IEEE TKDE, 31(4), 641-654, 2019.

A Rapid Hybrid Clustering Algorithm for Large Volumes of High Dimensional Data

Punit Rathore, Dheeraj Kumar, James C. Bezdek, *Life Fellow, IEEE*, Sutharshan Rajasegarar and Marimuthu Palaniswami, *Fellow, IEEE*

Abstract—Clustering large volumes of high-dimensional data is a challenging task. Many clustering algorithms have been developed to address either handling datasets with a very large sample size or with a very high number of dimensions, but they are often impractical when the data is large in both aspects. To simultaneously overcome both the 'curse of dimensionality' problem due to high dimensions and scalability problems due to large sample size, we propose a new fast clustering algorithm called FensiVAT. FensiVAT is a hybrid, ensemble-based clustering algorithm which uses fast data-space reduction and an intelligent sampling strategy. In addition to clustering, FensiVAT also provides visual evidence that is used to estimate the number of clusters (cluster tendency assessment) in the data. In our experiments, we compare FensiVAT with seven state-of-the-art approaches which are popular for large sample size or high-dimensional data clustering. Experimental results suggest that FensiVAT, which can cluster large volumes of high-dimensional datasets in a few seconds, is the fastest and most accurate method of the ones tested.

Index Terms—Big Data Cluster Analysis, Random Projection, Ensemble Clustering, Visual Assessment of Cluster Tendency, Single Linkage, Curse of Dimensionality.

1 Introduction

Data clustering [1] is an essential method of exploratory data analysis in which data are partitioned into several subsets of similar objects. With the rapid advancement of the *Internet of Things* (IoT) technologies, and social network services, we witness tremendous growth of data not only in the volume of the data, but also in the number of features collected for each data object. In many applications such as biomedical imaging, sequencing, and time series matching, the dataset may consist of millions of instances in hundreds to thousands of dimensions [2]. The two most important ways a dataset can be big are: (1) it has a very large number (N) of instances, and (2) each instance has many features (p) i.e. it is high-dimensional data.

A variety of clustering algorithms have been developed for a dataset that has either (1) large N but small p, or (2) small N but large p, but most clustering algorithms are impractical for handling datasets that are large jointly in N and p [3]. Most existing clustering algorithms encounter serious problems related to computational complexities and/or cluster quality for big datasets.

Many papers and surveys [4], [5] discuss different clustering approaches for big datasets. The most popular algorithms are based on partitioning and hierarchical techniques. Among them, single pass k-means [6], mini-batch k-means [7], CLARA (*CLustering LARge Applications*) [8] and CURE (*Clustering Using REpresentatives*) [9] are the most widely known for big datasets. A *single linkage* (SL) type algorithm called *clustering with improved visual assessment of tendency* (clusiVAT) [10], [11], has shown promising results for big datasets. Most of these clustering algorithms use sampling based strategies to reduce computational time. However, they still take a lot of time to cluster very large volumes of high-dimensional data.

There are a number of surveys [12], [13], [14] of high-dimensional data clustering techniques available in the literature. Several widely known clustering algorithms for high-dimensional data [13] are based on dimensionality reduction [15] (from 'up-space' to 'down-space') and subspace clustering [16]. In practice, most of the subspace clustering approaches such as CLIQUE [17], and PROCLUS [18] suffer from long run-times and/or low accuracies for large volumes of high-dimensional data. Dimensionality reduction based approaches such as global projection (e.g., *singular value decomposition* (SVD)) and *random projection* (RP) based ensemble approaches [19], [20] reduce computational time by clustering the projected data in a lower dimensional space. However, they too suffer from space and/or time complexity problems for big datasets, and clusters in the projected space do not necessarily correspond to clusters in the original space.

There has been a limited amount of work on hybrid algorithms that work efficiently on datasets that are jointly large in (N) and (p). These algorithms use random sampling or dimensionality reduction techniques either together [21] or with some other approach such as axis-parallel partitioning [22] or indexing [23], to reduce computation time. However, random sampling may fail [11] to provide a faithful representation of cluster structure in the input data, which may degrade clustering accuracy. The authors of [9] give a theorem that (in probability) insures representative random samples, but this result often leads to samples that are roughly half the size of the original data. This is still too large

- *Punit Rathore, and Marimuthu Palaniswami are with the Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville, Victoria, Australia.*
  E-mail: {prathore.student, palani}@unimelb.edu.au.
- *Dheeraj Kumar is with the Lyles School of Civil Engineering, Purdue University, USA.*
  E-mail: kumar299@purdue.edu
- *James C. Bezdek is with the School of Computing and Information Systems, The University of Melbourne, Victoria, Australia.*
  E-mail: jbezdek@unimelb.edu.au.
- *Sutharshan Rajasegarar is with the School of Information Technology, Deakin University, Geelong, Victoria, Australia.*
  E-mail: srajas@deakin.edu.au.

when N is big, say $N > 10^7$. Therefore, these approaches either take hours for large size datasets having hundred to thousands of dimensions, and/or sacrifice accuracy for faster computation time. Moreover, datasets used in these papers are not considered large in today's computing environment.

To deal with large amounts of high-dimensional data, this paper introduces a rapid, hybrid clustering algorithm, which efficiently integrates (i) a new *random projection* (RP) based-ensemble technique; (ii) an improved visual assessment of cluster tendency (iVAT) algorithm [24], and (iii) a smart sampling strategy, called *Maximin and Random Sampling* (MMRS) [25], [26]. The proposed method achieves fast clustering by combining ensembles of random projections with scalable version of iVAT, hence we call it FensiVAT.

FensiVAT aggregates multiple distance matrices, computed in a lower-dimensional space, to obtain the iVAT image in a fast and efficient manner, which provides visual evidence about the number of clusters to seek in the original dataset. MMRS sampling picks distinguished objects from the dataset, hence it requires relatively very few samples compared to random sampling to yield a diverse subset of the big data, that represents the cluster structure in the original (big) dataset.

Our major contributions are as follows:

- We propose a hybrid clustering algorithm, FensiVAT, for clustering large volumes of high-dimensional data, and perform experiments to compare its performance with nine big data clustering methods, viz., single pass k-means, mini-batch k-means, CLARA, CURE, clusiVAT, GARDENkm [27], and FastSpec [21] and two high-dimensional data clustering approaches, PROCLUS, and *random projection based ensemble clustering* (RP-EN) [19], [20].
  - We perform experiments on two synthetic (having 100,000 samples in 1000 dimensions) and six real labeled (except US Census 1990) datasets that are large in sample size (N) and dimension (p), to compare FensiVAT to the nine other methods in terms of CPU time and clustering accuracy.
  - We illustrate the utility of FensiVAT for one big, unlabeled dataset (US Census 1990). Unlike other clustering algorithms (except clusiVAT), which rely on intuition or need prespecification of the number of clusters in the dataset, FensiVAT is able to subjectively determine, via visual inspection of a reordered image of the distance matrix on the data, the number of clusters to seek.
- We use a statistical measure to compare the cluster distributions in samples obtained from three sampling strategies: random sampling, MMRS sampling in the p-dimensional upspace, and MMRS sampling in the q-dimensional downspace (we will call this type of sampling *Near-MMRS*). Our experiments shows that Near-MMRS samples accurately portray the distribution of the original data in lower dimensions.
- We illustrate that the iVAT image, obtained with our new ensemble-based distance matrices aggregation technique, provides reliable visual evidence about the number of clusters that may be present in big high-dimensional data, in a few seconds.

In summary, our hybrid scheme can cluster millions of data points having hundred to thousands dimensions, *in a few seconds time* without sacrificing accuracy, and it is orders of magnitude faster than the other state-of-the-art algorithms discussed in the paper.

Here is an outline of the rest of this article. Section 2 presents a brief summary of the VAT and iVAT algorithms and random projection methods. Section 3 reviews related work. The proposed algorithm, FensiVAT is discussed in Section 4. Section 5 presents the experiments and results, followed by our conclusions in Section 6.

2 Preliminaries

2.1 Visual Assessment of Tendency (VAT)

Consider a set of N objects $O = \{o_1, o_2, ..., o_N\}$, partitioned into $k \in \{2, .., N-1\}$ subsets, where each object o_i is represented by a p-dimensional feature vector, $\mathbf{x}_i \in \mathbb{R}^p$. The problem of estimating the number of clusters k prior to actual clustering is known as cluster tendency assessment. The data can also be presented in the form of dissimilarity matrix $D_N = [d_{ij}]$, where d_{ij} represents dissimilarity between o_i and o_j.

The FensiVAT clustering algorithm finds its root in the *visual assessment of tendency* (VAT) [28] algorithm. The VAT algorithm is based on (but not identical to) Prim's algorithm for finding the *minimum spanning tree* (MST) of a weighted undirected graph. It reorders the dissimilarity matrix D_N to D_N^* using a modified Prim's algorithm, such that dark blocks along the diagonal of the reordered image $I(D_N^*)$ potentially represent different clusters. The VAT algorithm also provides the VAT reordering indices P of D_N, and ordering of the MST cut magnitudes, c, used by single linkage clustering as inputs.

An *improved VAT* (iVAT) [24] provides a much sharper reordered diagonal matrix image by replacing input distance d_{ij} in distance matrix D_N by distances $D_N^{'} = [d_{ij}^{'}]$,

$$d_{ij}^{'} = \min_{r \in P_{ij}} \max_{1<h<|r|} D_{N_{r[h]r[h+1]}}, \tag{1}$$

where $r \in P_{ij}$ is an acyclic path in the set of all acyclic paths from object (o_i) and (o_j) (vertices i and j) in O. VAT and iVAT suffer from resolution and memory constraints that limit their usefulness for input matrix sizes of order of 10^5 and so. To overcome these limitations, *scalable single linkage algorithms* sVAT/siVAT [24], [29] were proposed, which first find $n << N$ MMRS samples and then construct an image of this sample using distance matrix $D_n^{'*}$. This image $I(D_n^{'*})$ usually provides a useful visual estimate of k without the need to calculate the very large distance matrix, D_N of the big dataset, and circumvents the problem that $I(D_N^*)$ is not computable. The siVAT scheme does not involve any sensitive threshold parameter, and requires the user to supply only two parameters: n the desired sample size, and $k^{'}$, an overestimate of k, the assumed number of clusters, to obtain $k^{'}$ distinguished objects in the sample.

Since single-linkage clusters are always diagonally aligned in the VAT/iVAT ordered images, we merely have to cut the largest $(k-1)$ edges in the MST and form the corresponding k aligned partition. For big data clustering, siVAT-SL and clusiVAT [10], [30] use this idea to extend the k partition of D_n (or D_n^*) noniteratively to the $(N-n)$ unlabeled objects in O using the *nearest (object) prototype rule (NOPR)*. Both siVAT-SL and clusiVAT are adequate for large sample size datasets, however, they still suffer from large computation time when the dataset is large in the number of dimensions.

