Privacy sets revisited

Werner G. Müller

joint work with Eva Benková and Radoslav Harman

Mascot-Num, Paris, March 19\textsuperscript{th} 2019
But first: Happy Birthday Luc!

Figure 1: Participants of mODa3, St. Petersburg, 1992.
Overview of the talk

- Introduction to optimal experimental design
- A more fundamental way of bridging designs
- Privacy sets / algorithm (PSA)
- Minimum-distance PSs
But again: Happy Birthday Luc!

Figure 2: Participants of mODa4, Spetses, 1995.
Optimal experimental design

Let us consider a linear regression model

\[ y(x) = f(x)^\top \beta + \varepsilon(x), \]

where
- \( x \in \mathcal{X}, \mathcal{X} \subseteq \mathbb{R}^d \) is a non-empty set called the design space,
- \( y(x) \) is an observation in the design point \( x \),
- \( \beta \in \mathbb{R}^m \) is a vector of unknown parameters,
- \( f(x) \in \mathbb{R}^m \) is a known regression function in the point \( x \),
- \( \varepsilon(x) \), are uncorrelated homoscedastic errors with mean values equal to zero.

The problem of optimal experimental design is to select points \( x \in \mathcal{X} \) such that the variance of the estimate \( \hat{\beta} \) is minimized.

Formally, let \( \xi \) be a finite subset of \( \mathcal{X} \) representing an experimental design. The problem of optimal experimental design (cf. e.g. Pázmán 1986) is to maximize some criterion in the set of all permissible designs \( \Gamma \), that is, to find

\[ \xi^* \in \arg\max_{\xi \in \Gamma} \Phi(\xi), \]

where \( \Gamma \subseteq \Xi \) and \( \Xi \) denotes the set of all designs.
Space-filling designs.

**Space-filling** designs cover the design space $\mathcal{X}$ as uniformly as possible, being often used in computer simulation experiments, where a deterministic computer code replaces the real (stochastic) data generating process.

We distinguish two approaches to space-filling:

- **Soft methods** of space-filling focus on proposing and tuning the criterion $\Phi$ and then optimize on the set $\Gamma$ of all designs of a given size.

- **Hard methods** pose **constraints** on the design, such that all designs from $\Gamma$ satisfy a required minimum level of space-fillingness and another criterion $\Phi$ can be optimized.
Soft space-filling.

Among the criteria $\Phi$, two are the most fundamental: Maximizing

$$\phi_{Mm}(\xi) = \min_{x, y \in \xi, x \neq y} \rho_2(x, y),$$

and minimizing

$$\phi_{mM}(\xi) = \max_{y \in \mathcal{X}} \min_{x \in \xi} \rho_2(x, y),$$

see Pronzato & Müller, 2012:

![Figure 3: Maximin and minimax distance designs for $N = 7$ points in $[0, 1]^2$.](image)
Hard space-filling.

Very common are *Latin hypercube designs (LHDs)* introduced in McKay et al., 1979: design points within the discrete regular grid of \( n = N^d \) points, with levels \( 0, 1/(N-1), \ldots, 1 \) for every dimension, such that for all \( x, y \in \xi \) it holds that \( x_i \neq y_i \) for all \( i = 1, \ldots, d \).

Morris, 1995 suggests to search for the optimal Maximin design within the class of Latin hypercube designs, minimizing

\[
\phi_{MmLHD}(\xi) = \left\{ \sum_{\substack{x, y \in \xi \\text{ s.t. } x \neq y \\text{ and } x_i \neq y_i \text{ for all } i = 1, \ldots, d}} \frac{1}{\rho_2^k(x, y)} \right\}^{1/k}.
\]

*Figure 4:* Maximin Latin hypercube designs for \( N = 7 \) points in \([0, 1]^2\).
Combining with optimal designs

Among all LHDs it may now be for example desired to search for a $D$-optimal design.

