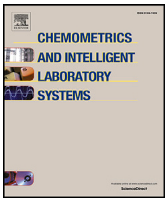
Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

journal homepage: www.elsevier.com/locate/chemometrics



Using hierarchical information-theoretic criteria to optimize subsampling of extensive datasets

Belmiro P.M. Duarte [a,b,*], Anthony C. Atkinson [c], Nuno M.C. Oliveira [b]

[a] Polytechnic Institute of Coimbra, ISEC, Department of Chemical & Biological Engineering, Rua Pedro Nunes, 3030–199 Coimbra, Portugal
[b] University of Coimbra, CIEPQPF, Department of Chemical Engineering, Rua Sílvio Lima — Pólo II, 3030–790 Coimbra, Portugal
[c] Department of Statistics, London School of Economics, London WC2A 2AE, United Kingdom

ARTICLE INFO

Keywords:
Subsampling
Information-theoretic criteria
Large datasets
Hierarchical data partitioning
Semidefinite programming

ABSTRACT

This paper addresses the challenge of subsampling large datasets, aiming to generate a smaller dataset that retains a significant portion of the original information. To achieve this objective, we present a subsampling algorithm that integrates hierarchical data partitioning with a specialized tool tailored to identify the most informative observations within a dataset for a specified underlying linear model, not necessarily first-order, relating responses and inputs. The hierarchical data partitioning procedure systematically and incrementally aggregates information from smaller-sized samples into new samples. Simultaneously, our selection tool employs Semidefinite Programming for numerical optimization to maximize the information content of the chosen observations. We validate the effectiveness of our algorithm through extensive testing, using both benchmark and real-world datasets. The real-world dataset is related to the physicochemical characterization of white variants of Portuguese *Vinho Verde*. Our results are highly promising, demonstrating the algorithm's capability to efficiently identify and select the most informative observations while keeping computational requirements at a manageable level.

1. Motivation

Subsampling has emerged as a highly effective strategy for addressing challenges associated with big (and large) datasets. Over time, numerous subsampling methodologies have emerged to tackle this issue. Diverse techniques and models, such as leverage sampling in linear regression, have been repeatedly introduced in the literature, as evidenced by reviews by Stewart [1] and Yao and Wang [2]. In practical terms, subsampling is a statistical technique employed to extract a smaller, representative sample from a larger dataset while minimizing information loss. Its primary objective is to retain the majority of the essential information present in the original dataset by carefully selecting observations or records to form a smaller sample, often referred to as a subsample or data subset. This approach proves particularly valuable when dealing with extensive datasets, as it facilitates more manageable data analysis techniques. Frequently, data quality is degraded by the deficiency of information value due to the non-controlled experimental conditions under which observations are obtained. Furthermore, when the number of observations is substantial, even unimportant variables can become statistically significant.

To address the challenge of subsampling from large datasets, a variety of methods have been developed, as exemplified by the work of Mahoney [3], Drineas et al. [4], Ma et al. [5] and Wang [6], among others. These methods share a common underlying principle: the utilization of sampling criteria that ensure a high degree of similarity between the descriptive statistics of the sample and those of the entire population. Several strategies have been employed to achieve this goal: (i) leverage scores: Ma et al. [5] introduced the concept of leverage scores as a means to guide subsampling; (ii) optimal Poisson subsampling: Yu et al. [7] put forward an optimal Poisson subsampling approach, offering another avenue to address this challenge; and (iii) asymptotic distribution in linear regression: Ma et al. [8] find the asymptotic distribution of the sampling estimator within the context of linear regression. In addition to these developments, there exists a rich historical context concerning the use of information-theoretical criteria for subsampling. Pioneering work by Wynn [9,10], Fedorov [11] and Pronzato [12], among others, has paved the way for further advancements in the field. Given the growing importance of this topic, further contributions have emerged. Some authors have proposed methods that combine information criteria with linear regression models, as exemplified by the work of Deldossi and Tommasi [13], Reuter and Schwabe [14] and Wang et al. [15]. Additionally, fully Bayesian

* Corresponding author at: Polytechnic Institute of Coimbra, ISEC, Department of Chemical & Biological Engineering, Rua Pedro Nunes, 3030–199 Coimbra, Portugal.
E-mail addresses: bduarte@isec.pt (B.P.M. Duarte), A.C.Atkinson@lse.ac.uk (A.C. Atkinson), nuno@eq.uc.pt (N.M.C. Oliveira).

https://doi.org/10.1016/j.chemolab.2024.105067
Received 18 September 2023; Received in revised form 11 January 2024; Accepted 13 January 2024
Available online 15 January 2024

experimental designs have been developed, aiming, as a comprehensive approach to addressing the subsampling problem, to maximize the utility function representing the Shannon information gain [16].

The concept of utilizing information-theoretical criteria to identify a subset of observations that maximizes the information content is appealing, as it aligns with the core principle of subsampling: reducing the size of the dataset while retaining substantial information. Moreover, this approach offers a solution based on the optimal design of experiments, a problem that can be effectively tackled using various systematic algorithmic tools. Several algorithms have proven useful in this context, including: (i) Semidefinite Programming: as illustrated by Duarte et al. [17] provides a systematic approach to address this challenge; (ii) Second Order Cone Programming: Sagnol [18] demonstrated the efficacy of Second Order Cone Programming in optimizing experimental designs; and (iii) Mixed Integer Linear Programming: Vo-Thanh et al. [19] have successfully employed Mixed Integer Linear Programming techniques for this purpose. Furthermore, the ever-increasing computational power at our disposal enables the utilization of Bayesian experimental designs through simulation. Researchers can harness this power to explore such designs, as exemplified by the work of Huan and Marzouk [20], Overstall and Woods [21].

While the concept is undeniably appealing, the challenges associated with handling large datasets persist, despite the advances in optimization algorithms and computational capabilities. Even when employing algorithms of polynomial complexity (referred to as P-type algorithms) to tackle optimization problems, the large volume of data can still present limitations, particularly when working in batch mode. To mitigate these challenges, a common strategy involves the use of data partitioning, as exemplified by Mahmud et al. [22]. Data partitioning serves as a pivotal technique, allowing for the organization of vast datasets into manageable subsets. This not only enhances the tractability of the problem, as demonstrated by Singh et al. [23], but also paves the way for the integration of information across these subsets using hierarchical approaches and iterative procedures. In practical terms, information gleaned from a specific level of analysis is passed on to the subsequent level, as articulated by Stergiou and Poppe [24]. Through this iterative process, a representative data subset that encapsulates the essence of the original dataset is progressively obtained. Finally, Wang et al. [25] introduced the Information-Based Optimal Subdata Selection (IBOSS) algorithm, which exhibits connections with our proposed algorithm. These are discussed in Section 6.

Our paper is centered on the subsampling of extensive datasets which we assume are characterized by linear relationships between response variables and covariates. To address this challenge, we employ a comprehensive approach that combines hierarchical strategies with data partitioning based on optimal design of experiments. This approach enables us to identify the most informative observations within smaller subsets of the original dataset. The selection of these informative observations is guided by optimality information criteria, which allow us to make judicious choices. To optimize the information content within each subset, we utilize optimality criteria and Semidefinite Programming (SDP) formulations to find the most informative observations. In our investigation, we focus on the most commonly used optimality criteria, namely D-, A-, and E-criteria, owing to their simplicity and the ease with which they can be translated into tractable SDP formulations. The results depend both on the optimality criterion and the assumed linear model. The output is a subset of observations and the optimal experimental design for this subset. We discuss applications of this design in Section 6.

1.1. Novelty statement and organization

This paper presents several novel contributions: (i) automated subsampling approach: We introduce an innovative, computationally automated method for subsampling large datasets; (ii) a hierarchical-based strategy: Our approach incorporates a hierarchical framework that combines data partitioning with the optimized selection of the most informative observations through SDP; (iii) application to a benchmark test: We apply our algorithm to a benchmark test, demonstrating its practical utility and effectiveness; and (iv) impact analysis: We conduct a detailed analysis to evaluate the influence of both partition size and the prior sorting of observations based on response variables on the performance of the algorithm. This analysis sheds light on key factors affecting the effectiveness of the proposed method.

The paper is organized as follows. Section 2 introduces the background and the notation used to formulate the problem of the choice of the most informative observations, as well as the fundamentals of Semidefinite Programming used to compute the support points and the hierarchical partitioning of data. Section 3 introduces the algorithm proposed to automate the subsampling. Comparisons for different setups using a benchmark dataset are presented in Section 4. Section 5 demonstrates the application of the algorithm to a real dataset related to data characterizing physicochemically white variants of Portuguese *Vinho Verde*. Section 6 reviews the formulation and offers a summary of the results obtained with the proposed tool.

2. Notation and background

In our notation bold face lowercase letters represent vectors, bold face capital letters stand for continuous domains, blackboard bold capital letters are used to denote discrete domains and capital letters are adopted for matrices. The transpose operation of a matrix or vector is represented by "$^\intercal$". The trace of a matrix is represented by $\mathrm{tr}(\bullet)$, and $\mathrm{hcat}(A, B)$ represents the horizontal concatenation of two matrices with equal number of rows into one, and $\mathrm{vcat}(A, B)$ stands for the vertical concatenation of matrices with equal number of columns.

