Erasmus University Rotterdam
Erasmus School of Economics
Bachelor Thesis Econometrics and Operations Research

Navigating Complexity: Evaluating the Effectiveness of Dimension Reduction and Clustering Approaches on Complex Datasets

Ani Mkheidze (568785)

Supervisor:	Marina Khismatullina
Second assessor:	Name of your second assessor
Date final version:	2nd July 2023

Abstract

The research replicates and extends the paper - Renjith et al. (2021), where the central aim was to compare different Dimension Reduction (DR) techniques for clustering tasks. The research has added a promising technique Uniform Manifold Approximation Manifold (UMAP) to the comparison and found that it outperforms other techniques for the dataset presented in Renjith et al. (2021). In pursuit of understanding the strengths and weaknesses of DR techniques better, the research examines them by clustering various artificial datasets with complex geometrical structures. It has been found that UMAP outperformed the counterparts for all the artificial datasets, with the exception of the helictical structure.

1 Introduction

In the day and age of rapid technological advancements, it is important to understand an instrumental driving force of these innovations - the collection of high-dimensional data, and how to leverage it. A fundamental method for working with high-dimensional data, lowering complexity, and identifying essential information is Dimension Reduction (DR). DR techniques attempt to maintain important information while reducing information loss by converting high-dimensional data into lower-dimensional space.

Several different applications can be discussed for high-dimensional data, but this paper focuses on one of the crucial techniques in machine learning, clustering. In the setting of a clustering problem, the main goal is to collect related data points based on their innate patterns or traits. Domains involving segmentation, pattern recognition, and data exploration often employ clustering. The cluster's data points tend to be more similar to each other, than to those in different clusters. Hence clustering aims to find natural groupings or clusters within a dataset (Hastie et al., 2005).

Renjith et al. (2021) have conducted a comparative review of different DR techniques for Clustering applications. In this paper, their research will be replicated and extended. The previous research has examined two linear and non-linear techniques. As part of replication the best-performing non-linear technique, t-Distributed Stochastic Neighbor Embedding, and a popular linear technique, Principal Component Analysis (PCA) will be examined. To extend their research another non-linear technique, Uniform Manifold Approximation and Projection (UMAP) will be a part of the comparison. In this research, k-means and Agglomerative Nesting (AGNES) clustering algorithms will be used, which have been applied in Renjith et al. (2021).

Various DR techniques capture the relationships within data differently, thus the performance

of the DR technique is significantly dependent on the dataset at hand. In order to gain more understanding of how different techniques perform this research will extend Renjith et al. (2021) by inferring on simulated datasets with different geometrical and dimensional structures.

The structure of this paper is as follows, Section 2 provides an overview of relevant literature that was integral to this research. Section 3 details the research methods, such as the DR techniques and clustering algorithms. Section 4 presents the dataset used by Renjith et al. (2021) and the artificial datasets generated. The results derived can be found in Section 5, and conclusions will be given in Section 6.

2 Literature Review

The discussion regarding dimension reduction methods has become increasingly relevant due to the volume of data that is retrieved daily, (Yin & Kaynak, 2015). Clustering has become a valuable tool in various domains and therefore pre-processing data for these techniques has become more crucial than ever, (Shirkhorshidi et al., 2014).

Hotelling (1933) was one of the first researchers to formulate the well-known dimension reduction technique PCA, which still to this day is widely used in data analytics. One of the notable techniques for the dimension reduction domain is t-Distributed Stochastic Neighbour Embedding (t-SNE) (Van der Maaten & Hinton, 2008), which allows better visualization of high-dimensional data. In light of the developments for non-linear DR techniques, this research will expand the understanding of these methods, through a comparison of two innovative techniques, t-SNE and UMAP, and the most popular linear technique, PCA.

Clustering techniques are often applied to data with high dimensions, noise, missing observations, and sampled data, thus the performance of algorithms differs significantly on a case-by-case basis (Rodriguez et al., 2019). Various algorithms have been researched and compared to discover the best-suited one for a specific application. Bhatnagar et al. (2017) have applied k-means, hierarchical, Gaussian mixture modelling, fuzzy c-means, and self-organized map clustering algorithms to the manufacturing business domain, where the problem was to cluster various firms. It notable that by using the simulated datasets research will also be able to further infer how well clustering algorithms tackle datasets with different geometrical structures.

The comparison of dimension reduction techniques is an ongoing effort in the data analytics community. Various datasets and techniques are used to discover the best DR technique for a particular situation. Van Der Maaten et al. (2009) have given a comprehensive review of various

linear and non-linear techniques by applying them to natural and simulated datasets, which are the primary inspiration for the datasets used in this research. They have used classification methods for the assessment of techniques and discovered that PCA outperforms other posed techniques when applied to natural datasets, but falls short when used in artificial ones. The comparison of DR techniques was conducted using Morphometric data (Du, 2019), where the main goal was to identify if PCA was an appropriate method to use in this domain. Research showed that even though PCA was deemed a powerful technique, the non-linear DR techniques were able to preserve the differences between morphologies better. The comparison of the two DR techniques, UMAP and t-SNE, has also been conducted by Becht et al. (2018) using single-cell RNA data, who found that UMAP outperformed the counterpart. It is important to note that the non-linear DR techniques that are introduced in this research have not yet been compared with PCA using various datasets and clustering. The comparisons presented lack the implementation of above mentioned non-linear techniques, as well as the analysis of how DR techniques affect the clustering process.

3 Methodology

3.1 Clustering

In this section two clustering techniques will be covered, k-means and Agglomerative Nesting (AGNES). Both of these methods have been used in Renjith et al. (2021). For both of the algorithms, there is a common context. A dataset is given consisting of n data points, $X = \{x_1, x_2...x_n\}$, where each data observation $x_i \in \mathbb{R}^d$. The algorithms split the data into k clusters, $\{C^1, C^2, ...C^k\}$. Each cluster has a centroid, that is the central point of the cluster calculated as the mean average of all the points within the cluster, denoted by $\{c_1, c_2, ...c_k\}$.

3.1.1 k-means

A popular clustering algorithm is k-means, which iteratively rearranges the clusters to find the best partition. Given the dataset and the desired number of final clusters - k, the split of data is performed such that the Within-Cluster Sum of Squared distances (WCSS) is minimized. The WCSS is given by $WCSS = \sum_{j=1}^{k} \sum_{x_i \in C^j} d(x_i - c_j)^2$, where d() is a distance function. The steps of the algorithm are the following (Tan et al., 2005).

1. The k observations are selected randomly from X as initial centroids.

2. For every point in X, the distance to each centroid is calculated, and the point gets assigned to the cluster of its closest centroid. The Euclidean distance will be used as it has been shown to outperform other popular distance metrics by Singh et al. (2013) for k-means clustering.

3. For newly formed clusters the centroids are updated using the formula: $c_j = \frac{1}{|C^j|} \sum_{x_i \in C^j} x_i$.

4. The previous two steps are repeated until the chosen threshold for change in WCSS or the maximum number of allowed iterations is reached.

Notably, the algorithm is sensitive to the initial random selection, (Arora et al., 2016), thus, it is beneficial to perform k-means a number of times. It is also sensitive to outliers as they may form a cluster of their own, (Olukanmi & Twala, 2017). As a result, making decisions such as dealing with outliers and scaling the data have a crucial impact on the final results.

3.1.2 AGNES

AGNES (Agglomerative Nesting) algorithm is a clustering method that starts off by placing each observation in an individual, singleton, cluster and iteratively merges the most similar clusters. It is a Hierarchical clustering method and is often described as a greedy algorithm (Sasirekha & Baby, 2013). Given the dataset, the process of this algorithm is following, (Tan et al., 2005). The observations start as singleton clusters, so there are n clusters, denoted as $C^{i,0} = \{x_i\}$ where 0 stands for the iteration of the algorithm.

1. The pairwise distance between each pair of singleton clusters is calculated using a specific distance type. In this research, the Euclidean distance is used as it was shown to perform best for the AGNES clustering with different linkage methods, which is a way to measure the similarity between different clusters, (Stefan et al., 2015). This process yields a distance matrix D, where $D_{(i,0),(j,0)}$ is the distance between $C^{i,0}$ and $C^{j,0}$.

2. Given the matrix D two closest clusters are found, let $C^{i,0}$ and $C^{j,0}$ be the closest clusters. These two clusters merge resulting in a new cluster $C^{i,1} = C^{i,0} \cup C^{j,0}$. The cluster is assigned index i arbitrarily.

3. The distance matrix D is recalculated for the new structure of the clusters. The way distance is calculated for the merged cluster will depend on the type of linkage used. This research uses the Ward method, (Ward, 1963), as it was shown to outperform other methods (Blashfield, 1976; Ferreira & Hitchcock, 2009). This method differs from other linkage methods as it calculates distance based on the dissimilarity of clusters, and the objective is to minimize the

total within-cluster sum of squares (TSS). Given that any $C^{i,0}$ and $C^{j,0}$ clusters have been merged, the increase in the total within-cluster sum of squares (TSS) is calculated as a result of the merge, given by:

$$\triangle TSS = TSS_{i,p} - TSS_{i,(p-1)} - TSS_{j,(p-1)}, \qquad \text{where} \qquad TSS_{i,p} = \sum_{x \in C^{i,p}} \parallel x - c_{i,p} \parallel^2 \quad (1)$$

Where p is the iteration of the algorithm, in the singleton merger case p is 1. The $c_{i,p}$ is centroid of $C^{i,p}$, After this, the matrix D is updated given the merged cluster, for example, the dissimilarity between clusters $C^{i,1}$ and $C^{r,1}$ is: $D_{(i,1),(r,1)} = \frac{n_{i,0}}{n_{i,0}+n_{r,0}} D_{(i,0),(r,0)} + \frac{n_{j,0}}{n_{j,0}+n_{r,0}} D_{(i,0),(r,0)}$, where $n_{r,0}$ is the number of observations in cluster $C^{r,0}$. The last term of the calculation represents the distance to the newly merged part of the cluster $C^{i,1}$. For an arbitrary iteration p, where non-singleton cluster merger is performed $C^{k,p} = C^{k,p-1} \cup C^{m,p-1}$, the distance is updated as follows: $D_{(k,p),(r,p)} = \frac{n_{k,(p-1)}}{n_{k,(p-1)}+n_{r,(p-1)}} D_{(k,p-1),(r,p-1)} + \frac{n_{m,(p-1)}}{n_{m,(p-1)}+n_{r,(p-1)}} D_{(m,p-1),(r,p-1)}$. The distances of the previous iteration can be found in the distance matrix corresponding to that iteration of the algorithm.

4. Steps 2 and 3 are repeated till the termination criterion is reached, which can be a desired number of clusters, a specific distance value, or a maximum number of iterations allowed.

There are some notable characteristics of the AGNES algorithm, such as it does not require a predefined number of clusters. But it still requires choosing the desired number between 1 and n. Other decisions include the maximum number of iterations allowed and the linkage method, which results in different systematic tendencies (or biases) in the manner that observations are grouped and potentially significantly different outcomes, (Ketchen & Shook, 1996).

3.1.3 Selection of Optimal Number of Clusters

The selection of an optimal number of clusters is a vital part of the clustering process. For this task the function NBClust, (Charrad et al., 2014) was used in Renjith et al. (2021). The inputs of this function are the data, clustering method, distance method, and minimum and maximum number of clusters. As a result, the function produces the optimal number of clusters by using 26 different evaluation indexes, such as the ones in Section 3.3. The function then iteratively creates a clustering for every size from the minimum up to the maximum cluster number. For every clustering, the function records the value for evaluation indexes, and once the procedure is complete each evaluation index "votes" for the number of clusters where the index is at its best.

Finally these "votes" are counted and the number of clusters with the highest "votes" is selected as optimal. For this research, the Euclidean distance was selected as the distance measure, as it is used for both k-means and AGNES clustering. For the method input "ward.D2", which is equivalent to AGNES clustering with Ward linkage, and "k-means", which is simply the k-means method, were both calculated. Given the size of the data maximum number of clusters was limited to 10 and the minimum was set at 2, identical to Renjith et al. (2021).

3.2 Dimension Reduction (DR)

In this section, Dimension Reduction (DR) techniques will be covered, Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE), and Universal Manifold Approximation and Projection (UMAP). PCA and t-SNE have been used by Renjith et al. (2021), and the addition of UMAP is an extension of their research. All the DR techniques have a common context. Techniques transform the given dataset $\mathbf{X} \in \mathbb{R}^{n \times d}$, into m-dimensional space, $\mathbf{Y} \in \mathbb{R}^{n \times m}$, where $m < d$. Let x_i be an observation in $\mathbf{X}$ and y_i be an observation in $\mathbf{Y}$.

3.2.1 PCA

Principal Component Analysis (PCA) is a well-known linear DR technique. This technique discovers the directions in the data where there is the most variation. These directions are called principal components and they are created with linear combinations of the d dimensions.

