mathbf{x}_j, y_j\}_{j=1}^{l-1}) = \log \frac{(lh_l^d)^{-1} \sum_{j=1}^{l-1} k(\frac{\mathbf{x}-\mathbf{x}_j}{h_l})}{\hat{f}^{(l-1)}(\mathbf{x})}$ .

当核函数 $k(\mathbf{t})$ 依然选为多元指数阶分布时, 带宽 $h_l$  选为

$$h_l = \frac{\min_{i \neq j, i, j \in \{1, \dots, (l-1)\}} \|\mathbf{x}_i - \mathbf{x}_j\|_2}{l^{\frac{5}{d+4}}}, d \geq 2.$$

利用上述算法得到的KL点称为自适应的KL点(AKL points).

算法3.1 给出了基于 $D_m$ 生成接下来 $n - m$ 个AKL点的详细步骤. 与算法2.1 类似, 算法3.1的输入为:  $n - m$ 是生成AKL点的个数, 参数 $n_{\text{init}}, n_{\text{delay}}, \mu_0, \Sigma_0, \{\mathbf{x}_j^{\text{curr}}\}_{j=1}^n, \lambda, l, u$ 的含义同算法2.1.

---

**算法 3.1** 通过随机Nelder-Mead方法生成 $n - m$  个AKL点

---

**输入:**  $n, n_{\text{init}}, n_{\text{delay}}, \mu_0, \Sigma_0, \lambda, l, u$

**输出:**  $\mathbf{x}_{m+1}, \dots, \mathbf{x}_n$

```

1: for  $l = (m + 1), \dots, n$  do
2:   基于 $\{\mathbf{x}_i, y_i\}_{i=1}^{l-1}$ 估计 $\hat{f}^{(l-1)}(\mathbf{x})$ , 并且计算 $M'(\mathbf{x}|\{\mathbf{x}_j, y_j\}_{j=1}^{l-1})$ .
3:   for  $i = 1, \dots, n_{\text{init}}$  do
4:     if  $l \leq n_{\text{delay}}$  then
5:        $\mathbf{x}_i^{\text{init}} \leftarrow \text{trunc}_l^u[\mathcal{N}(\mu_0, \Sigma_0)]$ 
6:     else
7:        $\mathbf{x}_i^{\text{init}} \leftarrow \text{trunc}_l^u[\Pi(\{\mathbf{x}_j\}_{j=1}^{n_{\text{curr}}}, \lambda)]$ 
8:     end if
9:      $\mathbf{x}_i^{\text{local}} \leftarrow \text{NelderMead}_x[M'(\mathbf{x}|\{\mathbf{x}_j, y_j\}_{j=1}^{l-1}), \mathbf{x}_i^{\text{init}}, l, u]$ 
10:   end for
11:    $i^* \leftarrow \text{argmin}_{i \in \{1 \dots n_{\text{init}}\}} M'(\mathbf{x}|\{\mathbf{x}_j, y_j\}_{j=1}^{l-1})$ 
12:    $\mathbf{x}_l \leftarrow \mathbf{x}_{i^*}^{\text{local}}$ 
13: end for

```

---

接下来介绍自适应KL点的两个重要的应用.

### §3.2.1 复杂密度函数的AKL点

当概率密度函数比较复杂且计算耗时, 不能直接通过最小化Kullback-Leibler偏

差的方法来生成KL. 本节利用一个高斯过程模型(GP)来近似复杂的真实密度函数. 由于密度函数均是非负的, 因此需要对密度函数作对数变换, 如下所示:

$$\log f(\mathbf{x}) \sim GP(\mu, \sigma^2 R(\cdot)),$$

这里相关函数 $\text{cor}(\log f(\mathbf{x}_i), \log f(\mathbf{x}_j)) = R(\mathbf{x}_i - \mathbf{x}_j)$ . 如果初始的设计是一个包含 $n_0$ 个点的极大极小距离的拉丁超立方设计<sup>[35]</sup>  $D_{n_0} = \{\mathbf{x}_1, \dots, \mathbf{x}_{n_0}\}$ , 且 $\mathbf{y}^{n_0} = (y_1, \dots, y_{n_0})'$ , 其中 $y_i = f(\mathbf{x}_i)$ , 则利用平稳高斯过程模型<sup>[62]</sup>得到的拟模型为:

$$\hat{f}^{(n_0)}(\mathbf{x}) = \exp\{\hat{\mu}^{(n_0)} + \mathbf{r}^{(n_0)}(\mathbf{x})' R_{(n_0)}^{-1} (\mathbf{y}^{(n_0)} - \hat{\mu}^{(n_0)} \mathbf{1}_{n_0})\},$$

这里 $\mathbf{r}^{(n_0)}(\mathbf{x})$ 是一个长度为 $n_0$ 的向量, 其第 $i$ 个元素为 $R(\mathbf{x} - \mathbf{x}_i)$ ,  $R_{n_0}$ 是 $n_0 \times n_0$ 的相关矩阵, 其第 $(i, j)$ 个元素为 $R(\mathbf{x}_i - \mathbf{x}_j)$ ,  $\mathbf{1}_{n_0}$ 向量长度为 $n_0$ 的全1向量, 且

$$\hat{\mu}^{(n_0)} = (\mathbf{1}_{n_0}' R_{(n_0)}^{-1} \mathbf{1}_{n_0})^{-1} \mathbf{1}_{n_0}' R_{(n_0)}^{-1} \mathbf{y}^{(n_0)}.$$

这里的相关函数结构选择为高斯相关

$$R(\mathbf{t}) = \exp\left\{-\sum_{i=1}^d \theta_i t_i^2\right\}.$$

利用算法3.1可以得到密度函数 $f(\mathbf{x})$ 的 $n - n_0$ 个自适应的KL点.

例如考虑二维香蕉形状的概率密度函数. 利用算法3.1生成 $n=50$ 个点, 其中前 $n_0 = 20$ 个点是一个极大极小距离的拉丁超立方体设计. 剩下的30个AKL点通过算法3.1得到, 在图3.1中标记为\*. 为了比较, 本小节根据相同的初始设计生成30个序贯的最小化能量设计点(SMED), 标记为图3.1中的\*. 从图中可以看出, 显然自适应的KL点比序贯的最小化能量设计点更具有代表性. 进一步, 发现序贯的最小化能量设计点对初始点集特别敏感, 序贯的最小化能量设计需要剔除初始点集中密度值较小的点, 否则序贯产生的点就会粘连在一起, 如图3.1所示.



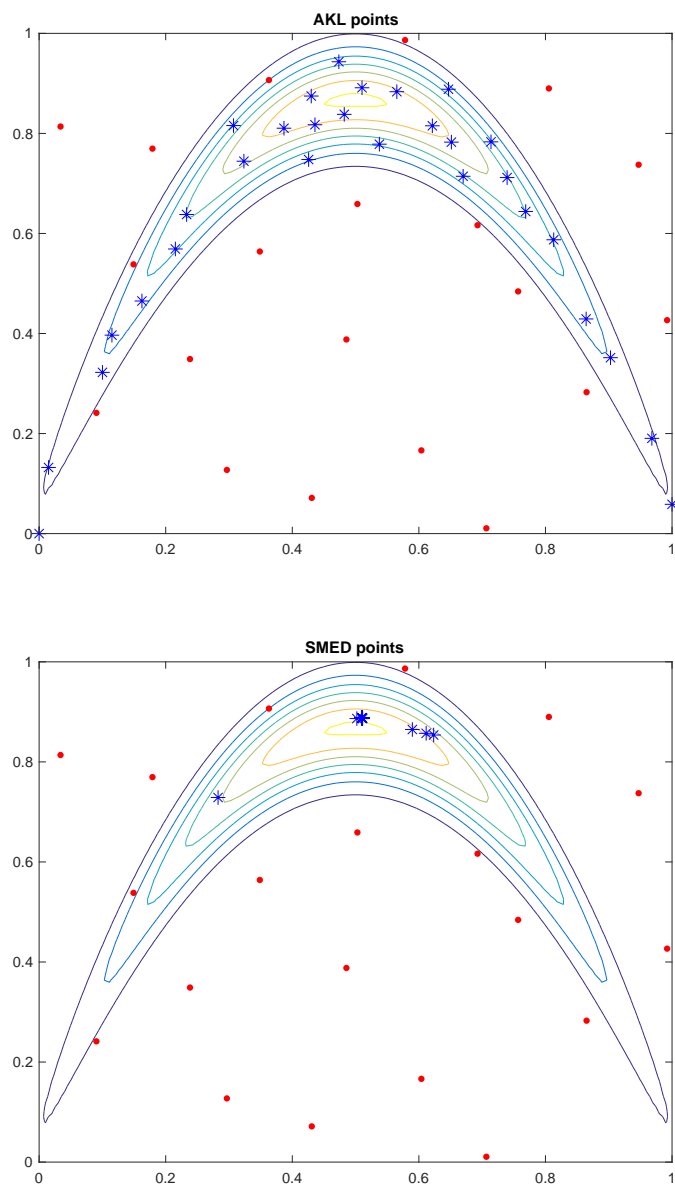


图 3.1 自适应的KL点(AKL)与序贯的最小化能量设计点(SMED)的比较. 初始设计是一个极大极小拉丁超立方体设计标记为●.

### §3.2.2 探索和优化昂贵的黑盒子函数

全局最优化是一个重要而又困难的问题. 许多算法可以用来搜索全局最优解, 如模拟退火算法和遗传算法等. 然而, 这些算法需要对函数进行大量的计算, 如果目标函数的计算成本很高, 那么算法的成本就会很高.

不失一般性, 本节考虑的优化问题是在某个有界区域 $\mathcal{X}$ 中寻找使得 $f(\mathbf{x})$ 最大的 $\mathbf{x}^*$ . 如果目标是寻找 $f(\mathbf{x})$ 的最小值, 则可以用 $-f(\mathbf{x})$ 来代替 $f(\mathbf{x})$ . 在本小节中, 依然不能直接生成 $f(\mathbf{x})$ 的KL点, 因为 $f(\mathbf{x})$ 的函数值可能是负数. 因此, 需要定义一个辅助函数:

$$g(\mathbf{x}) = f(\mathbf{x}) - f_{\min},$$

这里 $f_{\min} = \min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$ . 显然辅助函数 $g(\mathbf{x}) \geq 0$ 并且与 $f(\mathbf{x})$ 有相同的函数走势与最优值. 进而, 可以利用算法3.1生成 $g(\mathbf{x})$ 的自适应的KL点. 然而, 对于昂贵的函数 $f(\mathbf{x})$ 很难得到 $f_{\min}$ . 因此, 与上一节类似, 依然利用一个容易计算的拟模型序贯的近似函数 $f(\mathbf{x})$ .

本节仍然选择平稳的高斯过程模型近似一个黑盒子函数:  $f(\mathbf{x}) \sim GP(\mu, \sigma^2 R(\cdot))$ . 跟上一节类似, 基于一个包含 $n_0$ 个点的极大极小距离的拉丁超立方体设计, 得到的拟模型如下:

$$\hat{f}^{(n_0)}(\mathbf{x}) = \hat{\mu}^{(n_0)} + r^{(n_0)}(\mathbf{x})' R_{(n_0)}^{-1}(\mathbf{y}^{(n_0)} - \hat{\mu}^{(n_0)} \mathbf{1}_{n_0}).$$

通过最小化 $\hat{f}^{(n_0)}(\mathbf{x})$ 可以得到 $f_{\min}$ 的近似值 $\hat{f}_{\min}^{(n_0)}$ . 进一步, 可以得到 $\hat{g}^{(n_0)}(\mathbf{x}) = \hat{f}^{(n_0)}(\mathbf{x}) - \hat{f}_{\min}^{(n_0)}$ . 进而, 利用算法3.1 可以生成 $f(\mathbf{x})$ 的自适应的KL点. 在每一个 $n = n_0 + 1, \dots$ , 通过更新 $\hat{f}^n(\mathbf{x})$ 和 $\hat{f}_{\min}^{(n)}$ 可以得到新的 $\hat{g}^{(n)}(\mathbf{x})$ .

例如, 本小节考虑函数<sup>[63]</sup>

$$\begin{aligned} f(\mathbf{x}) = & \frac{3}{4} \exp\{-\frac{1}{4}(9x_1 - 2)^2 - \frac{1}{4}(9x_2 - 2)^2\} + \frac{3}{4} \exp\{-\frac{1}{49}(9x_1 + 1)^2 - \frac{1}{10}(9x_2 + 1)^2\} \\ & + \frac{1}{2} \exp\{-\frac{1}{4}(9x_1 - 7)^2 - \frac{1}{4}(9x_2 - 3)^2\} - \frac{1}{5} \exp\{-(9x_1 - 4)^2 - (9x_2 - 2)^2\}. \end{aligned}$$

初始设计依然选择一个包含 $n_0 = 20$ 个点的极大极小距离的拉丁超立方体设计. 对于辅助函数 $g(\mathbf{x})$ , 应用算法3.1可以生成接下来的20个自适应的KL点, 如图3.2所示. 该黑盒子函数 $f(\mathbf{x})$ 的全局最优解是 $\mathbf{x}^* = (0.21, 0.17)$ , 在图3.2-3.3 中标记为\*. 从

图3.2 中可以看出, 自适应的KL点找到的全局最优解 $\mathbf{x} = (0.19, 0.17)$ 与 $\mathbf{x}^*$ 的距离较近.

为了比较, 本小节利用相同的初始点集生成接下来的20个序贯的最小化能量设计点, 如图3.3所示. 从图中可以看出, 序贯的最小化能量设计点找到的全局最优解与 $\mathbf{x}^*$  的距离较大.

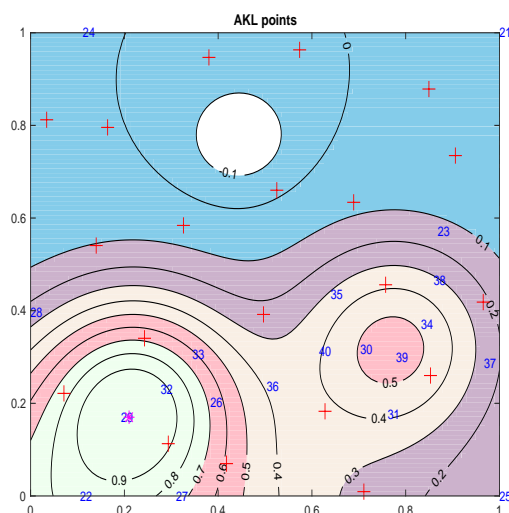


图 3.2 自适应的KL(AKL)点: + 表示初始的点集( $n_0 = 20$ ), 接下来的20个点按顺序出现.

