Introduction to Design and Analysis of Computer Experiments

Hongquan Xu
Department of Statistics and Data Science
UCLA

What are computer experiments?

- Computer experiments are increasingly being used to explore the behavior of complex physical systems.
- A computer model is a large computer code that implements a complex mathematical model of a physical process.
  e.g., simultaneous differential solver, finite element analysis computational fluid dynamics.

References:

- Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical Science*, **4**, 409–423.
- Wu, C. F. J. and Hamada, M. S. (2021). Experiments: Planning, Analysis and Optimization. Chapter 14.

Computer Experiments

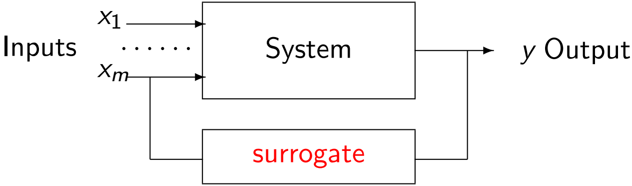


Figure 1: Computer experiment

- Computer model: (complex, expensive to compute)

$$y(\boldsymbol{x}) = f(\boldsymbol{x}) = f(x_1, \ldots, x_m)$$

- Surrogate: (fast to compute)

$$y(\boldsymbol{x}) = \hat{f}(\boldsymbol{x}) = \hat{f}(x_1, \ldots, x_m)$$

Characteristics of computer experiments

- Mostly deterministic (lack of random error)
- May take hours or even days to produce a single output
- Many input variables
- The performance of the predictor depends upon the choice of the training data (design).

Principles in traditional DOE are irrelevant

- Replication
- Blocking
- Randomization

Modeling Computer Experiments: Kriging

For $\boldsymbol{x} = (x_1, \ldots, x_m) \in \mathcal{R}^m$, treat the deterministic response $y(\boldsymbol{x})$ as a realization of a stochastic process

$$Y(\boldsymbol{x}) = \sum_{j=1}^{k} \beta_j f_j(\boldsymbol{x}) + Z(\boldsymbol{x}),$$

where $f_j(\boldsymbol{x})$ are known functions, β_j are unknown parameters and $Z(\cdot)$ is a **Gaussian process** with mean 0 and covariance

$$\text{cov}\left(Z(\boldsymbol{w}), Z(\boldsymbol{x})\right) = \sigma^2 R(\boldsymbol{w}, \boldsymbol{x}).$$

- This is the Kriging model used in spatial statistics.
- Also called Gaussian process model in Machine Learning.

R packages: `DiceKriging`, `kergp`, etc.

Prediction

Given a design $D = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ and data $y_D = (y(\boldsymbol{x}_1), \ldots, y(\boldsymbol{x}_n))^T$.
Consider the linear predictor

$$\hat{y}(\boldsymbol{x}) = c(\boldsymbol{x})^T y_D.$$

Frequentists replace y_D by the random vector
$Y_D = \{Y(\boldsymbol{x}_1), \ldots, Y(\boldsymbol{x}_n)\}^T$, and compute the MSE.
The **Best Linear Unbiased Predictor** (BLUP): choose $c(x)$ to minimize

$$\text{MSE}\,[\hat{y}(\boldsymbol{x})] = E[c(\boldsymbol{x})^T Y_D - Y(\boldsymbol{x})]^2$$

subject to

$$E[\hat{y}(\boldsymbol{x})] = E[c(\boldsymbol{x})^T Y_D] = E[Y(\boldsymbol{x})]$$

Notation

Kriging model:

$$Y(\boldsymbol{x}) = f(\boldsymbol{x})^T \beta + Z(\boldsymbol{x}),$$

where

$$f(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))^T,$$

$$\beta = (\beta_1, \ldots, \beta_k)^T$$

$$\mathrm{cov}\,(Z(\boldsymbol{w}), Z(\boldsymbol{x})) = \sigma^2 R(\boldsymbol{w}, \boldsymbol{x}).$$

In matrix form:

$$Y_D = F\beta + Z,\ \mathrm{cov}\,(Z) = \sigma^2 R$$

$$F = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n))^T = (f_j(\boldsymbol{x}_i))_{n \times k}$$

$$R = (R(\boldsymbol{x}_i, \boldsymbol{x}_j))_{n \times n}$$

$$r(\boldsymbol{x}) = (R(\boldsymbol{x}_1, \boldsymbol{x}), \ldots, R(\boldsymbol{x}_n, \boldsymbol{x}))^T$$

GLS and BLUP

The generalized LS estimate and BLUP are

$$\hat{\beta} = (F^T R^{-1} F)^{-1} F^T R^{-1} Y_D$$

$$\hat{y}(\boldsymbol{x}) = f(\boldsymbol{x})^T \hat{\beta} + r(\boldsymbol{x})^T R^{-1}(Y_D - F\hat{\beta})$$

$$MSE[\hat{y}(\boldsymbol{x})] = \sigma^2 \left[1 - (f(\boldsymbol{x})^T r(\boldsymbol{x})^T) \begin{pmatrix} 0 & F^T \\ F & R \end{pmatrix}^{-1} \begin{pmatrix} f(\boldsymbol{x}) \\ r(\boldsymbol{x}) \end{pmatrix} \right]$$

The GP interpolates the observed data: for any $\boldsymbol{x}_i \in S$,

$$\hat{y}(\boldsymbol{x}_i) = y(\boldsymbol{x}_i) \text{ and } MSE(\hat{y}(\boldsymbol{x}_i)) = 0.$$

Correlation Functions

The correlation $R(\boldsymbol{w}, \boldsymbol{x})$ has to be specified. Commonly used functions:

$$
\begin{aligned}
R(\boldsymbol{w}, \boldsymbol{x}) &= \prod \exp(-|w_j - x_j|^{p_j}/\theta_j), \; 0 < p_j \leq 2, \\
R(\boldsymbol{w}, \boldsymbol{x}) &= \prod K(|w_j - x_j|; \theta_j)
\end{aligned}
$$

where $K()$ is a correlation function such as Matérn correlation function with parameter $\nu = 5/2$.

