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

Validating the Clustering Quality of K-Means Clustering and Agglomerative Hierarchical Clustering by means of Multiple Dimension Reduction Techniques

Dilara Genç (494808)

Supervisor:	J. Durieux
Second assessor:	M. Khismatullina
Date final version:	2nd July 2023

Validating the Clustering Quality of K-Means Clustering and Agglomerative Hierarchical Clustering by means of Multiple Dimension Reduction Techniques

Dilara Genç (494808dg)

Erasmus University Rotterdam
Burgermeester Oudlaan 50
Rotterdam, the Netherlands
494808dg@student.eur.nl

Abstract. In order to find the "best-performing" data transformation technique this research makes use of two linear dimension reduction techniques, namely Principal Component Analysis and Independent Component Analysis, and four non-linear dimension reduction techniques, called Isometric mapping, t-Distributed Stochastic Neighbor Embedding, Locally Linear Embedding, and Uniform Manifold Approximation and Projection. For the purpose of validating these dimension reduction techniques, the clustering methods k-means and Agglomerative Hierarchical Clustering were used. The obtained clusters were then internally validated with the Dunn Index, Calinski-Harabasz Index, Silhouette Index, and Davies-Bouldin Index. Intriguingly, the optimal cluster count $k = 2$ showed promising results. There also were some implications that the dimension reduction technique t-Distributed Stochastic Neighbor Embedding was the "best" method, but more research was needed due to different perplexity values influencing the t-SNE results.

Keywords: Cluster Optimization · Clustering Validation · Principal Component Analysis · Independent Component Analysis · t-Distributed Stochastic neighbor Embedding · Locally Linear Embedding · k-means Clustering · Agglomerative Hierarchical Clustering · Clustering Analysis · Dimensionality Reduction

1 Introduction

The data science and business analytics fields have become more refined throughout the years. That causes business owners to obtain very useful information from their tremendously huge data set. An example is the killer app for the field of data mining, made by a team from an e-commerce company [19,20]. The huge data sets obtained by the e-commerce company due to the clicking history of Web activity [21], results in the need for various complex data analytic techniques. These huge data sets can result in some problems, such as the curse of dimensionality which occurs when data scientists try to visualize and analyze high-dimensional data.

Therefore, this research reproduces the results found in [35], which took many data analytic techniques into consideration in order to tackle the problems that can occur during the use of huge databases, mainly focusing on dimension reduction and clustering techniques. This work extends the work of [35] by also implementing the dimension reduction techniques, Isometric mapping, and Uniform Manifold Approximation and Projection. This work aims to validate the quality of the k-means and the Agglomerative Hierarchical clustering techniques by using multiple dimension reduction techniques to find the best-performing technique.

Hence the research question in this work is: **which method is the best dimension reduction technique?** This is researched by using a multiple-step approach. Starting with obtaining the optimal cluster count for the data set in Sect. 3. Then, the data transformation is either done through Principal Component Analysis, Independent Component Analysis, Isometric Mapping, t-Distributed Stochastic neighbor Embedding, Locally Linear Embedding, or Uniform Manifold Approximation and Projection. The clustering step is done with the k-means or the Agglomerative Hierarchical clustering techniques. Lastly, end with the internal cluster validation techniques to find the best performing techniques.

Due to the results found in the work of [35], the t-Distributed Stochastic neighbor Embedding is expected to be the best-performing dimension reduction technique. But other results could occur during the process of this research because the Isometric mapping and Locally Linear Embedding techniques are also recognized as well-performing techniques.

This work consists of Sect. 2, which summarizes the results of relevant research. Sect 3 gives an elaborate explanation of the used data. Sect 4 dives deep into the four-step research approach in this work and gives a detailed description of the used methods. Sect 5. shows the obtained results in this work. Lastly, Sect. 6 gives a summary and concludes the found results in Sect. 5.

2 Related Work

There are many dimensionality reduction techniques available, both linear and non-linear. Each technique has its own requirements for the different types of

data available. The t-Distributed Stochastic neighbor Embedding (t-SNE) introduced by [27,26], which is based on the Stochastic neighbor Embedding (SNE) developed by [17], is an example of a non-linear technique. This technique showed very intriguing results in the recent work of [35]. The results of [35] showed that the t-Distributed Stochastic neighbor Embedding technique is indeed a state-of-the-art approach for dimensionality reduction, even more so if combined with the k-means and Agglomerative Hierarchical clustering methods.

Besides the t-Distributed Stochastic neighbor Embedding technique there are many other well-performing non-linear dimensionality reduction methods as well in order to resolve the limitations of Principal Component Analysis [28]. Another example of a non-linear technique is Isometric Mapping, which makes use of Dijkstra's algorithm [7] or the Floyd-Warshall algorithm [9]. The work of [28] showed that the non-linear techniques, such as Isometric Mapping, perform worse on real-world tasks but do very well on artificial tasks. This means that techniques such as Isometric Mapping do have some limitations. An example of a limitation of Isometric mapping is that this technique has fixed viewing angles, often a 30-degree angle. This limitation makes it difficult to portray objects from different angles, which means that this method is not the best at expressing the relationships in a huge data set [45]. This limitation also shows that Isometric Mapping may be improved.

The work of [11] noted that Principal Component Analysis [1,18] and Hierarchical Clustering Analysis [23] is a widely used combination of techniques in research. The work of [25] tries to optimize the Principal Component Analysis technique. The results of [25] show that optimization of the Principal Component Analysis is successful and shows that the technique performs very accurately. [25] also stated that Principal Component Analysis performs better if used for dimensionality reduction in the deep learning framework. Therefore, it is very intriguing to see how this technique performs on a rating data set discussed in Sect. 3.

The techniques described above show very interesting results in the research mentioned. For that reason, it is very enlightening to combine those methods with other linear and non-linear dimensionality reduction techniques in order to see which technique performs the best. To add more depth to the research different clustering methods are used as well. Further explanation of the dimensionality reduction techniques and clustering methods can be found in Sect. 4.2 and Sect. 4.3 respectively.

3 Data

For this research, the Jester data set 1 from the work of [10] is used. The data can be downloaded from here: `https://eigentaste.berkeley.edu/dataset/`. The data consists of the Jester jokes data set and the Jester rating data set. A brief explanation of these data sets can be found in Table 1 and Table 2 respectively. The jokes data set of the Jester data set 1 contains in total 100 HTML files,

where each file contains the joke that an user gave a rating to, which resulted in the Jester rating data set from the period between April 1999 to May 2003.

Table 1: Jokes data of the Jester data set 1 - a brief explanation

Data set	Jester_dataset_1_joke_texts
File count	100 files
File format	HTML (.html)
File name explanation	Every file has the name init1.html, ..., init100.html, where the number refers to the ID of the jokes in the Excel file

The rating data set retrieved from the Jester data set 1 [10] is an Excel file that contains the ratings of 73,421 users in total, where each user rates 100 jokes. The ratings ranged from -10.00 to +10.00 with a value of 99 corresponding to a joke not being rated. Due to missing ratings because of users not rating all 100 jokes, there are approximately 4.1 million useful anonymous ratings from the 73,421 users used for this work. For this research, the full rating data from the Jester data set is cleaned from the 99 values. Thereafter, a random sample of 5,000 observations were taken from the 4.1 million useful anonymous ratings.

