Erasmus School of Economics

Master Thesis Econometrics and Management Science

Comparative Analysis of Dimension Reduction Techniques: Evaluating the Performance of ICA and t-SNE Against PCA

Author: Yme Bartels (541602)
Supervisor: drs. J Durieux
Second Assessor: dr. M van de Velden

March 2024

Abstract

This thesis aims to give multiple insights in the behavior of various dimension reduction techniques in both a unsupervised and supervised setting. More complex and (non-)parametric dimension reduction methods like ICA and t-SNE are compared with more conventional and standard techniques like PCA and MDS. The considered techniques are used for medical applications where the data is retrieved from UCI Machine Learning Repository in the supervised setting. In the unsupervised setting, clustering performances are extensively analyzed for multiple data generated processes (DGP) by conducting a simulation study and the effects of the addition of noise, the amount of clusters and extra dimensions in the reduced subspaces are estimated and analyzed for each DGP. It is concluded that complex dimension reduction techniques do outperform the classic methods for more non-linear and non-Gaussian scenario's in the simulation study even though crucial assumptions are (slightly) violated. However the conventional methods do outperform more complex techniques in more linear inputspaces. In the supervised setting the more complex models do also outperform the more standard techniques for harder classification tasks. When the classification task is considered as too simplistic almost no distinction can be observed between the model performances.

Keywords: Dimensionality reduction, Subspace Learning, Independent Component Analysis (ICA), t-Distributed Stochastic Neighbor Embedding (t-SNE)

Table of Contents

1 Introduction

Machine learning techniques are becoming more and more popular nowadays as these methods are extremely useful for data-intensive analyses. On the other hand previous research has shown that both the performance and reliability of machine learning algorithms are brought down when the amount of features - or in other words the dimensionality of the input data - becomes significantly large (Aremu et al., 2020). This phenomena was first described by Bellmann (1961) and is in existing literature known as *"the curse of dimensionality"*. This curse now often refers to the difficulty of finding latent structures in highly dimensional data (Banks & Fienberg, 2003). Solving this curse (partially) is a sophisticated and difficult problem, however promising results in multiple scientific fields are produced when using dimension reduction techniques (Ray et al., 2021). A dimension reduction technique compresses high-dimensional data into a lower-dimensional subspace. These methods are for example widely used in in proteome data sets to extract insightful information based on raw gene expressions (Liu et al., 2019). Mostly dimension reduction techniques are used for unsupervised learning purposes with the aim to find hidden clusters structures in highly dimensional data, but these techniques are also able to improve prediction performances in a supervised learning setting. Ma & Dai (2011) showed that standard dimension reduction techniques such as Principal Component Anaylsis (PCA) and PCA-like methods improve the performance of classification problems for bioinformatics data. One of the drawbacks of classical dimension reduction techniques is that they are often sensitive to different types of outliers and noise (Meinecke et al., 2004). However Archimbaud et al. (2018) showed that invariant coordinate selection (ICS) gives promising results regarding the detection of outliers in high-dimensional spaces meaning that such a method can potentially solve the issue of existing outliers in the multivariate data. However this method substantially differs from all other techniques which will be proposed later and therefore it will not be taken into consideration in this thesis. Another option to handle multivariate outliers is to modify the dimensionality reduction techniques such that they become as robust as possible to outliers as it is hard to detect these outliers in a high dimensional spaces (Kaur & Datta, 2019).

In this thesis multiple dimension reduction techniques will be investigated on their performances in both an unsupervised and a supervised setting. Principal Component Analysis (PCA) is widely used in practice as this dimension reduction technique is a standard linear and fundamental method for subspace learning and dimensionality reduction. It is also easy to compute for high-dimensional data which makes it suitable for data-intensive analysis (Hastie et al., 2009). Therefore PCA will be considered as a benchmark dimensional reduction method and it will be compared with more sophisticated dimension reduction techniques. Another relatively standard dimensionality reduction technique is Multidimensional Scaling (MDS) which is often used for visualisation purposes (Buja et al., 2012). This technique extends the scope of PCA as the subspace is determined by the choice of dissimilarity or distance measure. The PCA solution can be obtained from the MDS technique when applying the correct distance measure (Williams, 2002). Yet another related dimension reduction method to PCA is Independent Component Analysis (ICA) which compresses high-dimensional non-Gaussian data into a subspace with independent components and is extensively used for bio-medical data and reducing noise in natural images (Hyvarinen & Oja, 2000). van der Maaten & Hinton (2008) proposed a non-parametric dimension reduction technique known as t-distributed stochastic neighbor embedding

(t-SNE) which is a tool mainly used in order to visualize high-dimensional computational and cancer biology data in a two or three dimensional subspace such that both the local and global structure of the original space is preserved in the lower dimensional subspace (Tripathy & Ghela, 2022). This balance between the preservation of the global and local structure can be influenced by hyperparameters in the model. Although t-SNE is mainly used for visualisation purposes existing literature showed that this non-parametric t-SNE method can also be used for unsupervised learning problems by using this method for the clustering of hyperspectral paper data (Devassy et al., 2020). Moreover, Hajderanj et al. (2019) showed promising results for a supervised generalization of the t-SNE method in breast cancer data. Even though the more sophisticated dimension reduction methods are already developed, it is of importance to know how these method perform under different circumstances which can be easily exploit with an unsupervised study as there is total control over the data generating process (DGP). For a supervised study this control is much harder to get as specific latent data characteristics are often not known. This thesis therefore tries to answer the following research question: *"Do ICA or t-SNE outperform the standard PCA-like methods as dimension reduction technique in both a supervised and unsupervised setting?"* Although the dimensionality reduction techniques proposed above do all look different at first sight, they all do have a (mathematical) relation with each other. It is important to understand these underlying relations as the connections may explain the different behavior of the models in different situations.

To answer the unsupervised part of this research question a simulation experiment is done to investigate the performances of the considered techniques. The methods will be compared with each other based on the clustering performance of latent structures in the unique compressed subspace for different data generating processes (DGPs). Moreover, for every DGP the effects of adding gaussian noise, extra dimensions in the subspaces and clusters are analyzed per method. In the supervised learning context real-world medical data is used to compare the dimension reduction techniques with various classification performance measures. The real-world data are retrieved from University of California (UCI) Machine Learning Repository and are publicly available and are therefore ideal for researchers who want to replicate the used procedures in this study. The first data set which will be looked at is the Breast Cancer Wisconsin Diagnostic database which contains multiple features of a digitized image of a fine needle aspirate (FNA) of a breast mass (Wolberg et al., 1995). The second data set contains handwriting data of participants having potentially Alzheimer's Disease which is also obtained from the UCI database (Fontanella, 2022). All dimension reduction techniques have to be generalized or adjusted in order to be used as a classification tool, because the compressed subspaces are normally not based on a dependent variable which can potentially lead to a loss of information about this response variable (Malhi & Gao, 2004).