The two main time consuming steps in both algorithms for large, high-dimensional data clustering are (i) Maximin step of MMRS sampling; and (ii) Extension. In the Maximin step, k' distinguished objects are chosen which are furthest from each other in the dataset. This requires the computation of a $k' \times N$ distance matrix, $\hat{D} \subset D_N$, in p dimensions. In the extension step, the labels of n samples are used to label the remaining $(N-n)$ objects in the data using the NOPR. This requires the computation of an $n \times (N-n)$ distance matrix $\hat{\hat{D}} \subset D_N$, the distances again being in p dimensions. In an intermediate step of clusiVAT, SL clustering is applied to an $n \times n$ matrix, D_n. Because, N and $(N-n)$ can be very large for big datasets, and the distance computations ($\hat{D}$ and $\hat{\hat{D}}$) are performed in the original (high) p-dimension, siVAT-SL and clusiVAT take a large amount of time to cluster large volumes of high-dimensional dataset.

2.2 Random Projection

Dimensionality reduction methods, such as *principal component analysis* (PCA) or *linear discriminant analysis* (LDA) incur high computational cost for large volumes of high dimensional data. Two key properties, namely low computational complexity and (approximate) distance preservation [31] in lower dimension subspaces, make *random projection* (RP) an attractive choice for dimensionality reduction in our approach.

Random projections are based on the *Johnson-Lindenstrauss* (JL) lemma [32], which states that a set of N points $X = \{x_1, .., x_N\} \subset \mathbb{R}^p$ (denoted as the 'upspace') can be linearly projected (with approximate preservation of distances in probability) into a set of points $Y = \{y_1, .., y_N\} \subset \mathbb{R}^q, q \ll p$, (denoted as the 'downspace') using a random projection matrix $T \subset \mathbb{R}^{p \times q}$. In this paper, we adopt a variant of the JL lemma proposed by Achlioptas in [33]. The theorem is as follows:

Theorem 1. *Given a set of N points $X \subset \mathbb{R}^p$, and $\varepsilon, \beta > 0$, for any integer q*

$$q \geq q_0 = (4+2\beta)(\varepsilon^2/2 - \varepsilon^3/3)^{-1} log(N). \quad (2)$$

Let $Y = \frac{1}{\sqrt{q}} XT$ be the projection matrix of the N points in $\mathbb{R}^q$, where T is a $p \times q$ random matrix, whose entries t_{ij} are *independently and identically distributed* (i.i.d) with $t_{ij} = +1$ or -1, with equal probability. Let $f : \mathbb{R}^p \to \mathbb{R}^q$ map the i^{th} row of X to the i^{th} row of Y. Then for any $u, v \in X$ with probability at least $1 - N^{-\beta}$, we have

$$(1-\varepsilon)||u-v||_2 \leq ||f(u) - f(v)||_2 \leq (1+\varepsilon)||u-v||_2 \quad \blacksquare$$

The parameter ε controls the accuracy of distance preservation, while β controls the probability that distance preservation is within $1 \pm \varepsilon$. According to Theorem 1, if the reduced (downspace) dimension q obeys inequality (2), then pairwise Euclidean distances are preserved within a multiplicative factor of $1 \pm \varepsilon$, and we say that Y has a *JL certificate* (in probability). However, the authors in [34] assert that the JL result often holds for $q \ll q_0$. They termed such projections "*rogue random projections*" (RRP). We will study the use of rogue random projections in our experiments.

3 Related Work

Clustering algorithms for big data can be broadly categorized into two classes: partitional and hierarchical-based methods. The partitional algorithms attempt to determine partitions that optimize a given objective function. The k-means algorithm is one of the most popular, and computationally efficient partitional clustering algorithms. A fast, scalable version of k-means, scaleKM aka *single pass k-means* (spkm) was presented in [6] for big datasets. In this paper, sphkm, which is a crisp adaptation of the single pass fuzzy k-means algorithm [35] for big data, is used for comparison. It first divides the N points into s chunks and requires only a portion of the data (one of the s chunks) to be stored in the memory for k-means clustering. Then, it updates the current model (k weighted centroids) over the contents of the buffer (other chunks) iteratively until the whole dataset is loaded and processed. After obtaining the final k centroids, the big data are labeled based on the label of the nearest object, i.e., the standard k-means nearest prototype rule. A major limitation of almost all variants of the k-means algorithms is the requirement that the number of clusters k is a user-defined input, which is, in the case of unlabeled data, never known.

Mini-batch k-means (MBKM) [7], also known as web-scale k-means, was proposed as an alternative to k-means for large datasets. This scheme processes small random batches (mini-batches) of the dataset to reduce the computation time, while attempting to optimize the same objective function. The algorithm iterates between two major steps. In the first step, samples are drawn randomly from the dataset, to form a mini batch. Then, each data point in the batch is assigned to a cluster (nearest centroid). In the second step, a new random sample is obtained and used to update the centroid until termination is achieved. For each sample in the batch, the assigned centroid is updated using a convex combination of the samples of mini batch and previous samples assigned to that centroid, applying a learning rate. The learning rate is the inverse of number of samples in each batch assigned to a centroid during the process. The effect of new samples decreases as the number of iterations increases. These steps are performed until termination or until a pre-determined number of iterations is obtained. This scheme reduces the number of distance computations per iteration at the cost of lower cluster quality.

Another partition-based algorithm, CLARA (*CLustering LARge Applications*) [8] also relies on sampling. CLARA begins by randomly drawing $(40+2k)$ samples from the big data using uniform random sampling. *Partitioning Around Medoids* (PAM) then operates on these samples to find the best k-medoids. The remaining objects are labeled with the nearest prototype rule, and the average dissimilarity between crisp clusters is computed. If this improves the objective, retain these k medoids and continue.

In hierarchical clustering, data are organized into hierarchical clusters based on the proximity between pairs of objects. CURE [9] is a well known clustering algorithm for large datasets, which uses random sampling and hierarchical partitioning to speed up the computations. It randomly samples a constant number of points from the large dataset so that the selected (representative) points (hopefully) retain the geometry of the entire dataset. In CURE, each cluster is represented by a fixed number c of well-scattered points, which are shrunk towards the centroid of a cluster by a fraction α. These scattered points after shrinking are defined as representatives of the cluster. Then, the clusters with the nearest representative points are merged at each step until the desired value of k is attained, akin to SL. Heaps and k-d trees are the main data structures used by CURE for efficient search. As the number of clusters in the data increases, the probability of CURE samples retaining the data geometry decreases, and hence, the accuracy of

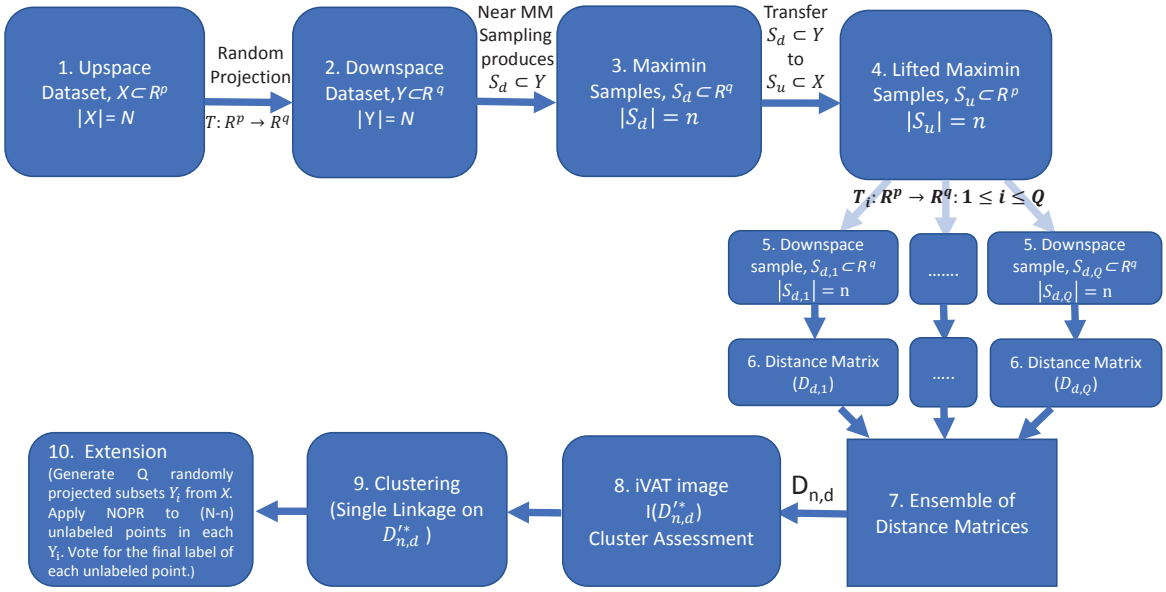


Fig. 1: The FensiVAT architecture.

CURE decreases.

These methods depend on nearest neighbor(s) information, so they are ineffective when clustering high-dimensional data, due to diminishing differences in distance in high-dimensional upspaces [13]. Hierarchical clustering algorithms such as CURE, siVAT-SL, and clusiVAT are fast for large N; however, they are inefficient for datasets jointly large in N and p. At the other extreme, there are methods that excel for large p and small N. Subspace clustering methods [16] do not suffer from nearest neighbor problems in high-dimensional space. PROCLUS is a subspace clustering approach, which first samples the data, then selects a set of k medoids, and iteratively improves the clustering. PROCLUS is capable of discovering arbitrarily shaped clusters in high-dimensional datasets. However, PROCLUS is very sensitive to input parameters, and is not efficient for very large N.

Random projection is a simple, and efficient dimension reduction method that has been shown to work surprisingly well in practice [15]. Since the clustering results using a single random projection are often unstable, ensemble-based techniques [19], [20] aggregate the result of various lower dimensional clustering results to form an affinity matrix, and use a clustering technique on its rows to get final results. In this paper, a *random projection-based ensemble technique* [20], called RP-EN, is compared to FensiVAT.

A few algorithms have been proposed for clustering data that are jointly large in (N) and (p). Almost all these approaches use sampling and/or dimensionality reduction for clustering high-dimensional massively large datasets. *O-cluster* [22] (*Orthogonal partitioning CLUSTERing*) combines active random sampling with an axis-parallel partitioning strategy to identify continuous areas of high density in the input space. O-cluster works well for high-dimensions, but it does not function optimally when the dimensionality is low. The low number of dimensions makes the use of axis-parallel partitioning algorithm problematic. O-cluster uses a parameter *sensitivity*, ρ, which require careful tuning when it is applied to a dataset where the number of clusters is unknown, and also, it requires larger buffer size to correctly identify all original clusters in the dataset. *GARDEN k-means* [27] (we denote it GARDENkm) begins with Gamma region density partitioning scheme for data summarization. Using this partitioning technique, it reduces the empty regions in the data space so that only tight, high-dense regions are retained. Then, it utilizes k-means to cluster summarized information. Like k-means, this algorithm also requires the number of clusters (k) as an input prior to clustering. Another hybrid approach, *fast spectral clustering* (we denote it FastSpec) [36], combines random sampling and FastMap projection with spectral clustering to identify clusters. In an intermediate step, it computes an affinity matrix of $N \times r$ dimension in the downspace (r is the number of random samples, $r = 300k$ [36]), and a diagonal matrix of $N \times N$ size, which can be very big for big datasets. Therefore, FastSpec has very high space complexity. In summary, all the methods reviewed either take hours for large size datasets having hundreds to thousands of dimensions, and/or sacrifice accuracy for faster computation time. In the next section, we discuss the FensiVAT algorithm.