The resulting data set was used for conducting the whole research in this paper. Table 2 shows the descriptive statistics of the 5,000 random samples from the Jester rating data set 1. Table 2 shows that the rating range is between -9.81 and 9.22, the standard deviation is -5.15, and on average, the users gave a rating of 1.86. These descriptive statistics indicate that the user ratings are, on average, positively skewed because the mean value is closer to the maximum value instead of the minimum value. This means that on average the users have a tendency to be more positive in their ratings. The standard deviation of 5.15 gives insights into the dispersion of the used rating data. Due to the standard deviation being moderately centered in the middle of the rating range, it shows that there is a moderate degree of variability in the ratings obtained from the users.

Table 2: Descriptive statistics of the subset of the Jester rating data set

Data set	Subset from jester_dataset_full.xlsx
File format	Excel (.xls)
Observations	5,000
Minimum Rating	-9.81
Maximum Rating	9.22
Mean Rating	1.86
Standard Deviation	5.15

4 Methodology

This research used the same four-step approach as [35]. For a visualization of the used steps, see Fig. 1. starting with Sect. 4.1 which explains how the optimal clus-

ter count (k) was obtained. Thereafter Sect. 4.2 explains the data transformation methods used for dimensionality reduction. Sect. 4.3 follows by describing the clustering analyzing methods used to verify the data transformation methods. Afterwards, in Sect 4.5 we find the explanation for the internal cluster validation methods used for evaluating the quality of the clustering methods. Lastly Sect. 4.5 explains the tools used to implement the whole research. A short overview of the four-step approach in this work can be seen from Fig. 1.

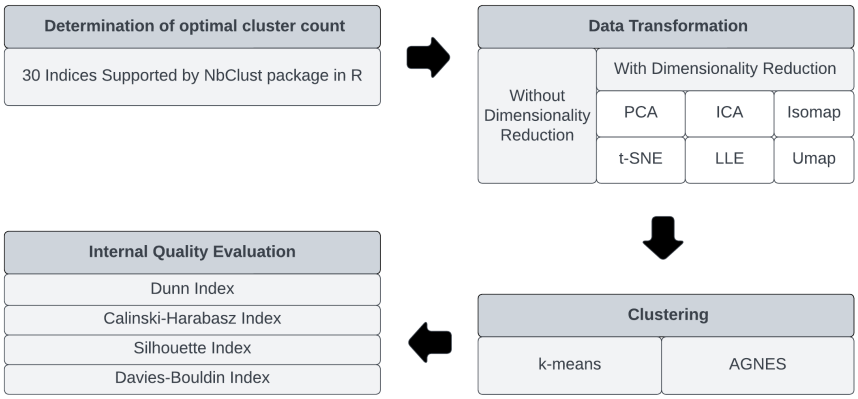


Fig. 1: Four-step approach used in this research

4.1 Optimization of Cluster Count (Step 1)

Before the clustering algorithms in Sect. 4.3 can be used, the cluster count (k) needs to be specified. In order to obtain the optimal cluster count in this work the NbClust() function from the NbClust package in R was used. The determination of the optimal cluster count is supported by the 30 indices provided by the NbClust package and by varying combinations of cluster counts. The optimal cluster count was then validated by the majority rule, Dunn Index, and Hubert index. A more elaborate explanation of the R package NbClust can be found in [4].

4.2 Data Transformation with Dimensionality Reduction (Step 2)

This section elaborates more on the used linear and non-linear dimension reduction techniques. In total six methods were used for the data transformation step. Two of the six dimension reduction techniques are linear, and the remaining four are non-linear.

Principal Component Analysis. Principal Component Analysis (PCA) was invented by [33] and later further developed by [18,1]. PCA is the first linear dimension reduction technique in this work, that is very useful for large data sets. The objective of data transformations by means of PCA is to transform the existing variables into a different set of variables by means of a linear combination

of the existing variables, without losing useful information [30]. This work makes use of the prcomp() function in R [43], which was pre-installed in the R program. The steps followed for PCA can be found in Algorithm 1.

Algorithm 1 Principal Component Analysis

Input: Data matrix (X) with n observations and p variables
Output: Matrix (Y) containing the principal components of the transformed data

–

Step 1 - Standardise: Standardise all the variables in X.
Step 2 - Covariance Matrix: Calculate the covariance matrix C_{xx} such that the correlations can be identified.
Step 3 - Eigenvectors: Calculate the eigenvalues λ_k and corresponding eigenvectors u_k of C_{xx} in order to identify the principal components.
Step 4 - Feature Vector: Create the feature vector in order to select the principal components that will be kept.
Step 5 - Principal Components: Remodel the data along the axes of the found principal components.

In Algorithm 1 the X was the rating data from the Jester data set 1. Y was the reduced data set obtained after performing the PCA technique. The number of observations n, also known as the users, is 73,421 and there are 100 variables p each representing the number of the joke. Step 3 in Algorithm 1 for the PCA technique is based on the eigenvalue equation in Eq. (1). C_{xx} was the covariance matrix, calculated in step 2. λ_k was the k^{th} eigenvalue with u_k as its corresponding k^{th} eigenvector and I being an identity matrix.

$$(C_{xx} - \lambda_k I)u_k = 0 \tag{1}$$

Independent Component Analysis. Independent Component Analysis (ICA) is the second linear dimension reduction technique in this work, first introduced by [13] in a very general form. Later, this technique was further researched and developed by [14,15,16]. But ICA only got popular a few years later through the work of [5]. More recent findings on ICA can be found in [32,42]. The goal of ICA is to linearly rotate the data while retrieving statistically independent components. In order to do ICA it needs to be assumed that there are non-Gaussian attributes and that these attributes need to be independent from each other. This research used the fastICA() function from the fastICA package in the program R [43]. The steps followed in order to implement ICA can be found in Algorithm 2.

Algorithm 2 Independent Component Analysis

Input: Data matrix (X) with n observations and p variables
Output: Matrix (Y) containing the independent components of the transformed data
Assume: (i) independent attributes and (ii) non-gaussian attributes

–

Step 1 - Decomposition Decompose matrix X into the mixing matrix A of the components and signal matrix s, where $s \approx \hat{s}$ in Eq. (2), such that $X = As$.
Step 2 - Whitening Whitening of X with the help of Eq. (3) such that the E^T rotates such that the entropy of Eq. (4) is maximized, resulting in non-gaussian attributes and a whitened data set X_W.
Step 3 - Independent Components The independent components, used for Y, are obtained.
Step 4 - Reduced Data The output matrix Y is made, where $Y = A_k.S_k$.

$$\hat{s} = WX \tag{2}$$

$$X_W = (D^{-\frac{1}{2}}E^T)X \tag{3}$$

In Algorithm 2 W is an approximation of A^{-1}. E and D were the eigenvectors and eigenvalues obtained from the covariance matrix of X. For more information on the whitening step consult [39]. Y is the reduced data set obtained by selecting the top independent components through A_k and s_k, the mixing and basis matrices containing the top independent components. The number of observations n, also known as the users, is 73,421 and there are 100 variables p each representing the number of the joke. Non-gausianity in Algorithm 1, used in the fastICA() function, was obtained through maximizing negentropy (J). Eq. (4) and Eq. (5) show how J was found.

$$J(v) = H(v_{gaussian}) - H(v) \tag{4}$$

where,

$$H(v) = -\int f(v)log(f(Y))dv \tag{5}$$

with $v = (v_1, ..., v_n)$ which was a vector containing random variables, where the random variable has density $f(.)$. $v_{gaussian}$ is a Gaussian random variable. $v_{gaussian}$ has the same density structure as v.

