Subsampling under distributional constraints

Florian Combes, Ricardo Fraiman, Badih Ghattas

Subsampling under distributional constraints

Florian COMBES

Aix Marseille University, CNRS, I2M, Marseille, France
Renault Group, Customer usage, Guyancourt, France

Ricardo Fraiman

Centro de Matemática, Universidad de la República, Uruguay

Badih Ghattas

Aix Marseille University, CNRS, I2M, Marseille, France

Abstract

Some complex models are frequently employed to describe physical and mechanical phenomena. In this setting we have an input X in a general space, and an output $Y = f(X)$ where f is a very complicated function, whose computational cost for every new input is very high. We are given two sets of observations of X, S_1 and S_2 of different sizes such that only $f(S_1)$ is available. We tackle the problem of selecting a subsample $S_3 \in S_2$ of smaller size on which to run the complex model f, and such that distribution of $f(S_3)$ is close to that of $f(S_1)$. We suggest three algorithms to solve this problem and show their efficiency using simulated datasets and the Airfoil self-noise data set.

Keywords: Optimal sampling, numerical models, nearest neighbours, Kolmogorov–Smirnov

1. Introduction

Numerical models are often used to model physical or mechanical phenomena [1] [2]. Such models are used to generate some scenarios using the solution of partial derivative equations (PDEs). Their input is often composed of border and initial conditions denoted by X, and their output $f(X)$ may be the values of some variables which may be multidimensional and depend on space and time. The use of such models consists in solving complicated PDEs, and each generated scenario corresponds, in the machine learning paradigm, to an inference for a new input X, thus the computation of $f(X)$. In practice, this computation can take several hours and even days depending on the complexity of the model f and on the granularity of time and space chosen to compute the solution. In this context, machine learning and deep learning algorithms may be used to replace such complicated models by learning an approximation of f, often based on small samples. In what follows we consider the subsampling [3] problem in this framework.

Suppose we have an iid sample $S_1 = \{X_1, \ldots, X_{n_1}\}$, with the same distribution as X, of data defined on a space, that we will assume to be just a complete separable metric space $(\mathcal{E}, \rho)$. We apply to each of these observations a very complicated, expensive and deterministic smooth function $f : \mathcal{E} \to \mathbb{R}$, which we will consider as black box. The pair $(S_1, f(S_1))$ may also be seen as the result of a large establishment survey.

Next, another large iid sample S_2 of size n_2 is provided with the same distribution as the first one, but for which the values $f(S_2)$ are not provided.

The main problem we address is how to provide a subsample $S_3 \subset S_2$ of size n_3 smaller than n_2 and such that the distribution of $f(S_3)$ will be close to that of $f(S_1)$. The idea is that in the future the values of $f(X)$ will be only computed for S_3.

This problem appears quite often in practice, in particular in some industrial applications, semi-supervised learning, neuroscience, big data regression and clustering, among many other problems. At first glance this problem is that of

a classical subsampling. It can be approached by sampling techniques used in surveys, or more recent techniques adapted to unsupervised ([4]) or supervised ([5], [6],[7]) situations. These approaches are tailored to sample from within a population (a large sample, accessible or not). They may be used partially to solve our problem which is semi supervised.

Other approaches consider also the sampling problem with different type of constraints and under uncertainties ([8]). In Design of experiments ([9]) the same problem is faced with other specifications; the sample S_1 is in general of very small size (~ 15) and the target sample S_3 is built sequentially, observations are sampled one by one. Moreover, the set S_3 is multidimensional space, and not a fixed sample. In our context, we need to sample from S_2 with constraints related to S_1.

This manuscript is organized as follows. In Section 2 we fix some notation that will be used throughout the manuscript and we specify the framework of the problem to be solved. In Section 3 we introduce three different algorithms with different alternative procedures to solve the problem. In Section 4 we provide sharp bounds for the probability of being far from the target distribution, which motivates the last algorithm, which is based on these results. Section 5 is devoted to illustrate, with some simulated experiments, the behaviour of the proposed algorithms. Lastly, in Section 6 some concluding remarks are provided.

2. The problem setting

Let $\mathcal{S}_1 = \{X_1, \ldots, X_{n_1}\}$ be a set of n_1 iid random elements in a complete separable metric space $(\mathcal{E}, \rho)$, with the same common distribution μ as X, and $\mathcal{S}_2 = \{X'_1, \ldots, X'_{n_2}\}$ a second iid sample of size n_2 with the same distribution μ. Let $f : \mathcal{E} \to \mathbb{R}$, a deterministic function which is very complicated and hard to compute (which we may think as a regular black box). The unknown distribution of $f(X)$ will be denoted by F. We have a sample

$$\mathcal{Y}_1 =: \{Y_i = f(X_i) \text{ for } i = 1, \ldots n_1\}$$

of the images of the first sample $\mathcal{S}_1$. Images for S_2 are not available.

With this information on hand, we want to find a subsample $\mathcal{S}_3 \subset \mathcal{S}_2$, with size $n_3 << n_2$, such that the empirical distribution of $f(S_3) := \{f(X_j) : X_j \in S_3\}$ will be close to the distribution of $f(X_1)$.

We will consider several different approaches to this problem throughout the manuscript, with different complexities. Some of them do not make use of the sample $\mathcal{Y}_1$, while others do.

Let μ_1 be the empirical distribution of $\mathcal{S}_1$, and given a subset $\mathcal{S}_3 \subset \mathcal{S}_2$, write μ_3 for its empirical distribution. If f is regular, we can look for a subset $\mathcal{S}_3$ for which

$$d(\mu_1, \mu_3), \tag{1}$$

is minimum among all possible subsets of size n_3, and d is a distance that metrizes weak convergence, like the Prokhorov distance. However, this translates the problem into another one which is computationally hard.

In what follows, the corresponding empirical probability measures on the space $(\mathcal{E}, \rho)$ are denoted by $\mu_{ni},\ i = 1, \ldots, 3$ while the ones corresponding to $f(X) \in \mathbb{R}$ are denoted by $F_{ni}\ i = 1, \ldots, 3$.