4 FensiVAT Algorithm

The essential steps in FensiVAT are: (i) **Near-MMRS Sampling:** MMRS sampling is done in Y, the downspace (subscript d), to obtain a small and diverse subset $S_d \subset Y$ from the full dataset, which is then lifted by using the same indices to the upspace (subscript u), $S_u \subset X$; and (ii) **Ensemble:** Aggregation of Q $n \times n$ distance matrices, $\{D_i\}_{i=1}^{Q}$, computed from multiple random projections of S_u to obtain Q sets of Near-MMRS samples $\{S_{d,i}\}_{i=1}^{Q}$ in the downspace. This is done to obtain a reliable output iVAT image, $I(D'^*_{n,d})$, which visually suggests the number of clusters, k, in the dataset, (iii) **Clustering**: SL partitioning on the $D'^*_{n,d}$ to obtain k clusters, and (iv) **Extension in downspace** to label the remaining data points in the dataset Y by giving them the label of their nearest object from sample S_d. Our FensiVAT algorithm for large, high-dimensional data clustering is presented in Algorithm 1. Below, we explain each step of FensiVAT algorithm, whose architecture shown in Fig. 1.

4.1 Near-MMRS Sampling

The input data to FensiVAT is $X = \{x_1, ..., x_N\} \subset \mathbb{R}^p$; N and p are large. In the second step, random projection (as discussed in Section 2.2) is applied to $X \subset \mathbb{R}^p$ to obtain downspace data $Y \subset \mathbb{R}^q$. Unlike ensemble-based approaches, random projection is applied only once to the large dataset to obtain a downspace dataset, which is subsequently used for the sampling step. It is possible that the clusters in a sample from the downspace dataset Y are drastically different from the points that MMRS sampling would produce when applied to X. This point is discussed below with the Near-MMRS sampling (third) step of the FensiVAT algorithm.

Near-MMRS sampling begins by finding the k' Maximin (MM) samples (*distinguished objects*) in Y, which are furthest from each other. MM sampling starts at a random point, and then chooses as the second MM sample the point which is furthest from the initial point with respect to a chosen measure of distance on the set being sampled. The third object selected maximizes the distance from both of the first two points. This process continues until k' MM samples are chosen. Since MM is performed in downspace, we call it Near-MM. Then, each object in O is grouped with its nearest *distinguished object*. This stage divides the entire dataset O into k' groups, $\{Z_i\}_{i=1}^{k'}$ by associating $|Z_i|$ objects to the ith *distinguished object*, which provides a representation of each of the k' clusters. This grouping task requires the computation of a $k' \times N$ matrix $\hat{D}$ (Refer to Section 2.1) now done in downspace ($\mathbb{R}^q$), which reduces the computational time that would be needed for the calculations of a $k' \times N$ distance matrix of $p-$dimensional feature vectors. Finally, the sample S_d of size n (just a small fraction of N), is built by selecting random data points (Random sampling (RS)) from each of the k' clusters $\{Z_i\}_{i=1}^{k'}$. The number of points, n_i extracted from cluster Z_i is proportional to the number of datapoints in Z_i, namely, $n_i = \lceil n \times |Z_i|/N \rceil$, where $\lceil \cdot \rceil$ denotes the ceiling function. Hence the term MMRS is used for the overall process.

The approximate distance preservation (within $1 \pm \varepsilon$) property of randomly projected pairs from X asserted by Theorem 1 supports a belief that if the Near-MM distinguished objects in Y are generated by applying MM to it, beginning with the same initial point, that the MM samples in Y should be the same or close (due to approximation distance error) to the k' MM points in X (upspace) that would be produced by MM sampling in the upspace. Two Propositions from [26] about this procedure provide some justification for believing this.

Proposition 1. *Let O be a finite set of distinct objects that can be partitioned into k compact-separated (CS) [37] clusters and let $k' \geq k$, then*

***A**. Step 3(a) of FensiVAT algorithm (the first step of Near-MMRS sampling algorithm) selects at least one distinguished object from each cluster.*

***B.** In addition, if $n_i = n \times |Z_i|/N$ (Step 3(c) in Algorithm 1) is an integer for $i = 1, 2, ..., k'$ then the proportion of the objects in the MMRS sample from cluster $O^{(j)}$ equals the proportion of objects from same cluster $O^{(j)}$ in the original data, for $j = 1, 2, ..., k$.*

Proof. See [26] for proof. ■

In Near-MMRS sampling, MMRS sampling is performed in the randomly projected lower dimensional space Y (downspace). Therefore, if dataset X has k CS clusters and $k' \geq k$, and if downspace data Y has k CS clusters, and carries a JL certificate ($q \geq q_0$) as in Theorem 1, then Proposition 1A guarantees that Near-MMRS sampling will select at least one distinguished object from each of the k clusters, and Proposition 1B assures us that the proportion of the objects in each cluster in Near-MMRS sample would be similar to the proportion of objects in each subset in original data.

Four sets of empirical experiments support our intuition about Near-MMRS samples. (i) Bezdek *et al.* [34] assert that the JL bound $(1 \pm \varepsilon)$ often holds even in spaces with $q \leq q_0$. They called this *rogue random projection* (RRP). Their experiments using RRPs on various datasets demonstrated empirically that the JL result in Theorem 1 often holds for $q << q_0$. (even with $q = 2$ for some datasets); (ii) Havens *et al.* [29] and (iii) Kumar *et al.* [10] demonstrated using various numerical examples that, even for non-CS datasets (having overlapping clusters), MMRS sampling in sVAT/sVAT-SL/clusiVAT picks objects from each cluster ($k' \geq k$), and produces a good representation of the input cluster structure (present in original dataset) in the output sample; (iv) Our experiments on various CS and non-CS datasets in this paper, for $q \leq q_0$, demonstrate that Near-MMRS sampling approximately preserves the distribution of the original data in randomly projected lower dimensions.

The missing link in the theory is whether or not the CS clusters in X are also CS clusters in Y under random projection. If so, then Proposition 1 would be valid in Y when random matrix T carries a JL certificate. So far, we have been unable to prove this conjecture.

4.2 Distance Matrix using Ensemble Method

The third (previous) step provides n samples in the downspace, $S_d \subset \mathbb{R}^q$, which can be used to build an $n \times n$ distance matrix $D_{n,d}$. We need a reliable iVAT image in order to select the number of clusters obtained by SL in penultimate steps of FensiVAT. The VAT/iVAT image provides a subjective visual assessment of potential cluster substructure based on how distinctive the dark blocks (clusters) appear in the image. However, the quality of the image of the reordered distance matrix $D'^{*}_{n,d}$, obtained by applying VAT/iVAT to $D_{n,d}$, often turns out to be very poor due to the unstable nature of random projection. Hence, we turned to an ensemble-based approach to obtain a good quality iVAT image from multiple reordered distance matrices ($\{D'^{*}_{d,i}\}_{i=1}^{Q}$) in the downspace. Since the ordering of the data in every reordered matrix $D'^{*}_{d,i}$ may be different, it is not feasible to directly aggregate multiple reordered distance matrices ($\{D'^{*}_{d,i}\}_{i=1}^{Q}$). Therefore, we devised a new method to aggregate the Q $n \times n$ ensemble of distance matrices to obtain a better quality iVAT image.

Our ensemble-based approach to build the aggregate $n \times n$ distance matrix, $D_{n,d}$ is shown in Steps 4-7 of Algorithm 1. First (in the fourth step), the Near-MMRS samples S_d are back-projected to the upspace by using the sample indices in S_d to identify the corresponding samples S_u in X. Then we apply random projection to S_u Q times, resulting in the downspace sample sets $\{S_{d,i}\}_{i=1}^{Q}$ (Step 5 in Fig. 1).

Next, the Q downspace samples, $\{S_{d,i}\}_{i=1}^{Q}$ are used to compute Q distance matrices, $\{D_{d,i}\}_{i=1}^{Q}$ in the sixth step. Since, the downspace samples can be drastically different from each other due to the random nature of the mapping from upspace to downspace, the distance matrices will be diverse. Therefore, we aggregate the Q $n \times n$ distance matrices to obtain a more reliable distance matrix, which in turn yields a better iVAT image than the Q individual iVAT images. The aggregation (Step 7) is

Algorithm 1 FensiVAT

Step 1. Input: Dataset $X = \{x_1, .., x_N\} \subset \mathbb{R}^p$, downspace dimension q, Overestimate of true number of clusters k', number of approximated samples n, number of RPs Q
Output: $D'^{*}_{n,d}$ - iVAT reordered dissimilarity matrix of $D_{n,d}$.
U - cluster membership vector of data points in O.
Step 2. Dataset generation in downspace.
Generate downspace datasets $Y \subset \mathbb{R}^{N\times q}$ using $Y = \frac{1}{\sqrt{q}}XT$, where $T \in \mathbb{R}^{p\times q}$ is the random matrix as discussed in Section 2.2.
Step 3: Near-MMRS Sampling: MMRS on Y.
a. *Select the indices m of k′ distinguished objects (Maximin samples).*
Randomly select the first distinguished object x_{m_0}.
Distance of x_{m_0} from N points, $z = (z_1, ...z_N) = \{dist(x_{m_0}, x_1), ..., dist(x_{m_0}, x_N)\} = (r_{m_0 1}, ...r_{m_0 n})$.
for $i = 1$ to k' **do**
$z \leftarrow (min\{z_1, r_{m_{i-1}1}\}, ..., min\{z_N, r_{m_{i-1}N}\})$
$m_i = \arg\max_{1\le j\le N} z_j$
end for
b. *Group each object in O with its nearest distinguished object.*
$Z_1 = Z_2 .... = Z_{k'} = \emptyset$.
for $t = 1$ to N **do**
$l = \arg\min_{1\le i\le k'}\{r_{m_i t}\}$
$Z_l = Z_l \cup \{t\}$
end for
c. *Randomly select data near each distinguished point to obtain the n number of samples.*
$n_i = \lceil n \times |Z_i|/N \rceil \quad i = 1, 2, ..., k'$.
Draw n_i unique random indices from Z_i to build sample Z'_i.
$S_d = \bigcup_{i=1}^{k'} Z'_i$
Step 4-7: Ensemble method to obtain a reliable iVAT image.
Generate Q, downspace datasets $\{S_{d,i}\}_{i=1}^{Q} \subset \mathbb{R}^q$ from $S_u \subset \mathbb{R}^p$ ($S_d \to S_u$), using random matrices $\{T_i\}_{i=1}^{Q} \in \mathbb{R}^{q\times q}, |S_u| = |S_d| = n$.
Compute distance matrices $\{D_{d,i}\}_{i=1}^{Q}$ from $\{S_{d,i}\}_{i=1}^{Q}$.
$D_{n,d} \leftarrow 0$ (Initialize a $n \times n$ distance matrix).
for $i = 1$ to Q **do**
$W_i = NormalizeRows(D_{d,i})$
$V_i = \frac{1}{2}(W_i + W_i^{\mathsf{T}})$
$D_{n,d} = D_{n,d} + V_i$
end for
Step 8: Apply VAT/iVAT on $D_{n,d}$, returning $D'^{*}_{n,d}$, P, c.
Choose the number of clusters k using image of $D'^{*}_{n,d}$.
Step 9: Clustering:
Find indices u of k largest values in MST cut magnitudes c.
Form the aligned partition, $U^* = \{u_1 : u_2 - u_1 : ... : u_k - u_{k-1}\}$
$U_{S_u} = U^*_{P_i},\ 1 \le i \le k$.
Step 10: Extension in Downspace:
Generate downspace datasets $\{Y_i\}_{i=1}^{Q} \subset \mathbb{R}^q$ using RP, $|Y_i| = N$.
for each Y_i **do**
Consider sample $Y^{(i)}_{S_d} \subset \mathbb{R}^q$ and $Y_i - Y^{(i)}_{S_d} \subset \mathbb{R}^q$, where $|Y^{(i)}_{S_d}| = n$, and $|Y_i - Y^{(i)}_{S_d}| = N - n$.
for each data point, $\hat{y} \in Y_i - Y^{(i)}_{S_d}$ **do**
$l = \arg\min_{i\in S_d} \{dist\{\hat{y}, y_i\}\}$
$U^{(i)}_{\hat{y}} = U_l$
end for
end for
U = Mode of labels for each data points $U^{(i)}_{\hat{y}}$.

performed in three sub-steps: Normalization, Symmetrization, and Summation.

