Constructing low-discrepancy point sets via subset sampling

François CLÉMENT

18/05/2022

An introduction to discrepancy

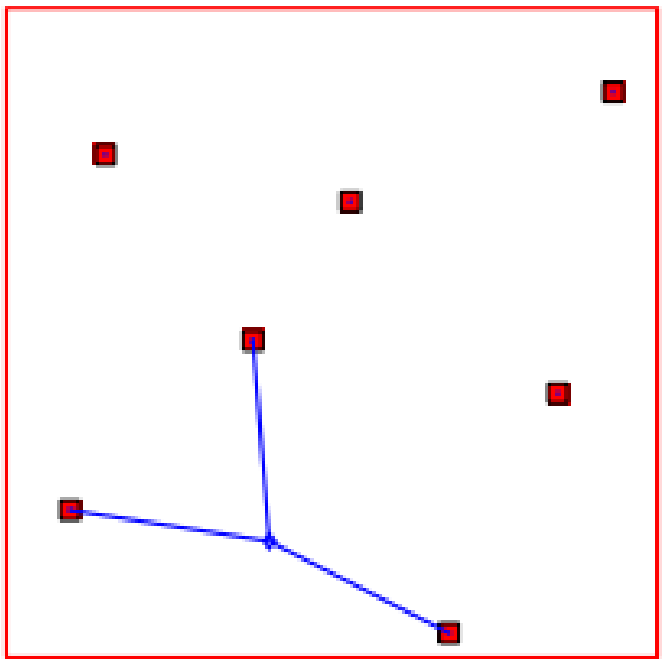
- What is a well-spread point set in $[0,1]^d$?

An introduction to discrepancy

- What is a well-spread point set in $[0,1]^d$?
- Natural approach: distance-based measures.

An introduction to discrepancy

- What is a well-spread point set in $[0,1]^d$?
- Natural approach: distance-based measures.
- **Dispersion**: Minimize the distance from any point to a point our set.



An introduction to discrepancy

- What is a well-spread point set in $[0,1]^d$?
- More complicated: uniformity-based measures.

An introduction to discrepancy

- What is a well-spread point set in $[0,1]^d$?
- More complicated: uniformity-based measures.
- **Discrepancy**: A box contains roughly the same proportion of points as its volume.

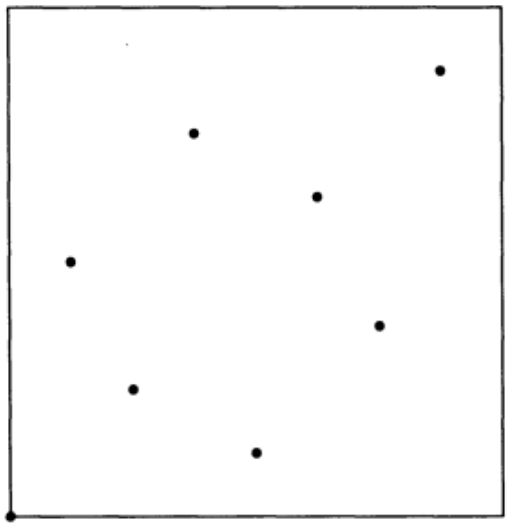


Figure: The first 8 points of the Van der Corput set

Definitions

L_∞-discrepancy

For P a point set in $[0;1]^d$,

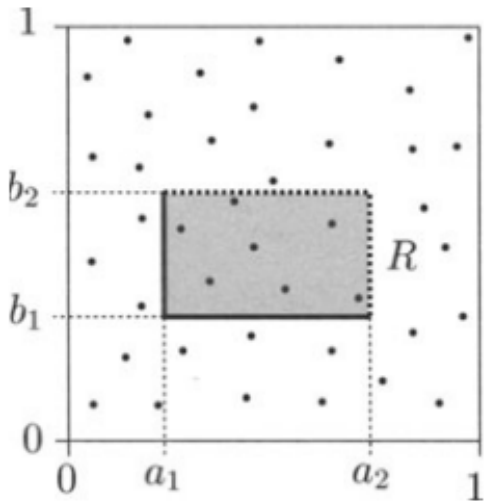
$$d_\infty(P) = sup_{a,b\in[0;1]^d, a\leq b}\left|\frac{\left|P\cap[a,b)\right|}{|P|} - \lambda\big([a;b)\big)\right|.$$

Definitions

L_∞-discrepancy

For P a point set in $[0;1]^d$,

$$d_\infty(P) = sup_{a,b\in[0;1]^d, a\leq b}\left|\frac{|P\cap[a,b)|}{|P|} - \lambda([a;b))\right|.$$