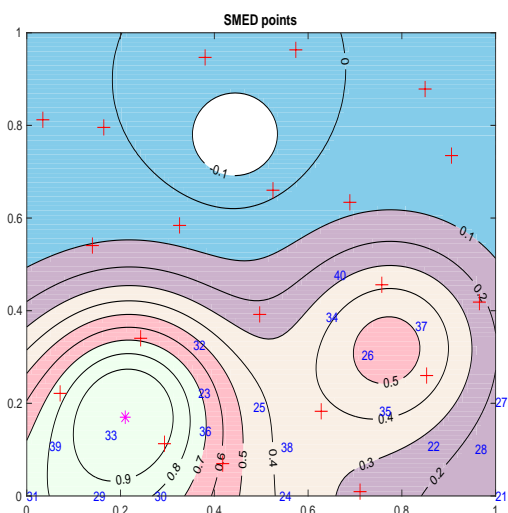


图 3.3 序贯的最小化能量设计(SMED)点: + 表示初始的点集( $n_0 = 20$ ), 接下来的20个点按顺序出现.

考虑到初始设计的选择可能会影响自适应的KL点和序贯的最小化能量设计点寻找全局最优解的能力, 因此, 本小节利用50个随机生成的拉丁超立方体设计作为初始设计来比较自适应的KL点和序贯的最小化能量设计点寻找全局最优解的能力. 图3.4比较了当两种方法进行50次寻找全局最优解时, 每次自适应的KL点和序贯的最小化能量设计点分别找到的全局最大值与真实的全局最大值的比较. 显然, 自适应的KL点的方法找到的全局最大值更接近于真实的全局最大值.

图3.5显示了当两种方法进行50次寻找全局最优解时, 自适应的KL点和序贯的最小化能量设计点找到的全局最优解与 $\mathbf{x}^*$ 之间的欧氏距离的箱线图. 正如预期的那样, 自适应的KL点的方法在寻找全局最优解方面表现更好.

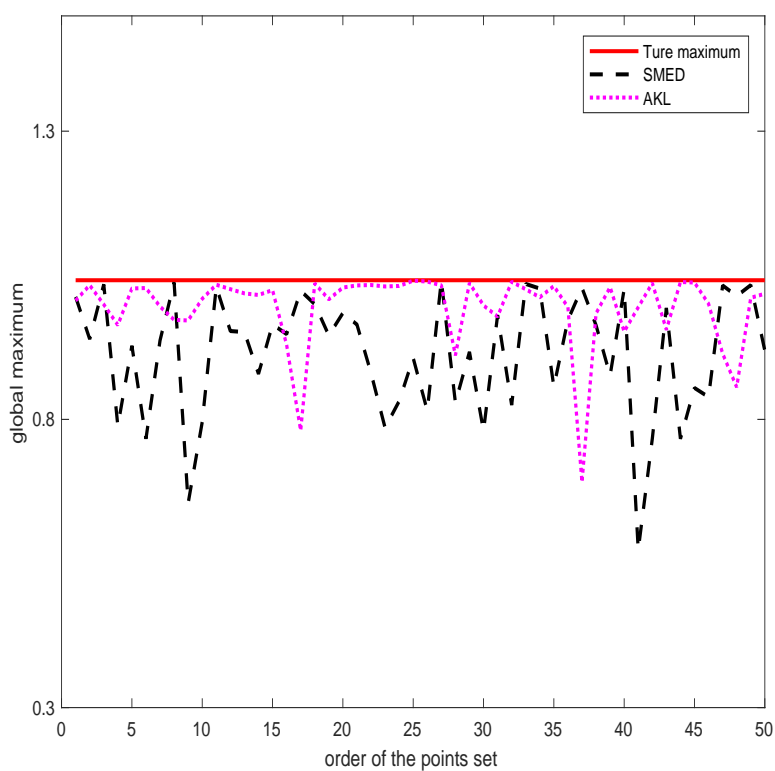


图 3.4 当两种方法进行50次寻找全局最优解时, 真实的全局最大值与AKL点和SMED方法找到的全局最大值的比较, 顶部实线表示真实的全局最大值.

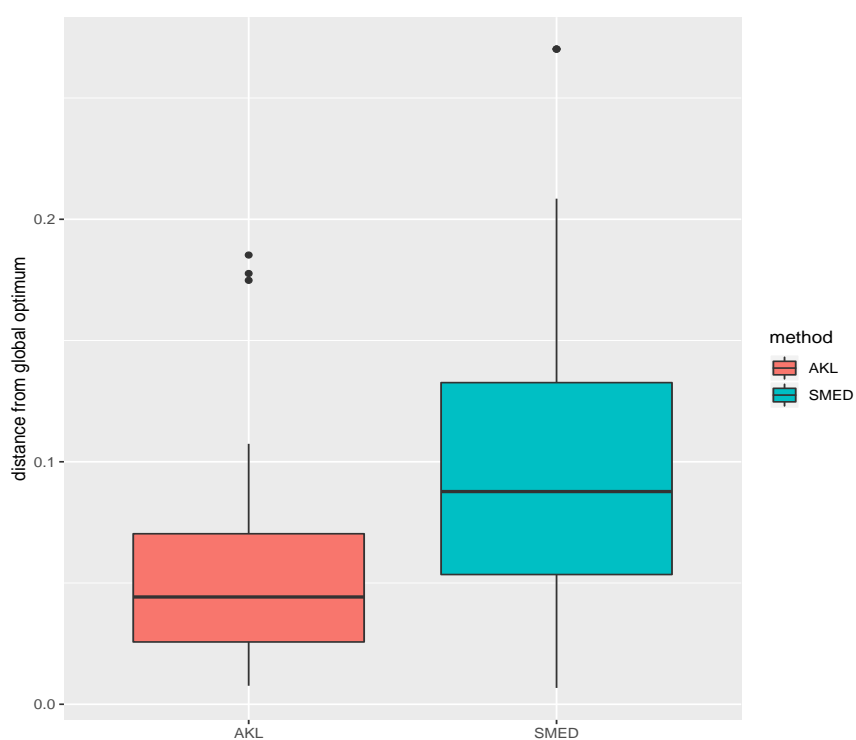


图 3.5 当两种方法进行50次寻找全局最优解时, AKL点和SMED两种方法找到的全局最优解与 $\mathbf{x}^*$ 之间的欧氏距离的箱线图.

### §3.3 基于最小邻近距离的Kullback-Leibler-nn点

为了提高KL点在高维响应曲面的表现, 依然可以将其转化为非标准化的密度函数. 本节通过最小化一组点密度的最小邻近估计与目标密度函数的Kullback-Leibler偏差来生成另一类推广的KL点, 并记为KL-nn点.

#### §3.3.1 Kullback-Leibler-nn点的定义

为了从目标密度函数中获取一组好的代表点, 本节利用一组点密度函数的最小邻近距离估计与目标密度函数之间的Kullback-Leibler偏差来度量一组点的代表性. 令

$$f_n^{Neib}(\mathbf{x}) = \frac{1}{nV_1(d)\rho_1^d(\mathbf{x}_i)} I(\mathbf{x} \in S(\mathbf{x}_i, \rho_1(\mathbf{x}_i))), \quad (3.1)$$

表示基于一组点 $\{\mathbf{x}\}_{i=1}^n$ 的最小邻近距离的密度估计, 这里 $V_1(d) = \frac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)}$ 表示 $R^d$ 上单位球的体积,  $\rho_1(\mathbf{x}_i) = \min_{j \neq i} \|\mathbf{x}_i - \mathbf{x}_j\|_2$ ,  $S(\mathbf{x}_i, \rho_1(\mathbf{x}_i))$ 表示以 $\mathbf{x}_i$ 为中心,  $\rho_1(\mathbf{x}_i)$ 为半径的圆, 并且 $S(\mathbf{x}_i, \rho_1(\mathbf{x}_i))$ 的体积表示为 $V_1(d)\rho_1^d(\mathbf{x}_i)$ .

基于(3.1)定义的 $f_n(\mathbf{x})$ , Kosachenko and Leonenko (1987)给出了 $f_n(\mathbf{x})$ 与 $f(\mathbf{x})$ 之间Kullback-Leibler偏差的无偏估计, 如下式所示:

$$D(f_n^{Neib} \| f) = - \left( \frac{d}{n} \sum_{i=1}^n \log \rho_1(\mathbf{x}_i) + \frac{1}{n} \sum_{i=1}^n \log f(\mathbf{x}_i) + \log V_1(d) + \log n + \gamma \right), \quad (3.2)$$

这里 $\gamma$ 是欧拉常数. 本小节通过最小化 $D(f_n \| f)$ 来得到Kullback-Leibler点, 并记为KL-nn点.

**定义3.3.1.** (KL-nn点) 对于固定的点集大小 $n \in \mathbb{N}$ , 函数 $f(\mathbf{x})$ 的Kullback-Leibler-nn点的定义如下:

$$\begin{aligned} \{\xi_i\}_{i=1}^n &= \arg \min_{\mathbf{x}_1, \dots, \mathbf{x}_n} D(f_n^{Neib} \| f) \\ &= \arg \max_{\mathbf{x}_1, \dots, \mathbf{x}_n} \left( \frac{d}{n} \sum_{i=1}^n \log \rho_1(\mathbf{x}_i) + \frac{1}{n} \sum_{i=1}^n \log f(\mathbf{x}_i) \right). \end{aligned} \quad (3.3)$$

上述KL-nn点的定义表明, 函数 $f(\mathbf{x})$ 的KL-nn点具有在响应值高的区域分布的

点多, 在响应值低的区域分布的点少, 并且点与点之间的距离不会很近等优点.

### §3.3.2 序贯地生成Kullback-Leibler-nn 点的算法

本节依然采用一次生成一个点的序贯策略生成 $n$ 个KL-nn点. 假定已经生成了 $m-1$ 个点, 则第 $m$ 个点可以通过下面的式子得到:

$$\begin{aligned} \mathbf{x}_m &= \arg \min_{\mathbf{x}} D(f_n^{Neib} \| f) \\ &= \arg \min_{\mathbf{x}} \frac{1}{m} \left\{ \sum_{i=1}^{m-1} \log \frac{1}{\rho_1^d(\mathbf{x}_i) f(\mathbf{x}_i)} + \log \frac{1}{\rho_1^d(\mathbf{x}) f(\mathbf{x})} \right\} \\ &= \arg \max_{\mathbf{x}} \frac{d}{m} \sum_{i=1}^{m-1} \log \rho_1(\mathbf{x}_i) + \frac{d}{m} \log \rho_1^d(\mathbf{x}) + \frac{1}{m} \log f(\mathbf{x}) \end{aligned} \quad (3.4)$$

$$= \arg \max_{\mathbf{x}} \hat{M}(\mathbf{x} | \mathbf{x}_1, \dots, \mathbf{x}_{m-1}), \quad (3.5)$$

这里

$$\hat{M}(\mathbf{x} | \mathbf{x}_1, \dots, \mathbf{x}_{m-1}) = \sum_{i=1}^{m-1} \log \rho_1(\mathbf{x}_i) + \frac{d}{m} \log \rho_1^d(\mathbf{x}) + \frac{1}{m} \log f(\mathbf{x}),$$

$$\rho_1(\mathbf{x}_i) = \min_{j \neq i} \{\min \|\mathbf{x}_j - \mathbf{x}_i\|_2, \|\mathbf{x}_i - \mathbf{x}\|_2\}, \rho_1(\mathbf{x}) = \min_{i=1:(n-1)} \|\mathbf{x}_i - \mathbf{x}\|_2.$$

特别地, 第一个点 $\mathbf{x}_1 = \arg \max_{\mathbf{x}} f(\mathbf{x})$ .

序贯生成每一个KL-nn点均需要在 $\mathbf{x} \in \mathcal{X}$ 上做全局优化. 算法2.1-3.1采用的是随机化的Nelder-Mead方法. 由于在生成KL-nn时, 需要计算 $\min_{j \neq i} \|\mathbf{x}_j - \mathbf{x}_i\|_2$ , 因此利用随机化的Nelder-Mead方法全局搜索最优值时, 随着维数的增加, 算法的时间也会加长. 为此, 本节通过基于MCMC的局部搜索方法代替随机化Nelder-Mead方法来生成KL-nn点, 具体见算法3.2.

### §3.3.3 数值模拟

从KL-nn点的定义可知, KL-nn点同样可视为从分布 $F$ 上得到一组最优样本点. 这些点集中在密度值高的区域, 同时点与点之间地距离尽量的隔开. 这样的空间填充特性可以减小数值积分的误差. 为了直观体现KL-nn点的空间填充性, 图3.6给出了最小化能量设计, 支撑点设计, KL点以及KL-nn点从2维正态分布函数 $N(0, \sigma^2 R)$ 中抽取 $n = 100$ 个样本点的散点图, 这里 $\sigma = 1, R_{ij} = 0.5^{|i-j|}, i, j = 1, 2$ .

---

**算法 3.2** 通过局部MCMC生成 $n$  KL-nn点

---

输入:  $n, \mathbf{x}_1$

输出:  $\mathbf{x}_2, \dots, \mathbf{x}_n$

```

1: for  $m = 2, \dots, n$  do
2:   根据 $\{\mathbf{x}_j\}_{j=1}^{m-1}$ , 计算 $\hat{M}(\mathbf{x}|\mathbf{x}_1, \dots, \mathbf{x}_{m-1})$ .
3:   从 $\{1, \dots, m-1\}$ 中随机的选择一个指标 $s^*$ ;
4:   以 $\mathbf{x}_{s^*}$ 为初始点, 生成一个长度为 $m_j$ 的马氏链 $\{\mathbf{y}_{m,l}\}_{l=1}^{m_j}$ ;
5:    $l^* \leftarrow \operatorname{argmax}_{l \in \{1 \dots m_j\}} \hat{M}(\mathbf{y}_{m,l}|\mathbf{x}_1, \dots, \mathbf{x}_{m-1})$ 
6:    $\mathbf{x}_m \leftarrow \mathbf{x}_{l^*}$ 
7: end for
    
```

---

从图3.6 中可以看出, KL-nn点的空间填充性质虽然没有KL点好, 但是其空间填充性质与最小化能量设计以及支撑点设计相类似.

