UNIVERSITY OF CAMBRIDGE

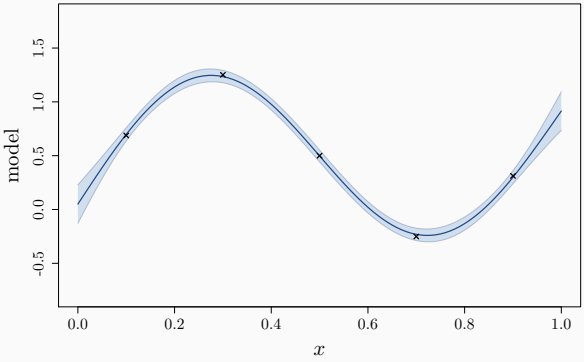
Lecture 10 – Design of Experiments

Machine Learning and the Physical World

Nicolas Durrande (Monumo) – 12th of November 2025

Introduction

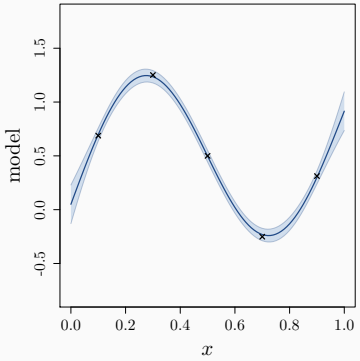
You've seen various ways to build a model from a given set of input/output tuples:



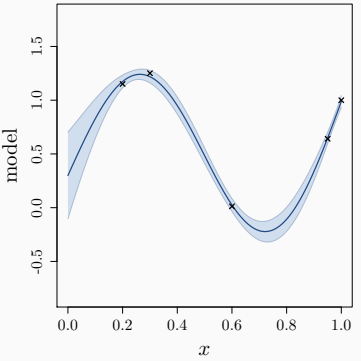
Today's question is: if the input points X can be chosen, how can we obtain the best model?

Motivating example

Same number of points but different input locations



$\text{IMSE} = 0.001$



$\text{IMSE} = 0.004$

Outline of the lecture

1. High-dimensional spaces are counter-intuitive
2. Traditional designs
3. Optimal designs for regression and GPR
4. Space-filling designs

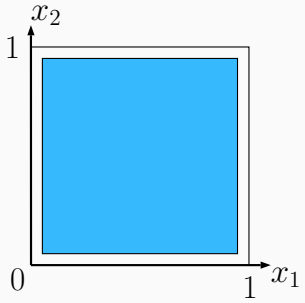
Note: I often use 1D examples, but input spaces are typically of dimension 5 to 100.

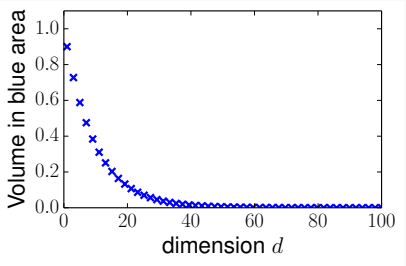
Intuition in high dimension...

Intuition is misleading in high dimension

Example 1

- Most of the volume is near the domain boundaries.
  $\rightarrow$ hypercube of size 0.9 included in the unit cube:

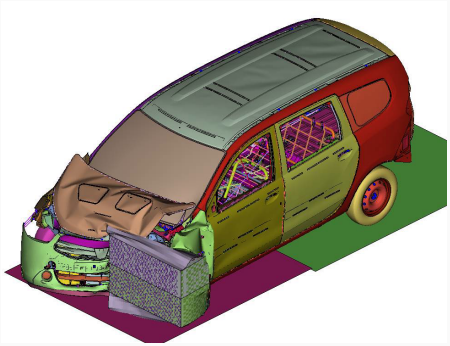




If 10,000 points are sampled uniformly in $[0, 1]^{100}$, the expected number of points in the blue area is $\sim$2.5.

Example 2

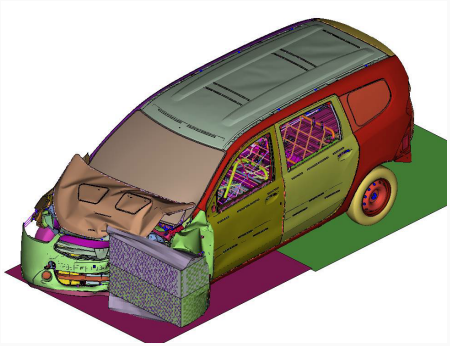
- The number of vertices of a hypercube increases very quickly with d.



If one simulation takes a minute, testing all min/max combinations of 50 input parameters requires...

Example 2

- The number of vertices of a hypercube increases very quickly with d.