One of the main results of the unsupervised study is that classic dimension reduction techniques do outperform the more sophisticated models when the DGP corresponds with a linear inputspace and the columns are all normally distributed variables. However when these characteristics are replaced non-linear and non-Gaussian variables the sophisticated models begin to perform relatively better compared to the PCA-like models. As long as the added Gaussian noise is not of an extreme high order it is observed that the ICA can handle this noise significantly well given that there exists a certain amount of non-Gaussianity in the input columns which is in line with the study of Durieux &

Wilderjans (2019). The classic methods all have more difficulties with an addition of noise regardless the characteristics of the DGP. For the supervised learning problem it is observed that for a relatively simple classification task none of the reduced features significantly predict the response variable better than the original features. This is potentially caused due to the fact that the proposed classification problem can be considered as too easy problem, because the original input features already ensure a extremely high prediction power. However for harder classification tasks an outperformance of the more sophisticated reduction techniques are observed where the ICA and t-SNE both have significantly higher performance measure values compared to the original features and all other more standard dimension reduction models.

In Section 2 a structured overview of already existing literature is given. Thereafter the methodology of each proposed technique and overall analyses can be found in Section 3. In Section 4 the generation of the simulated data is showed for the unsupervised learning part of this study. Subsequently all the information about the chosen data for the supervised study is given as well as the simulated cluster structures for the unsupervised learning part of this thesis. The main results are given in Section 5 and the paper is concluded in Section 6.

2 Literature

Principal Component Analysis (PCA) is a common and standard dimension reduction technique developed by Hotelling (1933) and Pearson (1901). The main idea of PCA is that uncorrelated components are created which maximize the total variance of the data. As a result the dimension of the data can be reduced in a way such that the information in the data is preserved as much as possible (Jolliffe & Cadima, 2016). A disadvantage of most dimension reduction techniques which also holds for standard PCA is that the created uncorrelated components are not very interpretable, even though rotation of components (and loadings) towards more simple structures exist (Kaiser, 1958). For PCA the components are equivalent to a specific linear combination of the input features. The principal components are obtained by the eigenvectors of the covariance matrix of the data and can therefore be calculated with both an eigendecomposition (EVD) and a singular value decomposition (SVD). The principal component with the highest corresponding eigenvalue is called the first principal component and contains most information about the initial data (Hastie et al., 2009). In multiple scientific fields PCA is applied prior to clustering analyses, because it might be expected that the cluster structure of the complete data set will be captured in the lower-dimensional subspace created by the first couple of principal components (Yeung & Ruzzo, 2001). However previous research has shown that the clustering structure is not always contained in the first couple of principal components. Chang (1983) showed that components with corresponding large eigenvalues do not have to contain much cluster structure when the data is a mixture of two multivariate normal distributions with different means but the same covariance structure within the two clusters. Similar results were found in a study of Chu & Herskowitz (1998) where PCA is used for clustering purposes to budding yeast data. Chu showed that a combination of components corresponding to lower eigenvalues had a better performance compared to the first components even though less variation of the data was captured in the used components. For this research the obtained principal components are used for partitioning the subspace with a clustering algorithm. This order of first reducing the original space and thereafter the partitioning based

on a clustering algorithm is known as tandem clustering. Dolnicar (2003) suggests that a tandem clustering procedure can be a statistically insupportable practice, because part of the global and local structure that should be mirrored by the clustering analysis is eliminated. A solution for this problem is a technique which simultaneously estimates the components using a dimension reduction technique and cluster the objects in the reduced subspace. Both reduced k-means and factorial k-means are techniques that estimate the subspace with PCA techniques and cluster the object in a simultaneous way (Timmerman & Vichi, 2010). However, reduced k-means (RKM) showed promising and better results compared to factorial k-means when the majority of the variables reflect the clustering structure and/or the features are standardized before the analysis (Timmerman & Vichi, 2010).

Classical PCA is an unsupervised learning tool and is therefore not suitable for classification and regression problems. However it is possible to expand the standard PCA framework such that the components maximize the dependency with the dependent variable instead of the variation in the reduced space (Barshan et al., 2011). This method is called supervised principal component analysis (SPCA) and can be used for classification and regression problems. The columns which span the compressed subspace serve as features which are as dependent to the dependent variable as possible (Ghojogh & Crowley, 2022). SPCA is an efficient technique for classification and regression problems as it reduces the computational problems for high-dimensional data. Previous literature has shown that SPCA outperforms more standard alternatives regarding the accuracy in classification problems such as Bair's Supervised Principal Components and kernel dimensionality reduction while on the other hand SPCA does have difficulties to extract local structures in the high-dimensional data as shown by Barshan et al. (2011).

Another drawback of PCA is a dimension reduction technique which is sensitive to outliers. To overcome this Filzmoser et al. (2009) proposed a robust alternative for PCA which uses simple robust estimates for the mean and sample covariance matrix in order to handle potential multivariate outliers better. Reimann showed that the robust PCA gives significantly better results for geochemical data compared to standard PCA and other non-robust dimension reduction techniques. Robust PCA is both applicable to conventional PCA and SPCA, but it will increase the computational time of both methods which is undesirable. Besides multivariate outliers, RPCA is also capable of handling noisy data better compared to conventional PCA techniques, mainly due to better estimation of the covariance matrix (Zhao et al., 2014).

Multidimensional scaling (MDS) is another more classical dimensionality reduction technique which will be analyzed within this thesis. The idea of MDS is that a high dimensional space is mapped onto a lower dimensional subspace such that the pairwise spatial distances between the points in the higher dimensional space are preserved (Tripathy & Ghela, 2022). MDS can be distinguished into two groups which refer to two different approaches. This thesis will only focus on the metric MDS (classic), also known as Principal Coordinate Analysis as this approach is closely related to the other dimensionality reduction techniques (Anowar et al., 2021). This approach has many applications in various fields of psychology and it also had significant successes in ecological studies where MDS outperformed other standard dimensionality reduction techniques (Davison, 1983; Kenkel & Orloci, 1986). The other approach of MDS is nonmetric MDS and this method is based on the rankings of the distances in

the higher dimensional space. This method is still very theoretical which often result in high computational times (Kenkel & Orloci, 1986). In this thesis MDS will mainly be used in order to show the mathematical relations between the considered dimension reduction techniques in this research as multiple methods are (closely) linked to the general idea of MDS. These mathematical relations are extensively discussed in Section 3.5.