接下来, 本节比较一下KL-nn点与KL点、最小化能量设计以及支撑点设计在高维数值积分上的表现. 模拟的设置如下, KL-nn点、KL点、最小化能量设计以及支撑点设计的点集大小 $n$ 分别取50, 100, 150, 200, 250. 由于生成KL-nn点、KL点、最小化能量设计以及支撑点设计时算法的随机性, 因此针对不同的点集大小 $n$ , 对这4种抽样方法均重复10次进而计算每种抽样方法下的平均误差. 本节的模拟实例采用的分布函数是 $d = 10$ 的多元正态分布 $N(0, \sigma^2 I)$ , 这里 $\sigma = 1$ ,  $I$ 是单位矩阵. 积分函数依然取作GAPK函数<sup>[61]</sup>:  $g(\mathbf{x}) = \exp\{-\sum_{l=1}^d \alpha_l^2 x_l^2\}$ . 这里尺度参数 $\alpha_l$ 均取为2.

图3.7给出了点集大小 $n$ 分别取50, 100, 150, 200, 250, KL-nn点、KL点、最小化能量设计以及支撑点设计重复10次时, GAPK函数在分布函数 $N(0, \sigma^2 I)$ 下的绝对误差. 图3.7表明对于不同的点集大小 $n$ , KL-nn点的积分误差都要比其他3种方法下的误差小.

表3.1给出了点集大小 $n$ 分别取50, 100, 150, 200, 250, KL-nn点、KL点、最小化能量设计以及支撑点设计重复10次时, GAPK函数在分布 $N(0, \sigma^2 I)$ 下的积分误差的方差. 从这个表中可以看出KL-nn点误差的方差要比最小化能量设计和支撑点设计方法下的方差小, 但是没有KL点方法下误差的方差小.



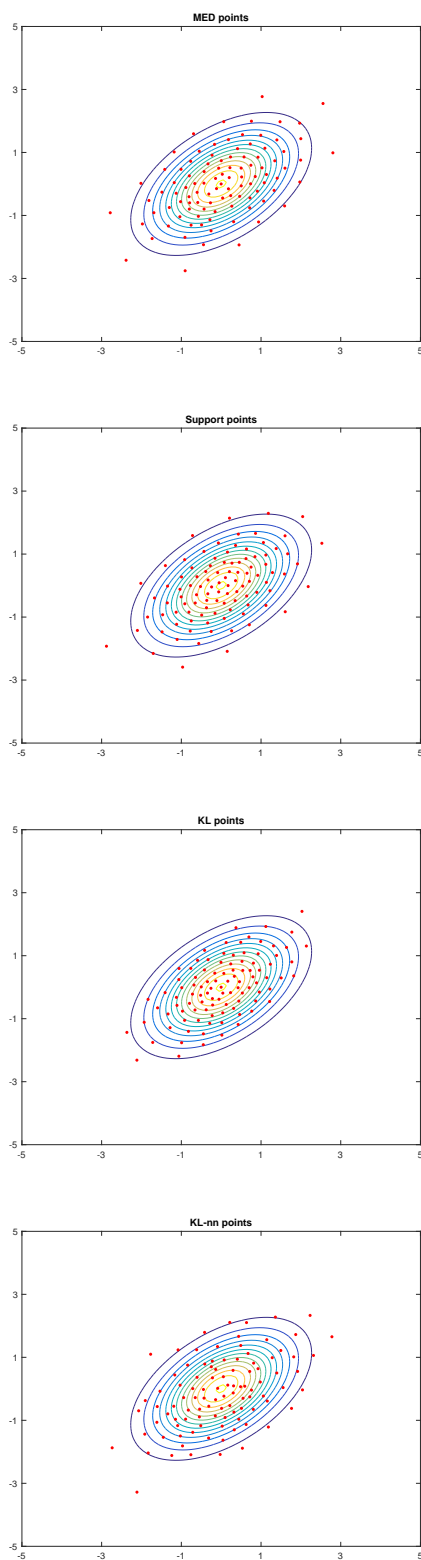


图 3.6 从正态分布中获得的 $n = 100$ 的MED点, Support points点, KL点以及KL-nn点

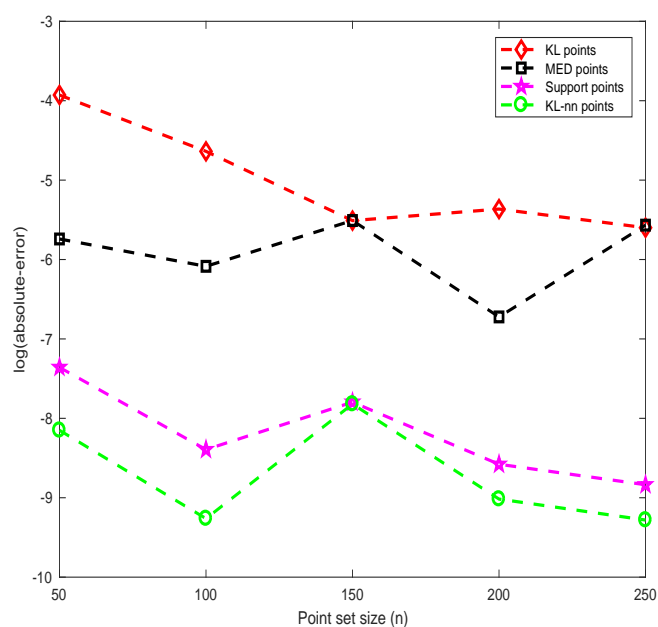


图 3.7 GAPK函数在分布函数取 $N(0, \sigma^2 I)$ 时的对数绝对误差.

表 3.1 GAPK函数在分布 $N(0, \sigma^2 I)$ 下的积分误差的方差, 方差单位取 $10^{-6}$ .

	$N(0, \sigma^2 I)$				
$n$	50	100	150	200	250
MED	2.58	1.41	0.88	0.12	0.19
KL points	0.003	0.008	0.003	0.004	0.002
Support points	0.96	0.034	0.53	0.023	0.029
KL-nn points	0.049	0.006	0.177	0.008	0.005

### §3.4 本章小结

本章基于Kullback-Leibler偏差导出了两类推广的Kullback-Leibler点. 首先, 当响应曲面计算耗时或没有显示表达时提出了自适应的Kullback-Leibler(AKL)点, AKL点在探索复杂响应曲面时也是一类比空间填充设计更高效的抽样方法. AK-L点可用于复杂密度函数的抽样与探索复杂响应曲面的最优值. 其次, 当响应曲面的维数较高时, 基于一组点密度函数的最小邻近距离估计提出了Kullback-Leibler-nn(KL-nn)点. 数值结果表明, KL-nn点不仅具有较好的空间填充性质, 而且在响应曲面的维数较高时具有较小的积分误差.



## 第四章 同时包含定量和定性因子的复杂响应曲面的试验设计

当复杂响应曲面同时包含定性和定量因子时, 传统的抽样设计方法不可用. 为处理该问题, Qian (2012)提出了一种新的设计方法, 称为切片拉丁超立方体设计. He and Qian (2016)研究了切片拉丁超立方体设计的大样本性质, 从理论上说明了该设计空间填充性的合理性. 当定性因子的个数和水平数增加时, 切片拉丁超立方体设计的试验次数呈指数增长, 因此实际的试验成本难以承受. 为这个问题, Deng et al. (2015), He et al. (2017)提出了边际耦合设计方法, 该设计可以大大降低试验次数, 从而节约试验成本. 由于边际耦合设计的结构与切片拉丁超立方体设计的结构不同, 所以He and Qian (2016)获得的大样本性质不能直接应用到边际耦合设计中. 为此, 本章导出了边际耦合设计的若干大样本性质, 进一步导出了参数估计的中心极限定理.

### §4.1 背景介绍

计算机试验中, 许多复杂响应曲面既含有定量因子又含有定性因子, 例如, Schmidt et al. (2005)描述的计算机试验中数据中心的例子既涉及定性因子(如扩散器位置和热风回流排气口的位置)又涉及定量因子(如电源的功率和扩散器流量).

Qian and Wu (2009)是第一个探讨同时包含定性和定量因子设计的方案. 随后, Qian (2012)又构造了切片拉丁超立方体设计(SLHDs). 这类设计由空间填充设计的几个切片组成, 其中定性因子的每一个水平组合对应的定量因子的设计均是一个小的切片空间填充设计. 因此, 即使对于中等数量的定性因子, 这些设计的试验次数都非常大. 为了解决这个问题, Deng et al. (2015) 提出了一种比较经济高效的设计, 称之为边际耦合设计(MCDs). 为了提高定量因子设计的空间填充性质, He et al. (2017)又介绍了三类边际耦合设计的构造方法.

设计的抽样性质对于数值积分、随机优化和不确定性的量化均是十分必要

的. Stein (1987), Owen (1992), Loh (1996), Ai et al. (2016), Kong et al. (2017) 研究了只涉及定量因子设计的抽样性质. Qian (2012), He and Qian (2016)分别导出了切片拉丁超立体设计的抽样性质和中心极限定理. 然而, 边际耦合设计的大样本性质是一个重要但尚未解决的问题. 本章导出了边际耦合设计下期望估计的渐近方差和相对应的中心极限定理. 此外, 为了比较不同试验次数设计的优劣, 本章利用设计的标准差 $D_{\text{value}}$ 对切片空间填充设计和边际耦合设计进行比较.

令 $\mathbf{w} = (\boldsymbol{\gamma}, \mathbf{x})$ 表示含有定性和定量因子计算机试验的输入, 其中 $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_p)$ 和 $\mathbf{x} = (x_1, \dots, x_q)$ 分别表示 $p$ 个定性和 $q$ 个定量因子,  $\mathbf{x}=(x_1, \dots, x_q)$ 服从 $[0, 1]^q$ 上的均匀分布. 为了研究边际耦合设计的抽样性质, 假设输入 $\mathbf{w}$ 下的响应表示为 $f(\mathbf{w})$

$$f(\mathbf{w}) = \mu + \tau_1(\gamma_1) + \dots + \tau_p(\gamma_p) + g(\mathbf{x}), \quad (4.1)$$

其中 $\mu$ 表示总平均值,  $\tau_i(\gamma_i)$ 表示第 $i$ 个定性因子在水平 $\gamma_i$ 处的效应, 这里 $i = 1, \dots, p$ .  $g(\mathbf{x})$ 表示定量因子 $\mathbf{x}$ 对应的响应值. 假定 $E(g(\mathbf{x})) = 0$  和 $\text{var}(g(\mathbf{x})) = \sigma^2$ , 且定性因子的效应满足

$$\sum_{\gamma \in S_i} \tau_i(\gamma) = 0 \text{ for } i = 1, \dots, p,$$

其中 $S_i$ 表示第 $i$ 个定性因子所有水平的集合.

## §4.2 边际耦合设计的构造

边际耦合设计首先由Deng et al. (2015)提出. 若复杂响应曲面包含 $p$ 个定性因子 $q$ 个定量因子, 令 $D_1$ 和 $D_2$ 分别是定性和定量因子对应的设计, 那么称一个设计 $D = (D_1, D_2)$ 是一个边际耦合设计, 如果 $D_1 = OA(n, s^p, 2)$ 是一个 $n$ 行,  $p$ 列, 每一列有 $s$ 个水平且强度为2的正交表,  $D_2$ 是一个 $n \times q$ 的拉丁超立体设计, 且 $D_1$ 中每个因子的每个水平行对应于 $D_2$ 中的行都是一个小的拉丁超立体设计, 记该边际耦合设计为 $\text{MCD}(D_1, D_2)$ . 为了提高定量因子设计的空间填充性质, He et al. (2017)又提出了三类新的边际耦合设计的构造方法.

为了后续的发展, 本节回顾一下He et al. (2017)提出的 $\text{MCD}(D_1, D_2)$ 的构造算法. 假设设计包含 $n$ 个点, 那么包含 $p$ 个定性因子和 $q$ 个定量因子的边际耦合设计可

以通过 $OA(n, s^p(n/s), 2)$ 和 $OA(n/s, s_1^q, 2)$ 构造, 其中 $s_1$ 和 $s$ 可以不同. 假设 $OA(n, s^p(n/s), 2)$ 和 $OA(n/s, s_1^q, 2)$ 存在并且分别表示为 $A$ 和 $B$ .

---

**算法 4.1** 边际耦合设计的具体构造算法<sup>[41]</sup>

---

- step 1. 基于正交表 $B$ 得到一个基于正交表的拉丁超立方设计 $L(n/s, q)$ , 记为 $L$ ;
- step 2. 将 $A$ 的最后一列的水平 $1, \dots, n/s$ 分别替换成从step1获得的 $L$ 的第 $1, \dots, (n/s)$ 行, 从而得到 $n \times q$ 的矩阵 $\tilde{D}_2$ ;
- step 3. 根据step 2中得到的 $\tilde{D}_2$ , 将其每一列的 $s$ 个水平 $i$ 替换成 $\{(i-1)s+1, \dots, is\}$ 的一个随机排列, 从而获得 $n \times q$ 矩阵 $D'_2 = (d'_{2,ik})_{n \times q}$ ,  $i = 1, \dots, n/s$ ;
- step 4.  $D_2 = (d_{2,ik})_{n \times q}$ 可由 $D'_2$ 通过下式得到

$$d_{2,ik} = \frac{d'_{2,ik} - u_{ik}}{n}, i = 1, \dots, n, k = 1, \dots, q,$$

其中 $u_{ik}$ 是相互独立的 $U(0, 1]$ 上的随机变量.

