IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Revue d'Intelligence Artificielle
Vol. 37, No. 6, December, 2023, pp. 1415-1421
Journal homepage: http://iieta.org/journals/ria

An Approximate Maximin-Directed Random Sampling for Clustering Applications

Khamees Khalaf Hasan[1], Omar A. Ibrahim[2,3*], Mahmood Ali A. Dham[1]

[1] Electrical Engineering Department Tikrit University, ALShirqat College of Engineering, Salahadin P.O. Box (42), Iraq
[2] Electrical Engineering Department, Tikrit University, College of Engineering, Salahadin, P.O. Box (42), Iraq
[3] Electrical Engineering Department, Tikrit University, ALShirqat College of Basic Education, Salahadin P.O. Box (42), Iraq

Corresponding Author Email: omar.a.ibrahim@tu.edu.iq

https://doi.org/10.18280/ria.370605

Received: 19 September 2023
Revised: 7 October 2023
Accepted: 2 November 2023
Available online: 27 December 2023

Keywords:
maximin-directed random sampling, approximate maximin-directed random sampling, Dunn's index, cluster analysis, big data

ABSTRACT

The Maximin-Directed Random Sampling (MMDRS) algorithm, a cornerstone of numerous visual assessment techniques and scalable single linkage clustering, is recognized for its unique three-part structure: (i) Maximin (MM) sampling for prototype identification; (ii) nearest prototype partition construction via maximin samples; and (iii) directed random sampling from partition subsets. Despite its diverse applications, the computational complexity of MMDRS presents significant challenges. In response to this issue, an approximate form of the MMDRS algorithm (AMMDRS) is proposed in this study, aiming to alleviate time complexity. Through experimental investigation, comparisons are drawn between the directed random sampling methods, assessing whether significant differences exist in the samples produced and evaluating the superiority of either method over simple random sampling. The results of this empirical study demonstrate that AMMDRS outperforms MMDRS in terms of speed across all datasets, without any compromise on sampling accuracy. This finding underscores the critical importance of such a method in big data applications, where the feasibility of processing the entire dataset is often limited. The study's revelations emphasize that undirected random sampling achieves more authentic representations of parent distributions than MM samples alone, thereby maximizing the diversity and representativeness of selected points within the feature space. Overall, this study introduces a promising avenue for enhancing the efficiency of MMDRS, opening the door to its broader application in data-intensive domains.

1. INTRODUCTION

Social networking giants like Facebook and Twitter boast billions of users, generating hundreds of gigabytes of content every minute. Retail establishments continuously amass extensive customer data, while platforms like YouTube, with over 1 billion unique users, churn out 100 hours of video content every hour. To illustrate the sheer magnitude, YouTube's content ID service scans an astounding 400 years' worth of video content each day [1, 2]. Notably, scientists and researchers refer to it as "Big Data". In the face of this deluge of data, the need for robust tools for knowledge discovery becomes imperative. Data mining techniques have firmly established themselves as indispensable instruments for this purpose. Among these techniques, clustering stands out as a method whereby data is partitioned into groups, ensuring that objects within each group share more similarity with one another than with objects in other groups [1].

Suppose n objects are represented as feature vectors $X_N = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subset \Re^p$. Classic cluster analysis for this kind of static data is discussed in many texts and numerous articles [3-11]. If the number of samples precludes clustering the data directly, there are two popular ways to approach the problem. First, we may split the data into chunks, process the chunks independently, and aggregate the results [12, 13].

A second popular approach is to sample the data, cluster the sample, and then extend the results to the rest of the data set non-iteratively by labeling the remaining points with the nearest prototype method [14]. The question addressed in this paper is: what method of sampling produces the "best" samples to use in this context? Certainly (true) Random Sampling (RS) is the best-known method. Progressive sampling using various termination criteria is advocated in [15-17]. The specification of the MMDRS algorithm requires a bit of notation.

Assume c is an integer number such that 1<*c*<*N*. The set $M_{hcN} = \{U \in \Re^{cN}:\ 0 \le u_{ik} \le 1\ \forall i,k;\ \sum_i u_{ik} = 1\ \forall\ k; \sum_k u_{ik} > 0\ \forall\ i\}$ contains all of the crisp c-partitions of N objects, represented as $c \times N$ matrices. Equivalently, each U (membership) can be represented as $X_N = \cup_{i=1}^{c} X_i\,;\ X_i \cap X_j = \emptyset\ \forall\ i \neq j$, where $\{X_i\}$ are the crisp subsets comprising the c clusters. We write $U \leftrightarrow \{X_i\}$. The MMDRS partition of X_N is $U_{MM} \in M_{hc'N}$ where *c*' is the desired number of smaples to be selected by maximin sampling (MM).

A third approach for sampling is based on a three step process comprising: (i) determination of *c*' Maximin (MM) prototypes $X_{MM} = \{x_{m_1}, \ldots, x_{m_{c'}}\} \subset X_N$; (ii) erection of the nearest prototype partition U_{MM} of X_N; and (iii) drawing a specified number of samples from each of the subsets in U_{MM}. This third method is not true random sampling; rather, it is random sampling constrained by drawing samples from specified locations. Since this RS scheme is directed by the MM samples, we will call it the Maximin-Directed Random Sampling (MMDRS) method which was first discussed in the study [18]. Since then, this method or some derivative of it have been used frequently in the literature of cluster analysis for static data. One of the challenges with MMDRS is that it is computationally expensive. Therefore, to enhance this aspect of MMDRS, we introduce a new approximate MMDRS (AMMDRS) sampling scheme. The goal of AMMDRS is to be faster and more applicable for big data applications.

