Subsampling bias and the best-discrepancy systematic cross validation

Liang Guo[1], Jianya Liu[2,*] & Ruodan Lu[3,4]

[1]*Data Science Institute, Shandong University, Weihai 264209, China;*
[2]*Data Science Institute, Shandong University, Jinan 250100, China;*
[3]*School of Architecture, Building and Civil Engineering, Loughborough University,*
Loughborough LE11 3TU, UK;
[4]*Darwin College, University of Cambridge, Cambridge CB3 9EU, UK*

Email: liangguo@sdu.edu.cn, jyliu@sdu.edu.cn, r.lu@lboro.ac.uk

Abstract Statistical machine learning models should be evaluated and validated before putting to work. Conventional k-fold Monte Carlo cross-validation (MCCV) procedure uses a pseudo-random sequence to partition instances into k subsets, which usually causes subsampling bias, inflates generalization errors and jeopardizes the reliability and effectiveness of cross-validation. Based on ordered systematic sampling theory in statistics and low-discrepancy sequence theory in number theory, we propose a new k-fold cross-validation procedure by replacing a pseudo-random sequence with a best-discrepancy sequence, which ensures low subsampling bias and leads to more precise expected-prediction-error (EPE) estimates. Experiments with 156 benchmark datasets and three classifiers (logistic regression, decision tree and naïve bayes) show that in general, our cross-validation procedure can extrude subsampling bias in the MCCV by lowering the EPE around 7.18% and the variances around 26.73%. In comparison, the stratified MCCV can reduce the EPE and variances of the MCCV around 1.58% and 11.85%, respectively. The leave-one-out (LOO) can lower the EPE around 2.50% but its variances are much higher than the any other cross-validation (CV) procedure. The computational time of our cross-validation procedure is just 8.64% of the MCCV, 8.67% of the stratified MCCV and 16.72% of the LOO. Experiments also show that our approach is more beneficial for datasets characterized by relatively small size and large aspect ratio. This makes our approach particularly pertinent when solving bioscience classification problems. Our proposed systematic subsampling technique could be generalized to other machine learning algorithms that involve random subsampling mechanism.

Keywords subsampling bias, cross validation, systematic sampling, low-discrepancy sequence, best-discrepancy sequence

MSC(2010) 62-07, 11J71, 62G09, 68T05

1 Introduction

All statistical machine learning processes start with a basic truism: garbage in, garbage out. This means that nonsense input data usually produce nonsense conclusions. However, "garbage in, garbage out" is not

---

* Corresponding author

only about data quality, but also about the way in which data scientists evaluate and choose statistical machine learning models. Such evaluation often relies on a cross-validation (CV) to estimate model generalization performance metrics. The holdout procedure is the simplest kind of CV, in which a data set is split into a training set and a test set. However, the holdout is not reliable as the evaluation depends heavily on which data points end up in the training set and which in the test set [23]. Thus, the evaluation may be significantly different depending on how the subsampling is made [23]. The k-fold Monte Carlo CV (MCCV), the most popular CV procedure, is an effective way to reduce the subsampling bias in the holdout procedure. Given a dataset of n instances, an MCCV procedure is to randomly generate k subsets, each of which contains m instances, where $n = km$. That is to say, each instance is first assigned an index between 1 and n to construct a subsampling frame. Using the simple random subsampling (SRS) method, the MCCV procedure then uses a pseudo-random number generator (PRNG) to generate a random integer sequence. This sequence consists of n distinct, randomly ordered integers between 1 and n, each of which refers to an element of the subsampling frame. Next, the random integer sequence is split into k equal-sized parts and the corresponding instances are partitioned into k subsets. Subsequently, k iterations of training and testing are performed such that in each iteration, a different subset is held out to estimate the EPE while the remaining $k - 1$ subsets are used for training [40].

However, relying on the SRS technique to subsample instances to training and test sets not only generates high computational costs, but also poses risks [5, 31]. The inductive bias of a model and the subsampling bias in the SRS may be confounded and lead to inflated EPEs [31]. This is especially true for small, imbalanced, and high dimensional datasets [15]. The random integer sequence used in the SRS poses a threat to the MCCV's validity and reliability, as the sequence may seem to be random, but not truly random and not truly uniformly distributed. This is because the elements in such sequence often clump at certain arbitrary intervals [16]. When performing SRS, similar instances could be included in one subset, meaning that information may be over- (clusters) or under-represented (voids) in that subset [6]. This is just like the random shuffle function of *Spotify* (a music player App) cannot ensure songs from a given artist are spread throughout the playlist. To avoid the risk of obtaining dissimilar subsets, the MCCV procedure should be repeated p times (usually $p = 50$) with different random seeds to circumvent the risk of obtaining greatly dissimilar subsets [31], as there are trillions of ways to partition n instances into k subsets.

The reliability of the MCCV increases as the k and p increase. When $k = n$, the MCCV becomes the leave-one-out (LOO) CV. In practice, the LOO is not widely used because it is more computationally demanding and has a larger variance than the MCCV [23]. The stratified MCCV procedure is another mean to alleviate the subsampling bias issue [36]. The stratified MCCV procedure starts by dividing a dataset into non-overlapping strata based on its class label vector. It then randomly and proportionally partitions instances into k subsets from each stratum. While this purposive sampling method ensures the same distribution of class values in all subsets, it may not ensure the same distributions of feature values. Thus, subsampling bias problem may remain unsolved. A low subsampling bias is crucial because no matter how sophisticated a statistical machine learning algorithm is, the resulting model may be biased if its training sets fail to represent the idiosyncrasies of the complete set [5, 31]. Likewise, subsampling bias and insufficiently informative test data will inflate the EPEs, jeopardizing the validity and reliability of a CV procedure [15]. We may run the risk of choosing a biased model and drawing a wrong conclusion.

This study follows the ordered systematic subsampling (OSS) theory [29] to propose a simple yet effective method, the best-discrepancy systematic subsampling method for k-fold CV (hereafter the BDSCV), which ensures a smaller subsampling bias than the MCCV, stratified CV and LOO. The contribution of our proposed systematic subsampling method lies in the fact that we make a step further to reduce subsampling bias than the conventional systematic subsampling one, i.e., the first subsampling instance is not chosen at random as in the conventional OSS method. Instead, it is chosen based on the best-discrepancy sequence (BDS), a sequence that is a uniformly distributed sequence, finite or infinite, in which the number of any proportion of its elements falling into an arbitrary interval proportionally converts to the distance of the interval [32]. Our subsampling interval is also determined by the BDS.

