Received: 9 January 2022 | Revised: 27 May 2022 | Accepted: 3 June 2022

DOI: 10.1002/qre.3165

SPECIAL ISSUE ARTICLE

WILEY

Large datasets, bias and model-oriented optimal design of experiments

Elena Pesce[1] | **Francesco Porro**[2] | **Eva Riccomagno**[2]

[1]Swiss Re Institute, Zurich, Switzerland

[2]Department of Mathematics, University of Genova, Genova, Italy

Correspondence
Eva Riccomagno, Department of Mathematics, University of Genova, Genova, Italy.
Email: riccomagno@dima.unige.it

Abstract

We review recent literature that proposes to adapt ideas from classical model based optimal design of experiments to problems of data selection of large datasets. Special attention is given to bias reduction and to protection against confounders. Some new results are presented. Theoretical and computational comparisons are made.

KEYWORDS

confounders, large datasets, model bias, optimal experimental design

1 | INTRODUCTION

For the analysis of big datasets, statistical methods have been developed, which use the full available dataset. For example, new methodologies developed in the context of Big Data and focussed on a 'divide-and-recombine' approach are summarised in Wang et al.[19] Other major methods address the scalability of Big Data through Bayesian inference based on a Consensus Monte Carlo algorithm[13] and sparsity assumptions.[16]

In contrast, other authors argue on the advantages of inference statements based on a well-chosen subset of the large dataset. Big datasets are characterised by few key factors. While usually data can be collected in scientific studies via active or passive observation, Big Data is often collected in passive way. Rarely their collection is the result of a designed process. This generates sources of bias, which either we do not know at all or are too costly to control. Nevertheless they will affect the overall distribution of the observed variables.[3,11]

Many authors in Ref.[15] argues that analysis of big dataset is effected by issues of bias and confounding, selection bias and other sampling problems (see, for example, Sharpes[14] for electronic health records). Often the causal effect of interest can only be measured on the average and great care has to be taken about the background population, for example, it is possible to consider and analyse every message on Twitter and use it to draw conclusions about the public opinion, but it is known that Twitter users are not representative of the whole population. The analysis of the full dataset might be prohibitive because of computational and time constraints. Indeed, in some cases, the analysis of the full dataset might also be not advisable.[4,6] To recall just one example, where the sample proportion of a self-reported big dataset of size 2300,000 unit has the same mean squared error as the sample proportion from a suitable simple random sample (SRS) of size 400 and a Law of Large Population has been defined in order to qualify this (see Meng[9]).

Recently, some researchers argued on the usefulness of utilising methods and ideas from design of experiment (DoE) for the analysis of big datasets, more specifically from model-based optimal experimental design. They argue that special

models are useful, or even needed, to guard against hidden sources of bias and that a well-chosen subset of the big dataset can deliver equivalent answers compared to the full dataset at considerably less effort. In this paper, we propose a unified general framework, which encompass the three main situations that can occur.

The remaining of the paper is organized as follows. Section 2 sketches the general theoretical framework, describes three model classes embedding the main situations that can occur (models without bias, models with bias and no confounders, and models with confounders and no bias) and describes main existing algorithms for optimal sub-sample selection based on those model classes. Section 3 provides a simulation study and the last section concludes. Theoretical and computational comparisons made using the software R lead us to conclude that so far these approaches are more suitable for tall dataset than for genuine large datasets and indicate that much work is needed to have efficient algorithms for sub-sample selection from large datasets in the presence of bias and confounders. To fix terminology, we recall that a dataset is tall if the number of observations is much larger than the number of predictors, and large when it has many observations and predictors.

2 | MODEL-ORIENTED SELECTION OF SUB-DATASETS

The most general form of the considered model is that of a linear model for a response variable $\boldsymbol{Y}$

$$\boldsymbol{Y}(\boldsymbol{x},\boldsymbol{z}) = \boldsymbol{f}'(\boldsymbol{x})\boldsymbol{\theta} + \boldsymbol{h}'(\boldsymbol{x})\boldsymbol{\psi} + \boldsymbol{g}'(\boldsymbol{z})\boldsymbol{\phi} + \epsilon \tag{1}$$

with $\boldsymbol{\theta} \in \mathbb{R}^p, \boldsymbol{\psi} \in \mathbb{R}^m$ and $\boldsymbol{\phi} \in \mathbb{R}^q$ and with $\boldsymbol{x} \in \mathcal{X}, \boldsymbol{z} \in \mathcal{Z}$. The observed values are the $\boldsymbol{x}$, while the $\boldsymbol{z}$ are from unmeasured (or unmeasurable) random variables. Both the $\mathcal{X}$ and $\mathcal{Z}$ spaces are assumed to be compact in the Euclidean topology, and $'$ indicates transpose. The usual assumptions are taken on the random errors: the ϵ_i's are independent and identically distributed with zero mean and variance $\mathrm{Var}(\epsilon_i) = \sigma^2$.

There are three terms in the model: $\boldsymbol{f}'(\boldsymbol{x})\boldsymbol{\theta}$ corresponds to a classical linear model, $\boldsymbol{h}'(\boldsymbol{x})\boldsymbol{\psi}$ to a bias term related to the variables $\boldsymbol{x}$ and $\boldsymbol{g}'(\boldsymbol{z})\boldsymbol{\phi}$ models a bias that may result from confounders, sources of bias, which either we do not know at all or are too costly to control. We assume it to be linear for simplicity of comparison. Special cases of the Model in Equation (1) have been addressed in order to adapt ideas from classical model-based optimal DoE: Montepiedra and Fedorov[10] and Wiens[21] consider

$$\boldsymbol{Y}(\boldsymbol{x}) = \boldsymbol{f}'(\boldsymbol{x})\boldsymbol{\theta} + \boldsymbol{h}'(\boldsymbol{x})\boldsymbol{\psi} + \epsilon \tag{2}$$

while Pesce et al.[11] considers

$$\boldsymbol{Y}(\boldsymbol{x},\boldsymbol{z}) = \boldsymbol{f}'(\boldsymbol{x})\boldsymbol{\theta} + \boldsymbol{g}'(\boldsymbol{z})\boldsymbol{\phi} + \epsilon. \tag{3}$$

The algorithms we report search for a design, which minimises the mean square error (MSE) of the least square estimate (LSE) of the $\boldsymbol{\theta}$ parameters, guarding against the two different sources of bias. Recently, authors of Refs.[1,2,17,18] proposed methods of data selection from large datasets in a DoE context, as a response to the more and more frequent need to analyse Big Data. However, they do not guard against different sources of bias.

The utility function we consider is the determinant of the information matrix after the marginalisation of the unobservable quantities. Its minimisation returns the so-called D-optimal designs. But equivalently other convex function of the information matrix could be considered. Under non-singularity of the information matrix observations and Gaussian errors, observations taken on a D-optimal design minimize the entropy of the LSEs of the unknown parameters and, from a geometrical point of view, minimize the volume of the ellipsoidal confidence interval. It can be shown that they are taken at points where the variance of the predicted response function is largest. This statement holds also in the cases of confounders and bias terms (see the Appendix).