In Section 2.1, we present the key measures employed for evaluating experimental designs. In Section 2.2, we introduce the foundational concepts of SDP. Following that, in Section 2.3, we provide a comprehensive overview of the fundamental aspects of hierarchical data partitioning.

With a slight abuse of notation, let $D = \mathrm{hcat}(Y, X) \in \mathbb{R}^{n_0\times(n_\mathrm{r}+n_\mathrm{c})}$ be a large dataset comprising n_0 observations of $n_\mathrm{r}+n_\mathrm{c}$ variables. The matrix $Y \in \mathbb{R}^{n_0\times n_\mathrm{r}}$ contains as lines the vectors $\mathbf{y}_i \in \mathbb{R}^{1\times n_\mathrm{r}}$ of the observed response variables, $i = 1,\dots,n_0$. Additionally, matrix $X \in \mathbb{R}^{n_0\times n_\mathrm{c}}$, contains the input variables, with ith row $\mathbf{x}_i \in \mathbb{R}^{1\times n_\mathrm{c}}$ representing the measurements of the n_c covariates and constant terms (if any) for the ith observation, $i = 1,\dots,n_0$. A subset of size $n_l < n_0,\ l > 0$ of the original dataset is denoted by $D^{\mathrm{strip}} = \mathrm{hcat}(Y^{\mathrm{strip}}, X^{\mathrm{strip}}) \in \mathbb{R}^{n_l\times(n_\mathrm{r}+n_\mathrm{c})}$, where Y^{strip} and X^{strip} are defined correspondingly. Notice that the number of observations in our algorithm is indexed by an iterative procedure where the hierarchical level, indexed to counter l, is increased; here n_0 holds for the observations in the original data matrix which corresponds to level $l = 0$, and $n_l,\ l \in \{1,\dots,L\}$ for the number of observations of the stripped data in the remaining hierarchical levels, with L indicating the total number of hierarchical partition levels required.

To maintain generality, we focus on a single response variable and n_c regression factors. We describe the relationship between the response and covariates using a linear model:

$$y_i = \beta_0 + \boldsymbol{\beta}\ \mathbf{x}_i^\intercal + \epsilon_\mathrm{c} = \boldsymbol{\theta}\ \mathbf{h}^\intercal(\mathbf{x}_i) + \epsilon_i. \tag{1}$$

Here, $\boldsymbol{\theta} = \{\beta_i | i \in \{0,\dots,n_\mathrm{c}\}\} \in \mathbb{R}^{1\times(n_\mathrm{c}+1)}$, is the vector of regression coefficients. The vector $\mathbf{h}(\mathbf{x}_i) \in \mathbb{R}^{1\times(n_\mathrm{c}+1)}$, where $\mathbf{h}(\mathbf{x}_i) = (1, \mathbf{x}_i)$, contains the measurements for the regressors, augmented by the constant 1. For simplicity, we compactly refer to $\mathbf{h}(\mathbf{x}_i)$ as $\mathbf{h}_i$.

Given a data matrix with n_0 observations, a design ξ with $n_k < n_0$ support points is a vector of n_0 weights w_i $(i \in \{1,\dots,n_0\})$, that are greater than zero for the (n_k) observations that form the support of ξ and are equal to zero for all the other observations in the dataset. The global Fisher Information Matrix corresponding to model (1) for a design ξ on the original dataset is given by

$$\mathcal{M}(\xi|X) = -\mathbb{E}\left[\frac{\partial}{\partial\boldsymbol{\theta}}\left(\frac{\partial\mathcal{L}(\xi)}{\partial\boldsymbol{\theta}^\intercal}\right)\right] = n_0\cdot\sum_{i=1}^{n_0} w_i\ M(\mathbf{x}_i) = n_0\cdot\sum_{i=1}^{n_0} w_i\ \mathbf{h}_i^\intercal\ \mathbf{h}_i, \tag{2}$$

where $M(\mathbf{x}_i)$ denotes the elemental FIM at $\mathbf{x}_i \in X$, $\mathcal{L}$ represents the log-likelihood, and $\mathbb{E}[\bullet]$ stands for the expectation. It is worth noting that X can be replaced by X^{strip} and n_0 by n_l in subsequent hierarchical partition levels.

In our iterative procedure for subset selection we find optimum designs for subsets of size n_x. We would like to find an exact design for each subset, that is one for which some weights w_i are zero and the other weights are equal and sum to one. The support points of this design then form part of a subset for the next level of iteration. Such an exact design problem is computationally forbidding. We therefore solve the computationally easier continuous or approximate optimum design problem for which $w_i \geq 0$ and $\sum_{i=1}^{n_x} w_i = 1$. These weights are then rounded to give exact designs. The details are in Section 3.1.

2.1. Measuring design efficiency

Here, we establish the metrics for assessing the efficiency of an experimental design ξ relative to a reference design ξ^{ref}. These results are used repeatedly in the subsequent sections of the paper. We define the p−efficiency of a design ξ wrt ξ^{ref} ($\text{p} = \{\text{D}, \text{A}, \text{E}\}$) as follows:

$$\text{Eff}_D(\xi, \xi^{\text{ref}}) = \left(\frac{\det[\mathcal{M}(\xi|X)]}{\det[\mathcal{M}(\xi^{\text{ref}}|X)]} \right)^{1/(n_c+1)} \tag{3a}$$

$$\text{Eff}_A(\xi, \xi^{\text{ref}}) = \frac{\text{tr}[\mathcal{M}(\xi^{\text{ref}}|X)^{-1}]}{\text{tr}[\mathcal{M}(\xi|X)^{-1}]} \tag{3b}$$

$$\text{Eff}_E(\xi, \xi^{\text{ref}}) = \frac{\lambda_{\min}[\mathcal{M}(\xi|X)]}{\lambda_{\min}[\mathcal{M}(\xi^{\text{ref}}|X)]} \tag{3c}$$

where, $\lambda_{\min}[\bullet]$ denotes the minimum eigenvalue.

In Section 4 we take ξ^{ref} to be the optimal continuous design for the optimal subset "stripped" from the original dataset. Practically, ξ^{ref} can be sub-optimal if the design space is continuous but is globally optimal for the set of observations forming the dataset. The tabulated efficiencies of our designs will therefore be less than one.

2.2. Semidefinite programming

In this Section, we introduce the fundamentals of Semidefinite Programming. This class of (convex) mathematical programming methods is employed to solve the optimal experimental design problems, given the discrete design domain $\mathbb{X}^n$ populated with n experimental candidate points.

Let $\mathbb{S}_+^{n_\theta}$ be the space of $n_\theta \times n_\theta$ symmetric positive semidefinite matrices, and $\mathbb{S}^{n_\theta}$ the space of $n_\theta \times n_\theta$ symmetric matrices. A convex set $\mathbf{S} \subset \mathbb{R}^{n_\theta}$ is considered semidefinite representable (SDr) when for all $\boldsymbol{\zeta} \in \mathbf{S}$, the projection of $\boldsymbol{\zeta}$ onto a higher-dimensional set $\mathbf{S}^{\text{exp}}$ can be precisely characterized using Linear Matrix Inequalities (LMIs). A SDr set is always a convex set given by finitely many polynomial inequalities (of strict and non-strict nature) involving semidefinite matrices [26, §3]. In turn, a convex (or concave) function $\varphi : \mathbb{R}^{m_1} \mapsto \mathbb{R}$ is SDr if and only if the epigraph of φ, $\{(t, \boldsymbol{\zeta}) : \varphi(\boldsymbol{\zeta}) \leq t\}$, or the hypograph, $\{(t, \boldsymbol{\zeta}) : \varphi(\boldsymbol{\zeta}) \geq t\}$, respectively, are SDr and can be cast by LMIs [26,27]. The values, $\boldsymbol{\zeta}$, that optimize specific SDr functions are then formulated as *semidefinite programs* of the form [27, §4.6.2]:

$$\max_{\boldsymbol{\zeta}} \quad \mathbf{d}^{\intercal}\, \boldsymbol{\zeta} \tag{4a}$$

$$\text{s.t.} \quad \sum_{i=1}^{m_1} \zeta_i \; M_{i,j} + M_{0,j} \preceq 0, \quad j \in \{1, \ldots, k\} \tag{4b}$$

$$M_0\, \boldsymbol{\zeta} \leq \rho \tag{4c}$$

$$M_{i,j} \in \mathbb{S}_+^k, \quad i \in \{0, \ldots, m_1\}, \quad j \in \{1, \ldots, k\}. \tag{4d}$$

In our design context, the decision variables in vector $\boldsymbol{\vartheta}$ are the weights of the design w_i, $i \in \{1, \ldots, n\}$, and other required auxiliary variables, and $\mathbf{d}$ is a vector of known constants that depends on the design problem. Semidefinite positive matrices $M_{i,j}$, $i \in \{1, \ldots, m_1\}$, $j \in \{1, \ldots, k\}$ contain elemental FIMs and other matrices produced by the reformulation of the function $\varphi(\xi)$ into LMIs. The problem of calculating an optimal design for a pre-specified set of candidate experimental points $\mathbb{X}^n = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ is solved with the formulation (4a)–(4d) complemented by the linear constraints on the weights: (i) $w_i \geq 0$; and (ii) $\sum_{i=1}^{n} w_i = 1$.