So, this article has the following contributions. First, we will introduce the new AMMDRS scheme. Then we will conduct some numerical experiments to compare the quality of samples produced by the three sampling methods: RS, MMDRS, and AMMDRS. Ultimately, we will demonstrate that adopting our approach yields sample quality comparable to MMDRS, all while requiring less computational complexity. The remainder of the paper is organized as follows. In Section 2, we dive into the MM and MMDRS algorithms. We then flesh out the new AMM scheme in Section 3, building on the foundations of the original MMDRS method. In Section 4, we tackle the nuanced idea of what "best" sample really means in the context of cluster analysis. Section 5 sheds light on the datasets used in the analysis and the metrics that gauge their quality. The details of our findings are in Section 6, and we wrap things up with our takeaways in Section 7.

2. THE MM AND MMDRS ALGORITHMS

The concept of MM sampling was initially introduced in the study [19], where it is characterized as a method for initializing a set of c prototypes, also known as cluster centers, for clustering purposes. Casey and Nagy [20] conducted an overview of the MM algorithm for setting up initial prototypes, which we refer to as the MM principle.

[MM Principle]. The initial sample in the batch serves as our first cluster center. From there, we calculate the distances of the other samples from this initial center. The sample farthest away becomes our second center. For every other sample, we consider the shorter of the two distances from these centers. The sample with the largest of these minimum distances is then selected next. Subsequent centers are selected to ensure maximum separation from those already chosen. This ensures that our initial cluster centers are spread widely across the sample space—a property that's intuitively appealing.

Hathaway et al. [18] appended two steps to this sampling scheme. First, the crisp nearest prototype rule (NPR) partition is computed using the MM samples as prototypes. Second, each of the subsets in this partition is subsequently sampled randomly a number of times proportional to the number of points in the subset. This produces a small subset of the larger parent set for approximate clustering and tendency assessment. The resultant sample is called a Maximin Directed Random Sample (MMDRS). The complete pseudo code for the MMDRS algorithm is depicted in Table 1 below where it is split into two sections, one is the MM sampling and the other one is the DRS sampling.

Lines 1-9 extract the *c*' MM samples from X_N. Ties in Line 6 are broken arbitrarily. Lines 10-19 build the elements of the crisp partition $U_{MM} \in M_{hc'N}$ of X_N. The matrix U_{MM} appearing in lines 10, 12 and 20 is commented out since it is not needed to secure the desired MMDRS samples outputted in line 20. We show it to instruct readers on how the partition is used to direct the random sampling. Hopefully this lends some transparency to the DRS scheme. You may recognize U_{MM} as the "k-means" or nearest prototype rule (NPR) partition of X_N built by applying Lloyd's algorithm [1] to the input data with *k*=*c*' using the *c*' MM samples as cluster centers.

Table 1. The algorithm of MMDRS sampling

1	**In: metric $d: \Re^p \times \Re^p \mapsto \Re^+ : X_N = \{x_1, \ldots, x_N\} \subset \Re^p :$ *c*'= desired # of MM samples:** ***n***n_s**=desired number of MMDRS samples**	**MM**
2	**Initialize: $X_{MM} = \varnothing$:**	
3	$x_{m_0} = x_1$:	
4	$Z = (z_{1,\ldots,}z_N) = (d(x_{m_o}, 1), \ldots, d(x_{m_o}, N))$:	
5	**For t ⟵1 to *c*' do**	
6	$Z = (\min\{z_1, d(x_{m_{t-1}}, x_1)\}, \ldots, \min\{z_N, d(x_{m_{t-1}}, x_N)\})$:	
7	$m_t = \underbrace{\mathrm{argmax}}_{1 \le j \le N}\{z_j\}$:	
8	$X_{MM} = X_{MM} \cup \{x_{m_t}\}$:	
9	**End for**	
10	**% Begin DRS: Initialize: S_1=S_2=…$S_{c'}$ = M_{n_s} =∅: % U_{MM}=[0]:**	**DRS**
11	**For t←1 to N do**	
12	$q = \underbrace{\mathrm{argmin}}_{1 \le j \le c'}\left\{d(x_{m_j}, x_t)\right\}$: $\%U_{MM}(q, t) = 1$	
13	$S_q = S_q \cup \{t\}$:$U_{MM}(q, t) = 1$	
14	**End for: % The sets**	
15	**For t ⟵1 to c' do**	
16	$n_t = \left\lceil n_s \left(\frac{\lvert S_t \rvert}{N}\right) \right\rceil$	
17	**Draw n_t unique indices $\{m_t\}$ from S_t**	
18	$M_{n_s} = \cup_{t=1}^{c'}\{m_t\}$	
19	**End for**	
20	**Out: n_s MMDRS indices $M_{n_s} = \{m_1, \ldots, m_{n_s}\}$: n_s MMDRS samples X_{MMDRS}:** **% MMDRS partition $U_{MM} \in M_{hc'N}$**	

The literature contains at least six ways to initialize MM sampling in Line 3. A recent study of this issue [21] determined that, on average, the original and fastest scheme (line 3) is as reliable as the other five methods, so that is the initialization we use. The primary requirement for good samples in the present context is that the cluster proportions in the *c*' samples from X_N be representative of the corresponding proportions for the subsets in X_N. If the data are unlabeled, there is no way to ascertain whether any sampling scheme satisfies this desire. But if the data are labeled, we can determine how well the samples match the distribution of the labeled subsets in X_N. This intuitive objective informs our definition for what constitutes a best set of samples. Our expectation is that the DRS methods which begin with MM sampling will produce better samples of labeled data than simple RS in terms of matching proportions of sample and parent (in this article we call X_N the parent of samples of it made by the three methods). There are three minor results about MMDRS sampling that provide weak guarantees that fuel our expectations. To describe the results, we need Dunn's index [22], discussed next.