If one simulation takes a minute, testing all min/max combinations of 50 input parameters requires...

$$2^{50}\text{min} \approx 1.13 \times 10^{15} \text{ minutes } \approx 2 \times 10^{9} \text{ years}$$

(twice the age of the universe!)

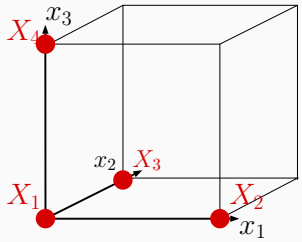
Traditional designs

One-at-a-time design

An intuitive way to check the influence of various variables is to change them one at a time.

- All variables are fixed at a reference value (0, for example).
- One variable is changed at a time to see if there is an influence.

Example



point	x_1	x_2	x_3
X_1	0	0	0
X_2	1	0	0
X_3	0	1	0
X_4	0	0	1

Pros and cons of this kind of design:

- $+$ requires only $d + 1$ observations
- $+$ is easy to interpret
- $-$ can only see linear effects:

$$m(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$$

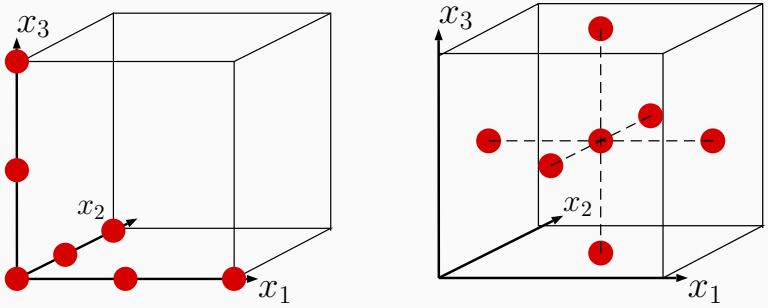
- $-$ does not cover the space

Question

How can this kind of design be adapted to estimate quadratic effects?

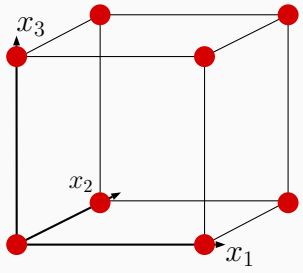
Solution

Quadratic effects can be estimated with either



We sometimes talk about *star-shaped* designs.

The principle of factorial design is to consider all combinations for $x_i \in \{0, 1\}$:



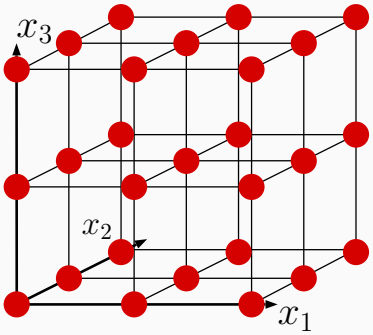
pros They allow us to get all interaction terms:

$$\beta_0 + \sum_k \beta_k x_k + \sum_{j,k} \beta_{j,k} x_j x_k + \beta_{1,2,3} x_1 x_2 x_3$$

cons The number of evaluations is unrealistic when d is large.

Factorial designs

It is also possible to build factorial designs with k levels:



This allows us to compute quadratic effects, but the number of evaluations k^d is even less realistic.

Conclusion on classical designs:

pros:

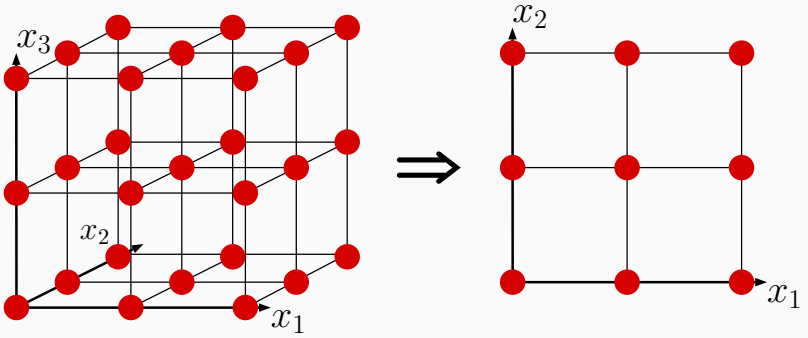
- Easy to use
- Adapted to continuous or discrete variables
- Can be combined (star + factorial, for example)
- Well suited (often optimal) for linear regression

cons:

- Number of evaluations is not flexible
- Number of evaluations too large in high dimension
- Points lie on top of each other when projected

Projection issues

Why don’t we want points to be superimposed when projected?
If one of the variables has no influence, most observations become redundant...