Normalization: Since each distance matrix is computed from randomly projected samples, the distance of each data point from the remaining data points may have a different range in different distance matrices. Therefore, the distance of each data point from the remaining datapoints is normalized to a unit scale in each distance matrix. The rows (or columns) of each $D_{d,i}$ are normalized such that the ij-th entry of $D_{d,i}$ is in $[0,1]$, and the row sum of each row is 1.

Symmetrization: The normalized distance matrices, W_i (in Algorithm 1), may be asymmetric. The input distance matrix to VAT/iVAT must be symmetric, so all normalized distance matrices are replaced by symmetric matrices using $V_i = \frac{1}{2}(W_i + W_i^{\mathsf{T}})$.

Summation: After symmetrization, the output distance matrix $D_{n,d}$ is obtained using element-wise summation of the Q distance matrices $\{V_i\}_{i=1}^{Q}$.

Cluster Assessment

In the eighth step, the VAT/iVAT algorithm is applied to distance matrix $D_{n,d}$, which returns a reordered matrix, $D'^{*}_{n,d}$ and the cut magnitudes of the MST links, c. The visualization of $D'^{*}_{n,d}$ using $I(D'^{*}_{n,d})$ suggests the number of clusters k present in the dataset. The comparison of iVAT images obtained using the Q single distance matrices $\{D_{d,i}\}_{i=1}^{Q}$ to the image based on $D_{n,d}$ is discussed in Section 5. In this article, human interpretation is used to estimate the number of clusters by viewing the output iVAT image, but there are also methods [38], [39], [40] to automatically determine the number of clusters from VAT/iVAT images or $D'^{*}_{n,d}$.

4.3 Clustering

All single linkage partitions are *aligned* partitions [41] in the VAT/iVAT ordered matrices, so SL is an obvious choice for the clustering algorithm in Step 9. Having the estimate of the number of clusters, k from the previous step, we cut the $k-1$ longest edges in the iVAT-built MST, resulting in k single linkage clusters.

If the dataset is complex and clusters are intermixed, cutting the $k-1$ longest edges may not always be a good strategy as the datapoints (outliers), which are typically furthest from normal clusters, might comprise most of the $k-1$ longest edges of the MST, leading to misleading partitions. Such datapoints need to be partitioned (usually in their own cluster) before a reliable partition can be found via the SL criterion. However, the iVAT image provides visual evidence as to how large the clusters should be. Thus, if the size of SL-clusters does not match well the visual evidence, then the partition can be discarded (perhaps choosing a different clustering algorithm to partition the sample of feature vectors in $\mathbb{R}^p$ or throwing out data from small clusters).

Next, the aligned partition $\{U^*_{P_i}\}_{i=1}^{k}$ is calculated using the indices of the $k-1$ longest edges. Since the objects in $U^*_{P_i}$ are arranged according to VAT reordering indices P, we reorder the cluster label vector $U^*_{P_i}$ to match the index-ordering of samples S_u in the original objets, resulting in the partition U_{S_u} of S_u.

4.4 Extension

In the extension step (Step 10) of FensiVAT, we label the remaining $\tilde{N} = (N-n)$ data points in O, by giving them the label of their nearest object in S_d. This requires the computation of an $n \times \tilde{N}$ size matrix, $\hat{D}$, with computational complexity $O(qn\tilde{N})$. In this step, we use the sample S_d and feature vectors Y in $\mathbb{R}^q$ (obtained

in Step 2) to compute the distance matrix $\hat{\tilde{D}}$. This further reduces the computation time which would be needed for the equivalent operation in $\mathbb{R}^p$.

Next, the remaining $\tilde{N}$ datapoints in O are labeled using this distance matrix, based on the label of the nearest object in S_d. Although, a single random projection (RP) might be sufficient to achieve comparable accuracy in the NOPR labeling step, several [42] RPs are used to best ensure robust nearest neighbour search in NOPR. First, multiple RPs are applied on the full dataset to get multiple Ys. Then, the sample labels are extended to each of these Ys using NOPR, which would give multiple sets of labels $\{U_{\hat{y}}^{(i)}\}_{i=1}^{Q}$ for full dataset. The final labels (U) are selected using voting, based on the labels cast by each voter from each RP, for each remaining data point in O.

Time Complexity

For dataset $Y \subset \mathbb{R}^q$, the computational complexity in the first and second stages of Near-MMRS sampling are $O(qk'N)$, and the last stage requires $O(qn^2)$ operations to build sample S_d. The complexity in computing multiple distance matrices in ensemble step is $O(qn^2Q)$ and the complexity of iVAT is $O(qn^2)$. The computational complexity to compute the aligned partition and reordering in clustering step is $O(N)$. The computational complexity of extension step is $O(qn\tilde{N}Q)$. So, the overall complexity of FensiVAT is $O(max\{qk'N, qn^2, qn^2Q, N, qn\tilde{N}Q\})$. In other words, FensiVAT is linear in N, i.e., it is scalable with respect to the number of samples while simultaneously reducing the dimensional complexity from p to q.

5 Experiments

We performed six set of experiments on two synthetic and six real datasets, that are relatively big in sample size (N) as well as in dimension (p). In the first experiment, we compare the cluster distribution obtained using three sampling schemes. In the second experiment, we compare the quality of iVAT images, obtained using Q distance matrices built with Q RPs to the quality of iVAT image obtained using ensemble distance matrix. In the third experiment, we explore the capability of FensiVAT to visually suggest the number of clusters in big datasets in the downspace dimension. In the fourth and fifth experiments, we demonstrate the performance of FensiVAT for different numbers (Q) of RPs in the ensemble step and for different downspace dimensions $q = 5, 10, 20, 30, 50$, and 100, respectively. In the last experiment, we compare the performance of FensiVAT with nine state-of-the-art methods, discussed in Section 3. These nine approaches are clusiVAT [10], MBKM [7], CLARA [8], spkm [6], CURE [9], RP-EN [20], PROCLUS [18], GARDENkm [27], and FastSpec [21]. While the comparison of our proposed algorithm with O-Cluster [22] would have been desirable, a publicly available code does not exist, and [22] does not offer sufficient implementation details to develop a reliable in-house version. The experiments were performed using MATLAB, WEKA and ELKI software on a Windows 7 (64 bit) PC with 16 GB RAM and Intel i7 @3.40 GHz processor.

5.1 Datasets

We performed our experiments on the following datasets.

Synthetic datasets: Two synthetic datasets, each having $N = 100,000$ data points in $p = 1000$ dimensions, were constructed by drawing labeled samples from a mixture of $k = 3$ Gaussian distributions. GM1 is a well separated Gaussian mixture, while GM2 has overlapping Gaussian clusters. The properties of these synthetic datasets are given in Table 1.

TABLE 1: Properties of two synthetic datasets GM1 and GM2

Component	1	2	3
Means			
GM1	$(-6,-6,...,-6)_{1000}$	$(0,0,...,0)_{1000}$	$(6,6,...,6)_{1000}$
GM2	$(-2,-2,...,-2)_{1000}$	$(0,0,...,0)_{1000}$	$(2,2,...,2)_{1000}$
Standard deviations in all directions			
GM1	$(1,1,...,1)_{1000}$	$(2,2,...,2)_{1000}$	$(3,3,...,3)_{1000}$
GM2	$(1,1,...,1)_{1000}$	$(2,2,...,2)_{1000}$	$(3,3,...,3)_{1000}$

TABLE 2: Properties of real datasets

Dataset	N	p	k	$DI(k,U_{gt})$
US Census 1990	2458285	68	Unknown	Unknown
KDD CUP'99	4898431	41	23	0 (Non-CS)
FOREST	581012	54	7	0.002 (Non-CS)
MiniBooNE	130064	50	2	0 (Non-CS)
MNIST	70000	784	10	0.15 (Non-CS)
ACT	9162	5625	19	0.01 (Non-CS)

Real datasets: Six publicly available real, high-dimensional (large volumes) datasets were chosen to demonstrate the applicability of our approach. The details of all real datasets[1] are given in Table 2. All datasets are labeled except the US Census 1990 dataset. We point out that the labeled subsets in these data sets may or may not correspond to computationally identifiable sets of clusters.

5.2 Evaluation Criteria

Partition Accuracy

For all datasets, except US Census 1990, the quality of the output crisp partition obtained by various clustering algorithms is assessed using ground truth information, U_{gt}. The similarity of computed partitions with respect to ground truth labels is measured using the *partition accuracy* (PA). The PA of a clustering algorithm is the ratio of the number of samples with matching ground truth and algorithmic labels to the total number of samples in the dataset. The value of PA ranges from 0 to 1, and a higher value implies a better match to the ground truth partition.