$$
K(h;\theta) = \left(1 + \frac{\sqrt{5}h}{\theta} + \frac{5h^2}{3\theta^2}\right) \exp\left(-\frac{\sqrt{5}h}{\theta}\right).
$$

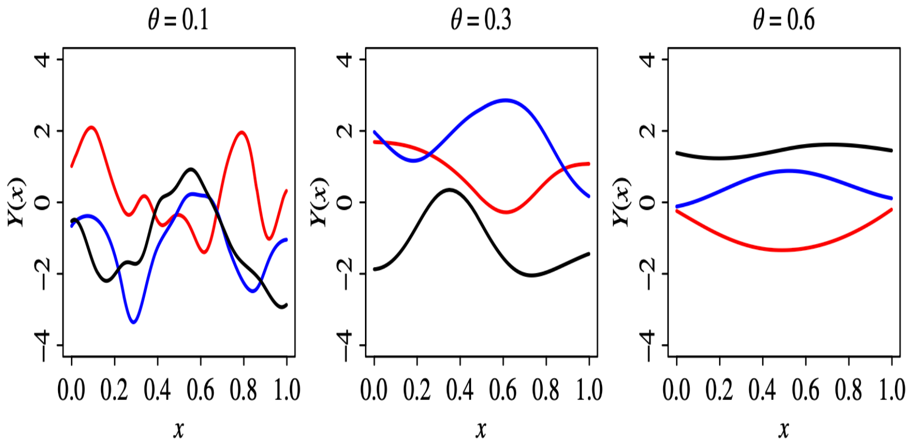
The correlation parameters (e.g., θ_j, p_j) need to be specified or estimated (by MLE or cross validation)

Given the correlation parameters, the MLEs are

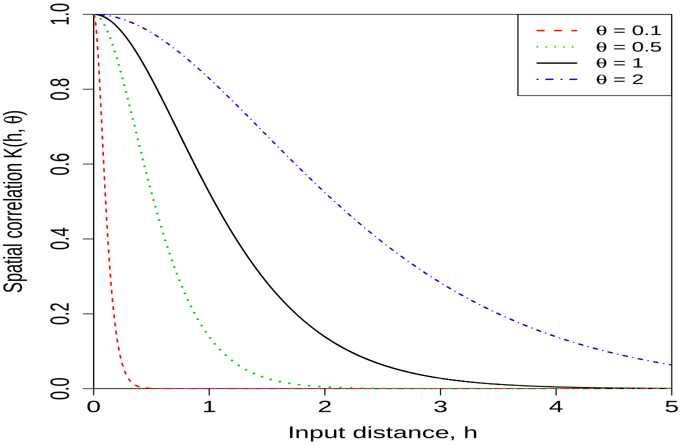
$$
\hat{\beta} = (F^T R^{-1} F)^{-1} F^T R^{-1} Y_D = \text{GLS estimate}
$$

$$
\hat{\sigma}^2 = \frac{1}{n}(Y_D - F\hat{\beta})' R^{-1} (Y_D - F\hat{\beta})
$$

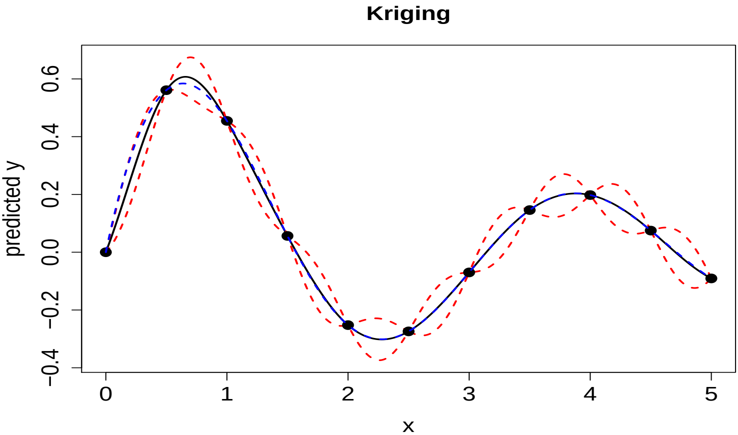
Three Sample Paths from a GP with Gaussian Correlation



Examples of Matérn $\nu = 5/2$ correlation functions

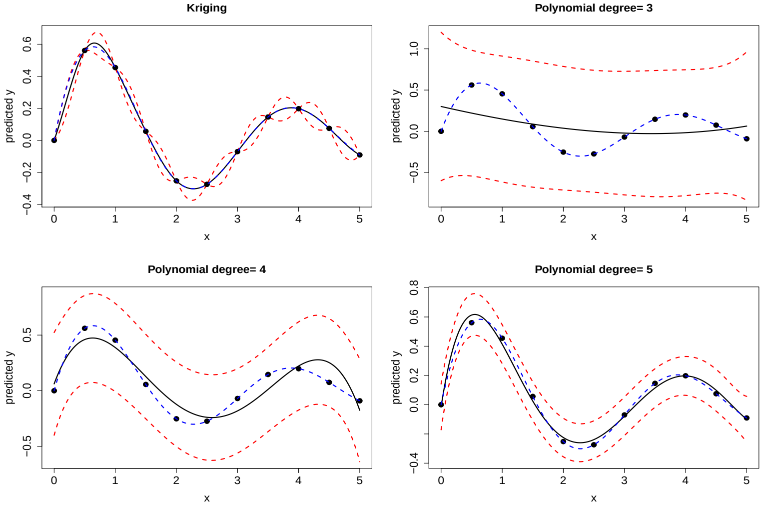


A toy example: Kriging



- Data: $n = 11$; Function: $y = \sin(2x)/(1+x)$;
- Kriging: $Y = \mu + Z(x)$.
- Black (solid) - prediction, red - 95% prediction intervals, blue - true.

A toy example: Kriging vs Polynomial models



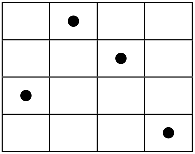
Data: $n = 11$; Function: $y = \sin(2x)/(1+x)$; Kriging: $Y = \mu + Z(x)$.