From 27 observations, we end up with only 9.

Optimal DoE for linear regression

Let's consider a classic linear regression problem:

$$y = X\beta + \varepsilon \qquad \text{with } \varepsilon \sim \mathcal{N}(0, \sigma^2 I_n)$$

Here X is an $n \times p$ design matrix (n observation points, p basis functions).

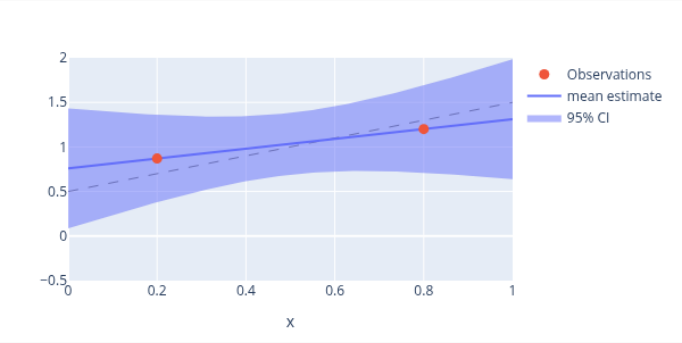
Given a vector of observations $\mathbf{y}$, the best linear unbiased estimator of β is:

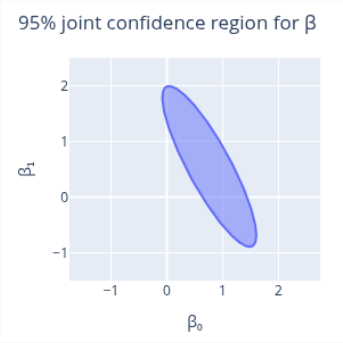
$$\hat{\beta} = (X^T X)^{-1} X^T \mathbf{y}$$

and its distribution is

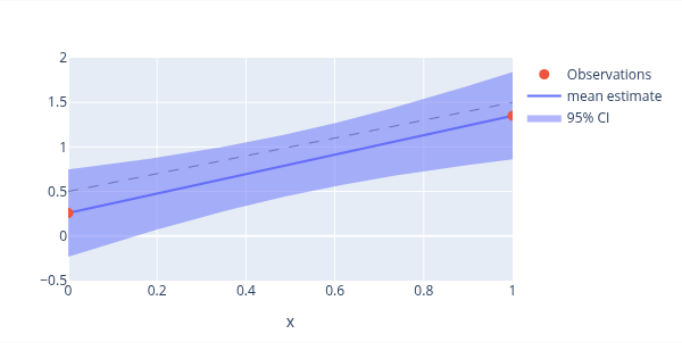
$$\hat{\beta} \sim \mathcal{N}\left(\beta, \sigma^2 (X^T X)^{-1}\right)$$

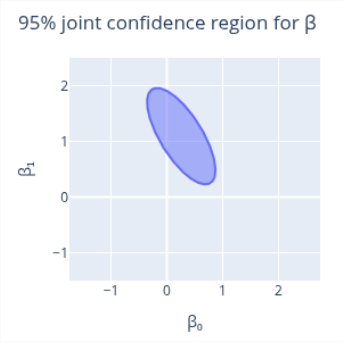
Simple example



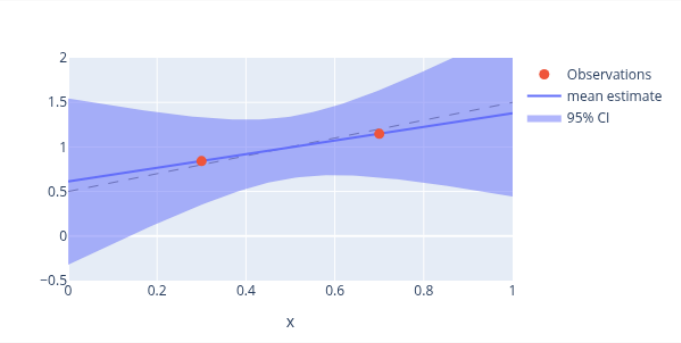


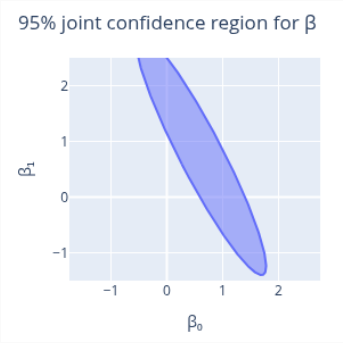
Simple example





Simple example





Let's have another look at the predicted mean and variance:

$$m(x) = B(x)(X^T X)^{-1} X^T \mathbf{y}$$
$$v(x) = \sigma^2 B(x)(X^T X)^{-1} B(x)^T$$

A striking property of the above is that the prediction variance does not depend on the observations!