Another commonly used dimension reduction technique is independent component analysis (ICA). This technique originates from separating signals into their original sources (Westad & Kermit, 2009). Separating sound sources from recorded mixed signals is known as the cocktail-party problem and can be solved with ICA. For ICA it must be assumed that the original sources are statistically independent and non-Gaussian. These assumptions are in many cases violated, however they do not have to be precisely true (Hyvarinen & Oja, 2000). There are multiple ways to calculate the independent components but this thesis will mainly focus on estimation using Fast-ICA proposed by Hyvarinen (1999). ICA is a convenient statistical tool for dimensionality reduction but the analysis also contains some model ambiguities. It is for instance not possible to calculate the variances of the independent components due to identification problems. The most efficient way to overcome this identification problem is to assume that all independent components have unit variance (Beckmann, 2012). Another inconvenient characteristic of ICA is that the order of the independent components cannot be determined unlike other dimensionality reduction techniques such as PCA where the first component corresponds with the highest eigenvalue and captures most variability. A permutation matrix can be added in the ICA framework which will result in the same independent components but in another order. It is only possible to determine the amount of independent components before the analysis and use all the obtained components as the span of the lower-dimensional subspace (Hyvarinen & Oja, 2000). Besides the cocktail-party problem ICA has many real-world applications. Yang et al. (2005) proposed a generalization of standard ICA for face recognition purposes. ICA is also one of the most often used methods for identifying biologically relevant networks in a brain that relates to functioning (e.g. the visual system in humans), but also to dysfunctioning in patients that suffer from dementia (Calhoun & Adali, 2012)

It is possible to extend the ICA framework towards a supervised setting such that is able for classification problems. To main idea is that a binary class label is appended to the ICA framework in a way such that the created independent components (features) are divided in two sets. One set of features which do not carry much information about the binary class label and one set which in statistically does carry more information about the class label. This is done by maximizing the joint mutual information between the binary class label and the created features. This feature extraction method which is based on ICA (ICA-FX) reduces the dimension of the feature space significantly while also improving the prediction power (Kwak & Choi, 2003).

T-distributed stochastic neighbor embedding (t-SNE) is a non-linear data visualisation method proposed by van der Maaten & Hinton (2008) which gives a lower-dimensional location to every datapoint while trying to preserve both the local and global structure of the data. This method differs significantly from previous discussed dimension reduction methods as t-SNE is more flexible due to the fact that the dimension reduction itself is based on non-linear assumptions. This method is an extension

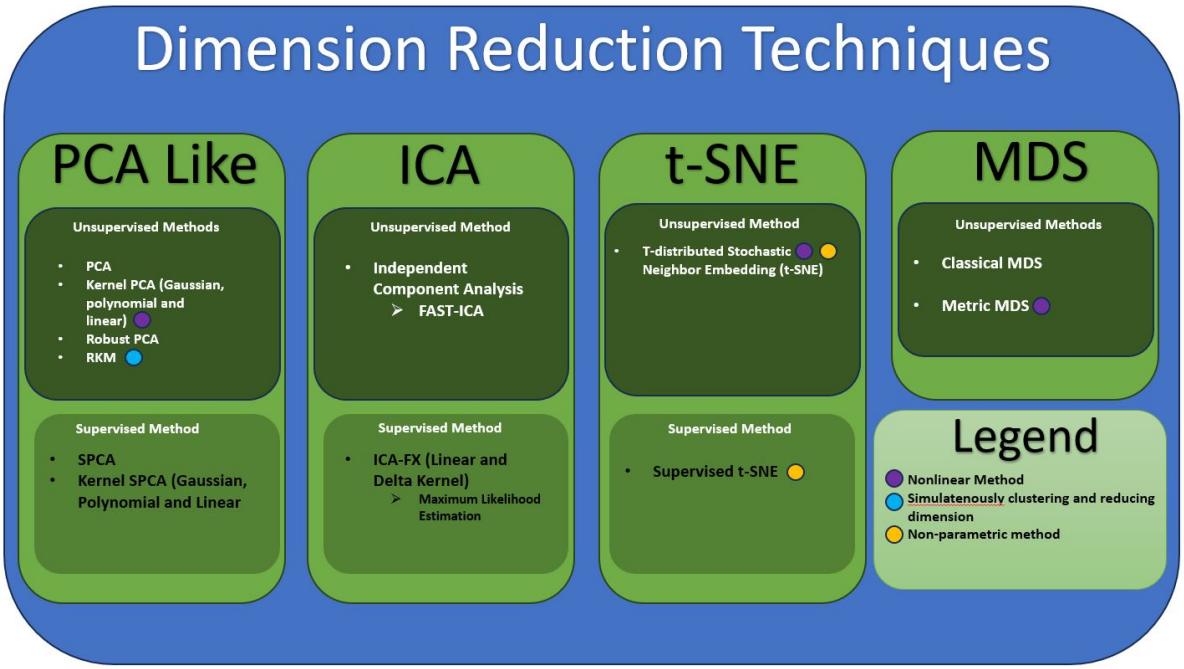
of the stochastic neighbor embedding (SNE) method which is easier to optimize and is able to partially resolve the crowding problem. The crowding problem occurs when the area of the two or three dimensional map which is available in order to accommodate datapoints with large distances is by far not large enough compared to the area of the map which is needed to accommodate datapoints with small distances. Therefore if we want to model nearby datapoints accurately the datapoints with large distances must be placed too far away on the two or three dimensional map. This crowding problem is partially solved with the t-SNE method as with t-SNE a heavier tailed student t-distribution is used to calculate the similarities between two points in the lower-dimensional subspace instead of a normal Gaussian distribution. The t-SNE method can be applied in several scientific fields and has had significant results by outperforming more standard linear dimension reduction techniques (Linderman et al., 2017). T-SNE is widely used for example in the medical field for analysing and clustering single-cell RNA sequences or single-cell transcriptomics (Kodak & Berens, 2019).

T-SNE is an unsupervised learning method which can be generalized in a way such that it becomes suitable also for supervised classification problems. Xu et al. (2020) proposed a supervised t-SNE method which is currently used for the classification of microbiome data. The general idea of this supervised t-SNE method is that the input features for the classification are a weighted average of k different datapoints in the reduced subspace which correspond with the k-nearest neighbors in the original input space. These weights are based on their relative distances to the new sample. Within the t-SNE algorithm a different distance measure must be used to avoid spurious correlations within the data. Therefore the Aitchison distance, which measures the distance between the composition of two input vectors, can be used because this measure has the property that it is invariant for scale and permutation and that it is sub-compositional coherent.

3 Methodology

In this section all the considered dimension reduction techniques will be explained and discussed separately. First the PCA technique and all techniques which are directly linked towards PCA are discussed including the generalization such that it becomes suitable for classification tasks. Subsequently, the ICA method will be introduced and the structure of this technique is mathematically analyzed. Again the adjustments in order to make it suitable for supervised learning purposes are also discussed. Thirdly, the t-SNE technique and supervised t-SNE will be discussed and thereafter the scope of PCA will be extended once more by introducing the classical and metric MDS method. At last the mathematical connections between all considered models will be discussed. Figre 1 shows a schematic overview of the considered dimension reduction methods used in this thesis. The color codes indicate if a specific method is (1) non-linear, (2)non-parametric or (3) a simultaneous dimension reduction clustering technique.