The comprehensive steps this method takes are described in Tharwat (2016). For the given dataset it is important to note that condition $n > d$, must hold so that there are more observations than dimensions. Firstly data is standardized, as the technique aims to find the directions with the highest variation. If the scales of dimensions are not the same, the dimensions with higher scales will be more influential. Given $x_{i,w}$ which is the value for dimension w of observation i, it is possible to calculate, $\mu_w = \frac{1}{n}\sum_{i=1}^{n} x_{i,w}$ and $\sigma_w = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\mu_w - x_{i,w})^2}$. The standardization is following: $x'_{i,w} = (x_{i,w} - \mu_w)/\sigma_w$, and standardized data matrix $\mathbf{X'}$ is constructed. Then the covariance matrix is calculated, given by $\sum = \frac{1}{n-1}\mathbf{X'}^T\mathbf{X'}$, and it is possible to find the eigenvalues and eigenvectors of it. The eigenvalues will be denoted by $\{\lambda_1, \lambda_2, ...\lambda_d\}$, and eigenvectors - $\{\mathbf{e}_1, \mathbf{e}_2, ...\mathbf{e}_d\}$. If one desires a dimension of size m, m eigenvectors that have the largest eigenvalues are selected. These eigenvectors are the principal components. The number of selected dimensions depends on desired dimension or the amount of variance that one wants to retain. It is important to note that selected eigenvectors, $\{\mathbf{e}_1, \mathbf{e}_2, ...\mathbf{e}_m\}$, have to be orthogonal, otherwise they need to be transformed. Lastly, the new transformed data matrix,

$\mathbf{Y}$ is derived: $\mathbf{Y} = \mathbf{X}'\mathbf{E}$, where $\mathbf{E}$ is a matrix that has selected components/eigenvectors as columns.

3.2.2 t-SNE

A non-linear dimensionality reduction technique, t-Distributed Stochastic Neighbor Embedding (t-SNE) has frequently used to visualize high-dimensional data in a lower-dimensional space. The local structure and relative distance between the data points are preserved with t-SNE (Y. Wang et al., 2020), unlike linear approaches like PCA, which makes it particularly effective for identifying clusters or structures within the data. t-SNE is a useful tool for uncovering hidden patterns and structures that can go unnoticed in data (L. Wang et al., 2023).

The way the algorithm achieves the reduction in dimension is described extensively in Linderman & Steinerberger (2019) and Draganov et al. (2023). In the algorithm, the joint probability p_{ij}^t concept is defined by the similarity between two points x_i and x_j.

$$p_{ij}^t = \frac{p_{i|j}^t + p_{j|i}^t}{2n}, \qquad \text{where} \qquad p_{i|j}^t = \frac{exp(-d(x_i, x_j)^2/2\sigma_i^2)}{\sum_{h \neq l} exp(-d(x_h, x_l)^2/2\sigma_h^2)} \tag{2}$$

The σ_i is called the bandwidth of the Gaussian Kernel. The chosen distance function d() is Euclidean distance, which is often used for t-SNE (Van der Maaten & Hinton, 2008). After the calculation of p_{ij}^t for all the points, it is possible to construct the pairwise similarity matrix $\mathbf{P}$. The similarity also needs to be defined for the points y_i and y_j, which are represented in the lower dimensional space.

$$q_{ij}^t = \frac{(1 + d(y_i + y_j)^2)^{-1}}{\sum_{h \neq l}(1 + d(y_h - y_l)^2)^{-1}} \tag{3}$$

After the calculation of q_{ij}^t, the pairwise similarity matrix of lower dimensional embedding is constructed $\mathbf{Q}$. Kullback-Leibler difference calculates how different two probability distributions are. By minimizing the Kullback-Leibler difference, between $\mathbf{P}$ and $\mathbf{Q}$, t-SNE creates $\mathbf{Y}$ data matrix. The initial combination of points of $\mathbf{Y}$ is selected using PCA initialization, which has been shown to outperform the random initialization (L. G. Kobak D., 2021), it is unclear what initialization Renjith et al. (2021) used. The $C(\mathbf{Y})$ is minimized using gradient descent. This objective can be expressed as follows:

$$C(\mathbf{Y}) = KL(\mathbf{P} \parallel \mathbf{Q}) = \sum_{i \neq j} p_{ij}^t \log\left(\frac{p_{ij}^t}{q_{ij}^t}\right) \tag{4}$$

t-SNE has various drawbacks despite its benefits. Crowding Problem is a significant issue

that arises when data points are tightly packed in the lower-dimensional space, particularly when perplexity levels are low or Gaussian kernel bandwidths are small (Wattenberg et al., 2016). Overcrowding can result in clusters that are difficult to differentiate from one another or use visualization to analyze data. To achieve a balance between the preservation of local structure and the avoidance of crowding, careful parameter tuning is crucial. It will be interesting to observe if the dataset we apply t-SNE to will have a tendency for overcrowding and how that will influence finding underlying clusters within the dataset and effective visualization.

3.2.3 UMAP

The Universal Manifold Approximation and Projection (UMAP) is a non-linear dimension reduction method that aims to retain the local structure of the dataset as well as the global structure when transforming the data into a lower dimension (McInnes et al., 2018). There are three essential assumptions that UMAP relies on as stated in Allaoui et al. (2020). Firstly, data should be uniformly distributed on the Riemannian Manifold, which as J. M. Lee (2006) defines is a "smooth manifold equipped with Riemannian metrics (smoothly varying choices of inner products on tangent spaces), which allow one to measure geometric quantities such as distances and angles". To put it simply, this assumption states that the data has to have an intrinsic lower dimensional manifold that can be approximated by Euclidean space. In real-world scenarios, this assumption can be reasonable and hold for various datasets, but for others, this assumption can be weak. Secondly, the Riemannian metric needs to be constant locally for the dataset. This assumption can be interpreted as the distance and angles of points with the dataset having to be constant, and it is crucial for retaining the clusters within data and capturing local structure. For datasets with outliers and overlapping cluster structures, this assumption will not hold. Lastly, the manifold of the dataset needs to be locally connected, meaning that the manifold can be continuously connected to nearby points. The main takeaway of this assumption is that the points should be close in the lower dimensional manifold if they are close to the original dimension, which ensures the preservation of the local structure of the data. In datasets with, for example, clusters that are not well separated this assumption is less likely to hold strongly. It is important to note that for high-dimensional data sets it is not possible to evaluate to what extent UMAP's assumptions hold. These core assumptions are used to define the algorithm of UMAP, but it nonetheless can be beneficial to apply this DR technique to the datasets where assumptions don't hold as UMAP is designed to be a relatively robust DR technique, which can uncover the patterns in the high dimensional space (Stolarek et al., 2022; McInnes et al., 2018).

The algorithm approaches the creation of a lower dimensional manifold by representing the high dimensional data as a weighted graph representing a likelihood that some given two points are connected. The concept of joint similarity, p_{ij}^U is defined as:

$$p_{ij}^U = p_{i|j}^U + p_{j|i}^U - p_{i|j}^U \cdot p_{j|i}^U, \qquad \text{where} \qquad p_{i|j}^U = exp\left(-\frac{d(x_i, x_j) - \rho_i}{\sigma_i}\right) \tag{5}$$

The $p_{i|j}^U$ is a one-way similarity measure, and the d() is the distance function for which a commonly used Euclidean distance is chosen (McInnes et al., 2018). The ρ_i is the local connectivity parameter, which measures the distance from x_i to its nearest neighbor, $\rho_i = \min_{l \neq i} d(x_i, x_l)$. The σ_i is a local connectivity parameter, which is set to match the local distance around x_i and a predetermined number of nearest neighbors called k which is a hyperparameter. p_{ij}^U is a symmetrization of the high dimensional probabilities. It is needed because $p_{i|j}^U$ does not necessarily equal to $p_{j|i}^U$. This symmetrization is defined through probabilistic T-conorm, which is an operation that calculates the probability of disjunction of two events (McInnes et al., 2018). As a result, the probabilistic similarity matrix $\mathbf{P}$ is constructed. Afterwards, the UMAP algorithm creates a lower dimensional graph such that it is similar to a high dimensional one. The similarity measurement for the lower dimensional embedding is the following:

$$q_{ij}^U = (1 + a(y_i - y_j)^{2b})^{-1} \tag{6}$$

Where a and b can be specified or calculated from the given number of neighbors k. As a result the probabilistic similarity matrix for the lower dimensional embedding, $\mathbf{Q}$ is constructed. The initial combination of data points of $\mathbf{Y}$ is determined using the Laplacian Eigenmaps initialization (Belkin & Niyogi, 2002), which has been shown to outperform the random initialization by D. Kobak & Linderman (2019). For an in-depth understanding of Laplacian Eigenmaps refer to Appendix A. The cost function that UMAP uses is the binary Cross Entropy (CE) given by:

$$CE(\mathbf{P}, \mathbf{Q}) = \sum_i^n \sum_j^n (p_{ij}^U \log\left(\frac{p_{ij}^U}{q_{ij}^U}\right) + (1 - p_{ij}^U) \log\left(\frac{1 - p_{ij}^U}{1 - q_{ij}^U}\right)) \tag{7}$$

The Cross-Entropy is minimized using Stochastic Gradient Descent (Bonnabel, 2013). UMAP has a number of hyperparameters, such as the learning rate, which affects the speed of convergence of the gradient descent. The number of neighbors k, which is also a hyperparameter, and a lower value will capture fine details, and a higher one will lead to estimation being influenced by larger regions. The minimum distance between points in embedded space needs to be de-

termined which is used in the optimization process and ensures that points in lower dimensions are separated, a lower value allows for more accurate capturing of the true manifold but leads to denser clusters that are not suitable for visualization. In order to determine the best hyperparameters, a limited grid search was performed, due to limited computational resources. For the grid search the embedding was calculated for combinations of hyperparameters and evaluated. The parameters used are: learning rate = $\{0.5, 1, 5\}$, number of neighbours = $\{15, 25, 50\}$ and minimum distance = $\{0.5, 0.1, 0.05, 0.001\}$. All are within the acceptable ranges proposed by McInnes (2018).

3.3 Internal Validation of Clustering Quality

The evaluation of clustering produced by the DR technique with clustering techniques is performed with the internal validation assessment. This procedure involves evaluating the clusterings for different validation indexes, which were used by Renjith et al. (2021).

The first index is the well-known Silhouette Index (Rousseeuw, 1987), which evaluates how well each observation is clustered. The range is from -1 to 1, where a value close to 1 indicated that the observation is in the correct cluster, a value close to 0 - close or near the decision boundary between two clusters, and a value close to -1 - observation should not be in the assigned cluster as it is more fitting in another one.

The second index is the Dunn index (Dunn, 1973), which evaluates the compactness and separation of clusters. This index can be interpreted as a trade-off between maximizing the distance between clusters and minimizing the distance within clusters. The higher value for this measure implies that clusters are compact and well-separated.

The third index is the Calinski-Harabasz index (Caliński & Harabasz, 1974), which measures the compactness and separation of clusters as well. This index calculates the ratio of the within-cluster and between-cluster dispersions. The aim is to maximize the between-cluster dispersion while minimizing the within-cluster one, and a higher value for this measure is desirable.

The last index is the Davies-Bouldin index (Davies & Bouldin, 1979), which measures how similar and dissimilar clusters are. The value ranges from 0 to 1, and the lower value is preferred, as it implies well-separated clusters, that are homogeneous. For a more in-depth understanding of all Internal Validation Indexes refer to Appendix B.

4 Data

The first dataset used in this research is the Jester dataset, (Goldberg, 2001). This dataset contains approximately 4.1 million anonymous ratings for jokes ranging between -10 and 10. The ratings are continuous and a total of 73,421 individuals rated 100 jokes. The visualizations that were presented by Renjith et al. (2021), for the clustering indicate that 5000 observations have been used for visualizations. It is important to note that it is unclear if the 5000 observations were used for only visualizations or for the whole process. Due to restricted computational resources, this research will use 5000 observations from the Jester dataset that will be randomly sampled. Renjith et al. (2021) has also omitted if they had missing values in the dataset and how they dealt with them. Thus the 5000 observations will be sampled from the subset of jokes that have all the observations, which adds up to 14,116. It is important to note that ensuring the reproducibility of the research conducted is an integral part of this research, thus, it is possible to view in-depth information about the steps taken to achieve results in Appendix 3. It is also important to note that as a number of techniques employed in this research use randomization, seeds have been set for every step in the code to ensure reproducability.

The simulated datasets will be used as they allow us to gain more insight into the performance of DR techniques. These datasets will challenge the ability of DR techniques of reducing cluster variance whilst preserving the global structure. Each dataset has a size of 5000.

The first type of dataset will exhibit a simple geometric structure. It can be accurately represented in a lower-dimensional space while keeping the distances between the points. In other words, the data is smooth and without any abrupt changes or curves; for this purpose a Swiss Roll, Figure a, is generated. The geometry of this dataset can be interpreted as uniformly distributed across a Riemannian manifold, where the manifold appears on the rolled-up surface. The assumption of a Riemannian metric that is locally constant denotes a roughly uniform distribution of lengths and angles on the manifold. Given that close spots on the rolled-up surface can be continually connected, the dataset is likewise locally connected. Due to this, it is expected that UMAP will outperform other techniques for this data set. There are three well-defined clusters in the dataset, which are separated by slight spacing.