This article is organised as follows. We first introduce the theories of the low discrepancy sequence

(LDS) and the BDS. We then illustrate subsampling traps in conventional k-fold CV. Next, we propose the best-discrepancy systematic subsampling method and the BDSCV procedure. We conduct experiments to provide empirical supports to our theories. Finally, we discuss our proposed procedure and its limitations.

2   Methods

2.1   Low-discrepancy sequences and best-discrepancy sequences

For brevity, we restrict our discussion to the key definitions of low-discrepancy sequence (LDS) and best-discrepancy sequence (BDS), which are necessary to understand our procedure. We refer interested readers to the comprehensive reviews of [13, 20, 26].

Let $\mathscr{X}$ denote a sequence $x_1, x_2, \ldots, x_n, \ldots$ in the unit interval $[0,1)$. The discrepancy D_N of its consecutive N elements is defined by

$$D_N = \sup_{0 \leqslant a < b < 1} \left| \frac{C([a,b); N)}{N} - (b - a) \right|, \tag{2.1}$$

where the counting function $C([a,b); N)$ denotes the number of element x_n with $1 \leqslant n \leqslant N$ for which $x_n \in [a, b)$ [26]. This $\mathscr{X}$ is said to be an LDS if D_N is low for all sufficiently large integers N. The sequence $\mathscr{X}$ is uniformly distributed in $[0,1)$ if $D_N \to 0$ as $N \to \infty$ [26]. In simple terms, the sequence $\mathscr{X}$ is uniformly distributed if every subinterval $[a, b)$ eventually obtains its proper share in $[0, 1)$. A sequence's level of uniformity is measured by its discrepancy function D_N. A classical theorem of Chung [9] states that, in general, the PRNG does not produce a sequence with D_N lower than $N^{-1/2}$. In any case, the order of the discrepancy D_N cannot be lower than $N^{-1} \log N$ by [38]. Thus, the best we can hope for D_N is something like

$$D_N \leqslant cN^{-1+\varepsilon}, \tag{2.2}$$

where $\varepsilon > 0$ is arbitrarily small, and $c > 0$ is some constant. This explains the following definition.

Definition 2.1.   A sequence $\mathscr{X}$ is called a BDS, if its discrepancy function D_N satisfies (2.2) for arbitrary $\varepsilon > 0$ and some constant $c > 0$.

In the following we will present some BDSs that are produced by methods from number theory. Of particular interests to this paper are those BDSs of the form

$$\{n\alpha\}, \quad n = 1, 2, 3, \ldots, \tag{2.3}$$

where $\{\alpha\}$ denotes the fractional part of a real number α. Plainly each term of (2.3) falls into the unit interval $[0,1)$. Denote by $\mathscr{X}(\alpha)$ the sequence (2.3).

Theorem 2.2.   *The sequence $\mathscr{X}(e)$ is a BDS, where e is the base of natural logarithm.*

Proof.   It follows from [1] that (2.2) holds for $\mathscr{X}(e)$ with the constant c therein depending only on ε, and by [30] the dependence of c on ε is effectively computable. See also the arguments in [20, Chapter 4, Section 5]. $\square$

Remark 2.3.   Of course the sequence $\mathscr{X}(\alpha)$ can be a BDS for many other α. For example, it is known [20, Chapter 4, Section 5] that if α is real algebraic, then the sequence $\mathscr{X}(\alpha)$ is a BDS. Examples of these α include $\sqrt{p}$ with p ranging over all prime numbers p. Prior studies also propose other ways to generate BDS, for example the Halton sequence [19] and the van der Corput sequence [44]. Note that our sequence $\mathscr{X}(e)$ in Theorem 2.2 belongs to none of the above examples since e is transcendental. In this paper we have followed the suggestion of Hua and Wang [20, Chapter 4, Section 5] to use $\mathscr{X}(e)$ rather than those previous sequences, and our numerical experiments support their claim that $\mathscr{X}(e)$ is of important practice implications.

The discrepancy properties of LDS/BDS have attracted a lot of attention [14]. Indeed, point sampling is an important research area in computational mathematics. For example, Xu and Zhou [46] proposed a deterministic method to produce the interpolation points and showed its outstanding performance

with $\mathscr{L}1$ minimization. Previous studies also report that the LDS can accelerate the convergence for Monte Carlo quadrature formulas [32] and its points are correlated to provide greater uniformity [41]. The LDS has been applied in many fields other than mathematics, including optimization [17, 25, 34, 43], machine learning [2, 3, 8], computer graphics and vision [22, 24, 27, 37, 45], finance [4, 35, 39, 42], and engineering [7, 11, 21, 28]. Our study is inspired from these studies, although they do not directly address CV procedures. We develop the best-discrepancy systematic subsampling method and the BDSCV procedure based on the BDS.

2.2 Numerical remarks on Subsection 2.1

Panels A–C in Figure 1 show the scatter plots of three random sequences, each of which contains 500 elements. The three sequences contain clear clumps and voids so that it is highly possible that the distance between two consecutive random elements may be either very small or very large. To formally compute the D_N of each sequence, we randomly choose 50 pairs of interval $[a, b]$ and compute the discrepancy to uniform distribution for these two sequences using (2.1), respectively. Results are shown in Figure 2, from which we find that the average D_N and the variance of 50 measures of each random sequence are quite large. That is to say, these three sequences are not quite uniformly distributed.

Panel D in Figure 1 illustrates a BDS of 500 elements. The interval between two consecutive elements of a BDS is determined by the property of uniformity so that the elements scatter evenly to avoid clumps and voids. We formally compute the D_N of BDS by randomly choosing 50 pairs of interval $[a, b]$ using (2.1). Figure 2 shows that the resulting discrepancy of BDS is close to zero and the variance of the 50 measures of BDS is greatly smaller than that of its random counterpart, i.e., unlike the elements of a random number sequence, the elements of a BDS are as uniformly distributed as possible. Successive elements are inserted in a position as far as possible from the other elements in order to avoid clustering and voids. The elements generated sequentially fill in the larger gaps between the previous elements of the sequence [8].

