ERASMUS UNIVERSITY ROTTERDAM

Erasmus School of Economics

Clustering high-dimensional rating data by using dimension reduction

Econometrics and Management Science

Quantitative Marketing and Business Analytics

Master Thesis

Author: Sietse Bennema
Student ID: 432425

Supervisor: dr. M. Van de Velden
Second Reader: dr. C. Cavicchia

August 16, 2021

The content of this thesis is the sole responsibility of the author and does not reflect the view of the supervisor, second assessor, Erasmus School of Economics or Erasmus University.

Abstract

In this research, we investigate the effect of using different dimension reduction techniques to efficiently cluster high-dimensional data. The idea behind using dimension reduction is to lift the *curse of dimensionality*, which is a term to describe that full dimensional clustering algorithms get less accurate when the amount of dimensions increases. Two approaches can be separated when using dimension reduction for clustering: The "tandem" approach, where we first apply dimension reduction and then cluster, or the simultaneous approach, where dimension reduction and clustering are applied at the same moment. Several tandem and simultaneous methods are practiced in a simulation study and we find that one simultaneous method using PCA, Reduced K-means, should be favoured when there are masking variables present, which are variables that do not contain taxonomic information.

Contents

1 Introduction

With the increased popularity of social media, streaming platforms and E-commerce, the collection and exploitation of data is gaining huge interest in business and science. One type of data that is often acquired is rating data. Rating data can be used to measure the customer satisfaction of certain products or services. Possible applications are the amount of stars for a movie you have seen on *Netflix* or rating your ordered package at a webshop with a number from 1 to 10 of how satisfied you are. Given that an amount of people have left a rating for a list of items, it would be interesting to segment this people into groups with similar ratings. Having identified the different groups of customers, you could treat them according to their preference, like give them more personalised advertisements.

The task of grouping observations based on their properties such that the identified groups are more homogeneous than the other groups, can also be called cluster analysis. This grouping is most effective when the amount of variables (amount of objects to be rated) for each observation is small, but tends to be less accurate when the amount of variables get large (Hinneburg and Keim, 1999). This problem is also called *curse of dimensionality*, which is a term introduced by Bellman (1961), and can be explained by the fact that the *volume* grows exponentially when adding dimensions. The volume in this case means all the possible amount of samples that can be generated in a certain amount of dimensions. As the volume increases, the possible amount of values to be generated increases. Adding a dimension will therefore increase the distances between points, even if they belong to the same cluster. The increase of distance between observations who actually belong to the same cluster may lead to the situation that these points are not being allocated to the same cluster, which is an undesirable situation.

The curse of dimensionality can also be explained by other factors. When the amount of variables is large, it is likely that some variables can be disregarded as these variables do not contain relevant information for clustering. These are so called "masking variables" (Vichi and Kiers, 2001). However, these variables are taken into account by the clustering algorithm and therefore lead to low clustering accuracy.

One solution to efficiently cluster high-dimensional data, is to reduce the amount of dimensions, while retaining as much information as possible, and then combine it with clustering. If the dimensions are reduced successfully, the distance between observations from the same cluster are similar, while observations from different clusters have different characteristics.

Dimension reduction can be done in 2 ways. First of all, we could select only a subset of features which carry the most information. However, the information carried by the disregarded features is evidently lost. Another way to decrease the amount of dimensions is feature projection. Feature projection reduces the amount of dimensions using linear combinations of the data. Different dimension reduction techniques are applied in different ways and thereafter the cluster allocations are compared. These cluster allocations are evaluated both on accuracy and cluster "quality", which measures the separation between the clusters and the cohesiveness within the clusters. Our goal of this research is to analyse the added value of different dimension reduction techniques for clustering high-dimensional rating data and which application of dimension reduction performs the best. Our main research question therefore is:

How do the different combined dimension reduction and clustering methods and the full dimensional clustering method compare with respect to clustering accuracy and quality when applied to high-dimensional rating data?

The first dimension reduction technique we consider is principal component analysis (PCA, Hotelling (1933)). PCA finds linear combinations of the variables, called loadings, that are orthogonal to each other. Multiplying these loadings with the data leads to the principal components, which are the projections of the data and are obtained by maximizing the variance of these components.

Next, we consider correspondence analysis (CA, Benzécri (1973)). This technique is developed to analyse contingency tables, which are cross tables that shows the frequency distribution of two categorical variables. Applying CA results in coordinates of rows and columns of the contingency table which can be displayed in a biplot and therefore explain the relationship between the two variables. Next to contingency tables, CA can also be used to analyse other matrices, given that the row and column totals are positive.

Greenacre (1984) concluded that applying CA to rating data does not satisfy the "scale invariance" condition. This condition means that if the scales of the rating data are reversed, analysis of such data should lead to the same outcome. To solve this problem, he proposed to double the columns of the matrix in such a way that the added columns are reflections of the original dataset.

An extension of correspondence analysis is multiple correspondence analysis (MCA, Greenacre (1984)). Where correspondence analysis is focused on the relationship between two categorical variables, MCA can be used to analyse the relationship of more than two categorical variables. The trick used is to transform the categorical matrix to an indicator matrix where the samples are the rows and all the categories of the variables are in the columns. Applying CA to this matrix results in low-dimensional coordinates of both the rows (observations) and columns (categories).

Our goal of applying PCA, CA and MCA is to obtain a low-dimensional projection of the rows that accurately represent the data. Given these projections, the observations can be clustered using a clustering algorithm. In our research, we use the K-means algorithm using Euclidean distance (MacQueen et al., 1967) for clustering the data. Applying dimension reduction and clustering in sequential order, is called a tandem technique (Arabie and Hubert, 1994).

Tandem techniques are relatively fast as dimension reduction and clustering only have to be done once and independently, but it also has the downside that dimension reduction and clustering both have their own objectives: the goal of dimension reduction is to accurately represent high dimensional data in low-dimensional space, while the goal of clustering is to split these observations into separate groups based on their values. The consequence may be that the observations in the reduced space are obtained from a subset of variables of the full dimensional dataset and that variables that are essential for clustering are left out. More generally, taxonomic information on the observations may be lost in the dimension reduction step.

This problem of the tandem approach has been addressed by Van de Velden et al. (2016) and Vichi and Kiers (2001). Because of this downside, they both propose a simultaneous approach, which incorporates both steps of the tandem approach into one function. In other words, this approach combines the cluster allocation with the data projection of dimension reduction technique in one function. This way, the dimension reduction step is dependent on the allocation of the clustering, and therefore, Van de Velden et al. (2016) and Vichi and Kiers (2001) argue that the variables that differentiate the observations are more likely to be taken into account for clustering.

Vichi and Kiers (2001) illustrate that their simultaneous method allocates the observations correctly to the clusters in an example where only a part of the continuous variables are important in allocating the clusters. However, in the simulation study of Van de Velden et al. (2016) concerning categorical data, the tandem technique and simultaneous technique both performed similar.

To find out how simultaneous methods compare to tandem approaches, we consider various simultaneous techniques which use PCA, CA and MCA as dimension reduction techniques, which are applied to rating data. These are Reduced K-means (De Soete and Carroll, 1994) and Factorial K-means (Vichi and Kiers, 2001), which both use PCA, MCA K-means (Hwang et al., 2006) and a simultaneous approach which uses correspondence analysis and clustering, similar bot not equal to Cluster Correspondence Analysis (CCA) in Van de Velden et al. (2016).

All the clustering allocations are evaluated based on accuracy, with a correction for the probability of coincidence that the allocation is correct. Furthermore, the quality of the clusters will be assessed by how cohesive the observations are within each cluster and how well the clusters are separated. All the combined clustering approaches are compared to full dimensional clustering, meaning that we apply the clustering algorithm (e.g. K-means) to all the variables of the data.

The data used for this study is simulated. The datasets vary in the amount of variables, the rating-scale, the amount of variance and whether there are masking variables present or not.

The results of the simulation study show that the tandem approach and full dimensional clustering may lead to high accuracy when there are no masking variables present, but tend to be less effective when there are masking variables present. In such a case, Reduced K-means tend to outperform all the other methods. Tandem MCA and MCA K-means generally outperform the other methods when there are no masking variables.

Factorial K-means and our version of CCA did lead to low clustering accuracy and therefore we do not recommend these methods for rating data, where we note that the low score for Factorial K-means could also be due to the low amount of "complement residuals" (Timmerman et al., 2010). Finally, we could not find that increasing the amount of dimensions leads to a decrease of the clustering accuracy when using full dimensional clustering. However, the curse of dimensionality can be explained by the presence of masking variables, in which case RKM is recommended.

The thesis is structured as follows: in Section 2, the present literature will be introduced. In Section 3, an overview of the used methods is given. Next, the data used for this research are discussed in Section 4. In Section 5, the results of the simulation study are presented. Subsequently, we apply Reduced K-means to a student survey and interpret the results. In Section 6, we conclude the thesis with our findings, limitations and recommendations for future research.

2 Literature

Using dimension reduction to efficiently cluster high-dimensional data has been studied extensively. Yeung and Ruzzo (2001) compared a tandem approach using PCA to full dimensional clustering and stated that PCA does not necessarily improve the clustering. Ben-Hur and Guyon (2003) note that this may be due to the standardisation of the variables beacause they find that standardisation may lead to a decrease in clustering quality. Moreover, Ben-Hur and Guyon (2003) note that increasing the principal components in the dimension reduction step does not necessarily lead to better clustering allocations.

Ciampi et al. (2005) used correspondence analysis in order to cluster both rows and columns. Moreover, they proposed an algorithm that can be interpreted as a formalisation of the "elbow rule": An algorithm that estimates the amount of dimensions that should be retained using dimension reduction.

Greenacre (1984) proposed to double the data column wise when applying correspondence analysis to rating data. This is necessary because rating data is *bipolar*, meaning that are two poles, in this case the highest and the lowest rating. Analysis of the rating data matrix where the poles are reversed, should lead to the same results, such that the data is scale invariant.

The content of the columns to be added should be the reverse of the original matrix, so when the original matrix contains the positive association of a sample with respect to a product, the appended matrix should contain the negative association for that sample. This datapoints are thus reflections and both have the same distance to the mean of the rating.

Arimond and Elfessi (2001) have applied MCA and clustering sequentially in analysing categorical survey data. Multiple correspondence analysis can be defined as an extension of correspondence analysis. In CA, the underlying structure of 2 variables are analysed, whereas in MCA, this can be done for more than 2 variables.

Vichi and Kiers (2001) and Desarbo et al. (1991) note that tandem techniques have the downfall that taxonomic information may be lost in the dimension reduction step and therefore favor a simultaneous approach. This approach incorporates both steps of the tandem approach into one function, so both dimension reduction and clustering are joined into one objective and are optimised simultaneously. Vichi and Kiers (2001) mark that their simultaneous method is better able to take advantage of the taxonomic information and therefore obtain better clusters.

For simultaneous clustering and dimension reduction of numerical data, De Soete and Carroll (1994) proposed reduced K-means and Vichi and Kiers (2001) proposed factorial K-means. Both methods simultaneously use principal component analysis with K-means clustering. The reduced K-means algorithm has the objective to minimize the squared distance between the data point and the centers of the clusters in the projected space, which are spanned by a loading matrix. The loading matrix linearly changes a high dimensional dataset to the reduced space or vice versa. The factorial K-means algorithm minimizes the sum of the squared distances between the centers of the clusters and the observations, both in the projected space.

For an appraisal of these techniques, we refer to the research by Timmerman et al. (2010). They show that it depends on the structure of the error which method leads to more accurate cluster allocations. In their research, they make a separation between "subspace" residuals and "complement" residuals. The subspace residuals are the errors in the reduced space, the complement residuals are the errors in the complement of this subspace. The presence of one of these errors determines which method should be preferred.

For the clustering and dimension reduction of categorical data, several methods have been proposed. Markos et al. (2018) refer to 3 joint methods, namely multiple correspondence analysis K-means (MCA K-means, Hwang et al. (2006)), iterative factorial clustering of binary variables (I-FCB, Iodice D'Enza and Palumbo (2013)) and cluster correspondence analysis (CCA, Van de Velden et al. (2016)), which are all included in their R package.

MCA K-means simultaneously applies MCA for dimension reduction with K-means for clustering. To optimise MCA K-means, the coordinates are obtained using MCA and are clustered using K-means and the other way around until convergence is reached.

Van de Velden et al. (2016) proposed a new method that is called cluster correspondence analysis. This method simultaneously applies K-means and correspondence analysis to categorical data. The correspondence analysis is applied to a multiplication of two matrices, which is $Z^T Z_K$. Z is the same for MCA K-means and Z_K is a binary matrix that assigns observations to clusters. However, we intend to use correspondence analysis differently than in CCA, namely that we double the matrix column wise. Moreover, the rating data will be interpreted as continuous and not as categorical when applying correspondence analysis. In other words, Z is a doubled rating data matrix in our application of CCA.

Furthermore, Van de Velden et al. (2016) evaluated the simultaneous methods, a tandem approach and the full dimensional clustering using a simulation study, where they used 2 different measures. First, as it's a simulation study, the accuracy of the allocation of the clusters can be checked. To measure this accuracy, they propose the Adjusted Rand Index by Hubert and Arabie (1985), which results in a score from -1 to 1, where the higher the score, the better the allocation.

It is likely that the "true" cluster allocation is not known. In such cases, the cluster allocation is evaluated differently. Therefore, they propose the average silhouette width Rousseeuw (1987). The silhouette width of a point is the the average distance of the points within the cluster, minus the average distance to the points of the nearest cluster, divided by the biggest of the two. A high score for the average silhouette width indicates how well the clusters are split among each other, while the observations are close to each other.

The results of the simulation study of Van de Velden et al. (2016) shows that when the amount of variables increase and when there are noise variables present, the full dimensional clustering technique tends to perform worse than the other clustering approaches. Among the simultaneous approaches and the tandem approach, none of them tend to perform significantly worse or better.