Isometric Mapping. Isometric Mapping (Isomap) is the first non-linear dimensionality reduction technique in this work. This method is used when low-dimensional embeddings need to be calculated from high-dimensional data points in a data set. Isomap might not be as accurate as LLE according to [40], but it is still a very efficient technique that can be used to interpret a wide range of dimensionalities and data sets. This research makes use of the isomap() function from the vegan package in R [43]. In order to implement Isomap the isomap() function made use of the steps in Algorithm 3.

Algorithm 3 Isometric Mapping

Input: Data matrix (X) with n observations and p variables
Output: Matrix (Y) containing the principal components of the transformed data

–

Step 1 - Neighborhood Graph Use the k Nearest neighbor (KNN) approach the find the nearest neighbor for each data point. Then develop the neighborhood graph, where the data points are connected to each other if they are neighbors.
Step 2 - Geodesic Distance Obtain the geodesic distances, also known as the shortest path between two data points.
Step 3 - MDS Apply Multi Dimensional Scaling (MDS) in order to obtain the lower-dimensional embeddings, then resulting in the reduced data set Y.

MDS visualized the similarities between individual points within a data set. More information about the MDS method can be found in [12].

t-Distributed Stochastic neighbor Embedding. t-Distributed Stochastic neighbor Embedding (t-SNE) is the second non-linear dimension reduction technique in this work. This technique reduces the data from a higher-dimension to a lower-dimensional, such that the data can be visualized into a three- or less-dimensional space. t-SNE is based on the SNE technique developed by [17]. Later, followed by the t-distributed version created by [27,26]. An advantage of the technique developed by [27] is that t-SNE keeps the local structure intact during the dimension reduction phase meaning that the overall geometry of the data does not change after the transformation. Thus, t-SNE works well in visualizing high-dimensional data. In order to obtain an optimal lower dimensional space gradient descent is used. Note that starting with lowering the data dimension and then followed by clustering could give varying results, due to the different perplexity values that could be used for t-SNE. This research makes use of the Rtsne() function from the Rtsne package in R [43], with a perplexity of 10. Where the perplexity balances the aspects between the local and global structure of the used data set. Therefore, changing this value would affect the obtained t-SNE results. The t-SNE technique minimizes the Kullback-Leibler formula, Eq (6), from the similarities between data points i and j. In the low dimensional space defined as q_{ij}, Eq. (7), and in the high dimensional space defined as p_{ij}, Eq. (8). The steps that were taken in order to implement t-SNE can be found in Algorithm 4.

$$Kullback.Leibler = \sum_{i \neq j} p_{ij} log(\frac{p_{ij}}{q_{ij}}) \tag{6}$$

where,

$$p_{ij} = \frac{\exp \frac{-||x_i - x_j||^2}{2\sigma^2}}{\sum_{k \neq l} \exp \frac{-||x_k - x_l||^2}{2\sigma^2}} \tag{7}$$

$$q_{ij} = \frac{\exp -||y_i - y_j||^2}{\sum_{k \neq l} \exp \frac{-||y_k - y_l||^2}{2\sigma^2}} \tag{8}$$

Algorithm 4 t-Distributed Stochastic neighbor Embedding

Input: Data matrix (X) with n observations and p variables
Parameters: Set perplexity to value 10
Output: Matrix (Y) containing the transformed low-dimensional data

–

Step 1 - Pairwise Affinities Compute p_{ij} with perplexity 10, with Eq. (7).
Step 2 - Initialization Set $p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n}$ and have an initial solution for Y.
Step 3 - Low-Dimensional Affinities Compute q_{ij} with Eq. (8).
Step 4 - Gradient Minimize Eq. (6) and use that result to find a new Y.
Step 5 - Repetition Repeat step 3 and step 4 for a 1000 times, resulting in a final Y.

Locally Linear Embedding. Locally Linear Embedding (LLE) is the third non-linear dimension reduction technique used in this work, which was developed by [38,36]. LLE is sensitive to outliers and noise, therefore the data set is cleared from these before the use of the LLE technique. The LLE technique transformed the high-dimensional space of the Jester rating data into a smaller dimension, through the use of linear combinations from multiple points in the data set, also known as neighbors. This work used the lle() function from the lle package in R, in order to implement the LLE technique. The exact steps for the LLE technique can be found in Algorithm 5.

Algorithm 5 LLE

Input: Data matrix (X) with n observations and p variables
Output: Matrix (Y) containing the the transformed data

–

Step 1 - KNN: Use the k Nearest neighbor (KNN) approach the find the nearest neighbor for each data point.
Step 2 - Weight Matrix: Construct the weight matrix W. Determine W by minimizing the error of the cost function, see Eq. (9), where each point is found through a linear combination of its neighbor.
Step 3 - Positioning: Find the positioning of each data point in the newly found lower dimensional embedding. This is done by minimising cost function C_y, see Eq. (10)

Step 2 in Algorithm 5 for the LLE technique is based on the cost function in Eq. (9), where x_i was data point i from X and w_{ij} was element i, j from W Step 3 in Algorithm 5 is based on minimizing the cost function C_y in Eq. (10).

$$W = \sum_{i=1}^{n} ||x_i - \sum_{j=1}^{n} w_{ij} x_j||^2 \tag{9}$$

$$C_y = \sum_{i=1}^{n} ||y_i - \sum_{j=1}^{n} w_{ij} y_j||^2 \tag{10}$$

Uniform Manifold Approximation and Projection. The Uniform Manifold Approximation and Projection (Umap) is the last non-linear dimension reduction technique used in this research. Umap developed by [31] has no embedding dimension restrictions, making it a very useful dimension reduction technique. This work makes use of the umap() function in the umap package in R [43]. Algorithm 6 shows the steps taken in order to implement Umap.

Algorithm 6 Uniform Manifold Approximation and Projection

Input: Data matrix (X) with n observations and p variables
Output: Matrix (Y) containing the the transformed data

–

Step 1 - KNN Use the k Nearest neighbor (KNN) approach the find the nearest neighbor for each data point
Step 2 - Neighbor Graph Develop the neighborhood graph, where the data points are connected to each other if they are neighbors
Step 3 - Dimensional representation Find the low-dimensional representation through the minimum distances between the data points and then obtain the reduced data set Y.

Note that Umap and t-SNE have a similar workflow according to [2]. These methods are even considered approximately the same if the ρ in Eq. (3) in [2] is manipulated correctly.

4.3 Clustering (Step 3)

This research checked the effect of dimensionality reduction techniques on the clustering quality by using two clustering methods. These techniques are k-means Clustering and Agglomerative Hierarchical Clustering. R packages were used in order to implement these algorithms.

k-Means Clustering. k-means clustering is an unsupervised machine learning method by [41,29,22] that makes use of a vector quantization technique. This method clusters the data set, consisting of p observations, into k. This work made us of the kmeans() function that is pre-installed in R [43]. Algorithm 7 shows the steps that were taken by the kmeans() function in order to implement k-means clustering.