Remark 1. *We will use in some of the algorithms that follow the notions of distinct and extended nearest neighbours. More precisely, for each data point in S_1 we will look for the nearest point in S_2 to it. It may happen that the set of the nearest neighbours of observations in S_1 from S_2 (having respectively n_1 and n_2 observations) may contain duplicates from S_2. If we simply remove these duplicates, the remaining set of neighbours will be of size less than n_1. We call this set the distinct nearest neighbours. The extended nearest neighbours of S_1 from S_2 refers to the set obtained by adding to the distinct nearest neighbours, further neighbours from S_2 (neighbours which are further away), i.e. in that case we look for the second nearest neighbour, and so on.*

3. Some Algorithms

We first propose a simple solution which does not make use of $\mathcal{Y}_1$. Then we will introduce two algorithms that make use of the output sample $\mathcal{Y}_1$ in different

ways. The idea is to get a subsample $\mathcal{Y}_3$ from $\mathcal{Y}_1$ whose distribution is close to that of $\mathcal{Y}_1$, consider its inverse image $f^{-1}(\mathcal{Y}_3)$, which is a subsample of S_1, and look for its neighbours in S_2. The selection of the optimal subsample $\mathcal{Y}_3$ is based on the results given in Section 4.

3.1. A simple extended nearest neighbours approach

Consider $S_1 = \{X_1, ...X_{n_1}\}$ and $S_2 = \{X'_1, ...X'_{n_2}\}$ with $n_2 > n_1$. Compute the nearest neighbours of S_1 in S_2, let $d_1, ..., d_{n_1}$ be their ordered distances and $j(1), ...j(n_1)$ their indices.

If two observations from S_1, X_i and X_j, have the same nearest neighbour, say X'_l, at distances d_i and d_j, such that $d_i < d_j$, than X'_l will be kept as a neighbour of X_i and for X_j we take its second nearest neighbour from S_2. If more than two observations have the same nearest neighbour, we will need to explore further away neighbours.

We end with the set $\mathcal{S}_3 = \{X_1^{'}, \ldots, X_{n_1}^{'}\}$ and its Prokhorov distance to μ_1 will be smaller than $d^{(n_1)}$ which will be small if n_2 is large and $n_2 >> n_1$. Indeed, $d^{(n_1)}$ will converge to 0 over any compact set $K \subset \mathcal{E}$.

Algorithm 1: Extended nearest neighbours.

$S_1 = [X_1, ..., X_{n_1}]$;

$S_2 = [X'_1, ..., X'_{n_2}]$;

$S_3 =$ Extended nearest neighbours of S_1 in S_2 ;

This simple approach is based on the idea that observations from S_2 close to S_1, should have images through f close to the images of S_1.

3.2. A histogram based approach

In the previous section we did not make use of $\mathcal{Y}_1 = f(S_1)$. Suppose now that we are interested in $\mathbb{P}(f(X) \in I)$, where $I \in \mathbb{R}$ is an interval, or a finite union of disjoint intervals in $\mathbb{R}$, say $I_1, \ldots, I_k$. We look for a subsample $\mathcal{S}_3 \subset \mathcal{S}_2$, with size $n_3 << n_2$, such that we can approach $\mathbb{P}(f(X) \in I)$. We start by considering the set

$$A_{n_1}(I) := \{X_j \in \mathcal{S}_1 : f(X_j) \in I\}.$$

Next, given $\epsilon > 0$, define

$$B_{n_1}(I) := B(A_{n_1}(I), \epsilon) := \bigcup_{X_j \in A_{n_1}(I)} B(X_j, \epsilon),$$

and

$$\mathcal{S}_3(\epsilon) = \{X_i \in \mathcal{S}_2 : X_i \in B(A_{n1}(I), \epsilon)\}.$$

The heuristic idea in this case is to look for a subsample S_3 such that the histogram of $f(S_3)$ is close to the one built up with the intervals I of the distribution of $f(X)$, assuming that the function f is smooth.

In this case, the size of $\mathcal{S}_3(\epsilon)$ is random and depends on ϵ. From an asymptotic point of view, we will need that $\epsilon \to 0$ slowly enough, since we can think of the problem as estimating the distribution of $X_1|f(X_1) \in I$. Some theory can be derived along this line. An alternative is to fix n_3 and choose ϵ in order to have a subsample of size approximately n_3. A special case of this approach is implemented in Algorithm 2 and consists in using bins obtained by adjusting a histogram to $f(S_1)$.

Algorithm 2: Histogram based algorithm.

$S_1 = [X_1, ..., X_{n_1}]$;

$S_2 = [X'_1, ..., X'_{n_2}]$;

$S_3 = \{\emptyset\}$;

Build a histogram for $f(S_1)$ using J bins I_j and consider the sets

$A_j = \{X \in \mathcal{S}_1 : f(X) \in I_j\}$;

for *j in 1..J* **do**

  Z = distinct neighbours of A_j in S_2 ;

  Append Z to S_3;

end

4. Bounds for the probability of being far from a target distribution

In what follows we provide sharp bounds for the probability of being far from a target distribution, when we make use of the sample $\mathcal{Y}_1$ as in Algorithm 2. These results will motivate our proposal of Algorithm 3 given in subsection 4.3 below.

Given a large sample of iid random variables $\mathcal{Y}_1 := \{Y_1, \ldots, Y_n\}$ with distribution F, we look for a subsample of $\mathcal{Y}_1$, of much smaller size, and which is as close as possible in distribution to the original one.

Suppose we fix the size m of the desired subsample. Searching within the class $\mathcal{C}$ of all possible subsamples of size m taken from $\mathcal{Y}_1$ is in general unfeasible in practice from a computational point of view. Thus, we will consider a smaller class defined as follows. We start by considering a partition $\mathcal{C}_n$ of the subset $\subset \{1, \ldots, n\}$ into L disjoint subsets $C_k \subset \{1, \ldots, n\},\ k = 1, \ldots, L$ each of size m. We denote by F_n the empirical distribution of $\mathcal{Y}_1$, F_k the empirical distribution of the set $\{Y_j : j \in C_k\}$, and F_{n-k} the empirical distribution of the set $\{Y_j : j \in \{1, \ldots, n\} \setminus C_k\}$.

Next, define

$$W_{n,k} = \min_{C_k \in \mathcal{C}_n} \|F_k - F_{n-k}\| \tag{2}$$

and

$$\hat{C}_k = \operatorname{argmin}_{C_k \in \mathcal{C}_n} \|F_k - F_{n-k}\|, \tag{3}$$

where $\|F - G\| = \sup_t |F(t) - G(t)|$ denotes the usual supremum distance. In other words, this amounts to using the classical Kolmogorov–Smirnov (KS) ([10]) statistic to assess the distance between two empirical distributions or, when F is known, the distance between the empirical distribution and the theoretical underlying distribution F.