Ben-Tal and Nemirovski [26] provide a list of SDr functions useful for solving continuous optimal design problems with SDP formulations, see Boyd and Vandenberghe [27, §7.3]. Recently, Sagnol [28] showed that each criterion in the Kiefer class of optimality criteria is SDr for all rational values of $\delta \in (-\infty, 0[$ and general Semidefinite Programming formulations exist for them. Here, δ is the coefficient in the Kiefer general class of criteria $\boldsymbol{\Phi}_\delta$ [29]. Notice that, in (3), A-optimality corresponds to $\delta = -1$, E-optimality to $\delta \to -\infty$ and D-optimality to $\delta \to 0$. Practically, the problem of finding optimal approximate experimental designs for the most common convex (or concave) criteria can be formulated as a Semidefinite Programming problem falling into the general representation, see Vandenberghe and Boyd [30] and Duarte and Wong [31] among others.

2.3. Hierarchical partitioning

In this Section we present the fundamentals of dataset partitioning, which serves two primary objectives: (i) reducing the computational complexity in the search for a set of candidate observations for subsampling; and (ii) ensuring a comprehensive analysis of the calculated subsample and so hopefully of the entire dataset. In the context of our discussion, "data partitioning" refers to the division of the dataset into distinct, subsets, characterized by tables featuring an equivalent number of columns and a limited number of rows. As datasets grow in size to a point where they, along with their associated processing metadata, exceed cache capacity, partitioning becomes instrumental in enhancing the performance of critical database operations. These operations encompass tasks such as joins, aggregations, and sorts, as emphasized by Lin et al. [32]. Furthermore, the application of hierarchical partitioning enables the efficient management of diverse segments and the subsequent merging of results into a consolidated dataset, as detailed in the work by Sasaki [33].

We employ the same nomenclature as introduced in Section 2; $F_k^{0,n_x} = (Y_k^{n_x}, X_k^{n_x})$ denote the kth subset of the original dataset D, comprising a fixed number n_x of consecutive observations. Likewise, as for the original dataset, we define $F_k^{l,n_x} = \text{hcat}(Y_k^{l,n_x}, X_k^{l,n_x}) \in \mathbb{R}^{n_x \times (n_c + n_r)}$ with X_k^{l,n_x} and Y_k^{l,n_x} containing observations for the regressors and the responses respectively, within the kth subset (of size n_x) of a reduced dataset $D^{l,\text{strip}}$ available at the hierarchical level l. Let n_l denote the size of the reduced dataset $D^{l,\text{strip}}$, available at the hierarchical level l; then the total number of subsets F_k^{l,n_x} of $D^{l,\text{strip}}$ is $m_l = \lceil n_l / n_x \rceil$, where $\lceil \bullet \rceil$ represents the ceiling operator.

Each data subset F_k^{l,n_x} undergoes a selection process employing an SDP-based tool, resulting in the identification of the most informative observations. This process leads to the creation of "stripped" data subsets characterized by a reduced number of observations $n_k^l < n_x$ denoted as $F_k^{l,\text{strip}}$, $k \in \{1, \ldots, m_l\}$. In other terms, n_k^l is the number of observations within the kth dataset that optimizes F_k^{l,n_x} at the lth hierarchical level. Subsequently, l is increased by 1 and the "stripped" data subsets, $F_k^{l-1,\text{strip}}$, $k \in \{1, \ldots, m_{l-1}\}$, are combined to form an extended dataset $D^{\text{strip},l} = \text{vcat}(F_1^{l-1,\text{strip}}, \ldots, F_{m_{l-1}}^{l-1,\text{strip}})$ which can then be subjected to further partitioning if necessary. The number of observations retained in $D^{l,\text{strip}}$ is represented as $n_l = \sum_{k=1}^{m_{l-1}} n_k^{l-1}$. The hierarchical level at convergence is denoted as l^* and the size of the respective "stripped" dataset as n_{l^*}.

The number of required partition levels is unknown initially, as it hinges on the unpredictability of the number of support points needed to optimize information extraction from each data subset. To address this lack of knowledge, we employ the following rule for increasing the partition level: when $m_l = \lceil n_l / n_x \rceil > 2$, $D^{l,\text{strip}}$ is divided into slots

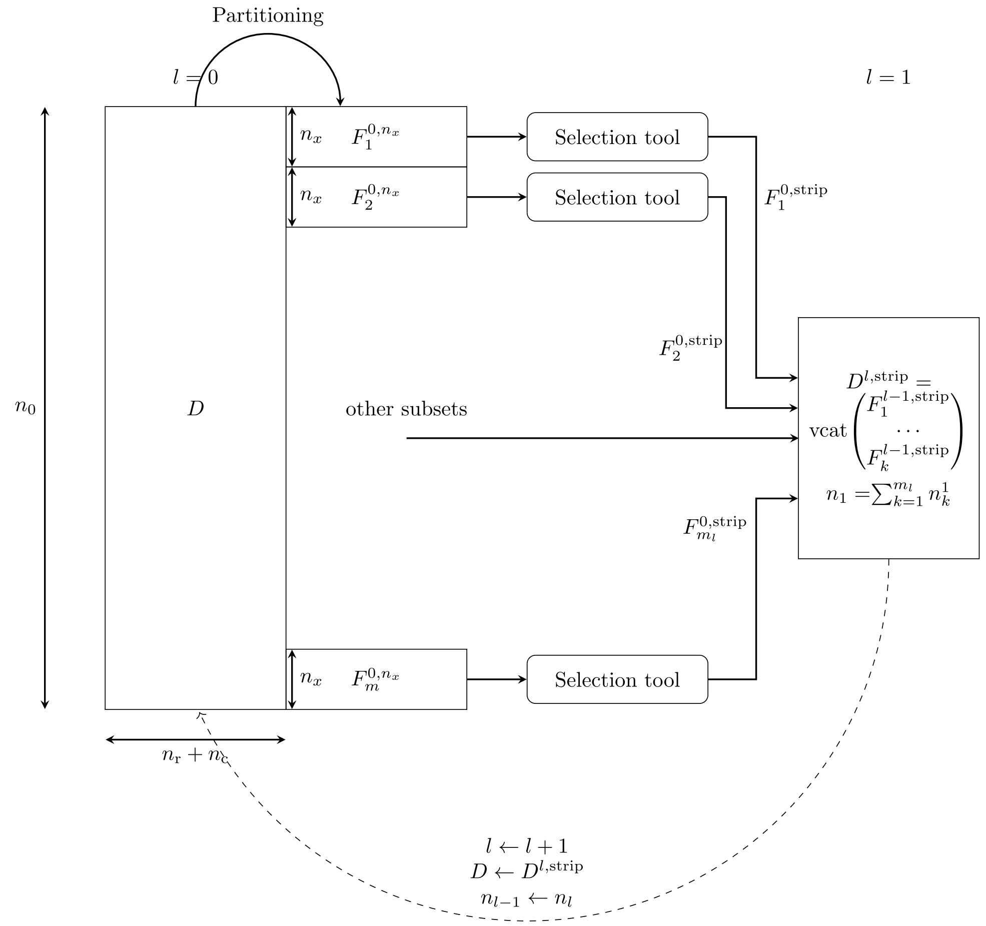


Fig. 1. Data partitioning-based algorithm. The “other subsets” each provide a F_i^{0,m_l} data matrix.

comprising n_x observations each and we increment the hierarchical level by one. Conversely, if the condition does not hold, the final hierarchical level l^* is reached. The dataset $D^{l^*,\text{strip}}$ of size n_{l^*} is not partitioned and it undergoes one final iteration through the *Selection tool* to obtain a final optimum design. In other words, we raise the partition level when the number of selected observations exceeds twice the value of n_x; otherwise, the dataset undergoes a last round of information-maximizing selection.

To address potential numerical errors, we introduce a lower threshold, denoted as “tol”, to the weights of individual observations obtained by solving the SDP problem, represented as w_i. This “tol” value is consistently set to 1×10^{-4} in all subsequent problems investigated in this study. In Fig. 1, we present an illustrative overview of our hierarchical data partitioning approach. Within this diagram, the *Selection tool* symbolizes the SDP solver module employed to determine the optimal design of experiments within the set of candidate observations X, excluding observations with weights below “tol”.

3. Algorithm for model subsampling

In this Section we introduce the framework proposed for data subsampling and performance measurement.

3.1. Finding alphabetic optimal experimental designs via semidefinite programming

This Section presents the formulations for obtaining optimal alphabetic designs using SDP, as illustrated by the *Selection tool* in Fig. 1. SDP can pose computational challenges, particularly when dealing with a substantial number of candidate observations. However, it offers the advantage of ensuring the identification of the global optimum within a grid of discrete candidate points. The Semidefinite Programming formulations for all the criteria share a common structure, as outlined in Eqs. (4a)–(4d). Detailed formulations for the D-, A-, and E-optimality criteria, which are currently considered state-of-the-art, are provided in Appendix.

