A brief introduction to the R library OptimalDesign

Radoslav Harman, Lenka Filová

February 9, 2020

Abstract

The R library OptimalDesign ([7]) provides a toolbox for computing efficient approximate and exact designs of regression experiments with uncorrelated observations[1]. This text presumes the general knowledge of the theory of optimal design of statistical experiments, see, e.g., the monographs [1], [2], [13]. Here we only provide an overview of the philosophy and some mathematical details of the library.

1 Design space

The current version of the library works with a design space $\mathfrak{X}$ of a finite size n.[2] We assume that we have an implicit ordering of "candidate" design points $x_1, \ldots, x_n$, each representing particular experimental conditions under which we can perform a "trial"[3], possibly multiple times. The design points typically correspond to all permissible combinations of the levels of discrete factors. The general aim is to construct an "efficient design", which is, abstractly put, a synergistic multi-set of design points selected from the candidates (a more concrete definition is in Section 10).

2 Matrix of candidate regressors

The central object of the library is the matrix $\mathbf{F} = (\mathbf{f}_1, \ldots, \mathbf{f}_n)' \in \mathbb{R}^{n \times m}$ (denoted by Fx in the codes) of all candidate regressors, representing the optimal design problem.

If we consider the standard linear regression model $y = \mathbf{f}'(x)\theta + \epsilon$ with the vector θ of parameters, where, for each trial, x can be selected from $\mathfrak{X}$, then

[1] The library OptimalDesign can also be used to for various other problems equivalent to optimal design of experiments such as for computing the minimum-volume data-enclosing ellipsoids. However, in this short guide we only focus on designing experiments.

[2] The design space is finite but it can be very large, up to millions of design points for some procedures. Moreover, using various simple strategies, the solvers for optimal designs on a finite design space can be used to compute efficient designs on continuous design spaces.

[3] "Trials" are sometimes referred to as "observations" or "measurements".

$\mathbf{f}_i = \mathbf{f}(x_i)$. For the application to the design of non-linear regression models $y = \eta(x, \theta) + \epsilon$, the $\mathbf{f}_i$ is the gradient of the function $\eta(x_i, \cdot)$ with respect to θ, evaluated in some nominal parameter value θ_0[4]. We assume that $\mathbf{F}$ has the full column rank, i.e., that $\mathbf{F}'\mathbf{F}$ is non-singular.

Note that in the package we assume that $m \geq 2$, because the optimal design problems with $m = 1$ tend to be trivial or directly equivalent to a problem of (possibly integer) linear programming.

To compute $\mathbf{F}$ for the most common linear and nonlinear regression models, the library `OptimalDesign` provides functions `Fx_cube`, `Fx_simplex`, `Fx_blocks`, `Fx_glm`, `Fx_dose`, `Fx_survival`. See the help pages of these functions for details.

3 Design

In our library, the design is formally a non-negative vector $\mathbf{w} = (w_1, \ldots, w_n)'$. Depending on the application, the interpretation of w_i is the approximate (possibly non-integer) or exact (integer) number of replications of independent trials in x_i. In the former case, $\mathbf{w}$ is called an "approximate" design, in the latter case it is called an "exact" design[5]. The value $\sum_i w_i$ is the "size" of the experiment, i.e., the total number of trials. The component w_i can also represent the relative proportions of the trials in x_i; then $\sum_i w_i = 1$ and $\mathbf{w}$ is called a "normalized" approximate design.

4 Information matrix

For any design $\mathbf{w} = (w_1, \ldots, w_n)' \in [0, \infty)^n$, the information matrix is defined as

$$\mathbf{M}(\mathbf{w}) = \sum_{i=1}^{n} w_i \mathbf{f}_i \mathbf{f}_i' = \mathbf{F}'\mathrm{diag}(\mathbf{w})\mathbf{F}.$$

The information matrix is a non-negative definite $m \times m$ matrix representing the information about the parameter gained from the experiment designs according to $\mathbf{w}$. For an efficient computation of the information matrix, the library provides the function `infmat`.

[4] The so-called "local" approach.