Note that this is also true for the distribution of $\hat{\beta} \sim \mathcal{N}\left(\beta, \sigma^2 (X^T X)^{-1}\right)$.

$\Rightarrow$ We can choose X in order to minimise the prediction variance or the uncertainty on $\hat{\beta}$!

Exercise

We consider a linear regression model over $[0, 1]$ with one basis function $b(x) = x$ and one observation at x_1.

1. Give the expression of m and v.
2. What value of x_1 minimises the maximum of v?
3. Give the expression of the variance of $\hat{\beta}$.
4. What value of x_1 minimises it?

Various criteria for the variability of the estimate:

D-optimality

The volume of the confidence ellipsoid is minimised

$$\min_X \det(X^T X)^{-1} = \max_X \det(X^T X).$$

A-optimality

The sum of the coefficient variances is minimised

$$\min_X \operatorname{tr}(X^T X)^{-1}.$$

E-optimality

Minimises the largest eigenvalue of $(X^T X)^{-1}$; equivalently, maximises the smallest eigenvalue of $X^T X$:

$$\min_X \lambda_{\max}\big((X^T X)^{-1}\big) \quad \Longleftrightarrow \quad \max_X \lambda_{\min}(X^T X).$$

Various criteria for the prediction variance:

G-optimality

The maximum of the prediction variance is minimised

$$\min_X \max_x \ \sigma^2 \, B(x)(X^T X)^{-1} B(x)^T.$$

IMSE-optimality (or I-optimality)

The integrated variance is minimised

$$\min_X \int \sigma^2 \, B(x)(X^T X)^{-1} B(x)^T \ \mathrm{d}.$$

In practice, the optimisation of these criteria is difficult:

- large number of variables ($n \times d$)
- multimodal objective (lots of symmetries)

Some algorithms (such as Fedorov) are based on one-at-a-time point replacement:

1. Find the worst point in the design
2. Find a critical region (large variance)
3. Replace the “bad” point by a point in the critical region

Equivalence theorem (Kiefer and Wolfowitz)

The three conditions are equivalent

- A design is D-optimal
- A design is G-optimal
- The maximum prediction variance is p
  (assuming $\sigma^2 = 1$ and total mass 1 on the design points)

Knowing a lower bound allows us to define the efficiency of a DoE:

$$G_{\text{eff}} = 100 \times \sqrt{\frac{p}{\max_x \; B(x)(X^T X)^{-1} B(x)^T}}.$$

Optimal DoEs for Gaussian process regression

As previously, we can discuss two kinds of optimality:

- in the parameter estimation
- in the prediction variance

We will distinguish two cases: when the covariance parameters are known or unknown.

Known covariance parameters

A GP Z with covariance k can be decomposed as a sum of two independent GPs:

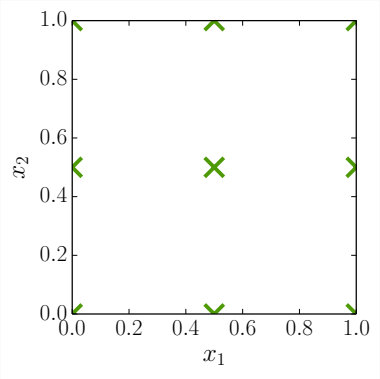
$$Z(x) = \underbrace{k(x,X)k(X,X)^{-1}Z(X)}_{Z_X(x)} + \underbrace{Z(x) - k(x,X)k(X,X)^{-1}Z(X)}_{Z_{X^\perp}(x)}$$

$$k(x,y) = \underbrace{k(x,X)k(X,X)^{-1}k(X,y)}_{k_X(x,y)} + \underbrace{k(x,y) - k_X(x,y)}_{k_{X^\perp}(x,y)}$$

In order to capture most of the variability of Z, we can:

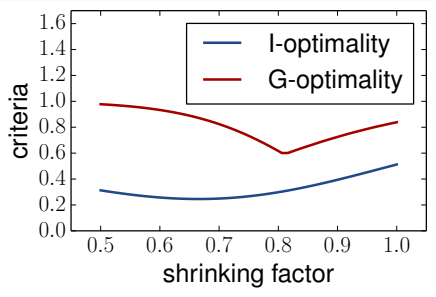
- maximise the variability of Z_X and apply previous D/A/E-optimality criteria to $k(X,X)$ instead of $B(X)^T B(X)$;
- minimise the prediction error: I/G-optimality applied to $k_{X^\perp}(x,x)$.