Dunn's Index

Since, the ground truth information is not available for US Census 1990 dataset, we use an internal cluster validity index, *Dunn's Index* (DI) [37], to evaluate the quality of output partitions for all clustering algorithms for this dataset. DI is a metric of how well a set of clusters represent *compact separated* (CS) clusters. DI for a partition U, is defined as:

$$DI(k,U) = \frac{min_{1\le i,j\le k\ i\neq j}\ \text{dist}(C_i,C_j)}{max_{1\le l\le k}\ \text{diam}(C_l)}, \tag{3}$$

1. These datasets can be found at the UCI machine learning data repository [43] and [44]. The features are normalized to the interval [0,1] by subtracting the minimum and then dividing by the subsequent maximum so that they all had the same scale.

where C_i is the ith cluster, $\text{dist}(C_i, C_j)$ is the distance between two clusters, and $\text{diam}(C_l)$ is the cluster diameter (maximum distance within a cluster). The diameter and distance functions are:

$$\text{dist}(C_i, C_j) = min_{o_i \in C_i, o_j \in C_j} \; D_{N_{ij}} \tag{4}$$

$$\text{diam}(C_l) = max_{o_i, o_j \in C_l} \; D_{N_{ij}}, \tag{5}$$

where $D_{N_{ij}}$ is the ijth element of D_N. Dunn defined CS clusters in X with a distance criteria, and showed that X contains CS clusters if and only there is a partition U^* of X for which $DI(k, U^*) > 1$. Havens *et al.* [45] related the effectiveness of VAT in showing cluster tendency to DI. The sVAT-SL [29] partition is equivalent to the SL partition for CS datasets. Since, the recursive version of iVAT [24] is used in our FensiVAT algorithm, the same rule applies to FensiVAT. If a dataset does not contain k-CS clusters, then FensiVAT is not guaranteed to find the same partition as SL. However, we show in our comparison experiments that FensiVAT produces good approximation for large datasets whether they are CS or not. The DI of the ground truth partition for all real data sets is shown in Table 2.

Chi-square distance

The similarity between two cluster distributions (histograms) A and B can be compared using the chi-square distance [46], $\chi^2(A, B)$, as follows

$$\chi^2(A, B) = \frac{1}{2} \sum_{i=1}^{f} \frac{(A_i - B_i)^2}{A_i + B_i}, \tag{6}$$

where f is the number of bins in histograms of the data. We take $f = k \in \mathbb{Z}$ as the number of bins. The value of χ^2 is in $[0, \infty]$, and a lower value implies higher similarity between two distributions.

Run-time

We also report the run-time (in seconds), another important criteria for comparison, which is related to the scalability of an algorithm.

5.3 Parameter settings

In all the experiments, FensiVAT and clusiVAT parameters, k' and n are randomly chosen between $2k$ and $4k$, and $10k$ and $30k$ respectively, where $k', n \in \mathbb{Z}$, and k is the number of labeled subsets in the ground truth data. The number of random projections Q in the ensemble step of our FensiVAT algorithm is chosen as 5, unless stated otherwise. For MBKM, the parameter batch size = 50, the iteration limit = 100, and the termination threshold = 0.001. The initial centroids for MBKM were built using 'kmeans++' method to speed-up convergence. For CLARA, the number of samples was 5 and sample size was $40 + 2k$ [8]. For spkm, n is 10% of N. The k-means++ seeding technique was used to choose k initial centroids in CLARA. For CURE, the number of representative (well-scattered) points c in clusters is 5, shrink factor α is 0.7, and the number of (random) samples is kept the same as n in FensiVAT. For PROCLUS, the average dimensionality of clusters, x_d, were chosen based on the grid search for best clustering performance. For the ensemble clustering method, RP-EN, we chose the number of random projections as 20, the weighting exponent as 2, termination threshold as 0.000001, and the iteration limit as 100. For FastSpec, we used $r = 300k$ for all datasets except KDD Cup, US Census, and FOREST. FastSpec [21] has very high space complexity, so we could not run it on our PC for datasets using $r = 300k$ [21] with very big N and k such as KDD Cup, US Census, and Forest dataset, so however, ran it with $r = 100k$ for FOREST dataset, and $r = 10k$ for KDD and US datasets, and using sparse MATLAB function to store diagonal matrix. The downspace dimensions for RP-EN and FastSpec were the same as those chosen for FensiVAT. The authors of [27] kindly provided us the GARDEN k-means code, written in C++. The density threshold in GARDENkm was chosen based on the best performance, for each dataset. All the experiments were performed 20 times on each dataset except KDD (5 times) and the average results are reported.

Distance Metric: The original JL lemma [32] as well as Dunn's index [37], [45] assume that the data are in an arbitrary metric space, so any distance metric will work in Theorem 1 and Proposition 1. In MM sampling, no matter what distance metric is used, the distinguished object selected will satisfy the Maximin rule in the sense of the chosen distance metric. The VAT/iVAT algorithms also work with arbitrary distance measures. Moreover, they have been shown to work well with dissimilarity measures that are not even metrics [47].

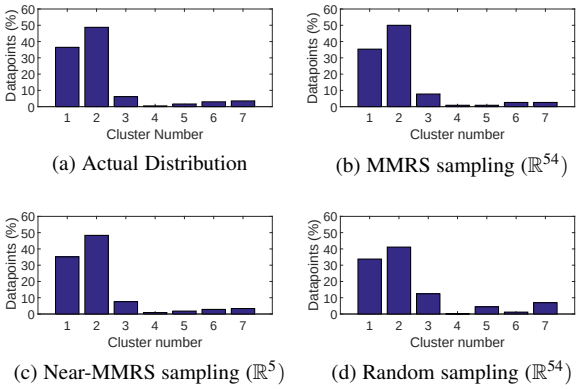
If a metric is Euclidean, the JL embedding can be done with an ε- distortion for every ε. For other metrics, the distortion is usually worse than the Euclidean, often with a constant or logarithmic distortion. In [48], random projection was used with the Hamming distance, which weakly preserves the distance in lower dimension space. The approximate distance preservation for higher order distances with weaker bounds using random projection is presented in [49]. So, Euclidean distance is used exclusively in this paper to ensure that Near-MM samples suffer minimum distortion.

5.4 Cluster Distribution using Various Sampling Schemes

In this experiment, we compare the cluster distribution in samples, obtained using three sampling schemes viz., random, MMRS, and Near-MMRS sampling for four datasets GM2, MNIST, ACT, and FOREST, which have different numbers of labeled subsets (k) and cluster distributions. First, for each sample, obtained from a sampling scheme, a histogram is computed using the label distribution in that sample in $\{1, 2, .., k\}$ integer bins. Then, the similarity of each cluster distribution is computed with respect to the actual (ground truth) distribution in the full dataset using chi-square distance.

The average distribution of datapoints in samples obtained using the three sampling schemes for FOREST dataset are shown in Fig. 2. The distribution of datapoints using MMRS and Near-MMRS sampling are very similar to each other, and to the actual distribution in the data, whereas, in random sampling, subsets 3, 5 and 7 are oversampled, while subsets 4 and 6 are undersampled. MMRS and Near-MMRS sampling both acquired at least one datapoint from each subset in every trial. On the other hand, random sampling did not select any data points from subset 4 on 4/10 trials (not shown in Fig. 2).

Table 3 shows the run-time and average (20 trials) chi-square values between cluster distributions for full and sampled datasets for each sampling scheme. The number of samples n for each sampling scheme, the number of distinguished objects k' for MMRS and Near-MMRS sampling scheme, and the downspace dimension q for Near-MMRS sampling scheme are also shown. The values in Table 3 show that the chi-square values for MMRS and Near-MMRS sampling differ from each other by either 0.01 or 0.02,



(a) Actual Distribution (b) MMRS sampling ($\mathbb{R}^{54}$)

(c) Near-MMRS sampling ($\mathbb{R}^{5}$) (d) Random sampling ($\mathbb{R}^{54}$)

Fig. 2: Histogram of data in the Forest Dataset. The MMRS and Near-MMRS parameters are $k' = 30$, and $n = 100$ samples, and $q = 5$ (for Near-MMRS).

TABLE 3: Average (20 trials) chi-square values and run-time (seconds) for each sampling scheme

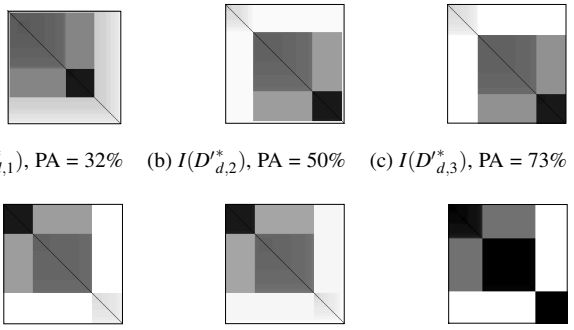
Dataset	Random		MMRS		Near-MMRS	
	χ^2	Time	χ^2	Time	χ^2	Time
GM2 ($n = 200, k' = 10, q = 50$)	0.3	0.00	0.06	20.5	0.05	0.8
MNIST ($n = 300, k' = 30, q = 100$)	1.25	0.00	0.59	25.2	0.60	1.2
ACT ($n = 100, k' = 30, q = 50$)	6.71	0.01	4.98	115	4.50	0.2
FOREST ($n = 100, k' = 30, q = 5$)	1.86	0.01	1.18	4.4	1.19	0.9

so these two methods yield essentially the same samples, which match the distribution quite well. The random sampling scheme has much higher χ^2 values, indicating a poorer match to the full distribution. Our experimental results indicate that Near-MMRS sampling accurately portrays the distribution of the original data in randomly projected lower dimensions, and takes significantly lesser time (around a second) than the MMRS sampling scheme.

5.5 Single Random Projection vs Ensemble RP for iVAT Image

In this experiment, we compare the quality of iVAT images obtained by applying VAT/iVAT to distance matrices $\{D_{d,i}\}_{i=1}^{Q}$, for $Q = 5$, computed from single random projections to the iVAT image of the distance matrix $D_{n,d}$, computed from our ensemble of multiple random projections. We also compare the PA values of NOPR partitions U, obtained by SL partitioning based on the VAT reordered distance matrices $\{D_{d,i}\}_{i=1}^{Q}$ and $D_{n,d}$.

Figs. 3 (a)-(e) show the iVAT images, $\{I(D'^{*}_{d,i})\}_{i=1}^{Q}$, obtained using single random distance matrices $\{D_{d,i}\}_{i=1}^{Q}$, for the GM2 dataset, which has three (true) clusters. It is clear from these five iVAT images and corresponding PA values that the qualities of these images vary due to random nature of RP, and none of them strongly suggests that actual number of clusters in GM2 is $k = 3$. Fig. 3 (f) shows the iVAT image $I(D'^{*}_{n,d})$ based on the ensemble distance matrix $D_{n,d}$, which is obtained by aggregating the five distance matrices, $\{D_{d,i}\}_{i=1}^{Q}$ using our ensemble scheme. View 3 (f) contains three dark blocks that are clearly visible along the diagonal in this image, and the PA value corresponding to this image is nearly perfect (99.6%).



(a) $I(D'^{*}_{d,1})$, PA = 32% (b) $I(D'^{*}_{d,2})$, PA = 50% (c) $I(D'^{*}_{d,3})$, PA = 73%

(d) $I(D'^{*}_{d,4})$, PA = 74% (e) $I(D'^{*}_{d,5})$, PA = 51% (f) $I(D'^{*}_{n,d})$, PA = 99.6%

Fig. 3: iVAT images obtained using single distance matrices (a-e) and ensemble distance matrix (f).

TABLE 4: Average PA (%) values (20 trials) using single distance matrices, $\{D_{d,i}\}_{i=1}^{Q=5}$ and ensemble distance matrix $D_{n,d}$ for VAT/iVAT in FensiVAT.