Random sequences cause subsampling bias in k-fold CV. To illustrate subsampling traps, we generate an artificial dataset of 16 instances with four binary features and one label vector with 16 classes (i.e., each combination of the values of the four binary features corresponds to one class). We duplicate this dataset five times and form a complete set of 80 instances. A logistic regression model is used to map
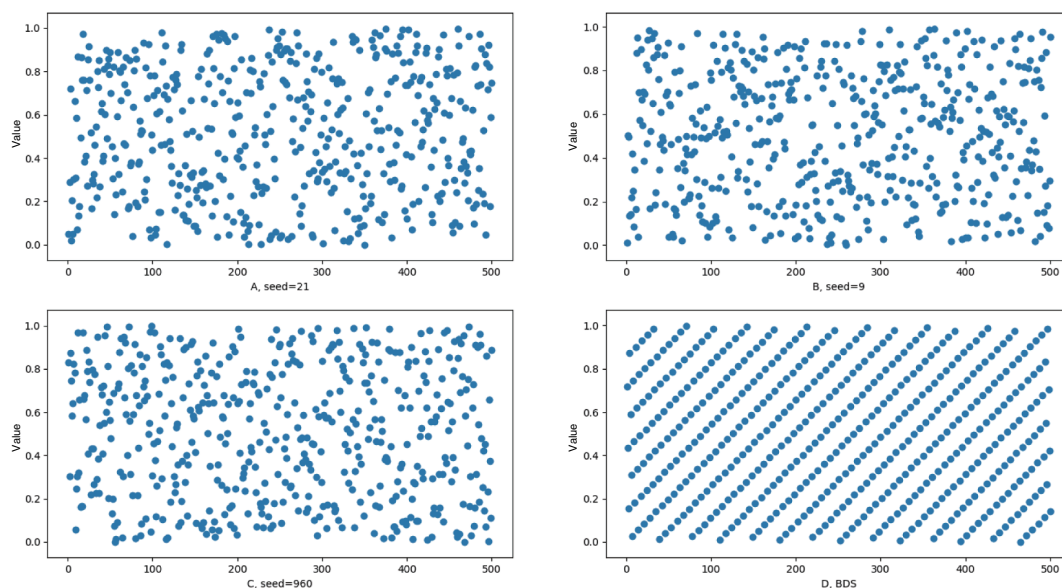
Figure 1   (Color online) The scatter plots of random sequences and of BDS
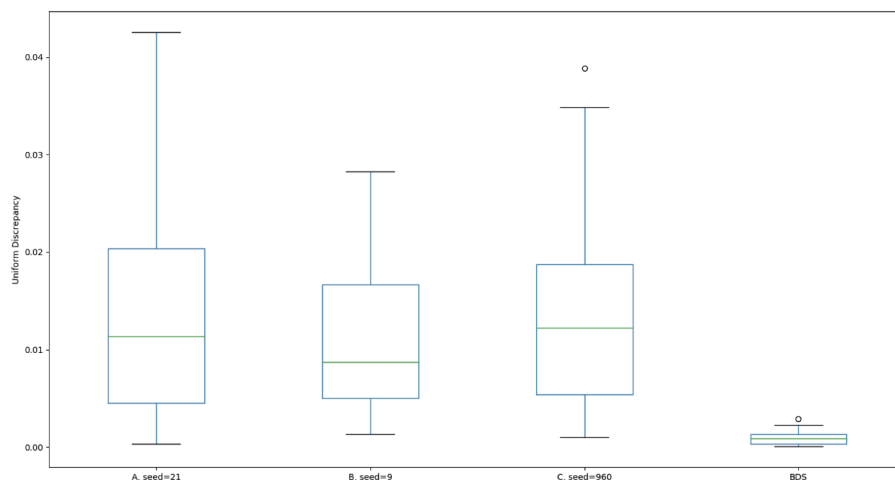
Figure 2   (Color online) The uniform discrepancy of random sequences and of BDS
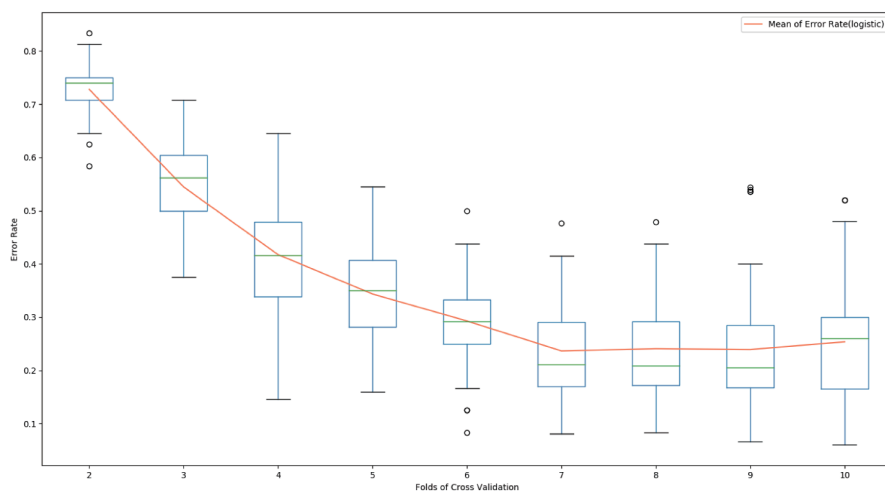
Figure 3   (Color online) The error rate in conventional k-fold CV

out the relations between features and labels. This should be an easy task, as the classification boundary of this dataset are perfectly clear. Therefore, if the 80 instances are well partitioned into representative subsets, then the error rate of both Holdout and CV procedure should be zero.

We conduct k-fold CV with the logistic regression model, where k ranges from 2 to 10. Each k-fold CV procedure is repeated 50 times with different random seeds to circumvent the risk of obtaining largely dissimilar subsets. Results are shown in Figure 3. The error rate of the k-fold CV is surprisingly high and the error rate drops when the number of k increases. However, the error rates are always larger than zero. The box plots also suggest that there are large variations over 50 repetitions. This illustration vividly demonstrates that there are actually traps in conventional k-fold CV. A major threat to the validity and reliability of k-fold CV lies in subsampling bias by the clumps and voids illustrated in Figure 1, which largely jeopardize the representativeness of subsets. Results of model evaluation and selection could be problematic.