[5] That is, formally speaking, exact design is a specific approximate design

5 Standard optimality criteria

The library implements the standard criteria of D-optimality, A-optimality, and c-optimality[6], in the following forms.

$$\begin{aligned}
\phi_{D,\mathbf{F}}(\mathbf{w}) &= \det[\mathbf{M}(\mathbf{w})]^{1/m}; &(1)\\
\phi_{A,\mathbf{F}}(\mathbf{w}) &= m/\operatorname{tr}[\mathbf{M}^{-1}(\mathbf{w})] \text{ if } \mathbf{M}(\mathbf{w}) \text{ is non-singular, 0 otherwise;} &(2)\\
\phi_{c,\mathbf{F}}(\mathbf{w}) &= \|\mathbf{h}\|^2/\,\mathbf{h}'\mathbf{M}^{-}(\mathbf{w})\mathbf{h} \text{ if } \mathbf{h} \in \mathcal{C}(\mathbf{M}(\mathbf{w})), 0 \text{ otherwise,} &(3)
\end{aligned}$$

where $\mathbf{h}$ is a user-defined non-zero vector and $\mathbf{M}^{-}$ is any generalized inverse of $\mathbf{M}$. All these criteria depend on $\mathbf{w}$ only via $\mathbf{M}(\mathbf{w})$. These criteria are concave, positive homogeneous, and normalized in such a way that they assign 1 to any $\mathbf{w}$ with $\mathbf{M}(\mathbf{w}) = \mathbf{I}_m$. See the function `optcrit`.

6 Derived optimality criteria

The library also implements two criteria derived from A-optimality: I-optimality and C-optimality (i.e., a regularized c-optimality).

$$\begin{aligned}
\phi_{I,\mathbf{F}}(\mathbf{w}) &= \phi_{A,\mathbf{F}\mathbf{R}_{\mathbf{F}}^{-1}}(\mathbf{w}) \text{ if } \mathbf{M}(\mathbf{w}) \text{ is non-singular, 0 otherwise;} &(4)\\
\phi_{C,\mathbf{F}}(\mathbf{w}) &= \phi_{A,\mathbf{F}\mathbf{H}_{\alpha}^{-1}}(\mathbf{w}) \text{ if } \mathbf{M}(\mathbf{w}) \text{ is non-singular, 0 otherwise.} &(5)
\end{aligned}$$

Above, the matrices $\mathbf{R}_{\mathbf{F}}$ and $\mathbf{H}_{\alpha}$, $\alpha \in [0, 1)$, are defined so that

$$\begin{aligned}
\mathbf{R}_{\mathbf{F}}'\mathbf{R}_{\mathbf{F}} &= \frac{m}{\sum_{i=1}^{n}\|\mathbf{f}_i\|^2}\sum_{i=1}^{n}\mathbf{f}_i\mathbf{f}_i',\\
\mathbf{H}_{\alpha}'\mathbf{H}_{\alpha} &= \alpha\mathbf{I}_m + \frac{(1-\alpha)m\mathbf{h}\mathbf{h}'}{\|h\|^2}.
\end{aligned}$$

where $\mathbf{h}$ is a user-defined non-zero vector. These criteria are also concave, positive homogeneous, and they are normalized in such a way that they assign 1 to any $\mathbf{w}$ with $\mathbf{M}(\mathbf{w}) = \mathbf{I}_m$.

It is possible to show that for a non-singular $\mathbf{M}(\mathbf{w})$ we have

$$\begin{aligned}
\phi_{I,\mathbf{F}}(\mathbf{w}) &= \|\mathbf{F}\|^2\left(\sum_{i=1}^{n}\mathbf{f}_i'\mathbf{M}^{-1}(\mathbf{w})\mathbf{f}_i\right)^{-1} &(6)\\
\phi_{C,\mathbf{F}}(\mathbf{w}) &= m\left(\operatorname{tr}\left[\alpha\mathbf{M}^{-1}(\mathbf{w}) + \frac{(1-\alpha)m}{\|h\|^2}\mathbf{M}^{-1}(\mathbf{w})\mathbf{h}\mathbf{h}'\right]\right)^{-1}. &(7)
\end{aligned}$$

[6] D-optimality aims at constructing designs minimizing the volume of the confidence ellipsoid for the vector of model parameters, A-optimality minimizes the sum of variance of the BLUEs of all model parameters and c-optimality minimizes the variance of the BLUE of a linear combination of the model parameters (the coefficients of the linear combinations are given by a vector $\mathbf{h}$). See the referenced monographs for more details.