Designs for Computer Experiments

- Constructing a "good" design is crucial for the success of a computer experiment.
- A "good" design should be space-filling (i.e., cover as much space as possible), and have good projection properties.
  - Latin hypercube designs (LHD) [McKay et al. (1979)]
  - Orthogonal Array-based designs [Owen (1992), Tang (1993), He and Tang (2013, 2014)]
  - Maximin and minimax distance designs [Johnson et al. (1990)]
  - Maximum projection designs [Joseph et al. (2015)]
  - Uniform designs [Fang et al. (2000, 2006)]
  - Uniform projection designs [Sun et al. (2019)]
- Optimality criteria: maximin distance, minimax distance, column-orthogonality, uniformity (discrepancy) etc.
- R packages: `lhs`, `LHD`, `SLHD`, `UniDOE`, `MaxPro`, `UniPro`, etc.

Latin Hypercube Design (LHD) - Introduction

- Mckay, Beckman and Conover (1979).
- An example: $p = 2, n = 4$:

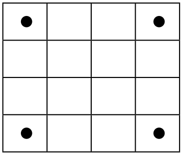


$$D = \begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 3 & 3 \\ 4 & 1 \end{bmatrix}$$

- Each row and column has one and only one point.
- Each factor has n levels.

Factorial Design 2^2

- On the other hand, a factorial 2^2 design is constructed as



$$D = \begin{bmatrix} 1 & 1 \\ 1 & 4 \\ 4 & 1 \\ 4 & 4 \end{bmatrix}$$

- If x_1 (or x_2) is not significant, replication is wasted for computer experiments.
- "Effect sparsity" principle: only a few factors are expected to be important.

Latin Hypercube Design (LHD)

Definition 1

A Latin hypercube design (LHD) with n runs and p inputs variables, denoted by LHD(n, p), is an $n \times p$ matrix, in which each column is a random permutation of $\{1, 2, \ldots, n\}$.

- How to construct an LHD(n, p)?

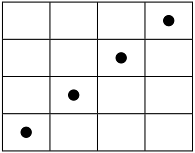
Step1 Randomly permute $\{1, 2, \ldots, n\}$ for each $x_1, \ldots, x_p$.

$$D = \begin{bmatrix} x_1 & x_2 & x_3 \\ 1 & 2 & 7 \\ 2 & n & 9 \\ 3 & 5 & 1 \\ \vdots & \vdots & \vdots \\ n & 3 & 2 \end{bmatrix}$$

Step2 $D' \leftarrow \frac{D-0.5}{n}$, where $D \in \{1, 2, \ldots, n\}^p$. Thus, $D' \in [0, 1]^p$.

Optimal LHD(n, p)

- Thus, there are $(n!)^{p-1}$ LHDs.
- Not all of them are good. For example,



$$D = \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 3 & 3 \\ 4 & 4 \end{bmatrix}$$

- This design is perfectly correlated and not space-filling.
- Finding an "optimal" LHD is a challenge.

Orthogonal Array-based LHD (OALHD) - Introduction

- Owen (1992) and Tang (1993).
- **LHD**: one-dimensional balancing property but two and higher dimensional projections can be very bad.
- Orthogonal Array (OA) of strength 2 has two dimensional balancing property, so use it to generate an LHD.
- Fractional factorial designs of resolution R are OAs of strength $t = R - 1$.
- Strong Orthogonal Arrays (He and Tang 2013) enjoys better projection properties.
- Tian and Xu (2022, *Biometrika*) introduced a **minimum-aberration type space-filling criterion** to rank General Strong Orthogonal Arrays (including OAs and OALHDs).

Orthogonal Array-based LHD (OALHD) - Example

- For example, $n = 9, p = 2$,

OA

$$\begin{bmatrix} x_1 & x_2 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 2 & 1 \\ 2 & 2 \\ 2 & 3 \\ 3 & 1 \\ 3 & 2 \\ 3 & 3 \end{bmatrix}$$

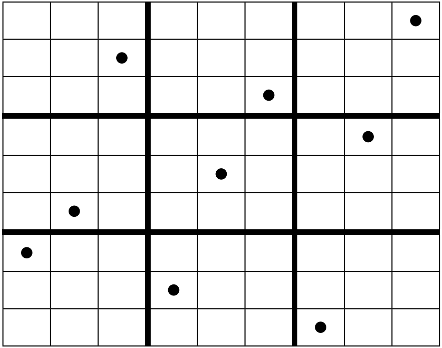
$$\begin{aligned} 1 &\to \{1,2,3\} \to \{3,2,1\} \\ 2 &\to \{4,5,6\} \to \{4,5,6\} \\ 3 &\to \{7,8,9\} \to \{8,7,9\} \end{aligned}$$

OA-LHD

$$\begin{bmatrix} x_1 & x_2 \\ 1 & 3 \\ 2 & 4 \\ 3 & 8 \\ 4 & 2 \\ 5 & 5 \\ 6 & 7 \\ 7 & 1 \\ 8 & 6 \\ 9 & 9 \end{bmatrix}$$

- Level expansion is not unique.
- Level collapsing (inverse map) is unique: $\lfloor (x+2)/3 \rfloor$.