The iterations in the k-means clustering method are done such that the Total Within-Cluster Variation (TWCV) in Eq. (11) was minimized, for the chosen clustering count k. In Eq. (11) K is the used cluster count k, E_i is the in-cluster object of C_k, with centroid μ_k. K-means is sensitive to local minima. Thus in order to find the best cluster assignments Algorithm 7 has been repeated 1000 times in this work.

$$TWCV = \sum_{k=1}^{K} \sum_{E_i = C_k} (E_i - \mu_k)^2 \tag{11}$$

Algorithm 7 k-Means

Input: Matrix (Y) containing the the transformed data
Output: Vector (Q) containing the clustering information

–

Step 1 - Cluster Count Select the optimal cluster count k.
Step 2 - Centroids Randomly select k centroids.
Step 3 - Form Clusters Assign each observation from the Y to their closest centroid, in order to form the k clusters.
Step 4 - Variance Find the variance in order to find new centroids.
Step 5 - Repeat Repeat step 3 and 4 until there is no more reassignment of clusters.

Agglomerative Hierarchical Clustering. Agglomerative Hierarchical Clustering (AGNES) is a clustering technique introduced by [24]. AGNES makes a cluster hierarchy through the bottom-up approach. The bottom-up approach means that each data point starts a cluster. Then these clusters are merged when the data point is moved up in the hierarchy, from the bottom. This work made use of the agnes() function from the cluster package in R [43]. Algorithm 8 shows the steps that were taken by the agnes() function in order to implement the AGNES clustering in this work.

Algorithm 8 Agglomerative Hierarchical clustering

Input: Matrix (Y) containing the the transformed data
Output: Vector (Q) containing the clustering information

–

Step 1 - Distance Matrix Find the distance matrix of Y.
Step 2 - Minimum Distance Calculate the minimum distance in the matrix of Y.
Step 3 - Combine Combine the two clusters that are nearest to each other, using the linkage method average.
Step 4 - Center Find the centroid of the new;y obtained cluster.
Step 5 - Repeat Repeat steps 2 up until and including step 4, until one cluster remains.

4.4 internal Cluster Validation (Step 4)

In order to check the quality of the clusters, obtained through the different clustering techniques combined with the dimensionality reduction techniques,

cluster validation methods were used. This work made use of the Dunn Index [8], Calinski-Harabasz Index [3], Silhouette Index [37], and Davies-Bouldin Index [6].

Dunn Index. The Dunn Index is the ratio between the smallest distance between two centroids and the largest distance between two objects in any cluster. In this work the Dunn Index was obtained by using the dunn() function of the clValid package in R [43]. Eq. (12) shows how the Dunn Index of object i was calculated. $\min_{1\leq i\leq j\leq m}\delta(C_i, C_j)$ was the smallest among-cluster distance object i and object j. $\max_{1\leq k\leq m}\triangle_k$ was the largest intra-cluster distance.

$$D_{Index} = \frac{\min_{1\leq i\leq j\leq m}\delta(C_i, C_j)}{\max_{1\leq k\leq m}\triangle_k} \tag{12}$$

Calinski-Harabasz Index. The Calinski-Harabasz Index is the ratio between the sum of inter-cluster dispersion and the sum of the intra-cluster dispersion. The Calinski-Harabasz Index is also called as the Variance Ratio Criterion (VRC). In this work the Calinski-Harabasz Index was obtained by using the calihara() function of the fpc package in R [43].

Eq. (13) shows how the Calinski-Harabasz Index was obtained in this research. In Eq. (13) $\sum_{k=1}^{K} n_k||C_k - C||^2$ was the inter-cluster dispersion. The inter-cluster dispersion is also called the in-between group sum of squares, with n_k being the observation count in cluster k, C_k being the centroid of cluster k, C being the centroid of the whole data set, and K were the number of clusters. $\sum_{i=1}^{n_k}||X_{ik} - C_k||^2$ was the intra-cluster dispersion, which is also known as the within-group sum of squares, with X_{ik} being observation i in cluster k. Lastly, the sum of all individual within-group sums of squares is calculated with $\sum_{k=1}^{K}\sum_{i=1}^{n_k}||X_{ik} - C_k||^2$.

$$CH_{index} = \frac{\sum_{k=1}^{K} n_k||C_k - C||^2}{\sum_{k=1}^{K}\sum_{i=1}^{n_k}||X_{ik} - C_k||^2}\frac{N-K}{K-1} \tag{13}$$

Silhouette Index. The Silhouette Index has a range from -1 to 1. A high value of the Silhouette Index shows that a specific object matches well with a cluster of its own. In this work the Silhouette Index was obtained by using the silhouette() function of the cluster package in R [43].

Eq. (14) shows how the Silhouette Index of object i was obtained. d_{near_i} was the average distance from i to all objects in the nearest cluster. d_{within_i} was the average distance from i to all objects within the same cluster as object i.

$$S_{Index} = \frac{d_{near_i} - d_{within_i}}{max(d_{near_i}, d_{within_i})} \tag{14}$$

Davies-Bouldin Index. The Davies-Bouldin Index is the average of the similarity measures of every cluster with greatest similarity. In this work the Davies-Bouldin Index was obtained by using the index.DB() function of the clusterSim package in R [43].

Eq. (15), Eq. (16), and Eq. (17) show how the Davies-Bouldin Index was obtained in this research. N were the total number of clusters. R_{ij} is a measure that shows how accurate the clustering is, with S_i and S_j being the within-cluster scatters for cluster i and j respectively, and M_{ij} was the separation between the clusters i and j.

$$DB_{Index} = \frac{1}{N}\sum_{i=1}^{N} D_i \tag{15}$$

with,

$$D_i = \max_{j \neq i} R_{ij} \tag{16}$$

$$R_{ij} = \frac{S_i + S_j}{M_{ij}} \tag{17}$$

4.5 Extra Extension

If time allows I would like to conduct a similar four-step approach research, described above, as an extra extension. However, this research will start with the data transformation step. This step also contains the two non-linear dimension reduction techniques Isomap and Umap. The next step is the cluster optimization step, then followed by the clustering step done by k-means and AGNES. Ending with the internal cluster validation step. Fig. 3 shows a visualization of the steps conducted in the extension of this research.

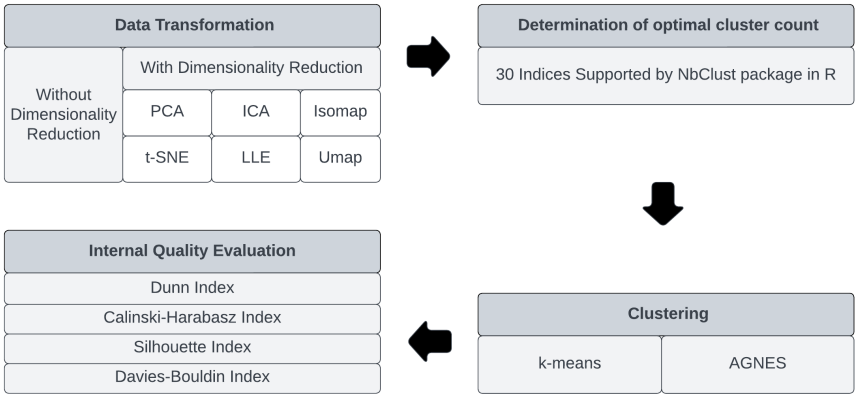


Fig. 2: Four-step approach used in the extension of this research

5 Results

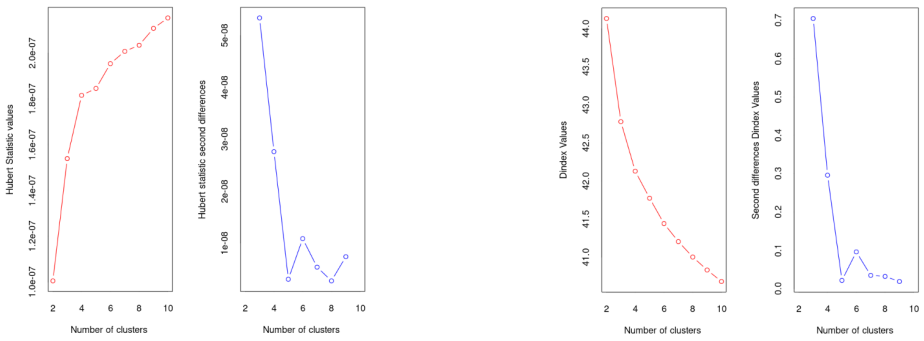
This section shows the results obtained from doing the research described in Sect. 4. Sect. 5.1 elaborates on the optimal cluster count obtained from the full Jester