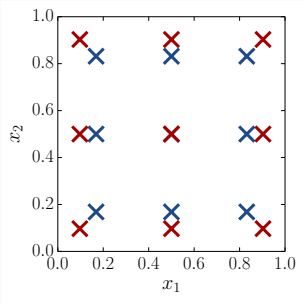
If we maximise the determinant of $k(X, X)$ for a 9-point DoE on $(0, 1)^2$, we find the following design:



However, this design is neither I-optimal nor G-optimal.

We can compute numerically the optimal shrinkage factor:





$\Rightarrow$ They all give different optimal DoEs: there is no D–G equivalence for GPR.

Unknown covariance parameters

What about **unknown** covariance parameters?
The kernel parameters can be estimated using maximum likelihood.

Can we find a design that gives a good parameter estimation of the variance and lengthscale?

- There are no strong theoretical results.
- Good estimation of the variance requires the points to be far apart.
- Good estimation of the lengthscale requires some points to be close together.

If the covariance structure itself is unknown, it is useful for the design to include a wide variety of inter-point distances.

Small recap on optimal design for GPR

- All criteria are difficult to compute.
- The optimisation problem is tricky.
- We don’t have strong theoretical results as in regression.

Good practice:

- space-filling designs such as LHDs
- optimisation inside a class of DoE

Space-filling DoE

We are now looking for model-agnostic DoE that

- provide good space coverage,
- have good projections on subspaces,
- have a flexible number of points.

How can we evaluate if a set of points fills the space?

1. Compute the distance between points

maximin the minimum distance between two points of the design should be large:

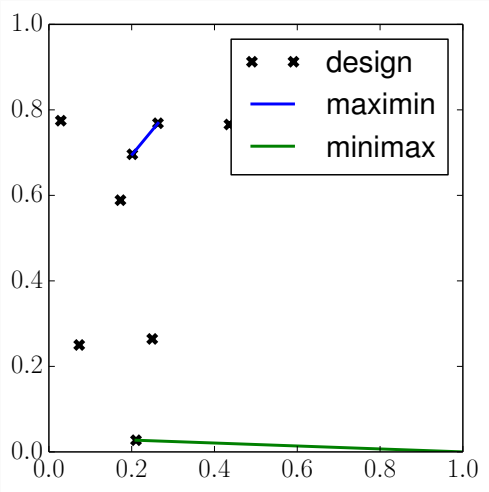
$$\text{Optimisation problem:} \quad \max_{X_1,\dots,X_n} \left[\min_{i \neq j} \operatorname{dist}(X_i, X_j) \right]$$

minimax the maximum distance between any point of the space and the closest design point should be small:

$$\text{Optimisation problem:} \quad \min_{X_1,\dots,X_n} \left(\max_{x \in D} \left[\min_{i} \operatorname{dist}(x, X_i) \right] \right)$$

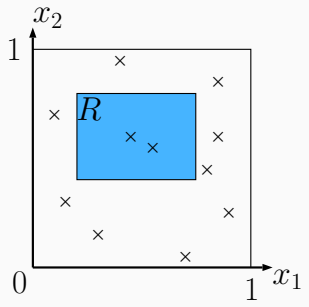
The second criterion is much more difficult to optimise.

2D illustration of maximin and minimax designs:



2. Compare the distribution with a uniform distribution

Discrepancy is a measure of non-uniformity. It compares the number of points in a hyper-rectangle with the expected number of samples from a uniform distribution.



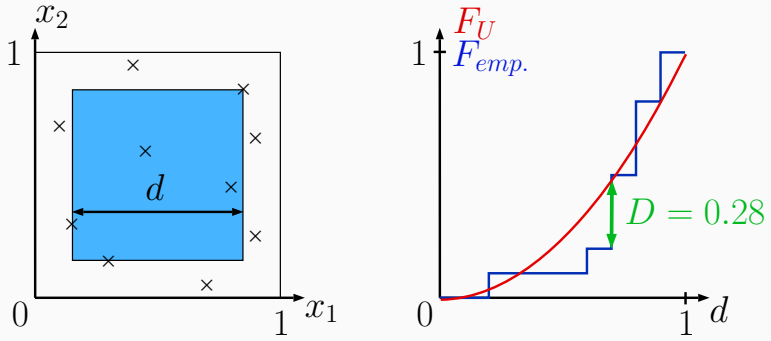
The probability for a uniform variable to be in R is 0.22 and we observe an empirical probability of $2/11$. The discrepancy (w.r.t. R) is then:

$$D_R = |0.22 - 2/11| = 0.038.$$

Discrepancy is defined as the sup of the distance between the empirical and analytical cdf.