Orthogonal Array-based LHD (OA-LHD) - Example



OA-based Latin Hypercube Design (OALHD) - Properties

- Monte Carlo Method: $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ is a random sample.
- To approximate the population mean $E[f(\boldsymbol{X})]$ by the sample mean

$$\overline{f(\boldsymbol{X})} = \frac{1}{n} \sum_{i=1}^{n} f(\boldsymbol{X}_i)$$

- Latin hypercube sampling (Stein 1987; Owen 1992) has a smaller variance than a simple random sampling.
- OA-based Latin hypercube sampling (Tang 1993) has an even smaller variance than Latin hypercube sampling

$$V_{oalhs}(\overline{f(\boldsymbol{X})}) < V_{lhs}(\overline{f(\boldsymbol{X})}) < V_{srs}(\overline{f(\boldsymbol{X})})$$

Designs Based on Measures of Distance

- Johnson, Moore, and Ylvisaker (1990).
- Let $\rho(\cdot,\cdot)$ be a metric.

$$\begin{aligned}
&\rho(\boldsymbol{x}_1,\boldsymbol{x}_2)=\rho(\boldsymbol{x}_2,\boldsymbol{x}_1)\\
&\rho(\boldsymbol{x}_1,\boldsymbol{x}_2)\geq 0\\
&\rho(\boldsymbol{x}_1,\boldsymbol{x}_2)=0 \Leftrightarrow \boldsymbol{x}_1=\boldsymbol{x}_2\\
&\rho(\boldsymbol{x}_1,\boldsymbol{x}_2)\leq \rho(\boldsymbol{x}_1,\boldsymbol{x}_3)+\rho(\boldsymbol{x}_3,\boldsymbol{x}_2)
\end{aligned}$$

- For example,

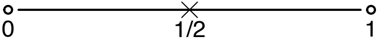
$$\rho(\boldsymbol{x},\boldsymbol{w})=\{\sum_{j=1}^{p}|x_j-w_j|^k\}^{1/k}$$

- $k=1$: rectangular distance
- $k=2$: Eulidean distance

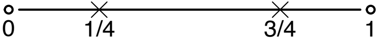
- Let $\rho(\boldsymbol{x}, D) = \min_{\boldsymbol{x}_i \in D} \rho(\boldsymbol{x}, \boldsymbol{x}_i)$ be the minimum distance to the design.
- Let $\chi = [0, 1]^p$ and
$$h = \max_{\boldsymbol{x} \in \chi} \rho(\boldsymbol{x}, D)$$
be the maximum distance in χ.
- h is called *fill distance*.
  - the largest gap
  - the radius of the largest ball that can be placed in χ which does not contain any point in D.
- Thus, find a D to minimize h. That is,
$$\min_{D} \max_{\boldsymbol{x} \in \chi} \rho(\boldsymbol{x}, D)$$
$\rightarrow$ miniMax distance design (mM).

miniMax Distance Design (mM Design) - Examples

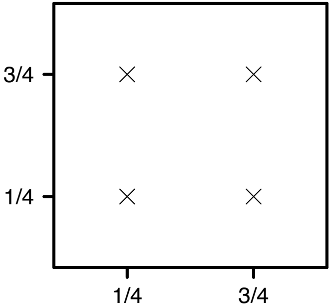
- $p = 1, n = 1$:



- $p = 1, n = 2$:



- $p = 2, n = 4$:



miniMax Distance Design (mM Design) - Properties

- mM designs ensure that all points in χ are not too far from the design.
- Consider the owner of a petroleum corporation who wants to open some gas stations. mM design ensures that no customer is too far from one of the company's gas stations.

Maximin Distance Design (Mm Design) - Introduction

- The minimum distance between any two points in D is

$$2q = \min_{\boldsymbol{x}_1, \boldsymbol{x}_2 \in D} \rho(\boldsymbol{x}_1, \boldsymbol{x}_2),$$

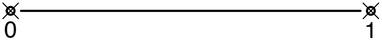
  where q is the separation distance or packing radius - the radius of the largest ball that can be placed around every design point such that no two balls overlap.
- A large q ensures numerical stability in Kriging.
- A large q tends to decrease h.
- Thus, find a D to maximize $2q$. That is,

$$\max_{D} \min_{\boldsymbol{x}_1, \boldsymbol{x}_2 \in D} \rho(\boldsymbol{x}_1, \boldsymbol{x}_2)$$

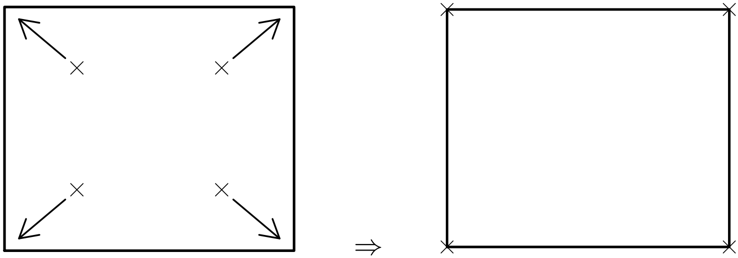
  $\rightarrow$ Maximin distance design (Mm).

Maximin Distance Design (Mm Design) - Example

- $p = 1, n = 2$:



- $p = 2, n = 4$:

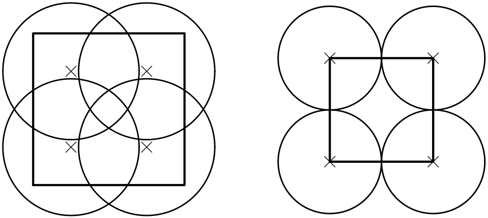


Maximin Distance Design (Mm Design) - Properties

- Mm designs ensure that the points in D are as far apart from each other as possible.
- Gas station example: Mm design ensures that no two gas stations are too close to each other. It minimizes the competition from each other by locating the stations as far apart as possible.
- Saturated $OA(n, 2^p)$'s are maximin distance designs when $p = n - 1$ (Xu 1999).

Summary

- In mathematics,
  - covering problems: cover $\mathbb{R}^p$ with spheres. $\rightarrow$ miniMax
  - packing problems: pack spheres in $\mathbb{R}^p$. $\rightarrow$ Maximin
- Experimental design problem is different because we need to cover $[0,1]^p$ or pack in $[0,1]^p$. $\rightarrow$ This introduces boundary effects.



(a) Minimum radius. (b) Maximum radius.