---

令 $D_1$ 为 $A$ 的前 $p$ 列,  $D_2$ 由算法4.1生成, 则得到的设计 $MCD(D_1, D_2)$ 是一个边际耦合设计.

### §4.3 边际耦合设计的大样本性质

在本节, 首先导出边际耦合设计( $MCD(D_1, D_2)$ )下参数估计的渐近方差, 然后进一步导出参数估计的极限分布. 不失一般性, 令 $D_1 = OA(n, s^p, 2)$ 表示含有 $p$ 个定性因子的设计,  $D_2$ 表示含有 $q$ 个定量因子的拉丁超立方设计, 其中 $p$ 和 $q$ 是有限正整数. 根据边际耦合设计的定义,  $D_1$ 中每个因子的每个水平对应于 $D_2$ 中的行都是一个具有 $m$ 个点的小拉丁超立方设计, 其中 $m = n/s$ . 换句话说, 对于任何一个定性因子, 定量因子的设计均是一个具有 $s$ 个切片的切片拉丁超立方设计.

假设响应曲面模型如(4.1)式所定义, 令 $d_{lc}$ 表示边际耦合设计中第 $l$ 个定性因子第 $c$ 个水平对应的设计点的集合, 其中 $l = 1, \dots, p$ ,  $c = 1, \dots, s$ . 符号 $i \in d_{lc}$ 表示

第 $i$ 个设计点 $\mathbf{w}_i = (\gamma_i, \mathbf{x}_i)$ 属于集合 $d_{lc}$ . 本节感兴趣的是 $\mu_{l1}, \dots, \mu_{ls}$ 的线性组合

$$\eta = \sum_{c=1}^s \lambda_c \mu_{lc}, \quad \mu_{lc} = \frac{1}{m} \sum_{i \in d_{lc}} E(f(\mathbf{w}_i)). \quad (4.2)$$

当 $\lambda_c = 1/s$ ,  $\eta$ 表示响应的总均值; 当 $\sum_{c=1}^s \lambda_c = 0$ ,  $\eta$ 表示第 $l$ 个定性因子的处理对照<sup>[70]</sup>.

**引理4.3.1.** 基于响应曲面模型(4.1)的假设和边际耦合设计的性质, 可得

1.  $\mu_{lc} = \mu + \tau_l(c)$ ,  $l = 1, \dots, p$ ,  $c = 1, \dots, s$ ;

2.  $\hat{\mu}_{lc} = \frac{1}{m} \sum_{i \in d_{lc}} f(\mathbf{w}_i)$ 是 $\mu_{lc}$ 的无偏估计;

3.  $\hat{\eta} = \sum_{c=1}^s \lambda_c \hat{\mu}_{lc}$ 是 $\eta$ 的无偏估计.

为了导出边际耦合设计下 $\hat{\mu}_{lc}$ 和 $\hat{\eta}$ 的渐近方差, 我们首先介绍一下函数型方差分析(ANOVA)<sup>[66, 67]</sup>. 令 $F$ 是 $[0, 1]^q$ 上的均匀分布, 且 $dF = \prod_{k=1}^q dx_k$ ,  $dF_{-k} = \prod_{l \neq k} dx_l$ , 其中 $k = 1, \dots, q$ . 假设 $g(\mathbf{x}) : [0, 1]^q \rightarrow R$ 是 $[0, 1]^q$ 上的连续函数, 均值 $\int g(\mathbf{x}) dF = 0$ , 方差 $\int g^2(\mathbf{x}) dF = \sigma^2$ . 则 $g(\mathbf{x})$ 可以被分解为

$$g(\mathbf{x}) = \sum_{k=1}^q g_{-k}(x_k) + r(\mathbf{x}), \quad (4.3)$$

其中 $g_{-k}(x_k) = \int g(\mathbf{x}) dF_{-k}$ 表示第 $k$ 个因子的主效应函数,  $r(\mathbf{x})$ 表示残差.

由于 $\int g_{-k}(x_k) dF_k = 0$ ,  $\int r(\mathbf{x}) dF_{-k} = 0$ , 因此, 对于 $k \neq l$ ,  $\int g_{-k}(x_k) g_{-l}(x_l) dF = 0$ 以及 $\int g_{-k}(x_k) r(\mathbf{x}) dF = 0$ . 故 $g(\mathbf{x})$ 的方差可以分解为

$$\sigma^2 = \sum_{k=1}^q \int g_{-k}^2(x_k) dx_k + \int r^2(\mathbf{x}) d\mathbf{x}. \quad (4.4)$$

在边际耦合设计下, 为了获得 $\hat{\mu}_{lc}$ 和 $\hat{\eta}$ 的渐近方差, 首先需要导出 $i, j \in d_{lc}$ 且 $i \neq j$ ,  $f(\mathbf{w}_i)$ 和 $f(\mathbf{w}_j)$ 的协方差, 即如下定理4.3.1.



**定理4.3.1.** 假设  $g : [0, 1]^q \rightarrow R$  是一个连续函数, 在边际耦合设计下,  $f(\mathbf{w}_i)$  如模型(4.1)所示. 则对于  $i, j \in d_{lc}$  且  $i \neq j$ , 当  $s$  固定,  $n = ms \rightarrow +\infty$  时, 可得

$$\text{cov}(f(\mathbf{w}_i), f(\mathbf{w}_j)) = -m^{-1} \sum_{k=1}^q \int_0^1 g_{-k}^2(x_k) dx_k + o(m^{-1}). \quad (4.5)$$

根据定理4.3.1, 我们可以导出  $\hat{\eta}$  和  $\hat{\mu}_{lc}$  的渐近方差, 其中  $l = 1, \dots, p$ ,  $c = 1, \dots, s$ .

**定理4.3.2.** 假设  $g : [0, 1]^q \rightarrow R$  是一个连续函数, 在边际耦合设计下,  $f(\mathbf{w}_i)$  如模型(4.1)所示. 则当  $s$  固定,  $n = ms \rightarrow +\infty$  时, 可得

1.  $\text{var}(\hat{\mu}_{lc}) = m^{-1} \sigma^2 - m^{-1} \sum_{k=1}^q \int_0^1 g_{-k}^2(x_k) dx_k + o(m^{-1})$ ,
2.  $\text{var}(\hat{\eta}) = m^{-1} \sum_{c=1}^s \lambda_c^2 \sigma^2 - m^{-1} \sum_{c=1}^s \sum_{k=1}^q \lambda_c^2 \int_0^1 g_{-k}^2(x_k) dx_k + o(n^{-1})$ .

通过定理4.3.2和方程(4.4),  $\hat{\mu}_{lc}$  和  $\hat{\eta}$  的方差可以简单的表示成如下形式:

$$\text{var}(\hat{\mu}_{lc}) = m^{-1} \int r^2(\mathbf{x}) d\mathbf{x} + o(m^{-1}), \quad (4.6)$$

$$\text{var}(\hat{\eta}) = m^{-1} \sum_{c=1}^s \lambda_c^2 \int r^2(\mathbf{x}) d\mathbf{x} + o(n^{-1}).$$

通过上述两个定理, 我们现在可以导出  $\hat{\mu}_{lc}$  和  $\hat{\eta}$  的极限分布.

**定理4.3.3.** 假设  $g : [0, 1]^q \rightarrow R$  是一个连续函数, 在边际耦合设计下,  $f(\mathbf{w}_i)$  如模型(4.1)所示. 则当  $s$  固定,  $n = ms \rightarrow +\infty$  时, 有

$$\begin{aligned} \sqrt{m}(\hat{\mu}_{lc} - \mu_{lc}) &\rightarrow N\left(0, \int_0^1 r^2(\mathbf{x}) d\mathbf{x}\right), \\ \sqrt{m}(\hat{\eta} - \sum_{c=1}^s \lambda_c \mu_{lc}) &\rightarrow N\left(0, \sum_{c=1}^s \lambda_c^2 \int_0^1 r^2(\mathbf{x}) d\mathbf{x}\right), \end{aligned}$$

## §4.4 数值模拟

本节给出了两个数值模拟的例子来证实上节导出的理论结果. 显然, 随机误差的方差可以忽略, 因为不同的试验设计具有相同的随机误差方差. 此外, 我们使

表 4.1 例4.4.1中, SLHD与MCD的比较

	MCD( $D_1, D_2$ )	SLHD		
$n$	32	32	48	64
$\text{var}(\hat{\eta})$	0.0303	0.0401	0.0303	0.0259
$\text{var}(\hat{\mu}_{11})$	0.0330	0.0975	0.0616	0.0453
$\text{var}(\hat{\mu}_{12})$	0.0325	0.0956	0.0622	0.0459
$D_{\text{value}}$	0.9696	3.0592	1.4544	1.6576

用设计的标准方差<sup>[71]</sup>.

$$D_{\text{value}} = n \times \text{var}(\hat{\eta}) \quad (4.7)$$

来比较包含不同试验次数设计的优劣, 即 $D_{\text{value}}$  越小, 设计越好.

**例4.4.1.** 假设响应曲面模型包含4个定性因子和4个定量因子, 并且每个定性因子均有2个水平, 如下所示:

$$f(\mathbf{w}) = 10 + \tau_1(\gamma_1) + \tau_2(\gamma_2) + \tau_3(\gamma_3) + \tau_4(\gamma_4) + g(\mathbf{x}),$$

其中 $\gamma_i \in \{1, 2\}, i = 1, 2, 3, 4, \tau_1 = (-1, 1)', \tau_2 = (-8, 8)', \tau_3 = (-10, 10)', \tau_4 = (-15, 15)',$  且 $g(\mathbf{x}) = 2/3e^{x_1+x_2} - x_4 \sin(x_3) + x_3 - 2.23.$  假定 $\mathbf{x}$ 的分布是 $[0, 1]^4$ 上的均匀分布. 这里 $g(\mathbf{x})$  经常应用在含有定量因子的计算机试验的模拟函数中<sup>[72]</sup>.

该试验中, 边际耦合设计MCD( $D_1, D_2$ )重复进行1000次, 这里定性因子的设计 $D_1$ 是一个对称的正交表 $OA(32, 2^4, 2)$ , 定量因子的设计 $D_2$ 是一个拉丁超立体设计LHD(32, 4). 该边际耦合设计可由算法4.1根据非对称正交表 $OA(32, 2^4 16^1, 2)$ 和对称正交表 $OA(16, 2^4, 2)$ 得到. 由于定性因子有16个水平组合, 因此, 如果我们使用切片拉丁超立体设计, 此设计应该包含16个切片, 每个切片包含 $m$ 个设计点. 若取 $m = 2, 3, 4$ , 则切片拉丁超立方体设计的试验次数分别为32, 48和64, 同样该切片拉丁超立方体设计也重复进行1000次. 不失一般性, 令 $l = 1$ 和 $\lambda_1 = \lambda_2 = 1/2$ , 基于试验得到 $\hat{\eta}$ 和 $\hat{\mu}_{1c}$ 的方差以及 $D_{\text{value}}$ 的值, 如下表所示.

表(4.1)比较了边际耦合设计和切片拉丁超立体设计下 $\hat{\eta}$ 和 $\hat{\mu}_{11}$ 及 $\hat{\mu}_{12}$ 的方差以

及标准差 $D_{\text{value}}$ . 通过表(4.1), 可以得出以下结论: (I) 对于相同的设计点的个数, 如表中第2栏和第3栏所示, 通过比较 $\text{var}(\hat{\eta})$ ,  $\text{var}(\hat{\mu}_{11})$ 和 $\text{var}(\hat{\mu}_{12})$ 以及 $D_{\text{value}}$ 这四个值, 可以看出边际耦合设计的性质优于切片空间填充设计; (II) 当切片空间填充设计的设计点数增加到48时, 对应于表中第4栏, 对于 $\hat{\eta}$ , 该切片拉丁超立方体设计可以实现与边际耦合设计相同的方差减小, 但是边际耦合设计在 $\text{var}(\hat{\mu}_{1i})$ 以及 $D_{\text{value}}$ 上依然比切片拉丁超立方体设计表现好; (III) 进一步, 当切片空间拉丁超立方体设计的设计点数增加到64时, 对应于表中第5栏, 尽管切片拉丁超立方体设计具有相同或者稍微较小的 $\text{var}(\hat{\eta})$ ,  $\text{var}(\hat{\mu}_{11})$ 和 $\text{var}(\hat{\mu}_{12})$ , 但是其设计点的个数是边际耦合设计的2倍. 因此, 边际耦合设计是切片拉丁超立方体设计的一个好的替代设计.

例4.4.1中, 对于边际耦合设计,  $\sqrt{m}(\hat{\eta} - \sum_{c=1}^s \lambda_c \mu_{1c})$  的密度曲线和渐近理论密度曲线如图4.1所示, 其中,  $m = 16, \lambda_1 = \lambda_2 = 1/2$ . 从图中可以看出真实的密度曲线(虚线曲线) 接近相应的渐近理论密度曲线(实线曲线). 该图证实了上节极限分布的理论结果.

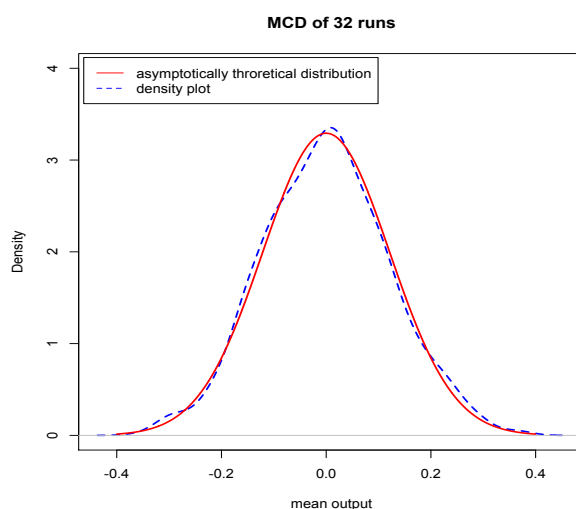


图 4.1 例4.4.1中  $\sqrt{m}(\hat{\eta} - \sum_{c=1}^s \lambda_c \mu_{1c})$  的真实的密度曲线(虚线曲线)和相应的渐近理论密度曲线(实线曲线).