rating data set and from a random sample consisting of 5,000 samples from the Jester rating data set. Sect. 5.2 show the results of the internal cluster validation from the different data transformation techniques, with optimal cluster count $k = 2$ and $k = 3$ respectively. Sect. 5.3 shows the internal validation results of the extension from this work. Sect. 5.4 elaborates more on the intriguing findings of this research. The second, third, and fourth columns of Table 3 up until and including Table 8 show the values for the Dunn Index, Calinski-Harabasz Index, and Silhouette Index respectively. Table 8 also includes the values for the Davies-Bouldin Index.

5.1 Optimal Cluster Count

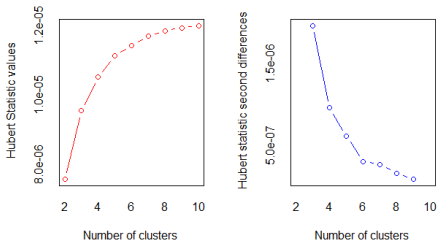
Fig. 3 shows the graphical representation of the optimal cluster count on the full Jester joke rating data set. Fig. 3 shows the optimal cluster count using the Hubert index and the Dunn index. From Fig. 3 can be seen that the optimal cluster count for the full data set is $k = 2$. The optimal cluster count is also used in the results of Sect. 5.2.

Fig. 4 shows the graphical representation of the optimal cluster count on the random sub-sample from the Jester joke rating data set, consisting of 5,000 samples. Fig. 4 also shows the optimal cluster count using the Hubert index and the Dunn Index. From Fig. 4, can be seen that the optimal cluster count for the sub-sample is $k = 3$.

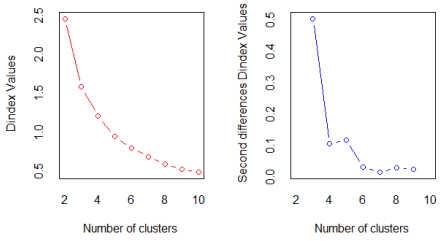


(a) (b)

Fig. 3: (a) Optimal cluster count found through the Hubert index method from the full data set (b) Optimal cluster count found through the Dunn Index method



(a)



(b)

Fig. 4: (a) Optimal cluster count found through the Hubert index method from the 5,000 samples (b) Optimal cluster count found through the Dunn Index method from the 5,000 samples

5.2 Results of Dimension Reduction Techniques

Table 3 and Table 4 show the results obtained from the internal cluster validation of this work for k-means and AGNES clustering, for $k = 2$ and $k = 3$ respectively. Fig. 5 and Fig. 6 show a visualization of the obtained clusters from the dimensionality reduction techniques.

From Table 3 can be seen that both PCA and t-SNE are performing best among all the dimension reduction techniques for k-means clustering, for the case $k = 2$. Do note that t-SNE might have a slightly better performance than PCA, as t-SNE has the highest index values for both the Calinski-Harabasz Index and the Silhouette Index. From Table 3 it is also apparent that both ICA and t-SNE are the best-performing data transformation techniques for AGNES clustering, for the case $k = 2$. Note that in the case of AGNES clustering, ICA is performing slightly better than t-SNE as ICA has the highest index values for both the Dunn Index and the Silhouette Index.

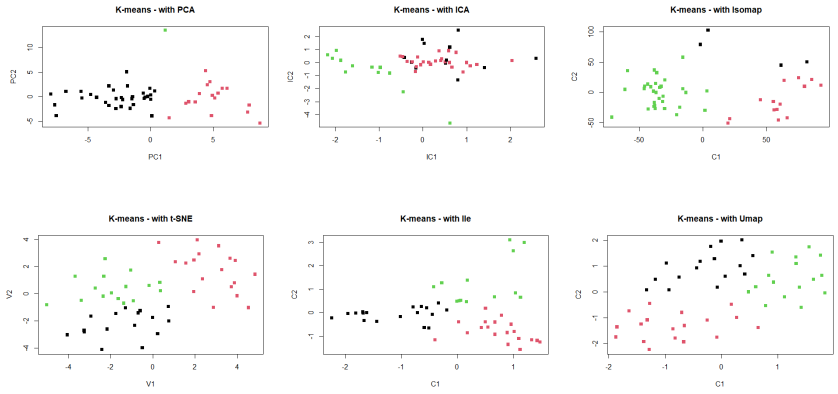


Fig. 5: Visualisation of dimension reduction techniques with k-means clustering

Table 3: Results of the internal cluster for $k = 2$

Technique	**D-Index**	**CH-Index**	**S-Index**
PCA with k-means	**0.2935**	19.9400	0.1518
ICA with k-means	0.0975	18.5867	0.2860
Isomap with k-means	0.2848	30.9340	0.2911
t-SNE with k-means	0.1030	**72.7000**	**0.4908**
LLE with k-means	0.1256	22.7461	0.3709
Umap with k-means	0.1816	69.1797	0.4703
PCA with AGNES	0.5666	4.8139	0.2837
ICA with AGNES	**0.9923**	14.1777	**0.5709**
Isomap with AGNES	0.5226	3.1335	0.3071
t-SNE with AGNES	0.1941	**66.5873**	0.4710
LLE with AGNES	0.6818	21.8517	0.5506
Umap with AGNES	0.1809	59.9661	0.4543

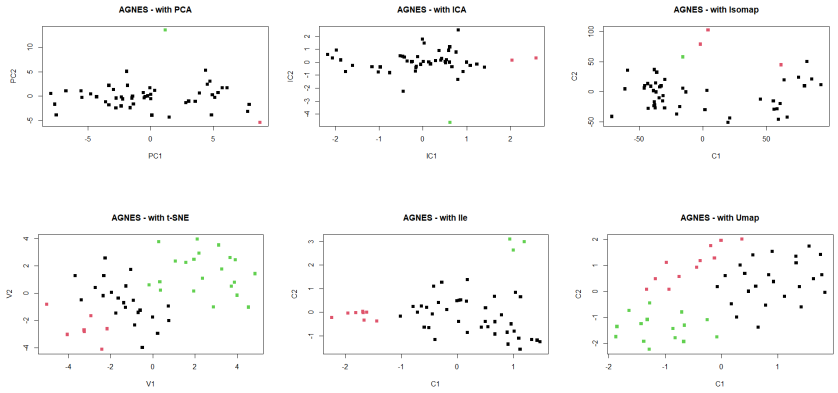


Fig. 6: Visualisation of dimension reduction techniques with AGNES clustering

From Table 4 it is apparent that PCA, t-SNE, and LLE are the three best-performing dimension reduction techniques for k-means clustering with $k = 3$. However, the found results in Table 4 can not be used in order to conclude which dimension reduction method performs best for the data transformation step. Therefore, in order to conclude which data transformation technique further research will be needed. From Table 4 can be seen that PCA, t-SNE, and LLE are the three best-performing dimension reduction techniques for AGNES clustering with $k = 3$ as well. It is intriguing to see that PCA, t-SNE, and LLE are the best-performing methods for both k-means and AGNES clustering, as seen in Table 4. This could be an indication that there is not a "best-performing" dimension reduction technique for the case $k = 3$. Instead, there could be multiple well-performing methods. However, in order to conclude this further research will be needed.