Summary

- Computationally,
  - miniMax:

$$\min_{D} \max_{x \in \chi} \min_{\mathbf{x}_i \in D} \rho(\mathbf{x}, \mathbf{x}_i) \rightarrow \text{very hard}$$

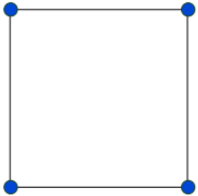
  - Maximin:

$$\max_{D} \min_{\mathbf{x}_1, \mathbf{x}_2 \in D} \rho(\mathbf{x}_1, \mathbf{x}_2) \rightarrow \text{hard, but easier than miniMax}$$

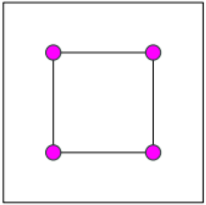
- Mak and Joseph (2017): a new hybrid algorithm combining particle swarm optimization and clustering for generating minimax designs on any convex and bounded design space.
- R package: `minimaxdesign`

Maximin Latin Hypercube Design (MmLHD) - Introduction

- Morris and Mitchell (1995).
- Advantage of mM/Mm designs: run size flexibility and "optimal".
- Disadvantage of mM/Mm designs: projections are poor.

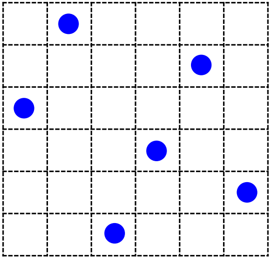


(a) Maximin design



(b) Minimax design.

- LHD: good 1-dimensional projections, but can be poor in terms of space-filling in higher dimensions.



Maximin Latin Hypercube Design (MmLHD) - Introduction

- Combine these two ideas $\rightarrow$ Maximin Latin hypercube designs (MmLHD)

$$\max_{D} \min_{\boldsymbol{x}_1, \boldsymbol{x}_2 \in D} \rho(\boldsymbol{x}_1, \boldsymbol{x}_2),$$

  where D is an LHD.

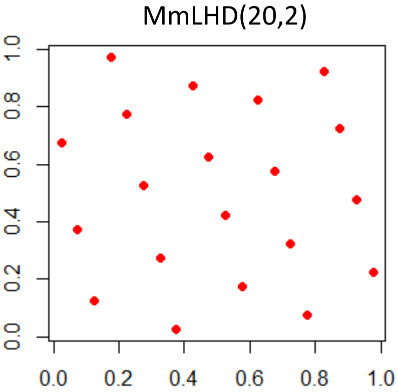
- It is the same as

$$\min_{D} \left(\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{\rho^k(\boldsymbol{x}_i, \boldsymbol{x}_j)} \right)^{1/k}$$

  and $k \rightarrow \infty$, where D is an LHD. Typically take $k = 15$ or 50.

- R package SLHD constructed maximin LHDs via simulated annealing algorithm (Morris and Mitchell 1995).

Maximin Latin Hypercube Design (MmLHD) - Examples



Maximum Projection Design (MaxPro) - Introduction

- Joseph, Gul, and Ba (2015).
- MmLHDs only ensure good space-fillingness in p dimensions and uniform projections in a single dimension, but projection properties in $2, 3, \ldots, p-1$ dimensions may not be good.
- In practice, we often have $1 <$ the number of important factors $< p$.

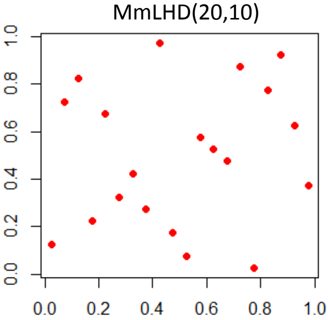


Figure 4: Projection in two dimensions for a 10 dimensional MmLHD.

Maximum Projection Design (MaxPro)

- How to calculate distances in a projected subspace? **Answer:** Put weights of 1 on the defining factors and 0 on the remaining factors.
- **Weighted Euclidean Distance**:

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{\theta}) = \{\sum_{l=1}^{p} \theta_l (x_{il} - x_{jl})^2\}^{1/2}.$$

  Let $0 \leq \theta_l \leq 1$ be the weight assigned to the factor l and let $\sum_{l=1}^{p} \theta_l = 1$.
- Then the MmLHD criterion can be modified to

$$\min_D \phi_k(D; \boldsymbol{\theta}) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{d^k(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{\theta})},$$

  where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{p-1})'$ and $\theta_p = 1 - \sum_{l=1}^{p-1} \theta_l$.

Maximum Projection Design (MaxPro)

- In practice, we often have no idea about the importance of the factors before the experiment.
- Assigning non-informative prior:

$$p(\boldsymbol{\theta}) = \frac{1}{(p-1)!}, \boldsymbol{\theta} \in S_{p-1},$$

where $S_{p-1} = \{\boldsymbol{\theta} : \theta_1, \ldots, \theta_{p-1} \geq 0, \sum_{i=1}^{p-1} \theta_i \leq 1\}$.

- The design criterion becomes

$$\min_D \mathbb{E}\{\phi_k(D; \boldsymbol{\theta})\} = \int_{S_{p-1}} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{d^k(x_i, x_j; \boldsymbol{\theta})} p(\boldsymbol{\theta}) d\boldsymbol{\theta}.$$

Maximum Projection Design (MaxPro)

Theorem 2

If $k = 2p$, then under the given prior

$$\mathbb{E}\{\phi_k(D;\boldsymbol{\theta})\} = \frac{1}{\{(p-1)!\}^2} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{\prod_{l=1}^{p}(x_{il} - x_{jl})^2}.$$

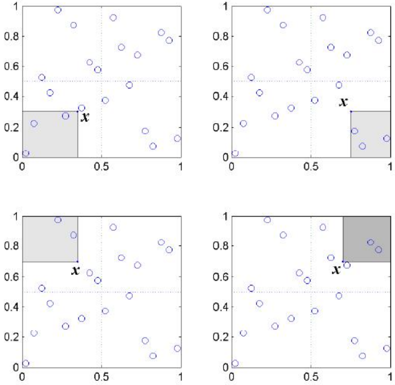
- Therefore, the *MaxPro Criterion* is

$$\min_{D} \psi(D) = \left(\frac{1}{\binom{n}{2}} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{\prod_{l=1}^{p}(x_{il} - x_{jl})^2} \right)^{1/p}.$$

- **Maximum projection (MaxPro) design**: maximizes space-filling properties on projections to all possible subsets of factors.
- R package `MaxPro`: computed at a cost no more than a design criterion that ignores projection properties.