2.1 | Model without bias

In this section, we consider the model $\mathrm{E}[Y(\boldsymbol{x})] = \boldsymbol{f}(\boldsymbol{x})'\boldsymbol{\theta}$ and the four algorithms presented in Refs.[1,2,17,18] An optimal retrospective sample set is drawn in accordance with a sampling plan or experimental design. The four algorithms are

ALGORITHM 1 Pseudocode for ROS[2]

Input: $\mathcal{D}$, $\mathcal{G}$, $\boldsymbol{f}$, U, distance function, n_t, n

Output: subset of n data points from $\mathcal{D}$

Sample randomly a subset of size $n_t < n$ from $\mathcal{D}$ and obtain $\hat{\boldsymbol{\theta}}$ or form a prior $p(\boldsymbol{\theta})$

Set the current sample size $n_c = n_t$

while $n_c \leq n$ or when a certain criterion is not met

Find the optimal design $\boldsymbol{g}^*$ such that

$$\boldsymbol{g}^* = \underset{\boldsymbol{g}_i \in \mathcal{G}}{\operatorname{argmax}}\, \mathbb{E}[U(\boldsymbol{g}_i, \hat{\boldsymbol{\theta}})] \qquad \text{or} \qquad \boldsymbol{g}^* = \underset{\boldsymbol{g}_i \in \mathcal{G}}{\operatorname{argmax}}\, \mathbb{E}[U(\boldsymbol{g}_i, p(\boldsymbol{\theta}))]$$

Find $\boldsymbol{x}_i$ in $\mathcal{D}$ not already sampled, which minimizes the distance $||\boldsymbol{x}_i - \boldsymbol{g}^*||$

Add $\{\boldsymbol{x}_i, \boldsymbol{y}(\boldsymbol{x}_i)\}$ into the data subset and remove the observation from $\mathcal{D}$

Set $n_c \leftarrow n_c + 1$

Re-estimate $\hat{\boldsymbol{\theta}}$ or update the prior distribution $p(\boldsymbol{\theta})$ **end while**

Obtain an estimate $\hat{\boldsymbol{\theta}}$ with the selected n data points

ALGORITHM 2 Pseudocode for IBOSS[17]

Input: $\mathcal{D}$, n

Output: subset of n data points from $\mathcal{D}$

Let $r = \lfloor n/(2p) \rfloor$

Initialise $\delta_i = 0$ for $i = 1, \dots, N$

for $j = 1, \dots, p$ **do**

Find the r data points with the smallest $x_i j$ values and set $\delta_i = 1$

Find the r data points with the largest $x_i j$ values and set $\delta_i = 1$

end for

Obtain an estimate of $\boldsymbol{\theta}$ with the selected n data points with $\delta_i = 1$

ALGORITHM 3 Pseudocode for OSS[18]

Input: $\mathcal{D}$, n

Output: a subset of n data points from $\mathcal{D}$

Scale values of each covariate to $[-1, 1]$

Find x_1^* = the point in $\mathcal{D}$ with the largest Euclidean norm

Add x_1^* into data subset and remove the observation from $\mathcal{D}$

for $i = 2, \dots, n$ **do**

Compute for each $x \in \mathcal{D}$ the discrepancy $l(x|x_{i-1}^*)$

Find x_i^* = the point in $\mathcal{D}$ with the smallest $l(x|x_{i-1}^*)$

Add x_i^* into data subset and remove the observation from $\mathcal{D}$

end for

targeted towards applications of regression models with large number of observations and relative small number of predictors, otherwise the problem of finding the best subset of data becomes computationally hard or infeasible due to the curse of dimensionality. We label them ROS,[2] IBOSS,[17] OSS[18] and ODB.[1] Their pseudocodes are in Algorithms 1–4, respectively. They all receive in input a tall dataset $\mathcal{D} \in \mathcal{X}$ with typical rows $\{\boldsymbol{x}_i, \boldsymbol{y}(\boldsymbol{x}_i)\}_{i=1}^N$ and the sample size of the sub-sample to be returned $n \ll N$. Their output is a subset of n data from $\mathcal{D}$ with specific properties.

We consider now each algorithm separately and highlight their differences. ROS (or retrospective optimal sampling) requires as input also the support vector of the mean values of the linear model, $\boldsymbol{f}$, a utility function U, a distance function in $\mathcal{X}$ (e.g. Euclidean), and the number of initial points to be sampled $n_t \leq n$. The output of the algorithm is a subset of $\mathcal{D}$ of size n, which maximises the given utility function U. Starting from a random subset of size n_t, the key idea behind the ROS algorithm is to find, sequentially, which point of the design space maximizes U and search for its nearest observation

ALGORITHM 4 Pseudocode for ODB[1]

Input: $\mathcal{D}, \boldsymbol{F}, \Phi[\cdot], n$

Output: subset of n data points from $\mathcal{D}$

Compute the design $\boldsymbol{\xi}^* = \underset{\boldsymbol{\xi}}{\text{argmax}}\, \Phi[M(\boldsymbol{\xi})]$ as in Equation (8) and the corresponding ideal design matrix $\boldsymbol{F}^* = \boldsymbol{F}(\boldsymbol{x}^*)$

if $n\omega_j^*$ for $j = 1, \dots, k$ is not an integer number **then**

Apply the rounding multiplier rule proposed by Fedorov[12] and obtain $\ddot{n}_j$ the updated weight for $j = 1, \dots, k$

end if

for $j = 1, \dots, k$ **do**

Compute the distance $||\boldsymbol{f}(\boldsymbol{x}_j)' - \boldsymbol{f}(\boldsymbol{x}_j^*)'||$

Let $\{d_1, d_2, \dots, d_N\}$ be the ranks of $||\boldsymbol{f}(\boldsymbol{x}_j)' - \boldsymbol{f}(\boldsymbol{x}_j^*)'||$ arranged in ascending order

Select from $\boldsymbol{F}$ the rows $d_1, \dots, d_{\ddot{n}_j}$

end for

in $\mathcal{D}$, then update the estimates of the parameters or the prior distribution (if one wants to consider a fully Bayesian experimental design framework). This is repeated until the desired sample size n is reached.

The major features of Algorithm 1 is that it returns a subset of $\mathcal{D}$ via an optimal, sequential and response adaptive procedure. The computations of the distances and the optimisation problem in the algorithm can be parallelised, thus speeding it up considerably. Parallelisation is particularly useful when the stopping criterion, the utility function and/or the distance function are costly to evaluate or when the sampling grid is large. The major drawback of Algorithm 1 is that it requires full trust in the model. Also, although it can be adapted for variable selection, the algorithm is efficient only for tall datasets. Finally, the obtained optimal design can be used as training set of, for example, a random forest.

Information-Based Optimal Sub-data Selection (IBOSS) is a deterministic algorithm to select the most informative data points for the model $\mathrm{E}[Y(\boldsymbol{x})] = \boldsymbol{f}(\boldsymbol{x})'\boldsymbol{\theta}$ with p regressors. Its pseudocode is given in Algorithm 2. The rational behind IBOSS is that D-optimal designs tend to be on the boundary of the available space, hence IBOSS tends to select data points that are on the boundary of the observed range of each covariate. The output of the algorithm is a subset of $\mathcal{D}$ of size n, which minimizes a univariate optimally criterion function $\Phi[\cdot]$ of the information matrix

$$M(\delta) = \frac{1}{\sigma^2}\sum_{i=1}^{N}\delta_i x_i x_i' \text{ subject to } \sum_{i=1}^{N}\delta_i = n$$

where σ^2 is the model variance, $\delta_i = 1$ if point $i \in \mathcal{D}$ is selected and $\delta_i = 0$ otherwise. For D-optimality, that is when Φ equals the determinant of $\boldsymbol{M}(\boldsymbol{\delta})$, in Wang et al.,[17] an upper bound on Φ is derived, which clearly suggests to collect a sub-sample with extreme covariate values.

The algorithm is very simple and computationally very fast. Its major drawback is that it requires full trust not only in the model formulation, but also on the representativeness of the response Y in $\mathcal{D}$, since it is not response adaptive. Furthermore, it is not robust with respect to permutation order of the covariates, so that different sub-samples may be obtained with different orders as shown in Figure 1 in a very simple case.