I-optimality is a well-known and very useful criterion (e.g., [1]). The expression 6 implies that I-optimality is aimed at constructing designs that minimize the sum of the variances of the BLUEs of estimators of the mean responses in $x_1, \ldots, x_n$. Note also that many of generalizations of I-optimality can also be easily converted to A-optimality and the corresponding optimal designs can be computed using the procedures of this library.

On the other hand C-optimality is not yes an established criterion. The expression 6 shows that C-optimality is constructed with the aim to produce a non-singular optimal design efficient with respect to the criterion of c-optimality (the true c-optimal designs are often singular, i.e, dangerous to use). We use the fixed value of $\alpha = 0.05$ and do not yet have theoretical underpinnings of the criterion, although the criterion seems to work reasonably well.

See the functions `optcrit`, `Fx_ItoA`, and `Fx_CtoA`.

7 Directional derivatives

Directional derivatives of the criteria provide a useful inspection tool. We use the following forms of the directional derivatives of ϕ at the normalized design $\mathbf{w}$ (with a non-singular $\mathbf{M}(\mathbf{w})$) in the direction of the singular designs $\mathbf{e}_i$.

$$\partial\phi_{D,\mathbf{F}}(\mathbf{w}, \mathbf{e}_i) = \frac{\det(\mathbf{M}(\mathbf{w}))^{1/m}}{m}\left[\mathbf{f}_i'\mathbf{M}^{-1}(\mathbf{w})\mathbf{f}_i - m\right];$$

$$\partial\phi_{A,\mathbf{F}}(\mathbf{w}, \mathbf{e}_i) = \frac{m}{\mathrm{tr}^2(\mathbf{M}^{-1}(\mathbf{w}))}\left[\mathbf{f}_i'\mathbf{M}^{-2}(\mathbf{w})\mathbf{f}_i - \mathrm{tr}(\mathbf{M}^{-1}(\mathbf{w}))\right];$$

$$\partial\phi_{c,\mathbf{F}}(\mathbf{w}, \mathbf{e}_i) = \frac{\|\mathbf{h}\|^2}{(\mathbf{h}'\mathbf{M}^{-1}(\mathbf{w})\mathbf{h})^2}\left[(\mathbf{f}_i'\mathbf{M}^{-1}(\mathbf{w})\mathbf{h})^2 - \mathbf{h}'\mathbf{M}^{-1}(\mathbf{w})\mathbf{h}\right].$$

The directional derivatives for the criteria of I- and C-optimality follow from the definitions (6) and (7).

See the function `dirder`.

8 Variance functions

"Variance functions" are closely related to the directional derivatives. Since we have an n-point design space, we can represent the variance functions by vectors of length n. The i-th component of the variance functions depends on the criterion and we define it as follows: $(v_{D,\mathbf{F}}(\mathbf{w}))_i = \mathbf{f}_i'\mathbf{M}^{-1}(\mathbf{w})\mathbf{f}_i$, $(v_{A,\mathbf{F}}(\mathbf{w}))_i = \mathbf{f}_i'\mathbf{M}^{-2}(\mathbf{w})\mathbf{f}_i$, and $(v_{c,\mathbf{F}}(\mathbf{w}))_i = (\mathbf{f}_i'\mathbf{M}^{-1}(\mathbf{w})\mathbf{h})^2$. See the function `varfun`.

9 Design constraints

Performing a trial incurs costs; therefore, the set $\mathcal{W}$ of practically permissible designs is always restricted.[7] Often, we only have the size of the design constrained by some number N.[8] Sometimes, however, we have more complex design constraints. Several procedures of the library allow for setting multiple linear constraints $\mathbf{A}\mathbf{w} \leq \mathbf{b}$, where $\mathbf{A}$ is a $k \times n$ matrix and $\mathbf{b}$ is a $k \times 1$ vector[9]. These can be either general linear constraints or the "resource" linear constraints, depending on the procedure. If the constraints are of the resource type, $\mathbf{A}$ has non-negative elements (but no column is $\mathbf{0}_k$), and $\mathbf{b}$ has positive elements. Many practical constraints are of the resource type and their special form makes it easier to handle for some optimal design algorithms. See the next section and Table 1.