More generally Bridge designs (BDs) introduced in Jones et al., 2014 aim to “bridge” the gap between $D$-optimal designs $D$-optimal LHDs. Any two points $x, y : x \neq y$ of a Bridge design $\xi$ must satisfy

$$|x_i - y_i| > \delta > 0 \text{ for all } i = 1, \ldots, d.$$ 

Alternatively we could ask for all $x, y \in \xi, x \neq y$ such that

$$\rho_2(x, y) > \delta > 0.$$ 

These designs will be called minimum-distance designs (MDDs).
And again: Happy Birthday Luc!

Figure 5: Participants of mODa5, Luminy, 1998.
Privacy sets

Let $\mathcal{X} \subset \mathbb{R}^d$ be a design space. For each $x \in \mathcal{X}$, let $\mathcal{P}(x) \subseteq \mathcal{X}$, $x \in \mathcal{P}(x)$, be a given privacy set of the point $x$. “Respecting” privacy sets means posing additional (linear) constraints on the design, yielding the optimization problem

$$\xi^* \in \arg\max_{\xi \in \Xi} \{ \Phi(\xi); |\xi| \leq N \text{ and } y \not\in \mathcal{P}(x) \text{ for all } x, y, \in \xi, x \neq y \}.$$ 

![Diagram of privacy sets](image)

(a) Latin hypercube designs  
(b) Bridge designs  
(c) Replication-free designs  
(d) Minimum-distance designs

Figure 6: Examples of privacy sets in dimension $d = 2.$
A more fundamental bridge

Our approach is a hard method, where constraints are given by the privacy sets.

Can be viewed as a generalization of the Bridge designs given in Jones et al, 2014.

We thus think that privacy sets represent a much more fundamental “bridge” between the two important ways of designing an experiment.
Privacy sets algorithm (PSA)

Using privacy sets yields the optimization problem

$$\xi^* \in \arg\max_{\xi \in \Xi} \{ \Phi(\xi); |\xi| \leq N \text{ and } x \notin \mathcal{P}(y) \text{ for all } x, y, \in \xi, x \neq y \}$$  \hfill (1)

- In Benková et al. (2016), we propose a general exchange-type heuristic for solving (1). It is a general framework with some steps left to be specified, depending greatly on the design space $\mathcal{X}$, on the constraints given by the sets $\mathcal{P}(x), x \in X$, as well as on the optimization criterion $\Phi$.

- One of the key differences is that our algorithm can temporarily violate “privacy” of one or more design points, which results in a broader range of possibilities.

- PSA does not pose any restrictions on the design space $\mathcal{X}$. Due to implementation reasons, we assume $|\mathcal{X}| = n \in \mathbb{N}$.

- If $n$ is relatively small (up to thousands of design points, say), the implementation of PSA is rather simple for all kinds of privacy sets.
Privacy sets algorithm

1. Construct an initial design $\xi$.
2. Construct the set of “candidate addition points” $A(\xi) \subseteq \mathcal{X} \setminus \xi$.
3. Pick up $x \in A(\xi)$:
   1. Add point $x$ to the design $\xi$, that is $\xi^{\text{new}} := \xi \cup \{x\}$.
   2. Delete from $\xi^{\text{new}}$ all the points belonging to the $\mathcal{P}(x)$ (except for $x$).
   3. If
      - $|\xi^{\text{new}}| = N + 1$, remove the design point from $\xi^{\text{new}}$ that leads to the smallest drop in the criterial value.
      - $|\xi^{\text{new}}| = N$, stop.
      - $|\xi^{\text{new}}| < N$, augment the design $\xi^{\text{new}}$ in a greedy way to a design of size $N$.
4. If $\Phi(\xi^{\text{new}}) > \Phi(\xi)$, set $\xi = \xi^{\text{new}}$ and continue by Step 2.
5. Else Continue by Step 3
6. If for all $x \in A(\xi)$ no design is better than $\xi$, return $\xi$.