**例4.4.2.** 考虑响应曲面模型包含3个定性因子和4个定量因子, 每个定性因子

表 4.2 例4.4.2中,  $MCD(D_1, D_2)$ 与SLHD 的比较.

	$MCD(D_1, D_2)$	SLHD		
$n$	27	27	54	81
$\text{var}(\hat{\eta})$	0.0086	0.0105	0.0067	0.0050
$\text{var}(\hat{\mu}_{11})$	0.0363	0.2699	0.0957	0.0529
$\text{var}(\hat{\mu}_{12})$	0.0356	0.2752	0.0958	0.0523
$\text{var}(\hat{\mu}_{13})$	0.0376	0.2656	0.1011	0.0509
$D_{\text{value}}$	0.2322	0.2835	0.3618	0.4050

有3个水平时的情形. 如下所示:

$$g(\mathbf{w}) = 10 + \tau_1(\gamma_1) + \tau_2(\gamma_2) + \tau_3(\gamma_3) + g(\mathbf{x}),$$

其中  $\gamma_i \in \{1, 2, 3\}, i = 1, 2, 3, \tau_1 = (0, -2/3, 2/3)', \tau_2 = (0, -1, 1)', \tau_3 = (8, -4, -4)',$   
 $g(\mathbf{x}) = x_1(\sqrt{1 + (x_2 + x_3^2)x_4/x_1 - 1})/2 + x_1 + 3x_4 - 2.16.$

在这个试验中, 我们用一个边际耦合设计 $MCD(D_1, D_2)$ 和三个切片拉丁超立方体设计SLHD来估计 $\eta, \mu_{11}, \mu_{12}, \mu_{13}$ , 每个设计均重复进行1000次. 该边际耦合设计 $MCD(D_1, D_2)$ 可由算法4.1通过正交表 $OA(27, 3^3 9^1, 2)$ 和正交表 $OA(9, 3^4, 2)$ 生成. 由于定性因子具有27个水平组合, 所以, 如果使用切片拉丁超立方体设计, 此设计应该包含27个切片, 若每个切片都包含 $m$ 个设计点,  $m = 1, 2, 3$ , 则相应的切片拉丁超立方体设计的设计点个数为27、54、81. 令 $l = 1, \lambda_1 = \lambda_2 = \lambda_3 = 1/3$ , 表4.2比较了边际耦合设计和切片拉丁超立方体设计下 $\hat{\eta}, \hat{\mu}_{11}, \hat{\mu}_{12}, \hat{\mu}_{13}$ 的方差, 其结果与表4.1的结果相似. 图4.2给出了 $\sqrt{m}(\hat{\eta} - \sum_{c=1}^s \lambda_c \mu_{1c})$ 的真实的密度曲线和其渐近理论密度曲线, 这里 $m = 9, \lambda_1 = \lambda_2 = \lambda_3 = 1/3$ , 该图同样证实了上节极限分布的理论结果.

## §4.5 本章小结

切片拉丁超立方体设计和边际耦合设计均可用于同时包含定性和定量因子

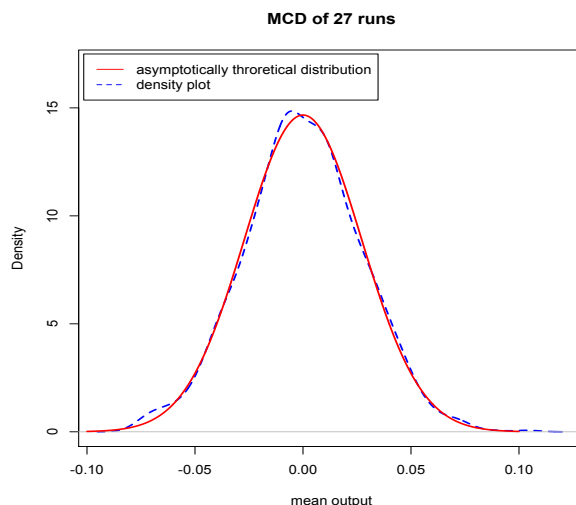


图 4.2 基于例4.4.2的  $\sqrt{m}(\hat{\eta} - \sum_{c=1}^s \lambda_c \mu_{1c})$  的密度曲线(虚线曲线)和相应的渐近理论密度曲线(实线曲线)

的复杂响应曲面的设计,但从经济的角度看,边际耦合设计比切片拉丁超立方体设计有较少的设计点个数.在一定的条件下,本节导出了边际耦合设计下,参数估计的若干大样本性质和对应的中心极限定理.这些结果在统计中广泛应用于建立参数估计的置信区间等.此外,本章还给出了两个例子用于比较边际耦合设计和切片拉丁超立方体设计的优劣,进一步证实,当试验成本较高时,边际耦合设计是一类比切片拉丁超立方体设计更高效的设计.

## §4.6 理论证明

本节给出了定理4.3.1-定理4.3.3 的证明.首先证明定理4.3.1. 为了证明定理4.3.1,我们需要以下引理:

**引理4.6.1.** 假设  $g : [0, 1]^q \rightarrow R$  是连续函数, 令

$$\rho_m(\mathbf{x}; g) = m^q \int_{(j_1-1)/m}^{j_1/m} \dots \int_{(j_q-1)/m}^{j_q/m} g(\mathbf{y}) d\mathbf{y},$$

这里  $\mathbf{x} \in \prod_{i=1}^q [(j_i - 1)/m, j_i/m]$ , 其中  $1 \leq j_1, \dots, j_q \leq m$ . 则有

$$\lim_{m \rightarrow \infty} m^{-q} \sum_{j_1=1}^m \dots \sum_{j_q=1}^m \rho_m^2(\mathbf{x}; g) = \int_{[0,1]^q} g^2(\mathbf{x}) d\mathbf{x}.$$

证明. 我们以  $q = 1$  为例,  $q > 1$  的证明类似.

$$m^{-1} \sum_{i=1}^m \rho_m^2(x; g) = m \sum_{i=1}^m \left[ \int_{(i-1)/m}^{i/m} g(y) dy \right]^2.$$

根据积分中值定理, 存在  $\xi_i \in [(i-1)/m, i/m]$ , 使得  $\int_{(i-1)/m}^{i/m} g(y) dy = g(\xi_i)/m$ , 故

$$m \sum_{i=1}^m \left[ \int_{(i-1)/m}^{i/m} g(y) dy \right]^2 = m^{-1} \sum_{i=1}^m g^2(\xi_i) \xrightarrow{m \rightarrow \infty} \int_0^1 g^2(x) dx.$$

定理4.3.1的证明.

对于  $i, j \in d_{lc}, i \neq j$ ,

$$\text{cov}(f(\mathbf{w}_i), f(\mathbf{w}_j)) = E(g(\mathbf{x}_i)g(\mathbf{x}_j)).$$

假设  $\mathbf{x}$  在  $[0, 1]^q$  上服从均匀分布, 且各分量相互独立. 当  $0 \leq r_i, r_j \leq 1$  时, 定义如下的示性函数:

$$\delta_m(r_i, r_j) = \begin{cases} 1 & [mr_i] = [mr_j]; \\ 0 & [mr_i] \neq [mr_j], \end{cases} \quad (4.8)$$

其中  $[x]$  是小于或等于  $x$  的最大整数.  $\mathbf{x}_i, \mathbf{x}_j$  的联合分布<sup>[12]</sup>为

$$h(\mathbf{x}_i, \mathbf{x}_j) = m^q (m-1)^{-q} \prod_{k=1}^q (1 - \delta_m(x_{ik}, x_{jk})). \quad (4.9)$$

由于

$$\prod_{k=1}^q (1 - \delta_m(x_{ik}, x_{jk})) = 1 - \sum_{k=1}^q \delta_m(x_{ik}, x_{jk}) + \sum_{|u|=2}^q \sum_{k_1 < \dots < k_{|u|}} (-1)^{|u|} \delta_m(x_{ik_1}, x_{jk_1}) \cdots \delta_m(x_{ik_{|u|}}, x_{jk_{|u|}})$$

其中, 对任意的  $u = \{k_1, \dots, k_{|u|}\} \subseteq Q = \{1, \dots, q\}$ ,  $|u|$  是  $u$  的基数,  $g(x_i) = \sum_{k=1}^q g_{-k}(x_{ik}) + r(x_i)$ , 则有

$$\begin{aligned}
 E[g(\mathbf{x}_i)g(\mathbf{x}_j)] &= m^q(m-1)^{-q} \int_{[0,1]^{2q}} g(\mathbf{x}_i)g(\mathbf{x}_j) \prod_{k=1}^q (1 - \delta_m(x_{ik}, x_{jk})) d\mathbf{x}_i d\mathbf{x}_j \\
 &= m^q(m-1)^{-q} \left[ \int_{[0,1]^{2q}} g(\mathbf{x}_i)g(\mathbf{x}_j) d\mathbf{x}_i d\mathbf{x}_j - \sum_{k=1}^q \int_{[0,1]^{2q}} g(\mathbf{x}_i)g(\mathbf{x}_j) \delta_m(x_{ik}, x_{jk}) d\mathbf{x}_i d\mathbf{x}_j \right] + R \\
 &= 0 - m^q(m-1)^{-q} \sum_{k=1}^q \int_{[0,1]^{2q}} g(\mathbf{x}_i)g(\mathbf{x}_j) \delta_m(x_{ik}, x_{jk}) d\mathbf{x}_i d\mathbf{x}_j + R \\
 &= -m^q(m-1)^{-q} \sum_{k=1}^q \int_0^1 \int_0^1 g_{-k}(x_{ik})g_{-k}(x_{jk}) \delta_m(x_{ik}, x_{jk}) dx_{ik} dx_{jk} + R \tag{4.10}
 \end{aligned}$$

其中

$$R = m^q(m-1)^{-q} \sum_{|u|=2}^q \sum_{k_1 < \dots < k_{|u|}} (-1)^{|u|} \int_{[0,1]^{2q}} g(\mathbf{x}_i)g(\mathbf{x}_j) \delta_m(x_{ik_1}, x_{jk_1}) \cdots \delta_m(x_{ik_{|u|}}, x_{jk_{|u|}}) d\mathbf{x}_i d\mathbf{x}_j.$$

(4.10)的第一项

$$\begin{aligned}
 &-m^q(m-1)^{-q} \sum_{k=1}^q \int_0^1 \int_0^1 g_{-k}(x_{ik})g_{-k}(x_{jk}) \delta_m(x_{ik}, x_{jk}) dx_{ik} dx_{jk} \\
 &= -[1 + o(1)] \sum_{k=1}^q \sum_{i=1}^m \int_{(i-1)/m}^{i/m} g_{-k}(x_{ik}) dx_{ik} \int_{(i-1)/m}^{i/m} g_{-k}(x_{jk}) dx_{jk} \\
 &= -[1 + o(1)] m^{-2} \sum_{k=1}^q \sum_{i=1}^m (m \int_{(i-1)/m}^{i/m} g_{-k}(x_{ik}) dx_{ik}) (m \int_{(i-1)/m}^{i/m} g_{-k}(x_{jk}) dx_{jk}) \\
 &= -[1 + o(1)] m^{-2} \sum_{k=1}^q \sum_{i=1}^m \rho_m^2(x_k; g_{-k}).
 \end{aligned}$$

根据引理4.6.1, 可得

$$-[1 + o(1)] m^{-2} \sum_{k=1}^q \sum_{i=1}^m \rho_m^2(x_k; g_{-k}) = -m^{-1} \int_0^1 g_{-k}^2(x_k) dx_k + o(m^{-1}). \tag{4.11}$$

显然,

$$\begin{aligned}
 |R| &< [1 + o(1)] \sum_{|u|=2}^q \sum_{k_1 < \dots < k_{|u|}} \int_{[0,1]^{2q}} g(\mathbf{x}_i) g(\mathbf{x}_j) \delta_m(x_{ik_1}, x_{jk_1}) \cdots \delta_m(x_{ik_{|u|}}, x_{jk_{|u|}}) d\mathbf{x}_i d\mathbf{x}_j \\
 &= [1 + o(1)] \sum_{|u|=2}^q \sum_{k_1 < \dots < k_{|u|}} \sum_{j_{k_1}=1}^m \cdots \sum_{j_{k_{|u|}}=1}^m \left[ \int_{(j_{k_1}-1)/m}^{j_{k_1}/m} \cdots \int_{(j_{k_{|u|}}-1)/m}^{j_{k_{|u|}}/m} g_u(\mathbf{x}_i) d\mathbf{x}_i \right] \\
 &\quad \left[ \int_{(j_{k_1}-1)/m}^{j_{k_1}/m} \cdots \int_{(j_{k_{|u|}}-1)/m}^{j_{k_{|u|}}/m} g_u(\mathbf{x}_j) d\mathbf{x}_j \right] \\
 &= [1 + o(1)] \sum_{|u|=2}^q \sum_{k_1 < \dots < k_{|u|}} m^{-2|u|} \sum_{j_{k_1}=1}^m \cdots \sum_{j_{k_{|u|}}=1}^m \rho_m^2(\mathbf{x}; g_u).
 \end{aligned}$$

再根据引理4.6.1, 有

$$\lim_{m \rightarrow \infty} m^{-|u|} \sum_{j_{k_1}=1}^m \cdots \sum_{j_{k_{|u|}}=1}^m (\rho_m(\mathbf{x}; g_u))^2 = \int g_u^2(\mathbf{x}) d\mathbf{x} < \infty.$$

因为 $|R|$ 包含  $2^q - q - 1$  项且 $q$ 是一个有限整数, 所以

$$|R| < \frac{M}{m^{|u|}} \leq \frac{M}{m^2} = o(m^{-1}), \quad (4.12)$$

其中 $M$ 是一个常数.

将(4.11),(4.12)代入(4.10)从而可以得到定理4.3.1.