2.3   The best-discrepancy systematic subsampling method

We propose a systematic subsampling method based on the uniformity property of the BDS. The theoretical foundation of our proposed method stems from sampling theory, which proves that the ordered

systematic subsampling (OSS) method can obtain smaller sampling error than the SRS and the stratified sampling [29]. The OSS method works as follows. A complete dataset should be firstly sorted either ascendingly or descendingly so that adjacent elements tend to be more similar than elements that are farther apart. Then, a fixed subsampling interval (also called a "skip") k is defined as $k = n/m$, where n is the dataset size and m is the subset size. The subsampling starts by selecting an element from the ordered complete dataset at random and then every k-th element is selected until a subset of m elements is formed. The ordered systematic subsampling outperforms the SRS because the former can avoid choosing two elements adjacent to each other and the resulting subsets are forced to compose of various values, meaning that the discrepancy between subsets caused by subsampling bias can be reduced [29].

We improve the OSS by replacing the fixed subsampling interval with a BDS. Suppose $D = (d_1, \ldots, d_n)$ is a one-dimensional dataset with n instances. We use this vector notation to emphasise the importance of the order of D. Let

$$\widetilde{D} = (\widetilde{d}_1, \ldots, \widetilde{d}_n)$$

be the sorted D, meaning that $\widetilde{D}$ is a permutation of the coordinates of D, so that

$$\widetilde{d}_1 < \cdots < \widetilde{d}_n.$$

This $\widetilde{D}$ is to be divided into k subsets, each of which contains m instance (i.e., $n = km$). To do so, an integer sequence derived from a BDS is needed that refers to the subsampling frame of a dataset.

Let $X = (x_1, \ldots, x_n)$ be the vector of the first n elements of $\mathscr{X}(e)$. Let $\widetilde{X} = (\widetilde{x}_1, \ldots, \widetilde{x}_n)$ be the sorted X, meaning that $\widetilde{X}$ is a permutation of the coordinates of X, so that

$$\widetilde{x}_1 < \cdots < \widetilde{x}_n.$$

The ranking vector $R = (r_1, \ldots, r_n)$ of X is such that $\widetilde{x}_{r_j} = x_j$ for $j = 1, \ldots, n$. Then, given $n = km$, we can chunk R into k disjoint subsets of the same size m, i.e.,

$$R = \bigcup_{1 \leqslant j \leqslant k} R_j, \tag{2.4}$$

where $R_1 = (r_1, \ldots, r_m)$, $R_2 = (r_{m+1}, \ldots, r_{2m})$, etc. This R inherits the property of uniformity. We can use R to partition $\widetilde{D}$ into k subsets

$$\widetilde{D} = \bigcup_{1 \leqslant j \leqslant k} \widetilde{D}_j,$$

where for $j = 1, \ldots, k$,

$$\widetilde{D}_j = (\widetilde{d}_r : r \in R_j). \tag{2.5}$$

Note that each subset $\widetilde{D}_j$ is written as a vector for the same reason.

Our proposed best-discrepancy systematic subsampling method can make sure the instances in a subset as heterogeneous as possible while different subsets as homogenous as possible. According to the theory of analysis of variance [29], we know that the sum of squares of total (SSTO) is decomposed into the sum of squares within subsets (SSW) plus the sum of squares between subsets (SSB). The level of heterogeneity of instances within subsets can be measured by the intra-class correlation coefficient (ICC) [29],

$$\text{ICC} = 1 - \frac{km}{km - 1} \frac{\text{SSW}}{\text{SSW} + \text{SSB}}, \tag{2.6}$$

where

$$\text{SSW} = \sum_{i=1}^{k} \sum_{j=1}^{m} (d_{ij} - \overline{d}_i), \tag{2.7}$$

$$\text{SSB} = n \sum_{i=1}^{k} (\overline{d}_i - \overline{d}), \tag{2.8}$$

$$\text{SSTO} = \text{SSW} + \text{SSB}. \tag{2.9}$$

In addition, Equation (2.6) can be transformed as

$$\text{SSW} = \frac{(1 - \text{ICC})(n - 1)\text{SSTO}}{n}. \tag{2.10}$$

Equation (2.10) implies that SSW becomes larger when ICC becomes negative (i.e., the instances within a subset are dispersed more than a randomly chosen subset would be). Prior study [29] proves that to have a negative ICC, the dataset should be ordered, so that it exhibits a positive autocorrelation (i.e., subsequent instances have similar or the same values) and instances should be chosen with a skip from the ordered dataset so that a subset is forced to compose of both small values and large values. The best-discrepancy systematic subsampling method follows exactly these guidelines to allocate instances. In addition, it is important to mention that the subsampling interval is naturally and uniformly set by the uniformity property of a BDS. The subsampling interval of the BDS guarantees that a subset is composed of heterogeneous instances.

2.4 Numerical remarks on Subsection 2.3

We illustrate the relations among the BDS, ICC and SSW in the following example. Suppose a one-dimensional dataset D is given, and the 21 instances of D are to be partitioned into three subsets with (2.5). We use $\mathscr{X}(e)$ to generate a tiny BDS using e's 38 digits after the decimal place (see the first column of Table 1). The property of uniformity of the BDS ensures that the distribution of values in a subset should be the same or close to that of the complete set, and the between-subset variance should equal or be close to zero. For example, the second column of Table 1 demonstrates that the variances of the three subsets ($\sigma_1^2 = \sigma_2^2 = \sigma_3^2 = 0.08$) are all close to that of the complete set $\sigma_X^2 = 0.08$, provided that the complete set is evenly chunked into three subsets using the BDS. The third and fourth columns of Table 1 show that the ranking vector R of the BDS exhibits the same uniformity property as the means of the three subsets are quite close (10, 10, and 12 respectively).

Then, D is ascendingly sorted as $\widetilde{D}$ so that adjacent instances tend to be more similar than instances that are farther apart. The 21 instances are partitioned into three subsets with (2.5) (see the fifth to seventh columns of Table 1). The three subset variances are quite close to that of D ($\sigma_D^2 = 0.01$) and the SSB is almost zero (0.0082). As expected, the ICC is negative (-0.1242) and the SSW is 0.2168. The instances with adjunct or the same values (for example, the 9th, 10th and 11th elements of D) are forced to spread out between different subsets. The ranking vector R makes the distributions of values highly similar among three subsets, as none of them consists of unrepresentative (i.e., mainly low or high) values.

To compare, we use the SRS to split D into three subsets and repeat the procedure 50 times. The average of SSB of these SRS procedures is 0.0208, which is 2.531 times larger than that of the best-discrepancy systematic subsampling method. This experiment shows that subsampling bias can be greatly reduced by the proposed method, even though the sample size of this tiny dataset is only 21. We conduct an additional experiment with 500 datasets of different sizes (ranging from 54 to 999) and with different levels of variance (ranging from 0 to 0.025), skewness (ranging from 0 to 0.116), and kurtosis (ranging from 0.506 to 31.398). The results suggest that regardless the characteristics of a dataset, the SSB of the best-discrepancy systematic subsampling method is much smaller (23.16%) than that of the SRS.