The second type of dataset has a piece-wise or discontinuous geometric structure that cannot be accurately represented by a single low-dimensional manifold. For this purpose a Broken Swiss Roll is generated, Figure b. This dataset has a visibly more compressed inner layer compared to the traditional Swiss Roll and breakage on the outer layer, the clusters consist of inner and

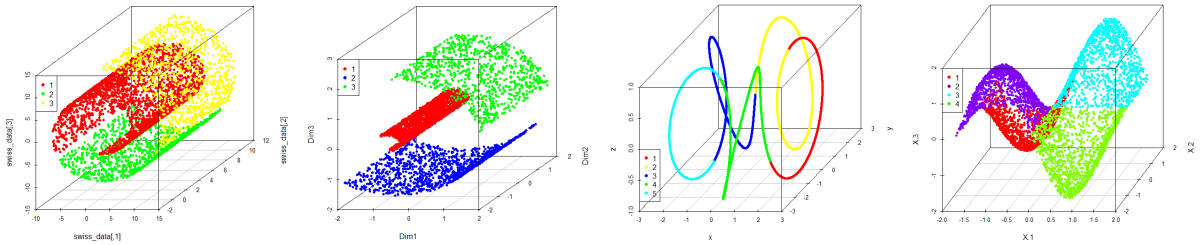
2 outer layers. The compressed inner layer of the Broken Swiss Roll dataset makes it likely that it fails to fully satisfy the requirement of a Riemannian manifold with a locally constant metric. Non-uniform distances and angles can arise from differences in the metric caused by the compression in the inner layer. Local connectivity remains evident in the dataset, though, as surrounding spots on the split surface can still be continually connected. As this data set does not fully satisfy the locally constant metric assumption posed by UMAP, it can not be expected that UMAP outperforms other DR techniques.

The third type of dataset is one that exhibits complex topological features, such as holes, loops, or self-intersections, that prevent it from being reduced to a Euclidean space, it can be transformed into a lower-dimensional space, but the distances between the observations will not be retained. For this purpose, Twinpeaks, Figure d, and Helix, Figure c, will be generated. For the TwinPeaks the two lower and two upper peaks intentionally have a denser structure compared to other surface points to create inherent clusters. The assumption of a uniformly distributed dataset on a Riemannian Manifold may not hold true for the Twinpeaks dataset having denser peaks. Local non-uniform distribution and non-constant metric could arise from changes in density in the Twinpeaks. The surface of the peaks can still be continually connected, hence the dataset can still show local connectivity. The Helix dataset has a lower scale for the third dimension in comparison to the other two, which results in clusters that are following the shape of the Helix, rather than, for example, those that divide it vertically. The Helix dataset can be described as having an even distribution on a Riemannian manifold according to the Helix's shape. The helix's distances and angles are assumed to be roughly uniform throughout small regions of a locally constant Riemannian metric. As close locations on the helix can be continually connected, the dataset is locally connected. Given that these datasets exhibit complex topological patterns, and don't fully satisfy the assumptions of UMAP, it is hypothesized that the t-SNE technique will outperform others, as it does not have concrete assumptions that datasets violate.

Lastly, we will examine a High-Dimensional dataset that has a complex and rich geometric structure that cannot be effectively represented in a lower-dimensional space without significant loss of information due to its dimensionality. More specifically this dataset will be generated by randomly creating points from a 5-dimensional non-linear space which will be structured in a 10-dimensional space. The dataset is created by applying a number of trigonometric and linear transformations to the random variables and creating a defined pattern within the dataset. For this dataset, it will be interesting to observe whether or not the DR techniques are able to

capture the underlying manifold or if clustering without DR is more beneficial.

For a detailed explanation of data generation and more visualizations consult Appendix C.



(a) Swiss Roll (b) Broken Swiss Roll (c) Helix (d) TwinPeaks

Figure 1: Artificial datasets with size 5000

5 Results

5.1 Number of Clusters

Table 1 presents the results for the NbClust function for the Jester dataset, with the optimal number chosen by each index and the final verdict. The function concluded that the optimal number of clusters is 3 for both methods, which is the same result as the Renjith et al. (2021). It is important to note that even though the final verdict of the functions is the same as the authors, the actual cluster votes for indexes are different. These differences can be explained by several factors, such as the randomly sampled dataset, and lack of information for their function inputs (method and distance). It is also notable that in case Renjith et al. (2021), have used the "k-means" method for the NbClust function, the results can differ due to the random initialization of the k-means. For artificial datasets the two NbClust functions were able to identify the intrinsic number of clusters, Swiss Roll and Broken Swiss Roll - 3 clusters, Helix - 5 clusters, and TwinPeaks - 4 clusters. For the High-Dimensional dataset, both functions concluded that 3 clusters are optimal and thus that number is used to derive further results. For more information on the Output of NbClust, view Appendix E.

dataset	Method	Scott	Cindex	Ball	McClain	Dindex	KL	Silhouette	Duda	PseudoT2	SDbw	Hartigsn	Marriot	TrCovW	TraceW	Friedman	Beale	Ratakowsky	CH	CCC	Rubin	DB	Frey	Hubert	PtBiserial	Dunn	SDindex	Verdict
Jester	**K-Means**	-inf	2	3	6	9	2	3	3	3	2	3	5	3	3	3	2	3	2	3	4	2	2	2	1	0	0	**3**
	Ward	-inf	2	4	10	9	7	NA	NA	NA	7	3	6	3	3	3	2	3	2	3	3	2	2	2	1	0	0	**3**

Table 1: Jester NbClust result

5.2 Replication of Renjith et al. (2021)

After applying the DR and clustering techniques to the Jester dataset the visualizations can be seen in Figure 2. The clustering achieved by using AGNES has a tendency to create overlapped clusters, Renjith et al. (2021) has also found this behavior in their visualizations. The visualizations of t-SNE and PCA also have a similar shape to that of Renjith et al. (2021). The visualizations of Renjith et al. (2021) can be found in Appendix G.

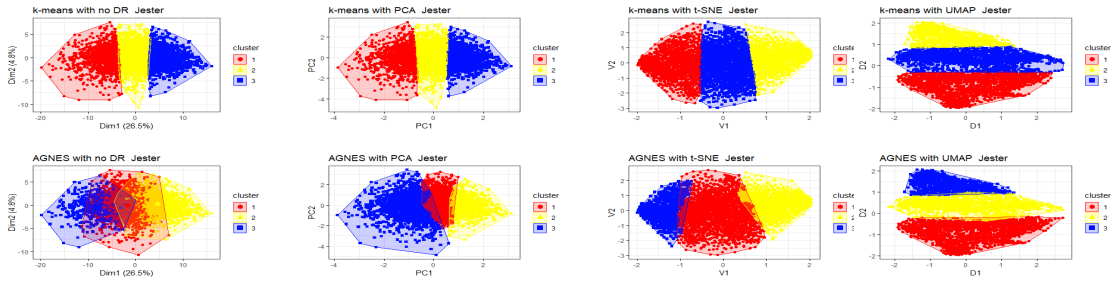


Figure 2: Visualizations of Clusters formed by DR techniques. dataset size is 5000

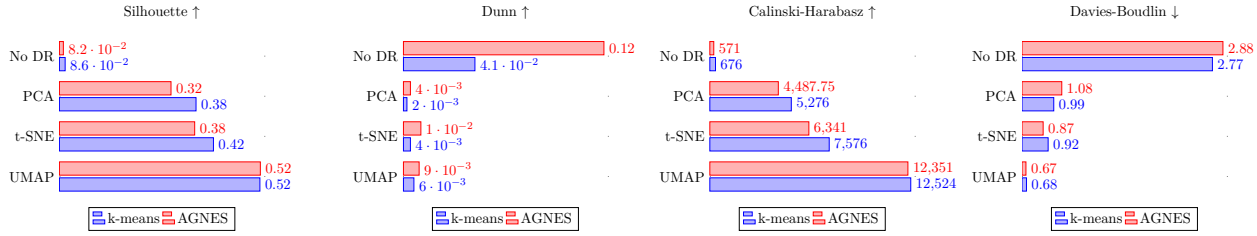


Figure 3: Internal Validation Indexes Results

Figure 3 summarizes the results for Internal Validation Indexes for all clustering and DR combinations. It is important to note that Renjith et al. (2021) omitted the results for the non-reduced dataset so it is not possible to validate if those results are in line with theirs. Figure 3 also indicates if a higher value indicated better clustering for an index with ↑, and if a lower one indicates better clustering the ↓ is used. In this section the focus will be on No DR, PCA, and t-SNE techniques as they are used by Renjith et al. (2021), and an additional technique, UMAP, will be discussed in Section 5.3.

For the Silhouette index, it is possible to see that t-SNE outperforms PCA for k-means and AGNES. The clustering using No DR data is unable to outperform either one of the methods. Thus it is possible to conclude that t-SNE is able to classify observations into correct clusters better. Renjith et al. (2021) has come to the same conclusion. The techniques perform better when k-means are used, which can be explained by the overlapping clusters produced by AGNES.

The Dunn index evaluates how compact clusters are and how well separated they are from

each other. The dataset that had no DR techniques applied to it outperformed the other two, for both clustering techniques. This is an interesting result, and more investigation is necessary such as examining if this result is driven by the relatively high minimal distance between different clusters, or by the low maximal distance of observations within clusters. One reason can be the structure of the original data set that has 100 dimensions and a limited range, and clusters appear more separate and compact when Dunn is calculated. One other hypothesis can be that the DR techniques may inflate the outliers and noise of the dataset, which creates a less compact cluster structure. The clustering using t-SNE outperforms one with PCA for both methods, which is a conclusion that Renjith et al. (2021) derived as well. There is one difference with Renjith et al. (2021) is that all techniques perform better when AGNES is applied.

The Calinski-Harabasz index indicates how well clusters are separated and how compact they are as well. It is possible to see that clustering using t-SNE outperforms PCA, and no DR performs worst. All the techniques perform better when k-means is used. These results are fully in line with the conclusion of Renjith et al. (2021). This result is quite interesting as both this index and Dunn measure the same behavior for clustering. The reason why they disagree needs more investigation. This can be due to Dunn only calculating the minimal distance between two points from different clusters and the maximal distance of two points in the same cluster, whilst Calinski-Harabasz calculates the compactness and separation of all the points in every cluster.

The Davies-Bouldin index evaluated how similar in-cluster observations are and how dissimilar clusters are from each other. It is clear that t-SNE yields lower results for both clustering methods compared to PCA and no DR, which is in line with Renjith et al. (2021). The No DR dataset clustering falls short for this metric compared to t-SNE and PCA. It is crucial to note that the value for this index ranges from 0 to 1 typically, so observing high values for the No DR dataset is atypical, and can be due to the high dimensionality of the dataset. Renjith et al. (2021) have found that using k-means yields better results, which is also the case in this research, but they found values that are above 1.5 for both PCA and t-SNE with AGNES.

One of the main discrepancies that can be pointed out in this research in comparison to Renjith et al. (2021) is the results derived when AGNES is used. Renjith et al. (2021) has found that the difference in performance between clustering techniques was greater than the results presented. This can be due to the linkage method used, which was not specified in the original research and could be different from Ward linkage, which has not been explored.

5.3 Extension

Firstly the results of UMAP performance for the Jester dataset will be discussed. From Figure 2, it is clear that the application of UMAP produces a visually less uniform visualization compared to other DR techniques. It is visible that the right side of the dataset is more dispersed, whilst the left side is more compact. UMAP was also assessed using Internal Validation Indexes, and the results can be viewed in Figure 3. UMAP was able to outperform all the other techniques for Silhouette, and Davies-Bouldin indexes, and alike other techniques performed slightly better when k-means was used. For Calinski-Harabasz, UMAP with AGNES outperformed all others. For the Dunn index, UMAP was unable to outperform the clustering with No DR. It performed better than PCA for both clustering methods, but was able to outperform t-SNE only for k-means clustering.