Consider two non-empty subsets, S and T$\in \Re^p$, with an arbitrary metric denoted by $d: \Re^p \times \Re^p \mapsto \Re^+$. The diameter of S can be defined as S as $\Delta(\mathrm{S}) = \underbrace{\max}_{\mathbf{x},\mathbf{y}\in \mathrm{S}}\{\mathrm{d}(\mathbf{x},\mathbf{y})\}$. Similarly, we define the set distance δ between S and T as $\delta(\mathrm{S},\mathrm{T}) = \underbrace{\min}_{\substack{\mathbf{x}\in \mathrm{S}\\ \mathbf{y}\in \mathrm{T}}}\{\mathrm{d}(\mathbf{x},\mathbf{y})\}$. For any given partition $\mathrm{U} \in \mathrm{M}_{\mathrm{hcN}} \leftrightarrow \{\mathrm{X}_\mathrm{i}\}$, the separation index of *U*, widely recognized as Dunn's index (DI, [22]) is:

$$\mathrm{DI(U;X)} = \min_{1\le i\le c}\left\{ \min_{\substack{1\le j\le c\\ j\ne i}}\left\{ \frac{\delta(\mathrm{X_i},\mathrm{X_j})}{\max\limits_{1\le k\le c}\{\Delta(\mathrm{X_k})\}} \right\}\right\} \qquad (1)$$

Dunn characterized set U as compact and separated (CS) in relation to metric d under the following conditions: For all subsets *s*, *q*, and *r*, where *q*≠*r*, any pair of points x and y from X_S are closer to each other (based on metric d) than any other pair *u* and *v*, where u is from X_q and *v* is from X_r. Dunn established that a set *X* possesses a clear CS partition with respect to *d* if and only if $\underbrace{\max}_{\mathrm{U}\in \mathrm{M_{hcn}}} \{\mathrm{DI(U;X)}\} > 1$, the maximum of DI(U;X) over all *U* in M_{hcN} is greater than 1. Subsequent results tie this particular characteristic of Dunn's index to the MMDRS samples extracted obtained from X_N by Algorithm 1:

Proposition MM. Let *c*'≥*c*. Suppose there is a CS c-partition of $\mathrm{X_N} = \{\mathbf{x}_1,\ldots,\mathbf{x}_N\} \subset \Re^\mathrm{p}$. Then lines 1-9 of the MMDRS Algorithm will select at least one object from each of the c clusters.

Proof. Proposition 1, Hathaway et al. [18].

The MM theorem tells us that when the input da have c CS clusters, lines 1-9 of Algorithm 1 will extract at least one sample from each cluster. Please observe that proposition MM applies to the seeds (the prototypes) which are used to build the MMDRS partition.

Proposition MMDRS-1. Let $\mathrm{X_N} = \{\mathbf{x}_1,\ldots,\mathbf{x}_N\} \subset \Re^\mathrm{p}$. Let metric $\mathrm{d}: \Re^\mathrm{p} \times \Re^\mathrm{p} \mapsto \Re^+$. If X_N can be partitioned into c compact and separated clusters CS clusters, and *c*'=*c*, then $\mathrm{U_{MM}} \in \mathrm{M_{hcN}}$ is the CS partition of X_N.
Proof. Theorem 1, Hathaway et al. [23].

Proposition MMDRS-1 tells that when the input data have *c* CS clusters and we choose *c*'=*c*, that lines 10-19 of Algorithm 1 find the CS clusters. The number of samples drawn from the t-th subset in Line 16 of the MMDRS algorithm is $n_t = \left\lceil n_s\left(\frac{|S_t|}{N}\right)\right\rceil; 1 \le t \le c'$. The number $|S_t|/N$ scales the number of desired samples n_s drawn from the t-th row of U_{MM} by the proportion of samples in that row. Because of the ceiling function, the overall number of samples is approximate, $\sum_{t=1}^{c'} n_t \approx n_s$. The number and the proportions drawn will be exact under the extra condition that the sampling proportions are all integers, so the ceiling function is not used and $\sum_{t=1}^{c'} n_t = n_s$.

Proposition MMDRS-2. Let $\mathrm{X_N} = \{\mathbf{x}_1,\ldots,\mathbf{x}_N\} \subset \Re^\mathrm{p}$. Let metric $\mathrm{d}: \Re^\mathrm{p} \times \Re^\mathrm{p} \mapsto \Re^+$. Suppose X_N can be partitioned into c CS clusters for *c*'≥*c*, and suppose that $|S_t|/N$ is an integer for all t. Then the proportion of objects in the MMDRS sample from subset *t* equals the proportion of objects in the parent population for *t*=1 to c.

Proof. Proposition 2, Hathaway et al. [18].