Given m, L and $t > 0$, we want to lower bound the following probability

$$\mathbb{P}(\min_{C_k \in \mathcal{C}_n} \|F_k - F_{n-k}\| \leq t), \tag{4}$$

and to provide an algorithm to find $\hat{C}_k$, for a given family $\mathcal{C}_n$.

Alternatively, we will also consider another version, denoted by $V_{n,k}$. It is obtained by replacing the KS statistic in (2) and (3) with the Cramer–von Mises discrepancy,

$$\mathrm{M}(H_1, H_2) = \int_{\mathbb{R}} (H_1(t) - H_2(t))^2 dH_2(t).$$

That is, we will use the statistic

$$V_{n,k} = \min_{C_k \in \mathcal{C}_n} \int_{\mathbb{R}} (F_k(t) - F_{n-k}(t))^2 dF_{n-k}(t) = \min_{C_k \in \mathcal{C}_n} \mathrm{M}(F_k, F_{n-k}). \quad (5)$$

Remark 2. *The relation between m (the size of the subsample of each C_k) and L, the size of $\mathcal{C}_n$ where we will perform the search, must take into account two different problems. A larger m will make the approximation better, but our purpose is to look for small values of m relative to n_2. On the other hand, this will increase L and therefore the computation time.*

We start by considering the unrealistic situation where the distribution F is known. In the case where it is unknown it will be replaced by F_{n-k}. Note that when F is continuous (which we will assume throughout the paper), the statistic $\sqrt{m}W_{n,k}$ has a continuous distribution not depending on F, due to the distribution-free property of the Kolmogorov–Smirnov statistic. This is also the case for $mV_{n,k}$.

4.1. The case where F is known

In order to lower bound the probability given in (4) we first recall the well known Dvoretzky–Kiefer–Wolfowitz (DKW)[11] inequality.

$$\mathbb{P}(\|F_k - F\| \leq t) \geq 1 - 2e^{-2mt^2}. \quad (6)$$

Since the subsets in $\mathcal{C}_n$ are disjoint, we have independence, and therefore

$$\mathbb{P}(\min_{C_k \in \mathcal{C}_n} \|F_k - F\| \leq t) = 1 - \mathbb{P}(\|F_k - F\| > t)^L \geq 1 - \left(2e^{-2mt^2}\right)^L, \quad (7)$$

which will be small if $e^{-2mt^2} < 0.5$ (that is, $mt^2 > \frac{\ln(2)}{2}$) for L large. For instance, if $L = 10000$, $m = 890$ and $t = 0.02$ the bound is 1.

The corresponding result for the Cramer–von Mises discrepancy follows directly from the fact that

$$\{mM(F_k, F) \geq t\} \subset \{\sqrt{m}\|F_k - F\| \geq \sqrt{t}\}.$$

4.2. The case where F is unknown

In the above development, we assumed that the distribution F was known. The whole approach can be adapted to the case where F is continuous but unknown, relying on the distribution-free properties of the statistics we use. We assume throughout the continuity of F.

Observe that $\|F_k - F_{n-k}\|$ is nothing but the two-sample KS-statistic which is distribution free (whenever F is continuous).

$$\begin{aligned}
&\mathbb{P}(\min_{C_k \in \mathcal{C}_n} \|F_k - F_{n-k}\| \leq t) \\
&\qquad = 1 - \mathbb{P}(\min_{C_k \in \mathcal{C}_n} \|F_k - F_{n-k}\| > t) \\
&\qquad \geq 1 - \mathbb{P}(\min_{C_k \in \mathcal{C}_n} (\|F_k - F\| + \|F - F_{n-k}\|) > t) \\
&\qquad = 1 - \left(\mathbb{P}(\|F_k - F\| > t/2)^L + \mathbb{P}(\|F_{n-k} - F\| > t/2)^L\right) \\
&\qquad \geq 1 - \left(2e^{-mt^2L/2} + 2e^{-(n-m)t^2L/2}\right). \qquad (8)
\end{aligned}$$

Figure 1 shows the behaviour of the obtained lower bounds for simulated Gaussian datasets using $m = 1000$ and $L = 1000$, which shows the good accuracy of inequalities (7 - left) and (8 - right).

4.3. A partition based algorithm

From the previous result we obtain $k << n_1$ for which the distribution of F_k is close to the distribution of F_{n-k}. Assuming that the function f is smooth enough we propose an algorithm with the following steps:

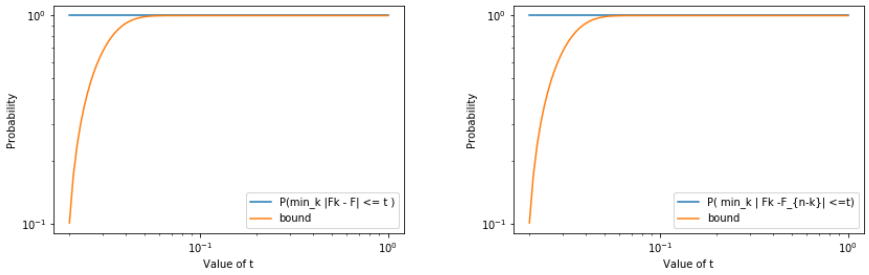

Figure 1: Bound from Equations 7(left) and 8 (right) for $m = 1000$ and $L = 1000$.

- Let $\hat{C}_k = \{Y_{i1}, \ldots, Y_{ik}\}$ be the minimizer obtained in the one dimensional space.
- Consider a $\tilde{C}_k = \{X_{i1}, \ldots X_{ik}\} \subset S_1$ fulfilling $f(\tilde{C}_k) = \hat{C}_k$.
- For each $X_{ij} \in \tilde{C}_k$, find its nearest neighbour in the set S_2. If there are ties, we put the data multiple times.

A detailed description of the algorithm is given below.

Algorithm 3: Partition based algorithm.

$S_1 = [X_1, ..., X_{n_1}]$;
$S_2 = [X'_1, ..., X'_{n_2}]$;
$S_3 = \{\emptyset\}$;
$Y_i = f(X_i)$, with empirical distribution F;
Partition the set $Y_1, .., Y_{n_1}$ into L clusters of size m s.t. $n = mL$;
Denote the clusters by C_k, and their complements by $C_{n-k} = S_1 \setminus C_k$;
Find the partition $\hat{C}_k$ which minimizes $\|F_k - F_{n-k}\|$;
Find the subset $\tilde{C}_k = \{X_{i1}, \ldots X_{ik}\} \subset S_1$ fulfilling $f(\tilde{C}_k) = \hat{C}_k$;
for *i in* $\tilde{C}_k$ **do**
    Z = nearest neighbour of i in S_2 ;
    Append Z to S_3;
end

The partitions in this algorithm may be obtained at random or using a randomized clustering algorithm like k-means. Note that the size of the obtained subsamples in this case are fixed and equal to m.