Uniform designs

Idea: choose design points from the design region with empirical distribution as “uniform” as possible (Fang et al, 2000, 2006).



For an $n \times m$ design D over $[0,1]^m$,

$$Disc(D) = \left\{ \int_{[0,1]^m} \left| Vol(J(a_x, x)) - \frac{N(D \cap J(a_x, x))}{n} \right|^2 dx \right\}^{1/2}.$$

The (squared) centered L_2-discrepancy is defined by

$$CD(D) = \left\{ \sum_{u \subseteq \{1:m\}} |Disc(D_u)|^2 \right\},$$

where u is a subset of $\{1, 2, \ldots, m\}$ and D_u is the projected design of D onto dimensions indexed by the elements of u.

Uniform Designs and Centered L_2-Discrepancy

The centered L_2-discrepancy has an analytical expression. For $D = (z_{ik})$ over $[0,1]^m$:

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

- For a design $D = (x_{ij})$ with s levels $1, 2, \ldots, s$, use $z_{ij} = (x_{ij} - 0.5)/s$.
- Uniform designs may have poor projections in lower dimensional spaces (Zhou et al., 2013).
- There are other types of discrepancy measures, but they have similar formulas and problems.
- R package: UniDOE

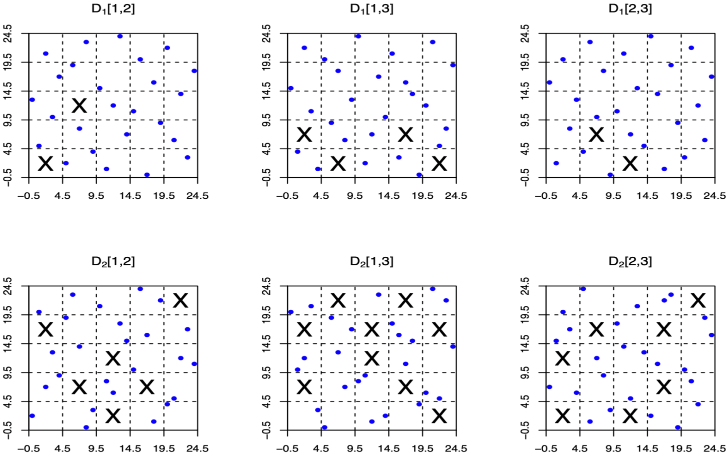
Uniform Projection Designs

- Focus on 2-dim projection uniformity
- *Uniform projection criterion* (Sun, Wang and Xu, 2019, Annals of Statistics)
$$\phi(D) = \frac{2}{m(m-1)} \sum_{|u|=2} CD(D_u), \tag{1}$$
where the summation is over all possible 2-column subsets u.
- A design achieving the minimum $\phi(D)$ value is a **uniform projection design** (UPD).
- R package UniPro and Meta4Design

Why we need a new criterion? Four 25×3 LHDs

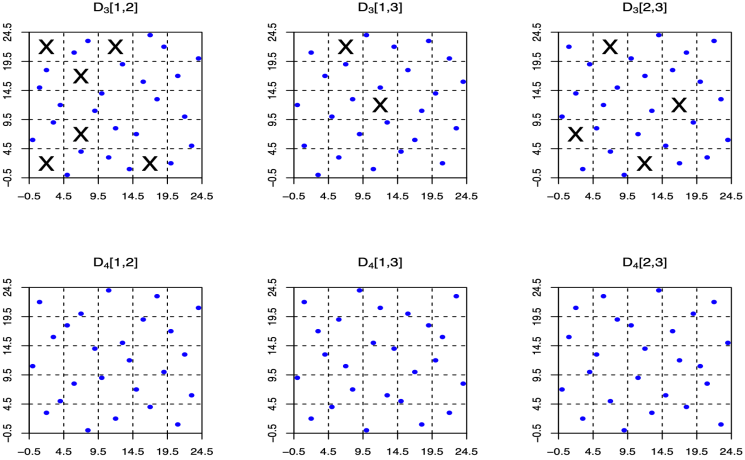
Uniform D_1			Maximin D_2			MaxPro D_3			UPD D_4		
18	16	14	0	2	20	0	6	12	2	3	2
19	9	0	1	20	10	1	15	5	4	5	13
11	1	2	2	7	12	2	18	21	0	11	9
16	20	3	3	13	21	3	9	0	3	16	17
20	22	12	4	9	3	4	12	17	1	22	22
14	7	10	5	19	0	5	0	10	8	0	7
4	17	1	6	23	19	6	21	3	6	8	19
12	12	7	7	14	13	7	4	19	9	14	24
10	15	24	8	0	7	8	23	13	5	18	4
22	14	5	9	3	17	9	11	7	7	20	11
2	21	22	10	21	8	10	14	24	12	2	21
15	11	21	11	8	9	11	3	1	10	9	0
1	5	4	12	6	1	12	8	15	14	12	14
3	10	11	13	18	23	13	19	9	13	15	6
23	3	8	14	15	2	14	1	22	11	24	15
0	13	15	15	10	18	15	7	4	17	4	10
8	23	6	16	24	16	16	16	18	15	7	5
7	8	18	17	16	11	17	24	6	19	10	18
9	4	13	18	1	15	18	13	11	16	19	20
6	19	9	19	22	4	19	22	23	18	23	1
24	18	19	20	4	6	20	2	14	21	1	16
21	6	23	21	5	24	21	17	2	23	6	23
13	24	17	22	12	5	22	10	20	22	13	3
17	0	16	23	17	22	23	5	8	20	17	12
5	2	20	24	11	14	24	20	16	24	21	8

Bivariate projections of Uniform D_1 and Maximin D_2



Note: ‘X’ means that there are no points in the grid.

Bivariate projections of MaxPro D_3 and UPD D_4



Note: 'X' means that there are no points in the grid.