Definitions

L_∞-star discrepancy

For P a point set in $[0;1]^d$,

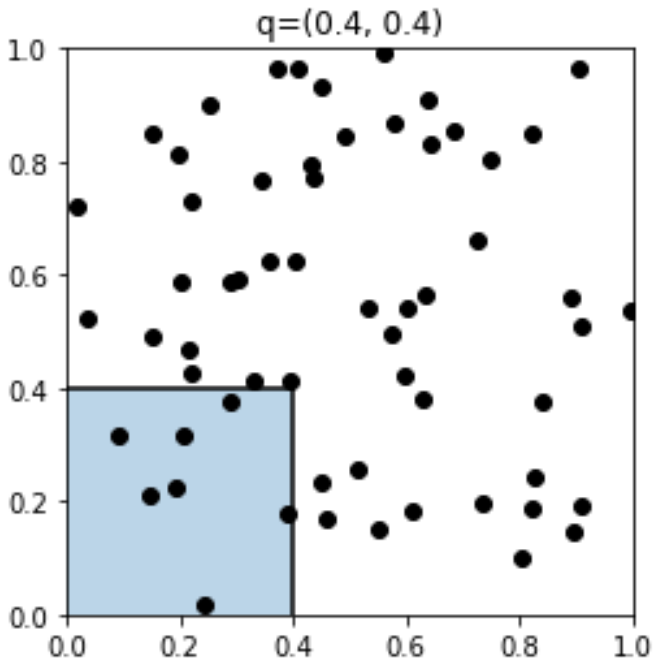
$$d^*_\infty(P) = sup_{q\in[0;1]^d}\left|\frac{|P\cap[0,q)|}{|P|} - \lambda([0,q))\right|.$$

Definitions

L_∞-star discrepancy

For P a point set in $[0;1]^d$,

$$d^*_\infty(P) = sup_{q\in[0;1]^d}\left|\frac{|P\cap[0,q)|}{|P|} - \lambda([0,q))\right|.$$