Next to combined clustering and dimension reduction, there are other methods to efficiently cluster high-dimensional data. Domeniconi et al. (2004) found that dimension reduction techniques are not effective in separating clusters that exist in different subspaces, or to put it in another way, are dependent of different variables. They therefore propose to select the relevant features locally for each cluster. This way, the variables that are relevant for a group of observations are more likely to be taken into account.

3 Methodology

In this section, the introduced methods will be further discussed. First, the dimension reduction techniques used for tandem analysis will be explained. Next, a description of K-means and full dimensional clustering will be given. Finally, a further elaboration of the simultaneous techniques and our evaluation methods will be presented.

3.1 Principal component analysis

Principal component analysis is a data reduction method introduced by Pearson (1901) and Hotelling (1933). This method is further developed numerous times, such as in Jolliffe and Cadima (2016). Moreover, Gower and Hand (1996) describe PCA in relationship with CA and MCA when creating a biplot. PCA finds linear combinations of the variables that are orthogonal to each other. These linear combinations are obtained by maximizing the variance of the vectors multiplied with data, which are called principal components.

These principal components can also be described more geometrically. Consider the data matrix P with I observations and J variables, such that the observations are represented in J dimensional space. P is therefore a multivariate dataset, and the rows and columns cannot easily be exchanged for this analysis. The distance between two row points is given in Euclidean distance, which is given in Equation 1, where $p_{i,j}$ is the value for P at row i for variable j.

$$d(i,k) = \sqrt{\sum_{j=1}^{J}(p_{i,j} - p_{k,j})^2} \tag{1}$$

Then, PCA finds a D dimensional subspace that orthogonal projects the I datapoints such that the Euclidean distance between the projections and the datapoints are minimized, where $1 \leq D \leq J$. The projections of the datapoints are the principal components.

The principal components can also be found by solving the following eigenequation. First, consider our data matrix to be P with I rows and J columns. Then we alter P to a column centred matrix P^*, which means that the average of the column is subtracted for each value in P. Therefore, $P^{*T}1_I = 0_J$, where 1_I a column vector of ones of dimension I and 0_J a column vector of zeros of dimension J. Geometrically, this centring could also be interpreted as setting the centroid at the origin.

We can obtain the eigenvalues and eigenvectors of P^* by computing the spectral decomposition of $P^{*T}P^*$, which is shown in Equation 2. More information on the spectral decomposition can be found in Appendix A.

$$P^{*T}P^*V = V\Lambda \tag{2}$$

In Equation 2, V contains the eigenvectors in the columns ($V = [v_1, v_2, ..., v_J]$) and Λ is the diagonal matrix of the eigenvalues of P. Given that V contains the orthonormal eigenvectors, $V^TV = I$. Also, the values in Λ are sorted in decreasing order, such that the largest squared singular value is λ_1^2, the squared singular in the first row, and the smallest value is λ_J^2, the squared singular value in the last row. The eigenvectors multiplied with the data (v_jP) are the so called principal components and the corresponding eigenvalue (λ_j^2) indicate the amount of variance explained by each component.

These vectors can also be obtained by the singular value decomposition as in Equation 3, where U and V are unitary matrices and Σ the matrix containing the singular values. More information on the singular value decomposition is given in Appendix B.

$$P^* = U\Sigma V^T \tag{3}$$

Given Equation 3, we can formulate Equation 4, which shows the relationship with the spectral decomposition by Equation 2:

$$P^{*T}P^* = V\Sigma U^T U\Sigma V^T = V\Sigma\Sigma V^T = V\Lambda V^T \tag{4}$$

where we use that U is a unitary matrix. Moreover, given that Σ is a diagonal matrix containing the singular values of P^*, $\Sigma\Sigma$ contains the squared singular values on the diagonal. Given that $\Lambda = \Sigma\Sigma$, the squared singular values of P^* are equal to the eigenvalues of $P^{*T}P^*$.

Given that Σ is a diagonal matrix, we can rewrite P^* as $\sum_{k=1}^{J} \lambda_k u_k v_k'$. We know that the vectors u_k and v_k are all standardised and the elements in Σ are decreasing, the first elements of the sum explain the most amount of the variance of P^*. Our goal is to find linear combinations that maximize the variance of the data, and therefore we can use the columns of V as these are the eigenvectors of $P^{*T}P^*$. The first D columns of V can be used to project the samples. Also, the eigenvectors are orthogonal to the other vectors, meaning that multiplying column vectors of V, such as v_l and v_d, are equal to 1 for $l = d$ and are equal to 0 for $l \neq d$. The projections can be denoted as $Q = PV_D \ = U_D\Sigma_D$, where Q denote the datapoints in D dimensions, where $D < J$. This means that the I observations are denoted in the columns in D dimensional space. The amount of variance explained by this projections is dependent of the proportion of the first D squared singular values, which can be denoted as

$$\frac{\lambda_1^2 + \lambda_2^2 + ... + \lambda_D^2}{\lambda_1^2 + \lambda_2^2 + ... + \lambda_D^2 + ... + \lambda_J^2} \tag{5}$$

where λ_l is equal to the diagonal element of row l of Σ, which holds for all l. If this fraction is small, the projection of the data is not a good representation of the original dataset.

The error term can be estimated by calculating the difference between the data and the spanned projections, which can be estimated as follows:

$$P - PV_DV_D^T = P(I - V_DV_D^T) \tag{6}$$

When the scales of the variables are very different, the data should be standardised. This can be done by dividing each column by it's standard deviation, such that P^TP is a correlation matrix, instead of a covariance matrix.

3.2 Correspondence analysis

The next method we propose is correspondence analysis (CA, Benzécri (1973)). This technique is developed to analyse contingency tables, such that the relations between two categorical variables can be further explained. CA can be used to obtain the coordinates of both rows and columns in the reduced space, and can thereafter be displayed in a biplot. Next to the analysis of contingency tables, CA can also be used to analyse other matrices, but it is required that the sum of all rows and columns are positive (Gower and Hand, 1996).

The distance between two row points in CA is given in χ^2 distance, which is given in Equation 7. In this equation, r_i is the row total of P for row i and c_j as the column total of P for column j. Next, the value at row i and column j for P is given by $p_{i,j}$.

$$d(i,k) = \sum_{j=1}^{J} \frac{1}{c_j}(\frac{p_{i,j}}{r_i} - \frac{p_{k,j}}{r_k})^2 \tag{7}$$

In CA, the distance between columns can be calculated similarly as for the rows in Equation 7. From the χ^2 distance of 2 rows can be obtained that scores with low row totals have high influence in comparison to the same scores with higher row totals.

In this research, we apply CA to rating data. In the first step of correspondence analysis, we transform the rating data matrix P with dimensions I and J and the total sum of n to $N = (1/n)P$. The sum of all elements in N is now equal to 1. Next, we define r and c as the vectors where the elements are equal to the sum of each row and column of N respectively. Also, we define D_r and D_c as diagonal matrices with r and c on the diagonal.

Next, we use the row and column totals to standardize and center the matrix, such that each element can be written as follows: $(n_{i,j} - r_i c_j)/\sqrt{r_i c_j} = (n_{i,j} - e_{i,j})/\sqrt{e_{i,j}}$. Note that $e_{i,j}$ can be interpreted as the expected value of $n_{i,j}$ given the row and column totals. Moreover, the singular value decomposition of the transformed matrix will be used, which is defined in Equation 8.

$$D_r^{-1/2}(N - rc^T)D_c^{-1/2} = \tilde{P} = U\Sigma V^T \tag{8}$$

In Equation 8, the singular values of diagonal matrix Σ are represented in descending order. Also, matrices U and V are unitary matrices, meaning that $U^TU = V^TV = I$. Furthermore, we want to approximate $\tilde{P}$ using two matrices that represent the reduced row and column values, which will be denoted as X and Y. Using Equation 8 and Van de Velden et al. (2016), X and Y are obtained by the following minimization function

$$\min_{X,Y} ||\tilde{P} - D_r^{1/2} X Y^T D_c^{1/2}||^2 \tag{9}$$

subject to $Y^T D_c Y = I$. This formulation leads to the same principal coordinates as in Greenacre (1984), but in that particular research a minimization form is not used. Given the SVD in Equation 8, $X = D_r^{-1/2} U \Sigma$ and $Y = D_c^{-1/2} V$. The results for X and Y are the principal row coordinates and standard column coordinates respectively. If we would impose that $X^T D_r X = I$ instead of $Y^T D_c Y = I$, the results for X and Y would be the standard row coordinates and principal column coordinates respectively. The coordinates in the reduced space are the first D columns of both X and Y. The representation where X are standard (principle) coordinates and Y are principle (standard) coordinates is called an asymmetric mapping.

A plot where the principal row coordinates and the principal column coordinates are shown are called a symmetric mapping. Since the research is limited to clustering the observations using dimension reduction, only the first D columns of the principal row coordinates will be used. The reason for choosing principal coordinates is the fact that the scaling of the singular values are an indication of the influence of a dimension. Singular values are not included in the display of standard coordinates and therefore they do not include that information.

The diagonal elements of Σ are the singular values. The accuracy of the low-dimensional representation of the data using CA can be analysed similarly as for PCA, namely by Equation 5, using the squared singular values.

As PCA and CA both can be obtained using the singular value decomposition, the differences of the projections of the rows can be explained further. First, consider that in PCA, P^* is column centred, and that for CA, the matrix to be decomposed is obtained by dividing by n and then by standardizing and centering for both rows and columns (Equation 8). Due to the fact that $n_{i,j}$ is measured in χ^2 distance (Equation 7), scores who are high with respect to the column and row average are very influential in CA. A correction for columns and rows with low column and row totals is not automatically included for PCA. Even so, in most applications the columns of the PCA are standardized, as a high variance in a certain column is very influential.

3.3 Multiple correspondence analysis

Multiple correspondence analysis is another dimension reduction method, which can be applied to categorical data. Recall that CA can be used to analyse the frequency distribution of two variables. In this case, MCA can be interpreted as CA where the amount of variables is 3 or larger. The trick behind MCA is to denote the categorical matrix as an indicator matrix and then analyse this matrix using CA. It can thus be considered as CA to an indicator matrix. However, Gower and Hand (1996) refer to it as PCA to categorical data.

3.3.1 Transformation to binary matrix

Consider matrix P with I observations and J variables, and let $p_{i,j}$ be the value for row i and column j of P. For variable j, there are e_j possible values, or in a rating data matrix, the scale of variable j is e_j. The indicator matrix Z now has I observations and $E = \sum_{j=1}^{J} e_j$ columns, such that each possible value for each variable has an own column. If $z_{i,1j} = 1$, then observation i corresponds to value 1 of variable j. The other values will be 0 for observation i at variable j. Multiple correspondence analysis can be interpreted as correspondence analysis of the indicator matrix. Therefore, we can again obtain coordinates for the rows and columns and analyse their similarity, for example in a biplot.

The alteration of the indicator matrix is further illustrated by an example. Let Table 1 be a survey where 5 people rate 3 products. Each product is rated from 1 to 5 from each respondent, where a score of 1 indicates that the respondent is very unsatisfied, and a score of 5 is equal to the highest score of satisfaction of the product. Each columns corresponds to a possible rating from a product, where the possible amount of ratings for a single column is $e_j = 5$, such that the total amount of columns can be calculated as $E = \sum_{j=1}^{J} e_j = \sum_{j=1}^{3} 5 = 15$. Next, we let a score of 1 correspond to the first column of a product, a score of 2 for the second, et cetera. Finally, we can obtain the indicator matrix Z, which is given by Table 2.

Table 1: example of numerical data

Respondent	**Product 1**	**Product 2**	**Product 3**
a	5	3	2
b	3	2	3
c	3	1	3
d	4	2	4
e	4	3	4

Table 2: Indicator matrix of categorical data

Respondent	Product 1					Product 2					Product 3				
a	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0
b	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0
c	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0
d	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0
e	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0

3.3.2 MCA explained as extension of CA

MCA can be explained simply by interpreting it as CA to the indicator matrix Z (Greenacre, 1984). First, we set the sum of all element equal to 1, such that $G = Z/n$, where n is the sum of all elements of Z. Next, the singular value decomposition of the standardized and centered matrix is computed, which van be denoted as

$$D_r^{-1/2}(G - rc^T)D_c^{-1/2} = U\Sigma V^T$$

where r and c are the row and column totals of G respectively and where D_r and D_c are diagonal matrices with r and c on the diagonal. Then, the coordinates of the rows in the projected space can be computed as

$$Q = D_r^{-1/2}U\Sigma$$

where the coordinates of the rows in the D dimensional space can be interpreted as the first D columns in Q.

3.3.3 Derivation of MCA using PCA

MCA can also be explained as PCA applied to categorical data. Consider D_c to be a diagonal matrix with the column frequencies of the indicator matrix Z at the diagonal. Then, MCA can be interpreted as PCA to $X = D_r^{-1/2}ZD_c^{-1/2}$, which can be interpreted as the standardised indicator matrix. Gower and Hand (1996) use $J^{-1/2}$ instead of $D_r^{-1/2}$, but as the row totals in D_r are equal to the amount of variables J, the results are the same.

The distance between samples can be calculated using Euclidean distance, given $X = D_r^{-1/2}ZD_c^{-1/2}$, is estimated as

$$d(i, j) = D_r^{-1}(z_i - z_j)D_c^{-1}(z_i - z_j)^T$$

where z_i is sample i of Z in row form.

According to Gower and Hand (1996), the projections of the sample coordinates in D dimensional space can also be obtained as the columns of

$$Q_D = XV_D = D_r^{-1/2} Z D_c^{-1/2} V_D = U_D \Sigma \tag{10}$$

where U, Σ and V are obtained by the singular value decomposition of X.

There is one issue when applying PCA to X in order to obtain the row coordinates, namely that X should be centred. For the centring, Gower and Hand (1996) prove that the first term of the singular value decomposition ($\lambda_1 u_1 v_1^t$) is equal to NX, where N is used for column centring such that $1^T(I-N)X = 0$. Therefore, PCA can be done without centring X, but keep in mind that the first term of the SVD should then be left out.