Discrepancy is often computed by fixing:

- one vertex of the hyper-rectangle at the origin,
- the hyper-rectangle centre at the domain centre.



The maximum is located where the rectangle is tangent to points.

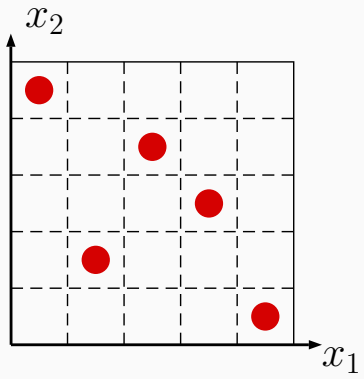
$\rightarrow$ The optimisation is over a finite space.

We will discuss three types of space-filling designs:

- Latin hypercubes,
- low-discrepancy sequences,
- centroidal Voronoi tessellations.

Latin hypercubes

Latin hypercubes are designs where the domain is sliced into n^d blocks and there is only one point per "column":



These designs have good projection properties.

A well-known example of an LHD in 2D is...

A well-known example of an LHD in 2D is… a Sudoku.

4	3	1	6	7	9	5	2	8
9	6	7	2	5	8	3	4	1
5	8	2	1	4	3	9	6	7
6	5	9	8	1	7	2	3	4
3	2	8	5	6	4	1	7	9
7	1	4	9	3	2	8	5	6
8	7	3	4	2	1	6	9	5
1	4	5	3	9	6	7	8	2
2	9	6	7	8	5	4	1	3

Any digit, say 4, is an LHD:

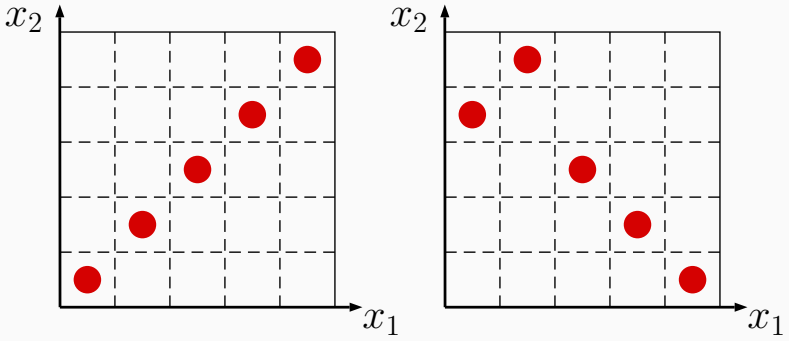
●	3	1	6	7	9	5	2	8
9	6	7	2	5	8	3	●	1
5	8	2	1	●	3	9	6	7
6	5	9	8	1	7	2	3	●
3	2	8	5	6	●	1	7	9
7	1	●	9	3	2	8	5	6
8	7	3	●	2	1	6	9	5
1	●	5	3	9	6	7	8	2
2	9	6	7	8	5	●	1	3

Note: Sudokus have additional constraints on the 3×3 blocks.

Exercise

- Generate a 5-point LHD in dimension 3.
- How would you programme a function $\mathrm{LHD}(n, d)$?

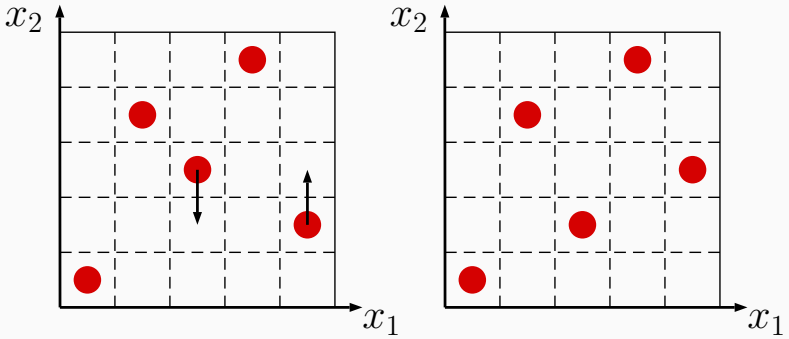
Latin hypercubes do not necessarily cover the space very well...



They have to be combined with a criterion such as maximin.

LHD optimisation

A common optimisation method is to exchange coordinates between two points:



LHD optimisation

The Morris and Mitchell algorithm is a popular approach based on *simulated annealing*.