These three results have limited utility because the majority of input datasets lack the CS property, and even when they do possess it, it is usually impossible to verify that this is the case. On the other hand, these results do provide some reassurance about the MMDRS procedure, in the sense that at least in some cases, Algorithm 1 obtains samples that do represent all c clusters in the data. Consequently, we expect the MMDRS samples to provide fairly representative proportions of the distribution of the input data.

As a final note, we remark that the actual MM samples drawn by MM lines 1-9 are not part of the sample output, but can easily be included in the output if this is desired. Our experience is that inclusion of the MM samples doesn't make much difference to their quality in terms of representing the distribution of the input data.

In summary, MMDRS demonstrates its effectiveness in generating representative samples from a dataset X_N when the cluster proportions in the *c*' samples derived from X_N align closely with the proportions found within the subsets of X_N. The generated samples can be used as input to any clustering algorithm to find structure in the data without the need to iteratively accessing the whole data samples. Thus, making it feasible to run most clustering algorithms for very large datasets which is impossible without sampling. However, one drawback of MMDRS is that it needs to span all the data which makes it challenging and time consuming for large datasets. Therefore, reducing the time complexity for this approach will be essential for big data applications.

3. APPROXIMATE MMDRS

Table 2. Approximate MM (AMM) sampling

1	In: metric $\boldsymbol{d}: \Re^p \times \Re^p \mapsto \Re^+$:$\boldsymbol{X_N} = \{\boldsymbol{x_1},\ldots,\boldsymbol{x_N}\} \subset \Re^p$: *c*'=desired # of MM samples: n_s=desired number of MMDRS samples, *T*: number of subsets in the data split
2	Initialize: $\boldsymbol{X_{MM}} = \emptyset$
3	$\boldsymbol{x_{m_0}} = \boldsymbol{x_1}$:
4	$\boldsymbol{Z} = (\boldsymbol{z_{1,\ldots,}z_N}) = (\boldsymbol{d}(\boldsymbol{x_{m_o}},\mathbf{1}),\ldots,\boldsymbol{d}(\boldsymbol{x_{m_o}},\boldsymbol{N}))$:
5	For t←1 to c' do
6	$X_N = X_N(rand(N))$#Data shuffle: $X_W = X_N(1{:}\frac{N}{T})$ Random partition of the data:
7	$\boldsymbol{Z} = (\mathbf{min}\{\boldsymbol{d}(\boldsymbol{x_{m_0},x_1}),\ldots,\boldsymbol{d}(\boldsymbol{x_{m_{t-1}},x_1})\},\ldots,$ $\boldsymbol{min}\{\boldsymbol{d}(\boldsymbol{x_{m_0},x_W}),\ldots,\boldsymbol{d}(\boldsymbol{x_{m_{t-1}},x_W})\})$:
8	$\mathrm{m_t} = \underbrace{\mathrm{argmax}}_{1\le j\le W}:\{\mathrm{z_j}\}$:
9	$\boldsymbol{X_{AMM}} = \boldsymbol{X_{AMM}} \cup \{\boldsymbol{x_{m_t}}\}$:
10	End for

Next, we turn to approximate MM (AMM) sampling. Several approximate MM schemes which don't use DRS have appeared [24, 25], but since they don't use directed random sampling as a second step, these methods will not be considered here. Table 2 describes our approximate version of MM sampling:

Lines 1-10 extract the *c*' AMM samples from X_N. The first AMM sample, selected in Line 3 of Algorithm 2, is the first sample in the data. For each additional MM sample, the data is shuffled and split into T chunks. Each successive MM sample is chosen from the new chunk (X_w) instead of the whole input data set (X_N). This process is repeated until *c*' samples are obtained. The DRS procedure (lines10-20 of Algorithm 1) is then used to find n_s AMMDRS samples. To summarize, the AMM procedure simply replaces the input data set X_N by a chunk X_w at each iteration in the MM part of the MMDRS algorithm. This reduces the computation time for the MM part of the sampling procedure. Now we turn to some ways to measure sampling quality, where the samples are explicitly constructed to support cluster analysis.

It is evident that AMDRS leverages its primary advantages in line 6, where the data is randomly partitioned into multiple segments. Subsequently, AMM operates on each of these chunks, obviating the need to access the entire dataset for sampling. This efficient approach significantly lowers the time complexity by diminishing the volume of data that needs to be processed, reducing it from N (the size of the data) to N/T, where T represents the number of partitions employed by AMDRS.

4. SAMPLE QUALITY

In our experiments, the datasets are labeled, which means they possess ground-truth *c*'-partitions, denoted by $U \in M_{hc'N}$ of X_N. Assume n_i represents the count of points in subset-i, then the total number of points is given by $N = \sum_{i=1}^{c'} n_i$. From this, we can define the proportion vector of X_N in $\Re^{c'}$ as:

$$v_N = (n_i / N, \ldots., n_{c'} / N) \in \Re^{c'} \tag{2}$$

Algorithm 1 or Algorithm 2, respectively, extracts *c*' MMDRS samples X_{MMDRS}, or AMMDR samples X_{AMMDRS} from the input data. Let $n_t', n'^{\prime}_t$ denote the number of samples drawn from the t-th subset, $1 \leq t \leq c'$ by these two algorithms. For these samples we have the corresponding sample proportion vectors in $\Re^c$:

$$\mathbf{V}_{\text{MMDRS}} = \left(n_1'/c', \ldots, n_c'/c'\right) \in \Re^c \tag{3a}$$

$$\mathbf{V}_{\text{AMMDRS}} = \left(n_1''/c', \ldots, n_c''/c'\right) \in \Re^c \tag{3b}$$

Our objective is to evaluate the degree of alignment between V_{MMDRS} and V_{AMMDRS}. Given that these samples are derived from labeled data, it is feasible to create histograms that contrast the counts of points within each labeled subset with those in the samples. This visual approach offers an assessment of how closely the proportions in the original dataset match those in the sample, all while being independent of both N and p. Especially for smaller values of c, a visual comparison can provide a fairly precise gauge of this alignment.