5. Some experiments

In this section we provide the results of some simulations for the three algorithms proposed; extended nearest neighbours (algorithm 1), histogram based (algorithm 2), and partition based (algorithm 3). As mentioned above, our objective is to find a subsample $S_3 \subset S_2$ for which the distribution of the set $f(S_3)$ will be close to that of $f(S_1)$, without using the values of $f(S_2)$.

To do this, we will apply each algorithm to data generated from various distributions, varying sample sizes (n_1 and n_2) as well as the dimension d of the inputs. The values of these parameters are fixed as follows: $n_1 = 100$, except for Algorithm 3 where we also experimented with the values $n_1 = 400$, $n_2 \in \{200, 1000, 5000\}$, $d \in \{1, 2, 50\}$ except for graphs where $d = 5$ is used instead of $d = 1$.

We use the following distributions; the coordinates are independent for $d > 1$:

- beta distribution $\mathcal{B}(0.5, 0.5)$.
- Gaussian distribution, $\mathcal{N}(0, 1)$.
- uniform distribution, $\mathcal{U}[-1, 1]$.
- truncated Gaussian distribution, where the intervals of truncation are $[-1, 1]$ and $[-2, 2]$. We denote it by $\mathcal{TN}(1)$ and $\mathcal{TN}(2)$.
- uniform distribution over graphs, where d in the number of nodes in the graph, and probability for each edge is 0.5.

Note that the graphs we generate are neither oriented nor acyclic. They are represented in our runs by their adjacency matrix reshaped into a d^2 length vector. For the function $f : \mathcal{E} \to \mathbb{R}$ we used $f(x) = ||x^2|| + \frac{<a.x>}{d}$ where a is the

fixed real sequence ranging from -0.5 to 0.5 with step $\frac{1}{d}$.

Once we get the output subset S_3 from any algorithm we use the Kolmogorov–Smirnov test to compare the observed empirical distributions of $f(S_1)$, with that of $f(S_3)$ which is not available in general. We report the values of the test statistic as well as the corresponding p-values, averaged over $K = 100$ runs for each configuration.

Tables 1 and 2 give the results for Algorithm 1, Tables 3 and 4 for Algorithm 2, and Tables 5 and 6 for Algorithm 3.
The three algorithms obtain a subset S_3 for which the distribution of $f(S_3)$ is very close to that of $f(S_1)$ according to the Kolmogorov test (with a significance value of 0.01), for all the values of n_2, except for the case of the Gaussian distribution in dimension 20. In all cases, the p-values decrease with the dimension, as expected. The values of the test statistic are often lower with Algorithm 1; this is due to the fact that Algorithm 1 outputs systematically a subsample S_3 of exactly the same size as that of S_1, whereas the two other algorithms give smaller subsamples.

The only case where the obtained subsamples are not satisfactory was for the Gaussian distribution in dimension 20. Using the truncated Gaussian distributions (over $[-1, 1]$ and $[-2, 2]$) improved the results. We think that the actual version of our algorithms might fail for tailed distributions in high dimensions.

Distribution	n_2	Dimension 1			Dimension 2			Dimension 20		
		n_3	Stat	Pvalue	n_3	Stat	Pvalue	n_3	Stat	Pvalue
	200	100	0.04	0.99	100	0.06	0.97	100	0.12	0.53
$\mathcal{B}(0.5, 0.5)$	1000	100	0.02	1	100	0.04	1	100	0.13	0.44
	5000	100	0.01	1	100	0.03	1	100	0.13	0.41
	200	100	0.05	0.99	100	0.08	0.88	100	0.22	0.07
$\mathcal{N}(0, 1)$	1000	100	0.02	1	100	0.05	1	100	0.30	0.01
	5000	100	0.02	1	100	0.04	1	100	0.31	0.01
	200	100	0.04	1	100	0.06	0.98	100	0.11	0.57
$\mathcal{U}[-1, 1]$	1000	100	0.02	1	100	0.04	1	100	0.13	0.45
	5000	100	0.01	1	100	0.03	1	100	0.12	0.48

Table 1: Average size of n_3, Kolmogorov-Smirnov test between S_1 and S_3 for beta, Gaussian and uniform distribution (Algorithm 1).

Distribution	n_2	$d = 1$			$d = 2$			$d = 20$		
		n_3	Stat	Pvalue	n_3	Stat	Pvalue	n_3	Stat	Pvalue
	200	100	0.05	1	100	0.07	0.94	100	0.20	0.14
$\mathcal{TN}(1)$	1000	100	0.02	1	100	0.05	1	100	0.27	0.02
	5000	100	0.02	1	100	0.05	1	100	0.26	0.02
	200	100	0.04	1	100	0.07	0.93	100	0.22	0.08
$\mathcal{TN}(2)$	1000	100	0.02	1	100	0.05	1	100	0.29	0.01
	5000	100	0.02	1	100	0.04	1	100	0.30	0.00
		$d = 5$			$d = 10$			$d = 20$		
	200	94	0.05	1	100	0.08	0.91	100	0.14	0.28
Graphs	1000	98	0.05	1	100	0.09	0.82	100	0.11	0.58
	5000	92	0.03	1	100	0.10	0.70	100	0.10	0.70

Table 2: Average size of n_3, Kolmogorov-Smirnov test between S_1 and S_3 for truncated Gaussian distribution and for graphs (Algorithm 1).

Distribution	n_2	$d=1$			$d=2$			$d=20$		
		n_3	Stat	Pvalue	n_3	Stat	Pvalue	n_3	Stat	Pvalue
	200	48	0.06	0.99	48	0.08	0.96	47	0.16	0.43
$\mathcal{B}(0.5, 0.5)$	1000	47	0.06	0.99	47	0.06	0.99	47	0.16	0.42
	5000	47	0.06	0.99	48	0.06	0.99	47	0.16	0.43
	200	48	0.08	0.92	48	0.11	0.80	47	0.30	0.03
$\mathcal{N}(0, 1)$	1000	48	0.08	0.93	47	0.09	0.93	48	0.32	0.01
	5000	48	0.08	0.92	47	0.08	0.96	47	0.33	0.01
	200	48	0.06	0.99	48	0.07	0.97	48	0.14	0.53
$\mathcal{U}[-1, 1]$	1000	47	0.06	0.99	48	0.06	0.99	47	0.15	0.47
	5000	47	0.06	0.99	48	0.06	0.99	47	0.15	0.45

Table 3: Average size of n_3, Kolmogorov-Smirnov test between S_1 and S_3 for beta, Gaussian and uniform distribution (Algorithm 2).