Another common method to evaluate MCA is by analysing the uncentred sum-of-squares matrix $X^T X$. This can be done by using the spectral decomposition, which results in

$$X^T X = D_r^{-1} D_c^{-1/2} Z^T Z D_c^{-1/2} = V \Lambda V^T$$

This method for evaluating $Z^T Z$, which is also called the *Burt* matrix, is computational more easy if the amount of columns in Z is considerable smaller than the amount of rows in Z. However, in our research we need to obtain the projections of the row coordinates and these can not be obtained by this method.

The accuracy of the representation of the samples in Equation 10 is given by

$$\frac{\lambda_2^2 + ... + \lambda_D^2}{\lambda_2^2 + ... + \lambda_D^2 + ... + \lambda_E^2} \tag{11}$$

where λ_1 is left out because of the centering problem. However, this measure tends to be very pessimistic with respect to the representation (Greenacre, 1984). An alternative is proposed by Kroonenberg and Greenacre (2004), where all λ are adjusted using

$$\lambda^* = (\frac{J}{J-1})^2(\lambda - \frac{1}{J})^2$$

The accuracy of the representation is now given more realistically using λ^* and Equation 11.

3.4 Correspondence analysis applied to rating data

Rating data can be seen as scaled data and is used often in surveys. The possible amount of values may differ and easy examples are 3 or 5-point scale data. Common example of answers of 3-point scale data are *agree*, *neutral* and *disagree*, and for 5-point scale data, *agree*, *slightly agree*, *neutral*, *slightly disagree* and *disagree*. These answers have a certain ordering and 2 extremes, in this case *agree* and *disagree*. In our research, the possible values of T-point rating data are integers ranging from 1 to T. Greenacre (1984) refer to them as poles and therefore call this data *bipolar*.

For the analysis of *bipolar* data, one theoretical condition should hold, and that is that the agreement of one statement which approves an entity, called A, should be equal to the disagreement of one statement that disapproves A. Emotionally this "scale invariance" may be questionable, but from a mathematical point of view, this should always be the case.

The coordinates acquired by applying CA to rating data is likely not to satisfy that condition. Consider a survey of 3 respondents and 4 questions and 5-point scale data, where a score of 5 corresponds to the highest score and 1 to the lowest score. One respondent gives one variable a 5 and the other three variables a 1. In the next matrix, the scales are reversed, meaning that a score of 1 corresponds to a high score and a score of 5 to a low score. For the same respondent, he now gives 3 variables a score of 5 and one variable a score of 1. The results of the analysis of both matrices should be the same as the content of matrices is still the same. However, due to the low row total for the respondent in the first matrix in comparison with the second matrix, the results are not equal when applying CA. Therefore, the condition of scale invariance will not hold. Note that we assume for both matrices that the column averages are equal.

To make sure this condition will hold, Greenacre (1984) proposed to double the matrix column-wise. The first J columns are the same as the original matrix, the second J columns are the reflected form of the data. As an example, in surveys, if one respondent gives a high rating to a certain value, it will give a low value to the variable in the second part of the doubled matrix. This low value is mirrored in the average of the scaled values, such that the distance between the low value and low pole is equal to the high value and the high pole.

This can also be explained more mathematically. Consider the rating matrix with I rows and J columns. Then, $y_{i,j}$ is the rating by observation i for variable j, where we consider the ratings to have positive associations with respect to the variable. The maximum rating for $y_{i,j}$ is given by T for all i, j. Next, we double the columns of the matrix, where the appended columns contain the negative associations, such that the matrix has I rows and $2J$ columns. The first J columns are denoted as $j+$, because these denote the positive associations of the ratings with respect to the variable and contains the value $y_{i,j}$. Next, the final J columns are denoted as $j-$ and measures the negative associations of a sample with respect to the variable, which are given by $T + 1 - y_{i,j}$.

Greenacre (1984) note that this doubling leads to a symmetry between the two poles of a scaled variable. Moreover, the scale invariance will hold, as both the positive and negative associations for all variables are included, and exchanging the poles would be equal to exchanging the columns.

After this doubling, we can apply regular correspondence analysis to this matrix.

However, there are some implications. First of all, the row total is the same for all observations, namely $J \cdot (T + 1)$. As there is no difference in row totals, this is not any more an essential part of the analysis that makes CA different from PCA.

3.5 K-means clustering

To cluster our data, we use K-means (MacQueen et al., 1967). This method is the most popular partitive clustering algorithm (Jain, 2010) and due to the fact that it's conceptually easy and the computation time is limited, it's suited for our research problem.

For initialisation, determine K as the amount of clusters and assign for each cluster one point that becomes a centroid. In the next step, every observation is assigned to the cluster of which the centroid is the most near, measured in Euclidean distance. Subsequently, the centroid of each cluster is computed, which is the average point of the allocated observations. In the next iterations, the assigning of the data points and computing of the centroids are repeated until convergence is reached. Convergence is reached if no points are allocated differently in subsequent iterations.

3.5.1 Full dimensional clustering

One application of clustering rating data is full dimensional clustering, where K-means clustering will be used. The distance between the centroids and the other data points in K-means is measured in Euclidean distance and the data of all dimensions will be used.

3.6 Simultaneous PCA and K-means

In this subsection, we propose two simultaneous methods, namely the reduced K-means algorithm (RKM, De Soete and Carroll (1994)) and the factorial K-means algorithm (FKM, Vichi and Kiers (2001)). Both methods use principal component analysis with K-means clustering in a coinciding approach.

To define both methods accurately, we introduce the matrices involved. First, consider P as the columnwise standardized and centred score matrix, which means that the mean and standard deviation of each column are equal to 0 and 1 respectively. Next, we define L as the loadings matrix with J rows and D columns, where J is the amount of variables and D is the amount of dimensions of the projected subspace. Next, we define Z_K as the indicator matrix that assigns every observation to one of the K clusters, and we denote G as the matrix containing the centroids of the clusters in the reduced space.

3.6.1 Reduced K-means

Given this matrices, both objective functions can be defined. The reduced K-means algorithm is solved in order to minimize the squared distance between the data points and the centroids in the reduced space, which are spanned by the loading matrix. The function which needs to be minimized in order to obtain the projections and the clusters is defined in Equation 12.

$$F_{Reduced}(Z_K, G, L) = \|P - Z_K G L^T\|^2 \tag{12}$$

3.6.2 Factorial K-means

Furthermore, the factorial K-means algorithm is formulated in such a way that the *within variance* of the separate clusters in the projected space is minimized. This means that implementing this algorithm leads to clusters where the differences between the estimated centroids and the data points, both in the reduced space, are relatively small. These clusters are obtained by minimizing Equation 13.

$$F_{Factorial}(Z_K, G, L) = \|PL - Z_K G\|^2 = \|PLL^T - Z_K G L^T\|^2 \tag{13}$$

3.6.3 Difference Factorial and Reduced K-means

RKM and FKM look very similar, but Timmerman et al. (2010) note that the difference in clustering accuracy can be pretty big. They show that the differences can best be explained by their error terms.

The objective function of Factorial K-means in Equation 12 can be rewritten such as in Equation 14. Timmerman et al. (2010) note that the the optimal value for $G = (Z_K^T Z_K)^{-1} Z_K^T PL$ such that P can be explained by H_K, L, P and E. Note that H_K is equal to $Z_K(Z_K^T Z_K)^{-1} Z_K^T$.

$$E_{Reduced} = P - Z_K G L^T = P - Z_K(Z_K^T Z_K)^{-1} Z_K^T PLL^T = P - H_K PLL^T \tag{14}$$

The error term for Factorial K-means is constructed in Equation 15, where we use the same optimal value for F and notation for H_K as for Reduced K-means.

$$E_{Factorial} = PLL^T - Z_K G L^T = PLL^T - Z_K(Z_K^T Z_K)^{-1} Z_K^T PLL^T = PLL^T - H_K PLL^T \tag{15}$$

Next, Timmerman et al. (2010) note that P can be expressed by 3 terms using above equations, namely a structural part ($Z_K G L^T$), the "subspace residuals" (EL^T), and the "complement residuals" ($E^\perp L^{\perp T}$). This partition is also given in Equation 16. Note that $E^\perp L^{\perp T}$ is equal to $P - PLL^T$ and that $E^T E^\perp = 0$.

$$P = Z_K G L^T + EL^T + E^\perp L^{\perp T} \tag{16}$$

Timmerman et al. (2010) speculate using algebraic analysis and verify by means of a simulation study that the quality of the clustering for Factorial K-means increases when the percentage of complement residuals inreases, and that the quality for the Reduced K-means clustering increases when the percentage of subspace residuals increases.

3.7 Simultaneous clustering and correspondence analysis to Rating data

In this section, we simultaneously apply K-means and correspondence analysis to rating data (CCA). This simultaneous approach has already been proposed for categorical data by Van de Velden et al. (2016). However, in this case the data is interpreted as continuous data. Moreover, the matrix needs to be doubled as discussed in Subsection 3.4, which did not happen in Van de Velden et al. (2016).

Consider the doubled matrix N. Next, we introduce the indicator matrix Z_K, which assigns I observations to one of the K clusters. Finally, we can introduce matrix $P = \frac{1}{n}Z_K^T N$, where n is chosen such that the sum of all elements of P sum to 1. P is now the matrix where the rows contains the sums of all the observations within a cluster. To be precise, the element $p_{k,j}$ is the sum of all elements for cluster k for variable j divided by n, such that, $p_{k,j} = \frac{1}{n}\sum_{i=1}^{I} z_{i,k} \cdot n_{i,j}$ (p, z and n are elements of P, Z and N respectively).

P remains scale invariant and therefore, we can still apply CA to P. This can be proven by

$$p_{k,J+j} = \frac{1}{n}\sum_{i=1}^{I} z_{i,k} \cdot n_{i,J+j} = \frac{1}{n}\sum_{i=1}^{I} z_{i,k} \cdot (T + 1 - n_{i,j})$$

such that $p_{k,J+j} + p_{k,j} = \frac{1}{n}\sum_{m=1}^{I} z_{m,k} \cdot (T+1)$, which is equal for all j given that they belong to the same cluster. Therefore, $p_{k,j}$ is a reflection of $p_{k,J+j}$, which results in a doubled matrix. Interesting is that we can now find n using

$$\sum_{j=1}^{J}\sum_{k=1}^{K} p_{k,j} + p_{k,J+j} = \frac{1}{n}\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{i=1}^{I} z_{i,k} \cdot (T+1) = \frac{J \cdot I \cdot (T+1)}{n} = 1$$

Next, consider $\tilde{P} = D_r^{-1/2}(P - rc^T)D_c^{-1/2}$, such that the matrix is standardized and centred. Moreover, we denote r as a vector of row totals corresponding to $P1_{2J}$ and c a column vector corresponding to $P^T 1_K$. Next, D_r and D_c are the corresponding diagonal matrices.

Note that we can rewrite $P - rc^T$ using that $n = I \cdot J \cdot (T+1)$ and $M = \mathbf{I}_I - \frac{1}{I}1_I 1_I^T$, where $\mathbf{I}$ is the indicator matrix and I is the amount of dimensions:

P - r c^T $= P - P1_{2J}1_K^T P = \frac{1}{n}Z_K^T N - \frac{1}{n^2}Z_K^T N 1_{2J} 1_K^T Z_K^T N$

$= \frac{1}{n}Z_K^T N - J \cdot (T+1)\frac{1}{(I \cdot J \cdot (T+1))^2)}Z_K^T 1_I 1_I^T N = \frac{1}{n}(Z_K^T \mathbf{I}_I N - \frac{1}{I}Z_K^T 1_I 1_I^T N) = \frac{1}{n}Z_K^T MN$

The formula to be minimized to obtain the cluster allocation using CA therefore becomes:

$$\min_{G,B,Z_K} ||\tilde{P} - D_r^{1/2} G B^T D_c^{1/2}||^2 = \min_{G,B,Z_K} ||D_r^{-1/2}(\frac{1}{n} Z_K^T M N) D_c^{-1/2} - D_r^{1/2} G B^T D_c^{1/2}||^2 \quad (17)$$

subject to $B^T D_c B = I$, where the projections of the rows and columns are the columns of G and B respectively.

This minimization problem can be solved by composing the SVD of $\tilde{P}$:

$$\tilde{P} = U \Sigma V' \quad (18)$$

Given that $B^T D_c B = I$, $\tilde{P}$ is approximated by the first D columns of $G = D_r^{-1/2} U \Sigma$, the principle coordinates of the rows, and $B = D_c^{-1/2} V$, the standard coordinates of the columns.

Next, we also have to minimize with respect to Z_K, which is captured in $\tilde{P}$. Van de Velden et al. (2016) note that minimizing Equation 17 is equal to

$$\max ||D_r^{1/2} G||^2 = max\ trace(G^T D_r G) = max\ trace(\Lambda^2) \quad (19)$$

subject to $B^T D_c B = I$.

Therefore, in order to obtain the cluster allocation, we can rewrite Equation 17, using Equation 18 and 19 and the fact that $G = D_r^{-1/2} U \Sigma = D_r^{-1/2} \tilde{P} V$, such that

$$\max_{Z_K} ||D_r^{1/2} D_r^{-1/2} \tilde{P} V||^2 = \max_{Z_K} ||D_r^{-1/2}(\frac{1}{n} Z_K^T M N) D_c^{-1/2} V||^2 \quad (20)$$

which can be rewritten as

$$\phi = ||\frac{1}{n} D_r^{-1/2} Z_K^T M N D_c^{-1/2} V||^2 = trace(V' D_c^{-1/2} N^T M Z_K D_r^{-1} Z_K^T M N D_c^{-1/2} V)$$

Moreover, we will show that $D_K = (Z_K^T Z_K) = I \cdot D_r$ by using

$$r = P 1_{2J} = \frac{1}{n} Z_K^T N 1_{2J}$$
$$r = \frac{J \cdot (T+1)}{I \cdot J \cdot (T+1)} Z_K^T 1_I$$
$$D_r = \frac{1}{I} diag(Z_K^T 1_I) = \frac{1}{I} Z_K^T Z_K = \frac{1}{I} D_K$$

such that

$$\begin{aligned}\max_{Z_K}\ \phi = \max_{Z_K}\ trace(V' D_c^{-1/2} N^T M Z_K D_r^{-1} Z_K^T M N D_c^{-1/2} V) \\ = \max_{Z_K}\ trace(V' D_c^{-1/2} N^T M Z_K D_K^{-1} Z_K^T M N D_c^{-1/2} V)\end{aligned}$$

Next, we introduce $Y = MND_c^{-1/2}V$. Moreover, the K-means objective can be denoted as (Van de Velden et al., 2016):

$$\min_{Z_K, G} ||Y - Z_K G||^2$$

where Y is the data matrix, Z_K the cluster allocation matrix and G the matrix containing the initial centroids.