Orthogonal sub-sampling (OSS) searches a sub-sample with points with the two following features: (i) extreme values and (ii) combinatorial orthogonality (see Hedayat et al.[8]). In Wang et al.,[18] it is remarked that '...the IBOSS approach searches sub-sample points with only feature (i) in some sense without any consideration of feature (ii)'. The OSS approach minimizes the discrepancy function

$$L(X_s) = \sum_{1\le i<j\le n}\left[p - ||x_i^*||/2 - ||x_j^*||/2 + \delta(x_i^*, x_j^*)\right]^2 \tag{4}$$

which contains two parts, targeting the two features. The function $\delta(x_i^*, x_j^*)$ counts the components of x_i^* and x_j^* with the same sign in the sample X_s. The OSS sub-sample X_s is obtained by solving the optimization problem:

$$X_s^* = \arg\min_{X_s} L(X_s) s.t. X_s \text{ contains } n \text{ points.} \tag{5}$$

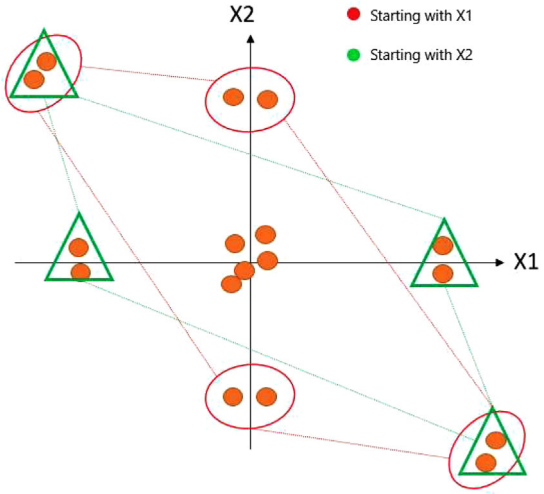

FIGURE 1 Effect of the initial ordering of the factors on the IBOSS output: in red, the design obtained starting the IBOSS algorithm with the variable x_1, in green, the design obtained starting with the variable x_2

Algorithm 3 is computationally fast enough, since at step i, it holds that

$$L(X_s^i) = l(x_i^*|X_s^{i-1}) + L(X_s^{i-1}) \tag{6}$$

where $l(x_i^*|X_s^{i-1})$ is the discrepancy introduced at step $i-1$:

$$l(x_i^*|X_s^{i-1}) = \sum_{x_j^* \in X_s^{i-1}} [p - ||x_i^*||/2 - ||x_j^*||/2 + \delta(x_i^*, x_j^*)]^2. \tag{7}$$

In the simulation studies provided in Wang et al.,[18] OSS overperforms IBOSS, also in model with interactions, because OSS selects sub-sample only based on the covariates, while IBOSS relies also on the interaction terms. A pseudocode for OSS is given in Algorithm 3.

Finally, optimal design based (ODB) is a two-step purposive selection strategy to make inferences about the parameters of the super-population model that is supposed to generate the big dataset. The design $\mathcal{D}$ is assumed to have been generated by a super-population model of the type $\boldsymbol{Y}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x})'\boldsymbol{\theta} + \boldsymbol{\varepsilon}$. The design matrix $\boldsymbol{F} = [\boldsymbol{f}(\boldsymbol{x}_1)', \ldots, \boldsymbol{f}(\boldsymbol{x}_N)']'$ and an optimality criterion $\Phi[\cdot]$ are additional inputs for ODB. The ODB output is a subset of $\mathcal{D}$ of size n, which is optimal with respect to $\Phi[\cdot]$.

The main idea is, first, to identify the theoretical most informative values of the explanatory variables according to $\Phi[\cdot]$ and then to select from the observed dataset, the data points that are closer to these theoretical optimal values. The rationale behind the ODB algorithm is that, given a super-population model, one can always compute the ideal continuous optimum design with respect to the criterion Φ as

$$\xi^* = \arg\max_{\xi} \Phi[M(\xi)] = \begin{Bmatrix} \boldsymbol{x}_1^* & \ldots & \boldsymbol{x}_j^* & \ldots & \boldsymbol{x}_k^* \\ \omega_1^* & \ldots & \omega_j^* & \ldots & \omega_k^* \end{Bmatrix} \tag{8}$$

with $0 \leq \omega_j^* \leq 1$, $\sum_{j=1}^k \omega_j = 1$ and where $M(\xi)$ is the information matrix for a continuous design ξ defined as $M(\xi) = \sum_{j=1}^k \boldsymbol{f}(\boldsymbol{x}_j)\boldsymbol{f}(\boldsymbol{x}_j)'\omega_j$. Here a design is defined by its distinct experimental points $\{\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_k^*\}$ with $k < N$.

ODB is robust to permutation order of the covariates, giving it an advantage with respect to IBOSS, and its main drawbacks are the computational time and the fact that it does not take into account misspecification in the super-population model.

In this paper, we considered only linear regression models, but the above methods can be extended to other classes of regression models. For instance, logistic regression is considered in Refs.[1,20] and softmax regression in Yao and Wang.[22]

ALGORITHM 5 Pseudocode for RA[21]

Input: $\mathcal{G}$, $\nu \in (0,1)$, loss function $\mathcal{D}_\nu$, n

Output: Minimax robust design

Let $\boldsymbol{e}_i$ be the ith column of the I_N identity matrix

Sample randomly a point $\boldsymbol{x}_i$ from $\mathcal{G}$, set the current sample size $n_c = 1$ and let $\boldsymbol{\xi}_{n_c,\boldsymbol{x}}$ be such that $\xi_i = 1$ and $\xi_j = 0$ for $j \neq i$

while $n_c \leq n$ or a certain criteria is not met **do**

Compute the matrix $T^{D_\nu}(\boldsymbol{\xi}_{n_c,\boldsymbol{x}})$

Take the largest diagonal element and assume it is in entry (i,i)

Update the weights of $\boldsymbol{\xi}_{n_c,\boldsymbol{x}}$ to $\boldsymbol{\xi}_{n_c+1,\boldsymbol{x}} = \left(\frac{n_c}{n_c+1}\right)\left(\boldsymbol{\xi}_{n_c,\boldsymbol{x}} + \frac{1}{n}\boldsymbol{e}_i\right)$

end while

2.2 | Models with no confounder terms

Author in Ref.[21] considers models of the form $Y(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x})'\boldsymbol{\theta} + \psi(\boldsymbol{x}) + \boldsymbol{\varepsilon}$, with a more general (unknown) bias term $\psi(\boldsymbol{x})$ than $\boldsymbol{h}(\boldsymbol{x})'\boldsymbol{\psi}$ in Equation (1). The main idea of the algorithm in Wiens[21] for optimal, sequential, design selection is to impose a neighbourhood structure on the standard regression response function, maximise a function of the MSE over this neighbourhood and then seek robust designs that minimize this maximum loss. The pseudocode is given in Algorithm 5. The inputs are a theoretical design space $\mathcal{G} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ based on $\mathcal{X}$ and with N distinct points, a value $\nu \in (0,1)$ chosen by the experimenter that corresponds to how much emphasis has to be given on bias reduction, a loss function $\mathcal{L}_\nu$ and the desired final size of the design n. The output is an exact n-point continuous robust design $\boldsymbol{\xi}_n = \{\xi_{n,i}\}_{i=1}^N$ placing mass $\xi_{n,i}$ on the ξ_i point of $\mathcal{G}$.