In our work, we addressed the Semidefinite Programming problems utilizing the `cvx` environment in conjunction with the `Mosek` solver, renowned for its efficient Interior Point algorithm [34]. To ensure computational precision, we set both relative and absolute tolerances of the SDP solver at 1×10^{-5}. Subsequently, we filter observations, retaining only those with w_i values greater than or equal to the tolerance threshold “tol”.

In essence, we keep observations in datasets F_k^{l,n_x} that result from the SDP treatment, where $w_i \geq \text{tol}$, while discarding others. To guarantee that the sum of weights equals 1 after the sweeping operation, we normalize the distribution of w’s. The number of observations extracted from each dataset is determined by the relation

$$n_k^l = \sum_{i=1}^{n_x} 1_{w_i \geq \text{tol}}, \;\; k \in \{1, \dots, m_l\}, \;\; l \in \{1, \dots, L\}. \tag{5}$$

The SDP-based tool facilitates the generation of subsampling sets that do not necessarily coincide with exact designs. By “exact design”, we refer to a design where all the weights are rational numbers, whereas “approximate designs” encompass designs that incorporate weights falling within the non-rational range in $[0, 1]$. To generate exact designs from previous approximate designs obtained via SDP one can employ the rounding procedure introduced by Pukelsheim and Rieder [35], provided that the total number of observations equals n_l. All computations presented in this paper were executed on a 64-bit

Windows 10 operating system running on an Intel Core i7 machine clocked at 2.80 GHz.

3.2. Hierarchic partitioning tool

This Section introduces the algorithm designed for subsampling. Initially, Algorithm 1 provides a description of the *Selection tool*, which accomplishes several tasks: solving the SDP problems for each data partition (as discussed in Section 3.1), normalizing the weights, and calculating equivalent exact designs from approximate designs. It is important to note that the acronym MIO stands for "Most Informative Observations".

Algorithm 1 Selection of the most informative observations in each dataset F_k^{l,n_x}.

```
procedure SelectObservations(Input: Criterion, $k, n_x, F_k^{l,n_x}$,tol; Output:$n_k^l$, $\xi_k^l$)
    Compute single observation FIM's
    Solve the SDP problem
    Find observations with $w_i \geq$ tol; $\mathbf{s} \equiv \{i : w_i \geq \text{tol}\}$
    $w_i \leftarrow w_i / \sum_{\mathbf{s}} w_i$                                  ▷ Re-normalize the weights
    $F_k^{l,\text{strip}} \leftarrow \{F_{k,i}^{l,n_x} : i \in \mathbf{s}\}$       ▷ Build the set of MIO
    $\xi_k^l \leftarrow (F_k^{l,\text{strip}}; \mathbf{w})$
    if Exact designs then
        Set $N \leftarrow \text{card}(\mathbf{s})$
        $\xi_k^l \leftarrow (F_k^{l,\text{strip}}; \mathbf{w}),\ \mathbf{w} = \mathbf{n}/N \in [0,1] \subset \mathbb{Z}$       ▷ Rounding
    end if
end procedure
```

Algorithm 2 outlines the hierarchical partition tool, which is schematically depicted in Fig. 1. This procedure comprises an external loop in which l is iterated as needed and an internal loop that iterates through the data subsets generated through partitioning. The iteration concludes when the number of observations has been reduced to $n_0 \leq 2 \cdot n$, at which point a final selection process is executed.

Algorithm 2 Hierarchical data partitioning algorithm.

```
procedure HierarchicalPartitioning(Input: Criterion, $D, n_0, n_x$; Output: $n_l$, $D^{\text{strip},l}$)
    tol$\leftarrow 1 \times 10^{-4}$
    $l \leftarrow 0,\ m_{\leftarrow} \lceil n_0/n_x \rceil$
    while $n_0 > 2\ n_x$ do
        for $k \in \{1, \cdots, m_l\}$ do
            SelectObservations (Criterion,$k, n_x, F_k^{l,n_x}$, tol; $n_k^l, \xi_k^l$)       ▷ Find MIO
        end for
        $l \leftarrow l + 1$
        $D^{\text{strip,l}} \leftarrow$ Vertical concatenation of $F_k^{l-1,\text{strip}}$, $k = 1, \cdots, m_l$
        $n_l \leftarrow \sum_{k=1}^{m_l-1} n_k^{l-1}$
        $D \leftarrow D^{\text{strip,l}}$
        $n_{l-1} \leftarrow n_l$
    end while
    $k \leftarrow 1$
    SelectObservations (Criterion,$k, n_x, F_k^{l,n_x}$, tol; $n_k^l, \xi_k^l$)       ▷ Find MIO
    $l \leftarrow l + 1$
    $D^{\text{strip,l}} \leftarrow$ Vertical concatenation of $F_k^{l-1,\text{strip}}$, $k = 1, \cdots, m_l$
    $n_l \leftarrow \sum_{k=1}^{m_l-1} n_k^l$
end procedure
```

4. Application examples

In this Section, we apply the formulations of Section 3 to assess the effectiveness of our proposed subsampling approach for large datasets. Initially, we apply our algorithm to a benchmark dataset, previously examined in Deldossi and Tommasi [13]. We evaluate the performance of our algorithm in Section 4.1. Subsequently, in Section 4.2, we expand its application to investigate the influence of partition size and a preliminary sorting step of observations based on the response variable, used in the initial partitioning process. We also analyze the impact of assuming a quadratic model to represent the relation between the response and the covariates.

The dataset used for testing comprises $n_0 = 1 \times 10^6$ observations. Each observation contains one response and $n_c = 10$ covariates. These covariates were generated through random sampling, following distinct distributions:

1. $x_i,\ i \in \{1, 2, 3\}$ follows uniform distributions in the interval $[0, 5]$, i.e. $(x_1, x_2, x_3) \sim (U[0; 5], U[0; 5], U[0; 5])$;
2. $x_i,\ i \in \{4, 5, 6, 7\}$ follows a multivariate normal distribution with $\boldsymbol{\mu}_1 = (0, 0, 0, 0)$ and covariance matrix
$$V_1 = \begin{pmatrix} 9 & -1 & -1 & -1 \\ -1 & 9 & -1 & -1 \\ -1 & -1 & 9 & -1 \\ -1 & -1 & -1 & 9 \end{pmatrix},$$
i.e. $(x_4, x_5, x_6,\ x_7) \sim N(\boldsymbol{\mu}_1, V_1)$;
3. $x_i,\ i \in \{8, 9\}$ follows a Student t distribution with 3 degrees of freedom, $\boldsymbol{\mu}_2 = (0, 0)$ and covariance matrix
$$V_2 = \begin{pmatrix} 4.0 & 0.5 \\ 0.5 & 4.0 \end{pmatrix},$$
i.e. $(x_8, x_9) \sim t_3(\boldsymbol{\mu}_2, V_2)$;
4. $x_i,\ i \in \{10\}$ follows a Poisson distribution with $\lambda = 5$, i.e. $x_{10} \sim P(\lambda)$.

The linear model, as expressed in Eq. (1), used for generating the response variable is defined by $\boldsymbol{\theta} = (0.25, 0.3, 0.2, 0.1, 0.35, 0.28, 0.16, 0.05, 0.08, 0.17, 0.06)^\intercal$, where the first component corresponds to β_0, and the subsequent components correspond to the remaining coefficients associated with the aforementioned random variables. Thus, the size of the FIM is $n_c + 1 = 11$. The observational noise ϵ_c is characterized by a normal distribution with mean 0 and a standard deviation of 0.2, represented as $N(0, 0.2)$.

To enhance the clarity of the forthcoming tables and figures, we will define the following terms:

(i) ξ^{ori}: This represents the uniform design derived from the original dataset X, and it can be expressed as $\xi^{\text{ori}} = (X^\intercal, \mathbf{w})$, with $w_i = 1/n_0,\ i \in \{1, \dots, n_0\}$;

(ii) ξ^{appr}: This denotes the approximate optimal design obtained for the stripped dataset $X^{\text{strip},l}$ at convergence using Algorithm 2. Specifically, ξ^{appr} is defined as $((X^{\text{strip},l})^\intercal, \mathbf{w})$, with $\mathbf{w}$ calculated through SDP;

(iii) ξ^{exac}: This holds for the exact optimal design obtained for the stripped dataset $X^{\text{strip},l}$ at convergence. It can be expressed as $\xi^{\text{exac}} = ((X^{\text{strip},l})^\intercal, \mathbf{w})$, where $\mathbf{w}$ is determined by rounding the SDP-obtained optimal design, while considering the allocation of n_l observations; and

(iv) ξ^{unif}: This represents the uniform optimal design for the stripped dataset $X^{\text{strip},l}$ at convergence, and is expressed as $\xi^{\text{unif}} = ((X^{\text{strip}}, l)^\intercal, \mathbf{w})$, where $w_i = 1/n_l,\ i \in \{1, \dots, n_l\}$.

To evaluate the efficiency of designs, we consider ξ^{appr} as the reference as it provides the optimal allocation of observations, maximizing the information contained within the original dataset.

4.1. Reference scenario

Here, we employ the algorithm on the dataset, D, defined above which contains $n_0 = 1 \times 10^6$ observations, and we set the initial number of partitions, m_0, to 100 which yields $n_x = 1 \times 10^4$.