Distrbution	n_2	$d=1$			$d=2$			$d=20$		
		n_3	Stat	Pvalue	n_3	Stat	Pvalue	n_3	Stat	Pvalue
	200	47	0.07	0.99	47	0.09	0.89	48	0.26	0.11
$\mathcal{TN}(1])$	1000	48	0.06	0.99	47	0.07	0.97	48	0.30	0.03
	5000	48	0.07	0.98	48	0.07	0.99	47	0.29	0.04
	200	47	0.07	0.98	48	0.09	0.89	47	0.28	0.04
$\mathcal{TN}(2)$	1000	48	0.06	0.98	48	0.08	0.97	48	0.32	0.03
	5000	47	0.06	0.99	48	0.07	0.98	47	0.32	0.01
		$d=5$			$d=10$			$d=20$		
	200	46	0.13	0.63	48	0.10	0.84	46	0.16	0.37
Graphs	1000	45	0.08	0.97	47	0.08	0.97	48	0.11	0.81
	5000	45	0.04	1	46	0.13	0.58	48	0.09	0.93

Table 4: Average size of n_3, Kolmogorov-Smirnov test between S_1 and S_3 for truncated Gaussian distribution and for graphs (Algorithm 2).

Distribution	n_1	n_2	$d=1$			$d=2$			$d=20$		
			n_3	Stat	Pvalue	n_3	Stat	Pvalue	n_3	Stat	Pvalue
$\mathcal{B}(0.5,0.5)$	100	200	50	0.10	0.86	50	0.11	0.82	50	0.17	0.43
		1000	50	0.09	0.85	50	0.10	0.83	50	0.17	0.38
		5000	50	0.08	0.92	50	0.09	0.88	50	0.16	0.43
	400	200	50	0.08	0.91	50	0.09	0.85	50	0.15	0.39
		1000	50	0.08	0.92	50	0.07	0.93	50	0.14	0.40
		5000	50	0.07	0.93	50	0.08	0.93	50	0.14	0.41
$\mathcal{N}(0,1)$	100	200	50	0.09	0.90	50	0.11	0.76	50	0.30	0.04
		1000	50	0.09	0.88	50	0.10	0.84	50	0.32	0.01
		5000	50	0.09	0.91	50	0.10	0.85	50	0.32	0.01
	400	200	50	0.08	0.91	50	0.09	0.83	50	0.29	0.02
		1000	50	0.08	0.93	50	0.08	0.88	50	0.32	0.00
		5000	50	0.08	0.93	50	0.08	0.92	50	0.30	0.00
$\mathcal{U}[-1,1]$	100	200	50	0.10	0.84	50	0.10	0.83	50	0.15	0.51
		1000	50	0.09	0.88	50	0.10	0.81	50	0.16	0.46
		5000	50	0.09	0.90	50	0.09	0.91	50	0.15	0.46
	400	200	50	0.08	0.91	50	0.09	0.86	50	0.15	0.43
		1000	50	0.08	0.93	50	0.08	0.89	50	0.14	0.39
		5000	50	0.07	0.95	50	0.07	0.94	50	0.14	0.44

Table 5: Average size of n_3, Kolmogorov-Smirnov test between S_1 and S_3 for beta, Gaussian and uniform distribution (Algorithm 3).

Distrbution	n_1	n_2	$d=1$			$d=2$			$d=20$		
			n_3	Stat	Pvalue	n_3	Stat	Pvalue	n_3	Stat	Pvalue
$\mathcal{TN}(1)$	100	200	50	0.10	0.87	50	0.11	0.78	50	0.27	0.08
		1000	50	0.09	0.88	50	0.10	0.80	50	0.29	0.03
		5000	50	0.09	0.88	50	0.09	0.88	50	0.28	0.05
	400	200	50	0.08	0.91	50	0.09	0.82	50	0.27	0.05
		1000	50	0.08	0.94	50	0.08	0.89	50	0.27	0.03
		5000	50	0.07	0.94	50	0.08	0.92	50	0.27	0.02
$\mathcal{TN}(2)$	100	200	50	0.10	0.85	50	0.11	0.77	50	0.31	0.04
		1000	50	0.09	0.86	50	0.10	0.82	50	0.33	0.02
		5000	50	0.09	0.91	50	0.09	0.87	50	0.31	0.02
	400	200	50	0.08	0.90	50	0.09	0.83	50	0.29	0.03
		1000	50	0.08	0.93	50	0.08	0.87	50	0.30	0.01
		5000	50	0.07	0.94	50	0.08	0.91	50	0.30	0.01
			$d=5$			$d=10$			$d=20$		
Graphs	100	200	50	0.11	0.85	50	0.13	0.67	50	0.14	0.51
		1000	50	0.10	0.83	50	0.10	0.90	50	0.13	0.56
		5000	50	0.13	0.61	50	0.17	0.28	50	0.11	0.80
	400	200	50	0.10	0.79	50	0.09	0.85	50	0.21	0.06
		1000	50	0.12	0.59	50	0.08	0.91	50	0.10	0.75
		5000	50	0.09	0.88	50	0.15	0.23	50	0.10	0.74

Table 6: Average size of n_3, Kolmogorov-Smirnov test between S_1 and S_3 for truncated Gaussian distribution and for graphs (Algorithm 3).

5.1. Comparing to other sampling approaches

Our algorithms aim to construct a subsample S_3 from S_2 using information or constraints related to S_1. Existing subsampling approaches cannot achieve this task but may be used as an alternative to some steps in our algorithms. We will use such approaches to obtain a subsample S_3 directly form S_2 without considering neither S_1 or $\mathcal{Y}_1$ in algorithm 1, and as an alternative in algorithms 2 and 3 to select a subsample from S_1. To do that, we consider two recent unsu-

pervised approaches; the "support points" [4] and the "D-optimality" sampling [5].