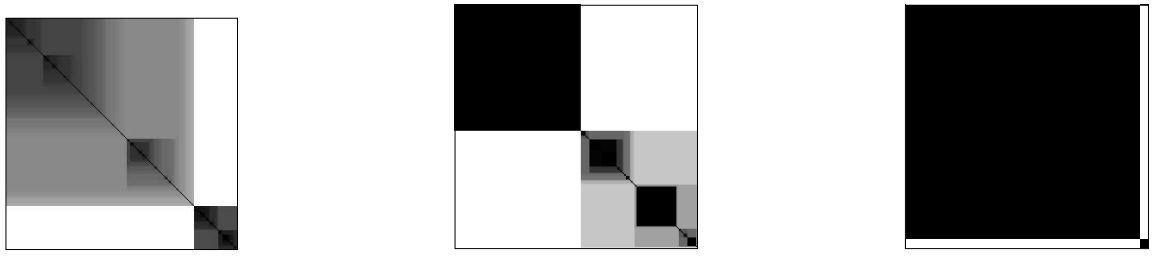
Distance Matrix	$D_{d,1}$	$D_{d,2}$	$D_{d,3}$	$D_{d,4}$	$D_{d,5}$	$D_{n,d}$
GM1 ($q = 20, k' = 10, n = 200$)	86	99	100	100	89	100
GM2 ($q = 50, k' = 10, n = 200$)	32	50	73	74	51	99.6

Table 4 show the PA values for the GM1 and GM2 datasets. The PA values corresponding to individual distance matrix for GM1 dataset show better accuracy than those obtained for the GM2 dataset. This is because the clusters in GM1 are much more separated that in the GM2 data, which has overlapping clusters. In contrast, the PA value corresponding to ensemble distance matrix is almost perfect for both datasets, which demonstrate the effectiveness of our ensemble scheme to obtain accurate NOPR partitions with the ensemble approach.

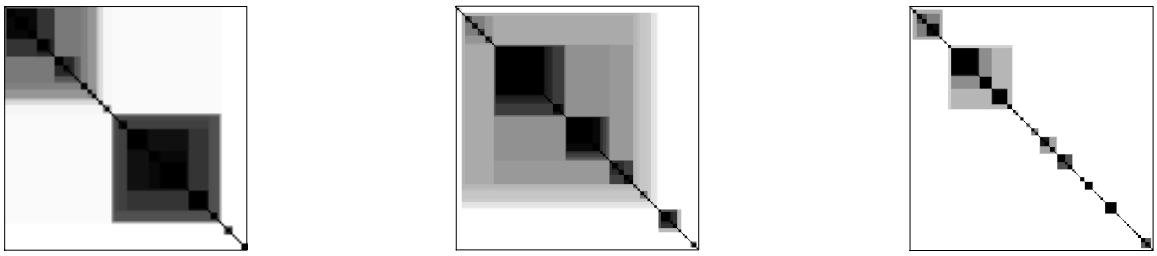
5.6 Cluster Assessment

The FensiVAT algorithm can be used to assess the potential number of clusters present in large, high dimensional data in significantly less time (discussed in Section 5.4) than clusiVAT, and with similar iVAT image quality. In this experiment, we compare iVAT images obtained using clusiVAT and FensiVAT for GM1 and GM2, and then we will showcase the ability of FensiVAT to correctly estimate the number of labeled subsets for the real datasets.

The iVAT images obtained using clusiVAT and FensiVAT for GM1 and GM2 are shown in Fig. 4 with corresponding algorithm parameter values. The ground truth partition of GM1 dataset has CS clusters because its DI $= 1.26$ (> 1). Figs. 4 (a) and (b) show that both clusiVAT and FensiVAT exhibit three (well-separated) dark blocks along the diagonal, suggesting that $k = 3$ for GM1. The ground truth partition for GM2 is non-CS because its DI is 0.66 (< 1). Figs. 4 (c) and (d) show that FensiVAT produces three dark blocks along the diagonal for GM2, whereas clusiVAT shows three light blocks including many tiny blocks (datapoints) along the diagonal. Both views show the two darker blocks superimposed on a lighter dark block, which indicates that this data has a high degree of overlap. While clusiVAT and FensiVAT both show three blocks for GM1 and GM2, FensiVAT provides the more convincing assessment because of the sharper

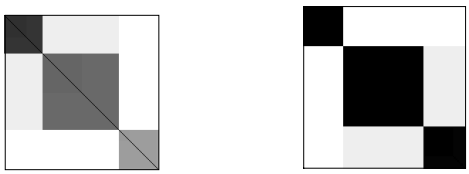


(a) Census, $k' = 10$, $n = 100$, $q = 5$, Time$= 2.9s$ (b) KDD, $k' = 47$, $n = 322$, $q = 5$, Time$= 14.5s$ (c) MiniBooNE, $k' = 5$, $n = 30$, $q = 5$, Time$= 0.05s$

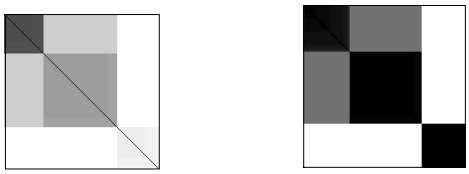


(d) FOREST, $k' = 20$, $n = 108$, $q = 5$, Time$= 0.7s$ (e) MNIST, $k' = 28$, $n = 313$, $q = 100$, Time$= 1.5s$ (f) ACT, $k' = 38$, $n = 314$, $q = 50$, Time$= 0.5s$

Fig. 5: iVAT images of $D'^{*}_{n,d}$ for each of the datasets obtained by FensiVAT algorithm.



(a) GM1, ClusiVAT (Time=20.1s) (b) GM1. FensiVAT (Time=0.35s)



(c) GM2, ClusiVAT (Time=21.3s) (d) GM2, FensiVAT (Time=0.84s)

Fig. 4: ClusiVAT (a) and (c), and FensiVAT images (b) and (d) for GM1 and GM2. The parameters are $k' = 9$, $n = 205$ for GM1 and $k' = 12$, $n = 206$ for GM2 dataset. The downspace dimensions for FensiVAT are $q = 20$ for GM1 and $q = 50$ GM2.

contrast between diagonal blocks and the background. Moreover, FensiVAT takes only a fraction of second for both datasets, whereas, clusiVAT takes around 20s to obtain poorer quality iVAT images. The sizes of the diagonal blocks in all four images show the relative size of each cluster accurately, which supports our claim that Near-MMRS sampling replicates (approximately) the same cluster distribution in the sample as the MMRS sampling used by clusiVAT.

The iVAT images of $D'^{*}_{n,d}$ for six real datasets are shown in Fig. 5 with corresponding FensiVAT algorithm parameter values. A (large) zoom is required to see all tiny dark blocks. The US Census 1990 data is an example of a real-world unlabeled, big data that is very large in both the number of records (N) and the number of attributes (p). Fig. 5 (a) shows the FensiVAT image for the US Census 1990 data. It can be seen that it shows two distinguished dark blocks along the diagonal in which the lower dark block comprises two small dark blocks. This suggests that there are two or three clusters in this data. Several previous researches [50], [51] also suggest $k = 2$ or 3 as the best estimate of the number of clusters for this dataset. FensiVAT just takes approximately 3 seconds to make this estimate.

The KDD CUP'99 is a big, labeled dataset that specifies attack types (normal or attack). It has 23 labeled subsets (22 simulated attacks and a normal subset), that fall into four main categories: DOS, R2L, U2R, and probing. The FensiVAT image of KDD-99 in View 5 (b) suggests 4 primary dark blocks and $18-20$ tiny dark blocks. The top left big dark block represents the 'smurf' attack (60% of the total dataset) in the DOS category. The right bottom dark block represents 'normal' data, which comprises approximately 18% of the data, and the middle dark block represents the 'neptune' attack in the DOS category, which comprises approximately 20% of the data. The remaining attacks are represented by $18-20$ tiny (hard to see) dark blocks along the diagonal.

The MiniBooNE data consists of $N = 130064$ instances divided into $36,499$ signal events of electron neutrinos and $93,565$ background events of muon neutrinos. The FensiVAT image (Fig. 5 (c)) for the MiniBooNE dataset shows two dark blocks along the diagonal. Although it shows two blocks, their sizes are not relative to the actual number of points in both classes. This is because some of the signal events are grouped with the background events due to similar attribute values. The FensiVAT image (Fig. 5 (d)) for Forest dataset shows 2 big dark blocks on low resolution, $6-7$ dark blocks (of moderate size) in medium resolution, and $14-15$ tiny dark blocks on high-resolutions. The Forest dataset has overlapping clusters due to heterogeneous features, so it has inter-mixed dark blocks along the diagonal. This

is a case where the physically labeled subsets do NOT form well-defined clusters, at least not in the sense of SL distance, the basis of the iVAT image.

The MNIST dataset is a big, fairly dimensional ($p = 784$) dataset. This is also a challenging dataset for clustering because handwritten images of a single character can be executed in many often quite different ways, which causes overlapping clusters in the data. Fig. 5 (e) shows the FensiVAT image for the MNIST dataset, which indicates $10-12$ dark blocks. The ACT dataset is a high-dimensional ($p = 5625$), time-series dataset, which contains 19 activity types such as sitting, walking, jumping etc. The FensiVAT image (Fig. 5 (f)) shows $18-24$ tiny and middle size dark blocks along the diagonal. Thus, the FensiVAT recommendation for this data set is to cluster it at every k from 18 to 24, and use a post clustering validation method to select the "best" partition of the data. The MNIST and ACT datasets contain inter-mixed clusters, so we removed outliers using the strategy mentioned in Section 4.3 to improve the quality of the FensiVAT images. Finally, please note that FensiVAT takes only about a second for most of the datasets (14.5s maximum for KDD) to provide visual evidence about potential cluster structure, which makes it one of the best cluster assessment tools for big, high-dimensional dataset.

5.7 Synthetic Dataset for Different Numbers of RPs in Ensemble Step

In this experiment, we compare the PA of FensiVAT algorithm for different number of RPs (or distance matrices) used in the ensemble step in our FensiVAT algorithm. For datasets having high diversity (overlapping clusters) like GM2, increasing Q in the ensemble method may be beneficial because there will probably be much more diversity in the random projections due to the mixed clusters in the upspace. Table 5 shows the average (20 trials) PA values with standard deviation and run-time of FensiVAT for $Q = 2, 3, 5, 10, 20$, and 30, for a fixed value of q for GM1 ($q = 20$) and GM2 ($q = 50$). FensiVAT achieves 100% accuracy for all $Q \geq 3$ on GM1, and for all $Q > 5$ on GM2, respectively. As expected, the accuracy of FensiVAT for GM2 increases as Q increases. Furthermore, increasing the ensemble size has very little effect on FensiVAT CPU time.