Table 1 presents the outcomes achieved for D as detailed above. In this table, Column 1 represents the chosen optimality criterion employed in our algorithm. Column 2 contains l^*, the number of iterations (hierarchical levels of partitioning) necessary to solve the problem. Column 3 specifies the count of sampling points within the final dataset (i.e., the size of the subsample derived from the original dataset). Moving forward, Column 4 is for the CPU time, Column 5 quantifies the efficiency of the original dataset in comparison to the approximate design, while Column 6 gives the efficiency of the exact design obtained through rounding. Finally, Column 7 elucidates

Table 1
Results of the subsampling procedure applied to dataset D ($n_0 = 1 \times 10^6$, $m_0 = 100$, $n_x = 1 \times 10^4$). The reference design in Eq. (3) is ξ^{appr}. The number of iterations at convergence is l^*.

Optimality criterion	l^*	n_l^*	CPU time (s)	$\text{Eff}(\xi^{\text{orig}}, \xi^{\text{appr}})$	$\text{Eff}(\xi^{\text{exac}}, \xi^{\text{appr}})$	$\text{Eff}(\xi^{\text{unif}}, \xi^{\text{appr}})$
D-	3	36	498.53	0.2881	0.8868	0.8868
A-	3	39	512.11	0.1699	0.6318	0.6318
E-	3	27	481.12	0.0839	0.3335	0.3335

the efficiency of the corresponding uniform design. These efficiency measurements are calculated using Eqs. (3a)–(3c).

To find exact designs from approximate designs we need to set the number of experiments. Here, we take the number of experiments to be the number of support points of the approximate designs, that is n_{l^*}. Since the primary rule of rounding is allocating an observation to each support point, the exact and uniform designs are often close.

A manifestation of the closeness of the designs in Table 1 is the identity of the numbers in the last two columns. Unless some design points have $w_i < tol$ the allocation is unitary. If some $w_i < tol$, we can have support points with 0 replication and others with replication 2. Due to the small value of tolerance considered, 1×10^{-4}, this situation has not occurred here, nor in the results shown in Tables 2–5.

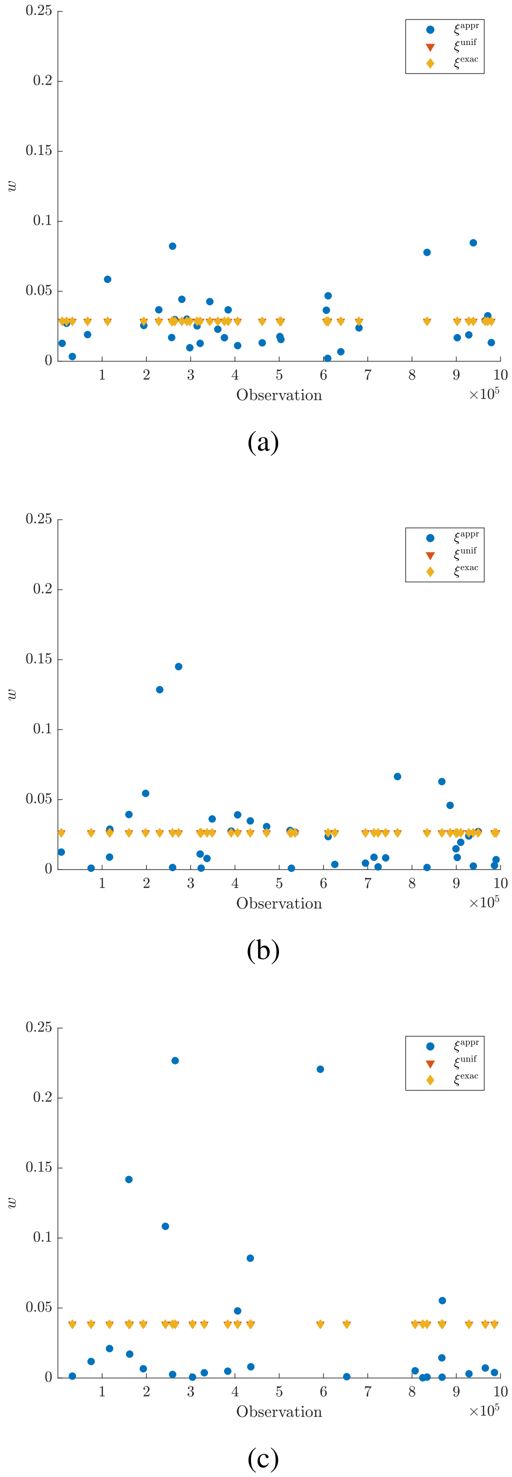
The number of observations in the subsampling approach remains consistently close to 40 for all the optimality criteria we analyzed, that is roughly four times n_c. The efficiencies are below 1.0 since the subsampled observations are chosen to be more informative on a per-observation basis. However, the amount of information derived from the complete dataset surpasses that of the subsample datasets. As expected, due to the rounding process, the efficiencies of the exact designs are lower than their approximate counterparts. This difference is particularly pronounced when we consider the E-optimality criterion.

Figs. 2(a)–2(c) depict the weight distributions of the optimal designs obtained through Algorithm 2, presenting the approximate, exact, and uniform designs. Note that the weights of observations excluded from the subsampling set are set to zero. The rounded optimum designs contain around 20 support points, with the exact design for E-optimality being smallest.

To assess the optimality of the formulations used for identifying the most informative observations within data subsets, we conducted a comparative analysis between our SDP-based tool and the randomized exchange algorithm (REA) proposed by Harman et al. [36], implemented in the R language, as detailed in Harman and Filová [37]. This comparison focused on D-optimal and A-optimal designs and was restricted to the approximate designs generated for the initial data subset ($k = 1$) at the first partition level ($l = 0$) with $n_x = 100$ (implying $n_0 = 1 \times 10^4$) for the example presented above. For both optimality criteria considered, our designs exhibited remarkable efficiency, slightly outperforming the designs obtained using the REA. The efficiency of the D-optimal design achieved through the REA relatively to the equivalent SDP-based design was 0.9974, while the efficiency of the A-optimal design to our SDP-based design was 0.9884. It is important to note that comparing computational efficiency presents challenges. Nevertheless, our experiments demonstrate that the REA outperforms the SDP-based tool in terms of speed for this particular dataset. In both algorithms the size of the optimal subsample is determined by the design algorithm, rather than being pre-specified.

4.2. Extensions

In this Section we extend the application of the algorithm introduced in Section 3.2 by examining three key aspects: (i) we investigate the influence of partition size on the final subsample; (ii) we assess the effects of ordering the data based on the response variable, so replacing contiguous partitions with partitions corresponding to strata of the response. It is worth noting that this technique is not applicable to time series data but can be valuable for unordered data structures;



(a)

(b)

(c)

Fig. 2. Results for simulated dataset ($n_0 = 1 \times 10^6$, $m_0 = 100$). Weights for (a) D-optimality criterion; (b) A-optimality criterion; (c) E-optimality criterion.

Table 2
Results of the subsampling procedure applied to dataset D ($n_0 = 1 \times 10^6$, $m_0 = 200$, $n_x = 5 \times 10^3$). The reference design in Eq. (3) is ξ^{appr}. The number of iterations at convergence is l^*.

Optimality criterion	l^*	n_l^*	CPU time (s)	$\text{Eff}(\xi^{\text{orig}}, \xi^{\text{appr}})$	$\text{Eff}(\xi^{\text{exac}}, \xi^{\text{appr}})$	$\text{Eff}(\xi^{\text{unif}}, \xi^{\text{appr}})$
D-	3	36	572.55	0.2881	0.8868	0.8868
A-	3	41	564.59	0.1699	0.6452	0.6452
E-	3	25	523.48	0.0841	0.4107	0.4107

Table 3
Results of the subsampling procedure applied to dataset D^{sorted} ($n_0 = 1 \times 10^6$, $m_0 = 100$, $n_x = 1 \times 10^4$) after sorting based on the response variable. The reference design in Eq. (3) is ξ^{appr}. The number of iterations at convergence is l^*.

Optimality criterion	l^*	n_l^*	CPU time (s)	$\text{Eff}(\xi^{\text{orig}}, \xi^{\text{appr}})$	$\text{Eff}(\xi^{\text{exac}}, \xi^{\text{appr}})$	$\text{Eff}(\xi^{\text{unif}}, \xi^{\text{appr}})$
D-	3	42	484.89	0.2858	0.8709	0.8709
A-	3	37	483.89	0.1749	0.6442	0.6442
E-	3	28	493.14	0.0870	0.3445	0.3445

and (iii) we explore the impact of employing quadratic polynomials instead of first-order models in Eq. (1).

Table 2 displays the outcomes for the identical dataset, where again $n_0 = 1 \times 10^6$, but now m_0 is increased to 200 with n_x being 5×10^3. These results exhibit strong agreement with those presented in Table 1; the efficiencies are similar. This finding serves as evidence that the algorithm can identify the most informative observations, and that, as expected, this capability remains consistent regardless of the partition size. The CPU times increase around 10 % for all optimality criteria.