**定理4.3.2 的证明.**

首先固定水平 $s$ ,  $n \rightarrow \infty$ 意味着 $m = ns^{-1} \rightarrow \infty$ . 对于 $c = 1, \dots, s$  和 $l = 1, \dots, p$ ,  $d_{lc}$ 是一个普通的拉丁超立方体设计. 由

$$\begin{aligned}
 \text{var}(\hat{\mu}_{lc}) &= \text{var}(m^{-1} \sum_{i \in d_{lc}} f(\mathbf{w}_i)) \\
 &= m^{-1} \sigma^2 + (1 - m^{-1}) \text{cov}(f(\mathbf{w}_i), f(\mathbf{w}_j)),
 \end{aligned} \quad (4.13)$$

以及定理4.3.1, 可以得到当 $m \rightarrow \infty$ 时,

$$\text{var}(\hat{\mu}_{lc}) = m^{-1} \sigma^2 - m^{-1} \sum_{k=1}^q \int_0^1 g_{-k}^2(x_k) dx_k + o(m^{-1}).$$



对于  $l = 1, \dots, p$ ,  $\hat{\eta} = \sum_{c=1}^s \lambda_c \hat{\mu}_{lc}$  是  $\mu_{l1}, \dots, \mu_{ls}$  的一个线性组合. 因此,

$$\text{var}(\hat{\eta}) = \sum_{c=1}^s \lambda_c^2 \text{var}(\hat{\mu}_{lc}) + \sum_{c_1 \neq c_2} \lambda_{c_1} \lambda_{c_2} \text{cov}(\hat{\mu}_{lc_1}, \hat{\mu}_{lc_2}).$$

固定水平  $s$ , 当  $n \rightarrow \infty$  时, 对于  $i \in d_{lc_1}, j \in d_{lc_2}, c_1 \neq c_2$ , 由 Qian (2012)<sup>[3]</sup> 的引理1, 则有

$$\text{cov}(g(\mathbf{x}_i), g(\mathbf{x}_j)) = o(n^{-1}) \text{ and } \text{cov}(\hat{\mu}_{lc_1}, \hat{\mu}_{lc_2}) = o(n^{-1}) \quad (4.14)$$

因此, 当  $n \rightarrow \infty$  时, 可得

$$\text{var}(\hat{\eta}) = m^{-1} \sum_{c=1}^s \lambda_c^2 \sigma^2 - m^{-1} \sum_{c=1}^s \sum_{k=1}^q \lambda_c^2 \int_0^1 g_{-k}^2(x_k) dx_k + o(n^{-1}).$$

**定理4.3.3 的证明.**

由  $\hat{\mu}_{lc} = \frac{1}{m} \sum_{i \in d_{lc}} f(\mathbf{w}_i)$  和方程(4.6), 则  $\sqrt{m}(\hat{\mu}_{lc} - \mu_{lc})$  的均值为0, 并且方差趋向于  $\int_0^1 r^2(\mathbf{x})$ . 接下来只需要证明其极限分布是正态即可. 定义  $\bar{R}_c = \frac{1}{m} \sum_{i \in d_{lc}} r(\mathbf{x}_i)$ ,  $\bar{g}_{-k}(x_k) = \frac{1}{m} \sum_{i \in d_{lc}} g_{-k}(x_{ik})$ , 由方程(4.3), 可得

$$\sqrt{m}(\hat{\mu}_{lc} - \mu_{lc}) = \sqrt{m} \left( \sum_{k=1}^q \bar{g}_{-k}(x_k) + \bar{R}_c \right). \quad (4.15)$$

根据方程(4.6) 和(4.15), 有  $\text{var}_{LHD}(\sqrt{m} \bar{g}_{-k}(x_k)) = o(1)$ . 因为  $\int_0^1 g_{-k}(x_k) dx_k = 0$ , 所以  $E[\sqrt{m} \bar{g}_{-k}(x_k)] = 0$ . 因此, 可得  $E[\sqrt{m} \bar{g}_{-k}(x_k)]^2 \xrightarrow{m \rightarrow \infty} 0$ . 故而, 基于 Chebyshev 不等式,  $\sqrt{m} \bar{g}_{-k}(x_k)$  依概率收敛到0. 因此, 我们只需要证明  $\sqrt{m} \bar{R}_c$  的极限分布是正态分布即可. 显然, 由于残差  $r(\mathbf{x})$  是有界的, 故在 IID 样本下,  $\sqrt{m} \bar{R}_c$  的极限分布是正态分布. 由 Owen (1992) 中的引理2, 对于整数  $k \geq 1$ ,

$$E_{LHD}\{(\sqrt{m} \bar{R}_c)^k\} = E_{IID}\{(\sqrt{m} \bar{R}_c)^k\} + O(m^{-1}).$$

因此, 根据矩方法<sup>[74]</sup>, 在拉丁超立方体设计下  $\sqrt{m} \bar{R}_c$  的极限分布是正态分布. 从而得到  $\hat{\mu}_{lc}$  的极限分布是正态分布.

显然, 对于  $l = 1, \dots, p$ , 由引理4.3.1 和方程(4.6), 可得  $E(\sqrt{m}(\hat{\eta} - \sum_{c=1}^s \lambda_c \mu_{lc})) = 0$ ,

且  $\sqrt{m}(\hat{\eta} - \sum_{c=1}^s \lambda_c \mu_{lc})$  的方差趋向于  $\sum_{c=1}^s \lambda_c^2 \int_0^1 r^2(x) dx$ . 根据  $\hat{\mu}_{lc}$  的极限分布是正态分布, 且  $\hat{\mu}_{l1}, \dots, \hat{\mu}_{ls}$  的任意一个线性组合的极限分布也是正态分布. 因此, 可以得到  $\hat{\eta} = \sum_{c=1}^s \lambda_c \hat{\mu}_{lc}$  的极限分布也是正态分布. 故定理4.3.3得证.

## 第五章 总结与展望

现代科学试验往往需要探究复杂响应曲面, 复杂响应曲面的抽样与设计在寻找复杂响应曲面的最优值, 贝叶斯计算以及数值积分等领域中有广泛的应用. 当没有响应曲面的先验信息时, 空间填充设计是探索响应曲面比较流行的设计方法, 该设计方法使试验点均匀地分散在区域内. 但是当获取响应曲面一定的信息后, 仍然使用空间填充设计有时会抓不住响应曲面的趋势特征. 例如, 若试验的目的是寻找响应曲面的最大值, 且知道曲面具有多个局部最大值时, 空间填充设计难免会使设计点落入响应值较低的区域, 进而造成设计点的浪费. 因此, 本文基于Kullback-Leibler偏差提出三类新的确定性抽样方法来获得复杂响应曲面的代表点集. 此类代表点能够抓住响应曲面的趋势特征, 具有在响应值高的区域多布点, 响应值低的区域少布点, 并且不同点之间保持一定的距离等优点. 这些优点可用于寻找响应曲面 $f(\mathbf{x})$ 的最大值. 若目标是寻找响应曲面的最小值, 只须将 $f(\mathbf{x})$ 变换为 $-f(\mathbf{x})$ 即可. 此外, 当复杂响应曲面即包含定性因子又包含定量因子时, 常用的抽样设计方法是边际耦合设计. 边际耦合设计是一类比普通的切片空间填充设计更经济有效的抽样设计方法. 本文研究了边际耦合设计的大样本性质, 进一步导出了边际耦合设计的中心极限定理.

### §5.1 本文的主要工作

当复杂响应曲面只包含定量因子时, 传统复杂响应曲面的抽样设计方法一般是空间填充设计, 例如常用的均匀设计和拉丁超立方设计等. 但是当获取复杂响应曲面一定的信息时, 例如已知响应曲面包含多个局部最大值时或者已知响应函数的解析式时, 传统空间填充设计会使许多设计点落在响应值较低或者为常数的区域, 这样会造成设计点的浪费. 因此, 本文提出了一种加权的空间填充设计方法, 该方法是通过确定性抽样实现的. 本文将响应曲面转化为非标准化的密度函数, 进而通过最小化一组点的核密度估计与该目标密度的Kullback-Leibler偏差来生成确定性样本点, 得到的点集称为Kullback-Leibler(KL)点. 进一步本文还从理论上证明了KL点作为响应曲面代表点的合理性.

针对复杂结构的响应曲面, 基于Kullback-Leibler偏差提出两类推广的KL点. 当响应曲面没有显式表达或者计算耗时, 基于Kullback-Leibler偏差和函数的核估计方法, 提出一类推广的KL点称为自适应的KL(AKL)点. 即通过序贯的近似复杂的响应曲面, 进而自适应地更新KL点. AKL点可用于复杂密度函数的抽样和响应曲面最优值的寻找. 当响应曲面的维数较高时, 基于Kullback-Leibler偏差和函数的最小邻近距离估计方法, 提出了第二类推广的KL点称为KL-nn(KL-nn)点. KL-nn点也能抓住响应曲面的趋势特征, 也可看作是一类加权的空间填充设计. 数值结果表明, KL-nn点不仅有较好的空间填充性质, 而且当响应曲面的维数较高时在数值积分上与最小化能量设计和支撑点相比有较小的积分误差.

当复杂响应曲面既包含定性因子又包含定量因子时, 切片的空间填充设计是第一类解决此类问题的设计, 即使得定性因子的每个水平组合对应的定量因子的设计都是一个小的空间填充设计. 然而, 即使对于中等数量的定性因子, 这些设计的试验次数都非常大. Deng et al. (2015)提出了一类经济高效的设计, 称之为边际耦合设计. 本文研究了边际耦合设计的大样本性质, 并进一步导出了边际耦合设计的中心极限定理. 这些大样本性质可被用于求参数估计的置信区间, 量化参数估计的不确定性等.

## §5.2 未来的研究工作

近年来的研究热点之一是大数据的子抽样问题. Yu et al. (2020)基于拟似然估计提出A-最优、L-最优准则下的泊松抽样方法, Wang et al. (2017)基于线性回归模型提出D-最优准则下的子抽样方法. Ai et al. (2020)基于分位数回归模型提出A-最优准则下的泊松抽样方法. 将来我们想利用KL点和KL-nn点好的代表性, 即在感兴趣的区域内多布点, 并且不同的点保持一定的距离, 用于探索不依赖模型的大数据子抽样问题.

尽管本文已经给出了三类好的确定性抽样方法, 即KL点、AKL点以及KL-nn点, 但是依然还存在很多问题等待研究. 根据Hickernell (1998)的思想, 高维的KL点、AKL点以及KL-nn点不仅应该很好地代表分布函数 $F$ , 而且也应该很好地代表 $F$ 的边际分布. 这种投影性质在大多数QMC点的文献中都有介绍, 为此我们

需要将这种好的投影性质也纳入到KL点、AKL点以及KL-nn 点的框架中.

本文在生成KL点、AKL点以及KL-nn点时利用的是序贯地一次生成一个点的策略, 因此算法的时间会随着维数的升高而增大, 未来我们也需要研究更高效的算法来生成KL点、AKL点以及KL-nn点.

第四章在研究边际耦合设计的大样本性质时, 我们基于的是定性因子与定量因子是独立时的情形, 当定性因子与定量因子具有交互作用时, 相应的边际耦合设计的大样本性质也将在今后的工作中进行研究.



## 参考文献

- [1] Joseph V R, Wang D P, Gu L, et al. Deterministic sampling of expensive posteriors using minimum energy designs[J]. *Technometrics*, 2019, 61(3): 297-308.
- [2] Mak S, Joseph V R. Support points[J]. *Annals of Statistics*, 2018, 46(6A): 2562-2592.
- [3] Qian P Z G. Sliced Latin hypercube designs[J]. *Journal of the American Statistical Association*, 2012, 107(497): 393-399.
- [4] He X, Qian P Z G. A central limit theorem for nested or sliced Latin hypercube designs[J]. *Statistica Sinica*, 2016, 26(3): 1117-1128.
- [5] Deng X W, Hung Y, Lin C D. Design for computer experiments with qualitative and quantitative factors[J]. *Statistica Sinica*, 2015, 25: 1567-1581.
- [6] Sacks J, Schiller S B, Welch W J. Designs for computational experiments[J]. *Technometrics*, 1989, 31(1): 41-47.
- [7] Currin C, Mitchell T, Morris M, et al. Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments[J]. *Journal of the American Statistical Association*, 1991, 86(416): 953-963.
- [8] Singh R K, Joseph V R, Melkote S N. A statistical approach to the optimization of a laser-assisted micromachining process[J]. *The International Journal of Advanced Manufacturing Technology*, 2011, 53(1-4): 22-230.
- [9] Hodges J S, Reich B J. Adding spatially-correlated errors can mess up the fixed effect you love[J]. *The American Statistician*, 2010, 64(4): 32-334.
- [10] Tuo R, Wu C F J, Yu D. Surrogate modeling of computer experiments with different mesh densities[J]. *Technometrics*, 2014, 56(3): 37-380.
- [11] Paciorek C J. The importance of scale for spatial-confounding bias and precision of spatial regression estimators[J]. *Statistical Science*, 2010, 25(1): 10-125.
- [12] McKay M D, Santner T J, Conover W J. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code[J]. *Technometrics*, 1979, 21: 239-245.
- [13] Fang K T, Wang Y. Number-theoretic methods in statistics[M]. London: Chapman & Hall, 1994.
- [14] Schmidt R R, Cruz E E, Iyengar M K. Challenges of data center thermal management[J]. *IBM Journal of Research and Development*, 2005, 49(4/5): 709-723.
- [15] Ye K Q. Orthogonal column Latin hypercubes and their application in computer experiments[J]. *Journal of the American Statistical Association*, 1993, 93(444): 1430-1439.