	Jester					Swiss Roll					Broken Swiss Roll			
	Silhouette↑	Dunn↑	Calinski-Harabasz↑	Davies-Bouldin↓		Silhouette↑	Dunn↑	Calinski-Harabasz↑	Davies-Bouldin↓		Silhouette↑	Dunn↑	Calinski-Harabasz↑	Davies-Bouldin↓
No DR K	0.086	0.041	675.840	2.766	*No DR K*	0.370	0.005	3539.185	0.995	No DR K	0.362	0.002	2644.060	1.189
No DR A	0.082	**0.115**	571.149	2.884	*No DR A*	0.305	0.015	2785.185	1.147	No DR A	0.372	0.277	2076.311	1.465
PCA K	0.379	0.002	5275.639	0.986	*PCA K*	0.442	0.003	4689.605	0.863	PCA K	0.466	0.003	4643.480	0.898
PCA A	0.323	0.004	4486.820	1.076	*PCA A*	0.403	0.002	3938.793	0.933	PCA A	0.442	0.003	4243.757	0.942
t-SNE K	0.418	0.004	7576.000	0.923	*t-SNE K*	0.409	0.005	4406.693	0.889	t-SNE K	0.427	0.003	4487.774	0.869
t-SNE A	0.376	0.010	6341.444	0.867	*t-SNE A*	0.400	0.008	3798.222	0.831	t-SNE A	0.412	0.023	3981.897	0.965
UMAP K	0.522	0.006	**12523.525**	0.680	*UMAP K*	**0.712**	**0.598**	**12046.158**	**0.408**	UMAP K	**0.730**	**0.756**	**17164.629**	**0.432**
UMAP A	**0.523**	0.009	12351.303	**0.674**	*UMAP A*	**0.712**	**0.598**	**12046.158**	**0.408**	UMAP A	**0.730**	**0.756**	**17164.629**	**0.432**

	TwinPeaks					Helix					High- Dimensional			
	Silhouette↑	Dunn↑	Calinski-Harabasz↑	Davies-Bouldin↓		Silhouette↑	Dunn↑	Calinski-Harabasz↑	Davies-Bouldin↓		Silhouette↑	Dunn↑	Calinski-Harabasz↑	Davies-Bouldin↓
No DR K	0.475	0.003	4642.768	0.741	*No DR K*	0.356	**0.003**	3414.748	1.000	*No DR K*	0.218	0.061	1488.248	1.514
No DR A	0.446	0.016	4126.695	0.835	*No DR A*	0.352	**0.003**	3392.664	1.010	*No DR A*	0.152	0.060	1070.090	1.745
PCA K	0.459	0.006	6720.064	0.806	*PCA K*	0.420	0.001	**5440.197**	0.789	*PCA K*	0.386	0.002	3829.041	0.957
PCA A	0.423	0.012	5851.897	0.858	*PCA A*	0.424	0.002	4946.513	**0.758**	*PCA A*	0.328	0.006	2931.597	1.100
t-SNE K	0.438	0.002	5624.875	0.801	*t-SNE K*	0.347	0.002	4111.569	0.987	*t-SNE K*	0.455	0.006	6306.474	0.845
t-SNE A	0.385	0.008	4401.276	0.852	*t-SNE A*	0.307	0.002	3519.037	1.066	*t-SNE A*	0.416	0.016	5402.744	0.841
UMAP K	**0.647**	0.009	**14180.205**	**0.528**	*UMAP K*	**0.444**	0.002	5432.330	0.792	*UMAP K*	**0.884**	**2.03**	**109251.332**	**0.145**
UMAP A	**0.647**	**0.139**	14178.524	**0.528**	*UMAP A*	0.438	**0.003**	5176.989	0.801	*UMAP A*	**0.884**	**2.03**	**109251.332**	**0.145**

Table 2: Results of Clustering Of All Datasets, Best Results are Bolded. A-AGNES and K-k-means. ↑ means higher value implies better clustering, ↓ - lower value is better clustering.

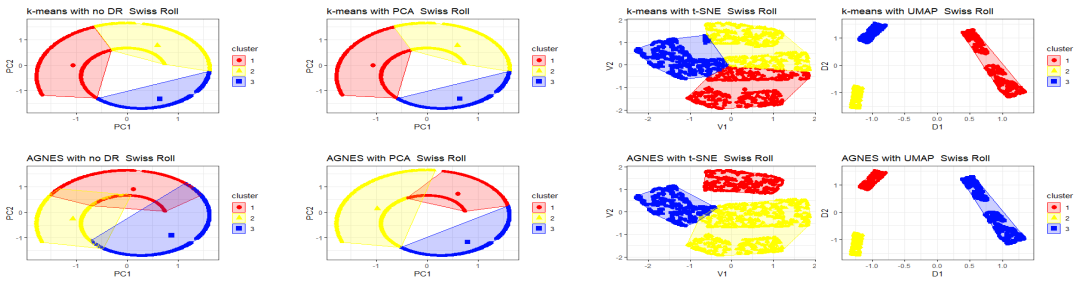


Figure 4: Visualizations of Clusters formed by DR techniques for Swiss Roll dataset.

Firstly the results for the Swiss Roll dataset will be discussed. The number of clusters chosen for the two clustering methods is 3. Table 2, clearly shows that both clustering achieved by using UMAP show equal and best performance. The Davies-Bouldin index also shows that all other DR techniques have a value that is either relatively close to 1 or more than 1. This indicates

that the observations are not well clustered for all the methods other ones using UMAP. This is not a surprising result as the Swiss Roll dataset satisfied all the assumptions of UMAP. It is also interesting to note that PCA has provided an accurate representation of a Swiss Roll with 2 dimensions, but as the distance within regions was not retained accurately was not able to cluster the Swiss Roll accurately. It is also interesting to observe that the clustering achieved by No DR also fell short of UMAP, so it can be argued that UMAP captured the small distance between clusters and enhanced it to make better-defined clusters.

The Broken Swiss Roll dataset has 3 clusters, the visualizations and results can be viewed in Figure 5. The table shows that both clusterings achieved by using UMAP show equal and best performance. As in the Swiss Roll case, the reduction achieved by UMAP creates 3 distinct clusters and thus there is no significant difference in the performance of clustering methods. The results that UMAP achieves for this dataset show that even when all the assumptions are not satisfied, such as uniform distance, the UMAP can still perform well.

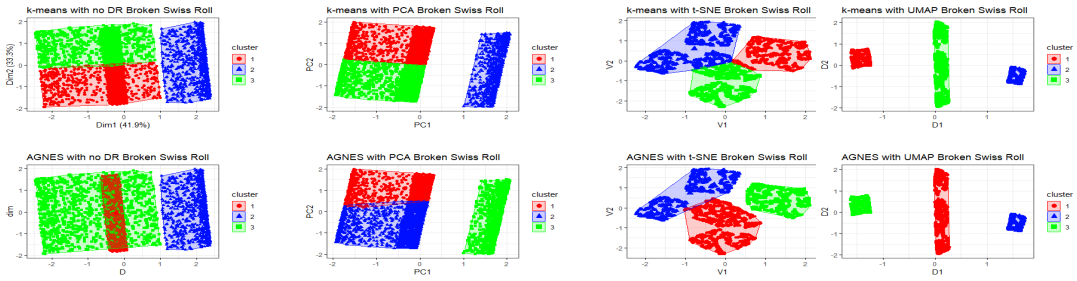


Figure 5: Visualizations of Clusters formed by DR techniques for Broken Swiss Roll.

The visualizations for the Helix dataset achieved using DR and clustering techniques are presented in Figure 6. It is possible to see that the PCA is able to capture the uniform geometry of the Helix, while the non-linear techniques are presenting coiled shapes. It is also interesting to note that all the DR techniques were able to create a shape that has no breaks, as the original Helix. The results from Table 2 show that clustering attained using PCA with AGNES outperforms all others for the Davies-Bouldin index. The PCA with k-means gives the best results for the Calinski-Harabasz index, which can be explained by the uniform structure of the clusters formed. For the Dunn index, the clusterings resulting from k-means and AGNES without using DR techniques, and UMAP with AGNES are best, and this might indicate that whilst reducing the dimensionality in this dataset the other DR techniques might not retain the same distance between the data points. The Silhouette index shows that UMAP with k-means yields the best results, followed by UMAP with AGNES. This disagreement in the result is interesting and for future work, it can be beneficial to explore other validation indexes. By

examining the 3-dimensional visualizations of clusterings, found in Appendix H.4, it was possible to see that PCA with AGNES is able to create clusters in a shape of a Helix. The clusterings using UMAP and t-SNE have breakages, so they don't capture the true clusters. This can be due to the complex structure of the Helix, resulting in DR techniques being unable to retain the true distance and geometry.

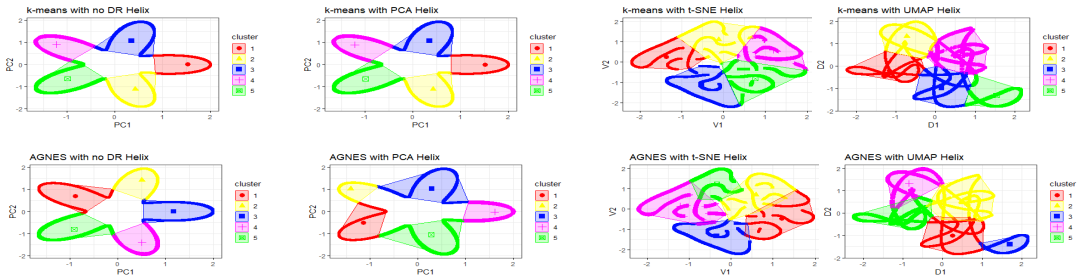


Figure 6: Visualizations of Clusters formed by DR techniques for Helix dataset.

The visualizations of TwinPeaks for different clustering and DR techniques can be viewed in Figure 7. The results of internal validation indexes, Table 2, indicate that UMAP has the best and equal for both clustering for, Silhouette, and Davies-Bouldin indexes. For the Dunn index UMAP with AGNES performs best, and for Calinski-Harabasz UMAP with k-means slightly outperforms the one with AGNES. This is an interesting result as UMAP was not performing well with the Helix dataset, which too exhibits complex topological features.

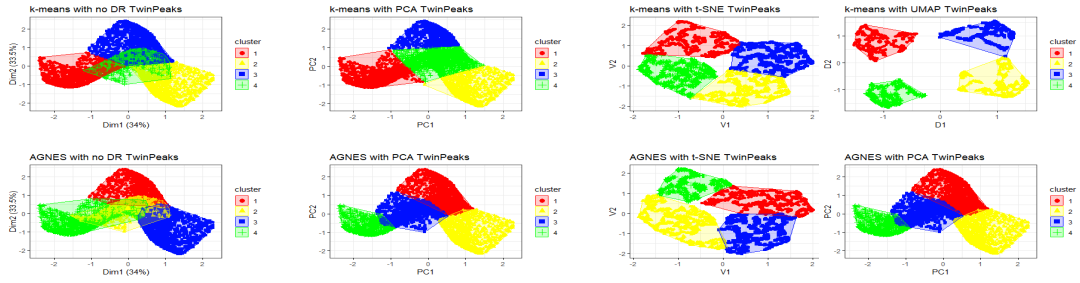


Figure 7: Visualizations of Clusters formed by DR techniques for TwinPeaks.

The visualizations of clusterings achieved by DR techniques transformed into 3 dimensions for all of the above data sets can be viewed in Appendix H.

Lastly, the High Dimensional dataset which had data points created on the 5-dimensional non-linear plane transformed into 10 dimensions, will be examined. From the visualizations in Figure 8, it is possible to see that the clusterings created using UMAP were able to identify the 3 distinct clusters present in the dataset. The fact that the use of UMAP for this dataset is beneficial can be seen in the results for internal validation indexes in Table 2, where both clustering techniques perform equally well and outperform others.

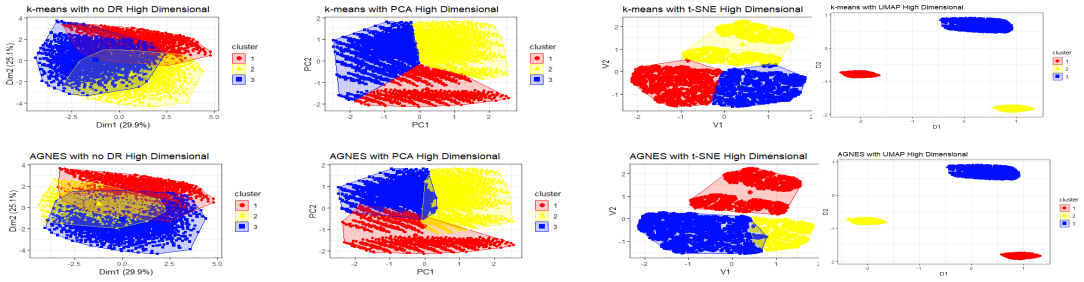


Figure 8: Visualizations of Clusters formed by DR techniques for HD dataset.

6 Conclusion

The central aim of this research was to gain a better understanding of the performance of DR techniques for clustering applications. Research achieved this goal through replication and extension of previous research, (Renjith et al., 2021), that had the same aim, which applied DR techniques and clustering to the Jester dataset. Renjith et al. (2021) found that the non-linear DR technique t-SNE outperformed others for two clustering techniques used, k-means and AGNES. In order to challenge the supremacy of t-SNE, a similar non-linear technique UMAP was applied. The research found that UMAP outperformed t-SNE and other methods for three out of four validation indexes. The only index that UMAP did not perform well for is the Dunn index, which assesses how compact clusters are and how well separated they are from each other. For this index, the clustering achieved by no DR performed best, especially when AGNES clustering was applied. This result is insightful as the calculation of the Dunn index and the nature of the dataset can drive it. For future research, it will be advised to examine what drives the favourable result achieved by no DR dataset the relatively high minimal distance between different clusters, or the low maximal distance of observations within clusters. Another suggestion would be to also assess Dunn Index Modifies (MDI) (Manochandar et al., 2020), which modifies the Dunn index by taking into account the density of clusters.

Dimension Reduction techniques and clustering methods both are influenced by the dataset at hand. In order to understand how these methods are able to tackle datasets with different characteristics five artificial datasets with varying geometrical structures were generated. The assessment of methods for these datasets is the same as the Jester dataset. The generated datasets allow the exploration of targeted questions about how well DR techniques tackle datasets with different geometrical properties.

For the simple geometrical dataset, Swiss Roll, UMAP performed best by a long shot and there is no significant difference in using different clustering algorithms. Notably, the visualiz-

ation presented by UMAP does not represent the Swiss Roll dataset accurately and PCA was able to visualize data better in a lower dimension.

The second type of dataset has a discontinuous, piece-wise structure with denser regions, the Broken Swiss Roll dataset. After evaluating the clusters formed by different cluster and DR methods, UMAP was found to achieve the best results once again. This is a notable result as one of the core assumptions of UMAP, the Riemannian manifold with a locally constant metric, is violated in this dataset. UMAP outperformed other techniques by a significant amount for Internal Validation Indexes and also created a visualization that resembles the inner and two outer layers of Broken Swiss Roll.