To investigate the impact of selecting initial partitions based on distinct strata of the response variable, we initially arrange the observations in ascending order using a sorting tool. In this process, D is transformed into $D^{\text{sorted}} \equiv \{Y^{\text{sorted}}, X^{\text{sorted}}\} \in \mathbb{R}^{n_0 \times (n_r + n_c)}$, and this sorted dataset is subsequently subjected to the algorithm. Similarly to reference scenario we set $m_0 = 100$; consequently, n_x is set to 1×10^4. This approach results in a reduction of information within each stratum while simultaneously increasing the information available in the final sample (at convergence). Table 3 presents the outcomes for X^{sorted}, and these results closely align with those in

Table 1, providing evidence for the minimal impact of the ordering procedure on the final subsample.

We now turn our attention to a quadratic model expressed as

$$y_i = \beta_0 + \boldsymbol{\beta}\ \mathbf{x}_i^{\intercal} + \boldsymbol{\gamma}\ \mathbf{x}_i^{\intercal} \circ \mathbf{x}_i^{\intercal} + \epsilon_i = \boldsymbol{\theta}\ \mathbf{h}^{\intercal}(\mathbf{x}_i) + \epsilon_i. \tag{6}$$

where $\circ$ stands for the elementwise product of vectors and $\mathbf{x}_i^{\intercal} \in \mathbb{R}^{n_c}$, $i \in \{1, \dots, n_0\}$.

Our objective is to identify the optimal design for estimating the $2 \cdot n_c + 1 = 21$ coefficients of this model using an expanded matrix of explanatory variables denoted as D^{expd}. Here, $\boldsymbol{\theta}$ is the vector of regression coefficients defined as $\boldsymbol{\theta} = \text{hcat}(\boldsymbol{\beta}, \boldsymbol{\gamma}) \in \mathbb{R}^{1 \times (2 n_c + 1)}$, and $\mathbf{h}(\mathbf{x}_i)$ is the basis expansion given by $\mathbf{h}(\mathbf{x}_i) \equiv (1, \mathbf{x}_i, \mathbf{x}_i \circ \mathbf{x}_i) \in \mathbb{R}^{1 \times (2 \cdot n_c + 1)}$. To create the expanded dataset, we augment the original dataset by including the squares of the columns of matrix X. Specifically, the inputs dataset is represented as $X^{\text{expd}} \equiv \{\mathbf{x}_{i,j}, \mathbf{x}_{i,j}^2 | i \in \{1, \dots, n_c, \}, \ j \in \{1, \dots, n_0\}\}$. Consequently, the dataset subjected to subsampling is referred to as $D^{\text{expd}} \equiv \text{hcat}(Y, X^{\text{expd}}) \in \mathbb{R}^{n_0 \times (n_r + 2 \cdot n_c)}$.

Due to the larger size of the Fisher Information Matrices (FIMs), the computational processing time increases to approximately 4 times per data partition. The size of the optimal subsets also increases. The results are detailed in Table 4, revealing an increase in the D-optimality efficiency of ξ^{appr} relatively to uniform design obtained by considering all the observations. Specifically, on a per-observation basis, the D-optimal approximate design provides nearly 9 times the information found in the original data, the A-optimal design offers roughly $1/0.1978 \approx 5$ times more information, and the E-optimal design delivers about $1/0.0814 \approx 12$ times more information. These comparisons assess the optimality of designs obtained through the subsampling algorithm when compared to the original datasets, where all observations have equal weight.

Notably, when considering A- and E-optimality criteria, their efficiencies closely match those presented in Tables 1–3. In contrast, the D-optimal efficiency for the quadratic model is appreciably higher. Furthermore, the loss of efficiency in exact and uniform designs relative to approximate designs is more pronounced. In each case, the size of the final subsample is at least twice as large as that achieved with the first-order model (as indicated in Columns 3 of Tables 1–4), and the CPU time required is approximately 5 times greater for all criteria. In summary, employing quadratic models to represent the relationship between the response and the covariates enhances the information content within the final subsample, attributable to the increased number of selected observations.

5. Application to a real dataset

We have employed our proposed methodology on a real-world dataset, specifically referring to white variants of Portuguese *Vinho Verde*. This particular type of wine, known as a “young wine”, is typically released between three to six months after the grapes are harvested. Additional information about *Vinho Verde* is at http://www.vinhoverde.pt/en/ as well as in the work by Cortez et al. [38].

The dataset comprises 11 covariates that provide a physicochemical characterization of the wine, along with one response variable that quantifies its quality through sensory testing. In total, the dataset encompasses 4898 wines. To rigorously evaluate our methodology in this scenario, we followed the step-by-step process outlined in Section Section 4:

(i) initially, we set the number of observations in each partition, denoted as n_x, to 490, which leads to $m_0 = 10$ partitions;
(ii) we, then conducted an assessment of the impact of the subsample size by halving the value of n_x, resulting in $n_x = 245$ and $m_0 = 20$, while keeping the other parameters fixed;
(iii) the dataset was subsequently sorted based on the output variable, and the resulting structure was subjected to our proposed methodology, maintaining n_x at 490 (equivalent to 10 strata);
(iv) lastly, we employed a quadratic polynomial model (see Eq. (6)), which naturally increased the number of parameters to be estimated, specifically to $2 \cdot n_c + 1 = 23$ where $n_c = 11$.

This systematic approach enabled us to thoroughly evaluate the effectiveness and reliability of our methodology across various conditions and configurations.

Fig. 3 illustrates the weights for D-optimal (Fig. 3(a)), A-optimal (Fig. 3(b)), and E-optimal (Fig. 3(c)) designs when n_x is set at 490 with m_0 equal to 10. The exact designs again agree with the uniform designs in each case.

In Table 5, we present a comparison of the subsampling sets generated for all simulated scenarios. For all optimality criteria, the

Table 4
Results of the subsampling procedure applied to dataset D^{expd} ($n_0 = 1\times 10^6,\ m_0 = 100,\ n_0 = 1\times 10^4$) considering the quadratic model (6). The reference design in Eq. (3) is ξ^{appr}. The number of iterations at convergence is l^*.

Optimality criterion	l^*	n^*_l	CPU time (s)	$\text{Eff}(\xi^{\text{orig}}, \xi^{\text{appr}})$	$\text{Eff}(\xi^{\text{exac}}, \xi^{\text{appr}})$	$\text{Eff}(\xi^{\text{unif}}, \xi^{\text{appr}})$
D-	3	207	2189.67	0.1123	0.7641	0.7641
A-	3	99	2796.94	0.1978	0.5695	0.5695
E-	3	79	1882.45	0.0814	0.4402	0.4402

Table 5
Results of the subsampling procedure applied to dataset related to the characterization of white variants of Portuguese *Vinho Verde*. The reference design in Eq. (3) is ξ^{appr}.

Optimality criterion	l	n_l	CPU time (s)	$\text{Eff}(\xi^{\text{orig}}, \xi^{\text{appr}})$	$\text{Eff}(\xi^{\text{exac}}, \xi^{\text{appr}})$	$\text{Eff}(\xi^{\text{unif}}, \xi^{\text{appr}})$
	$n = 490,\ m_0 = 10$, first-order model					
D-	2	21	12.16	0.2702	0.9275	0.9275
A-	2	26	14.23	0.1318	0.8475	0.8475
E-	2	24	11.86	0.0721	0.4555	0.4555
	$n = 245,\ m_0 = 20$, first-order model					
D-	2	21	23.77	0.2699	0.9316	0.9316
A-	2	26	26.73	0.1323	0.8221	0.8221
E-	2	21	21.72	0.0729	0.4376	0.4376
	$n = 490,\ m_0 = 10$, after sorting based on response variable, first-order model					
D-	2	24	12.56	0.2688	0.8964	0.8964
A-	2	26	14.70	0.1322	0.8447	0.8447
E-	2	24	11.72	0.0769	0.4988	0.4988
	$n = 490,\ m_0 = 10$, after dataset expansion, quadratic model					
D-	2	57	28.53	0.1794	0.8807	0.8807
A-	2	68	54.59	0.1193	0.5998	0.5998
E-	2	38	19.58	0.0778	0.4120	0.4120

number of subsampling points for first-order-based models is consistently around 20. Remarkably, the trends observed in the simulated dataset align with those in this specific dataset: (i) both exact and uniform designs yield subsamples with lower information per observation compared to the approximate design; (ii) all design criteria produce subsamples with higher information density per observation than the original dataset, although the absolute information amount in the latter is greater; (iii) for the quadratic model, the number of observations within subsample sets nearly doubles that obtained for the first-order model, except for the E-optimality criterion; (iv) the efficiencies of the designs for the quadratic model are notably higher than those for the first-order model; (v) notoriously, the uniform design, including all observations, is less efficient here than when a first-order model is considered. The reduction in efficiency is due to the increased efficiency of the approximate design relative to that for the first-order model. This consistent pattern highlights the robustness and generalizability of our findings across different datasets and models.

6. Conclusions

We have considered the problem of subsampling large datasets, a task of paramount importance in light of the vast amount of information generated in today's world. Effectively managing the original data without sacrificing significant underlying insights is imperative. To address this challenge, we have adopted an approach grounded in information criteria for the purpose of selecting the most informative observations within a sample. Our approach is rooted in linear models between the response and input variables. We have demonstrated its adaptability on both first-order and quadratic models.