Table 4: Results of the internal cluster validation for $k = 3$

Technique	**D-Index**	**CH-Index**	**S-Index**
PCA with k-means	**0.3432**	13.0197	0.1525
ICA with k-means	0.1136	16.9591	0.3270
Isomap with k-means	0.2069	20.7316	0.2794
t-SNE with k-means	0.0615	**49.4460**	0.3771
LLE with k-means	0.1034	26.0257	**0.4688**
Umap with k-means	0.1570	48.0723	0.4102
PCA with AGNES	**0.5378**	4.6632	0.2295
ICA with AGNES	0.2284	9.3764	0.2520
Isomap with AGNES	0.3799	4.1743	0.2310
t-SNE with AGNES	0.1787	**49.1483**	0.3470
LLE with AGNES	0.2323	31.3228	**0.4290**
Umap with AGNES	0.1807	44.0077	0.3906

5.3 Results of extension

Table 5 show the results obtained from the internal cluster validation for the extension of this work using the 5,000 samples, for k-means and AGNES clustering respectively.

From Table 5 it is apparent that for k-means clustering PCA, Isomap, and t-SNE are all well-performing data transformation techniques for their respective optimal cluster counts, which are 8, 2, and 2 clusters respectively. A best-performing method can not be concluded from Table 5, as not every dimension reduction technique has obtained the same optimal cluster count. Table 5 also shows that PCA and ICA are both well-performing dimension reduction techniques for AGNES clustering. Looking only at the techniques with optimal cluster count 2 in Table 8, it can be concluded that ICA performs best in this specific case.

Table 5 indicates that the optimal cluster count $k = 2$ appears most frequently. This could mean that the optimal cluster count of $k = 2$, found in Sect. 5.1, is a better optimal cluster count than $k = 3$. The reasoning could be because

the optimal cluster count is obtained from the full Jester joke rating data set instead of a smaller subset, consisting of 5,000 samples.

Table 5: Results of the internal cluster validation of the extension

Technique	**D-Index**	**CH-Index**	**S-Index**	**Optimal Cluster**
PCA with k-means	0.0152	**26008.7601**	**0.5577**	8
ICA with k-means	0.0974	18.5867	0.2860	**2**
Isomap with k-means	**0.2848**	30.9340	0.2911	**2**
t-SNE with k-means	0.1030	72.6999	0.4908	**2**
LLE with k-means	0.1437	47.9631	0.4899	4
Umap with k-means	0.1816	69.1797	0.4703	**2**
PCA with AGNES	0.0151	**23252.3982**	0.5263	8
ICA with AGNES	**0.9923**	14.1777	**0.5709**	**2**
Isomap with AGNES	0.5226	3.1335	0.3071	**2**
t-SNE with AGNES	0.1941	66.5873	0.4710	**2**
LLE with AGNES	0.2487	52.0143	0.4622	4
Umap with AGNES	0.1809	59.9661	0.4543	**2**

5.4 Intriguing Findings

An intriguing finding is that the dimension-reduction technique t-SNE is the best-performing technique for both $k = 2$ and $k = 3$. This could be an indication that t-SNE does in fact perform "slightly" better than the other "well-performing" techniques found in Sect. 5.2 and Sect. 5.3. But looking at the different perplexities for t-SNE indicates that further research is needed in order to conclude this with certainty. Because for this work, a perplexity of 10 is used for the t-SNE technique. Changing the perplexity value for t-SNE results in different internal validation indexes, as seen in Table 6 up until Table 8. For example, from Table 6 and Table 8 can be seen that a perplexity of 10 gives the highest index results for $k = 2$, while for $k = 3$ both perplexities 6 and 10 show high internal validation index values. These changing results for the t-SNE technique show that more research is needed on whether the t-SNE is the "best-performing" technique.

Table 6: Comparison of internal validation of t-SNE with different perplexities for $k = 2$

Perplexity	**D-Index**	**CH-Index**	**S-Index**
4 with k-means	0.0619	50.2538	0.3989
6 with k-means	0.1026	55.8035	0.3942
8 with k-means	0.0870	42.6315	0.3830
10 with k-means	**0.1030**	**72.7000**	**0.4908**
4 with AGNES	0.2260	50.7682	0.3705
6 with AGNES	**0.2621**	57.6237	0.3563
8 with AGNES	0.1967	50.5354	0.3551
10 with AGNES	0.1941	**66.5873**	**0.4710**

Table 7: Comparison of internal validation of t-SNE with different perplexities for $k = 3$

Perplexity	**D-Index**	**CH-Index**	**S-Index**
4 with k-means	0.0619	50.2538	0.3989
6 with k-means	0.1026	**55.8035**	0.3942
8 with k-means	0.0870	42.6315	0.3830
10 with k-means	**0.1034**	26.0257	**0.4688**
4 with AGNES	0.2260	50.7682	**0.3705**
6 with AGNES	**0.2621**	**57.6237**	0.3562
8 with AGNES	0.1967	50.5353	0.3551
10 with AGNES	0.1787	49.1483	0.3470

Table 8: Comparison of internal validation of k-means with different perplexities for the t-SNE extension

Perplexity	**D-Index**	**CH-Index**	**S-Index**	**DB-Index**
4 with k-means	0.1265	62.2177	0.4671	0
6 with k-means	0.1969	72.4891	0.4671	0
8 with k-means	**0.2528**	60.4512	0.4610	0
10 with k-means	0.1030	**72.6999**	**0.4908**	0
4 with AGNES	0.2260	50.7682	0.3705	**0.2956**
6 with AGNES	**0.2621**	57.6237	0.3563	0.2804
8 with AGNES	0.1967	50.5354	0.3551	0.2517
10 with AGNES	0.1941	**66.5873**	**0.4710**	0

From Table 3 up until Table 7 can be seen that there are no reported Davies-Bouldin Index values, while Table 8 has them for AGNES clustering with t-SNE. The reasoning for this result could be that the davies-bouldin does not incorporate the euclidian distance correctly for the other techniques, even though the Euclidian distance metric has been explicitly used in the agnes() function in R. Therefore, future research on this specific aspect has to be done.

It is also noteworthy that the recently found dimension reduction technique Umap never obtained the highest index value for internal validation among all the given techniques. In order to dive deeper into this finding, further research is needed. Future work could also check whether Umap needs more fine-tuning for this specific data set.

6 Conclusion

This research tried to validate the quality of the k-means and the AGNES clustering techniques, by means of six dimension reduction techniques. The used linear dimension reduction techniques were PCA and ICA, whereas the used non-linear techniques were Isomap, t-SNE, LLE, and Umap. These dimension-reduction techniques were combined with the clustering techniques in order to

answer the question: **which method is the best dimension reduction technique?**

From the results in Sect. 5 it is apparent that PCA, ICA, t-SNE, and LLE performed best among the given data transformation techniques. However, from the results, it could be seen that there was no explicit "best-performing" dimension reduction technique. There were however some implications that t-SNE performs best among the found "well-performing" techniques. But further research is needed in order to validate this finding. Therefore, other researchers that want to conduct similar research have to take into account some of these findings, in order to find even better results.