The third type of dataset exhibited complex topological features, with varying densities, for this type of dataset Helix and TwinPeaks datasets are generated. The results of the reduction of Helix showed that none of the techniques was unanimously better. For the validation Indexes clusterings with PCA, no DR and UMAP showed the best results for different indexes. Visualizations produced by non-linear techniques did not represent the uniform structure of the Helix, which was achieved by PCA. This result is quite interesting and more investigation into the performance of the techniques with looping geometrical shapes is required. For the TwinPeaks dataset, UMAP was able to outperform all others and yielded similar results for both clustering methods. The visualizations show that UMAP separated the peak regions and clustered them.

Lastly, a High-Dimensional dataset with a complex and rich structure was examined. UMAP, again, was able to prevail over other techniques with no difference in the clustering technique. The visualization provided by UMAP also separated clusters better than others.

In conclusion, this research showed that the use of UMAP is beneficial for rich and complex datasets. One of the hypotheses that can be drawn from the performance of UMAP on artificial datasets is that it is unable to perform well for not well-defined or non-separated cluster structures. All the datasets other than Helix had varying densities and gaps that allowed for the partitioning of clusters, but Helix did not have any breakages. Given UMAP's performance for Helix, it is suggested to research different datasets with complex geometries to understand the limitations of UMAP further. Another suggestion for further research is to implement clustering techniques such as Density-based clustering. It will also be interesting to examine the performance of DR techniques for other domains such as anomaly detection and time-series datasets. UMAP was able to capture and exaggerate the separation between different clusters. It is expected that UMAP will identify the anomalies and single them out. For time-series datasets, there are no concrete expectations, but this feature will definitely be interesting to investigate.

References

Allaire, J. (2012). Rstudio: integrated development environment for r. *Boston, MA*, *770*(394), 165–171.

Allaoui, M., Kherfi, M. L., & Cheriet, A. (2020). Considerably improving clustering algorithms using umap dimensionality reduction technique: A comparative study. , 317-325. doi: 10.1007/978-3-030-51935-3_34

Arora, P., Varshney, S., et al. (2016). Analysis of k-means and k-medoids algorithm for big data. *Procedia Computer Science*, *78*, 507-512. doi: 10.1016/j.procs.2016.02.095

Becht, E., Dutertre, C.-A., Kwok, I. W., Ng, L. G., Ginhoux, F., & Newell, E. W. (2018). Evaluation of umap as an alternative to t-sne for single-cell data.
doi: 10.1101/298430

Belkin, M., & Niyogi, P. (2002). Laplacian eigenmaps and spectral techniques for embedding and clustering.
doi: 10.7551/mitpress/1120.003.0080

Bhatnagar, V., Majhi, R., & Jena, P. R. (2017). Comparative performance evaluation of clustering algorithms for grouping manufacturing firms. *Arabian Journal for Science and Engineering*, *43*, 4071-4083. doi: 10.1007/s13369-017-2788-4

Blashfield, R. (1976). Mixture model tests of cluster analysis: Accuracy of four agglomerative hierarchical methods. *Psychological Bulletin*, *83*, 377-388. doi: 10.1037/0033-2909.83.3.377

Bonnabel, S. (2013). Stochastic gradient descent on riemannian manifolds. *IEEE Trans. Automat. Contr.*, *58*, 2217-2229. doi: 10.1109/tac.2013.2254619

Caliński, T., & Harabasz, J. (1974). A dendrite method for cluster analysis. *Comm. in Stats. - Theory Methods*, *3*, 1-27. doi: 10.1080/03610927408827101

Charrad, M., Ghazzali, N., Boiteau, V., & Niknafs, A. (2014). Nbclust: an r package for determining the relevant number of clusters in a data set. *Journal of statistical software*, *61*, 1–36.

Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.*, *PAMI-1*, 224-227. doi: 10.1109/tpami.1979.4766909

Draganov, A., Jørgensen, J. R., Nellemann, K. S., Mottin, D., Assent, I., Berry, T., & Aslay, C. (2023). Actup: Analyzing and consolidating tsne and umap. *arXiv preprint arXiv:2305.07320*.

Du, T. Y. (2019). Dimensionality reduction techniques for visualizing morphometric data: Comparing principal component analysis to nonlinear methods. *Evolutionary Biology*, *46*(1), 106–121.

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

Ferreira, L., & Hitchcock, D. B. (2009). A comparison of hierarchical methods for clustering functional data. *Communications in Statistics - Simulation and Computation*, *38*, 1925-1949. doi: 10.1080/03610910903168603

Goldberg, K. (2001). *Jester dataset.* `https://eigentaste.berkeley.edu/dataset/`.

Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2005). The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, *27*. doi: 10.1007/bf02985802

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, *24*, 417-441. doi: 10.1037/h0071325

Ketchen, D. J., & Shook, C. L. (1996). The application of cluster analysis in strategic management research: An analysis and critique. *Strat. Mgmt. J.*, *17*, 441-458. doi: 10.1002/(sici)1097-0266(199606)17:63.0.co;2-g

Kobak, D., & Linderman, G. C. (2019). Umap does not preserve global structure any better than t-sne when using the same initialization.
doi: 10.1101/2019.12.19.877522

Kobak, L. G., D. (2021). Initialization is critical for preserving global data structure in both t-sne and umap. *Nat Biotechnol*, *39*, 156-157. doi: 10.1038/s41587-020-00809-z

Lee, J. A., Verleysen, M., et al. (2007). *Nonlinear dimensionality reduction* (Vol. 1). Springer.

Lee, J. M. (2006). *Riemannian manifolds: an introduction to curvature* (Vol. 176). Springer Science & Business Media.

Linderman, G. C., & Steinerberger, S. (2019). Clustering with t-sne, provably. *SIAM Journal on Mathematics of Data Science*, *1*, 313-332. doi: 10.1137/18m1216134

Maaten, L. V. D. (2013). *Matlab-toolbox-for-dimensionality-reduction.* `https://github.com/UMD-ISL/Matlab-Toolbox-for-Dimensionality-Reduction`. (Version 0.8.1b)

Manochandar, S., Punniyamoorthy, M., & Jeyachitra, R. K. (2020). Development of new seed with modified validity measures for k-means clustering. *Computers Industrial Engineering*, *141*, 106290. doi: 10.1016/j.cie.2020.106290

Matlab, S. (2012). Matlab. *The MathWorks, Natick, MA*.

McInnes, L. (2018). *Basic umap parameters.* `https://umap-learn.readthedocs.io/en/latest/parameters.html`.

McInnes, L., Healy, J., & Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Olukanmi, P. O., & Twala, B. (2017). K-means-sharp: Modified centroid update for outlier-robust k-means clustering.
doi: 10.1109/robomech.2017.8261116

Renjith, S., Sreekumar, A., & Jathavedan, M. (2021). A comparative analysis of clustering quality based on internal validation indices for dimensionally reduced social media data. In *Advances in artificial intelligence and data engineering: Select proceedings of aide 2019* (pp. 1047–1065). doi: 10.1007/978-981-15-3514-7_78

Rodriguez, M. Z., Comin, C. H., Casanova, D., Bruno, O. M., Amancio, D. R., Costa, L. d. F., & Rodrigues, F. A. (2019). Clustering algorithms: a comparative approach. *PLoS ONE*, *14*, e0210236. doi: 10.1371/journal.pone.0210236

Rousseeuw, P. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53-65. doi: 10.1016/0377-0427(87)90125-7

Sasirekha, K., & Baby, P. (2013). Agglomerative hierarchical clustering algorithm - a review. *International Journal of Scientific and Research Publications*, *83*(3), 83.

Shirkhorshidi, A. S., Aghabozorgi, S., Wah, T. Y., & Herawan, T. (2014). Big data clustering: A review. , 707-720. doi: 10.1007/978-3-319-09156-3_49

Singh, A., Yadav, A., & Rana, A. (2013, 04). K-means with three different distance metrics. *International Journal of Computer Applications*, *67*, 13-17. doi: 10.5120/11430-6785

Stefan, R. A., Szöke, I.-A., & Holban, S. (2015). Hierarchical clustering techniques and classification applied in content based image retrieval (cbir). , 147–152. doi: 10.1109/saci.2015.7208188

Stolarek, I., Samelak-Czajka, A., Figlerowicz, M., & Jackowiak, P. (2022). Dimensionality reduction by umap for visualizing and aiding in classification of imaging flow cytometry data. *IScience*, *25*, 105142. doi: 10.1016/j.isci.2022.105142

Tan, P., Steinbach, M., & Kumar, V. (2005). *Introduction to data mining.* Addison-Wesley Longman, Boston, Mass, USA,.

Team, R., et al. (2016). R: A language and environment for statistical computing. *R Found. Stat. Comput.*, *1*, 409.

Tharwat, A. (2016). Principal component analysis - a tutorial. *IJAPR*, *3*, 197. doi: 10.1504/ijapr.2016.079733

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, *9*(11).

Van Der Maaten, L., Postma, E., Van den Herik, J., et al. (2009). Dimensionality reduction: a comparative review. *J Mach Learn Res*, *10*(66-71), 13.

Wang, L., Hare, B. M., Zhou, K., Stöcker, H., & Scholten, O. (2023). Identifying lightning structures via machine learning. *Chaos, Solitons Fractals*, *170*, 113346. doi: 10.1016/j.chaos.2023.113346

Wang, Y., Huang, H., Rudin, C., & Shaposhnik, Y. (2020). Understanding how dimension reduction tools work: An empirical approach to deciphering t-sne, umap, trimap, and pacmap for data visualization. *The Journal of Machine Learning Research*, *22*(1), 9129–9201. doi: 10.48550/arxiv.2012.04456

Ward, J. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, *58*, 236. doi: 10.2307/2282967

Wattenberg, M., Viégas, F., & Johnson, I. (2016). How to use t-sne effectively. *Distill*, *1*. doi: 10.23915/distill.00002

Yin, S., & Kaynak, O. (2015). Big data for modern industry: Challenges and trends [point of view]. *Proc. IEEE*, *103*, 143-146. doi: 10.1109/jproc.2015.2388958

Appendix A Laplacian Eigenmaps

Laplacian Eigenmaps (LEM) is a non-linear dimension reduction technique that focuses on the preservation of the local structure of the dataset during the transformation into a lower dimension. It is referred to as a topology-preserving technique rather than a distance-preserving one, which makes it more robust as it can alter the regions of the dataset in order to create a better representation in a lower dimensional space (J. A. Lee et al., 2007). The procedure taken by LEM is following as described by Du (2019). The first step is to construct a graph that represents the connections of data points within the dataset. The connection is established between a point and its closest k points. The pairwise similarity, given by Gaussian Kernel, has to be computed for the connected points, and for the points without connection, it is set to 0. The pairwise similarity of two connected observation vectors x_i and x_j is given by :

$$w_{(i,j)} = \exp \frac{\parallel (x_i - x_j)^2 \parallel}{2\sigma^2} \tag{8}$$

Where the σ^2 is the variance of the Gaussian Kernel. By calculating the pairwise similarities the matrix $\mathbf{W}$ is constructed. Afterwards the degree matrix is constructed, $\mathbf{D}$, which is given by $D_{ii} = \sum_{q=1}^{n} w_{(i,q)}$, for a dataset of size n. Then the Laplacian matrix, $\mathbf{L}$ is calculated given by $\mathbf{L} = \mathbf{D} - \mathbf{W}$. Given the Laplacian matrix the Eigenvectors and Eigenvalues need to be computed for a generalized eigenproblem:

$$\mathbf{LE} = \lambda \mathbf{DE} \tag{9}$$

Where the $\mathbf{E}$ is the matrix of Eigenvectors, and λ are eigenvalues. Lastly given the desired number of dimensions (m) for lower dimensional space, the m smallest (non-zero) eigenvalues are determined, and their eigenvectors are used as coordinates for the lower dimensional space.

Appendix B Internal Validation of Clustering Quality

B.1 Silhouette

The Silhouette value is determined by measuring the dissimilarity of the observation given by:

$$Silhouette(i) := \frac{b(i) - a(i)}{max(b(i), a(i)} \tag{10}$$

Where the b(i) is an average dissimilarity for observation i and all observations in the closest cluster. In the best case, this value is large. The a(i) is the average dissimilarity between

observation i and other observations in the same cluster, so ideally this value is low. In order to determine how good the overall clustering is the Silhouette is calculated for all the observations to find the average.

B.2 Dunn Index

The equation used to determine the value of the Dunn index is the following:

$$Dunn := \frac{min_{x \in C^i, y \in C^j, i \neq j} d(x, y)}{max_i(max_{x,y \in C^i} d(x, y)} \tag{11}$$

Where the numerator is the minimal distance between any two points from different clusters and the denominator calculates the maximal distance between any two points from the same cluster.

B.3 Calinski-Harabasz Index

The calculation of the Calinski-Harabasz index is given by:

$$Calinski - Harabasz := \frac{BCD}{WCD} * \frac{n-k}{k-1} \tag{12}$$

where the BCD is the between-cluster dispersion, which is measured by summing squared distances between cluster centroids and the centroid of the whole dataset. The WCD is within-cluster dispersion measured by summing squared distances between observations and their cluster centroids. The n is the total number of observations in the dataset, and k is the number of clusters.