Our procedure provides an optimum subset of the data of size n_{l^*} and an optimum design based on this subset. The parameter estimates from this design can be used to fit a linear model to any subset; plots of residuals against fitted values can then be used to check whether the data are homogeneous. For example, in Section 4.2 we ordered the data by response value. This had no effect on the analysis of our homogeneous simulated data. But if there were outliers in the data, their presence would be indicated by plots of residuals from subsets of differing sorted order. Close to optimal subsets can be investigated by considering subsets for which $l = l^* - 1$. Subsets of a few thousand are sufficiently small that their properties can be thoroughly checked using, for example, the methods for monitoring robust regression of [39]. Such procedures include the properties of the response variables in subset selection.

In Section 1 we mentioned the recent work of Wang et al. [25] who introduced the Information-Based Optimal Subdata Selection (IBOSS) algorithm, which exhibits connections with our proposed algorithm. IBOSS employs a partition-based selection approach to identify observations with extreme values which, based on a bound of the determinant of a Hadamard matrix [40,41], are deemed to be the most informative. The selection process unfolds iteratively, wherein a predetermined number of samples, linked to the extreme values of the first covariate, are initially selected. Subsequently, this procedure is replicated for the remaining covariates by replacing observations in the chosen set by others while upholding a consistent sample size. In essence, this algorithm aligns with D-optimality criteria commonly applied in experimental design, particularly when the goal is to identify a set of points with associated binary variable weights to describe the allocation of observations. This problem shares similarities with the optimization of sensor placement in distributed systems, as discussed by Uciński and Patan [42],Schäfer [43], and can be effectively addressed through the utilization of Mixed Integer Semidefinite Programming (MISDP) solvers, see Gally et al. [44],Duarte [45]. However, if the connections are many, the differences are also noticeable. Specifically, (i) our algorithm incorporates observations into the subsample by quantifying the information contained within each observation, thus accounting for the information across all covariates simultaneously; (ii) it utilizes state-of-the-art convex solvers to deterministically select the most informative observations; (iii) the subsample size is not predetermined but dynamically emerges as a result of the algorithm's observation selection, and tends to be minimally supported; and (iv) other optimality criteria can be easily applied to guide the selection process.

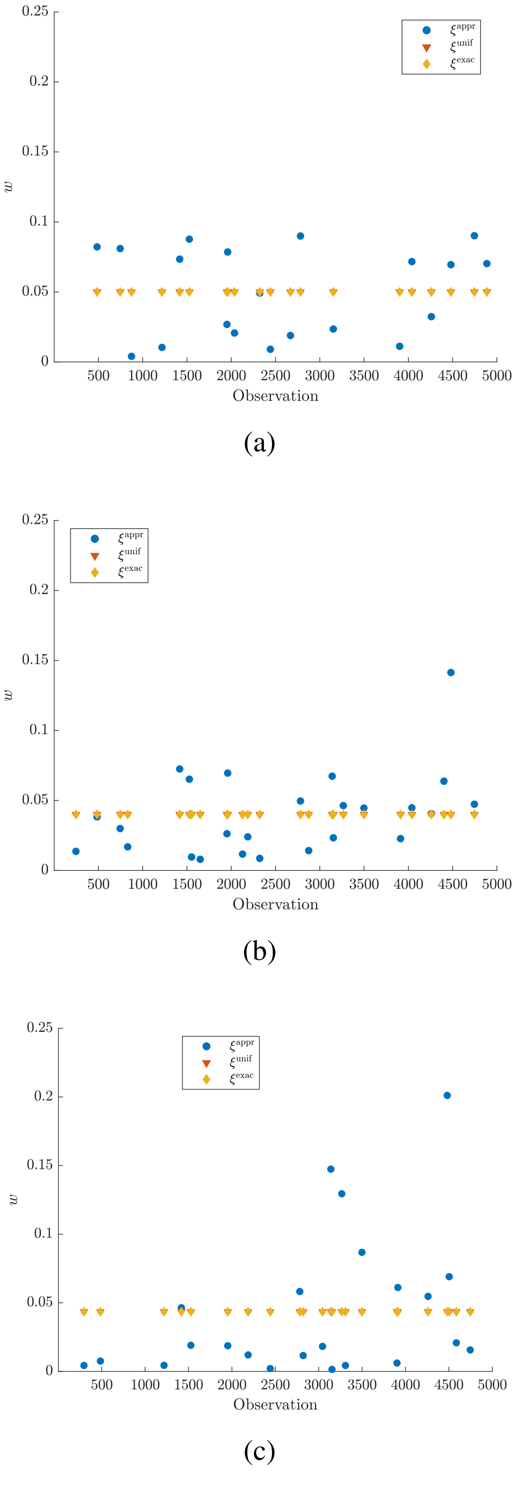


Fig. 3. Results for a real dataset characterizing white variants of Portuguese *Vinho Verde* ($n_0 = 4898$, $m_0 = 10$). Weights for (a) D-optimality criterion; (b) A-optimality criterion; (c) E-optimality criterion.

From a numerical perspective, we have tackled the problem of subset selection by employing SDP formulations within the space of available observations, as elucidated in Section 2.2. We avoid the complexity and prohibitive time of computation from simultaneous consideration of all observations in the dataset, by introducing a novel approach based on hierarchical partitioning. Our tool for implementing this methodology is presented in Section 2.3. The algorithms integrating both procedures are detailed in Section 3.2. The proposed tool partitions the original dataset into slices and identifies the most informative observations within each one. Subsequently, these selected observations are integrated into a "stripped" dataset, and this iterative process continues until the number of observations in the "stripped" dataset is small enough to be analyzed by customary statistical methods. Ultimately, each chosen observation should significantly contribute to representing the underlying information within the data. We successfully tested the proposed tool with a benchmark and a real-world dataset. The real-world dataset is related to the physicochemical characterization of white variants of Portuguese *Vinho Verde*.

It is clear from our results that the mathematical programming tools we have used enable us to efficiently address massive datasets with data mining techniques within reasonable computational times. When confronted with big datasets, hierarchical partitioning offers a valuable approach to surmount the complexities resulting from the high dimensionality. As we have demonstrated, subsampling based on information criteria represents a natural and numerically robust method for extracting significant observations, since it relies on a transparent metric to measure the information contained within each data point and accordingly allows choice of individual observations. An exciting avenue to extend this work involves exploring nonlinear models, such as Gaussian kernel regression models.

CRediT authorship contribution statement

Belmiro P.M. Duarte: Writing – review & editing, Writing – original draft, Validation, Methodology, Investigation, Conceptualization. **Anthony C. Atkinson:** Writing – review & editing, Methodology, Formal analysis. **Nuno M.C. Oliveira:** Writing – review & editing, Methodology, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

We wish to thank two anonymous reviewers for their many comments and suggestions that helped to improve the quality of the paper.

Appendix

A.1. Formulations to determine the optimal allocation via semidefinite programming

Here, we list the SDP formulations for the D–, A– and E–optimality criteria. The first three were introduced in Vandenberghe and Boyd [30, 46] and Ben-Tal and Nemirovski [26]. We start with the formulation for D-optimal designs:

$$\text{Opt} \equiv \max_{\xi, B} \quad t \tag{A.1a}$$

$$\text{s.t.} \quad \begin{pmatrix} M(\xi) & B^{\intercal} \\ B & \text{diag}(B) \end{pmatrix} \succeq 0_{n_\theta} \tag{A.1b}$$

$$t \leq \prod_{i=1}^{n_\theta} B_{i,i}^{1/n_\theta} \tag{A.1c}$$

$$\sum_{i=1}^{k} w_i = 1 \tag{A.1d}$$

$$0 \leq w_i \leq 1, \quad i \in \{1, \ldots, k\}. \tag{A.1e}$$

The formulation for computing A-optimal designs is:

$$\text{Opt} \equiv \min_{\xi, B} \quad t \tag{A.2a}$$

$$\text{s.t.} \quad \begin{pmatrix} M(\xi) & I_{n_\theta} \\ I_{n_\theta} & B \end{pmatrix} \succeq 0_{2 \times n_\theta} \tag{A.2b}$$

$$t \geq \sum_{i=1}^{n_\theta} B_{i,i} \tag{A.2c}$$

$$\sum_{i=1}^{k} w_i = 1 \tag{A.2d}$$

$$0 \leq w_i \leq 1, \quad i \in \{1, \ldots, k\}, \tag{A.2e}$$

Finally, for E-optimal designs, we have:

$$\text{Opt} \equiv \max_{\xi, t} \quad t \tag{A.3a}$$

$$\text{s.t.} \quad M(\xi) - t\ I_{n_\theta} \succeq 0_{n_\theta} \tag{A.3b}$$

$$\sum_{i=1}^{k} w_i = 1 \tag{A.3c}$$

$$0 \leq w_i \leq 1, \quad i \in \{1, \ldots, k\}. \tag{A.3d}$$

References

[1] K. Stewart, Subsampling, in: A.C. Michalos (Ed.), Encyclopedia of Quality of Life and Well-Being Research, Springer Netherlands, Dordrecht, 2014, pp. 6462–6464, http://dx.doi.org/10.1007/978-94-007-0753-5_2909.