Figure 1: Schematic Overview Considered Dimension Reduction Techniques



3.1 Principal Component Analysis (PCA) and Related Variants

Principal Component Analysis (PCA) is a linear method which projects d-dimensional data $\mathbf{X}$ onto a lower p-dimensional subspace, assuming that $p \leq d$ (and often $p \ll d$) (Hastie et al., 2009). For the derivation it is assumed that the columns of $\mathbf{X}$ are centered which means that the corresponding mean is subtracted from each value. If the data is not centered, this must be done before any PCA analysis. A linear projection of the d-dimensional data can be written as $\hat{\mathbf{X}} = \mathbf{U}\boldsymbol{\beta}$ where $\mathbf{U}$ is of dimension p. All mathematical notation is in line with Hastie et al. (2009) and Filzmoser et al. (2009). The distance between the d-dimensional $\mathbf{X}$ and the linear projection $\mathbf{U}\boldsymbol{\beta}$ is preferably as small as possible for an optimal projection. This residual is minimized when the residual vector $(\mathbf{X} - \mathbf{U}\boldsymbol{\beta})$ is orthogonal to the p-dimensional subspace. Therefore it must hold that $\mathbf{X} - \mathbf{U}\boldsymbol{\beta} \perp \mathbf{U}$ which is equivalent to equation 1:

$$\mathbf{X}^T(\mathbf{X} - \mathbf{U}\boldsymbol{\beta}) = 0 \iff \boldsymbol{\beta} = (\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{X} \tag{1}$$

This least-squares expression for $\boldsymbol{\beta}$ can be plugged into the first equation above such that the fitted $\mathbf{X}$ equals $\hat{\mathbf{X}} = \mathbf{U}(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{X}$. This expression can be simplified because it may be assumed that the columns which span the p-dimensional subspace $\mathbf{U}$ are orthonormal [1]. Therefore the equation is similar to $\hat{\mathbf{X}} = \mathbf{U}\mathbf{U}^T\mathbf{X}$.

One way of calculating the principal components which will be the column span of $\mathbf{U}$ that maximises the variance is by means of an EVD. The main idea is to maximize the variance of the projected data onto the PCA subspace. In mathematical terms this is equivalent to maximizing $Var(\mathbf{u}^T\mathbf{X})$ for every component $\mathbf{u}$. This variance expression can be rewritten as $\mathbf{u}^T Var(\mathbf{X})\mathbf{u} = \mathbf{u}^T\mathbf{S}\mathbf{u}$, where $\mathbf{S}$ is the sample covariance matrix of $\mathbf{X}$ which can be calculated using $\mathbf{S} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^T$. The

[1] When the columns of a matrix are orthonormal it means that the matrix $\mathbf{U}$ can be considered as orthogonal and that $\mathbf{U}^T\mathbf{U} = \mathbf{I}$

solution of the principal components $\mathbf{u}$ can be calculated by maximizing the constrained optimization problem in equation 2:

$$\begin{aligned} \max_{\mathbf{u}} \quad & \mathbf{u}^T\mathbf{S}\mathbf{u} \\ \text{s.t.} \quad & \mathbf{u}^T\mathbf{u} = 1 \end{aligned} \tag{2}$$

This constrained optimization problem can be solved using Lagrangian optimization techniques. The Lagrangian for this problem equals: $\mathcal{L} = \mathbf{u}^T\mathbf{S}\mathbf{u} - \lambda(\mathbf{u}^T\mathbf{u} - 1)$. Taking the derivative of the Lagrangian with respect to $\mathbf{u}$ and set it equal to zero gives the final result:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{u}} = 2\mathbf{S}\mathbf{u} - 2\lambda\mathbf{u} = 0 \iff \mathbf{S}\mathbf{u} = \lambda\mathbf{u} \tag{3}$$

Here λ and $\mathbf{u}$ are equivalent to the eigenvalues and eigenvectors of the covariance matrix respectively. The eigenvector with the corresponding highest eigenvalue contains the most variation of the sample covariance matrix and is called the first principal component. To reduce the dimensionality of the d-dimensional data, the first p principal components can be used as columns of the matrix $\mathbf{U}$.

3.1.1 Unsupervised Kernel Principal Component Analysis (KPCA)

One of the main limitation of standard PCA is that it is a linear method. As a result PCA is not really suitable in a situation where the existing data points are related in a non-linear way (Hastie et al., 2009). Kernel PCA (KPCA) expands the PCA framework such that it becomes suitable for non-linear data by first increasing the feature space in terms of dimensionality with the hope that the data will be linear in this higher dimensional space (Ghojogh & Crowley, 2022). How the dimensionality of the data increases relies on the chosen kernel function. In this paper three different kernel functions are taken into consideration: (1) A linear kernel, (2) a polynomial kernel and (3) a Gaussian kernel. The function $\Phi(\mathbf{X})$ maps the features of the data $\mathbf{X}$ to a higher-dimensional space based on the chosen kernel function. This kernel function $\mathbf{K}_x$ can be written as equation 4.

$$\mathbf{K}_x = \mathbf{K}(\mathbf{X},\mathbf{X}) = \Phi(\mathbf{X})^T\Phi(\mathbf{X}) \tag{4}$$

The mathematical notation of the three kernel functions taken into consideration are given in the Appendix 7.3. The idea of KPCA is to apply a matrix decomposition on K($\mathbf{X}$,$\mathbf{X}$). The matrix of eigenvectors $\mathbf{V}$ and the diagonal matrix of the square roots of the eigenvalues $\mathbf{\Sigma}$ can be used to calculate the higher dimensional space created by the function Φ projected on the PCA space. In mathematical terms this is equivalent to equation 5. This calculation is based on a singular value decomposition (SVD) of Φ ($\mathbf{X}$) and is proposed by Hofmann et al. (2008). Appendix 7.5 shows the mathematical relation between a SVD and EVD decomposition.

$$\Phi(\mathbf{X}) = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{5}$$

Where $\mathbf{U}$ and $\mathbf{V}$ refer to the p leading left and right singular vectors of $\Phi(\mathbf{X})$, respectively. $\mathbf{\Sigma}$ is a square matrix of dimension p where the diagonal elements correspond with the p largest non-zero singular values of $\Phi(\mathbf{X})$. A disadvantage of KPCA is that the results are heavily influenced by the decision of the kernel function and it is not clear which kernel gives the best results in general (Hastie et al., 2009). Therefore the researcher must take multiple kernels into consideration and do the KPCA analysis multiple times in order to compare the different outputs.

3.1.2 Supervised Principal Component Analysis (SPCA)

Standard (K)PCA is an unsupervised method and can therefore not be used for classification and regression problems. Barshan et al. (2011) proposed a technique which generalizes the PCA in a way such that the principal components have maximum dependency with the response variable instead of maximum variability. This method is called supervised principal component analysis (SPCA). The dependency between two random variables can be measured with the Hilbert-Schmidt independence criterion (HSIC) which is proposed by Gretton et al. (2005). The criterion makes use of the idea that two random variables $\mathcal{X}$ and $\mathcal{Y}$ are independent if and only if any bounded function of the two random variables is uncorrelated. The mathematical expression for the empirical HSIC is given in equation 6.

$$\mathbf{HSIC} = \frac{1}{(n-1)^2}\mathbf{tr}(\mathbf{KHLH}) \tag{6}$$

Here n refers to the sample size of the training data, $\mathbf{K}$ is a kernel of the data $\mathbf{X}$ such that $\mathbf{K} = \mathbf{K}(\mathbf{X},\mathbf{X})$. $\mathbf{L}$ is a kernel of the response variable $\mathbf{Y}$ such that $\mathbf{L} = \mathbf{L}(\mathbf{Y},\mathbf{Y})$ and $\mathbf{H}$ is a centering matrix which is defined by $\mathbf{H} = \mathbf{I}$ - n^{-1} $\mathbf{e}^T\mathbf{e}$ where $\mathbf{e}$ is a vector of ones.