Some Theoretical Results

Theorem 3

For a balanced (n, s^m) design D and any $2 \leq k \leq m$,

$$\frac{1}{\binom{m}{k}} \sum_{|u|=k} \phi(D_u) = \phi(D),$$

where D_u is the projected design onto k factors indexed by u.

- UPDs have good space-filling properties not only in two dimensions, but also in all dimensions.

Some Theoretical Results

Theorem 4

For a balanced (n, s^m) design $D = (x_{ik})$,

$$\phi(D) = \frac{g(D)}{4m(m-1)n^2s^2} + C(m, s), \tag{2}$$

where

$$g(D) = \sum_{i=1}^{n}\sum_{j=1}^{n} d_1^2(x_i, x_j) - \frac{2}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{n} d_1(x_i, x_j)\right)^2 \tag{3}$$

- $\phi(D)$ is a function of pairwise L_1-distances of the rows of D.
- $\phi(D)$ can be efficiently computed in $O(n^2m)$ operations.
- An equidistant design under the L_1-distance is a UPD (Sun, Wang, Xu 2019).

Application: Design and Modeling Comparison

- A 3-drug combination experiment on lung cancer (Al-Shyoukh et al. 2011; Xiao, Wang and Xu 2019).
- A 512-run and 8-level full factorial design to study 3 drugs.
- The response was the ATP level of the cells after the drug treatments.
- Kriging model with noise: $y(\boldsymbol{x}) = \mu + Z(\boldsymbol{x}) + \epsilon$

Table 1: Comparison of 1000×MSE for different models and designs

	Normal Cell				Cancer Cell			
	D_{512}	RD_{80}	MPD_{25}	UPD_{25}	D_{512}	RD_{80}	MPD_{25}	UPD_{25}
Kriging	0.002	0.21	0.62	0.22	0.003	0.37	1.87	0.21
NN	0.37	1.28	3.12	1.79	0.47	1.57	4.10	2.93
Polynomial	0.48	1.16	3.22	0.74	2.98	6.77	10.04	4.42

RD_{80}: Random 80-run design; MPD_{25}: MaxPro 25-run designs.

Comparison of projection properties

We compare four LHD(19, 18)'s:

1. The uniform design is from the uniform design website(UD)
2. The maximin distance design via R package SLHD (Ba, Myers and Brenneman, 2015, Technometrics).
3. The maximum projection (MaxPro) design were constructed via R package MaxPro (Joseph et al., 2015, Biometrika)
4. The uniform projection design (UPD): E_b.

We ran R commands maximinSLHD (with slice parameter $t = 1$) and MaxProLHD 100 times with default settings and chose the best designs.

Comparison of projection properties

Four criteria will be used in the comparison:

1. minimum Euclidean distance
2. maximum projection criterion (Joseph et al. 2015)

$$\psi(D) = \left\{ \frac{1}{\binom{n}{2}} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{\prod_{k=1}^{m} (x_{ik} - x_{jk})^2} \right\}^{1/m}$$

3. relative maximum centered L_2-discrepancy (CD)
4. maximum correlation ρ_{ave}.

For each k, we evaluate all $\binom{m}{k}$ projected designs and determine the worst projection with respect to four criteria.

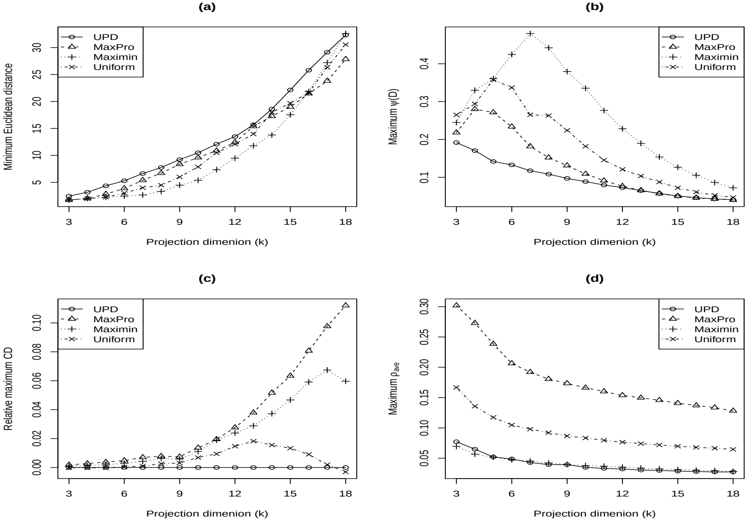


Figure 5: (a) minimum Euclidean distance (the larger the better), (b) maximum $\psi(D)$ (the smaller the better), (c) relative maximum CD (the smaller the better), and (d) maximum ρ_{ave} (the smaller the better).

Simulation: The Borehole Simulation

Consider the 8-dim borehole function

$$f(x_1, \ldots, x_8) = \frac{2\pi x_1(x_5 - x_8)}{\log(x_6/x_2)[1 + \frac{2x_3x_1}{\log(x_6/x_2)x_2^2x_7} + \frac{x_1}{x_4}]}.$$

Fit a Kriging model to approximate the borehole function

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

where $Z(\boldsymbol{x})$ is a Gaussian process with mean 0 and covariance function

$$\text{cov}\{Y(\boldsymbol{x} + \boldsymbol{h}), Y(\boldsymbol{x})\} = \sigma^2 \prod_{j=1}^{8} K(h_j; \theta_j),$$

$$K(h; \theta) = \left(1 + \frac{\sqrt{5}h}{\theta} + \frac{5h^2}{3\theta^2}\right) \exp\left(-\frac{\sqrt{5}h}{\theta}\right)$$

$K()$ is the Matérn correlation function with $\nu = 2.5$.