In short, this example proves that the best-discrepancy systematic subsampling method can satisfy Lohr's [29] two conditions that make the ICC negative. In contrast, the SRS cannot guarantee all subsets are of heterogeneous instances. Indeed, there are too many possibilities to randomly generate k subsets. It is then difficult to avoid "bad", quasi-homogeneous subsets with only or mainly small or large values. Therefore, the best-discrepancy systematic subsampling procedure outperforms the SRS in terms of subset representativeness.

Table 1   Elements with the best discrepancy systematic subsampling method

BDS		R		$\widetilde{D}$		
X_I	subset$_{X_I}$	R_I	subset$_{R_I}$	$\widetilde{D}_I$	$\widetilde{d}_i$ after partition	subset$_{d_i}$
0.7183		16		1.50_1	1.70_{16}	
0.4366		10		1.53_2	1.67_{10}	
0.1548	I	4	I	1.55_3	1.60_4	I
0.8731	$\mu_1 : 0.44$	19	$\mu_1 : 10$	1.60_4	1.80_{19}	$\mu_1 : 1.66$
0.5914	$\sigma_1^2 : 0.08$	13	$\sigma_1^2 : 36$	1.62_5	1.68_{13}	$\sigma_1^2 : 0.01$
0.3097		7		1.63_6	1.64_7	
0.0280		1		1.64_7	1.50_1	
0.7463		17		1.65_8	1.76_{17}	
0.4645		11		1.66_9	1.67_{11}	
0.1828	II	5	II	1.67_{10}	1.62_5	II
0.9011	$\mu_2 : 0.47$	20	$\mu_2 : 10$	1.67_{11}	1.90_{20}	$\mu_2 : 1.69$
0.6194	$\sigma_2^2 : 0.08$	14	$\sigma_2^2 : 36$	1.68_{12}	1.69_{14}	$\sigma_2^2 : 0.01$
0.3377		8		1.68_{13}	1.65_8	
0.0559		2		1.69_{14}	1.53_2	
0.7742		18		1.70_{15}	1.78_{18}	
0.4925		12		1.70_{16}	1.68_{12}	
0.2108	III	6	III	1.76_{17}	1.63_6	III
0.9291	$\mu_3 : 0.50$	21	$\mu_3 : 12$	1.78_{18}	1.92_{21}	$\mu_3 : 1.70$
0.6474	$\sigma_3^2 : 0.08$	15	$\sigma_3^2 : 36$	1.80_{19}	1.70_{15}	$\sigma_3^2 : 0.01$
0.3656		9		1.90_{20}	1.66_9	
0.0839		3		1.92_{21}	1.55_3	
SSW	1.6678		756		0.2168	
SSB	0.0109		14		0.0082	
ICC	-0.1591		-0.1455		-0.1241	

3   The BDSCV procedure

We propose a k-fold CV procedure that is built upon the best-discrepancy systematic subsampling procedure. The pseudocode of BDSCV is shown in Figure 4.

Suppose we are given an independently and identically distributed m dimensional dataset D of n instances, where $D \in \mathbb{R}^{n \times m}$. Each instance in D consists of a class label vector Y and vectors of measured feature covariates X. As discussed in the previous section, D is sorted before subsampling. However, it is difficult to decide along which axis to sort D. An arbitrary yet effective way is to transform D from multi-dimensional space to one-dimensional space using a dimensionality-reduction technique H, which produces a linear transformation $T \in \mathbb{R}^{m \times 1}$ and results in $E = DT$, where $E \in \mathbb{R}^{n \times 1}$ and E is the underlying structure of D and points to a direction where there is the most variance and where the elements in E are most spread out.

Simply stated, it is just like to project all dimensions of a dataset on the component vector E, which is a combination of all dimensions with weights. The instances with close or the same values in E are considered similar in the terms of all dimensions but not just of label classes. In addition, E can be served as an axis along which D is sorted so that instances that are similar to some extent are then gathered and separated from dissimilar ones. Therefore, we ascendingly sort E as $\widetilde{E}$ and then sort D as $\widetilde{D}$, so that $\widetilde{E} = \widetilde{D}T$. These $\widetilde{E}$ and T are then discarded and will not be included in subsequent analyses.

Next, we generate a BDS with n elements and then a ranking vector R accordingly. We partition $\widetilde{D}$ into k subsets, using the same allocation mechanism in (2.5).

Thirdly, the k subsets of $\widetilde{D}$ are used to conduct a standard k-fold CV procedure with a classifier A. Let F_k be the k-th fold that serves as the test set, and $\widetilde{D}_k$ be the training set obtained by removing

```
Read a dataset D of m dimensions and N instances
EPEs=[ ], VARs=[ ]
for H ← [PCA, FA, kPCA, tSVD] do:
        Use H to product a one-dimensional vector E from D
        E is attached to D as the last column
        D is ascendingly sorted along E as D̃
        E is deleted from D̃

        Generate a BDS X of N elements using Equation (3)
        Produce X's ranking vector R
        R is chunked to k subsets
        D̃ is partitioned into k subsets using Equation (4)

        Conduct k-fold CV with a classifier A and get k EPEs
    Obtain the mean of k EPEs using Equation (11)
    Save the mean EPE in EPEs
    Obtain the variance among k EPEs using Equation (12)
        Save the variance in VARs
Choose the smallest mean EPE from EPEs
Choose the corresponding variance from VARs
```

Figure 4   Pseudocode of the BDSCV procedure

the instances in F_k from $\widetilde{D}$. The estimated EPE over k folds with the dimensionality-reduction technique H is estimated as in [31]:

$$\mathrm{EPE}(\widetilde{D})_{H,A} = \frac{1}{K} \sum_{k=1}^{K} \widehat{\mathrm{EPE}(F_k)}.$$ (3.1)

In addition, the variance of the CV estimator is estimated as in [31]:

$$\sigma_{H,A}^2 = \frac{1}{K} \sum_{k=1}^{K} (\mathrm{EPE}(F_k)_{H,A} - \mathrm{EPE}(\widetilde{D})_{H,A})^2.$$ (3.2)

Finally, different dimensionality-reduction techniques may lose different amounts of information from the complete dataset [10]. Sweeping different candidate techniques, such as principal component analysis (PCA), factor analysis (FA), kernel PCA (kPCA), or truncated singular value decomposition (tSVD), will enable us to fine tune the BDSCV procedure, so that the original space of D can be reduced to one-dimensional space while retaining the most important information as much as possible in E. Therefore, previous steps are repeated using different dimensionality-reduction techniques in order to find the best H in terms of $\mathrm{EPE}(\widetilde{D})_{H,A}$ and its corresponding $\sigma_{H,A}^2$ as the final CV results for the classifier A.