- [16] Bingham D, Sitter R R, Tang B. Orthogonal and nearly orthogonal designs for computer experiments[J]. *Biometrika*, 2009, 96(1): 51 – 65.
- [17] Sun F S, Liu M Q, Lin D K J. Construction of orthogonal Latin hypercube designs[J]. *Biometrika*, 2009, 96(4): 971-974.
- [18] Sun F S, Liu M Q, Lin D K J. Construction of orthogonal Latin hypercube designs with flexible run sizes[J]. *Journal of Statistical Planning and Inference*, 2010, 140(11): 3236-3242.
- [19] Zhou Y D, Xu H Q. Space-filling fractional factorial designs[J]. *Journal of the American Statistical Association*, 2014, 109(507): 1134-1144.
- [20] Tang B X. Orthogonal array-based Latin Hypercubes[J]. *Journal of the American Statistical Association*, 1993, 88(424): 1392-1397.
- [21] Hedayat A S, Sloane N J A, Stufken J. Orthogonal array: Theory and applications[M]. New York: Springer, 1999.
- [22] Ai M, Jiang B, Li K. Construction of sliced space-filling designs based on balanced sliced orthogonal arrays[J]. *Statistica Sinica*, 2014, 24: 1685-1702.
- [23] Chen J, Qian P Z G. Latin hypercube designs with controlled correlations and multi-dimensional stratification[J]. *Biometrika*, 2014, 101(2): 319-332.
- [24] He X, Qian P Z G. A central limit theorem for general orthogonal array based space-filling designs[J]. *Annals of Statistics*, 2014, 42(5): 1725-1750.
- [25] He Y Z, Tang B X. Strong orthogonal arrays and associated Latin hypercubes for computer experiments[J]. *Biometrika*, 2013, 100(1): 254-260.
- [26] He Y Z, Tang B X. A characterization of strong orthogonal arrays of strength three[J]. *Annals of Statistics*, 2014, 42(4): 115-128.
- [27] Fang K T, Li R, Sudjianto A. Design and modeling for computer experiments[M]. New York: Chapman & Hal, 2006.
- [28] Fang K T, Liu M Q, Qin H, et al. Theory and application of uniform experimental designs[M]. Singapore & Beijing: Springer & Science Press, 2018.
- [29] Chen H, Huang H Z, Lin D K J, et al. Uniform sliced Latin hypercube designs[J]. *Applied Stochastic Models in Business and Industry*, 2016, 32(5): 574 – 584.
- [30] Zhou Y D, Fang K T, Ning J H. Mixture discrepancy for quasi-random point sets[J]. *Journal of Complexity*, 2013, 29(3 – 4): 283 – 301.
- [31] Hickernell F. A generalized discrepancy and quadrature error bound[J]. *Mathematics of Computation*, 1998, 67(221): 299-322.



- [32] Korobov N M. The approximate computation of multiple integrals[J]. Dokl. Akad. Nauk. SSSR, 1959, 124: 1207-1210.
- [33] Sun F, Wang Y P, Xu H. Uniform projection designs[J]. Annals of Statistics, 2019, 47(1): 641-661.
- [34] Johnson M E, Moore L M, Ylvisaker D. Minimax and maximin distance designs[J]. Journal of Statistical Planning and Inference, 1990, 26(2): 131-148.
- [35] Morris M D, Mitchell T J. Exploratory designs for computer experiments[J]. Journal of Statistical Planning and Inference, 2018, 43: 381 - 402.
- [36] van Dam E R, Rennen G, Husslage B. Bounds for maximin Latin hypercube designs[J]. Operations Research, 2009, 57(3): 595-608.
- [37] Joseph R V, Hung Y. Orthogonal-maximin Latin hypercube designs[J]. Statistica Sinica, 2012, 18(1): 171 - 186.
- [38] Joseph V R, Dasgupta T, Tuo R, et al. Sequential exploration of complex surfaces using minimum energy designs[J]. Technometrics, 2015, 57(1): 64-74.
- [39] Mak S, Joseph V R. Projected support points: A new method for high-dimensional data reduction. <https://arxiv.org/abs/1708.06897>, 2017.
- [40] Qian P Z G, Wu, C F J. Sliced space-filling designs[J]. Biometrika, 2009, 96(4): 733-739.
- [41] He Y Z, Lin C D, Sun F S. On construction of marginally coupled designs[J]. Statistica Sinica, 2017, 27(2): 665-683.
- [42] Kennedy M C, O'Hagan A. Bayesian calibration of computer models (with discussion)[J]. Journal of the Royal Statistical Society-Series B, 2001, 63(3): 425-464.
- [43] Brooks S, Gelman A, Jones G, Meng, X L. Handbook of Markov Chain Monte Carlo, Boca Raton[M]. FL: CRC Press, 2011.
- [44] Dunson D B, Johndrow J. The Hastings algorithm at Fifty[J]. Biometrika, 2020, 107(1): 1 - 23.
- [45] Fearnhead P, Bierkens J, Pollock M, et al. Piecewise deterministic Markov processes for continuous-time Monte Carlo[J]. Statistical Science, 2018, 33:386 - 412.
- [46] Hastings W K. Monte Carlo sampling methods using Markov chains and their applications[J]. Biometrika, 1970, 57(1): 97 - 109.
- [47] Korattikara A, Chen Y, Welling M. Austerity in MCMC land: Cutting the Metropolis-Hastings budget. in International Conference on Machine Learning, 2014, 181 - 189.
- [48] Li W, Ahn S, Welling M. Scalable MCMC for mixed membership stochastic blockmodels. in Artificial Intelligence and Statistics, 2016, 723 - 731.

- [49] Quiroz M, Kohn R, Villani M. Speeding up MCMC by efficient data subsampling[J]. Journal of the American Statistical Association, 2019, 114: 831 – 843.
- [50] Roberts G O, Rosenthal J S. Optimal scaling for various Metropolis – Hastings algorithms[J]. Statistical Science, 2001, 16(4): 351 – 367.
- [51] Roberts G O, Rosenthal J S. General state space Markov Chains and MCMC algorithms[J]. Probability Surveys, 2004, 1(1): 20 – 71.
- [52] Haario H, Saksman E, Tamminen J. Adaptive proposal distribution for random walk Metropolis algorithm[J]. Computational Statistics, 1999, 14(3): 375-395.
- [53] Sobol' I M. On the distribution of points in a cube and the approximate evaluation of integrals[J]. Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki, 1967, 7(4): 784–802.
- [54] Kullback S, Leibler R. On information and sufficiency[J]. Annals of Mathematical Statistics, 1951, 22(1): 79-86.
- [55] Csiszár I. Eine Informationstheoretische ungleichung und ihre anwendung auf den Beweis der Ergodizität on Markhoffschen Ketten[J]. Publications of the Mathematical Institute of the Hungarian Academy of Sciences, 1963, 8: 84-108.
- [56] Dudewicz E J, Van D M E C. Entropy-based tests of uniformity[J]. Journal of the American Statistical Association, 1981, 76(376): 967-974.
- [57] Jourdan A, Franco J. A new criterion based on Kullback-Leibler information for space filling designs[J]. <https://www.researchgate.net/publication/>, 2009.
- [58] Cover T M, Thomas J A. Elements of information theory[M]. Publications of the American Statal Association, 2006.
- [59] Härdle W G, Werwatz A, Müller M, et al. Nonparametric and semiparametric models[M]. New York: Springer, 2004.
- [60] Nelder J A, Mead R. A simplex method for function minimization[J]. The Computer Journal, 1965, 7(4): 308-313.
- [61] Genz A. Testing multidimensional integration routines[J]. In Proc. of International Conference on Tools, Methods and Languages for Scientific and Engineering Computation, 1984, 81-94.
- [62] Santner T J, Williams B J, Notz W I. The design and analysis of computer experiments[M]. New York: Springer, 2003.
- [63] Fasshauer G. Meshfree approximation methods with MATLAB. Singa- pore: World Scientific Publishers, 2007.

- [64] Kosachenko L F, Leonenko N N. Sample estimate of entropy of a random vector[J]. Problem of Information Transmission, 1987, 23(2): 95-101.
- [65] Stein M. Large sample properties of simulations using Latin hypercube sampling[J]. Technometrics, 1987, 29(2): 143-151.
- [66] Owen A B. A central limit theorem for Latin hypercube sampling[J]. Journal of the Royal Statistical Society. Series B, 1992, 54(2): 541-551.
- [67] Loh W L. On Latin hypercube sampling[J]. Annals of Statistics, 1996, 24(5): 2058-2080.
- [68] Ai M Y, Kong X S, Li K. A general theory for orthogonal array based Latin hypercube sampling[J]. Statistica Sinica, 2016, 26(2): 761-777.
- [69] Kong X S, Ai M Y, Mukerjee R. Central limit theorems for four new types of U-designs[J]. Statistics, 2017, 51(3): 655-667.
- [70] Mukerjee R, Wu C F J. A modern theory of factorial designs[M]. New York: Springer, 2006.
- [71] Kiefer J. Optimum designs in regression problems, II[J]. The Annals of Mathematical Statistics, 1961, 32(1): 298-325.
- [72] Cox D D, Park J S, Singer C E. A statistical method for tuning a computer code to a data base[J]. Computational statistics & data analysis, 2001, 37(1): 77-92.
- [73] Tsybakov A B. Introduction to nonparametric estimation[M]. New York: Springer, 2009.
- [74] Chung K L. A course in probability theory[M]. New York: Academic Press, 1974.
- [75] Dick J, Kuo F Y, Sloan I H. High-dimensional integration: the quasi-Monte Carlo way[J]. Acta Numerica, 2013, 22: 133-288.
- [76] Yu J, Wang H Y, Ai M Y, et al. Optimal distributed subsampling for maximum quasi-likelihood estimators with massive data[J]. Journal of the American Statistical Association, 2020, 00(0): 1-12.
- [77] Wang H Y, Yang M, Stufken J. Information-based optimal subdata selection for big data linear regression[J]. Journal of the American Statistical Association, 2017, 114(525): 292-405.
- [78] Ai M Y, Wang F, Yu J. Optimal subsampling for large-scale quantile regression[J]. Journal of Complexity, 2020, 62: 1-27.

## 在学期间公开发表论文及著作情况

文章名称	发表刊物（出版社）	刊发时间	刊物级别	第几作者
A central limit theorem for marginally coupled designs	Statistics and Probability Letters		SCI	1
Statistical inference-based research on sampling time of vehicle driving cycle experiments	Transportation Research Part D Transport and Environment		SCI	3
Constructing optimal designs for order-of-addition experiments using an exchange algorithm	Journal of Computational Statistics & Data Analysis	在投	SCI	2
Deterministic sampling based on Kullback-Leibler divergenc and its application.		Manuscript		1
A new Markov Chain Monte Carlo scheme based on the nearest neighbor distance for Bayesian inference.		Manuscript		1

## 致谢

行文至此,意味着我在东北师范大学的求学生涯即将结束,目之所及,皆是回忆.2010年,我高中毕业满怀激情与梦想来到东北师范大学,从本科到硕士再到博士,转眼就是11年,时间匆匆划过,如白驹过隙,忽然如梦.回首这11年的求学时光,有失败时的气馁,有受挫时的焦虑,有解决问题后的兴奋,有过后又发现错误时的无奈,但更多的是对这些年无数难忘时刻的追忆,以及对许多关心和帮助过我的老师,同学,朋友,家人的感谢。

经师易寻,人师难遇。首先,要感谢我的导师孙法省教授,从硕士到博士,无论从学习上还是从生活上,孙老师均给予了我无数的指导和帮助,见证了我一路的成长。在学习上,孙老师扎实渊博的学术造诣,严谨的科研态度,对科研的无限热情坚定了我读博的想法。我仍记得在我硕士一、二年级时,对科研论文没有任何想法,孙老师教育我们读一篇文章要读透包括其参考文献。任何学术科研上的创新都是基于前人的工作,根据孙老师总结的写文章的策略——“西红柿炒鸡蛋”,我在讨论班上讲过两篇文章后,开始着手写第一篇SCI论文时,尤记得当时由于经验不足,导致论文错误百出,孙老师不厌其烦的帮助我修改论文,传授给我撰写论文的方法与技巧。今年,在我撰写第二篇SCI论文时,孙老师从论文的最初想法,到论文中的逻辑关系以及繁琐的证明过程,均不辞辛苦的帮我修改和指正,尤其是考虑到我即将毕业,孙老师过年期间都没有休息来帮我修改文章。孙老师认真细致的学术态度,活跃的学术思维,从容随和的生活态度,以及处处为学生考虑的奉献精神是我值得学习一生的榜样。在生活上,孙老师见证了我一路从结婚到生子,孙老师时常教导我们学习重要,生活也重要。正是因为孙老师豁达的生活态度,才让我没有顾虑的结婚和生子,感谢孙老师给予我生活上的关心和帮助。

其次,要衷心的感谢学院的白志东教授、史宁中教授、郭建华教授、高巍教授、已故的陶剑教授、郑术蓉教授、朱文圣教授、刘秉辉教授等在校期间对我的知识传授和谆谆教诲。

感谢在我求学路上一路陪伴我的家人。感谢父母给予我的养育之恩,在东北师大求学的这十一年,我回家看望他们的次数屈指可数,对此,我内心十分内疚。尤其近两年由于疫情的原因,我一直都没有回家。尽管他们总是说让我安心学

习,但是我心里明白他们非常渴望我回家。此外,感谢我的兄弟姐妹对我生活上的关心,以及在我不在家的时候,他们对父母的照顾。感谢我的丈夫李帅先生对我学业上的支持。自从我们的宝贝赞赞出生后,他为了让我专心学习,辞去了沈阳的工作来到长春,主动承担大部分的家务以及带孩子,尤其周六日全天都在带孩子,照顾孩子的饮食起居等。此外,自从结婚后,感谢他一直以来对我的包容和理解,以及在我面对困难时给与我的鼓励和支持。感谢我的婆婆这两年一直在长春帮我们带孩子,给了我更多的时间专心搞学习。在来长春前,她几乎没有离开过河南老家,来到人生地不熟的长春帮我们带孩子需要牺牲很多,尤其还有一个妹妹在老家上高中,这种牺牲不是用几句感谢的语言能表达的。

感谢我的室友孟静波、荣鑫淼、王潇逸、朱红月、邹婷婷。我们几个从本科到现在,一路走过风华正茂的十一年,感慨万千。如今我们又一起毕业了,真的特别高兴。以前我们一起吃饭、一起上课、一起在寝室打打闹闹的景象历历在目。我们几个已经不仅仅是同学、室友的关系了,我们之间更像是家人。随着我们学位论文答辩的顺利结束,也意味着我们即将要分开了,她们时常羡慕我家庭、孩子都有了,如今也即将拿到博士学位。虽然我总是开玩笑劝说她们不要结婚,因为有了家庭和孩子后,生活会很累。但是,我心里却是希望她们赶紧拥有自己的家庭。

感谢我的师姐王东莹博士给与学习和生活上的帮助,尤记得东莹姐第一次帮我修改文章时,教给了我写文章时如何做到逻辑合理、语言贴切等。感谢师妹周怡杉、胡波、李鹏南、贾俊华、胡冰菁、文冰、夏新新、程紫童、王艺凝;师弟韩子健、朱甲庆、赵强、刘利峰、张新琦、高国丰,与他们交流使我收获颇丰。感谢他们创造了积极的学习氛围。此外,还要感谢我博士间的师妹陈晶、李铭、梁春辉,与他们一起学习提高了我的效率,感谢她们对我生活的帮助,她们创造的轻松的工作环境使我在艰难的学术探索中有了许多快乐。

感谢东北师大给我的十一年,让我收获了知识与快乐。感谢学院所有关心和帮助我的老师。感谢日益强大的祖国给我们创造的安全学习和生活环境。

祝福我的导师,身体健康!祝福我的同学,前程似锦!祝福母校,孕育英才!