Using this formulation, we can find the optimal value for G by using

$$G = (Z_K^T Z_K)^{-1} Z_K^T Y$$

Next, we will show that minimizing the K-means objective is equal to maximizing objective 20 using

$$\begin{aligned}\min_{Z_K, G} ||Y - Z_K G||^2 = traceY^T Y + traceG^T D_K G - 2traceG^T Z_K^T Y \\ = traceY^T Y + traceY^T Z_K D_K^{-1} D_K D_K^{-1} Z_K^T Y - 2traceY^T Z_K (D_K)^{-1} Z_K^T Y \\ = traceY^T Y - traceY^T Z_K D_K^{-1} Z_K^T Y \\ = traceY^T Y - trace(V' D_c^{-1/2} N^T M Z_K D_K^{-1} Z_K^T M N D_c^{-1/2} V)\end{aligned}$$

where the first term $(Y^T Y)$ is independent of Z_K and G. Consequently, we have shown that the two objectives are equal and that we can use K-means for clustering allocation.

The procedure of optimisation is started by assigning every observation to a random cluster. Next, we iteratively update B, G in one step and Z_K in the other step until the observations are not assigned to other clusters in sequential iterations or the maximum amount of iterations is reached.

3.8 MCA K-means

One example of coinciding multiple correspondence analysis and K-means is MCA K-means (Hwang et al., 2006). This method simultaneously reduces the dimensions of the data using MCA while allocating the clusters to observations in the reduced space using K-means.

First, we obtain indicator matrix Z by transforming rating matrix P, just as we have done before for MCA. Then, let Z_j be the indicator matrix corresponding to variable j, with dimensions I, q_j. Let F be a I, D matrix containing the values for the I observations in the reduced space of dimensions D. Next, we define W_j as the weight matrix of q_j rows and D columns, linearly transforming the columns of variable j to the reduced space. Next, we denote Z_K as the matrix that assigns I observations to K clusters and G the centroids of K clusters in D dimensions, so in the reduced space.

Moreover, 2 values have to be decided by the researcher, namely α_1 and α_2. The default choice for our research will be $\alpha_1 = \alpha_2 = 0.5$. Given these values and matrices, we can define MCA K-means by minimizing the following:

$$f = \alpha_1 \Sigma_{j=1}^{J} trace((F - Z_j W_j)^T (F - Z_j W_j)) + \alpha_2 trace((F - Z_K G)^T (F - Z_K G)) \quad (21)$$

with respect to F, W_j, Z_K and G. Moreover, the sum of α_1 and α_2 must be 1 and $F^T F = I$. The optimal matrices of Equation 21 can be obtained by an iterative procedure. For a detailed explanation of this optimisation, we refer to Hwang et al. (2006).

3.9 Evaluation measures

All cluster allocations will be evaluated based on two criteria. The first criteria is that we want the accuracy of the cluster allocations to be high. This is done by comparing the allocation with the real clustering. Next, the quality of the clusters will be assessed, regardless of the accuracy. Therefore, the clusters should be well separated from each other, while the clusters itself should be cohesive.

3.9.1 Adjusted Rand Index

First, we measure the accuracy of the allocation. As the "true" cluster allocation is known in our simulation study, we can assess the effectiveness of the different cluster allocations. Therefore, we will use the Adjusted Rand Index (ARI, Hubert and Arabie (1985)). The Rand Index is an index that measures the accuracy of the cluster, which is done by comparing the pairs of observations which belong to the same cluster in the estimation versus the true allocation. However, the Rand Index does not take into account chance, and therefore we use the Adjusted Rand Index.

The ARI is used to compare the true allocation with the estimated allocation and returns a value between -1 and 1, which indicates the precision of the estimated clustering. The higher the score, the more accurate the assignment of the observations. Also, a score of 0 indicates that the assignment of the clusters is equal to random assignment.

3.9.2 Average Overall Silhouette Width

Next, we also compare the algorithms using the Average Overall Silhouette Width (AOSW Rousseeuw (1987)). A high value for the AOSW corresponds to a cluster allocation where the observations within a cluster are close to each other (cohesive), while the distance to other clusters is relatively large (separation).

Consider observation i, which belongs to cluster A. Then, the average dissimilarity of observation i with respect to the other observations within cluster A is denoted as $a(i)$. All dissimilarity can estimated using Euclidean distance. Next, consider $d(i, C)$, which is the average dissimilarity of i to all observations within cluster C. The cluster with the smallest average dissimilarity, excluding cluster A, is denoted as $b(i) = \min_{C \neq A} d(i, C)$.

Finally, we can get silhouette width of i by

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))}$$

The AOSW is obtained by taking the average of the silhouette width for all points.

4 Data

In this research, the proposed methods will be applied to simulated data and on an existing dataset.

4.1 Simulation Study

Our goal of the simulation study is to further investigate the differences between the different approaches of clustering high-dimensional data using dimension reduction, in order to gain more insights which methods should be preferred when dealing with this problem.

For the simulated dataset, we mimic a survey with 500 respondents. The amount of variables varies among 16, 32 and 64. The possible answers vary among 3-point scale data, 5-point scale data and 9-point scale data. The predetermined amount of clusters is 4, where each cluster is equally sized. Moreover, the amount of variance varies among the simulations.

Next, we use 2 separate cases of distributions. In the first case, all variables are important in determining the correct cluster allocations. In the second case, only one part of the variables contain the taxonomic information, while the other variables have the same distribution for every cluster. These second part of variables do not carry any relevant information for clustering the observations. Vichi and Kiers (2001) refer to these variables as "masking variables", as improper analysis of these data may lead to the masking or obscuring of the taxonomic information of the relevant variables.

All the data will be simulated 50 times. An overview of the different settings is given in Table 3. The total amount of different settings will be $3x3x2x2 = 36$.

Table 3: Settings of simulation study

Simulation settings	
Amount of variables	16, 32, 64
rating scale	3, 5, 9
Amount of variance	High, Low
Presence of masking variables	No, Yes
Amount of clusters	4
Amount of observations	500
Amount of simulations	50

The ratings for the variables are simulated using a normal distribution. Given the outcome of the normal distribution, we round to the nearest integer to retain the structure of the rating data. The preference for the variables is the same for the respondents in the same cluster.

In the first case, all variables are important. The variables are split into 4 variable groups, which are all equal sized. The variables from the same variable group all have the same distribution. Each observation comes from 1 of the 4 clusters, and the observations from a cluster all like 1 of the 4 variable groups, dislike 1 of the 4 variable groups, and are neutral on 2 of the 4 variable groups. Eventually, all products are liked and disliked once by a cluster. An exact representation of this scheme is given in Table 4.

Table 4: Allocation table, which shows how the cluster corresponds to the variable groups, of case 1.

Clusters	**Positive**	**Negative**	**Neutral**
1	1	2	3, 4
2	2	3	1, 4
3	3	4	1, 2
4	4	1	2, 3

Values are the variable groups, the columns indicate the distribution corresponding to the preference, and the rows denote the cluster to which the observation belongs

In the second case, not all variables are important for clustering. The variables are split into 3 different variable groups, where the third group containing the masking variables is the biggest. The combined size of variable group 1 and 2 is equal to $6+\dfrac{amount\ variables}{4}$, such that the percentage of masking variables is decreasing when the amount of variables increase. Each observation comes from 1 of the 4 clusters. The observations differ in distribution for variable group 1 and 2, which means that the variables from this group carry the taxonomic information. All the observations have similar distribution for category group 3, which are the masking variables and do not contain relevant information in allocating the clusters. How each cluster is related to the variable groups, is given in Table 5.

Table 5: Allocation table, which shows how the cluster corresponds to the variable groups, of case 2.

Clusters	Positive	Negative	Masking variables
1	1	2	3
2	2	1	3
3	1,2	-	3
4	-	1,2	3

Values are the variable groups, the columns indicate the distribution corresponding to the preference, and the rows denote the cluster to which the observation belongs

Next, the data matrix P can be simulated. Let $p_{i,j}$ be the value for P at row i for variable j, then the data can be generated as

$$p_{i,j} \sim round(N(\mu, \sigma))$$

where μ is a high, normal or low value dependent on how cluster c is related to the variable group of j. The values of μ are also dependent of the rating scale. σ is dependent on if the presence of variance is high or low and on the scale of the data. Moreover, σ is higher for masking variables. An overview of the values for μ and σ is given in C. Given that T is the scale for the ratings, the simulated value is equal to $min(T, max(1, p_{i,j}))$, such that the data falls within the range of $[1, T]$.

The differences in accuracy and cluster quality will be interesting, and we have some expectations. First of all, we expect that the higher the amount of variables and the higher the variance, full dimensional clustering has a relative low cluster accuracy. This is due to the fact that the dimension reduction techniques try to capture the underlying structure of the data, and therefore are more likely to filter out the noise. Moreover, we expect that MCA is going to perform relatively well when the rating scale is low, because then the data is relatively more similar to categorical data then when the rating scale is high.

Subsequently, we expect the simultaneous methods to allocate more accurate with respect to the other methods when not all variables are important for clustering, as these methods jointly cluster and reduce dimensions. This has also been illustrated for numerical data by Vichi and Kiers (2001).

4.2 Survey data Malaysian Public University

Next, we illustrate the purpose of joint clustering and dimension reduction by applying our most effective method to survey data. This survey data is taken from 280 students at Malaysian Public University. The goal of the survey was to measure the satisfaction of the students of 14 different facilities, where the respondents could rate each facility from 1 (strongly dissatisfied) to 10 (strongly satisfied). The data can be found through this link.

The goal of this second part is to gain more insights in the application of combined dimension reduction and clustering, and how to interpret the results.

5 Results

In this section, we will discuss the goal and the results of the simulation study. Also, the usage of combined dimension reduction and clustering will be illustrated using an example. First of all, the goal of the simulation study is to compare the different clustering approaches and how they react to the different simulated datasets. Subsequently, it's interesting to verify if the results are in line with our conjectures and previous research.

The 8 methods that are applied are the following. First, K-means without dimension reduction is implemented, which is also called full dimensional clustering. Moreover, we apply 3 tandem approaches, which use different dimension reduction techniques, namely PCA, CA and MCA, and K-means for clustering. Next, we will implement 4 simultaneous techniques, namely Reduced K-means, Factorial K-means, MCA K-means and a simultaneous method that uses CA and K-means (CCA).

The methods will be evaluated based on clustering allocation accuracy and cluster quality. Moreover, the methods will be assessed on different datasets. The data is differentiated on the amount of variables, rating scale, amount of variance and the presence of masking variables. For all the methods a total amount of 25 random starts for initialisation will be used, such that we can make fair comparisons. The amount of dimensions we retain for every dimension reduction technique is 3. Moreover, we impose that the amount of clusters for each method is 4. The code used can be found at https://github.com/Sbennema/thesis and in Appendix E.

5.1 Results simulation study

The results for the clustering accuracy is given in Table 6 and Table 7. From both tables can be obtained that the Factorial K-means does not allocate the cluster better than random assignment, as most scores for the ARI are close to zero. According to Timmerman et al. (2010), FKM is likely to perform worse if the amount of masking variables is low. In that case, there is a relative high amount of "subspace" variance in comparison to variance of the complement residuals. However, we could not find an increase in accuracy of FKM when masking variables where present. Using the obtained matrices of Reduced K-means in Equation 16, we found that the average percentage of complement variance is indeed very low, below 10 %.

Moreover, CCA allocates worse with respect to the other methods. As the method had some interesting algebraic properties, we do not recommend to use this method due to the low clustering accuracy. However, we see that the method performs relatively better in the case with masking variables.

Furthermore, we can conclude that when there are no masking variables, tandem analysis using MCA and MCA K-means tend to perform slightly better than the other methods, especially when the rating scale is low. This is probably because these methods analyse the data as categorical data instead of continuous data, and the lower the rating scale, the less dependencies in the data MCA and MCA K-means miss in their analysis compared to the other methods. Next, MCA K-means has relative low clustering accuracy in the case with masking variables, which was not expected due to the fact that it was a simultaneous method.

From Table 7 can be obtained that when the masking variables are present, Reduced K-means tends to perform better than the other methods. This is line with the statements of Vichi and Kiers (2001) that simultaneous methods are better able to cluster on the variables that carry the taxonomic variables when there are masking variables present. One interesting observation is also that CCA and Reduced K-means both minimize the error of the full dimensional data points and centroids, while Factorial K-means and MCA K-means minimize based on the error of the projections of the observations and the centroids. This may be related to the fact that Reduced K-means and CCA perform well on Case 2 with respect to Case 1.

3 methods, namely tandem analysis using CA and PCA and full dimensional clustering all perform equally in all cases. These methods are generally outperformed by the methods using MCA when there are no masking variables, and outperformed by Reduced K-means when there are masking variables present. Hence, we do not recommend to use this clustering methods.

Overall, the Reduced K-means method has the highest clustering accuracy. This method has the highest accuracy when masking variables are present, and has an average accuracy when there are no masking variables.

Next to clustering accuracy, the clustering quality was also assessed by the average overall silhouette width. THe AOSW was estimated using the clustering allocation of the algorithms and the values of the observations from the full dimensional dataset. Due to the high overlap of the clusters, the scores for the average overall silhouette width were all very close to zero and therefore no interesting comparisons could me made. The results of the AOSW for all cases and all methods can be found in Appendix D.