5.1.1. *Support points*

In the Support points algorithm, we have a fixed distribution F and we look for a set of observations the best representing F. Those points are obtained by minimizing an energy distance (Székely and Rizzo 2004)

$$E(F, F_n) = \frac{2}{n}\sum_{i=1}^{n}\mathbb{E}||x_i - Y|| - \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbb{E}||x_i - x_j||,$$

where $Y \sim F$. The minimization of the energy distance can be formulated as a difference-of-convex program. For real valued random variables, energy distance is nothing but twice Cramer-von Mises discrepancy. For our setting the empirical version of E is optimized

$$\hat{E}(\{x_i\}, \{y_m\}) = \frac{2}{nN}\sum_{i=1}^{n}\sum_{m=1}^{N}||y_m - x_i||_2 - \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbb{E}||x_i - x_j||_2,$$

where $\{y_m\}$ is a sample from F. The algorithm for the support points using one sample batch is defined as follows [4]

- Sample $D^{(0)} = \{x_i^{(0)}\}_{i=1}^{n}$ i.i.d. from $\{y_m\}_{m=1}^{N}$
- Set $l = 0$, and repeat until convergence of $D^{(l)}$:
  - For $i = 1, .., n$ do parallel:
    Set $x_i^{(l+1)} \leftarrow M_i\left(D^{(l)}; \{y_m\}_{m=1}^{N}\right)$,.
  - Update $D^{(l+1)} \leftarrow \{x_i^{(l+1)}\}_{i=1}^{n}$, and set $l \leftarrow l + 1$.
- Return the converged point set $D^{(\infty)}$

where

$$M_i\left(\{x_j'\}_{j=1}^{n}; \{y_m\}_{m=1}^{N}\right) = q^{-1}\left(x_i'; \{y_m\}_{m=1}^{N}\right)\left(\frac{N}{n}\sum_{j=1;j\neq i}^{n}\frac{x_i' - x_j'}{||x_i' - x_j'||_2} + \sum_{m=1}^{N}\frac{y_m}{||x_i' - y_m||_2}\right),$$

and

$$q\left(x_i; \{y_m\}_{m=1}^{N}\right) = \sum_{m=1}^{N}||x_i - y_m||_2^{-1}.$$

Let $w_m^i = ||x_i - y_m||_2^{-1}$ and $\beta_j^i = ||x'_i - x'_j||_2^{-1}$. M may be written

$$M_i\left(\{x'_j\}_{j=1}^n; \{y_m\}_{m=1}^N\right) = \frac{N}{n\sum_{m=1}^N w_m^i} \sum_{j=1; j\neq i}^{n} \beta_j^i (x'_i - x'_j) + \frac{\sum_{m=1}^N w_m^i y_m}{\sum_{m=1}^N w_m^i}$$

It can be remarked that M_i is composed of two terms; the second term is a weighted average of the y_m sample with weights inversely proportional to the distance of y_m to x'_i;

Support points may be applied in our problem either to the input sample (S_1 or S_2) or to the output $\mathcal{Y}_1$. The obtained sample from support points is composed of arbitrary observations which are not part of the original sample. So when applying this approach we take as a final sample, the nearest neighbors of the obtained support points.

5.1.2. D-optimality

In [5] the authors suggest a procedure for subsampling for linear regression; however, the selection procedure is unsupervised and based on D-optimality. If $x = \{x_1, x_2, ..., x_N\}$ is the data set at hand, such that $x_i \in R^p$, and n is the size of the desired subsample, let $r = n/2p$, and $x_{\bullet,j} = \{x_{ij}, i = 1..N\}$ the observations of the j_{th} coordinate λ of X. The process of sampling is the following:

- Let $S_n = \emptyset$
- For $j = 1.., p$
  - Let X_j the set observations from X having the r largest values and the r smallest values within $x_{\bullet,j}$
  - Let $S_n = S_n \cup X_j$, and $X = X \setminus X_j$.
- The disared subsample is S_n

Distribution	Selection	n_2	$d=1$			$d=2$			$d=20$		
			n_3	Stat	Pvalue	n_3	Stat	Pvalue	n_3	Stat	Pvalue
$\mathcal{B}(0.5, 0.5)$	SP	200	50	0.04	1	50	0.06	0.99	50	0.13	0.61
		1000	50	0.03	1	50	0.05	1	50	0.13	0.64
		5000	50	0.02	1	50	0.04	1	50	0.13	0.63
	D-opt	200	50	0.26	0.02	48	0.13	0.63	40	0.14	0.58
		1000	50	0.26	0.02	48	0.12	0.65	40	0.15	0.57
		5000	50	0.26	0.02	48	0.13	0.63	40	0.14	0.59
$\mathcal{N}(0, 1)$	SP	200	50	0.05	1	50	0.08	0.95	50	0.32	0.02
		1000	50	0.03	1	50	0.06	0.99	50	0.36	0.01
		5000	50	0.03	1	50	0.06	1	50	0.37	0
	D-opt	200	50	0.31	0.01	48	0.29	0.01	40	0.29	0.06
		1000	50	0.32	0	48	0.31	0.01	40	0.30	0.04
		5000	50	0.31	0	48	0.32	0.01	40	0.29	0.06
$\mathcal{U}[-1, 1]$	SP	200	50	0.04	1	50	0.07	0.99	50	0.13	0.61
		1000	50	0.03	1	50	0.05	1	50	0.13	0.60
		5000	50	0.02	1	50	0.05	1	50	0.13	0.59
	D-opt	200	50	0.26	0.02	48	0.15	0.46	40	0.15	0.53
		1000	50	0.26	0.02	48	0.15	0.44	40	0.14	0.59
		5000	50	0.26	0.02	48	0.15	0.44	40	0.14	0.59

Table 7: Average size of n_3, Kolmogorov-Smirnov test between S_1 and S_3 for beta, Gaussian and uniform distribution (Algorithm 2 on X).