4   Experiments

We first test the logistic regression model in Subsection 2.2 with the proposed 10-fold BDSCV. The model achieves zero error rate in every fold, which suggests that the BDSCV can eliminate subsampling bias to a great extent. We then test the BDSCV with the Penn machine learning benchmark (PMLB) datasets [33]. Note that we eliminate 10 PMLB datasets that are too big to be computed on a personal computer. The characteristics of these 156 datasets are summarized in Table 2. We compare the performance of our proposed BDSCV to MCCV, stratified MCCV and LOO.

The MCCV and stratified MCCV procedures are conducted using the *scikit-learn* cross-validation module (version 0.19.2) with 50 repetitions for the three classifiers respectively. Our proposed BDSCV is mainly built on the same *scikit-learn* module except the subsampling part. We use the MCCV as the benchmark and compute the following three comparative performance metrics as:

(a) the comparative EPE ratio of a classifier A, which is

$$\varphi\mathrm{EPE}_A = \frac{\mathrm{EPE}_{\mathrm{BDSCV/Stratified/LOO}_A}}{\mathrm{EPE}_{\mathrm{MCCV}_A}};$$ (4.1)

Table 2   Descriptive statistics of 156 PMLB datasets

	#Instances	#Features	#Labels	Aspect Ratio
Minimum	32.00	2.00	2.00	0.00
Maximum	28056.00	240.00	26.00	0.54
Mean	2056.34	24.77	3.83	0.05
Median	2000.00	9.00	2.00	0.01

Table 3   Number of datasets that each CV reaches the lowest EPEs and variances

Classifier	BDSCV		MCCV		Stratified MCCV		LOO	
	EPE	Var	EPE	Var	EPE	Var	EPE	Var
L[1]	118	101	8	15	14	43	23	2
DT[2]	115	102	8	21	12	43	43	12
NB[3]	125	94	4	8	8	55	22	2

[1] L: Logistic
[2] DT: Decision Tree
[3] NB: Naïve Bayes

(b) the comparative variance ratio of a classifier A, which is

$$\varphi\sigma_A^2 = \frac{\sigma^2_{\text{BDSCV/Stratified/LOO}_A}}{\sigma^2_{\text{MCCV}_A}}; \tag{4.2}$$

and (c) the comparative computational burden ratio of the k-fold CV procedures, which is

$$\varphi\text{Time} = \frac{\text{Time}_{\text{BDSCV/Stratified/LOO}}}{\text{Time}_{\text{MCCV}}}. \tag{4.3}$$

Our experiments are conducted on a desktop personal computer (CPU: Intel Core i7-7700 eight cores, Memory: 64G, and SSD: 256GB). Note that computing time may slightly vary depending on the overall load and specifications of a computer. Table 3 shows that the number of datasets in which each of the four CV procedures achieve the same lowest EPEs and variances (it is possible that multiple CV procedures obtain the lowest performance metrices in a dataset). As expected, the BDSCV achieves the lowest EPEs and variances in most datasets, followed by the LOO and the stratified MCCV. Table 3 suggests that there are significant subsampling biases among different CV procedures and if we count on the most widely used 10-fold MCCV to do model selections, then we probably make the wrong choices.

We summarize the three comparative performance metrics in Table 4. The EPEs, variances and computational time of each classifier and of each dataset can be found in supplementary material.

The comparisons between the BDSCV and the MCCV are as follows (see Panel A in Table 4). On average, the $\text{EPE}_{\text{Logistic}}$ and $\sigma^2_{\text{Logistic}}$ of the former is 94.04% and 70.30% of that of the latter respectively; the $\text{EPE}_{\text{Decision Tree}}$ and $\sigma^2_{\text{Decision Tree}}$ of the BDSCV is 88.02% and 77.50% of that of the MCCV respectively; and the $\text{EPE}_{\text{Naïve Bayes}}$ and $\sigma^2_{\text{Naïve Bayes}}$ of the former is 96.41% and 72.00% of that of the latter respectively. On average, the BDSCV can lower the EPEs around 7.18% and the variances around 26.73% regardless classifiers. In comparison, the stratified MCCV can slightly reduce subsampling bias by decreasing the EPEs of the MCCV around 1.58% and the variances around 11.85%. The LOO can lower the EPEs around 2.50% but its variances are much higher than the any other CV procedure (see Panels B and C in Table 4). The computational time of the BDSCV is just 8.64% of the MCCV, 8.67% of the stratified MCCV and 16.72% of the LOO (see Panel D in Table 4). These results suggest that the proposed BDSCV outperforms the other CV procedures as it can extrude significant subsampling bias while using a small fraction of computational time of the other CV procedures. Being computationally efficient is especially important where it is expensive or impractical to train multiple models.

Finally, our proposed CV procedure may not be a cure-all and be more pertinent to a dataset with particular characteristics. We compute the aspect ratio (i.e., the number of features divided by the