1. Generate an LHD.
2. Find “bad” points according to maximin.
3. Choose a column of a critical point at random and exchange it with a randomly selected other point.
4. If the criterion is improved, accept the modification;
5. otherwise, accept it with probability $\exp\left(\frac{\text{maximin}_{\text{new}} - \text{maximin}_{\text{old}}}{T}\right)$.

Low-discrepancy sequences

Low-discrepancy sequences are deterministic sequences that converge toward the uniform distribution.

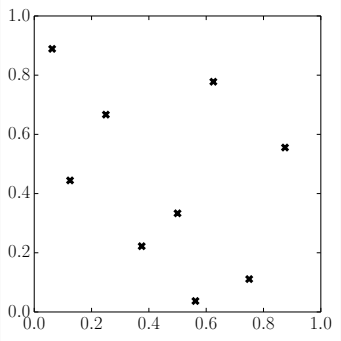
- They cover the space quickly and evenly.
- They are easy to build.
- It is easy to add new points.

Many low-discrepancy sequences can be found in the literature: Halton, Hammersley, Sobol', Faure, van der Corput, ...

Example (Halton sequence)

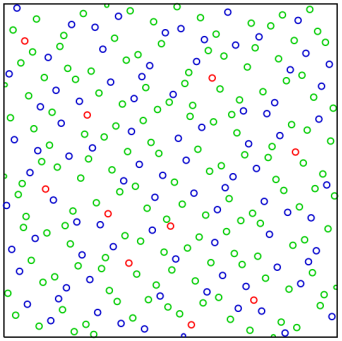
Let a and b be two integers with no common divisors (say 2 and 3). The x_1 and x_2 coordinates of the Halton sequence are:

$$x_1 = 1/2,\ 1/4,\ 3/4,\ 1/8,\ 5/8,\ 3/8,\ 7/8,\ 1/16,\ 9/16, \ldots$$
$$x_2 = 1/3,\ 2/3,\ 1/9,\ 4/9,\ 7/9,\ 2/9,\ 5/9,\ 8/9,\ 1/27, \ldots$$

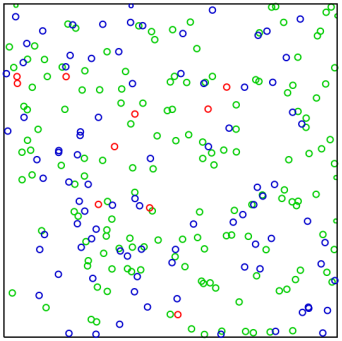


Example (Halton sequence)

Halton sequence



uniform pseudo-random



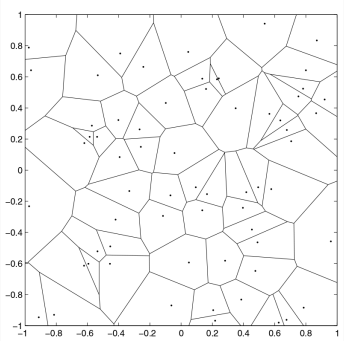
source: Wikipedia

Issues with low-discrepancy sequences:

- there can be alignments when projected,
- there can be holes in subspaces,
- points may be aligned (example: the first 16 points in bases (17,18)).

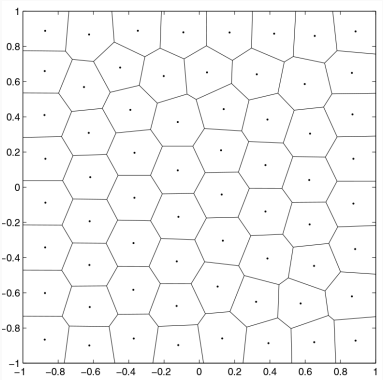
Centroidal Voronoi tessellations

Given a set of generator points X, the **Voronoi tessellation** associated with the point X_i is the region of the space such that X_i is the closest point from the set:



Source: Q. Du et al., *Centroidal Voronoi Tessellations: Applications and Algorithms*, SIAM Review, 41-4, 1999.

Centroidal Voronoi tessellations (CVT) are a special case of Voronoi tessellations where the generator points correspond to the centre of mass of the cells:



Source: Q. Du et al., *Centroidal Voronoi Tessellations: Applications and Algorithms*, SIAM Review, 41-4, 1999.

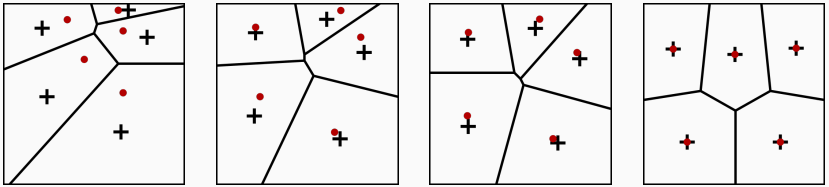
Properties of CVT:

- each point of the space is close to one generator point;
- the generator points cover the space.

