# Kirstine.jl: A Julia Package for Bayesian Optimal Design of Experiments

Ludger Sandig [1]

1 Department of Statistics, TU Dortmund University, Germany

Summary

Good design of planned experiments increases the precision of parameter estimates. In a dose-response study, for example, the credible intervals for the response curve's parameters are shorter when the dose levels have been chosen carefully. The general mathematical framework that guides the choice of values and weights for the covariates in a regression model is called optimal design theory. In the special case of nonlinear regression models, a researcher must specify prior knowledge about the model parameters. Accounting here for a full distribution of parameter values produces designs that are more robust than designs for a single best guess. To help compute such designs efficiently for general nonlinear regression models, we propose the `Julia` (Bezanson et al., 2017) package `Kirstine.jl`.

Mathematical Background

Consider a nonlinear regression model with mean function $\mu : X \times \Theta \to \mathbb{R}^m$, known covariance matrix Σ, and compact *design region* $X \subset \mathbb{R}^d$, where we want to design an experiment for estimating the unknown parameter θ. In nonlinear optimal design theory (Fedorov & Leonov, 2013), we represent the design by a probability measure ξ on X. For every such *design measure* we define the *normalized information matrix*

$$\mathrm{M}(\xi, \theta) = \int (\mathrm{D}_\theta \mu(x, \theta))' \Sigma^{-1} (\mathrm{D}_\theta \mu(x, \theta)) \, \xi(\mathrm{d}x),$$

where $\mathrm{D}_\theta \mu$ denotes the Jacobian matrix of μ with respect to θ. To obtain, on average, small confidence or posterior credible intervals, we aim to construct a design ξ^* that maximizes a functional ϕ of the normalized information matrix, with popular choices being the *D-* or *A-criterion*

$$\phi_{\mathrm{D}}(\mathrm{M}(\xi, \theta)) = \log \det(\mathrm{M}(\xi, \theta)), \quad \phi_{\mathrm{A}}(\mathrm{M}(\xi, \theta)) = -\mathrm{tr}(\mathrm{M}(\xi, \theta)^{-1}).$$

Since $\mathrm{M}(\xi, \theta)$ still depends on the unknown θ, we either plug in a best guess θ_0 and obtain a *locally optimal* design problem, or we try to find a *Bayesian optimal* design that maximizes the average of ϕ with respect to a prior distribution with density $p : \Theta \to [0, \infty)$ (Chaloner & Verdinelli, 1995). Having obtained a candidate design ξ^*, we then apply an equivalence theorem from infinite-dimensional convex analysis to verify that the design ξ^* is indeed optimal. The setup above can be generalized to designs that are optimal for estimating a transformed $T(\theta)$, or to models where Σ also depends on x.

To find a candidate design in practice, we must make three simplifications. We first have to approximate the prior expectation, since the integral is not tractable analytically. *Monte-Carlo (MC) integration*

$$\int \phi(\mathrm{M}(\xi, \theta)) \, p(\theta) \, \mathrm{d}\theta \approx \frac{1}{S} \sum_{s=1}^{S} \phi(\mathrm{M}(\xi, \theta^{(s)}))$$

is a versatile method for that because we can use it with any p from which we can draw a sample $\theta^{(1)}, \dots, \theta^{(S)}$, $S \in \mathbb{N}$. Next, we reduce the search space from all probability measures on X to the subset of those that are discrete and have $K \in \mathbb{N}$ design points. The optimal K^* is usually not known beforehand, but as long as we do not enforce unique design points, we may guess at a $K > K^*$ and will still be able to find the solution. Finally, we have to choose one of the many proposed algorithms (Ryan et al., 2015) for maximizing the objective function numerically.