B.4 Davies-Bouldin Index

The equation that is used to find the Davies-Bouldin index for a cluster C^i is given by:

$$Davies - Bouldin_i = max_{i \neq j} \frac{Si + S_j}{D_{i,j}} \tag{13}$$

Where S_i is the similarity measure in cluster C^i and $D_{i,j}$ is the distance between two clusters.

Appendix C Construction of Artificial datasets

The construction of the datasets is primarily inspired by Van Der Maaten et al. (2009), and the code provided by the researchers, (Maaten, 2013). The dataset generation was altered slightly for some of the datasets in order to create well-defined clusters within the datasets. Given two uniformly distributed random numbers on interval $[0, 1]$, p_i and q_i, and the dataset $\mathbf{X}$ is created as follows:

- **Swiss Roll:** Let $t_i = \frac{3\pi}{2} * (1+2p_i)$, the observation $x_=[t_i \cos(t_i), t_i \sin(t_i), 11q_i]$. Repetition of this process creates a Swiss Roll shaped dataset in 3D. Once the Swiss Roll with the desired number of observations is constructed, we remove the observations given condition: $-6 > X[, 3] > -5$. This is done to create a small boundary between clusters. Then the number of points removed are uniformly sampled from X. Lastly noise of size 0.01 is added to ensure no observations are identical.

- **Broken Swiss Roll:** $t_i = \frac{3\pi}{2} * (1+2p_i)$, the observation $x_=[(t_i/2) \cos(t_i), (3/2) \cdot t_i \sin(t_i), 11q_i]$. All observations for which $t_i \in \{\frac{2}{5}, \frac{4}{5}$ are rejected and resampled. Repetition of this process creates a Broken Swiss Roll shaped dataset in 3D. Then the dataset is scaled. Once the Broken Swiss Roll with the desired number of observations is constructed, we remove the observations given condition: $-0.8 > X[, 3] > -0.5 AND X[, 1] > 0$. Lastly, the noise of size 0.01 is added to ensure no observations are identical.

- **TwinPeaks:** $x_i = [1 - 2p_i, 1 - 2q_i, \sin(\pi - 2\pi p_i)]$, then to make the peaks denser the second dimension is multiplied 100 when it have a non-negative value and is a peak, which will result in denser peaks compared to the original Twinpeaks dataset.

- **Helix :** $x_i = [(2 + \cos(8p_i)) \cos(p_i), (2 + \cos(8p_i)) \sin(p_i), \sin(8p_i)]$

- **High Dimensional:**

Algorithm 1 High-dimensional dataset generation

Input: Total number of observations n, Desired number of dimensions no_dims, Scaling factor for random noise $noise$

Output: High-dimensional dataset X

Calculate the number of points per dimension: $no\_points\_per\_dim \leftarrow round(n^{(1/no\_dims)})$

Generate equally spaced points between 0 and 1: $l \leftarrow linspace(0, 1, no\_points\_per\_dim)$

Generate all combinations of points taken no_dims at a time: $t \leftarrow combn(l, no\_dims)$

Generate the high-dimensional dataset: **for** *each row in* t **do**

$x1 \leftarrow cos(t_{row,1})$ $x2 \leftarrow tanh(3 \cdot t_{row,2})$ $x3 \leftarrow t_{row,1} + t_{row,3}$ $x4 \leftarrow t_{row,4} \cdot sin(t_{row,2})$

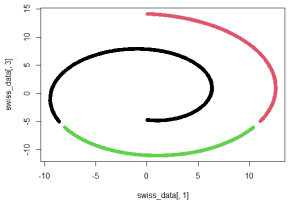
$x5 \leftarrow sin(t_{row,1} + t_{row,5})$ $x6 \leftarrow t_{row,5} \cdot cos(t_{row,2})$ $x7 \leftarrow t_{row,5} + t_{row,4}$ $x8 \leftarrow t_{row,2}$

$x9 \leftarrow t_{row,3} \cdot t_{row,4}$ $x10 \leftarrow t_{row,1}$ $X_{row,:} \leftarrow [x1, x2, x3, x4, x5, x6, x7, x8, x9, x10]$

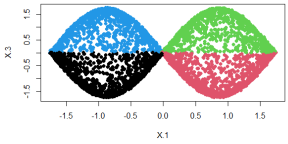
end

Add random noise to the dataset: $X \leftarrow X + noise \cdot randn(size(X))$

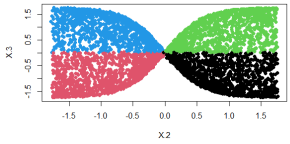
Some visualizations from different angles can also be viewed below:



(a) Swiss Roll



(b) TwinPeaks Dim 1 & Dim 2



(c) TwinPeaks Dim 2 & Dim 3

Figure 9: Visualizations of Swiss Roll and Twin Peaks from different angles

Appendix D Reproduction of Research

In this research, a number of tools have been used. The primary programming language is R (Team et al., 2016), (version 4.3.1), and an integrated development environment RStudio Allaire (2012), (version 2023.06.0+421). For a generation of one dataset, High-Dimensional, Matlab Matlab (2012), (version 9.12.0.1927505) was used.

Throughout this research there have been a number of steps taken that have randomization associated with them, such as sampling dataset, generating dataset, k-means, t-SNE, and UMAP. For t-SNE and UMAP when the cost functions are optimized Stochastic Gradient Descent is used, the order in which data points are processed or the specific batch size used can introduce randomness. In order to ensure that the results achieved can be reproduced random seeds have been used. Also, the code, datasets, and R workspaces are freely available at `https://github.com/ankaMkheidze/Navigating-Complexity-Thesis`. The functions that have been used and the inputs are given in Table 3.

Function	Library	Input	Description
NbClust()	NbClust	NbClust(data = DATA, distance = "euclidean", min.nc = 2, max.nc = 10, method = "kmeans", index ="all", alphaBeale = 0.1) NbClust(data = DATA, distance = "euclidean", min.nc = 2, max.nc = 10, method = "ward.D2", index ="all", alphaBeale = 0.1)	The function finds optimal number of Clusters
kmeans()	stats	kmeans(DATA, centers = k)	Implementation of k-means
hclust()	fastcluster	distance <- dist(DATA, method= "euclidean") agnes <- hclust(distance, method = "ward.D2")	Implementation of AGNES
prcomp()	stats	prcomp(DATA)	Implementation of PCA
Rtsne()	Rtsne	Rtsne(DATA, dims = d)	Implementation of t-SNE
umap()	uwot	umap(DATA, n_neighbors = nn , min_dist = md, n_components = d, init = "laplacian", learning_rate = lr)	Implementation of UMAP
silhouette()	cluster	silhouette(CLUSTERS, dist(DATA))	Calculation of Silhouette
dunn()	clusterValid	dunn(distance = NULL, CLUSTERS, Data = DATA, method = "euclidean")	Calculation of Dunn
calinhara()	fpc	calinhara(DATA, CLUSTERS, cn=max(CLUSTERS))	Calculation of Calinski- Harabasz
index.DB()	clusterSim	index.DB(DATA,CLUSTERS, d=NULL, centrotypes="centroids", p=2, q=2)$DB	Calculation of Davies-Bouldin
fviz_cluster()	factoextra		Visualization of Clusters
scatterplot3d()	scatterplot3d		3D visualization of Data

Table 3: Functions Used for Research

The functions for NbClust, k-means, PCA, t-SNE, and cluster visualization are the same as Renjith et al. (2021). The function for AGNES has been changed as the fastcluster package allows us to lower the computational time significantly. Renjith et al. (2021) used the cluster package for AGNES, which runs for approximately 3-5 minutes, whilst the fastcluster takes no more than 15 seconds. Given the times AGNES is used for tuning of UMAP and production of results for 6 datasets this change has lowered the computational strain significantly. Renjith et al. (2021) have used clusterCrit function for Internal Validation which is not supported by CRAN anymore due to its many flaws, thus the supported alternatives have been used. For Rtsne function it is unclear what initialization was used by Renjith et al. (2021), as PCA is the default for Rtsne and has been shown to outperform random initialization it has been used.

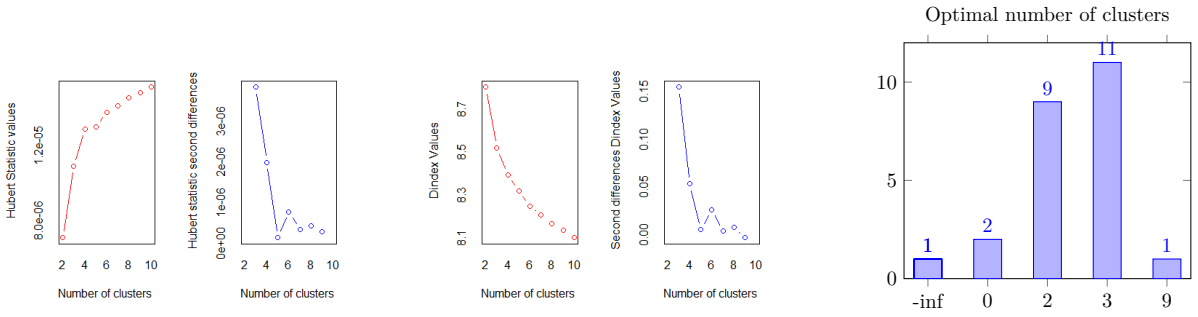
Appendix E Output of NbClust

E.1 k-means Method with Euclidean Distance Jester dataset

The output resulting by using the method k-means and Euclidean distance is the following, where 11 indexes indicated that 3 is the optimal number of clusters:

Scott	**Cindex**	**Ball**	**McClain**	**Dindex**	**KL**	**Silhouette**	**Duda**	**PseudoT2**	**SDbw**	**Hartigan**	**Marriot**	**TrCovW**	**TraceW**	**Friedman**	**Beale**	**Ratakowsky**	**CH**	**CCC**	**Rubin**	**DB**	**Frey**	**Hubert**	**PtBiserial**	**Dunn**	**SDindex**
-inf	2	3	2	9	2	3	3	3	2	3	5	3	3	3	2	3	2	3	3	2	2	2	1	0	0

Table 4: NbClust Results Jester Dataset with k-means

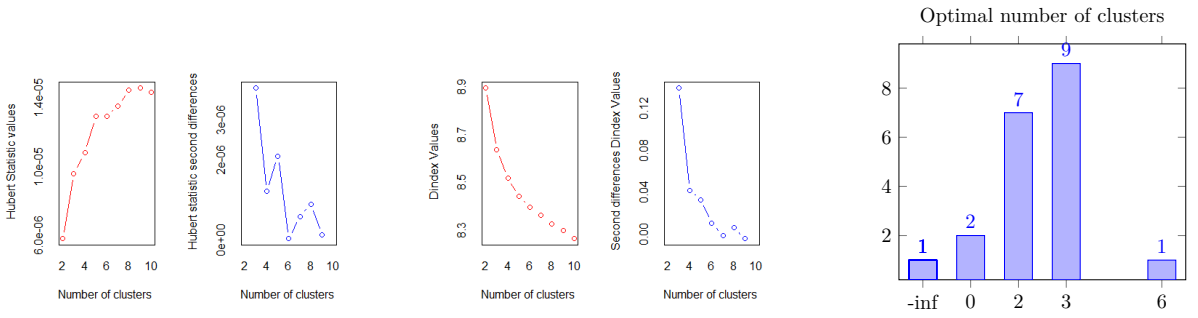


E.2 AGNES (Ward) Method with Euclidean Distance Jester dataset

The output for the NbClust when using the method Ward and Euclidean distance is the following, where 9 indexes indicated that 3 clusters are optimal:

Scott	**Cindex**	**Ball**	**McClain**	**Dindex**	**KL**	**Silhouette**	**Duda**	**PseudoT2**	**SDbw**	**Hartigan**	**Marriot**	**TrCovW**	**TraceW**	**Friedman**	**Beale**	**Ratakowsky**	**CH**	**CCC**	**Rubin**	**DB**	**Frey**	**Hubert**	**PtBiserial**	**Dunn**	**SDindex**
-inf	2	3	2	4	3	NA	NA	NA	3	3	6	3	3	3	2	3	2	3	5	2	2	2	1	0	0

Table 5: NbClust Results Jester dataset with AGNES

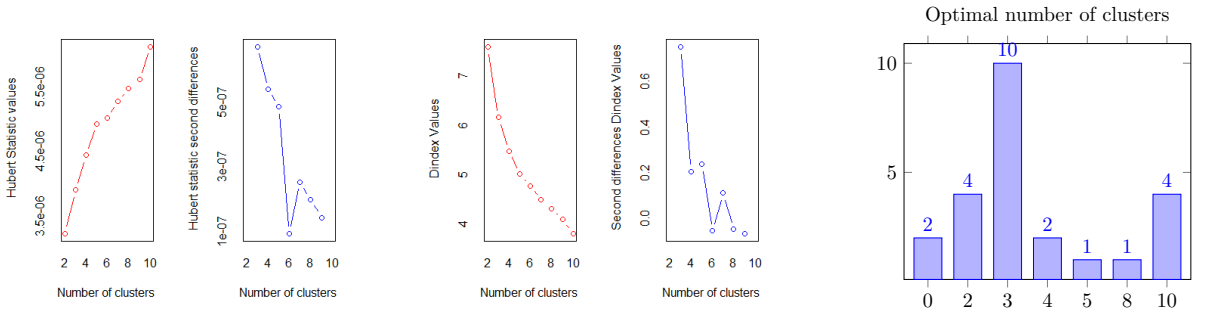


E.3 k-means Method with Euclidean Distance Swiss Roll

The output resulting by using the method k-means and Euclidean distance for the Swiss Roll dataset is the following, where 10 indexes indicated that 6 is the optimal number of clusters:

Scott	Cindex	Ball	McClain	Dindex	KL	Silhouette	Duda	PseudoT2	SDbw	Hartigan	Marriot	TrCovW	TraceW	Friedman	Beale	Ratakowsky	CH	CCC	Rubin	DB	Frey	Hubert	PtBiserial	Dunn	SDindex
3	10	3	2	0	10	3	8	2	10	3	5	3	3	10	2	3	4	4	5	3	1	0	3	8	3

Table 6: NbClust Output Swiss Roll k-means

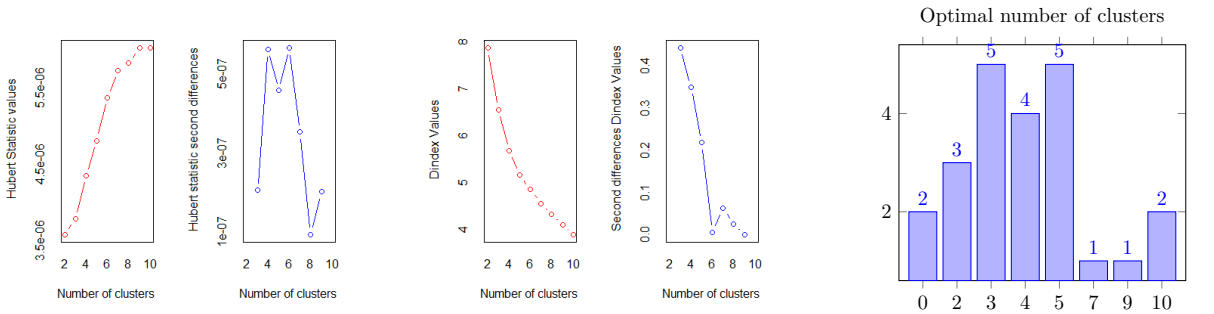


E.4 AGNES (Ward) Method with Euclidean Distance Swiss Roll

The output resulting by using the method AGNES with Ward linkage and Euclidean distance is the following, where 5 indexes indicated that 3 is the optimal number of clusters:

Scott	Cindex	Ball	McClain	Dindex	KL	Silhouette	Duda	PseudoT2	SDbw	Hartigan	Marriot	TrCovW	TraceW	Friedman	Beale	Ratakowsky	CH	CCC	Rubin	DB	Frey	Hubert	PtBiserial	Dunn	SDindex
4	10	3	2	0	5	2	NA	NA	10	3	5	3	3	9	2	3	5	5	5	4	1	0	4	7	4

Table 7: NbClust Output Swiss Roll AGNES

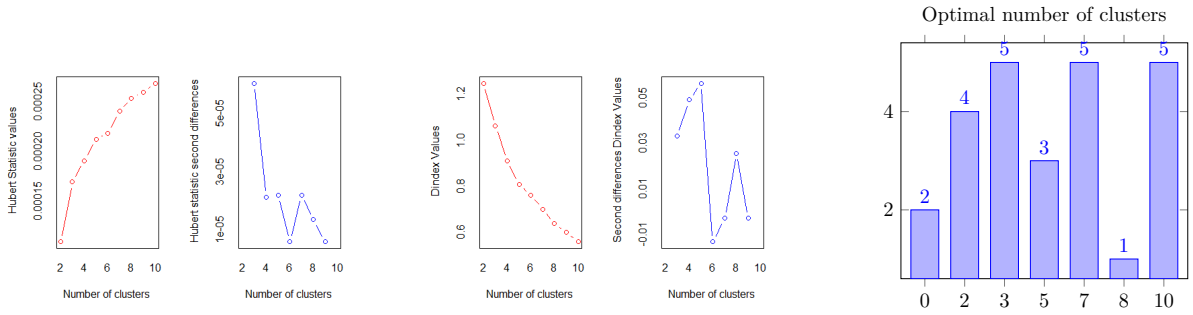


E.5 k-means Method with Euclidean Distance Broken Swiss Roll

The output resulting by using the method k-means and Euclidean distance for the Broken Swiss Roll dataset is the following, where 5 indexes indicated that 3 is the optimal number of clusters:

Scott	Cindex	Ball	McClain	Dindex	KL	Silhouette	Duda	PseudoT2	SDbw	Hartigan	Marriot	TrCovW	TraceW	Friedman	Beale	Ratakowsky	CH	CCC	Rubin	DB	Frey	Hubert	PtBiserial	Dunn	SDindex
5	10	3	5	0	7	7	2	2	10	3	5	3	3	5	2	3	10	10	7	10	1	0	7	8	7

Table 8: NbClust Output Broken Swiss Roll k-means

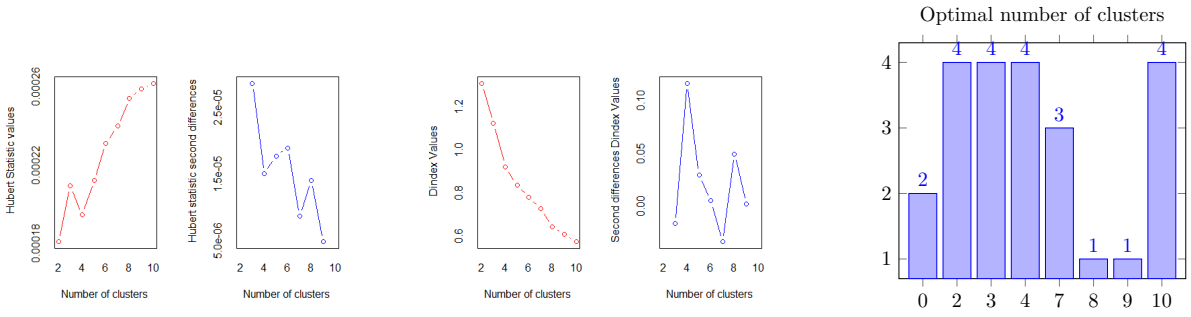


E.6 AGNES (Ward) Method with Euclidean Distance Broken Swiss Roll

The output resulting by using the AGNES method Ward linkage and Euclidean distance for Broken Swiss Roll dataset is the following, where 4 indexes indicated that 3 is the optimal number of clusters:

Scott	Cindex	Ball	McClain	Dindex	KL	Silhouette	Duda	PseudoT2	SDbw	Hartigan	Marriot	TrCovW	TraceW	Friedman	Beale	Ratakowsky	CH	CCC	Rubin	DB	Frey	Hubert	PtBiserial	Dunn	SDindex
3	9	3	2	0	2	7	NA	NA	10	4	3	4	4	3	2	4	10	10	8	10	1	0	7	2	7

Table 9: NbClust Output Broken Swiss Roll AGNES

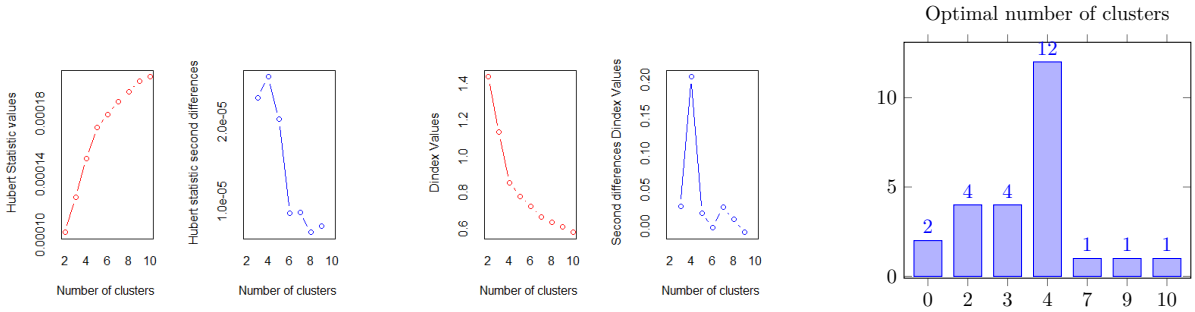


E.7 k-means Method with Euclidean Distance TwinPeaks

The output resulting by using the method k-means and Euclidean distance for TwinPeaks dataset is the following, where 12 indexes indicated that 4 is the optimal number of clusters:

Scott	Cindex	Ball	McClain	Dindex	KL	Silhouette	Duda	PseudoT2	SDbw	Hartigan	Marriot	TrCovW	TraceW	Friedman	Beale	Ratakowsky	CH	CCC	Rubin	DB	Frey	Hubert	PtBiserial	Dunn	SDindex
3	10	3	2	0	4	4	2	2	9	4	3	3	4	4	2	4	4	4	4	4	1	0	4	7	4

Table 10: NbClust Results Twin Peaks k-means

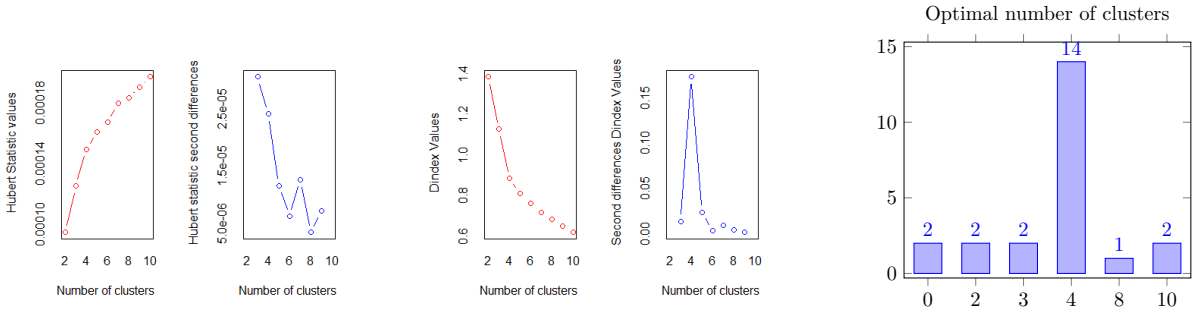


E.8 AGNES (Ward) Method with Euclidean Distance TwinPeaks

The output resulting by using the AGNES method with Ward linkage and Euclidean distance for TwinPeaks dataset is the following, where 14 indexes indicated that 4 is the optimal number of clusters:

Scott	Cindex	Ball	McClain	Dindex	KL	Silhouette	Duda	PseudoT2	SDbw	Hartigan	Marriot	TrCovW	TraceW	Friedman	Beale	Ratakowsky	CH	CCC	Rubin	DB	Frey	Hubert	PtBiserial	Dunn	SDindex
3	10	3	2	0	4	4	NA	NA	10	4	4	4	4	4	2	4	4	4	4	4	1	0	4	8	4

Table 11: NbClust Results for Twin Peaks dataset with AGNES

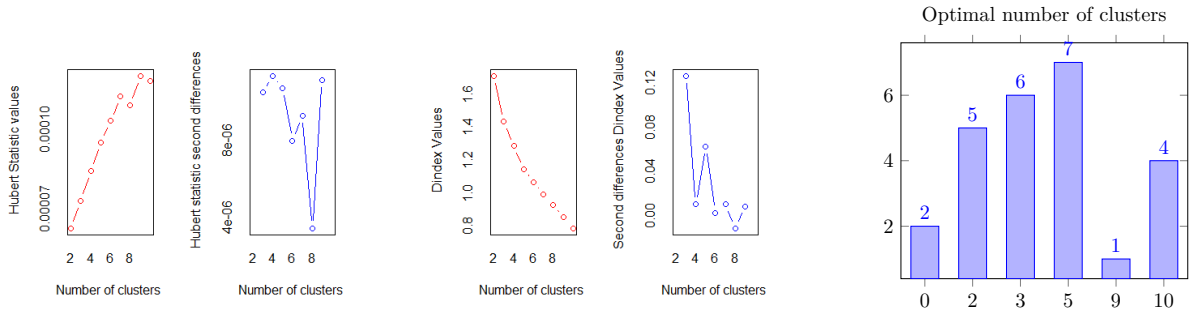


E.9 k-means Method with Euclidean Distance Helix

The output resulting by using the method k-means and Euclidean distance with Helix dataset is the following, where 7 indexes indicated that 5 is the optimal number of clusters:

Scott	Cindex	Ball	McClain	Dindex	KL	Silhouette	Duda	PseudoT2	SDbw	Hartigan	Marriot	TrCovW	TraceW	Friedman	Beale	Ratakowsky	CH	CCC	Rubin	DB	Frey	Hubert	PtBiserial	Dunn	SDindex
5	2	3	2	0	5	10	2	2	10	3	5	3	3	9	2	3	5	10	5	5	1	0	5	10	3