Simulation: Designs

Consider nine 121×8 designs:

- Three 11-level factorial designs (orthogonal arrays):
  - A regular design D
  - Level permuted regular design $D_b = D + b \pmod{11}$ (Tang and Xu, 2014)
  - Level permuted nonregular design $E_b = W(D_b)$ (Wang and Xu, 2021)
- Six LHDs (with 121 levels):
  - three LHDs obtained from the 11-level designs by rotation (Steinberg and Lin, 2006; Wang and Xu, 2021)
  - randomly generated LHD
  - maximin-distance LHD (Ba, Myers, and Brenneman, 2015)
    repeated 100 times, selected the best, each 6 secs, total 10 mins
  - maximum-projection LHD (Joseph, Gul, and Ba, 2015)
    repeated 100 times, selected the best, each 7 secs, total 12 mins

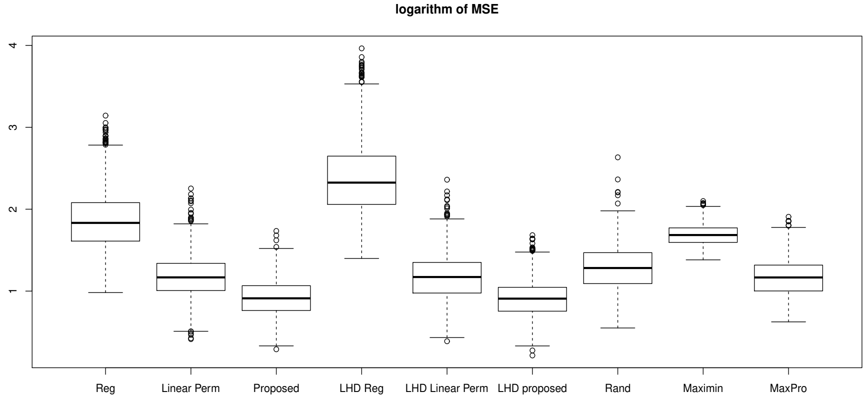
Simulation: Data generation

- Scale each variable to [0,1]: $\boldsymbol{x} = (x_1, \ldots, x_8)$ is scaled to $[0, 1]^8$
- Generate nine data sets based on the nine designs
- Build a Kriging model for each data set.
- Testing data: 5000 uniformly distributed points in $[0, 1]^8$
- MSE for the predicted responses at the testing data points are examined

$$\text{MSE} = \frac{1}{5000} \sum_{i=1}^{5000} (y_i - \hat{y}_i)^2,$$

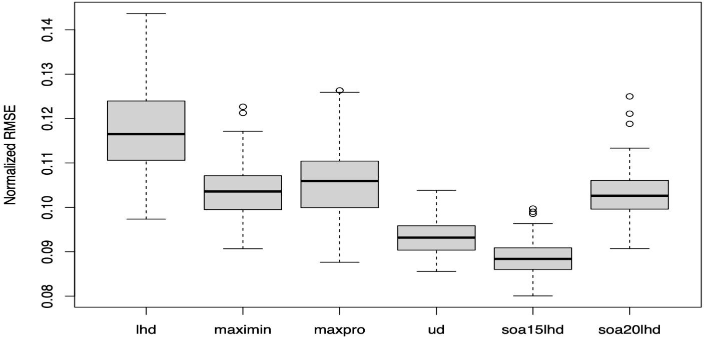
  where y_i is the true response and $\hat{y}_i$ is the predicted response from the fitted model at the ith testing data point
- The simulation process is repeated for 1000 times.

Simulation: Result (Comparing 121×8 Designs)



- Designs matter!
- Multilevel designs are as good as LHDs!

Simulation: Comparing 64×15 LHDs



- random (lhd), maximin, MaxPro, uniform (ud), and strong orthogonal arrays (soa15lhd, soa20lhd)
- Designs matter: Strong orthogonal arrays are better (Shi and Xu 2023+, JASA).

Summary on Space-Filling Designs

- Latin hypercube design:
  - Good 1-dimensional projections, but might not be space-filling in p dimensions.
- **Orthogonal array-based designs**:
  - Good space-filling properties in projections in low dimensions.
- miniMax/Maximin distance design:
  - Run size flexibility and good space-fillingness in p dimensions, but bad projections.
- Maximin Latin Hypercube design:
  - Good space-fillingness in p dimensions and uniform projections in a single dimension.
- Maximum projection design:
  - Good space-filling properties in projections to all subsets of factors.
- Uniform designs, **uniform projection designs**:
  - Better space-filling properties in projections to all subsets of factors.

Summary

- The Gaussian Process model provides an efficient framework for modeling.
  - The role of the regression part ?
  - Choice of correlation function – large vs. small correlation ?
- There are several types of space-filling designs
  - which type is the best in terms of prediction or optimization?
- Construction of space-filling designs
  - Meta-heuristic algorithms: simulated annealing, threshold accepting, genetic algorithms — flexible, but not efficient for constructing large designs
  - Theoretical methods such as good lattice point designs (Wang, Xiao and Xu 2018 AOS) — guaranteed efficiency, but not flexible
- Some new developments (Tian and Xu 2022 Biometrika, 2024 JRSSB; Shi and Xu 2023+ JASA)
  - space-filling hierarchy principle
  - minimum-aberration type space-filling criterion
  - How to construct/search space-filling designs?

Selected References

- Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical science*, **4**, 409–423.
- Wu, C. F. J. and Hamada, M. S. (2021). Experiments: Planning, Analysis and Optimization. Chapter 14.
- Johnson, M. E., Moore, L. M. and Ylvisaker, D. (1990). Minimax and maximin distance designs. *J. Statist. Plan. Infer.* **26**, 131–48.
- Joseph, V. R., Gul, E. and Ba, S. (2015). Maximum projection designs for computer experiments. *Biometrika* **102**, 371–380.
- Fang, K. T., Lin, D. K. J., Winker, P. and Zhang, Y. (2000). Uniform design: theory and application. *Technometrics* **42**, 237–248.
- Sun, F., Wang, Y. and Xu, H. (2019). Uniform Projection Designs. *Annals of Statistics*, 47, 641–661.
- Xiao, Q., Wang, L. and Xu, H. (2019). Application of Kriging Models for a Drug Combination Experiment on Lung Cancer. *Statistics in Medicine*, 38, 236–246.
- Tian, Y. and Xu, H. (2022). A Minimum Aberration Type Criterion for Selecting Space-Filling Designs. *Biometrika*, 109(2), 489–501.