Table 6: Adjusted Rand Index in the first case without masking variables

Low variance									
	Variables=16			Variables=32			Variables=64		
Methods	T=3	T=5	T=9	T=3	T=5	T=9	T=3	T=5	T=9
FDC	0.19	0.32	0.42	0.45	0.65	0.75	0.79	0.92	0.96
Tandem PCA	0.20	0.35	0.43	0.46	0.64	0.74	0.79	0.91	0.95
Tandem CA	0.20	0.35	0.43	0.45	0.64	0.74	0.78	0.91	0.95
Tandem MCA	0.22	0.35	0.40	0.60	0.79	0.77	0.88	0.97	0.97
RKM	0.18	0.32	0.43	0.46	0.66	0.76	0.77	0.91	0.95
FKM	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00
CCA	0.12	0.17	0.19	0.20	0.24	0.27	0.28	0.40	0.52
MCA K-means	0.20	0.36	0.37	0.59	0.74	0.76	0.85	0.95	0.95
High variance									
	Variables=16			Variables=32			Variables=64		
Methods	T=3	T=5	T=9	T=3	T=5	T=9	T=3	T=5	T=9
FDC	0.05	0.09	0.14	0.12	0.22	0.37	0.39	0.61	0.73
Tandem PCA	0.06	0.10	0.16	0.12	0.25	0.38	0.36	0.59	0.71
Tandem CA	0.06	0.10	0.15	0.12	0.25	0.38	0.36	0.59	0.71
Tandem MCA	0.07	0.14	0.10	0.18	0.33	0.34	0.58	0.78	0.78
RKM	0.05	0.09	0.13	0.11	0.23	0.36	0.31	0.58	0.69
FKM	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00
CCA	0.03	0.06	0.09	0.09	0.14	0.18	0.17	0.23	0.26
MCA K-means	0.07	0.11	0.09	0.21	0.38	0.36	0.55	0.71	0.73

Table 7: Adjusted Rand Index in the first case wit masking variables

Low variance									
	Variables=16			Variables=32			Variables=64		
Methods	T=3	T=5	T=9	T=3	T=5	T=9	T=3	T=5	T=9
FDC	0.31	0.61	0.74	0.37	0.70	0.83	0.57	0.81	0.90
Tandem PCA	0.32	0.57	0.73	0.38	0.73	0.82	0.59	0.84	0.92
Tandem CA	0.32	0.58	0.73	0.38	0.73	0.82	0.59	0.84	0.92
Tandem MCA	0.27	0.56	0.61	0.43	0.62	0.62	0.61	0.76	0.79
RKM	0.45	0.69	0.79	0.52	0.75	0.85	0.65	0.85	0.93
FKM	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
CCA	0.20	0.31	0.53	0.25	0.53	0.74	0.34	0.70	0.84
MCA K-means	0.26	0.37	0.42	0.30	0.44	0.43	0.39	0.62	0.62
High variance									
	Variables=16			Variables=32			Variables=64		
Methods	T=3	T=5	T=9	T=3	T=5	T=9	T=3	T=5	T=9
FDC	0.14	0.24	0.34	0.13	0.30	0.45	0.21	0.44	0.60
Tandem PCA	0.16	0.24	0.33	0.17	0.29	0.46	0.24	0.42	0.63
Tandem CA	0.16	0.24	0.33	0.17	0.29	0.46	0.24	0.43	0.63
Tandem MCA	0.16	0.23	0.21	0.18	0.27	0.22	0.23	0.33	0.30
RKM	0.22	0.41	0.53	0.23	0.47	0.60	0.30	0.60	0.72
FKM	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
CCA	0.03	0.11	0.20	0.09	0.18	0.28	0.14	0.24	0.39
MCA K-means	0.16	0.22	0.22	0.17	0.23	0.22	0.22	0.29	0.28

5.2 Results students survey

In this subsection an illustration is given of how a combined clustering and dimension reduction can be applied and interpreted. Our data is the student satisfaction survey considering 280 students rating 14 facilities from 1 tot 10. Given the results from the simulation study, Reduced K-means is the most suitable method.

Before executing RKM, the amount of dimensions to which the data is reduced and the amount of clusters is needed. The right amount of dimensions are chosen based on the percentage of variance explained using PCA, which can also be found at Equation 5. From Figure 1 can be observed that the first dimension is very influential, and after that the influence of dimensions is decreasing rapidly. Therefore, we have chosen to retain 2 dimensions using dimension reduction.

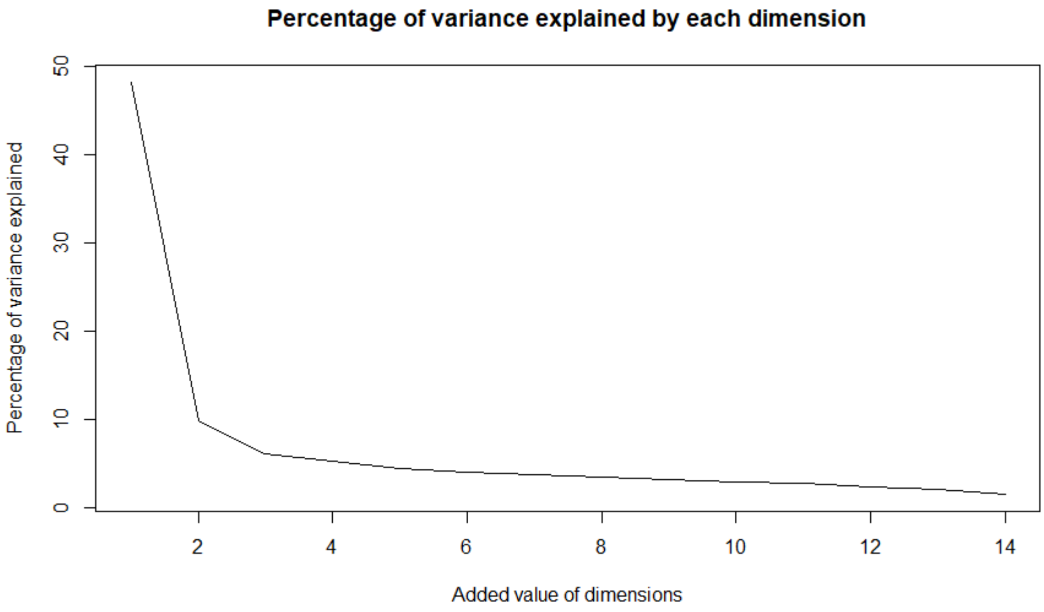

Figure 1

Next, the amount of clusters are based on the error term of the criterion, which is given in Figure 2. Given this figure, we have decided to obtain 5 clusters, as we see a slight decrease when that amount of clusters is used. However, 3 clusters could also have been chosen.

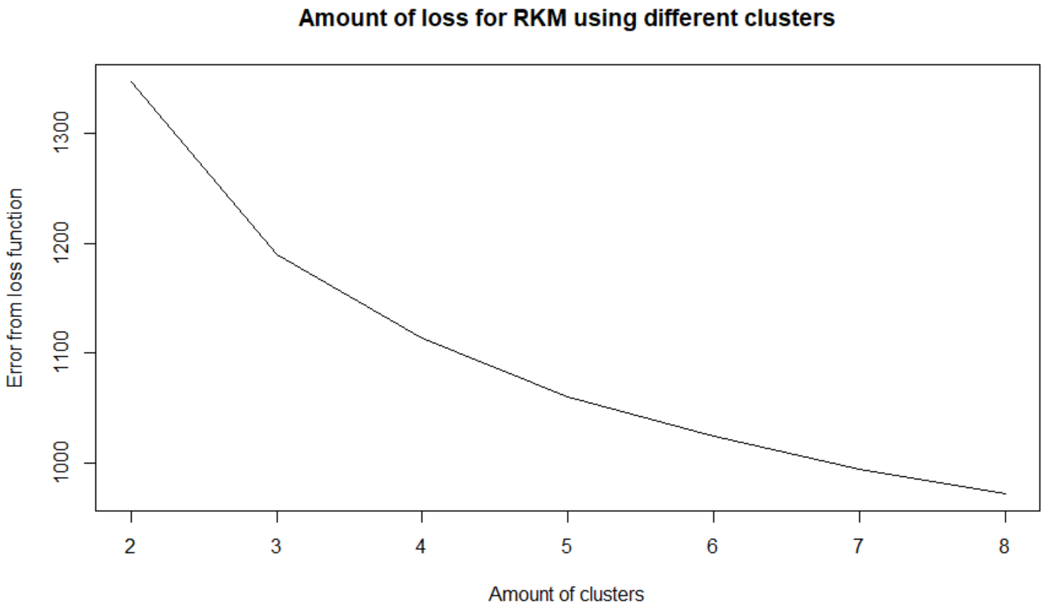

Figure 2

Given the amount of dimensions and the amount of clusters, the Reduced K-means is applied to the survey data with 100 random starts. The cluster allocation of this method is given in Figure 3, where respondents with the same figure belong to the same cluster.

To further interpret this result, we are interested in how the variables influence each dimension, which is given in Table 8. Note that this table is equal to L in Equation 12. The first dimension can be interpreted as the amount of pessimism of the respondents towards all the facilities, as all scores for the variables for that dimension are negative and in somewhat the same range. This means that a high score for this dimension generally leads to a low overall score for the respondent to the survey. The respondents belonging to the cluster indicated by the diamonds in Figure 3 can now be interpreted as the pessimistic group, while the respondents which are indicated by an "X" can be interpreted as optimistic respondents. As the percentage of explained variance is really big for this dimension, the influence of the scores for this dimension is huge.

The second dimension can be interpreted differently, as $Bursary$, $Health\ Centre$, $Sports\ Centre$ and $Islamic\ centre$ are all correlated negatively to the second dimension, and all the other variables are positively correlated to this dimension. This implicates that a high score from a respondent in this dimension generally corresponds to a negative feeling towards the above named variables and vice versa. However, note that percentage of explained variance is low for this dimension, and therefore the influence of the scores for this dimension is little.

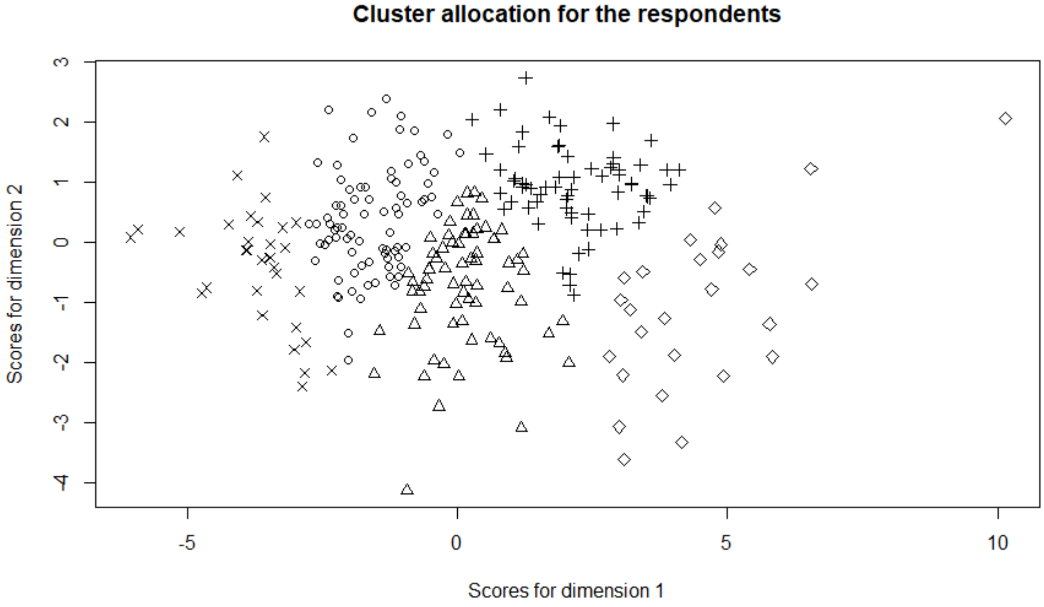

Figure 3

Table 8: Contribution of variables to first and second dimension

Variable	Dimension 1	Dimension 2
Bursary	-0.24871	-0.27507
Health Centre	-0.19425	-0.52685
Library	-0.24828	-0.35914
Sport Centre	-0.27519	-0.2639
Islamic Centre	-0.23707	-0.3245
Auto teller Machine	-0.27263	0.015982
Residential Collage	-0.25107	0.13287
Transportation Centre	-0.28089	0.120923
Wireless Internet	-0.26279	0.254405
Parking Lot	-0.25988	0.382486
Toilet	-0.29332	0.157442
Cafeteria	-0.29531	0.167657
Cleanliness	-0.2953	0.210734
Welfare	-0.30582	0.044508

6 Conclusion

In this research, the goal was to answer the following research question:

How do the different dimension reduction methods and full dimensional method compare with respect to clustering accuracy and quality when applied to high-dimensional rating data?

To answer this question, we used 8 different methods, namely K-means on the full dataset, 3 tandem methods using PCA, CA and MCA and 4 simultaneous methods. These simultaneous methods are Reduced K-means, Factorial K-means, MCA K-means and our own version of cluster correspondence analysis. This version of CCA also was the first application in which correspondence analysis to continuous data and K-means was executed simultaneously.

All these methods have been applied to different simulated datasets. The simulations varied in the amount of masking variables present, the amount of variance, the amount of variables and the scale of the rating data.

The results of the study showed that MCA and MCA K-means generally both obtained higher clustering accuracy than the other methods when there are no masking variables present, especially when the rating-scale is low.

When there are masking variables, Reduced K-means tends to perform better. This is in line with the conjecture that simultaneous methods tend to perform better when only a subset of variables are relevant for the clustering. This was due to the fact that the dimension reduction is dependent on the clustering. Therefore, the variables that carry the information for the clustering allocation are more likely to be taken into account in the dimension reduction step, which leads to a higher clustering accuracy.

However, as our version of cluster correspondence analysis was an interesting experiment, the values for the ARI were significantly lower than the other methods, especially when there were no masking variables present. Moreover, Factorial K-means did not allocate the clusters well, which is probably due to the high amount of "subspace" residuals.

Tandem PCA, Tandem CA and full dimensional clustering all performed equally well in all cases, but most of the occasions, one of the other methods achieved higher clustering accuracy.