The rationale behind the algorithm is that the experimenter will compute estimates assuming that $\psi(\cdot) \equiv 0$ and protects the LSE of $\boldsymbol{\theta}$ against the bias through a minimax robust design. For given n and a control parameter τ, the bias term ψ is restricted to the following class of function Ψ:

$$\Psi = \left\{ \psi \,\middle|\, \sum_{x\in\mathcal{G}} f(x)\psi(x) = 0 \text{ and } \sum_{x\in\mathcal{G}} \psi^2(x) \leq \tau^2/n \right\} \tag{9}$$

The two conditions in Ψ ensure, respectively, the identifiability of the parameters $\boldsymbol{\theta}$ and that, asymptotically, the bias of the LSEs remains of the same magnitude as the variance. In Wiens,[21] the classical notion of D-optimality is extended to D-robustness. Letting $\hat{\boldsymbol{\theta}}$ be the LSE of $\boldsymbol{\theta}$ based on a design $\boldsymbol{\xi}$, it is proved that the maximum of the loss function

$$\mathcal{D}(\psi,\boldsymbol{\xi}) = \left(\det \mathbb{E}\left[(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})'\right]\right)^{1/p} \tag{10}$$

can be factorised as $\max_\psi \mathcal{D}(\psi,\boldsymbol{\xi}) = \frac{\sigma^2}{n}\left(\frac{\sigma^2+\tau^2}{\sigma^2 \det[\boldsymbol{F}'\boldsymbol{F}]}\right)^{1/p} \times \mathcal{D}_\nu(\boldsymbol{\xi})$ where $\mathcal{D}_\nu$ depends only on the sought design and on known quantities. Here, $\boldsymbol{F} = [\boldsymbol{f}(\boldsymbol{x})]_{\boldsymbol{x}\in\mathcal{G}}$ is the full design matrix. The objective is to find a design $\boldsymbol{\xi}^*$, which satisfies $\min_{\boldsymbol{\xi}} \max_{\psi} \mathcal{D}(\psi,\boldsymbol{\xi})$.

The algorithms presented in Section 2.1 are used in order to select n data points from an observed dataset, while with algorithm in Wiens,[21] the entire design is constructed, sequentially one point at a time, before any data are collected. Furthermore, the latter is not suitable for Big Data, especially since it requires lots of matrix multiplications. The main feature for the purpose of this approach is that it accounts for bias. Ideally, one would like to exploit Algorithm 5 in the implementation of algorithms of Section 2.1.

2.3 | Models with no bias terms

Models of the form $\boldsymbol{Y}(\boldsymbol{x},\boldsymbol{z}) = \boldsymbol{f}(\boldsymbol{x})'\boldsymbol{\theta} + \boldsymbol{g}(\boldsymbol{z})'\boldsymbol{\phi} + \boldsymbol{\varepsilon}$ have been considered in Pesce et al.[11] Let $\xi_{\boldsymbol{x},\boldsymbol{z}}$ be a design measure on $\mathcal{X}\times\mathcal{Z}$ so that the corresponding information matrix M can be rewritten as a blocked matrix

$$M(\xi) = \int_{\mathcal{X}\times\mathcal{Z}} \begin{pmatrix} \boldsymbol{f} \\ \boldsymbol{g} \end{pmatrix} (\boldsymbol{f}', \boldsymbol{g}')\, d\,\xi_{\boldsymbol{x},\boldsymbol{z}} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \tag{11}$$

Then the MSE of the LSEs of $\boldsymbol{\theta}$ is $\sigma^2 N^{-1} R$ where

$$R = M_{11}^{-1} + \left(\frac{N}{\sigma}\right)^2 M_{11}^{-1} M_{12} \boldsymbol{\phi}\boldsymbol{\phi}' M_{21} M_{11}^{-1} \tag{12}$$

and the loss function for D-optimality depends on both $\boldsymbol{x}, \boldsymbol{z}$ and it holds

$$\det(R) = \det(M_{11}^{-1}) \left(1 + \left(\frac{N}{\sigma}\right)^2 \boldsymbol{\phi}' M_{21} M_{11}^{-1} M_{12} \boldsymbol{\phi}\right) \tag{13}$$

The same approach was introduced by authors in Montepiedra and Fedorov[10] with the difference that the bias was on the $\boldsymbol{x}$ and not on confounders $\boldsymbol{z}$.

Here, we assume $\boldsymbol{g}(\boldsymbol{z})'\boldsymbol{\phi}$ to be unknown but belonging to some function class and that for each $\boldsymbol{x} \in \mathcal{X}$, there is an unobserved $\boldsymbol{z}_{\boldsymbol{x}} \in \mathcal{Z}$. Let $\boldsymbol{G} = [\boldsymbol{g}(\boldsymbol{z}_{\boldsymbol{x}})]_{\boldsymbol{x}\in\mathcal{X}}$ and P_Z be a randomization distribution for the $\boldsymbol{z}_{\boldsymbol{x}}$'s. In the game theoretical approach in Pesce et al.,[11] a D-optimal design measure is one maximising the following expected value under the randomization measure

$$\min_{P_Z} \mathbb{E}_{P_Z} \left\{ \max_{\substack{\text{function} \\ \text{class}}} \boldsymbol{G}'\boldsymbol{\phi}\boldsymbol{F} M_{11}^{-2} \boldsymbol{F}'\boldsymbol{\phi}'\boldsymbol{G} \right\} \tag{14}$$

The same arguments used in Section 2.2 for Algorithm 5 can be applied in this case.

3 | SIMULATION STUDY

We compare the algorithms presented in the previous sections on four examples. The first example compares all presented algorithm and seeks a small design with 12 points subset of a 10^3 point simulated set from a Gaussian model with mean $0.5 - 3x_1 + 4x_2$; in the second example, 10^3 points are sought out of a simulated dataset of 10^5 from the same model; in the third example, a bias term is added to the model. For comparison reasons, also a SRS has been implemented; as expected, SRS provides the worst results with respect to all the other algorithms. The last example uses a dataset from the literature in this area. All computations are carried through in R on an Intel(R) Core(TM) i5-6200U CPU (2.30GHz) 8 GB RAM.

A measure of the goodness of the derived designs is the Φ-*efficiency*, where Φ is a positive, concave and homogeneous function. For a design ξ, the Φ-efficiency with respect to an optimal design $\xi^* = \arg\max_\xi \Phi[M(\xi)]$ is given by

$$0 \leq \text{Eff}_\Phi[M(\xi)] = \frac{\Phi[M(\xi)]}{\Phi[M(\xi^*)]} \leq 1. \tag{15}$$

In the following examples, we will consider the D-optimality criterion, so that Φ is the determinant of $M(\xi)$, and the theoretical optimal design is computed by the `od_REX` function in the `R` package `OptimalDesign`.[7]

3.1 | Example 1

We compare algorithms presented in Sections 2.1 and 2.2 on simulated data from the following model:

$$Y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \varepsilon \tag{16}$$

where $\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2) = (0.5, -3, 4)$, x_1 and x_2 are independent and generated from a *Uniform*$[-1, 1]$ and $\varepsilon \sim \mathcal{N}(0, 0.1^2)$. We simulated $N = 10^3$ values and the goal is to select $n = 12$ design points from the initial dataset. Here, the model has no bias and we do not use a big dataset in order to compare all the presented algorithms, indeed the computational time