Table 4   Comparisons between different CV procedures

A: BDSCV vs. MCCV					
Aspect ratio	EPE_L	σ_L^2	EPE_DT	σ_DT^2	EPE_NB	σ_NB^2
Q1	97.95%	64.29%	90.51%	87.88%	98.61%	75.11%
Q2	94.45%	67.06%	90.25%	64.22%	96.57%	57.31%
Q3	91.37%	66.80%	82.92%	78.72%	95.89%	72.47%
Q4	92.36%	83.06%	88.41%	79.17%	94.59%	83.11%
Average	94.04%	70.30%	88.02%	77.50%	96.41%	72.00%
B: Stratified MCCV vs. MCCV					
Aspect ratio	EPE_L	σ_L^2	EPE_DT	σ_DT^2	EPE_NB	σ_NB^2
Q1	99.48%	72.02%	98.91%	91.23%	99.59%	80.99%
Q2	97.96%	83.79%	99.53%	93.58%	99.91%	89.41%
Q3	97.25%	90.41%	94.35%	93.08%	98.79%	85.73%
Q4	98.40%	94.25%	97.62%	93.73%	99.25%	85.59%
Average	98.28%	85.12%	97.60%	92.90%	99.39%	86.43%
C: LOO vs. MCCV					
Aspect ratio	EPE_L	σ_L^2	EPE_DT	σ_DT^2	EPE_NB	σ_NB^2
Q1	99.72%	44036.96%	93.16%	36397.24%	100.01%	37667.87%
Q2	100.20%	7897.06%	92.52%	7693.12%	100.36%	7000.07%
Q3	97.43%	4977.84%	91.25%	4084.19%	101.24%	6016.48%
Q4	98.25%	4387.51%	98.45%	4411.73%	97.44%	3858.06%
Average	98.90%	15324.84%	93.85%	13146.57%	99.76%	13635.62%
D: Computational time comparisons					
Aspect ratio	$\frac{\text{BDSCV}}{\text{MCCV}}$	$\frac{\text{Stratified}}{\text{MCCV}}$	$\frac{\text{LOO}}{\text{MCCV}}$	$\frac{\text{BDSCV}}{\text{Stratified}}$	$\frac{\text{LOO}}{\text{Stratified}}$	$\frac{\text{BDSCV}}{\text{LOO}}$
Q1	9.15%	99.62%	886.01%	9.19%	889.44%	3.08%
Q2	8.48%	99.68%	152.33%	8.51%	153.16%	9.68%
Q3	8.52%	99.79%	126.15%	8.54%	126.61%	20.60%
Q4	8.42%	99.80%	90.95%	8.43%	91.23%	33.53%
Average	8.64%	99.72%	313.86%	8.67%	315.11%	16.72%

L: Logistic
DT: Decision Tree
NB: Naïve Bayes

number of instances) of the 156 datasets and divide them into four groups based on the quantiles of the aspect ratio. The first group includes 67 datasets whose number of features is smaller or equal to 1% of number of instances. The second group includes 20 datasets whose percentages of number of features to number of instances are in the range of $(1\%, 2\%]$. The third group includes 33 datasets whose aspect ratios are in the range of $(2\%, 5\%]$. In addition, the fourth group includes 36 datasets whose aspect ratios are larger than 5%. Table 4 shows that the BDSCV can significantly reduce subsampling bias in the third and fourth groups of datasets, as the EPEs of the BDSCV are significantly smaller. These findings imply that the BDSCV could be particularly effective in analysing in bioscience datasets, which usually are characterized as high dimensionality and relatively small sample sizes [12, 15].

In short, these experiments confirm that the MCCV suffers a lot from subsampling bias. The BDSCV can reduce subsampling bias in the EPEs and variances of the three classifiers to a significant extent. The improvements of the stratified MCCV are not as substantial as the BDSCV. The LOO does not have significant advantages compared with the other CV procedures. Both the stratified MCCV and the LOO are not as computationally efficient as the BDSCV. In addition, the decision tree is the most sensitive classifier to subsampling bias in terms of EPEs, especially in the datasets with high aspect ratios (i.e., the 3rd and 4th groups) in which the BDSCV can deflate the $\text{EPE}_\text{Decision Tree}$ around 15% on average. The logistic regression is the most sensitive classifier to subsampling bias in terms of variances, especially in the datasets with relatively low aspect ratio (i.e., the 1st, 2nd and 3rd groups) in which the BDSCV

can reduce variances around 34%.

5　Discussion and closing remarks

A statistical machine learning model could be affected by the idiosyncrasies of training data if the discrepancy between k subsets is too high. As a result, the EPEs and variances of an MCCV procedure are contaminated by subsampling bias and cannot reflect the true generalization performance of a model's inductive mechanism. The inflated metrics will interfere with data scientists making right decisions. Our experimental results confirm that subsampling bias in the MCCV procedure greatly jeopardizes its validity and reliability for model selection even after 50 repetitions. Our experiments also confirm that the subsampling bias can be reduced through Lohr's guidelines [29]—systematic subsampling from an ordered population with a skip. The BDSCV follows these guidelines so that it is relatively more precise, stable and computationally efficient than all the other three CV procedures.

It is important to point out that the BDSCV is not meant to completely replace other CV procedures. As the results of our experiments suggest that the advantages of the different types of CV may depend on dataset characteristics and decision boundary. We recommend using the BDSCV in the datasets characterized with high aspect ratios (i.e., small dataset size and high dimensionality) and for the non-parametric classifier such as the decision tree. Besides, the BDSCV is much more computational efficient than the other CV procedures so it is suitable for the time-critical statistical machine learning problems. Finally, it is beneficial to estimate the true generalization performance of a learning function using different CV procedures in order to choose the model with the lowest EPEs and variances. Further studies are also necessary to determinate what is the most appropriate CV procedure for a particular dataset and for a statistical machine learning model.

In conclusion, this study sheds more light on the literature of CV and statistical machine learning model evaluation by proposing the BDSCV procedure, which is a new way to estimate a model's relatively objective generalizability by significantly reducing subsampling bias in existing CV procedures. It also helps the model to pick out the hidden underlying mapping between input-output vectors and to remain stable when the training set changes, leading to more precise estimations and more confident inferences. What is more, our proposed procedure is straightforward and computationally efficient. Unlike other information-preserving subsampling techniques [18] that need to understand the complete dataset in great detail, our procedure does not make the CV computation more complex. Lastly, our subsampling method may replace the random mechanism used in many statistical machine learning algorithms.