Sect. 5 also showed some intriguing findings on the different perplexity values for the t-SNE method. The main finding was that the change in perplexity for t-SNE does influence the obtained results of the internal validation indexes, meaning that more research is needed in order to find the best perplexity value for this specific data set.

For future research, different extensions of the use of dimension reduction techniques or clustering analysis techniques can be used, such as the Linear Discriminant Analysis guided by Unsupervised Ensemble Learning (LDA-UEL) from [36]. LDA-UEL can be used to lower the dimensionality of more complex and higher-dimensional data sets. LDA-UEL is also very robust against outliers, which can be problematic for PCA and ICA according to [36]. As outliers could influence the principal components and independent components for PCA and ICA, LDA-UEL is a very intriguing technique to use in future research. Another possible extension for future research is to interpret the obtained clusters for k-means and AGNES. This can be done by researching the jokes that are put into certain clusters and then interpreting these findings. The fully written jokes can be found on `https://eigentaste.berkeley.edu/dataset/`. Lastly, the same four-step approach is repeated on a very different data set to validate the found results, as there are different data sets available at the Jester site.

References

1. Abdi, H., Williams, L.J.: Principal component analysis. Wiley Interdiscip Review Computer Statistics **2**, 433–459 (2010)
2. Böhm, J.N., Berens, P., Kobak, D.: Attraction-repulsion spectrum in neighbor embeddings. Journal of Machine Learning Research **23**(95), 1–32 (2022)
3. Caliński, T., Harabasz, J.: A dendrite method for cluster analysis. Communications in Statistics **3**, 1–27 (1974)
4. Charrad, M., Ghazzali, N., Boiteau, V., Niknafs, A.: Nbclust: An r package for determining the relevant number of clusters in a data set. Journal of Statistical Software **61**, 1–36 (2014)
5. Comon, P.: Independent component analysis, a new concept? Signal Processing **36**, 286–314 (1994)
6. Davies, D.L., Bouldin, D.W.: A cluster separation measure. Transactions on Pattern Analysis and Machine Intelligence **1**, 224–227 (1979)
7. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik **1**, 269—-271 (1959)

8. Dunn, J.C.: A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. Journal of Cybernetics **3**, 32–57 (1973)
9. Floyd, R.W.: Algorithm 97: Shortest path". communications of the acm. Communications of the Association for Computing Machinery **5**, 345 (1962)
10. Goldberg, K.Y., Roeder, T., Gupta, D., Perkins, C.: Eigentaste: A constant time collaborative filtering algorithm. Information Retrieval **4**, 133–151 (2001)
11. Granato, D., Santos, J.S., Escher, G.B., Ferreira, B.L., Maggio, R.M.: Use of principal component analysis (pca) and hierarchical cluster analysis (hca) for multivariate association between bioactive compounds and functional properties in foods: A critical perspective. Trends in Food Science Technology **72**, 83–90 (2018)
12. Green, P.E.: Marketing applications of mds: Assessment and outlook. Journal of Marketing **39**, 24–31 (1975)
13. Herault, J., Ans, B.: Réseau de neurones ́synapses modifiables : Décodage de messages sensoriels composites par apprentissage non supervisé et permanent. Comptes Rendus de l'Académie des Sciences **3**, 525–528 (1984)
14. Herault, J., Ans, B., Jutten, C.: Architectures neuromimétiques adaptatives : Détection de primitives. Cognitiva **2**, 593–597 (1985)
15. Herault, J., Ans, B., Jutten, C.: Détection de grandeurs primitives dans un message composite par une architecture de calcul neuromimétique en apprentissage non supervisé. Proceedings of the 10th Workshop Traitement du signal et ses applications **2**, 1017–1022 (1985)
16. Herault, J., Jutten, C.: Space or time adaptive signal processing by neural networks models. International Conference on Neural Networks for Computing **1**, 206–211 (1986)
17. Hinton, G., Roweis, S.: Stochastic neighbor embedding. Neural Information Processing Systems. **0**, 1–8 (2002)
18. Hotelling, H.: Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology **24**, 417–441 (1933)
19. Kohavi, R., Brodley, C.E., Frasca, B., Mason, L., Zheng, Z.: Kdd-cup 2000 organizers report: Peeling the onion. SIGKDD Explorer **2**, 86–89 (2000)
20. Kohavi, R., Provost, F.: Applications of data mining to electronic commerce. Applications of Data Mining to Electronic Commerce **5**, 5–10 (2001)
21. Kohavi, R., Rothleder, N.J., Simoudis, E.: Emerging trends in business analytics. Communications of the ACM **45**, 45–48 (2002)
22. Lloyd, S.: Least square quantization in pcm. IEEE Transactions on Information Theory **2**, 129–137 (1957)
23. Lukasová, A.: Hierarchical agglomerative clustering procedure. Pattern Recognition **11**, 365–381 (1979)
24. Lukasová, A.: Hierarchical agglomerative clustering procedure. Pattern Recognition **11**, 365–381 (1979)
25. Ma, J., Yuan, Y.: Dimension reduction of image deep feature using pca. Journal of Visual Communication and Image Representation **63**, 1047–3203 (2019)
26. van der Maaten, L.: Accelerating t-sne using tree-based algorithms. Journal of Machine Learning Research **15**, 3221–3245 (2014)
27. van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research **9**, 2579–2605 (2008)
28. van der Maaten, L., Postma, E., van den Herik, J.: Dimensionality reduction: A comparative review. Tilburg centre for Creative Computing **1**, 1–36 (2009)
29. MacQueen, J.: Some methods for classification and analysis of multivariate observations. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability **1**, 281–297 (1967)

30. Massy, W.F.: Principal components regression in exploratory statistical research. Journal of the American Statistical Association **60**, 234–256 (1965)
31. McInnes, L., Healy, J., Melville, J.: Umap: Uniform manifold approximation and projection for dimension reduction. Tutte Institute for Mathematics and Computing **0**, 1–63 (2018)
32. Naik, G.R., Kumar, D.K.: An overview of independent component analysis and its applications. School of Electrical and Computer Engineering **1**, 1–20 (2009)
33. Pearson, K.: On lines and planes of closest fit to systems of points in space. Philosophical Magazine and Journal of Science **2**, 559–572 (1901)
34. Racine, J.S.: Rstudio: A platform-independent ide for r and sweave. Department of Economics at McMaster University **2**, 1–6 (2012)
35. Renjith, S., Sreekumar, A., Jathavedan, M.: A comparative analysis of clustering quality based on internal validation indices for dimensionally reduced social media data. Advances in Artificial Intelligence and Data Engineering **1133**, 1047–1065 (2019)
36. de Ridder, D., Kouropteva, O., Okun, O., Pietikäinen, M., Duin, R.P.W.: Supervised locally linear embedding. Artificial Neural Networks and Neural Information Processing **2714**, 333—-341 (2003)
37. Rousseeuw, P.J.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics **20**, 53–65 (1987)
38. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**, 2323–2326 (2000)
39. Shlens, J.: A tutorial on independent component analysis. Computing of Research Repository **1404.2986**, 1–13 (2014)
40. Silva, V., Tenenbaum, J.: Global versus local methods in nonlinear dimensionality reduction. Advances in Neural Information Processing Systems **15**, 721–728 (2002)
41. Steinhaus, H.: Sur la division des corps matériels en parties. Bulletin of the Polish Academy of Sciences **4**, 801–804 (1957)
42. Takuya, I., Taro, T.: A local learning rule for independent component analysis. Scientific Reports **6**, 2045–2322 (2016)
43. Team, R.: R: a language and environment for statistical computing r. Foundation for Statistical Computing (2009)
44. Tierney, L.: The r statistical computing environment. Statistical Challenges in Modern Astronomy V **902**, 435–447 (2012)
45. Wang, C., Song, X.: Robust frontal view search using multi-camera constrained isomap **0**(0), 1017–1020 (2012)