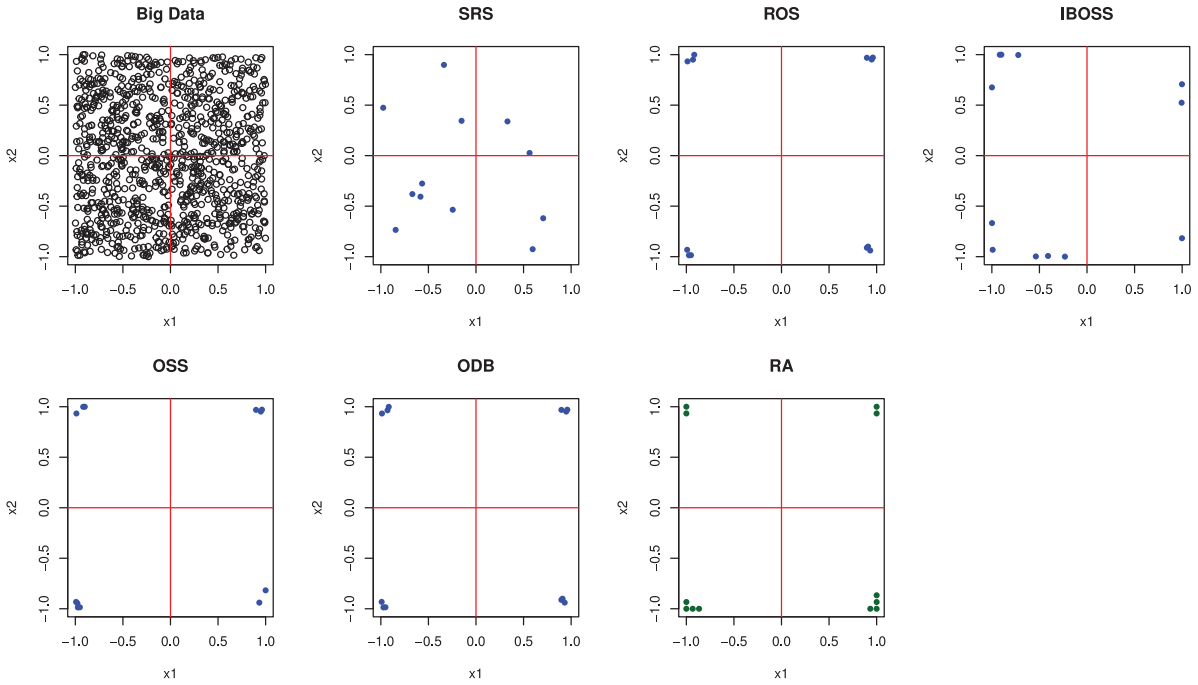

FIGURE 2 Example 1: Optimal design points selected according to each algorithm

TABLE 1 Efficiencies and computational times of each algorithm in Example 1

Algorithm	Efficiency	Timing (s)
SRS	0.4383372	–
ROS	0.9576251	0.19
IBOSS	0.8042624	0.03
OSS	0.9457539	0.08
ODB	0.9584127	8.12
RA	0.9784970	0.86

for Algorithm 5 is prohibitive. For Algorithms 1 and 4, we choose to use a grid of 31 equally space points in the interval $[-1, 1]$ for both x_1 and x_2.

In Figure 2, blue points correspond to data points in the original dataset, while green points are from the grid (and observations will be taken at them). The efficiencies of the selected designs and the computational times of each algorithm are reported in Table 1. With respect to a SRS (Figure 2), all D-optimal design tends to allocate points at the boundary of the sample space and at the corners with the exception of IBOSS. The most efficient algorithm is the last one and the longest to run is ODB, while the SRS design is obtained in virtually no time.

3.2 | Example 2

This second example considers again Model (16) and a much larger simulated dataset of $N = 10^5$ values from which to select $n = 10^3$ design points. Algorithm 5 is excluded from this comparison because of its computational time cost. For Algorithms 1 and 4, we choose to use a grid of 151 equally space points in the interval $[-1, 1]$ for both x_1 and x_2. The data points selected by each algorithm are reported in Figure 3.

The efficiencies of the selected designs and the computational times of each algorithm are reported in Table 2.

It is worth remarking the importance of the selection of the grid required in the ROS and the ODB algorithms for the computation of the efficiency. This grid is used to compute the theoretical optimal design for the computation of the efficiency and thus it is required to be fine enough, otherwise the algorithms can select data points 'more extreme' than those in the grid. A such event can provide a not reliable efficiency evaluations for the involved algorithms. The

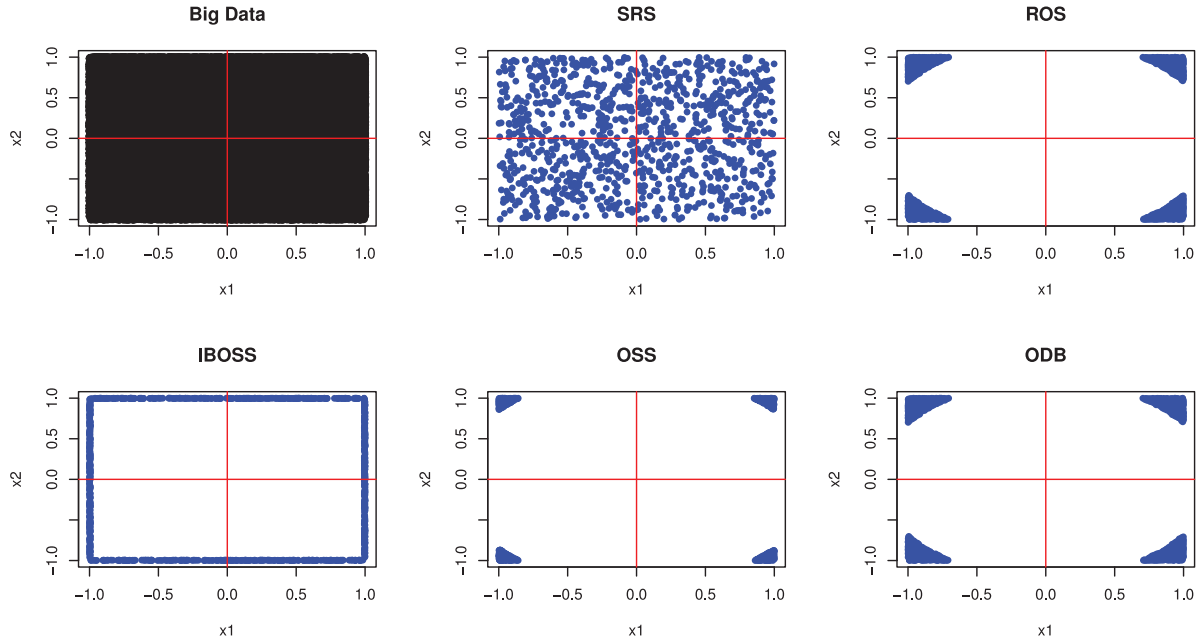

FIGURE 3 Example 2: Optimal design points selected according to each algorithm

TABLE 2 Efficiencies and computational times of each algorithm in Example 2

Algorithm	Efficiency	Timing (s)
SRS	0.4532854	–
ROS	0.8266456	776.54
IBOSS	0.7092132	0.05
OSS	0.8802289	698.52
ODB	0.8266642	611.36

RA algorithm does not conclude on the selected grid. OSS is the most efficient algorithm while computational times for ROS, OSS and ODB are comparable and IBOSS is very fast almost by definition in virtue of the easy geometry of the sample space.

3.3 | Example 3

In this third example, the bias term $\psi(x_1, x_2) = 2.22\, x_1^2$ is added to Model (16) and a dataset of $N = 10^3$ values is simulated from

$$Y(x_1, x_2) = 0.5 - 3x_1 + 4x_2 + \psi(x_1, x_2) + \varepsilon \tag{17}$$

The goal is to select $n = 12$ design points out of it. As in previous example, we do not use a big dataset in order to compare all the presented algorithms, otherwise the computational time for Algorithm 5 (RA) would have been too large. For Algorithms 1 (ROS) and 4 (ODB), we choose to use a grid of 41 equally space points in the interval $[-1, 1]$ for both x_1 and x_2.

In Table 3, the efficiencies of the selected designs and the computational times of each algorithm are reported. The algorithm with the best efficiency is RA, as expected since there is a bias in the model, but we need to consider that this algorithm is not based on the observed value. Figure 4 suggests the same comments made in the previous examples. In the two cases involving the search of a small 12 point design ODB is the slowest while for the search of a larger design in Example 2, ROS is the slowest because it requires to compute many distances.