Figure 7: Three different situations can occur during the mutation procedure. Figures 7a, 7b, 7c correspond to the steps 3,4,5 from the previous slide, in the respective order.
Privacy sets algorithm for bridge designs

Privacy sets of bridge designs are for a given constant $\delta > 0$ and any $x \in \mathcal{X}$ given by

$$\mathcal{P}(x) = \{ y \in \mathcal{X} : |x_i - y_i| < \delta \text{ for some } i \in \{1, \ldots, d\} \}.$$ 

We assume the design space $\mathcal{X}$ to be discretized into a grid of $n = L^d$, $L \in \mathbb{N}$ points.

Implementation specifics:

- In Step 1 of Greedy augmentation, it is enough to store just an $L \times d$ logical matrix representing permissible levels of factors. Points $x \in \mathcal{X} \setminus \mathcal{P}(\xi)$ can then be easily selected independently, coordinate by coordinate.

- The procedure does not require storing all design points of $\mathcal{X}$ in the computer memory $\implies$ very large design spaces can be considered.
Example: $D$-optimal bridge designs on $[-1, 1]^2$

Minimizing variance of $\hat{\beta} \Leftrightarrow$ maximizing information about $\beta$:

Let

$$M(\xi) = \frac{1}{N} \sum_{x \in \xi} f(x)f^\top(x)$$

be the standardized information matrix of the size $m \times m$. The inverse of a non-singular information matrix is proportional to the covariance matrix of the best linear unbiased estimator (BLUE) of $\beta$. The most common objective function $\Phi$ in optimal design is **D-criterion** given by

$$\Phi_D(\xi) = \det(M(\xi))^{1/m},$$

$D$-optimal design minimizes the volume of the confidence ellipsoid of the parameters $\beta$.

We present examples for 2 factors in $N = 21$ trials. The design space is a discrete grid on $[-1, 1]^2$ and we consider both the linear regression function $f(x) = (1, x_1, x_2)^\top$ and the full quadratic regression function $f(x) = (1, x_1, x_2, x_{21}, x_{22}, x_1 x_2)^\top$.

The size of the grid depends on the parameter $\delta$, which is set to 0.05 and 0.025, leading to $n = 41^2 = 1681$ and $n = 81^2 = 6561$. 
Once more: Happy Birthday Luc!

Figure 8: Participants of mODa6, Puchberg, 2001.
Example: Linear bridge designs

Figure 9: Resulting designs for linear model and $\delta = 0.05, 0.025$. Efficiencies of the best designs found by the algorithm of Jones et al. relative to the designs presented in Figs. 2(a), 2(b) are 0.79, 0.82, respectively.
Example: Quadratic bridge designs

Figure 10: Resulting designs for full quadratic model and $\delta = 0.05, 0.025$. Efficiencies of the best designs found by the algorithm of Jones et al. relative to the designs presented in Figs. 3(a), 3(b) are 0.96, 0.96, respectively.
Numerical study on bridge designs

We performed a small comparative numerical study on a few examples of **quadratic regression**, providing comparison of the algorithm of Jones et al. (2014) and PSA in various dimensions.

For every example, we ran the algorithms for a **restricted time** $T$, observing the time dependence of the criterion value of the actually **best design found by the algorithm**. We repeated this procedure 5 times in order to provide responses from multiple random starts.

The minimum spacing constant $\delta$ was in all four cases set to the value $\delta = 1/(N - 1)$, which corresponds to the value recommended in Jones et al. (2014). Every $t$ seconds, we plotted the criterion values of the best designs found so far.

If an algorithm terminated during the given time $T$, it was automatically restarted and the resulting value found by its run was stored in the memory. These restarts are denoted by the red and the blue diamonds.
Numerical study on bridge designs

Figure 11: Comparison of the performance of the algorithm of Jones et. al. (red lines) and PSA (blue lines). Y-axis shows the best $D$-optimality values found by the time displayed on the x-axis.
Another example

Criterion is average reciprocal distance (ARD) criterion (Draguljic et al., 2012) minimizing

\[
\phi_{\text{ARD}}(\xi) = \left( \frac{1}{\binom{N}{2}} \sum_{j \in J} (d_j) \sum_{j \in J} \sum_{Y \in \mathcal{X}_j} \sum_{x, y \in \xi, x \neq y} \left( \frac{j^{1/z}}{\rho(z(x^*_Y, y^*_Y))} \right)^{\lambda} \right)^{1/\lambda},
\]

where \( z \geq 1 \) and \( \lambda \geq 1 \) and \( x^*_Y \) is the projection of \( x \) onto subspace \( Y \).