$\Rightarrow$ The generator points of a CVT can be used as a design of experiments.

1. Lloyd's algorithm

1 Initialise X as a set of n points.
2 While $i <$ `nb_iter`:
3 compute the Voronoi diagram of X;
4 set X to the set of centres of mass of the cells.

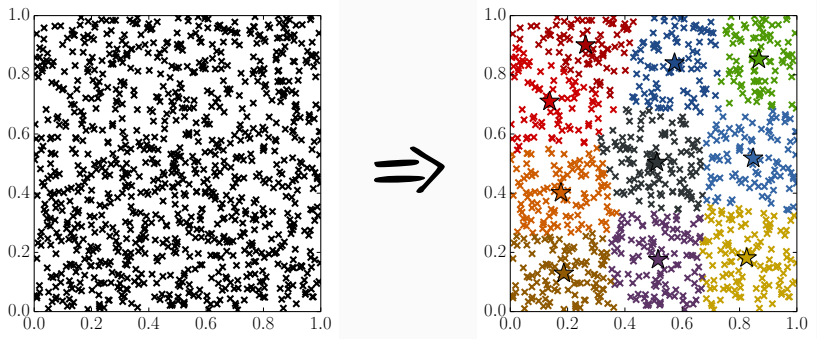


iteration 1 iteration 2 iteration 3 iteration 15

Source: Wikipedia, "Lloyd's algorithm".

2. k-means

This algorithm is very similar to Lloyd but it uses a large set of points covering the input space instead of the full continuous domain:



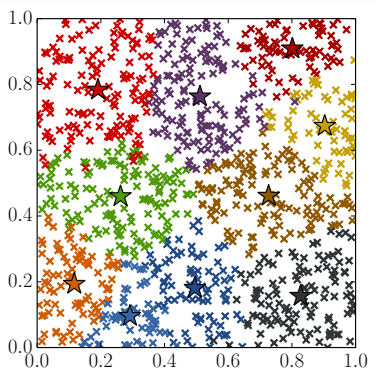
3. MacQueen algorithm

Approximate k-means if the latter is too slow:

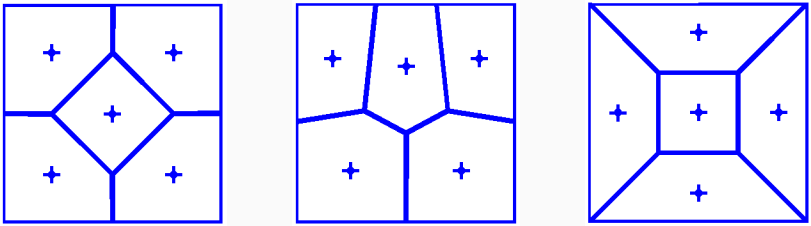
1 Initialise X as a set of n points.
2 Initialise k as a vector of ones with length n.
3 While $i <$ `nb_iter`:
4 generate one random point z in the input space;
5 find the X_i closest to z;
6 update $X_i = \frac{k_i X_i + z}{k_i + 1}$;
7 set $k_i = k_i + 1$.

3. MacQueen algorithm

We obtain the following design:



CVTs are not unique:



Source: Wikipedia, “Centroidal Voronoi Tessellations”.

Conclusion

Design of Experiments: principles

- Control of data generation
- Careful thinking about experimental choices

Objectives

- Measure the influence of the variables
- Get accurate models
- Get good inference for the models

Measure variable influence

The influence of a variable can be estimated by its linear regression coefficient.

Additive models: The number of points depends on the complexity of the univariate effects (linear, quadratic, ...)

- star-shaped designs
- one-at-a-time designs

Models with interaction

- factorial designs

Designs without model assumptions

Designs without model assumptions $\rightarrow$ space-filling designs

Various designs have been introduced

- Latin hypercubes
- low-discrepancy sequences
- centroidal Voronoi tessellations

Various criteria

- quality of projection
- discrepancy
- maximin
- minimax

Designs with model assumptions

When the form of the model is known, we can define various optimality criteria.

Best model estimation

- D-optimality
- A-optimality
- E-optimality

Lowest prediction error

- G-optimality
- I-optimality

For linear regression we have some interesting results...
For GPR, it's much more tricky!

In general, optimising DoE is difficult:

- large number of variables: $n \times d$;
- computationally expensive criteria.

Alternatives are:

- optimising within a given class of DoE (LHD),
- adaptive designs: add points one-at-a-time.