Finally, we found that increasing the amount of dimensions does not necessarily lead to lower clustering accuracy of the full dimensional clustering algorithm. Nevertheless, when the variables added are masking variables, Reduced K-means should be preferred over tandem approaches and full dimensional clustering.

The quality of the clusters were hard to access, as there was a high overlap in the simulated data and therefore, all values for the average overall silhouette width were close to zero.

To illustrate the usage of dimension reduction combined with clustering, we have applied Reduced K-means to a survey concerning student satisfaction of facilities. In that section, we have shown how to choose the right amount of dimensions, the right amount of clusters, and how to interpret the cluster allocations in the reduced space.

When comparing correspondence analysis to principal component analysis in this research, we can not confidently say that one method should be preferred over the other. In the Tandem approach, the accuracy is even, while CCA has lower overall clustering accuracy then RKM but higher than FKM. However, we can say that for this simulated data, simultaneous methods that minimize the error in the projected space should be avoided.

The most suitable method to cluster high-dimensional rating data using dimension reduction in the situation that there is no information on the amount of masking variables present is, according to this research, Reduced K-means. This method performs similar to most methods when there are no masking variables, and tends to outperform the other methods when these masking variables are present.

When there are no masking variables, other methods such as tandem approaches may lead to equal results. In the situation that rating scale is low and there are no masking variables, MCA should be favored as a dimension reduction technique when clustering.

For further research concerning the topic of clustering high-dimensional data using dimension reduction, we recommend to take a further look at the relationship between the increasing amount of masking variables and the accuracy of the clustering algorithms. We found an interesting shift in the accuracy of all methods by changing this parameter, which was less present when we changed the amount of variance, rating scale or amount of variables.

Note that in our research we remained the amount of clusters constant at 4 and that the size of each cluster was equal. This is not necessarily the case in most clustering problems, such that this could be an interesting factor to dive further into for further research. Moreover, we assumed the data was normally distributed, but rating-scale data could also follow a multinomial distribution.

Subsequently, we did use a low amount of random starts in order to to limit the running time. Increasing the amount of random starts will decrease the probability that the result is a local optimum, and therefore will lead to better results.

Furthermore, our research is limited to the influence of dimension reduction on clustering using sequential and contemporaneous approaches, but there are far more techniques that can be used to accurately cluster high-dimensional data, but that was beyond the scope of our research. Nevertheless, these methods may be more hard to interpret.

Finally, only K-means was used as a clustering method in combination with dimension reduction. Other clustering methods, such as Gaussian Clustering and K-mediods clustering, could also lead to interesting results.

A Spectral theorem

The spectral theorem, also called the eigendecompsition, is a tool used in linear algebra that splits a matrix into eigenvalues and eigenvectors. The spectral decomposition can be decomposed for any squared, symmetric A and can be denoted as in Equation 22.

$$AV = V\Lambda \tag{22}$$

where V is the matrix containing the eigenvectors in the columns and Λ the diagonal matrix containing the eigenvalues of A. Every eigenvalue λ_j at row j of the matrix corresponds to the eigenvector at columns j of V, v_j. This eigenvalues are also called the characteristic polynomial and can be obtained by solving $det(A - \lambda I) = 0$. The eigenvectors are orthogonal, such that $v_i^T v_j = 0$ if $i \neq j$, but for the same vectors hold that $v_i^T v_i = 1$. (scales are arbitrary, and therefore we impose standardisation).

Given corresponding eigenvalues and eigenmatrices, we can compose $Av_i = \lambda_i v_i$ for all i given that $1 \leq i \leq J$. Moreover, given that the vectors are orthogonal, we can also find that $V^T V = I$, we can rewrite Equation 22 as

$$A = V\Lambda V^T \tag{23}$$

Moreover, given that all eigenvalues are bigger than zero, than A is positive definite. Moreover, if the eigenvalues are equal to zero or positive, A is semi-positive definite. Further information on the spectral decomposition can be found in Gower and Hand (1996).

B Singular Value Decomposition method

The singular value decomposition method (SVD) is a factorisation technique used in matrix algebra. The projections of PCA, CA and MCA can all be derived using the SVD. The singular value decomposition has first been used by Eckart and Young (1936), although they named it the lower rank estimation. Greenacre (1984) proposed the method in relationship with CA, and therefore we will follow the theory in their book.

SVD denotes one real matrix as the product of three matrices, where each of the matrices has their own properties. This decomposition is given in Equation 24,

$$A = U\Sigma V^T \tag{24}$$

where Σ is a diagonal matrix of positive values and has K rows and columns. Next, U and V are matrices that contain unit vectors in the columns, such that $U^T U = V^T V = I$. Moreover, the columns in U and V are orthogonal, such that for columns $[u_1, ..., u_K]$

of U and $[v_1, ..., v_K]$ of V, $u_k^T u_{k'} = 0$ and $v_k^T v_{k'} = 0$ for $k \neq k'$. Furthermore, given that rank$(A) = K$ and A has n rows and p columns, U has n rows and V has p rows. The vectors u_k and v_k are called left singular and right singular vectors and the value α_k, the diagonal element on row k of Σ, is called the corresponding singular value. The singular value α_k indicates the magnitude of the corresponding matrix $u_k v_k^T$. Moreover, we can rewrite Equation 24 as $A = \sum_{k=1}^{K} \alpha_k u_k v_k^T$, such that is becomes the sum of all the outer products times the corresponding singular value. The matrix will be decomposed in such a way that the diagonals of Σ as descending, such that α_1 is the largest and α_K is the smallest. If all $\alpha_k \neq \alpha_{k'}$ for $k \neq k'$, this decomposition is unique.

Given the descending order of the singular values in Σ and the fact that these values correspond to the magnitude of outer product of singular vectors, we could only use a subset of vectors and singular values for approximation. Consider $K^* < K$, and only select the first K^* singular values of Σ and first K^* singular vectors of U and V. These new matrices will be denoted as $\Sigma^*, U^* and V^*$. Therefore, we can approximate matrix A by using the linear combinations with the most impact, which can be denoted as follows:

$$\tilde{A} = U^* \Sigma^* V^{*T}$$

The error term can be denoted as the complete decomposition minus the approximation, which is therefore:

$$A - \tilde{A} = U \Sigma V^T - U^* \Sigma^* V^{*T} \tag{25}$$

Given that $A = \sum_{k=1}^{K} \alpha_k u_k v_k^T$, Equation 25 can be solved as $A - \tilde{A} = \sum_{k=K*+1}^{K} \alpha_k u_k v_k^T$. Therefore, the approximation is accurate if $\sum_{k=K^*+1}^{K} \alpha_k$ is relatively small.

C Simulation settings

Table 9: Averages for normal distribution in simulation study

	T=3	T=5	T=9
low average	1,5	2	3
medium average	2	3	5
high average	2,5	4	7

Table 10: Standard deviation for normal distribution in simulation study

	case 1					
	T=3		T=5		T=9	
	low std	high std	low std	high std	low std	high std
negative/postive distribution	1	1,5	1.67	2.5	3	4.5
Neutral distribution	2	3	3.33	5	6	9
	case 2					
	T=3		T=5		T=9	
	low std	high std	low std	high std	low std	high std
negative/postive distribution	0.75	1	1.25	1.67	2.25	3
Masking variable	1.5	2	2.5	3.33	4.5	6

D Results AOSW

Table 11: Average silhouette width in the first case without masking variables

	Low variance								
	Variables=16			Variables=32			Variables=64		
Methods	T=3	T=5	T=9	T=3	T=5	T=9	T=3	T=5	T=9
FDC	0.06	0.06	0.07	0.04	0.05	0.06	0.04	0.05	0.07
Tandem PCA	0.05	0.06	0.07	0.04	0.05	0.06	0.04	0.05	0.07
Tandem CA	0.05	0.06	0.07	0.04	0.05	0.06	0.04	0.05	0.07
Tandem MCA	0.03	0.03	0.03	0.03	0.04	0.05	0.04	0.05	0.06
RKM	0.06	0.06	0.07	0.04	0.05	0.06	0.04	0.05	0.06
FKM	0.03	0.03	0.03	0.01	0.01	0.01	0.00	0.00	0.00
CCA	0.04	0.03	0.03	0.01	0.02	0.02	0.01	0.02	0.03
MCA K-means	0.03	0.02	0.03	0.01	0.03	0.00	0.03	0.01	0.04
	High variance								
	Variables=16			Variables=32			Variables=64		
Methods	T=3	T=5	T=9	T=3	T=5	T=9	T=3	T=5	T=9
FDC	0.05	0.06	0.06	0.03	0.03	0.04	0.02	0.03	0.03
Tandem PCA	0.05	0.05	0.05	0.03	0.03	0.03	0.02	0.03	0.03
Tandem CA	0.05	0.05	0.05	0.03	0.03	0.03	0.02	0.03	0.03
Tandem MCA	0.02	0.01	0.00	0.01	0.01	0.01	0.01	0.02	0.02
RKM	0.05	0.05	0.06	0.03	0.03	0.04	0.02	0.03	0.03
FKM	0.03	0.03	0.03	0.01	0.01	0.01	0.00	0.00	0.00
CCA	0.04	0.04	0.04	0.02	0.02	0.02	0.01	0.01	0.01
MCA K-means	0.01	0.05	0.01	0.03	0.01	0.05	0.02	0.06	0.02

Table 12: Average silhouette width in the second case with masking variables

Low variance									
	Variables=16			Variables=32			Variables=64		
Methods	T=3	T=5	T=9	T=3	T=5	T=9	T=3	T=5	T=9
FDC	0.06	0.06	0.07	0.03	0.03	0.04	0.02	0.02	0.02
FDC	0.06	0.06	0.07	0.03	0.03	0.04	0.02	0.02	0.02
Tandem PCA	0.06	0.07	0.08	0.03	0.04	0.05	0.02	0.03	0.03
Tandem CA	0.06	0.07	0.08	0.03	0.04	0.05	0.02	0.03	0.03
Tandem MCA	0.04	0.06	0.07	0.03	0.04	0.04	0.02	0.02	0.03
RKM	0.06	0.07	0.08	0.03	0.04	0.05	0.02	0.03	0.03
FKM	0.02	0.02	0.01	0.01	0.01	0.00	0.00	0.00	0.00
CCA	0.04	0.04	0.06	0.02	0.03	0.04	0.01	0.02	0.03
MCA K-means	0.04	0.03	0.04	0.03	0.05	0.02	0.02	0.02	0.03
High variance									
	Variables=16			Variables=32			Variables=64		
Methods	T=3	T=5	T=9	T=3	T=5	T=9	T=3	T=5	T=9
FDC	0.06	0.06	0.06	0.03	0.03	0.03	0.02	0.02	0.02
Tandem PCA	0.05	0.06	0.06	0.03	0.03	0.03	0.02	0.02	0.02
Tandem CA	0.05	0.06	0.06	0.03	0.03	0.03	0.02	0.02	0.02
Tandem MCA	0.03	0.03	0.02	0.02	0.02	0.01	0.01	0.01	0.01
RKM	0.06	0.06	0.06	0.03	0.03	0.03	0.02	0.02	0.02
FKM	0.03	0.02	0.02	0.01	0.01	0.01	0.00	0.00	0.00
CCA	0.05	0.05	0.05	0.02	0.02	0.02	0.01	0.01	0.01
MCA K-means	0.01	0.03	0.01	0.02	0.01	0.02	0.01	0.02	0.01