Distrbution	Selection	n_2	$d=1$			$d=2$			$d=20$		
			n_3	Stat	Pvalue	n_3	Stat	Pvalue	n_3	Stat	Pvalue
$\mathcal{TN}(1)$	SP	200	50	0.04	1	50	0.07	0.98	50	0.18	0.34
		1000	50	0.03	1	50	0.06	1	50	0.20	0.23
		5000	50	0.03	1	50	0.05	1	50	0.20	0.23
	D-opt	200	50	0.25	0.03	48	0.22	0.11	40	0.19	0.33
		1000	50	0.25	0.02	48	0.23	0.07	40	0.20	0.29
		5000	50	0.25	0.02	48	0.23	0.07	40	0.20	0.32
$\mathcal{TN}(2)$	SP	200	50	0.04	1	50	0.08	0.97	50	0.26	0.07
		1000	50	0.03	1	50	0.06	0.99	50	0.31	0.01
		5000	50	0.03	1	50	0.05	1	50	0.31	0.02
	D-opt	200	50	0.28	0.01	48	0.28	0.02	40	0.26	0.10
		1000	50	0.29	0.01	48	0.28	0.02	40	0.27	0.09
		5000	50	0.29	0.01	48	0.29	0.02	40	0.25	0.12
			$d=5$			$d=10$			$d=20$		
Graphs	SP	200	50	0.07	0.95	50	0.12	0.69	50	0.12	0.70
		1000	50	0.06	0.98	50	0.11	0.73	50	0.13	0.65
		5000	50	0.04	0.99	50	0.10	0.81	50	0.13	0.64
	D-opt	200	47.23	0.09	0.86	40	0.13	0.72	40	0.14	0.65
		1000	46.24	0.08	0.93	40	0.12	0.74	40	0.14	0.62
		5000	45.61	0.08	0.92	40	0.12	0.75	40	0.15	0.60

Table 8: Average size of n_3, Kolmogorov-Smirnov test between S_1 and S_3 for truncated Gaussian distribution and for graphs(Algorithm 2 on X).

Distribution	Selection	n_2	$d=1$			$d=2$			$d=20$		
			n_3	Stat	Pvalue	n_3	Stat	Pvalue	n_3	Stat	Pvalue
$\mathcal{B}(0.5, 0.5)$	SP	200	50	0.04	1	50	0.06	0.99	50	0.14	0.58
		1000	50	0.03	1	50	0.05	1	50	0.13	0.66
		5000	50	0.02	1	50	0.04	1	50	0.12	0.68
	D-opt	200	50	0.26	0.02	50	0.25	0.03	50	0.13	0.61
		1000	50	0.26	0.02	50	0.26	0.02	50	0.13	0.61
		5000	50	0.26	0.02	50	0.26	0.02	50	0.15	0.50
$\mathcal{N}(0, 1)$	SP	200	50	0.04	1	50	0.08	0.95	50	0.30	0.03
		1000	49.98	0.03	1	50	0.06	0.99	50	0.35	0
		5000	50	0.02	1	50	0.04	1	50	0.35	0.01
	D-opt	200	50	0.26	0.02	50	0.23	0.06	50	0.30	0.03
		1000	50	0.26	0.02	50	0.25	0.03	50	0.34	0.01
		5000	50	0.26	0.02	50	0.26	0.02	50	0.35	0.01
$\mathcal{U}[-1, 1]$	SP	200	50	0.04	1	50	0.07	0.99	50	0.14	0.57
		1000	50	0.03	1	50	0.05	1	50	0.13	0.61
		5000	50	0.02	1	50	0.04	1	50	0.13	0.63
	D-opt	200	50	0.26	0.02	50	0.24	0.04	50	0.14	0.55
		1000	50	0.26	0.02	50	0.26	0.02	50	0.14	0.58
		5000	50	0.26	0.02	50	0.26	0.02	50	0.14	0.51

Table 9: Average size of n_3, Kolmogorov-Smirnov test between S_1 and S_3 for beta, Gaussian and uniform distribution (Algorithm 2 on $\mathcal{Y}$).

Distribution	Selection	n_2	$d=1$			$d=2$			$d=20$		
			n_3	Stat	Pvalue	n_3	Stat	Pvalue	n_3	Stat	Pvalue
$\mathcal{TN}(1)$	SP	200	50	0.04	1	50	0.07	0.98	50	0.18	0.34
		1000	50	0.03	1	50	0.05	1	50	0.21	0.18
		5000	49.98	0.02	1	50	0.04	1	50	0.21	0.21
	D-opt	200	50	0.26	0.02	50	0.24	0.04	50	0.18	0.35
		1000	50	0.26	0.02	50	0.26	0.02	50	0.20	0.21
		5000	50	0.26	0.02	50	0.26	0.02	50	0.21	0.19
$\mathcal{TN}(2)$	SP	200	50	0.04	1	50	0.08	0.94	50	0.26	0.07
		1000	50	0.03	1	50	0.06	1	50	0.30	0.03
		5000	50	0.02	1	50	0.05	1	50	0.30	0.02
	D-opt	200	50	0.26	0.03	50	0.23	0.06	50	0.27	0.07
		1000	50	0.26	0.02	50	0.25	0.03	50	0.29	0.03
		5000	50	0.26	0.02	50	0.26	0.02	50	0.30	0.02
			$d=5$			$d=10$			$d=20$		
Graphs	SP	200	11	0.22	0.72	24	0.15	0.75	39	0.14	0.66
		1000	12	0.25	0.67	22	0.16	0.76	39	0.13	0.70
		5000	9	0.29	0.58	21	0.17	0.69	38	0.14	0.65
	D-opt	200	47	0.10	0.82	50	0.11	0.72	50	0.13	0.65
		1000	47	0.16	0.43	50	0.10	0.79	50	0.12	0.67
		5000	47	0.20	0.19	50	0.11	0.73	50	0.13	0.65

Table 10: Average size of n_3, Kolmogorov-Smirnov test between S_1 and S_3 for truncated Gaussian distribution and for graphs(Algorithm 2 on $\mathcal{Y}$).

We will use Support Points (SP) and D-optimality as alternatives in the histogram based approach (algorithm 2). Recall that in algorithm 2 we use a histogram to partition $\mathcal{Y}_1$, and sample 50% of the observations within each bin; we consider then their inverse image by f which is a subsample from S_1. We replace this process by either sampling from $\mathcal{Y}_1$ using support points and D-optimality, either sampling directly from S_1.

Tables 7 and 8 give the results obtained when using SP over X in on histogram based algorithm tables 9 and 10 give the results when apply SP and D-optimality over $\mathcal{Y}$.

Using SP gives better results when compared to the other approaches for all the simulations models mainly in dimension lower than 5. D-optimality is less efficient except for large dimensions (20) where its performance is close to that of SP. These observations are correct whether the selection is done over X or $\mathcal{Y}$.

Moreover, the SP and the D-optimality behave like the other algorithms with the normal distribution, i.e. they are less good on the normal distribution than on the truncated normal distributions.