Statement of Need

Currently, most open-source experimental design software is implemented in R. There is also a Julia package that focuses on factorial design problems but does not address nonlinear regression. Among the R packages on CRAN, only four deal with nonlinear regression models, and all of them have to make a tradeoff between speed and flexibility. With MC integration, thousands of information matrices, each built from K Jacobian matrices $\mathrm{D}_\theta \mu$, have to be computed for one evaluation of the objective function. In R, these matrix-valued functions are a performance bottleneck since each call has to allocate new matrix objects. To avoid the memory overhead, package authors can implement internals in C and pass around pointers to pre-allocated matrices. However, this requires the users to be proficient in C in order to supply the Jacobian matrices of their models. Consequently, packages either just accept the slowdown (Masoudi et al., 2020), recommend using C++ (Overstall et al., 2020), or come with a small set of models pre-specified (Bornkamp et al., 2023; Foracchia et al., 2004; Nyberg et al., 2012). Hence a design package is needed where knowledge of only one language is required for efficiently implementing arbitrary nonlinear regression models.

Kirstine.jl attempts to fill this gap in the design software ecosystem. The package achieves modeling flexibility through Julia's multiple dispatch mechanism, and performs matrix operations efficiently by passing object references to BLAS and LAPACK routines. It currently implements the D- and A-criterion, vector-valued measurements, posterior transformations of θ via the Delta method, box-shaped design regions of arbitrary dimension, particle swarm optimization (Kennedy & Eberhart, 1995), and a variant of Fedorov's coordinate exchange algorithm (Yang et al., 2013). Plotting functions for checking the equivalence theorem are also provided. Locally optimal design is supported implicitly. Since user-defined Julia code does not inherently incur performance penalties, specific regression models are not supplied. Instead, users should first define subtypes of NonlinearRegression, Covariate, Parameter, and CovariateParameterization, and then add methods for a few functions that dispatch on them. Optionally, one of Julia's automatic differentiation packages can be used. Kirstine.jl has a modular and readable code base, which enables users to extend the package's functionality with drop-in replacements for new criteria, design regions, or even custom optimization algorithms. Thanks to multiple dispatch, no changes are required in the package internals. This way, Kirstine.jl provides an additional level of flexibility without sacrificing efficiency.

Acknowledgments

References

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, *59*(1), 65–98. https://doi.org/10.1137/141000671

Bornkamp, B., Pinheiro, J., Bretz, F., & Sandig, L. (2023). *DoseFinding: Planning and analyzing dose finding experiments*. https://CRAN.R-project.org/package=DoseFinding

Chaloner, K., & Verdinelli, I. (1995). Bayesian experimental design: A review. *Statistical Science*, *10*(3), 273–304. https://doi.org/10.1214/ss/1177009939

Fedorov, V. V., & Leonov, S. L. (2013). *Optimal design for nonlinear response models*. CRC Press. https://doi.org/10.1201/b15054

Foracchia, M., Hooker, A. C., Vicini, P., & Ruggeri, A. (2004). POPED, a software for optimal experiment design in population kinetics. *Computer Methods and Programs in Biomedicine*, *74*. https://doi.org/10.1016/s0169-2607(03)00073-7

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, *4*, 1942–1948. https://doi.org/10.1109/icnn.1995.488968

Masoudi, E., Holling, H., & Wong, W. K. (2020). *ICAOD: Optimal designs for nonlinear statistical models by imperialist competitive algorithm (ICA)*. https://CRAN.R-project.org/package=ICAOD

Nyberg, J., Ueckert, S., Stroemberg, E. A., Hennig, S., Karlsson, M. O., & Hooker, A. C. (2012). PopED: An extended, parallelized, nonlinear mixed effects models optimal design tool. *Computer Methods and Programs in Biomedicine*, *108*. https://doi.org/10.1016/j.cmpb.2012.05.005

Overstall, A. M., Woods, D. C., & Adamou, M. (2020). acebayes: An R package for Bayesian optimal design of experiments via approximate coordinate exchange. *Journal of Statistical Software*, *95*(13), 1–33. https://doi.org/10.18637/jss.v095.i13

Ryan, E. G., Drovandi, C. C., McGree, J. M., & Pettitt, A. N. (2015). A review of modern computational algorithms for Bayesian optimal design. *International Statistical Review*, *84*(1), 128–154. https://doi.org/10.1111/insr.12107

Yang, M., Biedermann, S., & Tang, E. (2013). On optimal designs for nonlinear models: A general and efficient algorithm. *Journal of the American Statistical Association*, *108*(504), 1411–1420. https://doi.org/10.1080/01621459.2013.806268