[2] Y. Yao, H. Wang, A review on optimal subsampling methods for massive datasets, J. Data Sci. 19 (1) (2021) 151–172.

[3] M.W. Mahoney, Randomized algorithms for matrices and data, Found. Trends Mach. Learn. 3 (2) (2011) 123–224, http://dx.doi.org/10.1561/2200000035.

[4] P. Drineas, M. Mahoney, S. Muthukrishnan, T. Sarlos, Faster least squares approximation, Numer. Math. 117 (2011) 219–249, http://dx.doi.org/10.1007/s00211-010-0331-6.

[5] P. Ma, M. Mahoney, B. Yu, A statistical perspective on algorithmic leveraging, in: E.P. Xing, T. Jebara (Eds.), Proceedings of the 31st International Conference on Machine Learning, Proc. Mach. Learn. Res. 32 (1) (2014) 91–99, URL: https://proceedings.mlr.press/v32/ma14.html.

[6] H. Wang, More efficient estimation for logistic regression with optimal subsamples, J. Mach. Learn. Res. 20 (2018) 132:1–132:59.

[7] J. Yu, H. Wang, M. Ai, H. Zhang, Optimal distributed subsampling for maximum quasi-likelihood estimators with massive data, J. Amer. Statist. Assoc. 117 (537) (2022) 265–276, http://dx.doi.org/10.1080/01621459.2020.1773832.

[8] P. Ma, Y. Chen, X. Zhang, X. Xing, J. Ma, M.W. Mahoney, Asymptotic analysis of sampling estimators for randomized numerical linear algebra algorithms, J. Mach. Learn. Res. 23 (1) (2022).

[9] H.P. Wynn, Minimax purposive survey sampling design, J. Amer. Statist. Assoc. 72 (359) (1977) 655–657.

[10] H.P. Wynn, Optimum submeasures with applications to finite population sampling, in: S.S. Gupta, J.O. Berger (Eds.), Statistical Decision Theory and Related Topics III, Academic Press, 1982, pp. 485–495, http://dx.doi.org/10.1016/B978-0-12-307502-4.50033-7, URL: https://www.sciencedirect.com/science/article/pii/B9780123075024500337.

[11] V.V. Fedorov, Optimal design with bounded density: optimization algorithms of the exchange type, J. Statist. Plann. Inference 22 (1) (1989) 1–13, http://dx.doi.org/10.1016/0378-3758(89)90060-8.

[12] L. Pronzato, On the sequential construction of optimum bounded designs, J. Statist. Plann. Inference 136 (8) (2006) 2783–2804, http://dx.doi.org/10.1016/j.jspi.2004.10.020.

[13] L. Deldossi, C. Tommasi, Optimal design subsampling from Big Datasets, J. Qual. Technol. 54 (1) (2022) 93–101, http://dx.doi.org/10.1080/00224065.2021.1889418.

[14] T. Reuter, R. Schwabe, Optimal subsampling design for polynomial regression in one covariate, Statist. Papers (2023) 1–23, http://dx.doi.org/10.1007/s00362-023-01425-0.

[15] H. Wang, R. Zhu, P. Ma, Optimal subsampling for large sample logistic regression, J. Amer. Statist. Assoc. 113 (522) (2018) 829–844, http://dx.doi.org/10.1080/01621459.2017.1292914.

[16] C.C. Drovandi, C. Holmes, J.M. McGree, K. Mengersen, S. Richardson, E.G. Ryan, Principles of experimental design for big data analysis, Stat. Sci.: a Rev. J. Inst. Math. Stat. 32 (3) (2017) 385.

[17] B.P.M. Duarte, W.K. Wong, H. Dette, Adaptive grid semidefinite programming for finding optimal designs, Stat. Comput. 28 (2) (2018) 441–460.

[18] G. Sagnol, Computing optimal designs of multiresponse experiments reduces to second-order cone programming, J. Statist. Plann. Inference 141 (5) (2011) 1684–1708.

[19] N. Vo-Thanh, R. Jans, E.D. Schoen, P. Goos, Symmetry breaking in mixed integer linear programming formulations for blocking two-level orthogonal experimental designs, Comput. Oper. Res. 97 (2018) 96–110, http://dx.doi.org/10.1016/j.cor.2018.04.001.

[20] X. Huan, Y.M. Marzouk, Simulation-based optimal Bayesian experimental design for nonlinear systems, J. Comput. Phys. 232 (1) (2013) 288–317.

[21] A.M. Overstall, D.C. Woods, Bayesian design of experiments using approximate coordinate exchange, Technometrics 59 (4) (2017) 458–470.

[22] M.S. Mahmud, J.Z. Huang, S. Salloum, T.Z. Emara, K. Sadatdiynov, A survey of data partitioning and sampling methods to support big data analysis, Big Data Min. Anal. 3 (2) (2020) 85–101.

[23] T. Singh, R. Khanna, Satakshi, M. Kumar, Improved multi-class classification approach for imbalanced big data on spark, J. Supercomput. 79 (6) (2023) 6583–6611.

[24] A. Stergiou, R. Poppe, AdaPool: Exponential adaptive pooling for information-retaining downsampling, IEEE Trans. Image Process. 32 (2023) 251–266, http://dx.doi.org/10.1109/TIP.2022.3227503.

[25] H. Wang, M. Yang, J. Stufken, Information-based optimal subdata selection for big data linear regression, J. Amer. Statist. Assoc. 114 (525) (2019) 393–405, http://dx.doi.org/10.1080/01621459.2017.140.

[26] A. Ben-Tal, A.S. Nemirovski, Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications, Society for Industrial and Applied Mathematics, Philadelphia, 2001.

[27] S. Boyd, L. Vandenberghe, Convex Optimization, University Press, Cambridge, 2004.

[28] G. Sagnol, On the semidefinite representation of real functions applied to symmetric matrices, Linear Algebra Appl. 439 (10) (2013) 2829–2843.

[29] J. Kiefer, General equivalence theory for optimum design (approximate theory), Ann. Statist. 2 (1974) 849–879.

[30] L. Vandenberghe, S. Boyd, Applications of semidefinite programming, Appl. Numer. Math. 29 (1999) 283–299.

[31] B.P.M. Duarte, W.K. Wong, Finding Bayesian optimal designs for nonlinear models: A semidefinite programming-based approach, Internat. Statist. Rev. 83 (2) (2015) 239–262.

[32] J. Lin, T. Ji, X. Hao, H. Cha, Y. Le, X. Yu, A. Akella, Towards accelerating data intensive application's shuffle process using SmartNICs, Proc. ACM Meas. Anal. Comput. Syst. 7 (2) (2023) http://dx.doi.org/10.1145/3589980.

[33] Y. Sasaki, A survey on IoT big data analytic systems: Current and future, IEEE Internet Things J. 9 (2) (2022) 1024–1036, http://dx.doi.org/10.1109/JIOT.2021.3131724.

[34] Y. Ye, Interior Point Algorithms: Theory and Analysis, John Wiley & Sons, New York, 1997.

[35] F. Pukelsheim, S. Rieder, Efficient rounding of approximate designs, Biometrika 79 (1992) 763–770.

[36] R. Harman, L. Filová, P. Richtárik, A randomized exchange algorithm for computing optimal approximate designs of experiments, J. Amer. Statist. Assoc. 115 (529) (2020) 348–361, http://dx.doi.org/10.1080/01621459.2018.154.

[37] R. Harman, L. Filová, Package "OptimalDesign", 2022, https://CRAN.R-project.org/package=OptimalDesign.

[38] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, J. Reis, Modeling wine preferences by data mining from physicochemical properties, Decis. Support Syst. 47 (4) (2009) 547–553, http://dx.doi.org/10.1016/j.dss.2009.05.016.

[39] M. Riani, A. Cerioli, A.C. Atkinson, D. Perrotta, Monitoring robust regression, Electron. J. Stat. 8 (2014) 642–673.

[40] J. Hadamard, Résolution d'une question relative aux déterminants, Bull. des Sci. Math. 2 (1893) 240–246, URL: https://cir.nii.ac.jp/crid/1572824501240738944.

[41] J. Brenner, The Hadamard maximum determinant problem, Amer. Math. Monthly 79 (6) (1972) 626–630.

[42] D. Uciński, M. Patan, D-optimal design of a monitoring network for parameter estimation of distributed systems, J. Global Optim. 39 (2007) 291–322.

[43] C. Schäfer, Optimization Approaches for Actuator and Sensor Placement and Its Application to Model Predictive Control of Dynamical Systems (Ph.D. thesis), Fachbereich Mathematik, Technische Universität Darmstadt, Darmstadt, Germany, 2015.

[44] T. Gally, M.E. Pfetsch, S. Ulbrich, A framework for solving mixed-integer semidefinite programs, Optim. Methods Softw. 33 (3) (2018) 594–632, http://dx.doi.org/10.1080/10556788.2017.1322081.

[45] B.P.M. Duarte, Exact optimal designs of experiments for factorial models via mixed-integer semidefinite programming, Mathematics 11 (4) (2023) http://dx.doi.org/10.3390/math11040854.

[46] L. Vandenberghe, S. Boyd, Semidefinite programming, SIAM Rev. 8 (1996) 49–95.