References

1　Baker A. On some Diophantine inequalities involving the exponential function. Canad J Math, 1965, 17: 616–626

2　Bergstra J, Bardenet R, Bengio Y, et al. Algorithms for hyper-parameter optimization. Adv Neural Inf Process Syst, 2011, 1: 2546–2554

3　Bergstra J, Bengio Y. Random search for hyper-parameter optimization. J Mach Learn Res, 2012, 13: 281–305

4　Boyle P, Broadie M, Glasserman P. Monte Carlo methods for security pricing. J Econom Dynam Control, 1997, 21: 1267–1321

5　Braga-Neto U M, Dougherty E R. Is cross-validation valid for small-sample microarray classification? Bioinformatics, 2004, 20: 374–380

6　Braga-Neto U M, Zollanvari A, Dougherty G. Cross-validation under separate sampling: Strong bias and how to correct it. Bioinformatics, 2014, 30: 3349–3355

7　Branicky M, LaValle S, Olson K, et al. Quasi-randomized path planning. In: IEEE International Conference on Robotics and Automation, vol. 2. Piscataway: IEEE, 2001, 1481–1487

8   Cheng J. Computational investigation of low-discrepancy sequences in simulation algorithms for Bayesian networks. In: Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence, vol. 1. San Francisco: Morgan Kaufmann Publishers, 2000, 72–81

9   Chung K. An estimate concerning the Kolmogoroff limit distribution. Trans Amer Math Soc, 1949, 67: 36–50

10   Cunningham J, Ghahramani Z. Linear dimensionality reduction: Survey, insights, and generalizations. J Mach Learn Res, 2015, 16: 2859–2900

11   Dai H, Wang W. Application of low-discrepancy sampling method in structural reliability analysis. Struct Safety, 2009, 31: 155–164

12   Díaz-Uriarte R, DeAndrés S. Gene selection and classification of microarray data using random forest. BMC Bioinformatics, 2006, 7: 1–13

13   Dick J, Kuo F, Sloan I. High-dimensional integration: The quasi-Monte Carlo way. Acta Numer, 2013, 22: 133–288

14   Dick J, Pillichshammer F. The weighted star discrepancy of Korobov sets. Proc Amer Math Soc, 2015, 143: 5043–5057

15   Fu W, Carroll R, Wang S. Estimating misclassification error with small samples via bootstrap cross-validation. Bioinformatics, 2005, 21: 1979–1986

16   Gentle J. Statistics and Computing Random Number Generation and Monte Carlo Methods. New York: Springer, 2003

17   Georgieva A, Jordanov I. A hybrid meta-heuristic for global optimisation using low-discrepancy sequences of points. Comput Oper Res, 2010, 37: 456–469

18   Groot P De, Postma G, Melssen W, et al. Selecting a representative training set for the classification of demolition waste using remote NIR sensing. Anal Chimica Acta, 1999, 392: 67–75

19   Halton J H. Algorithm 247: Radical-inverse quasi-random point sequence. Comm ACM, 1964, 7: 701–702

20   Hua L K, Wang Y. Applications of Number Theory in Approximate Analysis. Beijing: Science Press, 1978

21   Kalagnanam J, Diwekar U. An efficient sampling technique for off-line quality control. Technometrics, 1997, 39: 308–319

22   Keller A. The fast calculation of form factors using low discrepancy sequences. In: Proceedings of the 12th Spring Conference on Computer Graphics, vol. 1. Bratislava: Comenius University Press, 1996, 195–204

23   Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of International Joint Conferences on Artificial Intelligence, vol. 14. San Francisco: Morgan Kaufmann Publishers, 1995, 1137–1143

24   Kollig T, Keller A. Efficient multidimensional sampling. Comput Graph Forum, 2002, 21: 557–563

25   Kucherenko S, Sytsko Y. Application of deterministic and low-discrepancy sequence in global optimisation. Comput Optim Appl, 2005, 30: 297–318

26   Kuipers L, Niederreiter H. Uniform Distribution of Sequences. New York: John Wiley & Sons, 1974

27   Li X, Wang W, Martin R, et al. Using low-discrepancy sequences and the crofton formula to compute surface areas of geometric models. Comput Aided Design, 2003, 35: 771–782

28   Lindermann R, Steven S, LaValle M. Incremental low-discrepancy lattice methods for motion planning. In: Proceedings of IEEE International Conference on Robotics and Automation, vol. 1. Piscataway: IEEE, 2003, 2920–2927

29   Lohr S. Sampling: Design and Analysis. Boston: Brooks/Cole, 2009

30   Mahler K. On a paper by A. Baker on the approximation of rational powers of *e*. Acta Arith, 1975, 27: 61–87

31   Molinaro A, Simon R, Pfeiffer R. Prediction error estimation: A comparison of resampling methods. Bioinformatics, 2005, 21: 307–330

32   Niederreiter H. Random Number Generation and Quasi-Monte Carlo Methods. Philadelphia: SIAM, 1992

33   Olson R, LaCava W, Orzechowski P, et al. PMLB: A Large benchmark suite for machine learning evaluation and comparison. BioData Mining, 2017, 10: 36

34   Pant M, Thangaraj R, Grosan C, et al. Improved particle swarm optimization with low-discrepancy sequences. In: Proceedings of the IEEE Congress on Evolutionary Computing, vol. 2. Piscataway: IEEE, 2008, 3011–3018

35   Paskov S, Traub J. Faster valuation of financial derivatives. J Portfolio Management, 1995, 22: 113–123

36   Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine learning in python. J Mach Learn Res, 2011, 12: 2825–2830

37   Quinn J, Langbein F, Martin R. Low-discrepancy sampling of meshes for rendering. Eurographics Symp Point-Based Graph, 2007, 1: 19–28

38   Schmidt W. Irregularities of distribution, VII. Acta Arith, 1972, 21: 45–50

39   Singhee A, Rutenbar R. From finance to flip flops: A study of fast quasi-Monte Carlo methods from computational finance applied to statistical circuit analysis. In: Proceedings of the 8th International Symposium on Quality Electronic Design, vol. 1. Washington: IEEE, 2007, 685–692

40   Stone M. Cross-validatory choice and assessment of statistical predictions. J R Stat Soc Ser B Stat Methodol, 1974, 36: 111–147

41   Struckmeier J. Fast generation of low-discrepancy sequences. J Comput Appl Math, 1995, 61: 29–41

42   Tan K, Boyle P. Applications of randomized low discrepancy sequences to the valuation of complex securities. J Econom Dynam Control, 2000, 24: 1747–1782

43   Uy N, Hoai N, McKay R, et al. Initialising PSO with randomised low-discrepancy sequences: The comparative results. In: Proceedings of the IEEE Congress on Evolutionary Computing, vol. 1. Piscataway: IEEE, 2007, 1985–1992

44   van der Corput J G. Verteilungsfunktionen (Erste Mitteilung). In: Proceedings of the Koninklijke Akademie van Wetenschappen te Amsterdam, vol. 38. Amsterdam: Elsevier, 1935, 813–821

45   Wenzel L, Dair D, Vazquez N. Pattern Matching System and Method with Improved Template Image Sampling Using Low Discrepancy Sequence. Washington: US Patent No. 6,229,921, 2001

46   Xu Z Q, Zhou T. On sparse interpolation and the design of deterministic interpolation points. SIAM J Sci Comput, 2014, 36: 1752–1769