There are multiple methods to analytically compare V_{MMDRS} or V_{AMMDRS} with V_N. One straightforward approach involves calculating the distances $d(V_N, V_{MMDRS})$ and $d(V_N, V_{AMMDRS})$, using a suitable metric in $\Re^{c'} \times \Re^{c'}$. A distance of zero signifies an impeccable alignment between the proportions in the main dataset and the sample. Secondly, the similarity between the two distributions (V_{MMDRS} or V_{AMMDRS} to V_N) can be calculated via different methods. The Kolmogorov-Smirnov (KS) test is a statistical test used to compare a sample distribution with a reference probability distribution, or to compare two sample distributions [26]. It is a non-parametric test, which means it does not make any assumptions about the shape or parameters of the distributions being compared. It can determine whether two independent samples are drawn from the same population or different populations. This is useful in comparing the characteristics of two groups. Therefore, KS is used to test against the null hypothesis that (V_N, V_{MMDRS}) or (V_N, V_{AMMDRS}) come from the same distribution. The returned p-value is used to interpret the results. For our experiments, we will choose a default significance level of α=0.05. Consequently, if $p>\alpha$=0.05, we uphold the hypothesis that the sample originates from the same distribution as the parent data. In such cases, we will note that the sample has successfully passed the KS test. It is worth mentioning that in our experiments, the number of "samples" for the KS test equates to c', the total count of labeled subsets. Given that the KS test tends to be less precise for smaller sample sizes, it might not offer highly informative outcomes in our context. We will consider a sample to "cover" the input data if every labeled subset gets represented at least once.

5. NUMERICAL EXPERIMENTS

We conducted all experiments on a system equipped with an INTEL Core i7-8700K CPU and 64 GB of RAM, utilizing MATLAB for implementation. The value of T used in line 6 of Algorithm 2 was 10. The horizontal axis on all of the histograms is the cluster number in the labeled data. So, for example, the horizontal axis for the X15 histograms has 15 ticks at k=1 to 15 corresponding to the 15 labeled subsets in the data. The vertical axis on all of the histograms is the ratio of the number of data points (n_i) in subset-i (or sample thereof) to the number of input points (N).

Table 3. Datasets

Name	N	p	c'
X6	399	2	6
X15	5000	2	15
X31	3100	2	31
WDBC	569	30	2

Table 3 lists the four datasets utilized in our experiments. These include three datasets, named as follows: X15 [27], X31 [28], and X6 [29], as well as the Wisconsin Diagnostic Breast Cancer (WDBC) dataset [30]. While each of these datasets underwent identical analysis, due to space constraints, we cannot showcase all the figures in this article. However, a comprehensive collection of graphs can be obtained upon request from the second author.

X15, as seen in Figure 1, showcases clusters visibly distinct, stemming from Gaussian distributions with varied means and covariance matrices. Each cluster has a size varying between

300 and 350. Figure 2 presents six histograms for the dataset X15 when *c*'=20. The input data's histogram is positioned on the upper left, while the random sample is on the upper right. Each histogram is labeled with two values: ED denotes the value of $d(V_N,V_{MM}(*))$ where d represents the Euclidean distance; p signifies the result of the 2-sample KS test (as provided by Matlab) against the significance level α=0.05. A p-value less than the significance level prompts us to reject the 05null hypothesis that both samples come from the same distribution. Conversely, we accept the two samples as being from the same distribution if p>0.05.