TABLE 5: Average (20 trials) PA (%) values (with standard deviation) and run-time (in seconds) of FensiVAT for different RPs Q in ensemble step

	$Q=2$		$Q=3$		$Q=5$	
	PA	Time	PA	Time	PA	Time
GM1	99.9 ± 0.04	1.01	100 ± 0	1.11	100 ± 0	1.14
GM2	97 ± 5.8	1.68	99 ± 0.74	1.72	99.99 ± 0.1	1.89
	$Q=10$		$Q=20$		$Q=30$	
GM1	100 ± 0	1.38	100 ± 0	1.75	100 ± 0	2.11
GM2	100 ± 0	1.95	100 ± 0	2.33	100 ± 0	2.62

5.8 Effect of Different Downspace Dimesions, q

In this experiment, we compare the performance of FensiVAT for different downspace dimensions $q = 5, 10, 20, 30, 50, 100$ for synthetic datasets GM1 and GM2. For the choices of $\varepsilon = \beta = 0.25$, and $n = 9162$, $q_0 = 1576$ (using (2)) and the probability of distance preservation = 0.9, so the chosen q values are well below the JL bound. These q values correspond to rogue random projections, which are chosen irrespective of ε and β. Table 6 shows the average (20 trials) PA values with standard deviation and run-time of FensiVAT for a fixed value of Q ($= 5$). As expected, the accuracy of FensiVAT increases with increasing q and the performance becomes more stable (as standard deviation decreases). This is because higher q's correspond to more dimensions, so there is a better chance to preserve distances and lose less information under the projection.

TABLE 6: Average (20 trials) PA (%) values (with standard deviation) and run-time (in seconds) of FensiVAT for different downspace dimensions, q.

	$q=5$		$q=10$		$q=20$	
	PA	Time	PA	Time	PA	Time
GM1	99.8 ± 0.2	0.80	99.9 ± 0.1	0.94	100 ± 0	1.11
GM2	92.4 ± 8.9	0.85	98.1 ± 1.2	0.96	99.2 ± 0.4	1.14
	$q=30$		$q=50$		$q=100$	
GM1	100 ± 0	1.33	100 ± 0	1.61	100 ± 0	2.51
GM2	99.9 ± 0.1	1.21	100 ± 0.0	1.53	100 ± 0.0	2.52

The values in Table 6 show that even at $q = 5$ downspace dimensions, FensiVAT achieves very good clustering results (PA$>$ 99%) and achieves perfect (PA$=$ 100%) results with $q \geq 30$ for GM1. This is because the clusters in this dataset are (probably) well separated (recall that the ground truth partition of GM1 has CS clusters in the sense of Dunn). Thus, FensiVAT takes a fraction of a second to achieve perfect accuracy for the big, high-dimensional data GM1. For GM2, FensiVAT achieves near perfect results with $q \geq 50$. Unlike GM1, the performance of FensiVAT for GM2 is unstable for $q < 30$, most likely due to overlapping clusters in GM2 in the input space. Overall, FensiVAT achieves very good and stable clustering solutions even with rogue random projections.

5.9 Comparison of Different Cluster Ensemble Methods

In this last experiment, we compare the performance of our approach with nine existing approaches, which are best known for big and/or high-dimensional data clustering. The downspace dimensions for FensiVAT are chosen based on its best performance for each dataset. These values are shown in Figs. 4 and 5 for each dataset. Table 7 shows the comparison of FensiVAT to nine other algorithm based on the accuracy (PA) and run-time. Since, US Census 1990 dataset is not labeled, we use DI as a measure of accuracy for different algorithms on the census data. The highest accuracy and smallest CPU time are shown in bold for each data set.

For GM1, GM2, and ACT, FensiVAT outperforms the other approaches in terms of accuracy and CPU time. For GM2, FensiVAT, CLARA, CURE and RP-EN achieve perfect results (ave. PA= 100%), however, CLARA, CURE, and RP-EN take 16.6s, 511.9s and 183.9s respectively, whereas FensiVAT just takes 1.7s. For the FOREST, FensiVAT and clusiVAT achieve the highest PA (48.9%), however, FensiVAT takes the least time (2.17s). For MNIST, clusiVAT achieves the highest PA (50.1%) in 25.3s, whereas FensiVAT just takes 2.6s to achieve (nearly) similar accuracy (50.0%). For KDD, FensiVAT and clusiVAT achieve the highest accuracy, but FensiVAT is about 10 times faster than clusiVAT.

For GM1, GM2, MNIST, and ACT data, which have relatively high dimensions, FensiVAT outperforms the other clustering methods. It achieves the highest PA values in less than 2.6s (maximum

TABLE 7: Average PA (%) values (DI for US Census) and run-time (in seconds) for all the approaches on all the datasets.

Dataset /	GM1		GM2		KDD		FOREST		MiniBooNE		US Census		MNIST		ACT	
Methods	PA	Time	PA	Time	PA	Time	PA	Time	PA	Time	DI	Time	PA	Time	PA	Time
FensiVAT	**100**	**1.12**	**100**	**1.72**	**96.1**	88.4	**48.9**	2.17	71.9	**0.12**	0.10	6.9	50.0	**2.6**	**49.5**	**1.2**
clusiVAT	100	22.3	75.6	24.9	96.1	798.8	48.9	5.39	71.9	0.38	0.08	21	**50.1**	25.3	49.2	123.7
MBKM	90.1	2.07	89.9	2.03	74.3	**33.2**	34.2	**1.75**	71.9	0.12	0.08	**3.12**	44.3	3.38	47.8	7.8
CLARA	100	16.7	100	16.6	73.9	223.4	37.2	10.30	71.9	0.94	0.07	31.8	37.5	19.94	45.7	44.1
spkm	100	39.5	95.8	37.4	78.6	147.6	45.3	53.3	64.6	1.97	**0.12**	33	19.8	2296.8	12.5	3315.6
CURE	100	505.2	**100**	511.9	95.2	828.8	44.7	17.6	**76.8**	11.36	0.12	270	18.5	78.2	19.8	1871.3
RP-EN	100	31.8	100	183.9	95.7	52584	45.4	1596.3	76.8	30.3	0.12	475	26.5	95.8	26.5	205.9
PROCLUS	79.3	19.8	75.5	23.3	94.5	14346	45.1	2901.3	71.9	36.7	0.02	9162355	17.9	1185.7	17.8	12479.5
GARDENkm	65.8	1564	51.3	1652	94.1	326	38.2	44.5	71.9	340	0.01	3617	17.8	1133	18.5	3458
FastSpec	100	30.3	100	31.2	71.8	66.38	42.4	86	65.7	14.5	0.06	420	33.5	68.4	45.2	21.7

2.6s for MNIST), whereas, clusiVAT takes more than 20s for GM1 and MNIST, and hundreds of seconds for ACT. For all datasets except GM2, clusiVAT and FensiVAT achieve approximately equal PA values, but clusiVAT is $5-100$ times slower than FensiVAT. Surprisingly, FensiVAT achieves perfect results for GM2 dataset, whereas clusiVAT achieves 75.6%. This is probably because of the robust distance matrix obtained using our ensemble scheme.

For the US Census data, spkm, CURE, and RP-EN achieve the highest DI value (0.12), which implies that their partitions are very slightly superior with regard to Dunn's validity measure. FensiVAT achieves the second highest DI value (0.1) and takes only 6.9s. ClusiVAT and MBKM achieves similar DI value, with MBKM the fastest algorithm. For KDD, which has millions of samples, FensiVAT achieves the best accuracy (96.1%) in just 88.4s, whereas the other approaches (except MBKM) take up to about $52,000$s average CPU time.

The MBKM approach is the second fastest method for all datasets except US Census, KDD, and FOREST (fastest), but, at the cost of lower clustering accuracy. CLARA is able to achieve the best PA for GM1 and GM2 dataset, but it is $10-60$ times slower than FensiVAT. The spkm algorithm achieves good accuracy for GM1, GM2, US Census, and FOREST, but performs poorly on MiniBoone, MNIST and ACT. It is approximately $30-50$ times slower than FensiVAT for GM1, GM2, KDD, and FOREST dataset, and $1500-6000$ times slower for the high-dimensional datasets MNIST and ACT. CURE achieves comparable accuracy on all datasets except MNIST and ACT. It is likely that with many clusters, the randomly drawn samples used by CURE do not adequately capture the geometry of the big data, therefore it suffers for the ACT and MNIST data, which appear to have many clusters. CURE is approximately $20-500$ times slower than FensiVAT for GM1, GM2, MNIST, and FOREST, and 3500 times slower for ACT dataset. PROCLUS is inaccurate for all datasets except KDD. The ensemble clustering method, RP-EN, achieves its highest PA values for GM1, GM2, and MiniBooNE and highest DI for US Census dataset, but at a time cost that is about $100-1500$ times higher than FensiVAT. RP-EN becomes intractable for KDD, and takes 52584s. Among high-dimensional clustering algorithms, RP-EN outperforms PROCLUS for all datasets except KDD. FensiVAT is about $4-3000$ times faster than GARDENkm. GARDENkm is inaccurate for all the datasets except for KDD and MiniBooNE. FastSpec performs better than GARDENkm based on the clustering accuracy and CPU time. FastSpec achieves perfect results for GM1 and GM2, and exhibits comparable accuracy for all other datasets except KDD and MNIST. FensiVAT is faster ($20-120$ times) than FastSpec for all datasets except KDD. FastSpec is little faster than FensiVAT on KDD, but at the cost of clustering accuracy. To summarize, FensiVAT seems superior to the nine comparison algorithms used in this paper.

6 Conclusions

This paper introduces a new, fast clustering algorithm, called FensiVAT, which can be used to cluster large volumes of high-dimensional data. FensiVAT integrates a new random projection-based distance matrix ensemble method with Maximin and Random sampling (MMRS) and a visual assessment of cluster tendency method. We showed that the samples obtained using MMRS sampling in the downspace dimension (Near-MMRS sampling) retain the same geometry in the downspace as samples in the upspace. This enables us to use random projection effectively with MMRS sampling and in our ensemble method to reduce the computation time.

We demonstrated the superiority of our FensiVAT approach by comparing it with nine state-of-the-art approaches on two Gaussian mixture datasets and six real datasets which have both large sample size and high dimensions. Our experimental results on eight large, high-dimensional datasets show that FensiVAT almost always outperforms the other nine approaches. FensiVAT is an order of magnitude faster than clusiVAT, and several order of magnitudes faster than the other nine approaches (except MBKM), without compromising accuracy.

References

[1] C. C. Aggarwal and C. K. Reddy, *Data clustering: algorithms and applications*. CRC Press, 2013.

[2] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: applications to image and text data," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2001, pp. 245–250.

[3] A. McCallum, K. Nigam, and L. H. Ungar, "Efficient clustering of high-dimensional data sets with application to reference matching," in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2000, pp. 169–178.

[4] A. S. Shirkhorshidi, S. Aghabozorgi, T. Y. Wah, and T. Herawan, "Big data clustering: a review," in *International Conference on Computational Science and Its Applications*. Springer, 2014, pp. 707–720.

[5] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Y. Zomaya, S. Foufou, and A. Bouras, "A survey of clustering algorithms for big data: Taxonomy and empirical analysis," *IEEE transactions on emerging topics in computing*, vol. 2, no. 3, pp. 267–279, 2014.

[6] P. S. Bradley, U. M. Fayyad, C. Reina *et al.*, "Scaling clustering algorithms to large databases." in *KDD*, 1998, pp. 9–15.

[7] D. Sculley, "Web-scale k-means clustering," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 1177–1178.

[8] P. J. Rousseeuw and L. Kaufman, *Finding Groups in Data*. Wiley Online Library, 1990.