For the uniform distribution, SP gives very good results, while the D-optimality is poor except for dimension 20 where both give the same results.

For graphs, SP works less well than for the other distributions in small dimensions (< 20). On the contrary the D-optimality works well even in small dimension.

5.2. Airfoil self-noise data set

The airfoil self-noise dataset [12] comes from NASA. It was obtained from a series of aerodynamic and acoustic tests of two- and three-dimensional airfoil blade sections conducted in an anechoic wind tunnel. The data set comprises NASA 0012 airfoils of different sizes at various wind tunnel speeds and angles of attack. It has 1503 observations and 6 real attributes, and comes with a regression problem. The input variables are Frequency (in Hertz), Angle of attack (in degrees), Chord length (in meters), Free-stream velocity (in meters per second) and Suction side displacement thickness (in meters). The only output variable is the Scaled sound pressure level (in decibels). There are no missing values.

In order to test our algorithms with this dataset we split it randomly into two disjoint parts S_1 of size n_1 and S_2 of size n_2. We tried two values for n_1: 100 and 400. The output values for S_2 are used only to assess the performance of our algorithms. The process was repeated $K = 100$ times and the results are reported in Table 11.

Note that compared to the simulations, the function f is unknown but the values of the output variable are considered as the observation of $f(X)$.

The results are satisfactory on most of the algorithms, accept the D-optimality. Moreover we notice that for Support Points the fact of sampling over Xor $\mathcal{Y}$ give similar results. Finally, we can see that Support Points on $\mathcal{Y}$ and Histogram based algorithm give the same results as Extended nearest neighbours but with twice less elements in S3.

Algorithm		n_1	n_2	n_3	Stat	Pvalue
Extended nearest neighbours		100	1403	100	0.11	0.58
		400	1103	314	0.04	0.89
Histogram based algorithm		100	1403	50	0.13	0.61
		400	1103	198	0.07	0.61
Partition based algorithm		100	1403	50	0.13	0.62
		400	1103	50	0.11	0.62
Support Point	Over X	100	1403	50	0.13	0.63
		400	1103	50	0.11	0.62
	Over $\mathcal{Y}$	100	1403	50	0.12	0.70
		400	1103	50	0.10	0.71
D-optimality	Over X	100	1403	50	0.12	0.71
		400	1103	200	0.07	0.49
	Over $\mathcal{Y}$	100	1403	50	0.10	0.88
		400	1103	200	0.12	0.13

Table 11: Average results for airfoil data over $K = 100$ runs, varying values of n_1 and n_2.

6. Conclusion

We have considered the problem of selecting a new subsample S_3 to use for inference with a model f whose output is available only for a small subsample S_1. The subsample should be small, and the distribution of $f(S_3)$ should be close to that of $f(S_1)$. Three algorithms were considered. The first one, an extended nearest neighbours approach, makes no use of the sample $f(S_1)$, while the other two (histogram based and partition based approaches) make use of it in different ways. For the last one a mathematical consistency result was given. All the algorithms showed a good behaviour when analysed through simulation using different distributions and different dimensions for the input, except for the Gaussian case in high dimensions. The results obtained on a real dataset showed that all the algorithms behave as expected.

Our algorithms are currently extended to the case where the input data are time series and the output is multidimensional.

References

[1] S. L. Brunton, B. R. Noack, P. Koumoutsakos, Machine learning for fluid mechanics, Annual Review of Fluid Mechanics 52 (1) (2020) 477–508. arXiv:https://doi.org/10.1146/annurev-fluid-010719-060214, doi:10.1146/annurev-fluid-010719-060214.
URL https://doi.org/10.1146/annurev-fluid-010719-060214

[2] M. S. Brunton Steven L., Hemati, T. Kunihiko, Special issue on machine learning and data-driven methods in fluid dynamics, Theoretical and Computational Fluid Dynamics 34 (2020) 333–337. doi:10.1007/s00162-020-00542-y.
URL https://doi.org/10.1007/s00162-020-00542-y

[3] C. Wu, M. E. Thompson, Sampling Theory and Practice, ICSA Book Series in Statistics, Springer, Cham, 2009. doi:10.1007/978-3-030-44246-0.

[4] R. Joseph V., S. Mak, Support points, The Annals of Statistics 46 (6A) (2018) 2562–2592. doi:10.1214/17-AOS1629.
URL https://doi.org/10.1214/17-AOS1629

[5] W. HaiYing, Y. Min, S. John, Information-based optimal subdata selection for big data linear regression, Journal of the American Statistical Association 114 (525) (2019) 393–405. arXiv:https://doi.org/10.1080/01621459.2017.1408468, doi:10.1080/01621459.2017.1408468.
URL https://doi.org/10.1080/01621459.2017.1408468

[6] W. HaiYing, Z. Rong, M. Ping, Optimal subsampling for large sample logistic regression, Journal of the American Statistical Association 113 (522) (2018) 829–844, pMID: 30078922. arXiv:https://doi.org/10.1080/01621459.2017.1292914, doi:10.1080/01621459.2017.1292914.
URL https://doi.org/10.1080/01621459.2017.1292914

[7] V. R. Joseph, S. Mak, Supervised compression of big data, Statistical Analysis and Data Mining: The ASA Data Science Journal 14 (3) (2021) 217–229. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/sam.11508, doi:10.1002/sam.11508.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/sam.11508

[8] R. E. Amri, R. Le Riche, C. Helbert, C. Blanchet-Scalliet, S. Da Veiga, A sampling criterion for constrained bayesian optimization with uncertainties, working paper or preprint (Mar. 2021).
URL https://hal-emse.ccsd.cnrs.fr/emse-03167452

[9] V. Fedorov, Theory of Optimal Experiments Designs, 1972.

[10] F. J. M. Jr., The kolmogorov-smirnov test for goodness of fit, Journal of the American Statistical Association 46 (253) (1951) 68–78. arXiv:https://www.tandfonline.com/doi/pdf/10.1080/01621459.1951.10500769, doi:10.1080/01621459.1951.10500769.
URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1951.10500769

[11] A. Dvoretzky, J. Kiefer, J. Wolfowitz, Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator, The Annals of Mathematical Statistics 27 (3) (1956) 642–669. doi:10.1214/aoms/1177728174.
URL https://doi.org/10.1214/aoms/1177728174

[12] D. Dua, C. Graff, UCI machine learning repository (2017).
URL http://archive.ics.uci.edu/ml