(a) \( J = \{1\} \)

(b) \( J = \{2\} \)

(c) \( J = \{1, 2\} \)
And now for something completely different: Happy Birthday Luc!

Figure 12: A party at Henry’s place, 2006.
Minimum-distance privacy sets

- in the eye of a chicken: the color-sensitive cone cells are distributed “hyperuniformly” - i.e., randomly, but sufficiently far away from each other at the same time, see [?];
- corresponds to very natural privacy sets for space-filling designs
- no appropriate efficient algorithm is available yet to handle these restrictions;
- privacy sets in the form \( \mathcal{P}(x) = \{ y : d(x, y) < \delta \} \), where \( d(., .) \) is the Euclidean metric and \( \delta > 0 \) is a given constant.

Figure 13: Olena Shmahalo/Quanta Magazine; Photography: MTSOfan and Matthew Toomey.
Mixtures of privacy sets.

If we separate the cone types according to the color, we can see that they never get “too close to each other”: hyperuniformity.
Minimum-distance privacy sets

Figure 15: Examples of designs resulting from a PSA-type algorithm, with the radius of privacy sets $\delta = 0.2$. The efficiencies with respect to D-optimal approximate designs are 98.48% for $N = 11$ design points (Fig. 15a) and 99.9% for $N = 18$ design points (Fig. 15b). Blue squares represent vertices of the so-called Voronoi diagram, which plays a crucial role in this implementation of PSA and can be efficiently computed e.g. by Bowyer-Watson incremental algorithm.
Benkova’s Lemma

**Lemma**

Let \( \xi \) be a permissible minimum-distance design with privacy sets \( P(x) = \{ y : \rho_2(x, y) \leq \delta \} \). Let \( x \notin \xi \) be a permissible point in \( X \). Then, there exists a Voronoi vertex \( v \) such that the whole line segment \( \overline{vx} \) is permissible.

This Lemma justifies the use of Voronoi vertices when sampling from the permissible area \( X \setminus P(\xi) \).

Efficient Delauny triangulation can be used to find the candidate set, but...
The same again: Happy Birthday Luc!

Figure 16: Participants of mODa9, Bertinoro, 2010.
Stratified designs from privacy sets

introduced in Harman, 2014 as a generalization of *marginally restricted designs*: Let $\mathcal{X}_1, \ldots, \mathcal{X}_k$ be a decomposition of $\mathcal{X}$ into non-empty non-overlapping partitions - strata. Let $\mathcal{X}_j \cap \xi \leq 1$ for $j = 1, \ldots, k$. For $x \in \mathcal{X}_j$, the privacy set $P(x)$ is then given by $P(x) = \mathcal{X}_j$.

Figure 17: An illustration of privacy sets of a stratified design with 3 strata. All the points in the “blue” stratum have the same privacy set $P(x)$; the same holds for the points in the “red” stratum and the points in the “green” stratum.
Conclusions

- The notion of privacy sets is central to the understanding and interpretation of “hard” space-filling.

- The Privacy sets algorithm based on this notion is very flexible and can be used in a great variety of situations to compute efficient designs.

- For small-size problems, PSA can be straightforwardly applied for all kinds of privacy sets.

- For large-size problems, specific particularities of the privacy constraints should be taken into account. This can be efficiently done e.g. for bridge designs, where our algorithm performs significantly better than the state-of-the-art method.

- **Future research (almost done):** Adapt PSA for another type of privacy sets - for example based on minimum distance.
One more time: Happy Birthday Luc!

Figure 18: Participants of mODa10, Lagow Lubuski, 2013.
References


One last time: Happy Birthday Luc and thank you for your attention!

Figure 19: Participants of mODa11, Hamminkeln-Dingden, 2016.