Local discrepancy: 0.044

Some examples

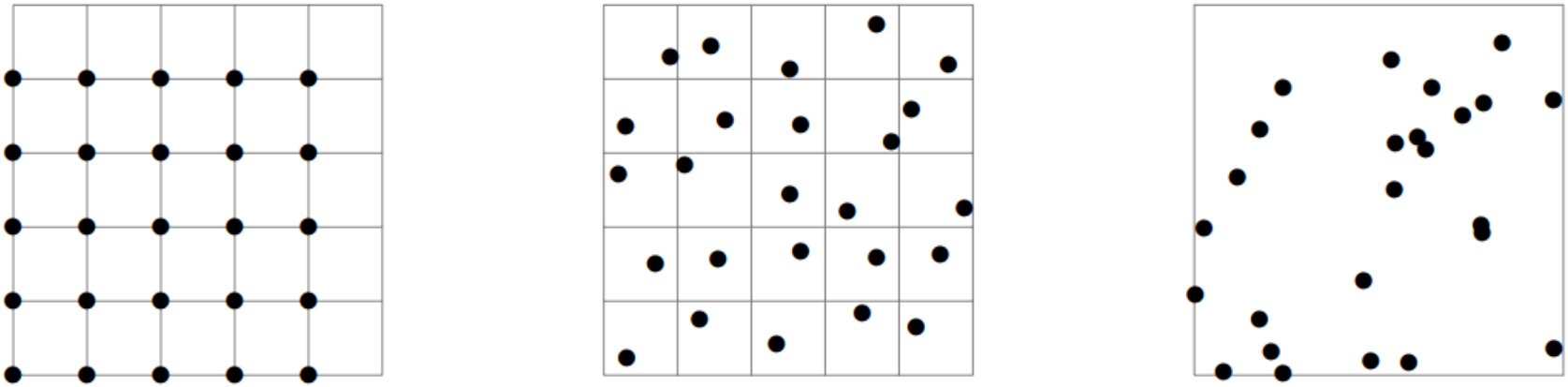


Figure: Grid, Stratified Sampling and Random points for $n = 25$

Some examples

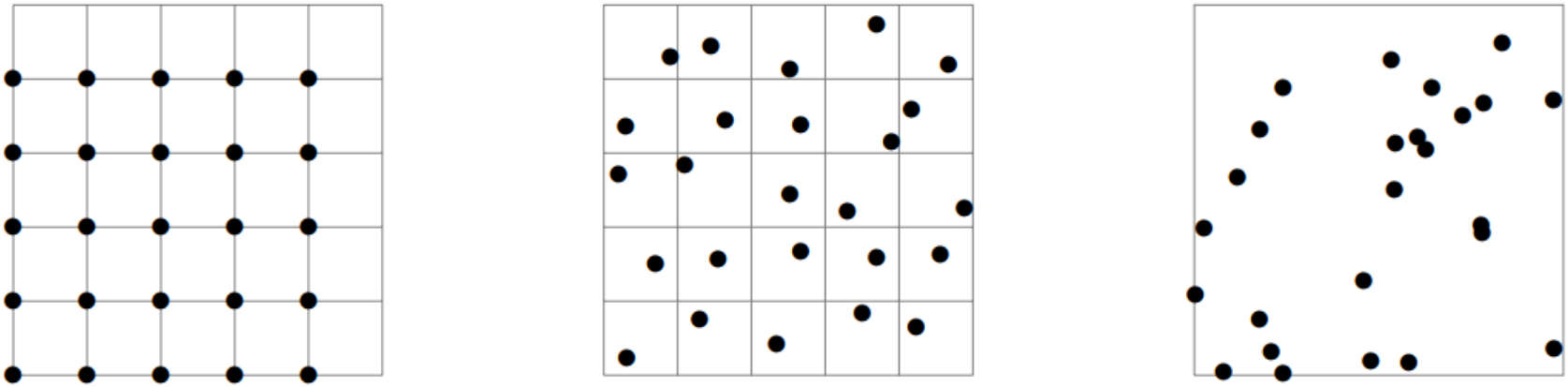


Figure: Grid, Stratified Sampling and Random points for $n = 25$

- Grid points: Looks OK but scales very badly with n and d.

Some examples

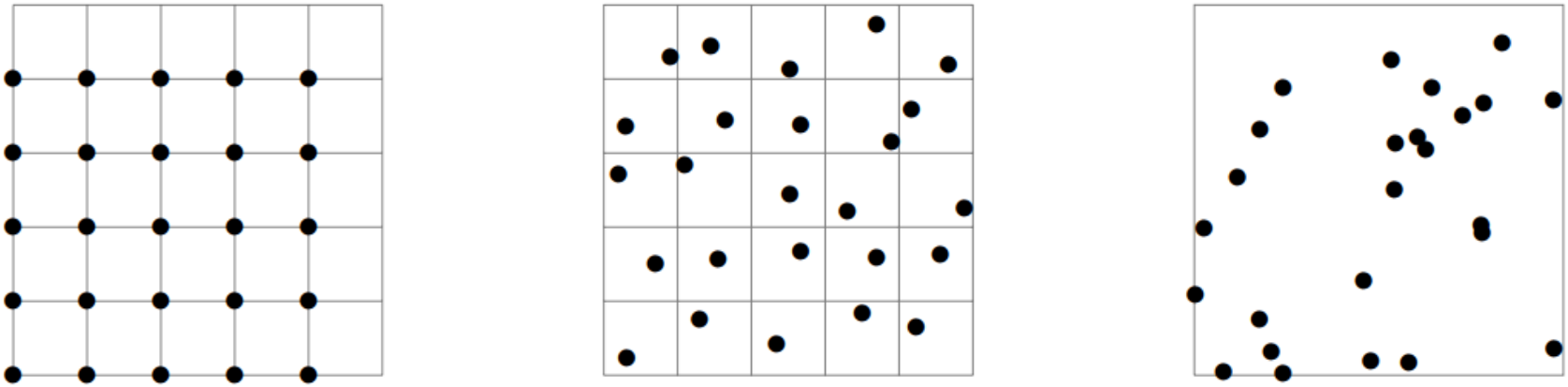


Figure: Grid, Stratified Sampling and Random points for $n = 25$

- Grid points: Looks OK but scales very badly with n and d.
- Stratified sampling: Avoids the long empty columns of the grid.