To come back to the original problem, again a subspace $\mathbf{U}^T\mathbf{X}$ must be found. However, a subspace $\mathbf{U}^T\mathbf{X}$ should be created which maximises the dependency between this subspace and the response variable $\mathbf{Y}$. The HSIC is used in order to maximize this dependency. Note that in the HSIC expression $\mathbf{U}^T\mathbf{X}$ can be used to calculate $\mathbf{K}$ and $\mathbf{L}$. In this research two different kernels for $\mathbf{Y}$ will be taken into consideration, namely a linear kernel $\mathbf{L} = \mathbf{Y}^T\mathbf{Y}$ and a delta kernel proposed by Ghojogh & Crowley (2022). To come to the final result it is necessary to rewrite the trace expression within the HSIC in a way such that a closed form solution can be obtained. The original expression can be rewritten as follows: $\mathbf{tr(KHLH)} = \mathbf{tr(HKHL)} = \mathbf{tr(HX}^T\mathbf{UU}^T\mathbf{XHL)} = \mathbf{tr(U}^T\mathbf{XHLHX}^T\mathbf{U)}$. The final constrained optimization problem is given in equation 7 and has a closed form solution which equals $\mathbf{Q} = \mathbf{XHLHX}^T$.

$$\begin{aligned} \underset{u}{argmax} \quad & \mathbf{tr}(\mathbf{U}^T\ \mathbf{XHLHX}^T\mathbf{U}) \\ \text{s.t.} \quad & \mathbf{U}^T\mathbf{U} = \mathbf{I} \end{aligned} \tag{7}$$

The eigenvectors of $\mathbf{Q}$ are the principal components maximizing the dependency with the response variable and can be used as features in classification and regression problems. The algorithm of SPCA can be summarized in algorithm 1.

Algorithm 1: Superviced Principal Component Analysis

Require: Training data $\mathbf{X}$, test data $\mathbf{X}^*$ and output $\mathbf{Y}$

1: $\mathbf{H} \leftarrow \mathbf{I} - \frac{1}{n}\mathbf{e}^T\mathbf{e}$
2: $\mathbf{Q} \leftarrow \mathbf{XHLHX}^T$
3: $\mathbf{U} \leftarrow$ Eigenvectors of $\mathbf{Q}$ corresponding to the top d eigenvalues
4: **Encode:** Trainingdata $\mathbf{Z} \leftarrow \mathbf{U}^T\mathbf{X}$
5: **Encode:** Testdata $\mathbf{Z}^* \leftarrow \mathbf{U}^T\mathbf{X}^*$

A small extension of SPCA is that instead of the data $\mathbf{X}$ a kernel of this data $\mathbf{K}_x$ is used. This

extension is similar to KPCA with respect to standard PCA for unsupervised learning problems as discussed in Section 3.1.1. For this extension the same kernel functions are taken into consideration as before and they can be found in the Appendix 7.3. These kernels ensure that SPCA is suitable for non-linear relations between the response variable and the original data. The procedure for kernel SPCA is summarised in algorithm 2 and the complete derivation of this method can be found in Barshan et al. (2011).

Algorithm 2: Kernel Superviced Principal Component Analysis

Require: Training kernel data $\mathbf{K}$, kernel test data $\mathbf{K}^*$ and output $\mathbf{Y}$

1: $\mathbf{H} \leftarrow \mathbf{I}_n - \frac{1}{n}\mathbf{e}^T\mathbf{e}$
2: $\mathbf{Q} \leftarrow \mathbf{KHLHK}^T$
3: **Compute Basis:** $\boldsymbol{\beta} \leftarrow$ generalized eigenvectors of **(Q,K)** corresponding to the top d eigenvalues.
4: **Encode:** Trainingdata $\mathbf{Z} \leftarrow \boldsymbol{\beta}^T\ \mathbf{K}$
5: **Encode:** Testdata $\mathbf{Z}^* \leftarrow \boldsymbol{\beta}^T\ \mathbf{K}^*$

Here $\mathbf{e}$ is a one vector of length n and $\mathbf{I}_n$ refers to an identity matrix. Kernel SPCA has a disadvantage namely that the results are again highly influenced by the chosen kernel. Therefore the researcher must take multiple kernels into consideration and compare the results. It is noteworthy to mention that SPCA and PCA are closely related. When we take an identity kernel for $\mathbf{Y}$ such that $\mathbf{L} = \mathbf{I}_n$ the results of SPCA and PCA are equivalent as now the dependency between the projection $\mathbf{U}^T\mathbf{X}$ and $\mathbf{I}_n$ is maximised. Or in other words, the variation of the projection is maximised which gives us a standard PCA solution (Barshan et al., 2011).

3.1.3 Robust Principal Component Analysis

For standard PCA the components are calculated with the sample covariance matrix. However, this sample covariance matrix is sensitive to outliers which can cause undesirable results for data with a potential amount of outliers or noise. The eigenvector which are based on this sample covariance matrix can therefore become inconsistent (Filzmoser et al., 2009). A solution for this is to calculate the location T($\mathbf{X}$) and scale estimator S($\mathbf{X}$) with robust alternatives. The principal component matrix are represented in equation 8. Here $\mathbf{V}$ represent the matrix of eigenvectors of scale estimator S($\mathbf{X}$).

$$\mathbf{X}^* = \mathbf{V}^T(\mathbf{X} - \mathbf{e}T(\mathbf{X})) \tag{8}$$

Here $\mathbf{e}$ is a vector of n ones. Filzmoser et al. (2009) proposed a minimum covariance determinant (MCD) technique in order to estimate the location and scale estimator in a robust way. The main idea of MCD is to take a subset h data points and calculate the sample covariance and mean for that particular subset, note that $h \leq n$. This subset is chosen in a way such that the determinant of a sample covariance matrix, based on h datapoints is minimised. As a result (large) outliers will not be contained in the subset. To take account for arisen inconsistencies a Fisher consistency correction is needed for the scale parameter S($\mathbf{X}$). For MCD estimation h is a hyperparameter which must be chosen before the analysis. Lopuhaa & Rousseeuw (1991) showed that the optimal value of h equals $\frac{n+p+1}{2}$ which maximizes the breakdown point for this estimator.