[9] S. Guha, R. Rastogi, and K. Shim, "Cure: an efficient clustering algorithm for large databases," *Information Systems*, vol. 26, no. 1, pp. 35–58, 2001.

[10] D. Kumar, J. C. Bezdek, M. Palaniswami, S. Rajasegarar, C. Leckie, and T. C. Havens, "A hybrid approach to clustering in big data," *IEEE transactions on cybernetics*, vol. 46, no. 10, pp. 2372–2385, 2016.

[11] J. C. Bezdek, *Primer on Cluster Analysis: Four Basic Methods that (Usually) Work*. First Edition Design Publishing, 2017, vol. 1.

[12] H.-P. Kriegel, P. Kröger, and A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 3, no. 1, p. 1, 2009.

[13] I. Assent, "Clustering high dimensional data," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 4, pp. 340–350, 2012.

[14] M. Steinbach, L. Ertöz, and V. Kumar, "The challenges of clustering high dimensional data," in *New directions in statistical physics*. Springer, 2004, pp. 273–309.

[15] T. Urruty, C. Djeraba, and D. A. Simovici, "Clustering by random projections," in *Industrial Conference on Data Mining*. Springer, 2007, pp. 107–119.

[16] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data: a review," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 90–105, 2004.

[17] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data," *Data Mining and Knowledge Discovery*, vol. 11, no. 1, pp. 5–33, 2005.

[18] C. C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, and J. S. Park, "Fast algorithms for projected clustering," in *ACM SIGMoD Record*, vol. 28, no. 2. ACM, 1999, pp. 61–72.

[19] X. Z. Fern and C. E. Brodley, "Random projection for high dimensional data clustering: A cluster ensemble approach," in *Proceedings of International Conference on Machine Learning (ICML)*, vol. 3, 2003, pp. 186–193.

[20] M. Popescu, J. Keller, J. Bezdek, and A. Zare, "Random projections fuzzy c-means (rpfcm) for big data clustering," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2015, pp. 1–6.

[21] T. Sakai and A. Imiya, "Fast spectral clustering with random projection and sampling," in *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer, 2009, pp. 372–384.

[22] B. L. Milenova and M. M. Campos, "O-cluster: Scalable clustering of large high dimensional data sets," in *IEEE International Conference on Data Mining (ICDM)*. IEEE, 2002, pp. 290–297.

[23] S. Gilpin, B. Qian, and I. Davidson, "Efficient hierarchical clustering of large high dimensional datasets," in *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. ACM, 2013, pp. 1371–1380.

[24] T. C. Havens and J. C. Bezdek, "An efficient formulation of the improved visual assessment of cluster tendency (ivat) algorithm," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 5, pp. 813–822, 2012.

[25] R. L. Thorndike, "Who belongs in the family?" *Psychometrika*, vol. 18, no. 4, pp. 267–276, 1953.

[26] R. J. Hathaway, J. C. Bezdek, and J. M. Huband, "Scalable visual assessment of cluster tendency for large data sets," *Pattern Recognition*, vol. 39, no. 7, pp. 1315–1324, 2006.

[27] Y. Lai, R. Orlandic, W. G. Yee, and S. Kulkarni, "Scalable clustering for large high-dimensional data based on data summarization," in *IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*. IEEE, 2007, pp. 456–461.

[28] J. C. Bezdek and R. J. Hathaway, "VAT: A tool for visual assessment of (cluster) tendency," in *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, 2002, pp. 2225–2230.

[29] T. C. Havens, J. C. Bezdek, and M. Palaniswami, "Scalable single linkage hierarchical clustering for big data," in *IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*. IEEE, 2013, pp. 396–401.

[30] D. Kumar, M. Palaniswami, S. Rajasegarar, C. Leckie, J. C. Bezdek, and T. C. Havens, "clusivat: A mixed visual/numerical clustering algorithm for big data," in *IEEE International Conference on Big Data*. IEEE, 2013, pp. 112–117.

[31] H. Gunadi, "Comparing nearest neighbor algorithms in high-dimensional space," 2011.

[32] W. B. Johnson and J. Lindenstrauss, "Extensions of lipschitz mappings into a hilbert space," *Contemporary Mathematics*, vol. 26, no. 189-206, p. 1, 1984.

[33] D. Achlioptas, "Database-friendly random projections," in *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 2001, pp. 274–281.

[34] J. C. Bezdek, X. Ye, M. Popescu, J. Keller, and A. Zare, "Random projection below the JL limit," in *Proceedings of International Joint Conference on Neural Network (IJCNN)*, 2016, pp. 2414–2423.

[35] P. Hore, L. O. Hall, and D. B. Goldgof, "Single pass fuzzy c means," in *2007 IEEE International Fuzzy Systems Conference*. IEEE, 2007, pp. 1–7.

[36] E. J. Otoo, A. Shoshani, and S.-w. Hwang, "Clustering high dimensional massive scientific datasets," *Journal of Intelligent Information Systems*, vol. 17, no. 2-3, pp. 147–168, 2001.

[37] J. C. Dunn, "A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters," *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, 1973.

[38] L. Wang, C. Leckie, K. Ramamohanarao, and J. Bezdek, "Automatically determining the number of clusters in unlabeled data sets," *IEEE Transactions on knowledge and Data Engineering*, vol. 21, no. 3, pp. 335–350, 2009.

[39] I. J. Sledge, T. C. Havens, J. M. Huband, J. C. Bezdek, and J. M. Keller, "Finding the number of clusters in ordered dissimilarities," *Soft Computing*, vol. 13, no. 12, pp. 1125–1142, 2009.

[40] T. C. Havens, J. C. Bezdek, J. M. Keller, and M. Popescu, "Clustering in ordered dissimilarity data," *International Journal of Intelligent Systems*, vol. 24, no. 5, pp. 504–528, 2009.

[41] T. C. Havens, J. C. Bezdek, J. M. Keller, M. Popescu, and J. M. Huband, "Is vat really single linkage in disguise?" *Annals of Mathematics and Artificial Intelligence*, vol. 55, no. 3, pp. 237–251, 2009.

[42] Y. S. Ahmed, *Multiple random projection for fast, approximate nearest neighbor search in high dimensions*. University of Toronto, 2004.

[43] A. Asuncion and D. Newman, "Uci machine learning repository," 2007.

[44] Y. LeCun, C. Cortes, and C. J. Burges, "The mnist dataset of handwritten digits," *URL http://yann. lecun. com/exdb/mnist*, 1998.

[45] T. C. Havens, J. C. Bezdek, J. M. Keller, and M. Popescu, "Dunn's cluster validity index as a contrast measure of vat images," in *19th International Conference on Pattern Recognition (ICPR)*. IEEE, 2008, pp. 1–4.

[46] O. Pele and M. Werman, "The quadratic-chi histogram distance family," in *European conference on computer vision*. Springer, 2010, pp. 749–762.

[47] S. Mahallati, J. C. Bezdek, D. Kumar, M. R. Popovic, and T. A. Valiante, "Interpreting cluster structure in waveform data with visual assessment and dunn's index," in *Frontiers in Computational Intelligence*. Springer, 2018, pp. 73–101.

[48] S. Arca, A. Bertoni, and G. Lipori, "Random projections weakly preserving the hamming distance between words," in *Italian Workshop on Neural Networks: WIRN*. IOS Press, 2009.

[49] P. Li, M. W. Mahoney, and Y. She, "Approximating higher-order distances using random projections," *arXiv preprint arXiv:1203.3492*, 2012.

[50] K. Chen and L. Liu, "Detecting the change of clustering structure in categorical data streams," in *Proceedings of the 2006 SIAM International Conference on Data Mining*. SIAM, 2006, pp. 504–508.

[51] ——, "ivibrate: Interactive visualization-based framework for clustering large datasets," *ACM Transactions on Information Systems (TOIS)*, vol. 24, no. 2, pp. 245–294, 2006.

Punit Rathore received the Master of Technology (M.Tech) in electrical engineering (Instrumentation) from the Indian Institute of Technology, Kharagpur, India in 2011. He has worked as Researcher in TATA Steel Limited, India for three and half years (2011-14).

Currently, he is pursuing the Ph.D. degree from Department of Electrical and Electronic Engineering, the University of Melbourne, Melbourne, Australia. His research interests include big data clustering, incremental clustering, spatio-temporal analytics, Internet of Things, machine learning, pattern recognition, and signal processing.

Dheeraj Kumar received the B.Tech. and M.Tech. dual degrees in electrical engineering from the Indian Institute of Technology, Kanpur, India in 2010, and Ph.D. degree from Department of Electrical and Electronic Engineering, the University of Melbourne, Melbourne, VIC, Australia in 2017.

Currently, he is a Postdoctoral research fellow in Lyles School of Engineering, Purdue University, USA. His current research interests include big data clustering, incremental clustering, spatio-temporal estimations, Internet of Things, machine learning, pattern recognition, and signal processing

James C. Bezdek (LF'10) received the PhD in Applied Math, Cornell University, 1973. Jim is past president of NAFIPS (North American Fuzzy Information Processing Society), IFSA (International Fuzzy Systems Association) and the IEEE CIS (Computational Intelligence Society as the NNC): founding editor the International Journal of Approximate Reasoning and the IEEE Transactions on Fuzzy Systems: Life fellow of the IEEE and IFSA; recipient of the IEEE 3rd Millennium, IEEE CIS Fuzzy Systems Pioneer, IEEE Frank Rosenblatt TFA and the Kempe de Feret IPMU awards. He retired in 2007. His research interests include optimization, pattern recognition, clustering in very large data, coclustering, and visual clustering.

Sutharshan Rajasegarar received his Ph.D. degree from the University of Melbourne, Melbourne, VIC, Australia, in 2009. He has worked as Research Fellow with the Department of Electrical and Electronic Engineering, the University of Melbourne and as a researcher in Machine Learning at National ICT Australia (NICTA). Currently, he is the lecturer in School of Information Technology, Deakin University, Geelong, Australia. His current research interests include anomaly/outlier detection, distributed machine learning, spatio-temporal estimations, pattern recognition, signal processing, Internet of Things, and wireless communication.

Marimuthu Palaniswami (F'12) received the M.E. degree in electrical, electronic and control engineering from the Indian Institute of Science, Bengaluru, India, the M.Eng.Sc. degree in electrical, electronic and control engineering from the University of Melbourne, Melbourne, VIC, Australia, and the Ph.D. degree from the University of Newcastle, N.S.W., Australia.

He is currently a Professor with the University of Melbourne. He is representing Australia as a core partner in EU FP7 projects such as SENSEI, SmartSantander, Internet of Things Initiative, and SocIoTal. He has been funded by several Australian Research Council (ARC) and industry grants (over 40 million) to conduct research in sensor network, Internet of Things (IoT), health, environmental, machine learning, and control areas. He has published over 400 refereed research papers, and leads one of the largest funded ARC Research Network on Intelligent Sensors, Sensor Networks and Information Processing Programme. His current research interests include SVMs, sensors and sensor networks, IoT, machine learning, neural network, pattern recognition, and signal processing and control.