Some examples

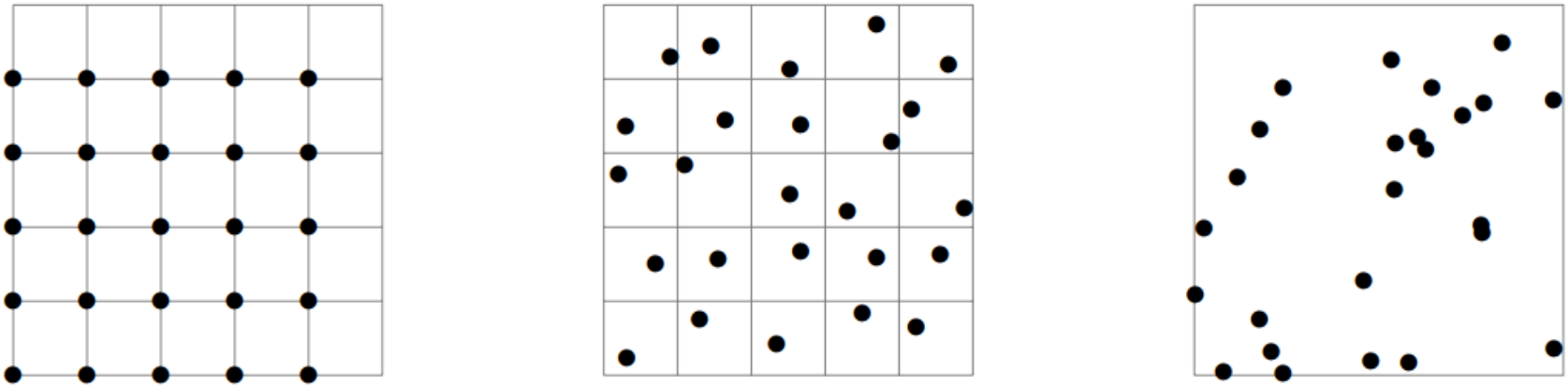


Figure: Grid, Stratified Sampling and Random points for $n = 25$

- Grid points: Looks OK but scales very badly with n and d.
- Stratified sampling: Avoids the long empty columns of the grid.
- Random points: They should converge to a uniform distribution, but are they good?

Low-discrepancy sequences

A lower-bound for the L_∞-star discrepancy (Bilyk et al. 2008)

For any point set P of size n in dimension d, there exists a constant $c = c(d)$ such that

$$d^*_\infty(P) \geq c \frac{\log^{c+(d-1)/2}(n)}{n}$$

Low-discrepancy sequences

A lower-bound for the L_∞-star discrepancy (Bilyk et al. 2008)

For any point set P of size n in dimension d, there exists a constant $c = c(d)$ such that

$$d^*_\infty(P) \geq c \frac{\log^{c+(d-1)/2}(n)}{n}$$

Conjecture

For any point set P of size n in dimension d, there exists a constant $c = c(d)$ such that

$$d^*_\infty(P) \geq c \frac{\log^{d-1}(n)}{n}$$

Low-discrepancy sequences

- With the previous conjecture, this would give a lower bound in $O(log^d(n)/n)$, for **sequences**.

Low-discrepancy sequences

- With the previous conjecture, this would give a lower bound in $O(log^d(n)/n)$, for **sequences**.
- In practice, there are many known sequences $(P_n)_{n\in\mathbb{N}}$ such that $d^*_\infty(P_n) \leq C_d log^d(n)/n$, with C_d a constant.

Low-discrepancy sequences

- With the previous conjecture, this would give a lower bound in $O(log^d(n)/n)$, for **sequences**.
- In practice, there are many known sequences $(P_n)_{n\in\mathbb{N}}$ such that $d^*_\infty(P_n) \leq C_d log^d(n)/n$, with C_d a constant.
- These sequences are known as low-discrepancy sequences.

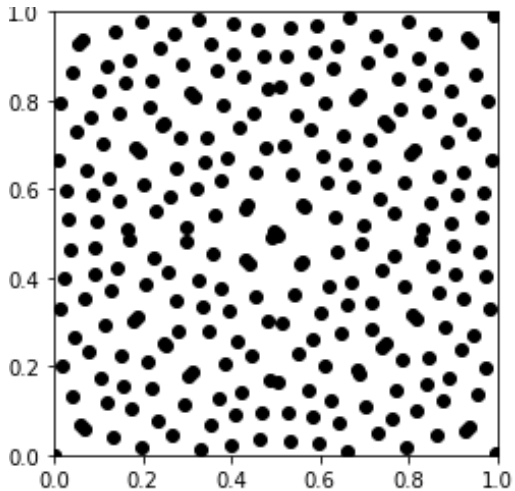
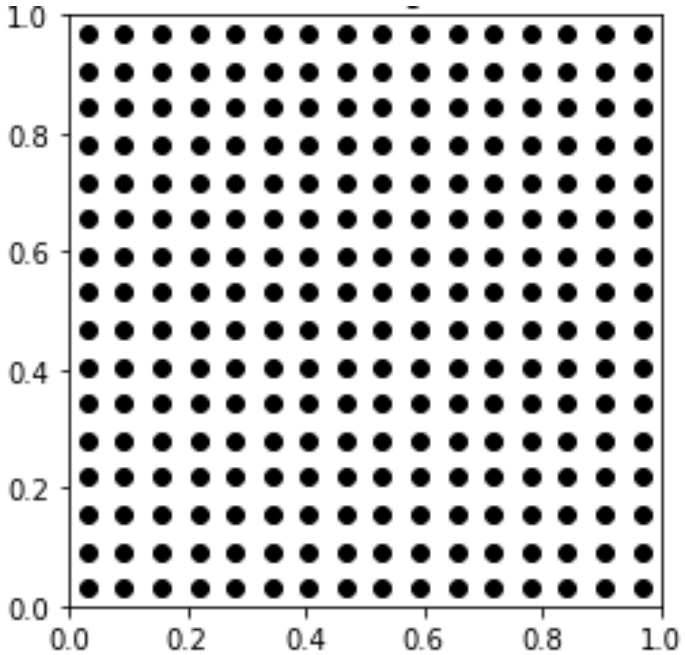