TABLE 3 Efficiencies and computational times of each algorithm in Example 3

Algorithm	Efficiency	Timing (s)
SRS	0.4870204	–
ROS	0.9330215	0.39
IBOSS	0.7984535	0.16
OSS	0.9389230	0.09
ODB	0.9514904	8.10
RA	1.0000000	2.60

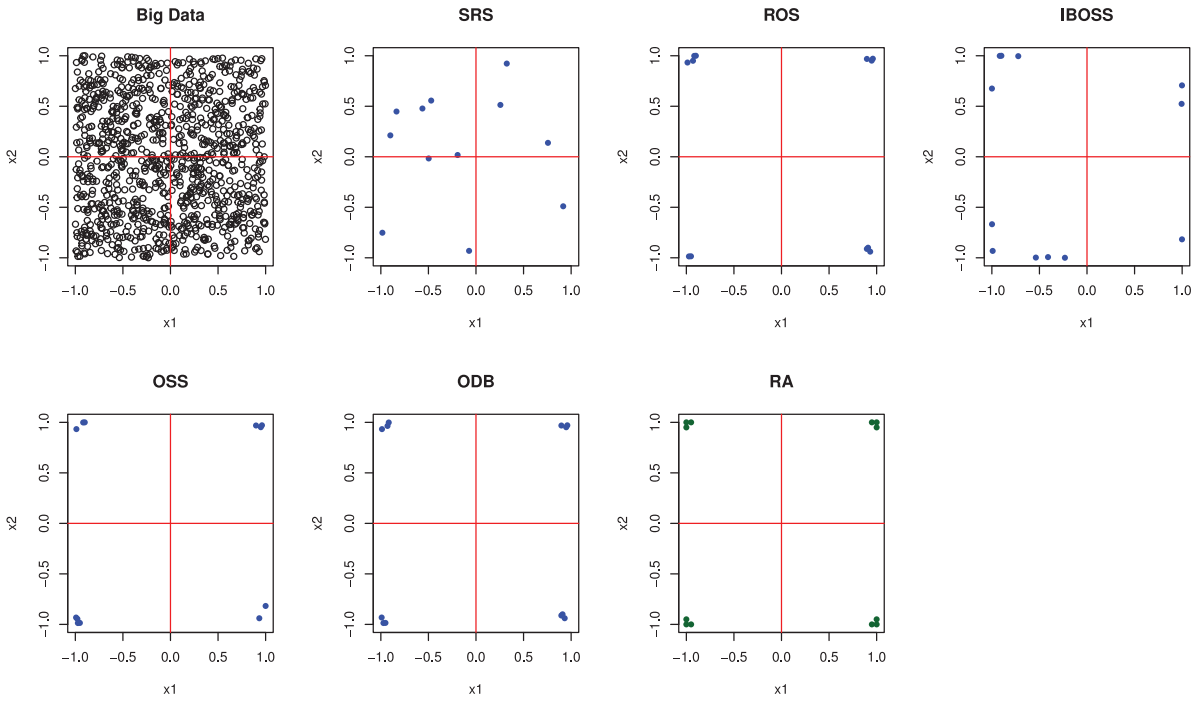

FIGURE 4 Example 3: Optimal design points selected according to each algorithm when a bias term is present

TABLE 4 Grid generated by the data points

Covariate	Grid
creditScore	−4, −3, −2, −1, 0, 1, 2, 3, 4
houseAge	−2, −1, 0, 1, 2, 3, 4
yearsEmploy	−2, −1, 0, 1, 2, 3, 4
ccDebt	−2, −1, 0, 1, 2

3.4 | Example 4: Mortgage dataset

Here, Algorithm 1 is applied on the simulated mortgage defaults (years 2000) dataset considered as test case also in other studies (see, e.g. Drovandi et al.[2]). The dataset has 1000, 000 data points, a binary response for the mortgage defaults $Y_i \sim Binary(\pi_i)$ and four covariates: Credit Rating Score (*creditScore*), the age of the house in years (*houseAge*), the number of years the mortgage holder has been employed at current job (*yearsEmploy*) and the amount of credit card debt (*ccDebt*). The scaled values of the data points are clustered around the grid in Table 4, so we take this as the grid used in Algorithm 1. The response is skewed: $Y_i = 1$ in 1031 units and $Y_i = 0$ for 998,969 units and following Drovandi et al.,[2] we assume $Y_i \sim Binary(\pi_i)$ and a logistic model $logit(\pi_i) = \theta_0 + \theta_1 x_{1i} + \theta_2 x_{2i} + \theta_3 x_{3i} + \theta_4 x_{4i}$

The maximum likelihood estimates of the parameters of the logit models are obtained starting with $n_t = 5000$ points in step 2, and with a final sample of size $n_d = 6,200$. The estimates are consistent with those presented by other authors

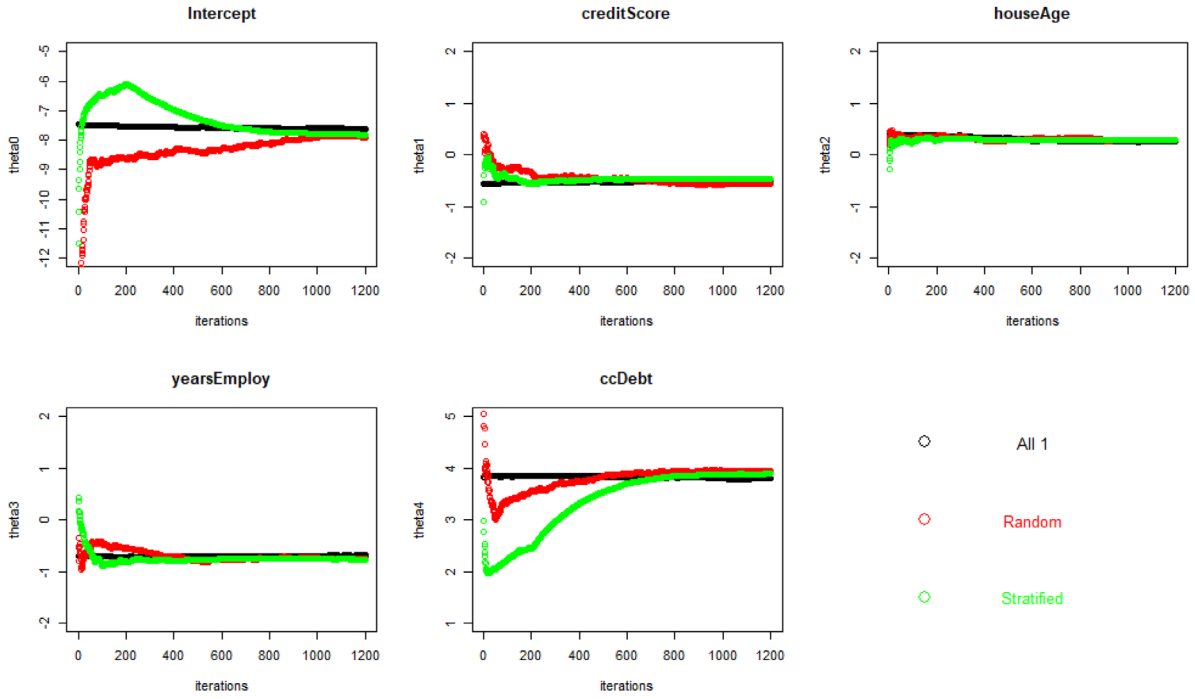

FIGURE 5 Effect of the initial sample on the output of Algorithm 1

in Ref.[2] Further to their work in Figure 5, we investigate the effect of the choice of the initial sample on the parameter estimates: **black** refers to an initial sample including all units for which $Y = 1$ (a dope training set), **red** to a randomly selected initial sample, **green** to a stratified sample: we considered the distribution of *ccDebt* for the sub-population for which $Y = 1$ and sampled one data points for each quantile, since preliminary analysis indicates that *ccDebt* affects most the response. All the estimates converge to the same value, but the black ones are faster, as expected.