3.1.4 Reduced k-means (RKM)

Reduced k-means (RKM) is a technique which simultaneously reduces the original feature space and cluster the object with a k-means clustering algorithm. This is an alternative procedure compared to a tandem clustering method where first the data is reduced to a lower-dimensional space and subsequently a clustering algorithm is used in order to cluster the different objects in the reduced space. The main idea is that the data $\mathbf{X}$ is decomposed into three different matrices: (1) the loading matrix $\mathbf{A}$, (2) the membership matrix $\mathbf{U}$ and (3) the centroid matrix $\mathbf{F}$. The centroid matrix is a matrix where the i'th column corresponds with the mean of cluster i. The membership matrix is a binary matrix where $\mathbf{U}_{ij} = 1$ if object i is clustered in cluster j. The loading matrix shows which variables reflect and which variables do not reflect the clustering structure. All mathematical notation is in line with Terada (2014). Factorial k-means (FKM) is a similar method which simultaneously reduces the feature space and groups object using a k-means algorithm which is proposed by Vichi & Kiers (2001). However this technique is not discussed in this thesis as RKM shows better results when the majority of the variables reflect the clustering structure (Dolnicar, 2003). As later will be discussed in Section 4, it is expected that RKM is preferred for simulated cluster structures. RKM can be seen as a minimization problem with the objective function given in equation 9.

$$RKM(\mathbf{A}, \mathbf{F}, \mathbf{U} \mid k, p) = \|\mathbf{X} - \mathbf{U}\mathbf{F}\mathbf{A}^{\mathbf{T}}\|_F^2 \tag{9}$$

Here k refers to the amount of clusters, p corresponds with the amount of components taken into consideration and $\|.\|_F$ denotes a Frobenius norm. All other variables are defined earlier in this subsection. This RKM loss function finds a solution which minimizes the distances between the original data points and clustercentroids in the reduced space spanned by the columns of the loading matrix $\mathbf{A}$ (Timmerman & Vichi, 2010). The algorithm of the RKM procedure is given in Algorithm 3.

Algorithm 3: Reduced K-means (RKM)

1: **Initialize:** The values of $\mathbf{U}$, $\mathbf{F}$ and $\mathbf{A}$
2: **Calculate:** $\mathbf{Q}\mathbf{\Sigma}\mathbf{V}^{\mathbf{T}}$ which is the SVD of $(\mathbf{U}\mathbf{F})^{\mathbf{T}}\mathbf{X}$. See 7.5 for the relation between SVD and EVD.
3: **Update:** The loadingsmatrix $\mathbf{A} = \mathbf{V}\mathbf{Q}^{\mathbf{T}}$
4: **Update:** For $i = 1, ..., n$ and $j = 1, ..., k$ update $u_{ij} = 1$ if $\|\mathbf{A}^{\mathbf{T}}x_i - f_j\|^2 < \|\mathbf{A}^{\mathbf{T}}x_i - f_{j'}\|$ for each $j' \neq j$ and zero otherwise.
5: **Update:** the centroidmatrix $\mathbf{F} = (\mathbf{U}^{\mathbf{T}}\mathbf{U})^{-\mathbf{1}}\mathbf{U}^{\mathbf{T}}\mathbf{X}\mathbf{A}$
6: **Repeat:** from step 2 until convergence.

Note that in step 4 a normal euclidean norm is used instead of the Frobenius norm. It is important to know that this algorithm potentially converges to a local minimum. Therefore it is necessary to run this algorithm multiple times with different starting values in order to increase the chance of finding the global minimum.

3.2 Independent Component Analysis (ICA)

Independent Component Analysis (ICA) is a statistical tool proposed by Comon (1994) which decomposes observable mixtures x_j into statistical independent components. In random vector notation, this can be written as equation 10. It is assumed that all vectors $\mathbf{x}$ and $\mathbf{s}$ have zero mean and if this assumption does not hold all $\mathbf{x}$ can be centered without loss of generality.

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{10}$$

In this equation both the matrix $\mathbf{A}$ as the vector $\mathbf{s}$ are unknown and have to estimated under the most general assumptions. It is assumed that the components $\mathbf{s}$ are statistically independent and that they must be non-Gaussian. First for simplicity it is also assumed that the matrix $\mathbf{A}$ is a square matrix [2]. Since $\mathbf{A}$ is a square matrix a matrix $\mathbf{W}$ can be defined such that the equation 11 holds. Note that $\mathbf{W}$ must equal $\mathbf{A}^{-1}$. All mathematical notation is in line with the paper of Hyvarinen & Oja (2000).

$$\mathbf{s} = \mathbf{W}\mathbf{x} \tag{11}$$

The vector $\mathbf{s}$ has to be constructed in a way such that the components are statistically independent as is assumed in the beginning of this Section. Two random variables, y_1 and y_2, are independent when any information of y_1 does not give information about the other random variable y_2 and vice versa. Note that independent components is a much stricter assumption than uncorrelated components which was assumed in the PCA analysis. Two components are uncorrelated when there does not exist a linear dependence between the two components. However for Gaussian variables uncorrelation implicates also independence. In mathematical terms independence between two random variables can be written as equation 12.

$$p(y_1, y_2) = p_1(y_1)p_2(y_2) \tag{12}$$

This definition of independence can be extended to any higher dimension of amount or variables.

Another assumption of the ICA framework is that the independent components have to be non-Gaussian distributed. This is because it is not possible to estimate the weighting matrix $\mathbf{A}$ when all components are completely symmetric. This will lead to multiple identification problems. It is possible to extend the ICA framework such that is becomes possible to estimate the matrix $\mathbf{A}$ when only some components are normally distributed. The main idea of ICA comes from the Central Limit Theorem which state that the sum of independent random variables tends towards a Gaussian distribution. Therefore it is correct to assume that the sum of two independent random variables are closer to a Gaussian distribution compared to any of the original random variables. Now a new variable $\mathbf{z}$ is defined such that $\mathbf{z} = \mathbf{A}^T\mathbf{w}$. Then it is possible to rewrite part of the equation which is done in equation 13.

$$\mathbf{y} = \mathbf{w}^\mathbf{T}\mathbf{x} \iff \mathbf{y} = \mathbf{w}^\mathbf{T}\mathbf{A}\mathbf{s} \iff \mathbf{y} = \mathbf{z}^\mathbf{T}\mathbf{s} \tag{13}$$

From this equation it is clearly visible that y is a linear combination of $\mathbf{s}$ which is determined by a weight vector $\mathbf{z}$. Since the sum of two independent random variables are more Gaussian than the two original variables, it must hold that $\mathbf{z}^T\mathbf{s}$ is more Gaussian than any of the independent components

[2] This assumption can be relaxed in a later stage of the ICA analysis

$\mathbf{s}_i$. Moreover $\mathbf{z}^T\mathbf{s}$ becomes least Gaussian when it exactly equals one of the $\mathbf{s}_i$. Therefore the vector $\mathbf{w}$ has to be estimated in a way such that it maximizes the non-Gaussianity of $\mathbf{w}^T\mathbf{x}$. As a result when this non-Gaussianity is maximized, $\mathbf{w}^T\mathbf{x}$ will equal an independent component. Now the question raises how to measure the non-Gaussianity of random variables. This will be discussed in the next subsection.