Figure: Beginning of the Sobol sequence in 2 dimensions

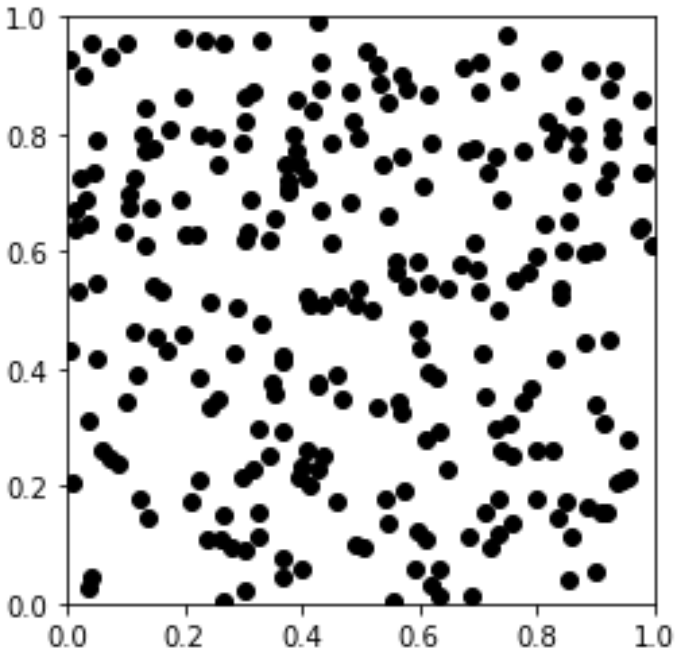
Some examples (again)

- Grid points: Discrepancy scales in $O(n^{-1/d})$



Some examples (again)

- Random points and Latin Hypercube Samples: Discrepancy is in $O(\sqrt{d/n})$ (Heinrich et al., Doerr, Doerr et al.).



Why do all this?

- "Random" searches: one-shot optimization, Quasi-Monte Carlo methods...

Why do all this?

- "Random" searches: one-shot optimization, Quasi-Monte Carlo methods...
- Covering a search space uniformly: Design of experiments.

Why do all this?

- "Random" searches: one-shot optimization, Quasi-Monte Carlo methods...
- Covering a search space uniformly: Design of experiments.
- Koksma-Hlawka inequality: Discrepancy is a bound for the error of approximating an integral.

$$\left| \int_{[0,1]^d} f(x) d\lambda^d(x) - \frac{1}{|P|} \sum_{p \in P} f(p) \right| \leq Var(f) d^*_\infty(P)$$

The L_∞-star discrepancy

- Thanks to the Koksma-Hlawka inequality, it's the most important one in practice.

The L_∞-star discrepancy

- Thanks to the Koksma-Hlawka inequality, it's the most important one in practice.
- Very difficult to compute: NP-hard (Gnewuch et al.) and even W[1]-hard with respect to the dimension (Giannopoulos et al.).

The L_∞-star discrepancy

- Thanks to the Koksma-Hlawka inequality, it's the most important one in practice.
- Very difficult to compute: NP-hard (Gnewuch et al.) and even W[1]-hard with respect to the dimension (Giannopoulos et al.).
- Low-discrepancy sequences are optimised for asymptotic results. What should we use in practice for specific n and d combinations?

Computing the star discrepancy

L_∞-star discrepancy

For P a point set in $[0;1]^d$,

$$d^*_\infty(P) = sup_{q\in[0;1]^d}\left|\frac{\left|P\cap[0,q)\right|}{|P|} - \lambda([0,q))\right|.$$

Computing the star discrepancy

L_∞-star discrepancy

For P a point set in $[0;1]^d$,

$$d^*_\infty(P) = sup_{q\in[0;1]^d}\left|\frac{|P\cap[0,q)|}{|P|} - \lambda([0,q))\right|.$$

- The problem looks continuous but it's actually a discrete optimization problem.