10 Optimal designs

The aim of optimal experimental design is to maximize the value of the criterion of optimality ϕ over the set $\mathcal{W}$ of all permissible designs, thus leading to an optimal design $\mathbf{w}^*$. This can be formulated as an optimization problem

$$\begin{array}{rl} \max_{\mathbf{w}} & \phi(\mathbf{w}) \\ \text{subject to} & \mathbf{w} \in \mathcal{W}. \end{array} \tag{8}$$

The primary purpose of the library is to compute optimal designs with a general matrix $\mathbf{F}$ of candidate regressors and various forms of $\mathcal{W}$ as indicated in Table 1. The available procedures are `od_REX` (cf. [8] and [9]), `od_KL` ([1]), `od_MISOCP` ([16]), `od_RC` ([6]), and `od_AQUA` ([5] and [3]).

Note that some of the functions also allow for setting a separate lower bound $\mathbf{w}_0$ on $\mathbf{w}$ which can represent the design to be optimally augmented.

Some of the functions (`od_MISOCP`, `od_AQUA`) require the gurobi solver ([4]). The solver is commercial, but it is very simple to obtain a free academic licence.

11 Efficiency of a design

Let $\mathbf{w}^*$ be a ϕ-optimal design within $\mathcal{W}$ and let ϕ be a concave, positive homogeneous criterion. The efficiency of a design $\mathbf{w}$ is defined as $\phi(\mathbf{w})/\phi(\mathbf{w}^*)$. To

[7] Please, note the difference between restrictions on the set $\mathfrak{X}$ of design points and restrictions on the space of designs on $\mathfrak{X}$. Computing efficient designs on a restricted $\mathfrak{X}$ is usually not much different from computing efficient designs on the original, full, $\mathfrak{X}$. On the other hand, computing the efficient design within a functionally constrained class of design on $\mathfrak{X}$ may be a fundamentally harder problem.

[8] If, for instance, each trial costs the same amount of money and we have a fixed budget (and there are no constraints on other resources).

[9] For the sake of simplicity, some of the functions of the library allow for setting $\mathbf{A}\mathbf{w} \leq b$ in the form of simultaneous constraints $\mathbf{A}_1\mathbf{w} \leq \mathbf{b}_1$, $\mathbf{A}_2\mathbf{w} \geq \mathbf{b}_2$, and $\mathbf{A}_3\mathbf{w} = \mathbf{b}_3$.

$\mathcal{W}$	criteria	function
$\mathbf{w} \in [0,\infty)^n : \sum_i w_i = 1$	D, A, I, C, c	`od_REX`
$\mathbf{w} \in \mathbb{N}_0^n : \sum_i w_i = N$	D, A, I, C	`od_KL`
$\mathbf{w} \in \{0,1\}^n : \sum_i w_i = N$	D, A, I, C	`od_KL` (with `bin=TRUE`)
$\mathbf{w} \in [0,\infty)^n : \mathbf{A}\mathbf{w} \leq \mathbf{b}$ general	D, A, I, C, c	`od_MISOCP`
$\mathbf{w} \in \mathbb{N}_0^n : \mathbf{A}\mathbf{w} \leq \mathbf{b}$ resource	D, A, I, C	`od_RC`
$\mathbf{w} \in \mathbb{N}_0^n : \mathbf{A}\mathbf{w} \leq \mathbf{b}$ general, smaller n	D, A, I, C, c	`od_MISOCP`
$\mathbf{w} \in \mathbb{N}_0^n : \mathbf{A}\mathbf{w} \leq \mathbf{b}$ general, larger n	D, A, I, C	`od_AQUA`

Table 1: Recommended functions to compute optimal design within various sets of permissible designs.