Non-Gaussianity Measures

There are multiple ways to measure the non-Gaussianity of a random variable. For the rest of this part standard normality is assumed, which means that the random variable has mean zero and a variance of one. This assumption holds for the whole ICA analysis as well and standard transformations are needed in case the assumption is not valid. One of the classical measures of a (non-)Gaussian variable x is the kurtosis which is given in equation 14.

$$K(x) = \mathbb{E}(x^4) - 3(\mathbb{E}(x^2))^2 \tag{14}$$

As unit variance is assumed, this equation can be rewritten as the fourth moment of x minus three. In mathematical terms this equals $K(x) = \mathbb{E}(x^4) - 3$. Both the absolute and squared kurtosis are often used as non-Gaussianity measures. These are both zero for Gaussian variables and (almost) always positive for every other distribution. The kurtosis is widely used in practice as it is easy to derive and the theoretical analysis can be simplified due to linear properties. These properties can be illustrated using previous notation. For the ICA analysis the kurtosis of the variable y is needed. $K(\mathbf{y}) = K(\mathbf{z}^T\mathbf{s}) = K(z_1s_1 + z_2s_2)$ for a two-dimensional dataset. Now the linear properties of the kurtosis simplify the previous equation to $K(\mathbf{y}) = z_1^4K(s_1)+z_2^4K(s_2)$. The absolute (or squared) value of this measure can be maximized under the constraint of unit variance of $\mathbf{y}$ in order to calculate the independent components. Although the kurtosis is a very practical measure which is easy to compute, it is sensitive for outliers, because the empirical kurtosis is based on only a few observations in the tails of the distribution (Huber, 1985). Therefore alternative must also be taken into account for estimating the non-Gaussianity of a random variable.

The negentropy is a second measure for the non-Gaussianity of a certain random variable. This measure is closely related to another measure in the information theory namely the entropy which calculates the degree of information that the sample of the random variable gives (Hyvarinen & Oja, 2000). As a result the more unpredictable the random variable is, the larger is its entropy. In mathematical terms the entropy is given in equation 15. Note that this expression only holds for continuous distributions. The entropy can also be calculated for any discrete distribution but this is not of interest in this thesis.

$$H(y) = -\int p(y)log(f(y))dy \tag{15}$$

The main idea of using the entropy as a measure for Gaussianity is that a Gaussian variable has the largest entropy among all random variables with equal variances. The negentropy of a variable y can be calculated using the following equation: $J(\mathbf{y}) = H(\mathbf{y_{gauss}}) - H(\mathbf{y})$. Where the first part equals a gaussian distribution with the same covariance structure as $\mathbf{y}$. This measure is always greater or equal to zero and can be used as a measure for non-Gaussianity. Although it is not sensitive to outliers, this measure can actually be hard to compute as a (non-parametric) estimate of the probability density

function is required. Therefore it is often necessary to use approximations of the negentropy in the analysis which were proposed by Hyvarinen (1999). These approximations have the structure as given in equation 16.

$$J(y) \approx \sum_{i=1}^{p} k_i[\mathbb{E}(G_i(y)) - \mathbb{E}(G_i(\nu))]^2 \tag{16}$$

Here k_i is a positive constant, G_i a non-quadratic function and ν a standardised Gaussian variable. In the Appendix 7.4 different options for the functions G_i are discussed.

Preprocessing Procedure for Independent Component Analysis

Before any analysis it is necessary to preprocess the data in a way such that it becomes suitable for ICA. Equivalently as in PCA, the first preprocessing is to center the variables of interests. This means that all variables will have be transformed such that they will have a zero mean. This can be done by simply subtracting the variable mean from every observation. This centering is done purely to simplify the ICA analysis. It is also possible to add the mean later in the analysis back, but this will not be considered in this thesis.

The second step in the preprocessing procedure after centering is whitening the data. This means that the observed vector $\mathbf{x}$ is transformed in a linear way into $\tilde{\mathbf{x}}$ such that $\mathbb{V}(\tilde{\mathbf{x}}) = \mathbf{I}_n$. This whitening transformation can be obtained with an EVD which is explained earlier in Section 3.1. Whitening of data matrix $\mathbf{X}$ can be done as described in the equation 17.

$$\tilde{\mathbf{X}} = \mathbf{E}\mathbf{D}^{-\frac{1}{2}}\mathbf{E}^{\mathbf{T}}\mathbf{X} \tag{17}$$

Here $\mathbf{E}$ refers to the eigenvector matrix where column i corresponds to the i'th eigenvector of $\mathbf{X}$. $\mathbf{D}$ equals a diagonal matrix with all eigenvalues on the diagonal elements. The whitening ensures that the amount of parameters which need to be estimated for ICA reduces from n^2 to $\frac{n(n-1)}{2}$ and therefore it is a convenient transformation simplifying this analysis. The amount of estimated parameters are reduced because the $\mathbf{A}$ matrix becomes orthogonal when the input variables are whitened and the amount of degrees of freedom of an orthogonal matrix with dimension n equals $\frac{n(n-1)}{2}$. For the rest of this thesis it is assumed that all variables are centered and whitened before any ICA analysis.

Fast-ICA

One of the most efficient ways to estimate the independent components is by means of Fast-ICA. This method is known for its high efficiency in the maximization task of ICA. This Fast-ICA procedure consists of two parts. The first part is the estimation of an individual independent component and the second part consists of estimating all components which makes use of the first part in the estimation procedure.

The estimation of a single independent component is done by choosing $\mathbf{w}$ in way such that the non-Gaussianity of the projection $\mathbf{w}^T\tilde{\mathbf{x}}$ is maximized under the constraint that the variance of this projection is equal unity. Here $\tilde{\mathbf{x}}$ refers to the whitened transformation of $\mathbf{x}$. The Fast-ICA procedure makes use of the functions g which are the first order derivatives of the G functions described earlier

Algorithm 4: Fast Independent Component Analysis (ICA)

1: Initialization of weight vector $\mathbf{w}$ at random
2: Let $\mathbf{w}^+ = \mathbb{E}(\tilde{\mathbf{x}}g(\mathbf{w}^T\tilde{\mathbf{x}})) - \mathbb{E}(g'(\mathbf{w}^T\tilde{\mathbf{x}}))\mathbf{w}$
3: Let $\mathbf{w} = \frac{\mathbf{w}^+}{\|\mathbf{w}^+\|}$
4: Repeat from step 2 until convergence

in this section. Again these g functions can be found in the Appendix 7.4. The Fast-ICA algorithm for a single independent component is summarized in algorithm 4.
The first term in the second step of the algorithm comes from maximizing an approximation of the negentropy namely maximizing $\mathbb{E}(G(\mathbf{w}^T\mathbf{x}))$. The second part in this equation ensures that the optimal solution of $\mathbf{w}$ has a variance which equals unity. The convergence in step 4 refers to the fact that the procedure must be repeated until the dot product of $\mathbf{w}$ converges towards one.

Now the procedure of Fast-ICA for multiple components will be discussed. The main idea of estimating multiple independent components is that algorithm 4 is used for multiple weight vectors $\mathbf{w}_1, ..., \mathbf{w}_n$. After every iteration in the algorithm $\mathbf{w}_1, ..., \mathbf{w}_n$ must be decorrelated to avoid that multiple weight vectors end in the same (optimal) solution. Hyvarinen (1999) proposed an efficient procedure to decorrelate the multiple weight vectors after every iteration. For this method matrix $\mathbf{W}$ represents a matrix with all n weight vectors as columns. First $\mathbf{W}$ must be transformed which is done in the following equation $\mathbf{W}^{(t)} = \frac{\mathbf{W}}{\sqrt{\|\mathbf{W}\mathbf{W}^\mathbf{T}\|}}$. In the second step the following transformation is repeated until convergence: $\mathbf{W}^{(\mathbf{t+1})} = \frac{3}{2}\mathbf{W}^{(\mathbf{t})} - \frac{1}{2}\mathbf{W}^{(\mathbf{t})}\mathbf{W}^{\mathbf{T}^{(\mathbf{t})}}\mathbf{W}^{(\mathbf{t})}$. The obtained $\mathbf{W}$ matrix is fully decorrelated which ensures that the multiple weight vectors will not end the same optimum.