Computing the star discrepancy

L_∞-star discrepancy

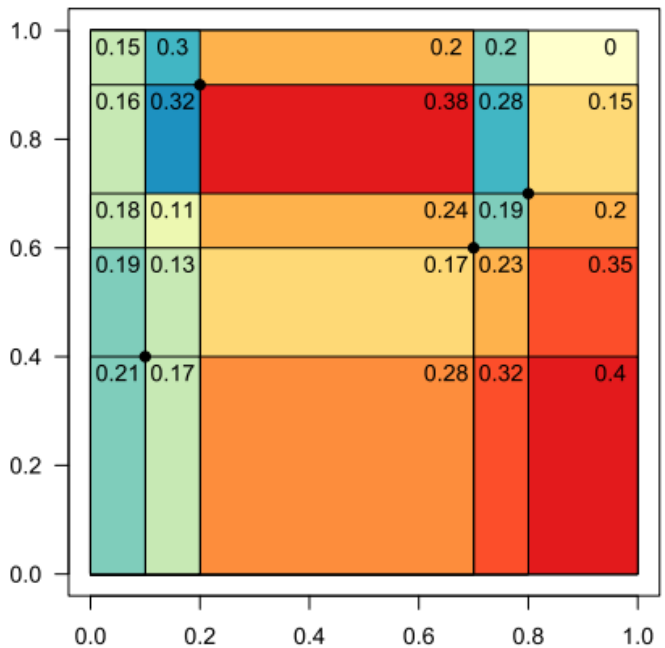
For P a point set in $[0;1]^d$,

$$d_\infty^*(P) = sup_{q\in[0;1]^d}\left|\frac{|P\cap[0,q)|}{|P|} - \lambda([0,q))\right|.$$

- The problem looks continuous but it's actually a discrete optimization problem.
- For a point set $P = (p^{(i)})$, $\Gamma(X) = \prod_{i=1}^d \left\{p_j^{(i)} : j \in \{1,...,n\}\right\} \cup \{1\}$ are the only possible positions for the worst-case box (either open or closed).

Computing the star discrepancy

For a point set $P = (p^{(i)})$, $\Gamma(X) = \prod_{i=1}^{d} \left\{ p_j^{(i)} : j \in \{1, \ldots, n\} \right\} \cup \{1\}$ are the only possible positions for the worst-case box (either open or closed).



Computing the star discrepancy

- This set can be further reduced: only boxes with a point on each "outer" side are **critical**.

Computing the star discrepancy

- This set can be further reduced: only boxes with a point on each "outer" side are **critical**.
- Going through all these boxes takes $O(n^d/d!)$ time.

Computing the star discrepancy

- This set can be further reduced: only boxes with a point on each "outer" side are **critical**.
- Going through all these boxes takes $O(n^d/d!)$ time.
- A more sophisticated algorithm using dynamic programming runs in $O(n^{d/2+1})$ (Dobkin et al.).

Computing the star discrepancy

- This set can be further reduced: only boxes with a point on each "outer" side are **critical**.
- Going through all these boxes takes $O(n^d/d!)$ time.
- A more sophisticated algorithm using dynamic programming runs in $O(n^{d/2+1})$ (Dobkin et al.).
- More recent approaches using threshold accepting have found "success" for higher dimensions (10, 20 and even 50 for very small sets).

Summary

- A very useful uniformity measure but extremely hard to compute.

Summary

- A very useful uniformity measure but extremely hard to compute.
- Most results/constructions are targeting asymptotic bounds. Practical applications just use low-discrepancy sequences or random points by default.

Summary

- A very useful uniformity measure but extremely hard to compute.
- Most results/constructions are targeting asymptotic bounds. Practical applications just use low-discrepancy sequences or random points by default.
- My PhD: develop methods to construct good low-discrepancy point sets from existing ones.

Low-discrepancy subset selection

- **Objective**: Being able to construct good low-discrepancy point sets tailored to applications: between hundreds and thousands of points, in low and medium dimensions.

Low-discrepancy subset selection

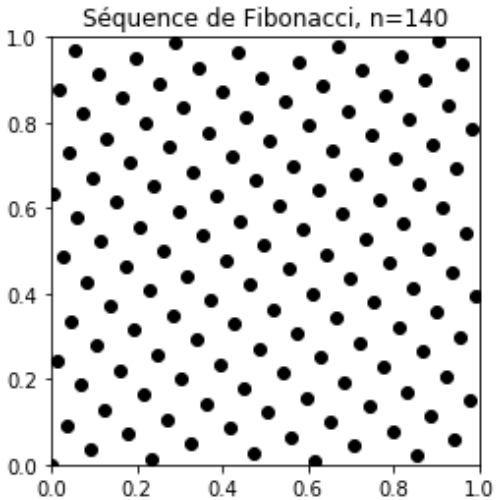
- **Objective**: Being able to construct good low-discrepancy point sets tailored to applications: between hundreds and thousands of points, in low and medium dimensions.
- Choose the best k-size subset of an n point low-discrepancy set.