4 | CONCLUSIONS AND FINAL REMARKS

In this paper, we proposed a unified general framework, which covers diverse algorithms from recent literature to approach the problem of model-oriented optimal sub-sample selection in large datasets. We reviewed them and we show that the most used algorithms address some special cases: models without bias, models with no confounder terms and models with no bias terms. We described the main advantages and drawbacks of the different algorithms.

We also presented some preliminary results on a unified theory to take into account selection bias, model bias and bias due to confounders in the choice of a sub-sample for an efficient estimation, in the least square sense, of parameters expressing the effect of interest.

ACKNOWLEDGEMENT

The support and collaboration of Swiss Re Corporate Solutions, PhD sponsor of Elena Pesce, are gratefully acknowledged.

Open Access Funding provided by Universita degli Studi di Genova within the CRUI-CARE Agreement.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in revolution analytics at http://packages.revolutionanalytics.com/datasets/.

ORCID

Eva Riccomagno https://orcid.org/0000-0001-7280-2772

REFERENCES

1. Deldossi L, Tommasi C. Big Data and model-based survey sampling. *arXiv preprint arXiv:2002.04255*. 2020.
2. Drovandi CC, Holmes C, McGree JM, Mengersen K, Richardson S, Ryan EG. Principles of experimental design for Big Data analysis. *Stat Sci*. 2017;32(3):385-404. https://doi.org/10.1214/16-STS604
3. Dunson DB. Statistics in the Big Data era: failures of the machine. *Stat Probab Lett*. 2018;136:4-9. https://doi.org/10.1016/j.spl.2018.02.028
4. Faraway JJ, Augustin NH. When small data beats Big Data. *Stat Probab Lett*. 2018;136:142-145. https://doi.org/10.1016/j.spl.2018.02.031
5. Flassig RJ, Schenkendorf R. Model-based design of experiments: where to go? 9th Vienna Conference on Mathematical Modelling, MATHMOD 2018 Extended Abstract Volume, Vienna, Austria. 2018:21-23.
6. Harford T. Big Data: are we making a big mistake? *Significance*. December 2014, 14-19, reprint from The Financial Times; 2014.
7. Harman R, Filova L. Optimaldesign: a toolbox for computing efficient designs of experiments. R package version 1.0.1. 2019. 20 June 2022. https://CRAN.R-project.org/package=OptimalDesign
8. Hedayat AS, Sloane NJA, Stufken J. *Orthogonal Arrays: Theory and Applications*. Springer; 1999. https://doi.org/10.1007/978-1-4612-1478-6
9. Meng XL. Statistical paradises and paradoxes in Big Data (I): law of large populations, Big Data paradox, and the 2016 US presidential election. *Ann Appl Stat*. 2018;12(2):685-726. https://doi.org/10.1214/18-AOAS1161SF
10. Montepiedra G, Fedorov VV. Minimum bias design with constraints. *J Stat Plan Inference*. 1977;63:97-111. https://doi.org/10.1016/S0378-3758(96)00199-1
11. Pesce E, Riccomagno E, Wynn HP. Experimental design issues in big data. The question of bias, In: Greselin F, Deldossi L, Bagnato L, Vichi M (eds) *Statistical Learning of Complex Data*. CLADAG 2017. Studies in Classification, Data Analysis, and Organization. Springer: Cham, 2017; pp. 193-201.
12. Pukelsheim F, Rieder S. Efficient rounding of approximate designs. *Biometrika*. 1992;79(4):763-770. https://doi.org/10.1093/biomet/79.4.763
13. Scott SL, Blocker AW, Bonassi FV, Chipman HA, George EI, McCulloch RE. Bayes and Big Data: the consensus Monte Carlo algorithm. *Int J Manag Sci Eng Manag*. 2016;11(2):78-88. https://doi.org/10.1080/17509653.2016.1142191
14. Sharpes LD. The role of statistics in the era of Big Data: electronic health records for healthcare research. *Stat Probab Lett*. 2018;136:105-110. https://doi.org/10.1016/j.spl.2018.02.044
15. Sangalli LM, ed. The role of Statistics in the era of Big Data. *Stat Probab Lett*. 136 (Special issue). https://doi.org/10.1016/j.spl.2018.04.009
16. Tibshirani R, Wainwright M, Hastie T. *Statistical Learning with Sparsity: the Lasso and Generalizations*. Chapman and Hall/CRC; 2015. https://doi.org/10.1201/b18401
17. Wang H, Yang M, Stufken J. Information-based optimal subdata selection for Big Data linear regression. *J Am Stat Assoc*. 2019;114(525):393-405. https://doi.org/10.1080/01621459.2017.1408468
18. Wang L, Elmstedt J, Wong WK, Xu H. Orthogonal subsampling for Big Data linear regression. *Ann Appl Stat*. 2021;15(3):1273-1290. https://doi.org/10.1214/21-AOAS1462
19. Wang C, Chen MH, Schifano E, Wu J, Yan J. Statistical methods and computing for Big Data. *Stat Interface*. 2016;9(4):399. https://doi.org/10.4310/SII.2016.v9.n4.a1
20. Wang H, Zhu R, Ma P. Optimal subsampling for large sample logistic regression. *J Am Statist Assoc*. 2018;113(522):829-844. https://doi.org/10.1080/01621459.2017.1292914
21. Wiens DP. I-robust and D-robust designs on a finite design space. *Statist Comput*. 2018;28(2):241-258. https://doi.org/10.1007/s11222-017-9728-8
22. Yao Y, Wang H. Optimal subsampling for softmax regression. *Stat Papers*. 2019;60:585-599. https://doi.org/10.1007/s00362-018-01068-6

AUTHOR BIOGRAPHIES

Elena Pesce is currently a Business Analyst in the Swiss Re Institute in Zurich. She received her PhD degree in Mathematical Methods for Data Analysis from the University of Genoa, Italy. Her research interest include optimal design of experiments and supply chain risk analysis.

Francesco Porro is currently an Assistant Professor at the University of Genoa, after a brief experience at University of Sassari. He got a PhD in Methodological and Applied Statistics from University of Milan-Bicocca. His main research topics are: economic inequality, design of the experiments, survival analysis and compositional data.

Eva Riccomagno is a Full Professor of Statistics at the University of Genova. She received her PhD degree in Statistics from the University of Warwick. She coauthored over 80 peer reviewed research contributions. Her main research interests are in Design of Experiments, Algebraic statistics, application of Statistics and Mathematics in a number of fields.

How to cite this article: Pesce E, Porro F, Riccomagno E. Large datasets, bias and model-oriented optimal design of experiments. *Qual Reliab Eng Int*. 2023;39:532–545. https://doi.org/10.1002/qre.3165

APPENDIX

We provide a proof of the General Equivalence Theorem (GET) in the presence of confounders and/or bias terms. It uses standard technique and we have not found it in the literature. The GET establishes the equivalence of the optimal design obtained under different convex criterion. In the notation of Model (1) consider the information matrix

$$M(\boldsymbol{\xi}) = \int_{\mathcal{X}\times\mathcal{Z}} \begin{pmatrix} \boldsymbol{f} \\ \boldsymbol{h} \\ \boldsymbol{g} \end{pmatrix} (\boldsymbol{f}^\top, \boldsymbol{h}^\top, \boldsymbol{g}^\top)\, d\,\xi_{\boldsymbol{x},\boldsymbol{z}} \tag{18}$$

for a generic convex design measure ξ. Consider also the variance function $d((\boldsymbol{x},\boldsymbol{z}),\boldsymbol{\xi}) = \boldsymbol{f}'(\boldsymbol{x})M(\boldsymbol{\xi})^{-1}\boldsymbol{f}(\boldsymbol{x})$. As ξ varies in the convex set of probability measures, we assume that the information matrices form a closed and bounded subset of the semi-definite positive matrices.