Table 12: NbClust Output for Helix dataset and k-means method

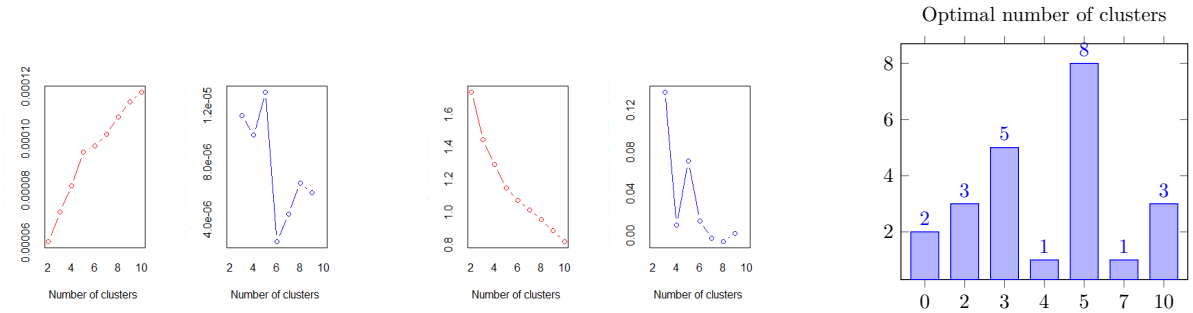


E.10 AGNES (Ward) Method with Euclidean Distance Helix

The output resulting by using the AGNES method with Ward linkage and Euclidean distance for Helix dataset is the following, where 8 indexes indicated that 5 is the optimal number of clusters:

Scott	Cindex	Ball	McClain	Dindex	KL	Silhouette	Duda	PseudoT2	SDbw	Hartigan	Marriot	TrCovW	TraceW	Friedman	Beale	Ratakowsky	CH	CCC	Rubin	DB	Frey	Hubert	PtBiserial	Dunn	SDindex
3	2	3	2	0	5	10	NA	NA	10	5	5	3	3	4	2	3	5	10	5	5	1	0	5	7	5

Table 13: NbClust Output for Helix dataset and Ward method

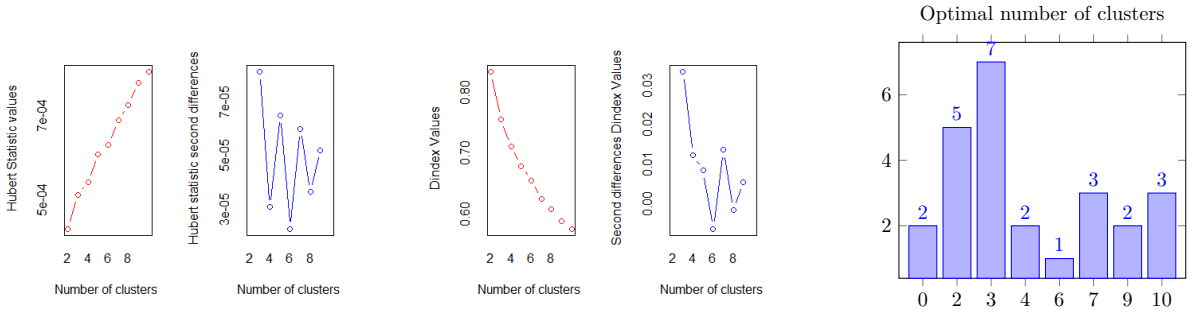


E.11 k-means Method with Euclidean Distance High Dimensional

The output resulting from using the method k-means and Euclidean distance for High Dimensional dataset is the following, where 7 indexes indicated that 3 is the optimal number of clusters:

Scott	Cindex	Ball	McClain	Dindex	KL	Silhouette	Duda	PseudoT2	SDbw	Hartigan	Marriot	TrCovW	TraceW	Friedman	Beale	Ratakowsky	CH	CCC	Rubin	DB	Frey	Hubert	PtBiserial	Dunn	SDindex
3	4	3	2	0	7	3	2	2	10	3	7	3	3	6	2	4	2	10	7	9	1	0	3	10	9

Table 14: NbClust Results for High-Dimensional set with k-means



E.12 AGNES (Ward) Method with Euclidean Distance High Dimensional

The output resulting by using the method AGNES with Ward linkage and Euclidean distance for High Dimensional dataset is the following, where 8 indexes indicated that 3 is the optimal number of clusters:

Scott	Cindex	Ball	McClain	Dindex	KL	Silhouette	Duda	PseudoT2	SDbw	Hartigan	Marriot	TrCovW	TraceW	Friedman	Beale	Ratakowsky	CH	CCC	Rubin	DB	Frey	Hubert	PtBiserial	Dunn	SDindex
3	10	3	2	0	3	2	NA	NA	10	3	8	3	3	7	10	4	3	2	3	10	1	0	6	8	4

Table 15: NbClust Results for High-Dimensional set with AGNES

Appendix F UMAP Tuning results

As mentioned in Section 3.2.3, there are several hyperparameters that needed tuning. These hyperparameters are the number of neighbors, minimum distance, and learning rate. Some of the other parameters such as the number of epochs were left as the default values. The best hyperparameters are determined by a limited grid search. The parameters used are: learning rate $= \{0.5, 1, 5\}$, number of neighbours $= \{15, 25, 50\}$ and minimum distance $= \{0.5, 0.1, 0.05, 0.001\}$, which all are within the acceptable ranges proposed by McInnes (2018). The results for the hyperparameter tuning are given in Table 16. The way grid search was structured was the embedding was calculated and the evaluation was done by clustering of the resulting data using k-means and AGNES and evaluating the clustering using the Internal Validation Indeces, given in Section 3.3. After the best combination was determined for both clustering methods and all the indexes the voting system is used. This system was simply looking at the best five combinations and seeing which one occurs more frequently and with what ranking. For example, if a combination is at first place for validation index one and clustering one it is awarded five points, and the combination with the highest points is chosen.

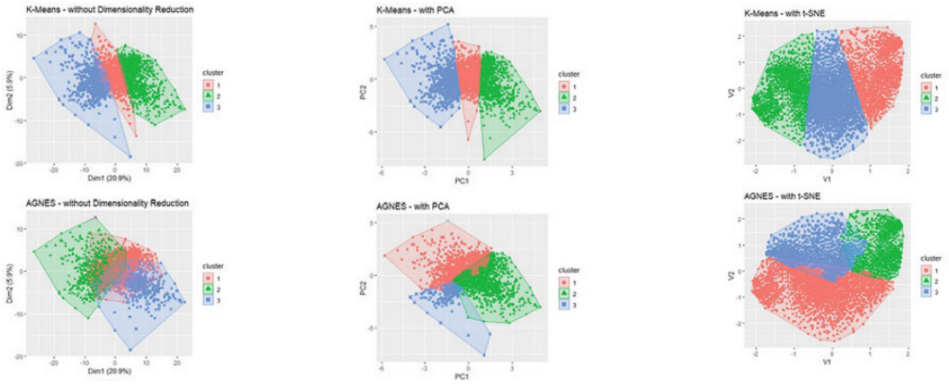
DataSet	Number Of Neighbours	Minimum Distance	Learning Rate
Jester	15	0.001	5.0
Swiss Roll	50	0.001	5.0
Broken Swiss Roll	50	0.05	5.0

DataSet	Number Of Neighbours	Minimum Distance	Learning Rate
Helix	25	0.05	5.0
TwinPeaks	25	0.001	5.0
HD	15	0.001	5.0

Table 16: UMAP Hyperparameters

Appendix G Renjith et al. (2021) Visualizations

In this section, the Visualizations of clusters produced by Renjith et al. (2021) are presented.



(a) No DR (b) PCA (c) t-SNE

Figure 22: Visualizations from Renjith et al. (2021). First row is k-means and Second row is AGNES.

Appendix H Visualizations 3D

In this section, the visualizations in 3 dimensions are presented. These clusterings were attained by first transforming data into lower dimensions, and producing clusters. These clusters are then applied to the data in 3-dimensions.

H.1 Swiss Roll

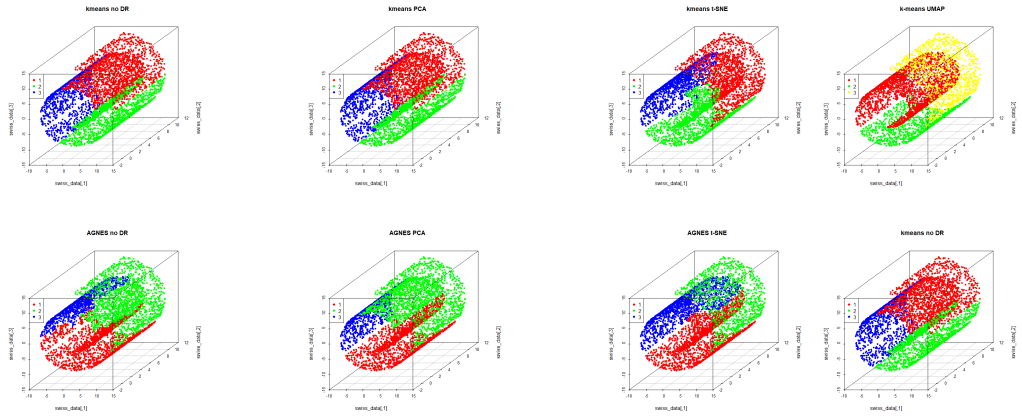


Figure 23: Visualizations of Clusters formed by DR techniques for Swiss Roll dataset.

H.2 Broken Swiss Roll

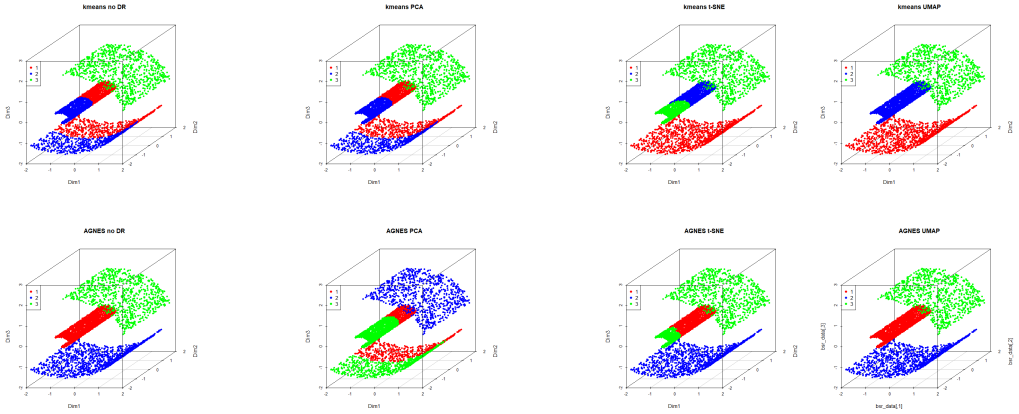


Figure 24: Visualizations of Clusters formed by DR techniques for Broken Swiss Roll dataset.

H.3 Twin Peaks

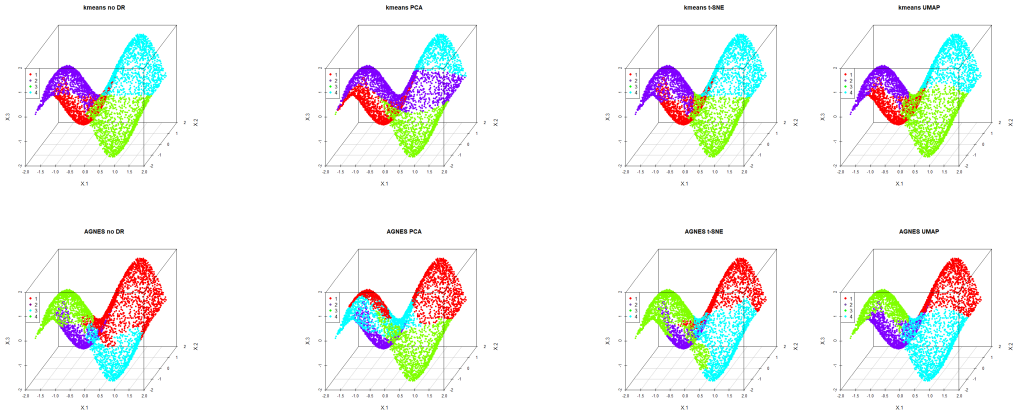


Figure 25: Visualizations of Clusters formed by DR techniques for twin peaks. dataset size is 5000

H.4 Helix

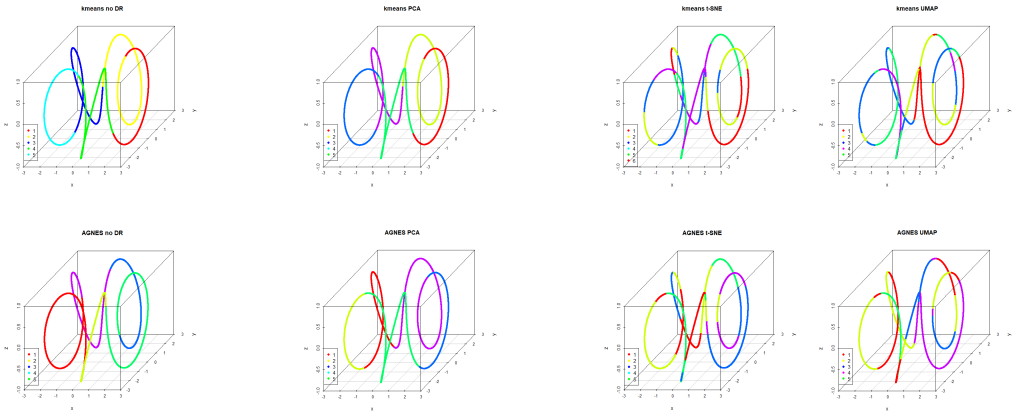


Figure 26: Visualizations of Clusters formed by DR techniques for Helix dataset. dataset size is 5000