感谢所有相遇,聚散终有时,未来方可期,山水一程,三生有幸!

王素敏

2021年5月于长春

## 附 录

在本章中,我们给出第二、三章部分程序的代码.

### A.1 利用随机化的NM算法求函数最小值的MATLAB代码

```

1 function [xMin, dMin, fMin, nEval, fLoc] = fminadaresseed(
2     f, X, nRes, mu0, Sigma0, sigsq, delay, lb, ub, A, b)
3 % 函数[xMin, dMin, fMin, nEval, fLoc] = fminadares(f, X, nRes,
4 % sigsq, delay, lb, ub, A, b) 通过随机化Nelder-Mead 最小化目标函数.
5 %
6     [nObs, nDim] = size(X);
7     if nObs ≤ delay
8         frnd = @()mvnrnd(mu0, Sigma0, 1);
9
10    else
11        Sigma = sigsq .* eye(nDim);
12        frnd = @()mvnrnd(X(randi(nObs), :), Sigma, 1);
13    end
14    % 多次局部搜索的约束Nelder-Mead方法
15    opt = optimset('tolfun', 1e-3, 'tolx', 1e-3, 'display', ...
16        'off');
17    XLoc = zeros(nRes, nDim);
18    fLoc = zeros(nRes, 1);
19    nEval=0;
20    X0 = genfeasible(frnd, nRes, nDim, lb, ub, A, b);
21    for i = 1:nRes
22        [XLoc(i, :), fLoc(i), ~, out] = fminsearchcon( ...
23            f, X0(i, :), lb, ub, A, b, [], opt);
24        nEval = nEval + out.funcCount;
25    end
26    % 求全局最优值
27    [fMin, iMin] = min(fLoc);
28    xMin = XLoc(iMin, :);
29    % 最后一次调用目标函数以获得得分向量

```

```

29     [¬, dMin] = f(xMin);
30
31     nEval=nEval+1;
32 end
33
34 function X = genfeasible(frnd, n, nDim, lb, ub, A, b)
35     X = zeros(n, nDim);
36     randn('seed', 53)
37     for i = 1:n
38         t = frnd();
39         while ¬isfeasible(t, lb, ub, A, b)
40             t = frnd();
41         end
42         X(i, :) = t;
43     end
44 end
45
46 function tf = isfeasible(x, lb, ub, A, b)
47     if ¬isempty(lb) && any(x < lb)
48         tf = false;
49         return
50     end
51     if ¬isempty(ub) && any(x > ub)
52         tf = false;
53         return
54     end
55     if ¬isempty(A) && any(A * x > b)
56         tf = false;
57         return
58     end
59     tf = true;
60 end

```



## A.2 响应曲面函数KL点的MATLAB代码

```

1 function [X, e, nEval] = klpointsbanagreedy(nDim, fpdf, fmin, ...
    nIter)
2 % 函数[X, e, nEval] =klpointsbanagreedy(nDim, fpdf, fmin, k,
    nIter)
3 % 通过贪婪算法生成Kullback-Leibler (KL) points.
4 %
5 % Input:
6 % nDim - 目标函数的维数.
7 % fpdf - 目标函数.
8 % fmin - 搜索函数最小值的算法.
9 % nIter - KL点的个数.
10 %
11 % Output:
12 % X -生成点矩阵.
13 % e -每次迭代时用于最小化的函数.
14 % nEval - 每次迭代的需要计算目标函数值的次数.
15 %
16
17 X = zeros(nIter, nDim);
18 y = zeros(nIter, 1);
19 e = zeros(nIter, 1);
20 nEval = zeros(nIter, 1);
21
22 % 生成x1
23 f = @(XNew)fq(XNew, fpdf);
24 [X(1, :), y(1), e(1), nEval(1)] = fmin(f, double.empty(0, ...
    nDim));
25 fprintf('n = 1\n');
26
27 % 生成接下来的点
28 for n = 2:nIter
29     f = @(XNew)fe(XNew, fpdf, X, y, n);
30     [X(n, :), y(n), e(n), nEval(n)] = fmin(f, X(1:(n - ...
        1), :));
31     fprintf('n = %d\n', n);
32 end
33 end
34 function [e, yNew] = fe(XNew, fpdf, X, y, n)
35     [nNew, nDim] = size(XNew);

```

```

36     yNew = fpdf(XNew);
37     exist=X(1:(n - 1), :);
38     matrix=[XNew;exist];
39     d=pdist2(matrix,matrix);
40     [m,t]=size(d);
41     dnew=[];
42     for i=1:m
43         dnew=[dnew;d(i,[1:i-1 i+1:t])];
44     end
45     end
46     c11=(n)^(-1/6);
47     %hn=0.06*c11;
48     dnew=exp(-dnew/hn);
49
50     d1=sum(dnew(1,:)) ;
51     he=sum(log(d1));
52     ce=log(yNew);
53     kl=he-ce-log(n*(hn)^nDim);
54     e=kl;
55
56
57     end
58
59     function [q, yNew] = fq(XNew, fpdf)
60         nDim = size(XNew, 2);
61         yNew = fpdf(XNew) .^ (1 ./ (2 .* nDim));
62         q = 1 ./ yNew;
63     end

```

### A.3 响应曲面函数AKL点的MATLAB代码

```

1 function [X, e, nEval] = aklpoinstsgreedy(nDim, fpdf, fmin, ...
    nIter,initial)
2 % 函数[X, e, nEval] = aklpoinstsgreedy(nDim, fpdf, fmin,
    nIter,initial)
3 % 通过贪婪算法生成adaptive Kullback-Leibler points (AKL) 点.
4 %

```

```

5 % Input:
6 % nDim - 目标函数的维数.
7 % fpdf - 目标函数.
8 % fmin - 搜索函数最小值的算法.
9 % nIter - KL点的个数.
10 % initial- 生成AKL点时的初始设计.
11 % Output:
12 % X -生成的AKL点矩阵.
13 % e -每次迭代时用于最小化的函数.
14 % nEval - 每次迭代的需要计算目标函数值的次数.
15 %
16 %
17     n0=size(initial,1);
18     X = [initial;zeros((nIter-n0), nDim)];
19     y = zeros(nIter, 1);
20     e = zeros(nIter, 1);
21     nEval = zeros(nIter, 1);
22
23
24
25     for n = (n0+1):nIter
26         f = @(XNew)efe(XNew, fpdf, X, n);
27         [X(n, :), y(n), e(n), nEval(n)] = fmin(f, X(1:(n - ...
28             1), :));
29         fprintf('n = %d\n', n);
30     end
31 end
32 function [e, yNew] = efe(XNew, fpdf, X, n)
33     [n, nDim] = size(XNew);
34     %S=X(1:(n - 1), :);
35     Y=zeros(n-1,1);
36     for i=1:(n-1)
37         Y(i)=log(fpdf(S(i, :)));
38     end
39     %yNew=dace(XNew,S,Y)
40     %
41     %exist=X(1:(n - 1), :);
42     %matrix=[XNew;exist];

```

```

43 d=pdist2(matrix,matrix);
44 [m,t]=size(d);
45 dnew1=[];
46 for i=1:m
47     dnew1=[dnew1;d(i,[1:i-1 i+1:t])];
48
49 end
50 de=pdist2(exist,exist);
51 [s,t]=size(de);
52 dd=[];
53 for i=1:s
54     dd=[dd;de(i,[1:i-1 i+1:t])];
55
56 end
57 %
58 hn=min(min(dd))./(n^(1/(nDim+4)));
59
60 dnew=exp(-dnew1(1,:)/hn);
61
62 d1=sum(dnew(1,:));
63
64 he=sum(log(d1));
65
66 %pre=yNew;
67
68 kl=he-pre-log(n*(hn)^nDim);
69
70 e=kl;
71
72 end

```

```

1 function [X, e, nEval] = blackboxseklpointsgreedy(nDim, fpdf, ...
    fmin, nIter, initial)
2 %函数[X, e, nEval] = blackboxseklpointsgreedy(nDim, fpdf, fmin, k,
    nIter)
3 %通过贪婪算法生成黑盒子函数的adaptive Kullback-Leibler points (AKL)点
4 %
5 % Input:

```

```

6 % nDim - 目标函数的维数.
7 % fpdf - 目标函数.
8 % fmin - 搜索函数最小值的算法.
9 % nIter - KL点的个数.
10 % initial- 生成AKL点时的初始设计.
11 % Output:
12 % X -生成的AKL点矩阵.
13 % e -每次迭代时用于最小化的函数.
14 % nEval - 每次迭代的需要计算目标函数值的次数.
15 %
16 %
17     n0=size(initial,1);
18     X = [initial;zeros((nIter-n0), nDim)];
19     y = zeros(nIter, 1);
20     e = zeros(nIter, 1);
21     nEval = zeros(nIter, 1);
22
23
24
25     for n = (n0+1):nIter
26         f = @(XNew)efe(XNew, fpdf, X, y, n);
27         [X(n, :), y(n), e(n), nEval(n)] = fmin(f, X(1:(n - ...
28             1), :));
29         fprintf('n = %d\n', n);
30     end
31 end
32 function [e, yNew] = efe(XNew, fpdf, X, y, n)
33     [nNew, nDim] = size(XNew);
34     S=X(1:(n - 1), :);
35     Y=zeros(n-1,1);
36     for i=1:(n-1)
37         Y(i)=fpdf(S(i,:));
38     end
39     f0=@(XNew)dace(XNew,S,Y);
40     x0=[0.2,0.2];
41     lb = [0, 0];
42     ub = [1, 1];
43     opt = optimset('tolfun', 1e-3, 'tolx', 1e-3, 'display', ...

```

```

        'off');
44     [f, f0min, f1, f2] = fminsearchcon( ...
45         f0, x0, lb, ub, [], [], [], opt);
46     yNew=dace(XNew,S,Y)-f0min;
47
48
49     exist=X(1:(n - 1), :);
50     matrix=[XNew;exist];
51     d=pdist2(matrix,matrix);
52     [m,t]=size(d);
53     dnew=[];
54     for i=1:m
55         dnew=[dnew;d(i, [1:i-1 i+1:t])];
56
57     end
58
59     de=pdist2(exist,exist);
60     [s,t]=size(de);
61     dd=[];
62     for i=1:s
63         dd=[dd;de(i, [1:i-1 i+1:t])];
64
65     end
66     %
67     hn=min(min(dd))/(n^(1/(nDim+4)));
68     dnew=exp(-dnew/hn);
69
70     d1=sum(dnew(1,:));
71     he=log(d1);
72
73     ce=log(yNew);
74     c2=log(n*(hn)^nDim);
75     kl=he-ce-c2;
76     e=kl;
77
78     end

```

## A.4 响应曲面函数KL-nn点的R代码

```

1  library(mcmc)
2  # 求当前点与已有点矩阵的距离.
3  fastpdist<-function(existmatrix,tempoint)
4  {
5    if( is.vector(existmatrix)){
6      diff<-existmatrix-tempoint
7      fastpdist<- sqrt(diff%*%diff)
8    }else{
9      diff<-existmatrix-matrix(tempoint,nrow=nrow(existmatrix),
10      ncol=length(tempoint),byrow = TRUE)
11
12      fastpdist<- sqrt(apply(diff*diff,1,sum))
13    }
14    return(fastpdist)
15  }
16  # 根据KL-nn准则求一个点的得分函数.
17  obj<-function(tempoint,j)
18  {
19    x<-klpoints[1:j-1,]
20    xnew<-rbind(x,tempoint)
21    d<-matrix(0.0,j,1)
22    for(i in 1:j)
23    {
24      existmatrix<-xnew[-i,]
25      tem<-xnew[i,]
26      dist<-fastpdist(existmatrix,tem)
27      orderid<-order(dist)
28      d[i]<-dist[orderid[1]]
29    }
30
31    he<-sum(log(d))*p/j
32    #logf表示目标函数.
33    F=matrix(0.0,j,1)
34    for(i in 1:j)
35    {

```

```

36     F[i,]=logf(xnew[i,])
37   }
38   ce<-(sum(F))/j
39
40   e<-he+ce+log((j-1)*pi)
41   return(e)
42 }
43
44
45 #生成具有n个KL-nn的函数, N为局部mcmc马氏链的长度.
46 kl<-function(N,n){
47   out <- metrop(logf, rep(0, p), 1000)
48   out <- metrop(out, scale = 0.9)
49   out$accept
50   out <- metrop(out, nbatch = N)
51   x=out$batch
52   klpoints=matrix(0.0,n,p)
53   #生成x1
54   logf_ini<-matrix(0.0,N,1)
55
56   for(t in 1:N)
57   {
58     logf_ini[t]<-logf(x[t,])
59   }
60   klpoints[1,]<-x[which.max(logf_ini),]
61
62   #生成x2,x3,...
63
64   for (j in 2:n){
65     index=c(1:(j-1))
66     id=sample(index,1)
67     x0=klpoints[id,]
68     out <- metrop(logf, x0, 1000)
69     out <- metrop(out, scale = 2.5)
70     out <- metrop(out, nbatch = 50)
71     xcand=out$batch
72     func=matrix(0.0,50,1)
73     for (m in 1:50){
74       func[m]=obj(xcand[m,],j)

```



```
75
76     }
77     klpoints[j,]=xcand[which.max(func),]
78
79 }
80 x=klpoints;
81 return(x)
82 }
```