Theorem 1. *For a design measure $\boldsymbol{\xi}^*$, the following statements are equivalent*

(i) $\boldsymbol{\xi}^*$ *maximizes* $\det(M(\boldsymbol{\xi}))$
(ii) $\boldsymbol{\xi}^*$ *achieves* $\min_{\boldsymbol{\xi}} \max_{(x,z)\in\mathcal{X}\times\mathcal{Z}} d((\boldsymbol{x},\boldsymbol{z}),\boldsymbol{\xi})$
(iii) $\max_{(x,z)\in\mathcal{X}\times\mathcal{Z}} d((\boldsymbol{x},\boldsymbol{z}),\boldsymbol{\xi}^*) = p+m+q-2.$

Proof. There are two further equivalent conditions, which allows a circular proof. One is a local D-optimality condition

$$(iv)\ \frac{\partial}{\partial(\boldsymbol{x},\boldsymbol{z})} \log\det\left(M\left((1-\alpha)\boldsymbol{\xi}^* + \alpha\boldsymbol{\xi}'\right)\right)\,|_{\alpha=0} \le \mathbf{0} \tag{19}$$

where $\boldsymbol{\xi}^*$ and $\boldsymbol{\xi}'$ are design measures. The fifth equivalent condition is

$$(v)\ d((\boldsymbol{x},\boldsymbol{z}),\boldsymbol{\xi}) \le p+m+q-2 \quad \text{for all } (\boldsymbol{x},\boldsymbol{z}) \in \mathcal{X}\times\mathcal{Z} \tag{20}$$

That *(i)* implies *(iv)* is straightforward. To show that *(iv)* implies *(v)* we use the matrix identity $\frac{\partial}{\partial\alpha}\log\det(A) = \mathrm{tr}(A^{-1}\frac{\partial A}{\partial\alpha})$. Thus

$$\begin{aligned}
&\frac{\partial}{\partial(\boldsymbol{x},\boldsymbol{z})} \log\det\left(M\left((1-\alpha)\boldsymbol{\xi}^* + \alpha\boldsymbol{\xi}'\right)\right)\,|_{\alpha=0} \\
&\quad= \mathrm{tr}\left(M^{-1}\left((1-\alpha)\boldsymbol{\xi}^* + \alpha\boldsymbol{\xi}'\right)\cdot\frac{\partial}{\partial\alpha}M\left((1-\alpha)\boldsymbol{\xi}^* + \alpha\boldsymbol{\xi}'\right)\right)\,|_{\alpha=0} \\
&\quad= \mathrm{tr}\left(M^{-1}\left((1-\alpha)\boldsymbol{\xi}^* + \alpha\boldsymbol{\xi}'\right)\cdot\frac{\partial}{\partial\alpha}\left((1-\alpha)M(\boldsymbol{\xi}^*) + \alpha M(\boldsymbol{\xi}')\right)\right)\,|_{\alpha=0} \\
&\quad= \mathrm{tr}\left(M^{-1}\left((1-\alpha)\boldsymbol{\xi}^* + \alpha\boldsymbol{\xi}'\right)\cdot\left(-M(\boldsymbol{\xi}^*) + M(\boldsymbol{\xi}')\right)\right)\,|_{\alpha=0} \\
&\quad= \mathrm{tr}\left(M^{-1}(\boldsymbol{\xi}^*)\cdot\left(-M(\boldsymbol{\xi}^*) + M(\boldsymbol{\xi}')\right)\right) = \mathrm{tr}\left(-I_{p+m+q-2} + M^{-1}(\boldsymbol{\xi}^*)M(\boldsymbol{\xi}')\right) \\
&\quad= -(p+m+q-2) + \mathrm{tr}\left(\int_{\mathcal{X}\times\mathcal{Z}} d((\boldsymbol{x},\boldsymbol{z}),\boldsymbol{\xi}^*)\,\boldsymbol{\xi}'(dx,dz)\right)
\end{aligned} \tag{21}$$

so that the statement in *(iv)* is equivalent to $\int_{\mathcal{X}\times\mathcal{Z}} d((\boldsymbol{x},\boldsymbol{z}),\boldsymbol{\xi}^*)\,\boldsymbol{\xi}'(dx,dz) \le p+m+q-2$ for all $\boldsymbol{\xi}'$. This holds in particular when $\boldsymbol{\xi}'$ places mass one at a specific point $(\boldsymbol{x},\boldsymbol{z})$. But this is $d((\boldsymbol{x},\boldsymbol{z}),\boldsymbol{\xi}^*) \le p+m+q-2$ for all $(\boldsymbol{x},\boldsymbol{z})$, so *(v)* is verified.

To prove that *(iii)* is equivalent to *(iv)*, we can show that

$$\max_{(x,z)\in\mathcal{X}\times\mathcal{Z}} d((\boldsymbol{x},\boldsymbol{z}),\boldsymbol{\xi}^*) \geq p+m+q-2 \text{ for all } (\boldsymbol{x},\boldsymbol{z}) \tag{22}$$

But this follows from the fact that a maximum is always greater than or equal to an average, so

$$\begin{aligned}\max_{(x,z)\in\mathcal{X}\times\mathcal{Z}} d((\boldsymbol{x},\boldsymbol{z}),\boldsymbol{\xi}^*) &\geq \int_{\mathcal{X}\times\mathcal{Z}} d((\boldsymbol{x},\boldsymbol{z}),\boldsymbol{\xi}^*)\,\boldsymbol{\xi}^*(dx,dz)\\ &= \operatorname{tr}\left(M^{-1}(\boldsymbol{\xi}^*)M(\boldsymbol{\xi}^*)\right) = \operatorname{tr}(I_{p+m+q-2}) = p+m+q-2\end{aligned} \tag{23}$$

As we assumed that $M(\boldsymbol{\xi})$ is a closed and bounded subset of the semi-definite positive matrices, $\boldsymbol{\xi}^*$ achieves the bound.

Lastly we prove that *(iii)* implies *(i)*. We use the identity $\operatorname{tr}(A) \geq n \cdot \det(A)^{\frac{1}{n}}$ for an $n \times n$ matrix A. Thus for $k = p + m + q - 2$ we have

$$\begin{aligned}k = \max_{(x,z)\in\mathcal{X}\times\mathcal{Z}} d((\boldsymbol{x},\boldsymbol{z}),\boldsymbol{\xi}^*) &\geq \int_{\mathcal{X}\times\mathcal{Z}} d((\boldsymbol{x},\boldsymbol{z}),\boldsymbol{\xi}^*)\,\boldsymbol{\xi}'(dx,dz)\\ &= \operatorname{tr}\left(M^{-1}(\boldsymbol{\xi}^*)M(\boldsymbol{\xi}')\right) \geq k\cdot\det\left(M^{-1}(\boldsymbol{\xi}^*)M(\boldsymbol{\xi}')\right)^{\frac{1}{k}}\\ &= k\cdot\left(\det\left(M^{-1}(\boldsymbol{\xi}^*)\right)\det\left(M(\boldsymbol{\xi}')\right)\right)^{\frac{1}{k}} = k\cdot\left(\frac{\det(M(\boldsymbol{\xi}'))}{\det(M(\boldsymbol{\xi}^*))}\right)^{\frac{1}{k}}\end{aligned} \tag{24}$$

From this $\det(M(\boldsymbol{\xi}^*)) \geq \det(M(\boldsymbol{\xi}'))$, which is *(i)*; so we have *(iii)* holds if and only if *(i)* holds. □