compute the actual efficiency of $\mathbf{w}$ we need the optimal design $\mathbf{w}^*$ (or at least $\phi(\mathbf{w}^*)$), which may be unavailable. However, in some situations we can quickly compute a lower bound on the efficiency of $\mathbf{w}$ (with a non-singular $\mathbf{M}(\mathbf{w})$). For instance if $\mathcal{W}$ is a set of designs with the same size as $\mathbf{w}$ then such a lower bounds are (based on the criterion):

$$\frac{\phi_D(\mathbf{w})}{\phi_D(\mathbf{w}_D^*)} \geq \frac{m}{\max_i \mathbf{f}'(x_i)\mathbf{M}^{-1}(\mathbf{w})\mathbf{f}(x_i)};$$

$$\frac{\phi_A(\mathbf{w})}{\phi_A(\mathbf{w}_A^*)} \geq \frac{\mathrm{tr}(\mathbf{M}^{-1}(\mathbf{w}))}{\max_i \mathbf{f}'(x_i)\mathbf{M}^{-2}(\mathbf{w})\mathbf{f}(x_i)};$$

$$\frac{\phi_c(\mathbf{w})}{\phi_c(\mathbf{w}_c^*)} \geq \frac{\mathbf{h}'\mathbf{M}^{-1}(\mathbf{w})\mathbf{h}}{\max_i(\mathbf{h}'\mathbf{M}^{-1}(\mathbf{w})\mathbf{h})^2}.$$

See the function `effbound`.

12 Conclusions

This text is constantly evolving and still does not cover the `OptimalDesign` functions `od_PIN` ([11]), `od_DEL` ([10], [12]), `od_PUK` ([14]), `od_SYM` (Subsection 5.1 in [8]), `od_pool`, `od_print`, `od_plot`. Do not hesitate to let us know your suggestions for improvements.

References

[1] Atkinson AC, Donev AN, Tobias RD (2007): *Optimum Experimental Designs, With SAS*. Oxford University Press.

[2] Fedorov V, Hackl P (1997) *Model-oriented design of experiments (lecture notes in statistics).* Springer, Berlin

[3] Filova L., Harman R. (2020). Ascent with Quadratic Assistance for the Construction of Exact Experimental Designs. To appear in *Computational*

Statistics, DOI: https://doi.org/10.1007/s00180-020-00961-9, arXiv preprint: https://arxiv.org/abs/1801.09124

[4] Gurobi Optimization, Inc. (2017): Gurobi Optimizer Reference Manual, http://www.gurobi.com

[5] Harman R, Filová L (2014): "Computing efficient exact designs of experiments using integer quadratic programming", *Computational Statistics & Data Analysis*, Vol 71, pp 1159-1167

[6] Harman R, Bachratá A, Filová L (2016): "Construction of efficient experimental designs under multiple resource constraints", *Applied Stochastic Models in Business and Industry* 32/1, pp. 3-17

[7] Harman R, Filová L (2017): R package OptimalDesign, https://CRAN.R-project.org/package=OptimalDesign

[8] Harman R, Filová R, Richtárik P (2020): A randomized exchange algorithm for computing optimal approximate designs of experiments, to appear in *Journal of the American Statistical Association*, DOI: https://doi.org/10.1080/01621459.2018.1546588, arXiv preprint: https://arxiv.org/abs/1801.05661

[9] Harman R, Jurík T (2008): Computing c-optimal experimental designs using the simplex method of linear programming, *Computational Statistics & Data Analysis* 53 (2), 247-254

[10] Harman R, Pronzato L (2007): Improvements on removing nonoptimal support points in D-optimum design algorithms, *Statistics & Probability Letters* 77, 90-94

[11] Harman R, Rosa S (2020): On greedy heuristics for computing D-efficient saturated subsets, to appear in *Operations Research Letters*, DOI: https://doi.org/10.1016/j.orl.2020.01.003, arXiv preprint: https://arxiv.org/abs/1905.07647

[12] Pronzato L (2013): A delimitation of the support of optimal designs for Kiefers Φ_p-class of criteria. *Statistics & Probability Letters* 83, 2721-2728

[13] Pukelsheim F (2006): *Optimal Design of Experiments*. SIAM.

[14] Pukelsheim F, Rieder S (1992): "Efficient rounding of approximate designs", *Biometrika*, Vol. 79, pp. 763-770

[15] R Core Team (2016): "R: A language and environment for statistical computing", R Foundation for Statistical Computing, Vienna, Austria, https://www.R-project.org/

[16] Sagnol G, Harman R (2015): "Computing exact D-optimal designs by mixed integer second order cone programming", *The Annals of Statistics* 43 (5), 2198-2224