A Appendix - Programming code

Hardware information: For this research, the open-source programming language R [43] and the development environment Rstudio [44,34] are used. The whole research has not yet been conducted therefore it could be that other programming languages were used as well. The Packages used for this research can be found in the methodology of the final Thesis paper. The used hardware in this work is Intel Core i7-8550U, 2.40 GHz, and 5.00 GHz dual-core x64-based processor with 8.00 GB RAM.

Prepare hardware:

Step 1: Download R version 4.3.0, from https://cran.rstudio.com/
Step 2: Dowload the IDE compatible with R version 4.3.0 called Rstudio, from https://posit.co/download/rstudio-desktop/
Step 3: Download Rtools43, from https://cran.rstudio.com/bin/windows/Rtools/rtools43/rtools.html
Step 4: Download code from Github, from `https://github.com/XDilalaX/Thesis_code/tree/main`
Step 5: Run the code!

Program inoformation: The whole program used for this research consists of 13 R program files, each fulfilling their own task, as explained below:

1. **Thesis_code_pca.R** This program shows step 2 up until 4 for the research of the thesis for k-means and AGNES clustering with PCA as its dimension reduction technique.
2. **Thesis_code_ica.R** This program shows step 2 up until 4 for the research of the thesis for k-means and AGNES clustering with ICA as its dimension reduction technique.
3. **Thesis_code_isomap.R** This program shows step 2 up until 4 for the research of the thesis for k-means and AGNES clustering with Isomap as its dimension reduction technique.
4. **Thesis_code_tsne.R** This program shows step 2 up until 4 for the research of the thesis for k-means and AGNES clustering with t-SNE as its dimension reduction technique.
5. **Thesis_code_lle.R** This program shows step 2 up until 4 for the research of the thesis for k-means and AGNES clustering with LLE as its dimension reduction technique.
6. **Thesis_code_isomap.R** This program shows step 2 up until 4 for the research of the thesis for k-means and AGNES clustering with Isomap as its dimension reduction technique.
7. **Thesis_code_pca_extension.R** This program shows the code for the extension part of the thesis. This code consists of step 0 up to 4 for the research of the thesis for k-means and AGNES clustering with PCA as its dimension reduction technique.
8. **Thesis_code_ica_extension.R** This program shows the code for the extension part of the thesis. This code consists of step 0 up to 4 for the research of the thesis for k-means and AGNES clustering with ICA as its dimension reduction technique.
9. **Thesis_code_umap_extension.R** This program shows the code for the extension part of the thesis. This code consists of step 0 up to 4 for the research of the thesis for k-means and AGNES clustering with Umap as its dimension reduction technique.
10. **Thesis_code_tsne_extension.R** This program shows the code for the extension part of the thesis. This code consists of step 0 up to 4 for the research of the thesis for k-means and AGNES clustering with t-SNE as its dimension reduction technique.
11. **Thesis_code_lle_extension.R** This program shows the code for the extension part of the thesis. This code consists of step 0 up to 4 for the research of the thesis for k-means and AGNES clustering with LLE as its dimension reduction technique.

12. **Thesis_code_isomap_extension.R** This program shows the code for the extension part of the thesis. This code consists of step 0 up to 4 for the research of the thesis for k-means and AGNES clustering with Isomap as its dimension reduction technique.
13. **Thesis_code_optimalclusters.R** This is the code used in order to obtain the optimal cluster count for this research, also known as step 1.

B Appendix - Visualization of the dimension reduction techniques for $k = 2$

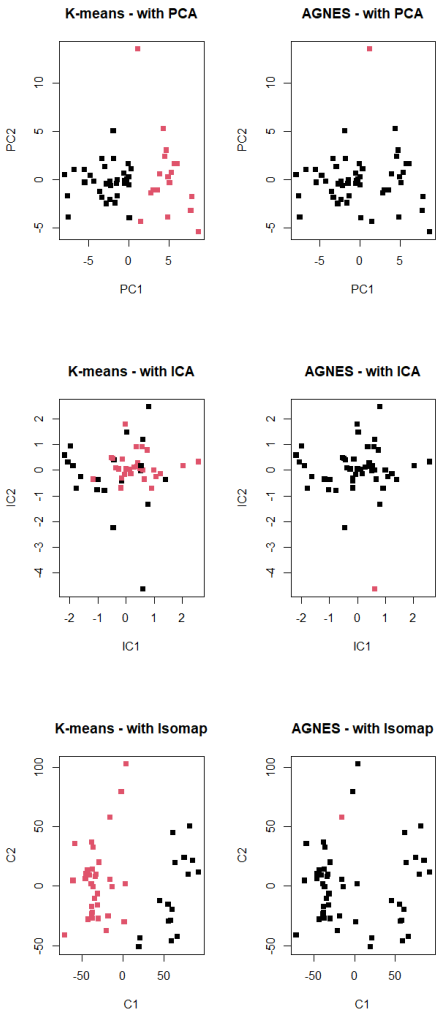


Fig. 7: (a) Visualization for PCA (b) Visualization for ICA (c) Visualization for Isomap

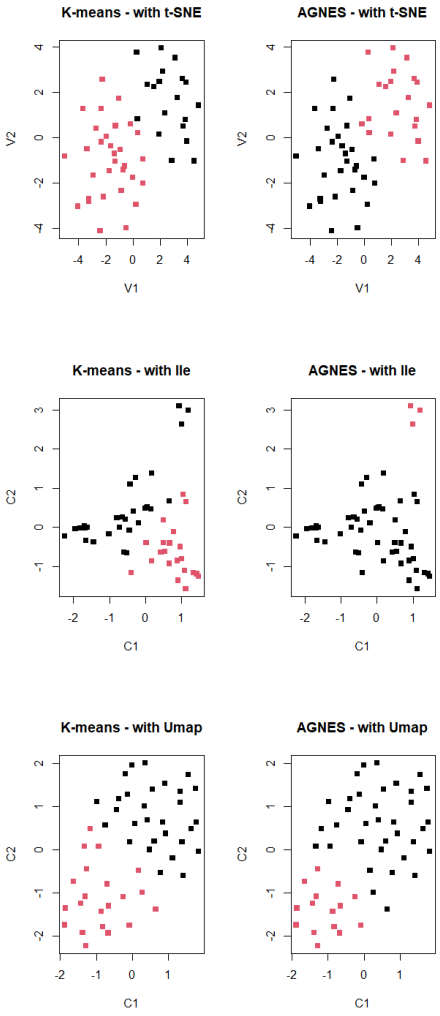


Fig. 8: (d) Visualization for t-SNE (e) Visualization for LLE (f) Visualization for Umap