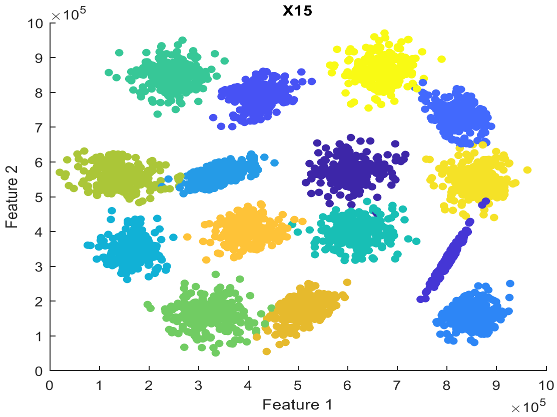

Figure 1. X15~N=5,000 points in p=2 dimensions, c=15

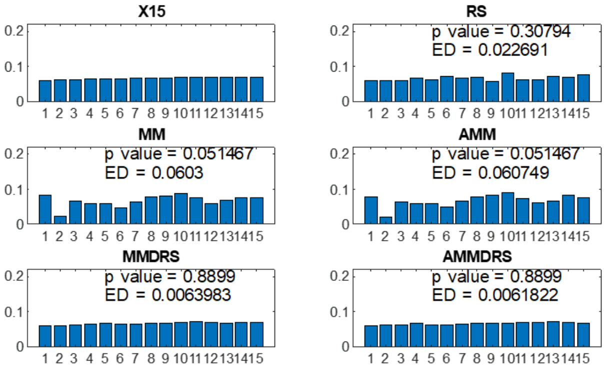

Figure 2. RS, MMDRS and AMMDRS sampling of X15 for c'=20 and n_s=1000

The values of Euclidean distance in Figure 2 show that Random Sampling produces a much higher value of ED (and hence, a lower quality match to the input distribution) than all four of the MM based methods. Comparing MM to AMM, we see that MM does slightly (but only slightly) better for the c' samples. After applying DRS to the two sets (MM and AMM), the ED values are an order of magnitude smaller, and AMMDR does slightly better than MMDRS. Visually, the two DRS sets are much closer to the input distribution than the RS, MM and AMM sets, confirming that the DRS portion of these two algorithms really improves the quality of the samples drawn. The KS test accepts all 5 samples, but clearly prefers the two DRS methods (equal p values of 0.8899) to the MM and AMM samples (p~0.060). The p value for RS (0.307) lies in-between these two pairs of values, which agrees with the visual assessment that RS matches the distribution of X15 better than both MM methods, but not as well as both DRS methods.

The dataset X31, illustrated in Figure 3, comprises 100 points distributed across 31 Gaussian clusters. As a result, the histogram representing the input data exhibits a uniform profile, each bin containing 1/31~0.0322 of the points, as seen in the upper left view of Figure 4, which exhibits the histograms and statistics (ED and KS test) for the five sampling methods at c'=50. The two DRS methods yield visually superior samples, and the ED for these two samples favors MMDRS, albeit slightly. The RS is visually inferior to the other four methods. The p-values for all 5 samples are quite small; the statistical implication of this is to reject the null hypothesis that any of these samples matches the input distribution at significance level α=0.05.

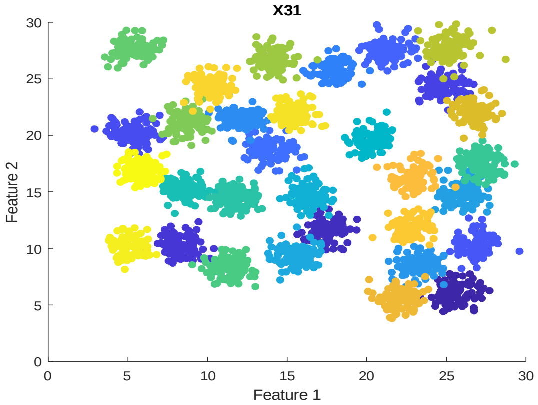

Figure 3. X31~N=3100 points in p=2 dimensions, c=31

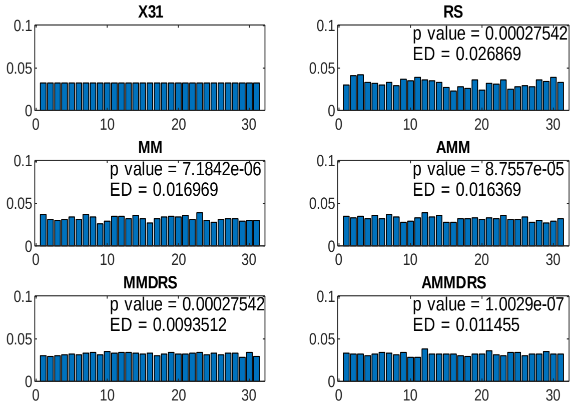

Figure 4. RS, MMDRS and AMMDRS samples of X31: c'=50, n_s=1000

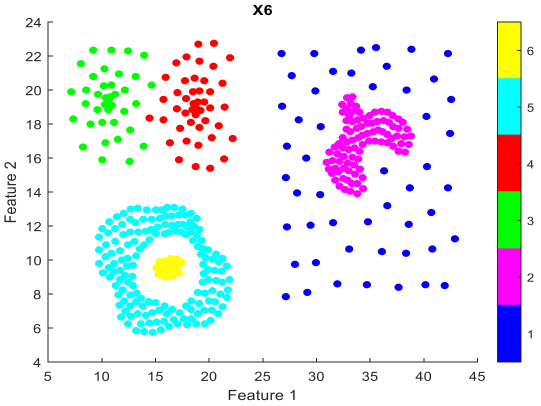

Figure 5. X6~N=399 points in p=2 dimensions, c=6

The dataset X6, displayed in Figure 5, consists of six labeled clusters. The upper left showcases two Gaussian clusters. To the right, a thick cluster of magenta points nestles within a sparser blue subset. Notably, the lower left section of the scatterplot presents a unique clustering reminiscent of a "fried egg". This configuration consists of a vibrant yellow center (depicting the "yolk") encased by a cyan perimeter, symbolizing the "egg white". The specific sizes of these six clusters are as follows: 50, 92, 38, 45, 158, and 16.