E Code

```
### Data simulation case 1

library(matrixStats)
library(csv)


dist_emo = function(emo, am_cat, am_rat, std_dev){
# Function that returns value given the emotion, the rating scale, and whether the std_dev is high/low
    if (am_rat==3){
    mult_rat= 0.5
  }
  else if(am_rat==5){
    mult_rat=1
  }else{
    mult_rat=2
  }
  if(std_dev=="low"){
    std_dev = am_rat/3
  }
  else{
    std_dev = am_rat/2
  }
   # given the emotion, the data is simulated from the normal distribution
  if(emo=="neg"){
    a= rnorm(am_cat, (1+mult_rat), std_dev)
  }
  if(emo=="neu"){
    a= rnorm(am_cat*2, (1+am_rat)/2, 2*std_dev)
  }
  if (emo=="pos"){
    a= rnorm(am_cat, am_rat-mult_rat, std_dev)
  }
  # Data is rounded such we get rating data
  a = round(a)
  a[a<1] =1
  a[a>am_rat] = am_rat
  return(a)
}


sim_mat = function(am_var, am_rat, std_dev){
  # Funtion that simulates 500 obsevrations, where the observations belong to one of the clusters
  # Amount of variables, rating scale and high or low standard deviation is dependent on the input
  n_obs=500
  mat = data.frame()
  for (i in c(1:4)){
    for (j in c(1:(n_obs/4))){
      ob_i = cl_i(i, am_var, am_rat, std_dev)
      # print(ob_i)
      mat = rbind(mat, ob_i)
    }}
  return(mat)
}

cl_i = function(clus_i, am_var, am_rat, std_dev){
  # Given the cluster of an observation, the variables are simulated
  am_cat = am_var/4
  row_v= c(dist_emo("pos", am_cat, am_rat, std_dev), dist_emo("neg", am_cat, am_rat, std_dev), dist_emo("neu", am_cat, a
  val =  er(row_v, clus_i*am_cat-am_cat)
  val = cbind(t(c(val)), clus_i)
  return(val)
}

er <- function(x, n = 1) {
  ## makes sure the simulations are different for the different functions
  if (n == 0) x else c(tail(x, -n), head(x, n))
}



 write_file = function(am_sim, am_var, am_rat, std_dev){
  ### given the settings, the files are generated to this path
  path = paste("C:\\Users\\siets\\Documents\\E&OR\\Thesis\\Proposal\\Simulatie_case_1",am_var,am_rat,std_dev,sep="_")
  dir.create(path)
  for (i in c(1:am_sim))
  {
    sample = sim_mat(am_var, am_rat, std_dev)
    file = paste(i, "sim.csv", sep="_")
    pa_file= paste(path, file, sep="\\")
    write.csv(sample, file = pa_file, row.names = FALSE)
  }
  return(print("done"))
}

## write 50 simulation files
Sim_case_1 = function(am_sim){
  ### simulates all datasets with the different settings, am_sim times
  for (i in c(3,5,9)){
    for (j in c(16, 32, 64)){
      for (std in c("low", "high")){
        set.seed("1234")
        sample = write_file(am_sim, am_var=j, am_rat=i, std_dev=std)
      }
    }
  }
  return(print("case_1"))}



### Data simulation case 2

library(matrixStats)
library(csv)


dist_emo = function(emo, am_cat, am_rat, std_dev){
  # Function that returns value given the emotion, the rating scale, and whether the std_dev is high/low
  if (am_rat==3){
    mult_rat= 0.5
  }
  else if(am_rat==5){
    mult_rat=1
  }else{
    mult_rat=2
  }

  if(std_dev=="low"){
    #USed to bed std_dev = am_rat/4
    std_dev = am_rat/4
  }
  else{
    std_dev = am_rat/3
  }
  # given the emotion, the data is simulated from the normal distribution
  if(emo=="neg"){
    a= rnorm(am_cat, (1+mult_rat), std_dev)
  }
  if(emo=="neu"){
    a= rnorm(am_cat, (1+am_rat)/2, std_dev*(2))
  }
  if (emo=="pos"){
    a= rnorm(am_cat, am_rat-mult_rat, std_dev)
  }
  # Data is rounded such we get rating data
  a = round(a)
  a[a<1] =1
  a[a>am_rat] = am_rat
  return(a)
}


sim_mat = function(am_var, am_rat, std_dev){
  # Funtion that simulates 500 obsevrations, where the observations belong to one of the clusters
  # Amount of variables, rating scale and high or low standard deviation is dependent on the input
  n_obs=500
  mat = data.frame()
  for (i in c(1:4)){
    for (j in c(1:(n_obs/4))){
      ob_i = cl_i(i, am_var, am_rat, std_dev)
      mat = rbind(mat, ob_i)
    }}
  return(mat)
}

cl_i = function(clus_i, am_var, am_rat, std_dev){
  # Given the cluster of an observation, the variables are simulated
  am_relev = (6+am_var/8)/2
  am_neutral = am_var-2*am_relev
  if (clus_i==1){
  row_v= c(dist_emo("pos", am_relev, am_rat, std_dev), dist_emo("neg", am_relev, am_rat, std_dev), dist_emo("neu", am_n
  }
  if (clus_i==2){
    row_v= c(dist_emo("neg", am_relev, am_rat, std_dev), dist_emo("pos", am_relev, am_rat, std_dev), dist_emo("neu", am_
  }
  if (clus_i==3){
    row_v= c(dist_emo("pos", am_relev, am_rat, std_dev), dist_emo("pos", am_relev, am_rat, std_dev), dist_emo("neu", am_
  }
  if (clus_i==4){
    row_v= c(dist_emo("neg", am_relev, am_rat, std_dev), dist_emo("neg", am_relev, am_rat, std_dev), dist_emo("neu", am_
  }

  val = cbind(t(c(row_v)), clus_i)
  return(val)
}

er <- function(x, n = 1) {
  ## makes sure the simulations are different for the different functions
  if (n == 0) x else c(tail(x, -n), head(x, n))
}



write_file = function(am_sim, am_var, am_rat, std_dev){
  ### given the settings, the files are generated to this path
  path = paste("C:\\Users\\siets\\Documents\\E&OR\\Thesis\\Proposal\\Simulatie_case_2",am_var,am_rat,std_dev,sep="_")
  dir.create(path)
  for (i in c(1:am_sim))
  {
    sample = sim_mat(am_var, am_rat, std_dev)
    # print(dim(sample))
    # colMeans(sample)
    # colSds(data.matrix(sample))
    file = paste(i, "sim.csv", sep="_")
    pa_file= paste(path, file, sep="\\")
    write.csv(sample, file = pa_file, row.names = FALSE)
  }
  return(print("done"))
}


Sim_case_2 = function(am_sim){
  ### simulates all datasets with the different settings, am_sim times
  for (i in c(3,5,9)){
    for (j in c(16, 32, 64)){
      for (std in c("low", "high")){
        set.seed("1234")
        sample = write_file(am_sim, am_var=j, am_rat=i, std_dev=std)
      }
    }
  }
  return(print("case_2_done"))
}

Tan_Res = function(am_sim){
  ## Tandem clustering
  library("factoextra")
  library("gplots")
  library("FactoMineR")
  library("ggfortify")
  library("pdfCluster")
  library("cluster")
  library(ncpen)

  dim_DR = 3
  ARI_PCA = function(am_sim, path, dim_DR){
    # Returns average ARI and ASW using tandem PCA in 50 simulations
    ARIcol = data.frame()
    SILcol = data.frame()
    for (i in c(1:am_sim)){
      num = i
      name = paste(num,"sim.csv", sep="_")
      pa_na = paste(path, name, sep="\\")
      matrix = read.csv(pa_na)
      dt <- as.table(as.matrix(matrix[,1:(dim(matrix)[2]-1)]))
      res.pca <- prcomp(dt, scale = FALSE)
      df = res.pca$x[,1:dim_DR]
      k2 <- kmeans(df, centers = 4, nstart = 25)
      ARI = adj.rand.index(c(k2$cluster), c(matrix[,dim(matrix)[2]]))
      sil = silhouette(c(k2$cluster), dist(dt))
      meansil = mean(sil[,3])
      ARIcol = rbind(ARIcol, ARI)
      SILcol = rbind(SILcol, meansil)
    }
    ARICIL = cbind(ARIcol, SILcol)
    return(ARICIL)
  }

  ARI_CA = function(am_sim, path, dim_DR){
    # Returns average ARI and ASW for Tandem CA in 50 simulations
    ARIcol = data.frame()
    SILcol = data.frame()
    for (i in c(1:am_sim)){
      num = i
      name = paste(num,"sim.csv", sep="_")
      pa_na = paste(path, name, sep="\\")
      matrix = read.csv(pa_na)
      dt <- as.table(as.matrix(matrix[,1:(dim(matrix)[2]-1)]))
      doubled = cbind(dt, max(dt)+1-dt)
      res.ca  <- CA(doubled, ncp=dim_DR, graph = FALSE)
      df = res.ca$row$coord
      k2 <- kmeans(df, centers = 4, nstart = 25)
      ARI = adj.rand.index(c(k2$cluster), c(matrix[,dim(matrix)[2]]))
      sil = silhouette(c(k2$cluster), dist(dt))
      meansil = mean(sil[,3])
      ARIcol = rbind(ARIcol, ARI)
      SILcol = rbind(SILcol, meansil)
    }
    ARICIL = cbind(ARIcol, SILcol)
    return(ARICIL)
  }


  ARI_MCA = function(am_sim, path, dim_DR){
    # Returns average ARI and ASW for Tandem MCA in 50 simulations
    ARIcol = data.frame()
    SILcol = data.frame()
    for (i in c(1:am_sim)){
      num = i
      name = paste(num,"sim.csv", sep="_")
      pa_na = paste(path, name, sep="\\")
      matrix = read.csv(pa_na)
      dt <- as.table(as.matrix(matrix[,1:(dim(matrix)[2]-1)]))
      # dt = data.frame(factor(dt[,1]), factor(dt[,2]), factor(dt[,3]), factor(dt[,4]))
      mat = c(1:(dim(matrix)[1]))
      for (i in c(1:(dim(matrix)[2]-1))){
        column = data.frame(factor(dt[,i]))
        mat = cbind(mat, column)
      }
      res.mca  <- MCA(mat[,2:dim(matrix)[2]], ncp=dim_DR, graph = FALSE)
      df = res.mca$ind$coord
      k2 <- kmeans(df, centers = 4, nstart = 25)
      ARI = adj.rand.index(c(k2$cluster), c(matrix[,dim(matrix)[2]]))
      sil = silhouette(c(k2$cluster), dist(dt))
      meansil = mean(sil[,3])
      ARIcol = rbind(ARIcol, ARI)
      SILcol = rbind(SILcol, meansil)
    }
    ARICIL = cbind(ARIcol, SILcol)
    return(ARICIL)
  }

  ARI_FDC = function(am_sim, path, dim_DR){
    # Returns average ARI and ASW for full dimensional clustering in 50 simulations
    ARIcol = data.frame()
    SILcol = data.frame()
    for (i in c(1:am_sim)){
      num = i
      name = paste(num,"sim.csv", sep="_")
      pa_na = paste(path, name, sep="\\")
      matrix = read.csv(pa_na)
      dt <- as.matrix(matrix[,1:(dim(matrix)[2]-1)])
      k2 <- kmeans(dt, centers = 4, nstart = 25)
      sil = silhouette(c(k2$cluster), dist(dt))
      meansil = mean(sil[,3])
      ARI = adj.rand.index(c(k2$cluster), c(matrix[,dim(matrix)[2]]))
      ARIcol = rbind(ARIcol, ARI)
      SILcol = rbind(SILcol, meansil)
    }
    ARICIL = cbind(ARIcol, SILcol)
    return(ARICIL)
  }

  tandem =function(am_sim, path, dim_DR){
    #Run all tandem functions and full dimensional clustering and obtain average for all functions
    ARIcol_PCA = ARI_PCA(am_sim, path, dim_DR)
    ARIcol_CA = ARI_CA(am_sim, path, dim_DR)
    ARIcol_MCA = ARI_MCA(am_sim, path, dim_DR)
    ARIcol_FDC = ARI_FDC(am_sim, path, dim_DR)
    return(c(colMeans(ARIcol_PCA), colMeans(ARIcol_CA), colMeans(ARIcol_MCA), colMeans(ARIcol_FDC)))
  }
  ## Execute all tandem methods and full dimensional clustering on all the simulated data
  Tandem_res = matrix( nrow =0 , ncol =12)
  colnames(Tandem_res)=c("case", "am_var", "am_rat", "Std_dev", "PCA_ARI", "PCA_SIL", "CA_ARI", "CA_SIL", "MCA_ARI", "MC
  for (c in c(1,2)){
    for (i in c(16, 32, 64)){
      for (j in c(3,5,9)){
        for (std in c("low", "high")){
        set.seed("1234")
        path = paste("C:\\Users\\siets\\Documents\\E&OR\\Thesis\\Proposal\\Simulatie_case",c,i,j,std,sep="_")
        if (std=="low"){
          std_dev=1
        }else{
          std_dev=2
        }
        res = cbind(c, i, j, std_dev, t(as.numeric(tandem(am_sim, path, dim_DR))))
        Tandem_res = rbind(Tandem_res, res)
        }
      }
    }
  }
  name =paste("Tandem_Res")
  write.csv(Tandem_res, file = name, row.names = FALSE)
  return(Tandem_res)
}

library("fastDummies")
library("philentropy")
library(stats)
library("factoextra")
library("gplots")
library("FactoMineR")
library("ggfortify")
library("pdfCluster")
library("clustrd")
library("tictoc")
library("cluster")
library(doParallel)
library(plyr)



ARI_RKM = function(am_sim, path, dim_DR){
  # Returns average ARI and ASW for RKM in 50 simulations
  ARIcol = data.frame()
  SILcol = data.frame()
  for (i in c(1:am_sim)){
    num = i
    name = paste(num,"sim.csv", sep="_")
    pa_na = paste(path, name, sep="\\")
    matrix = read.csv(pa_na)
    dt <- as.table(as.matrix(matrix[,1:(dim(matrix)[2]-1)]))
    outRKM = cluspca(dt, 4, dim_DR, method = "RKM", nstart = 25)
    sil = silhouette(c(outRKM$cluster), dist(dt))
    meansil = mean(c(sil[,3]))
    ARI = adj.rand.index(c(outRKM$cluster), c(matrix[,dim(matrix)[2]]))
    ARIcol = rbind(ARIcol, ARI)
    SILcol = rbind(SILcol, meansil)
  }
  ARICIL = cbind(ARIcol, SILcol)
  return(ARICIL)
}

ARI_FKM = function(am_sim, path, dim_DR){
  # Returns average ARI and ASW for FKM in 50 simulations
  ARIcol = data.frame()
  SILcol = data.frame()
  for (i in c(1:am_sim)){
    num = i
    name = paste(num,"sim.csv", sep="_")
    pa_na = paste(path, name, sep="\\")
    matrix = read.csv(pa_na)
    dt <- as.table(as.matrix(matrix[,1:(dim(matrix)[2]-1)]))
    outFKM = cluspca(dt, 4, dim_DR, method = "FKM",rotation = "varimax",  nstart = 25)
    sil = silhouette(c(outFKM$cluster), dist(dt))
    meansil = mean(c(sil[,3]))
    ARI = adj.rand.index(c(outFKM$cluster), c(matrix[,dim(matrix)[2]]))
    ARIcol = rbind(ARIcol, ARI)
    SILcol = rbind(SILcol, meansil)
  }
  ARICIL = cbind(ARIcol, SILcol)
  return(ARICIL)
}

ARI_MCA_kM = function(am_sim, path, dim_DR){
  # Returns average ARI and ASW for MCA_KM in 50 simulations
  ARIcol = data.frame()
  SILcol = data.frame()

  for (i in c(1:am_sim)){
    num = i
    name = paste(num,"sim.csv", sep="_")
    pa_na = paste(path, name, sep="\\")
    matrix = read.csv(pa_na)
    dt <- as.table(as.matrix(matrix[,1:(dim(matrix)[2]-1)]))
    mat = c(1:(dim(matrix)[1]))
    # Makes an indicator matrix of the numerical matrix
    for (i in c(1:(dim(matrix)[2]-1))){
      column = data.frame(factor(dt[,i]))
      mat = cbind(mat, column)
    }
    out_MCA_kM = clusmca(mat[,2:dim(matrix)[2]], 4, dim_DR, method = "MCAk", nstart = 25, seed = 1234)
    sil = silhouette(c(out_MCA_kM$cluster), dist(dt))
    meansil = mean(c(sil[,3]))
    ARI = adj.rand.index(c(out_MCA_kM$cluster), c(matrix[,dim(matrix)[2]]))
    ARIcol = rbind(ARIcol, ARI)
    SILcol = rbind(SILcol, meansil)
  }
  ARICIL = cbind(ARIcol, SILcol)
  return(ARICIL)
}

ARI_CCA = function(am_sim, path, dim_DR){
  # Returns average ARI and ASW for MCA_KM in 50 simulations
  ARIcol = data.frame()
  SILcol = data.frame()
  for (i in c(1:am_sim)){
    num = i
    name = paste(num,"sim.csv", sep="_")
    pa_na = paste(path, name, sep="\\")
    matrix = read.csv(pa_na)
    dt <- as.table(as.matrix(matrix[,1:(dim(matrix)[2]-1)]))
    doubled = cbind(dt, max(dt)+1-dt)
    CCAZK = CCA(doubled, dim_DR,25)
    sil = silhouette(c(CCAZK$alloc%*%c(1:4)), dist(dt))
    meansil = mean(c(sil[,3]))
    ARI = adj.rand.index(c(CCAZK$alloc%*%c(1:4)), c(matrix[,dim(matrix)[2]]))
    ARIcol = rbind(ARIcol, ARI)
    SILcol = rbind(SILcol, meansil)
  }
  ARICIL = cbind(ARIcol, SILcol)
  return(ARICIL)
}

CCA = function(doubled, dim_DR, ran_starts){
  # Code for own version of CCA
  minloss= Inf
  # Ran starts indicate the amount random initialisations
  for (i in c(1:ran_starts)){
    ret = ran_start(doubled, dim_DR)
    if (ret$loss < minloss){
      minloss = ret$loss
      optimAlloc = ret$alloc
      coord = ret$coord
    }
  }
  return(list("alloc"=optimAlloc, "coord"=coord))
}
```

```
ran_start = function(doubled, dim_DR){
  # In this function, we try to obtain the best possible solution given the random start
  random= floor(runif(500,min=1, max=5))
  ZK = as.matrix(dummy_cols(random, remove_selected_columns = TRUE))
  ZKprev = as.matrix(matrix(1:2000, nrow = 500, ncol = 4))
  am_var = dim(doubled)[2]/2
  iter=0
  # In this while loop, we iterate between K-means and CA until convergence is reached
  while((!identical(ZK,ZKprev)) && iter<300 ){
    ZKprev = ZK
    iter= iter+1
    ### obtain G and Z using CA
    N = as.matrix(doubled)
    ZKN = t(ZK) %*% N
    P = ZKN
    res.ca  <- CA(P, ncp=dim_DR, graph = FALSE)
    if (am_var==16){
      colnames(N) = c(1:32)
    }
    else if(am_var==32){
      colnames(N) = c(1:64)
    }else{
      colnames(N) = c(1:128)
    }
    G = res.ca$row$coord
    M = diag(500) - 1/500
    Dc = sqrt(solve(diag(colSums(N))))
    B = res.ca$col$coord
    ### Y = M N B
    Y = M %*% N %*% B
    # execute Km-means using Y and G
    Km = kmeans(Y, centers=G, algorithm="Lloyd", iter.max=20)
    ZK = Km$cluster
    ZK = as.matrix(dummy_cols(ZK, remove_selected_columns = TRUE))
    loss = Km$tot.withinss
    if(dim(ZK)[2]<4)
    {loss=Inf}
    my_list= list("alloc" = ZK, "loss" = loss, "coord"=Y)
  }
  return(my_list)
}
```

```
Simul = function(am_sim, path, dim_DR){
  #Run all simultaneous function and obtain average for all functions
  ARIcol_RKM = ARI_RKM(am_sim, path, dim_DR)
  ARIcol_FKM = ARI_FKM(am_sim, path, dim_DR)
  ARIcol_CCA=ARI_CCA(am_sim, path, dim_DR)
  ARIcol_MCA_kM = ARI_MCA_kM(am_sim, path, dim_DR)
  return(c( colMeans(ARIcol_RKM), colMeans(ARIcol_FKM), colMeans(ARIcol_CCA), colMeans(ARIcol_MCA_kM)))
}
```

```
Simul_Res = function(am_sim){
  ## Execute all simultaneous methods on all the simulated data
  Simul_Res = matrix( nrow =0 , ncol =12)
  dim_DR=3
  colnames(Simul_Res)=c("case", "am_var", "am_rat", "Std_dev", "RKM_ARI", "RKM_SIL", "FKM_ARI", "FKM_SIL", "CCA_ARI", "C
   for (c in c(1, 2)){
    for (i in c(16, 32, 64)){
      for (j in c(3,5,9)){
        for (std in c("low", "high")){
          set.seed("1234")
          path = paste("C:\\Users\\siets\\Documents\\E&OR\\Thesis\\Proposal\\Simulatie_case",c,i,j,std,sep="_")
          dim_DR=3
          if (std=="low"){
            std_dev=1
          }else{
            std_dev=2
          }
          res = cbind(c, i, j, std_dev, t(as.numeric(Simul(am_sim, path, dim_DR))))
          Simul_Res = rbind(Simul_Res, res)
          ### Give an update of all the results
          print(Simul_Res)
        }
      }
    }
  }
  return(Simul_Res)
}

##### Main
###Run sim case 1
Sim_case_1(50)
###Run sim case 2
Sim_case_2(50)



###Tandem Analysis
Tan_Res(50)

###Simultaneous Analysis
Simul_Res(50)

##### Check RKM vs FKM
library("fastDummies")
library("philentropy")
library(stats)
library("factoextra")
library("gplots")
library("FactoMineR")
library("ggfortify")
library("pdfCluster")
library("clustrd")
library("cluster")



Tandem_res = matrix( nrow =0 , ncol =2)
# Run FKM and RKm on all the simulated data
for (c in c(1,2)){
  for (i in c(16,32,64)){
    for (j in c(3,5,9)){
      for (std in c("low", "high")){
      am_sim=50
      path = paste("C:\\Users\\siets\\Documents\\E&OR\\Thesis\\Proposal\\Simulatie_case",c,i,j,std,sep="_")
      dim_DR=1
      res = RKM_FKM(am_sim, path, dim_DR)}
      Tandem_res = rbind(Tandem_res, res)
}
}
}
Tandem_res


RKM_FKM = function(am_sim, path, dim_DR){
  ##Function used to obtain the complement residuals and the subspace residuals
  RKMcol = data.frame()
  FKMcol = data.frame()
  for (i in c(1:am_sim)){
    num = i
    name = paste(num,"sim.csv", sep="_")
    pa_na = paste(path, name, sep="\\")
    matrix = read.csv(pa_na)
    dt <- as.table(as.matrix(matrix[,1:(dim(matrix)[2]-1)]))
    outRKM = cluspca(dt, 4, dim_DR, method = "RKM", nstart = 25)
    ARIRKM = adj.rand.index(c(outRKM$cluster), c(matrix[,dim(matrix)[2]]))

    outFKM = cluspca(dt, 4, dim_DR, method = "FKM",rotation = "varimax",  nstart = 25)
    ARIFKM = adj.rand.index(c(outFKM$cluster), c(matrix[,dim(matrix)[2]]))

    #### resid XAA-UFA following FKM
    #compl res = X - XAA
    compl_res = dt - outRKM$obscoord %*%  t(outRKM$attcoord)
    #subsp_resid = XAA' - UFA'
    subsp_resid = outRKM$obscoord %*%  t(outRKM$attcoord) -  to.indicators(outRKM$cluster, exclude.base = FALSE)%*% outF
    obscoors = t(outFKM$attcoord) %*% outFKM$attcoord
    varcomp = var(c(compl_res))
    varsubsp = var(c(subsp_resid))
    RKMcol = rbind(RKMcol, varcomp)
    FKMcol = rbind(FKMcol, varsubsp)
  }
  ARICIL = list("compl_res"=mean(RKMcol), "subsp_res"=mean(FKMcol))
  return(ARICIL)}
```

```
set.seed("1234")
### Reads student satisfaction file
data = read.csv("C:\\Users\\siets\\Documents\\E&OR\\Thesis\\Rcode\\StudentSF.csv")
matrix = data[,9:22]

res.pca <- PCA(matrix, scale = TRUE)
res.pca$eig[,2]

var= res.pca$sdev^2/sum(res.pca$sdev^2)
### Plots amount of variance explained by each dimension
plot(c(1:14), res.pca$eig[,2], type = "l", xlab = "Added value of dimensions", ylab = "Percentage of variance explained"

crit = data.frame()
for (i in c(2:8)){
  outRKM = cluspca(matrix, i, 2, method = "RKM", nstart = 100)
  crit = rbind(crit, outRKM$criterion)
}
### Plots amount of error with different amount of clusters
plot(x=c(2:8),y=t(crit), type="l", xlab = "Amount of clusters", ylab = "Error from loss function", main="Amount of loss

outRKM = cluspca(matrix, 5, 2, method = "RKM", nstart = 100)
### Plots the cluster allocation using RKM
plot(outRKM$obscoord, pch = outRKM$cluster,  xlab = "Scores for dimension 1", ylab = "Scores for dimension 2",
main = "Cluster allocation for the respondents")

path = "C:\\Users\\siets\\Documents\\E&OR\\Thesis\\Results\\contrib_dim.csv"
write.csv(outRKM$attcoord, file= path)
```