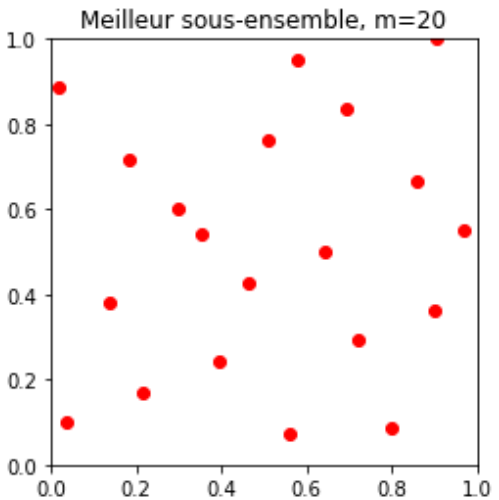
Low-discrepancy subset selection

- **Objective**: Being able to construct good low-discrepancy point sets tailored to applications: between hundreds and thousands of points, in low and medium dimensions.
- Choose the best k-size subset of an n point low-discrepancy set.
- Still NP-hard, even if we don't compute the discrepancy.

An example

The point set on the right has discrepancy 0.0731 against 0.0930 for the 20-point Fibonacci set.





Exact algorithmic approaches

- **Branch and bound**: add points one by one in our chosen set. Bounds are given by the number of points we could potentially add in each box.

Exact algorithmic approaches

- **Branch and bound**: add points one by one in our chosen set. Bounds are given by the number of points we could potentially add in each box.
- Works reasonably well up to $n = 140$ in dimension 2 and $n = 100$ in dimension 3.

Exact algorithmic approaches

- **Branch and bound**: add points one by one in our chosen set. Bounds are given by the number of points we could potentially add in each box.
- Works reasonably well up to $n = 140$ in dimension 2 and $n = 100$ in dimension 3.
- Even with the final result as the bound, the algorithm struggles if we increase n.

MILP formulation

- We are trying to find the minimal discrepancy value over all possible subsets of size k.

MILP formulation

- We are trying to find the minimal discrepancy value over all possible subsets of size k.
- Calculating the discrepancy is a discrete optimization problem: we need to check the local discrepancy value for each critical box.

MILP formulation

- We are trying to find the minimal discrepancy value over all possible subsets of size k.
- Calculating the discrepancy is a discrete optimization problem: we need to check the local discrepancy value for each critical box.
- We know for each box exactly which points of our original set are inside.

$$\Rightarrow \textbf{We can formulate an MILP!}$$

MILP formulation

$$
\begin{aligned}
\min \quad & z \\
\text{s.t.} \quad & z \geq h_{i,j} - \frac{1}{k} \sum_{\ell \in \Delta(P,i,j)} x_\ell && \text{for all } i,j \in [1..n+1] \\
& z \geq -h_{i,j} + \frac{1}{k} \sum_{\ell \in \overline{\Delta}(P,i,j)} x_\ell && \text{for all } i,j \in [1..n] \\
& \sum_{i=1}^{n} x_i = k \\
& x_i \in \{0,1\} && \text{for all } i \in [1..n] \\
& z \in \mathbb{R}_{\geq 0}
\end{aligned}
$$