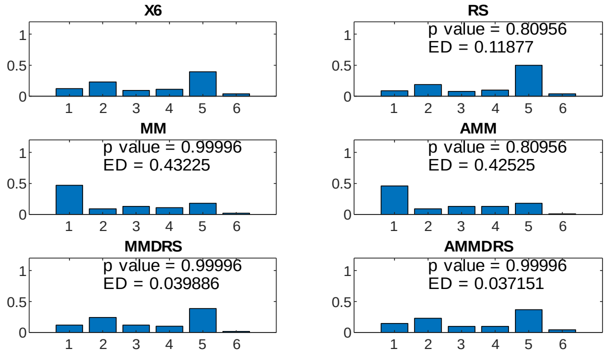

Figure 6. MMDRS and AMMDRS samples of X6: *c'*=10, n_s=100

From Figure 6, first, notice that RS produces a much better visual match to the input data than either MM or AMM, but when DRS is added to the sampling procedure, the visual match of both DRS schemes is slightly better than RS. The ED values agree: RS is better than MM or AMM, but not as good as MMDRS or AMMDRS. All 5 samples pass the KS test, i.e., they accept the match between the samples and parent distributions.

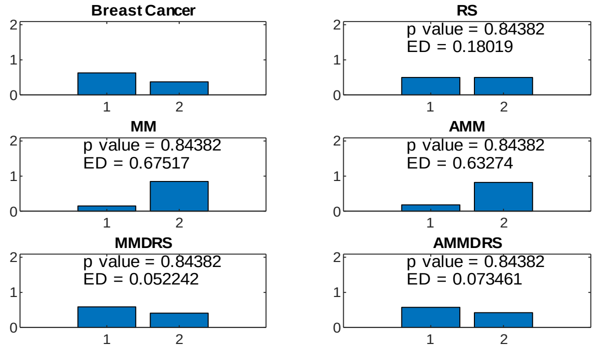

Figure 7. RS, MMDRS and AMMDRS samples of the WDBC data: *c*'=10, n_s=100

In our final experiment, we utilized the Wisconsin Diagnostic Breast Cancer dataset. Figure 7 contains the results. This data set is an odd one, because it has feature vectors in 30 dimensions (p=30), but only N=569 samples. All 5 samples yield the same p value for the KS test, so it is not a useful discriminator for sample quality. Visually, the MM, AMM, and RS samples are poor matches to the input data, while the two DRS samples all look the same and are a better match the actual data. The ED values for the two DRS methods are lower than the MM values and the RS value. From the ED values, we conclude that for this experiment, MMDRS was the best and AMMDRS was next best.

Table 4 shows the CPU time used to compute samples for data set X31. The time required to compute AMM samples is about 1/7 of the time required for MM samples because AMM works on a subset of size N/T of the original dataset, which has N samples. The smaller the subset size (the larger the T value), the smaller will be the time required to compute the AMM samples. But the cost of large T values is the risk of missing samples from the partition of the datasets that does not exist in that subset. Since AMMDRS relies on AMM, it is slightly faster than MMDRS, as can be seen in Table 4.

Table 4. Computational times of the proposed sampling methods on X31 dataset

Method	Time (Seconds)	Dataset	Number of Samples
MM	3.55	X31	1000
AMM	0.57	X31	1000
MMDRS	0.021	X31	1000, c'=50
AMMDRS	0.015	X31	1000, c'=50
RS	0.00035	X31	1000

6. CONCLUSIONS

In this manuscript, we introduced an innovative Approximate MMDRS (AMMDRS) algorithm designed to facilitate the generation of faithful and representative samples from large datasets. This approach empowers the application of traditional clustering algorithms without the necessity of processing the entire dataset, a critical advantage in scenarios where accessing the complete dataset is computationally challenging or impossible due to resources constraint. The significance of this research lies in its potential to make data-driven decision-making more accessible and practical, particularly in situations where working with big data sets is otherwise infeasible. Consequently, this manuscript contributes to the growing body of knowledge aimed at bridging the gap between data analysis and real-world applications, further underscoring the importance of efficient and accurate sampling techniques for handling big data challenges.

The experiments presented here do suggest that the approximate MM method is faster than MM, without a significant loss in sampling accuracy. This is especially important for big data applications where processing the entire datasets is not feasible. Table 4 shows that simple (undirected) random sampling is faster than either of the MM based DRS methods because no time is expended in building the NPR partition. This will be true for any input data set. But in terms of sample quality for cluster analysis, both of the DRS methods produce samples that provide a more faithful representation of the distribution of the input structure than simple random sampling in the experiments reported here. We have used several with different number of cluster and samples for our experiments, but our experience with these methods suggests that as the size of the input data grows, AMDRS will eventually be superior to MMDRS due to computation complexity of MMDRS which needs to access the whole data. We will test this conjecture with a more extensive empirical study in the future.

REFERENCES

[1] Havens, T.C., Bezdek, J.C., Palaniswami, M. (2013). Scalable single linkage hierarchical clustering for big data. In 2013 IEEE eighth international conference on intelligent sensors, sensor networks and information

processing, pp. 396-401. https://doi.org/10.1109/ISSNIP.2013.6529823

[2] YouTube Statistic. (2014). http://www.youtube.com/yt/press/statistics.html

[3] Bezdek. J.C. (2017). A primer on cluster analysis: Four basic methods that (usually) work. First Edition Design Publishing, Sarasota, FL. https://doi.org/10.1201/9781003338086

[4] Dubes, B.C., Jain, A. (1988). Algorithms for clustering data. http://dl.acm.org/citation.cfm?id=SERIES10022.42779

[5] Theodoridis, S., Koutroumbas, K. (2009). Pattern recognition. 6th ed., Academic Press, NY. https://doi.org/10.1016/B978-1-59749-272-0.X0001-2