References

Phipps Arabie and L. Hubert. Cluster analysis in marketing research. *Advanced methods of marketing research*, pages 160–189, 1994.

George Arimond and Abdulaziz Elfessi. A clustering method for categorical data in tourism market segmentation research. *Journal of Travel Research*, 39(4):391–397, 2001. doi: 10.1177/004728750103900405. URL `https://doi.org/10.1177/004728750103900405`.

Richard E Bellman. *Adaptive control processes: a guided tour*. Princeton university press, 1961.

Asa Ben-Hur and Isabelle Guyon. Detecting stable clusters using principal component analysis. In *Functional genomics*, pages 159–182. Springer, 2003.

Jean-Paul Benzécri. *L'Analyse des Données. Volume II. L'Analyse des Correspondances*. 1973. ISBN 2-04-015515-5.

Antonio Ciampi, Ana González Marcos, and Manuel Castejón Limas. Correspondence analysis and 2-way clustering. *SORT. 2005, Vol. 29, Núm. 1 [January-June]*, 2005.

Geert De Soete and JD Carroll. K-means clustering in a low-dimensional euclidean space, 1994.

Wayne Desarbo, Kamel Jedidi, Karel Cool, and Dan Schendel. Simultaneous multidimensional unfolding and cluster analysis: An investigation of strategic groups. *Marketing Letters*, 2(2):129–146, 1991.

Carlotta Domeniconi, Dimitris Papadopoulos, Dimitrios Gunopulos, and Sheng Ma. *Subspace Clustering of High Dimensional Data*, pages 517–521. 2004. doi: 10.1137/1.9781611972740.58. URL `https://epubs.siam.org/doi/abs/10.1137/1.9781611972740.58`.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(1):211–218, 1936. URL `https://doi.org/10.1007/BF02288367`.

J. C. Gower and D. J. Hand. *Biplots: J. C. Gower and D. J. Hand (1996) London: Chapman Hall, ISBN 0-412-71630-5, [pound sign] 32.00, pp. 277*, volume 22. 1996. URL `https://EconPapers.repec.org/RePEc:eee:csdana:v:22:y:1996:i:6:p:655-651a`.

M.J. Greenacre. *Theory and Applications of Correspondence Analysis*. Academic Press, 1984. ISBN 9780122990502. URL `https://books.google.nl/books?id=LsPaAAAAMAAJ`.

A. Hinneburg and D. Keim. Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering. In *VLDB*, 1999.

Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.

Heungsun Hwang, William Dillon, and Yoshio Takane. An extension of multiple correspondence analysis for identifying heterogeneous subgroups of respondents. *Psychometrika*, 71:161–171, 02 2006. doi: 10.1007/s11336-004-1173-x.

Alfonso Iodice D'Enza and Francesco Palumbo. Iterative factor clustering of binary data. *Comput. Stat.*, 28(2):789–807, April 2013. ISSN 0943-4062. doi: 10.1007/s00180-012-0329-x. URL https://doi.org/10.1007/s00180-012-0329-x.

Anil K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2009.09.011. URL https://www.sciencedirect.com/science/article/pii/S0167865509002323. Award winning papers from the 19th International Conference on Pattern Recognition (ICPR).

Ian T. Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016. doi: 10.1098/rsta.2015.0202. URL https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2015.0202.

Pieter Kroonenberg and Michael Greenacre. *Correspondence Analysis.* 07 2004. ISBN 9780471667193. doi: 10.1002/0471667196.ess6018.

James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

Angelos Markos, Alfonso Iodice D'Enza, and Michel van de Velden. Beyond tandem analysis: Joint dimension reduction and clustering in r. *Journal of statistical software*, 91, 05 2018. doi: 10.18637/jss.v091.i10.

Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901. doi: 10.1080/14786440109462720. URL https://doi.org/10.1080/14786440109462720.

Peter Rousseeuw. Rousseeuw, p.j.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. comput. appl. math. 20, 53-65. *Journal of Computational and Applied Mathematics*, 20:53–65, 11 1987. doi: 10.1016/0377-0427(87)90125-7.

Marieke E. Timmerman, Eva Ceulemans, Henk A. L. Kiers, and Maurizio Vichi. Factorial and reduced k-means reconsidered. *Computational Statistics Data Analysis*, 54(7): 1858–1871, July 2010. ISSN 0167-9473. doi: 10.1016/j.csda.2010.02.009.

Michel Van de Velden, Alfonso Iodice D'Enza, and Francesco Palumbo. Cluster correspondence analysis. *Psychometrika*, 82, 09 2016. doi: 10.1007/s11336-016-9514-0.

Maurizio Vichi and Henk AL Kiers. Factorial k-means analysis for two-way data. *Computational Statistics & Data Analysis*, 37(1):49–64, 2001.

Ka Yee Yeung and Walter L. Ruzzo. Principal component analysis for clustering gene expression data. *Bioinformatics*, 17(9):763–774, 2001.