Results

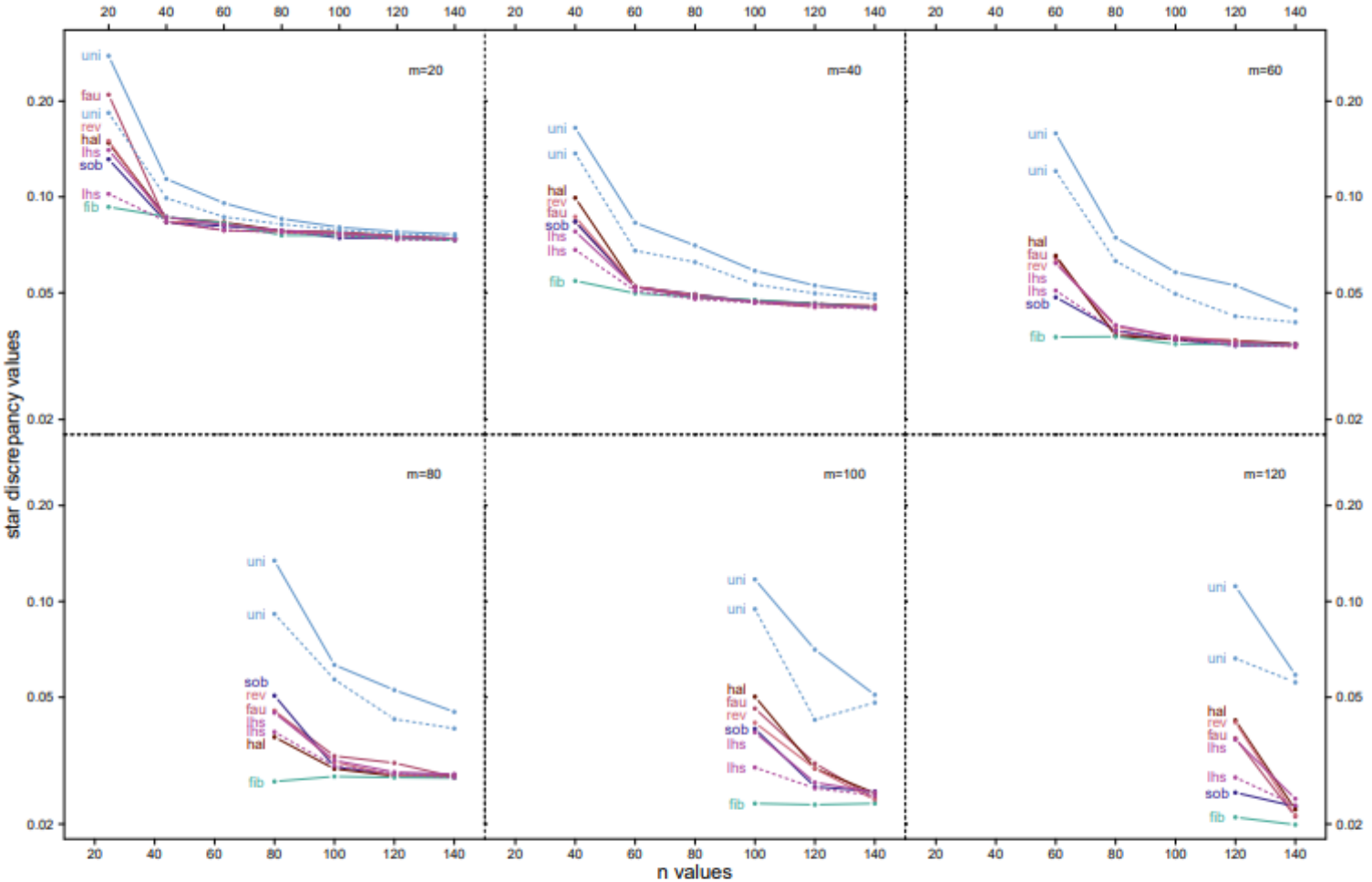


Figure 3: Star discrepancy values for each tested combination of m and n in $2d$. For the two randomized constructions `iLHS` and `unif`, minimum (dashed lines) and median (solid lines) values across the ten independent runs are shown

Open questions

- Can we find better exact algorithms? Are there better subset selection methods we could adapt here?

Open questions

- Can we find better exact algorithms? Are there better subset selection methods we could adapt here?
- Can we find heuristics with decent performance guarantees?

Open questions

- Can we find better exact algorithms? Are there better subset selection methods we could adapt here?
- Can we find heuristics with decent performance guarantees?
- Do the methods extend to other discrepancies? L_2-star discrepancy subset selection can be represented as an Unconstrained Binary Quadratic Problem.

Open questions

- Can we find better exact algorithms? Are there better subset selection methods we could adapt here?
- Can we find heuristics with decent performance guarantees?
- Do the methods extend to other discrepancies? L_2-star discrepancy subset selection can be represented as an Unconstrained Binary Quadratic Problem.
- Any specific applications we could target are interesting!

Open questions

- Can we find better exact algorithms? Are there better subset selection methods we could adapt here?
- Can we find heuristics with decent performance guarantees?
- Do the methods extend to other discrepancies? L_2-star discrepancy subset selection can be represented as an Unconstrained Binary Quadratic Problem.
- Any specific applications we could target are interesting!

Thank you for your attention!

References (images)

- Slide 2: L.Pronzato, Maximum Mean Discrepancy, Bayesian integration and kernel herding for space-filling design, séminaire UQSay, dec 2021.
- Slide 4: Jirí Matousek, Geometric Discrepancy, An illustrated guide, Springer, 2009.
- Slide 6: M. Kiderlen, F.Pausinger, Discrepancy of stratified samples from partitions of the unit cube, Monatshefte für Mathematik, 2021.

References

- D. Bilyk, M.T. Lacey, A. Vagharshakyan, On the small ball inequality in all dimensions, J. Funct. Anal. 2008.
- S. Heinrich, E. Novak, G.W. Wasilkowski, H. Wozniakozski, The inverse of the star discrepancy depends linearly on the dimension, Acta Arith. (96), 2001.
- B. Doerr, A lower bound for the discrepancy of a random point set, J. Complex (2014)
- B. Doerr, C.Doerr, M. Gnewuch, Probabilistic lower bounds for the discrepancy of Latin Hypercube Samples, 2018.

References

- M. Gnewuch, A. Srivastav, C. Winzen, Finding optimal volume subitnervals with k points and alulcating the star discrepancy are NP-hard problems, J. Complex, 2009.
- P. Giannopoulos, C. Knauer, M. Wahlström, D.Werner, Hardness of discrepancy calculation and ϵ-net verification in high dimension, J. Complex., 2012.
- F. Clément. C.Doerr. L. Paquete, Star discrepancy Subset Selection: Problem Formulation and efficient approaches for low dimensions, J.Complex., 2022.