[6] McLachlan, G.J., Basford, K.E. (1988). Mixture models: Inference and applications to clustering (Vol. 38). New York: M. Dekker. https://doi.org/10.2307/2348072

[7] Bezdek, J.C. (1981). Pattern recognition with fuzzy objective function algorithms. Plenum Press. https://doi.org/10.1007/978-1-4757-0450-1

[8] Keller, J.M., Liu, D., Fogel, D.B. (2016). Fundamentals of computational intelligence: Neural networks, fuzzy systems, and evolutionary computation. John Wiley & Sons. https://doi.org/10.1002/9781119214403

[9] Xu, R., Wunsch, D.C. (2009). Clustering. IEEE Press, Piscataway, NJ. https://doi.org/10.1002/9780470382776

[10] Hartigan, J. (1975). Clustering algorithms. Wiley, NY.

[11] Rayala, V., Kalli, S.R. (2020). Big data clustering using improvised Fuzzy C-Means clustering. Revue d'Intelligence Artificielle, 34(6): 701-708. https://doi.org/10.18280/ria.340604

[12] Hore, P., Hall, L.O., Goldgof, D.B. (2007). Single pass fuzzy c means. In 2007 IEEE International Fuzzy Systems Conference, pp. 1-7. https://doi.org/10.1109/FUZZY.2007.4295372

[13] Hore, P., Hall, L.O., Goldgof, D.B., Gu, Y., Maudsley, A.A., Darkazanli, A. (2009). A scalable framework for segmenting magnetic resonance images. Journal of Signal Processing Systems, 54: 183-203. https://doi.org/10.1007/s11265-008-0243-1

[14] Kumar, D., Bezdek, J.C., Palaniswami, M., Rajasegarar, S., Leckie, C., Havens, T.C. (2015). A hybrid approach to clustering in big data. IEEE transactions on cybernetics, 46(10): 2372-2385. https://doi.org/10.1109/TCYB.2015.2477416

[15] Provost, F., Jensen, D., Oates, T. (1999). Efficient progressive sampling. In Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 23-32. https://doi.org/10.1109/ICDM.2002.1183923

[16] Hathaway, R.J., Bezdek, J.C. (2006). Extending fuzzy and probabilistic clustering to very large data sets. Computational Statistics & Data Analysis, 51(1): 215-234. https://doi.org/10.1016/j.csda.2006.02.008

[17] Pal, N.R., Bezdek, J.C. (2002). Complexity reduction for" large image" processing. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 32(5): 598-611. https://doi.org/10.1109/TSMCB.2002.1033179

[18] Hathaway, R.J., Bezdek, J.C., Huband, J.M. (2006). Scalable visual assessment of cluster tendency for large data sets. Pattern Recognition, 39(7): 1315-1324. https://doi.org/10.1016/j.patcog.2006.02.011

[19] Thorndike, R.L. (1953). Who belongs in the family?. Psychometrika, 18(4): 267-276. https://doi.org/10.1007/BF02289263

[20] Casey, R.G., Nagy, G. (1968). An autonomous reading machine. IEEE Transactions on Computers, 100(5): 492-503. https://doi.org/10.1109/TC.1968.226928

[21] Ibrahim, O.A., Keller, J., Bezdek, J.C., Popescu, M. (2020). Experiments with maximin sampling. In 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1-7. https://doi.org/10.1109/FUZZ48607.2020.9177681

[22] Dunn, J.C. (1973). A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. Journal of Cybernetics, 3(3): 32-57. http://doi.org/10.1080/01969727308546046

[23] Hathaway, R.J., Bezdek, J.C., Huband, J.M. (2006). Maximin initialization for cluster analysis. In Progress in Pattern Recognition, Image Analysis and Applications: 11th Iberoamerican Congress in Pattern Recognition, CIARP 2006 Cancun, Mexico, November 14-17, 2006 Proceedings 11, pp. 14-26. https://doi.org/10.1007/11892755_2

[24] Steponavičė, I., Shirazi-Manesh, M., Hyndman, R.J., Smith-Miles, K., Villanova, L. (2016). On sampling methods for costly multi-objective black-box optimization. Advances in Stochastic and Deterministic Global Optimization, 273-296. https://doi.org/10.1007/978-3-319-29975-4_15

[25] Shao, H., Zhang, P., Chen, X., Li, F., Du, G. (2019). A hybrid and parameter-free clustering algorithm for large data sets. IEEE Access, 7: 24806-24818. https://doi.org/10.1109/ACCESS.2019.2900260

[26] Fränti, P., Virmajoki, O. (2006). Iterative shrinking method for clustering problems. Pattern Recognition, 39(5): 761-775. https://doi.org/10.1016/j.patcog.2005.09.012

[27] Veenman, C.J., Reinders, M.J.T., Backer, E. (2002). A maximum variance cluster algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(9): 1273-1280. https://doi.org/10.1109/TPAMI.2002.1033218

[28] Zahn, C.T. (1971). Graph-theoretical methods for detecting and describing gestalt clusters. IEEE Transactions on Computers, 100(1): 68-86. https://doi.org/10.1109/T-C.1971.223083

[29] https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+ Wisconsin+(Diagnostic).

[30] Massey Jr, F.J. (1951). The Kolmogorov-Smirnov test for goodness of fit. Journal of the American statistical Association, 46(253): 68-78. https://doi.org/10.2307/2280095