3.2.1 Feature Extraction with Independent Component Analysis (ICA-FX)

In the previous section, ICA is described as an unsupervised learning tool which can help with for instance data visualisation and the blind source separation problem. Kwak & Choi (2003) generalizes the ICA framework such that it becomes suitable for feature extraction in binary classification problems (ICA-FX). Further research also proposed a method which generalizes the ICA-FX such that it becomes suitable for multinomial classification and regression problems, but this is beyond the scope of this thesis (Kwak & Kim, 2006). The main idea of the ICA-FX method is that two sets of variables are created; One set containing variables that carry information about the class label and one set that does not. Only the features in the first set will be used in the ICA-FX analysis and this results in a reduction of the dimensionality of the feature space. The general idea of ICA-FX is to maximize the mutual information between the class label and created features as this ensures that the lower-bound of the error probability is minimized, also known as Fano's inequality. Another result which is crucial in the ICA-FX technique is that any transformation of the input variable will have a lower mutual information compared to the original variable. Therefore the problem is to extract new features (with a dimension lower then n) from the input variables $\mathbf{X}$ such that the mutual information between the new extracted features and the class labels becomes as close to the mutual information between the input variables $\mathbf{X}$ and the class labels.

Now the ICA-FX algorithm will be discussed. It is assumed that the set of features which contain the information of the class label has dimension M and can be summarized as a linear combination of the input features $\mathbf{F_a} = [\mathbf{f}_1, .., \mathbf{f}_M]$. The other features are as independent of class label c as much as possible and is equivalent to $\mathbf{F_b} = [\mathbf{f}_{M+1}, ..., \mathbf{f}_n]$. If the mutual information between $\mathbf{F_b}$ and the class labels c equals zero it is possible to reduce the dimensionality from n to M without any loss of information about the class label. Just like before it is assumed that there exists n independent and non-Gaussian sources $[\mathbf{s}_1, ..., \mathbf{s}_n]$ which are also independent of the class labels c, where $c_i \in \{-1, 1\}$. The observed $\mathbf{x}$ can be decomposed as described in equation 18.

$$\mathbf{x} = \mathbf{As} + \mathbf{bc} \tag{18}$$

The unmixing part of the algorithm is slightly different compared to the conventional ICA algorithm where only the matrix $\mathbf{W}$ was introduced. In the ICA-FX algorithm the unmixing phase can be described like equation 19.

$$\mathbf{u} = \mathbf{Wx} + \mathbf{vc} \iff \mathbf{\Lambda\Pi s} = \mathbf{Wx} + \mathbf{vc} \tag{19}$$

Here $\mathbf{\Lambda}$ represents a scale matrix and $\mathbf{\Pi}$ is a permutation matrix. The algorithm of ICA-FX is based on a maximum likelihood (ML) estimation which results in a learning rule for an optimal $\mathbf{W}$. For the rest of this method it is assumed that every element of $\mathbf{u}$ is both independent with the class label and all other elements of $\mathbf{u}$. Now the (log)likelihood function of the observed data $\mathbf{x}$ can be written as equation 20

$$\mathbb{L}(x, c, \mathbf{W}) = log(|\mathbf{W}|) + \sum_{i=1}^{N} log(p_i(u_i)) + log(p(c)) \tag{20}$$

This results is obtained due to the fact that the variables u and the class label c are independent, therefore $p(u, c) = p(u)p(c)$ as described earlier in the ICA section. The full mathematical derivation of this ML method is given in Kwak & Choi (2003) and their final result for the learning rules for both $\mathbf{W}$ and $v_a = [w_{1,N+1}, ..., w_{M,N+1}]$, is given in equations 21 and 22.

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} + \eta_1[\mathbf{I}_N - \phi(u)f^T]\mathbf{W}^{(t)} \tag{21}$$

$$v_{\mathbf{a}}^{(t+1)} = v_{\mathbf{a}}^{(t)} - \eta_2\phi(u_{\mathbf{a}})c \tag{22}$$

Here η_1 and η_2 are the learning rates which need to be tuned as their does not exist a closed form optimal solution for these variables. $f^T = \mathbf{W}x$ and can be interpreted as the projection of iteration i and $\phi(u) = [\phi_1(u_1), ..., \phi_N(u_N)]$ can be derived from the following equation: $\phi_i(u_i) = \frac{dp_i(u_i)}{du_i}\frac{1}{p_i(u_i)}$. After both variables are repeated until convergence the M features which contain information about c can be obtained using $\mathbf{F_a} = \mathbf{W}x$.

3.3 t-Distributed Stochastic Neighbor Embedded (t-SNE)

The t-Distributed Stochastic Neighbor Embedded (t-SNE) technique is the third dimension reduction method which is taken into consideration in this thesis. The first step of the t-SNE algorithm is that Euclidean distances of high dimensional datapoints should be converted into similarity measures. Here the conditional probability is used as a similarity measure which is mathematically denoted in equation 23. Later the data $\mathbf{X} = \{x_1, ..., x_n\} \in \mathbb{R}^d$ will be compressed into a lower dimensional

subspace $\mathbf{Z} = \{z_1, ..., z_n\} \in \mathbb{R}^p$ where $p \ll d$ (where often $p = 2 \vee p = 3$) such that the local structure in maintained.

$$p_{i|j} = \frac{exp\{-\|x_i - x_j\|^2/2\sigma_i^2\}}{\sum_{i \neq k} exp\{-\|x_i - x_k\|^2/2\sigma_i^2\}} \tag{23}$$

Here σ_i is the variance of the Gaussian that is centered around the datapoint x_i. How this variance is estimated will be discussed later in this section. The conditional probability is sensitive to outliers in the data. Therefore it is more convenient to use the symmetrized conditional probability p_{ij} as similarity measure in order to avoid these types of problems (Linderman et al., 2017). This measure can be calculated by $p_{ij} = \frac{p_{i|j} + p_{j|i}}{2N}$. Now in a similar way the symmetrized conditional probabilities of the datapoints in $\mathbf{Z}$ can be calculated using a Cauchy distribution instead of an earlier used Gaussian distribution which is done in equation 24. Note that the Cauchy distribution is equivalent to a t-distribution with one degree of freedom.

$$q_{ij} = \frac{(1 + \|z_i - z_j\|^2)^{-1}}{\sum_{k \neq l}(1 + \|z_k - z_l\|^2)^{-1}} \tag{24}$$

In an ideal world points z_i and z_j are mapped in a way such that their conditional probability q_{ij} is exactly equal to the (symmetrized) conditional probability of the original point p_{ij}. The Kullback-Leibler divergence is a measure which calculates the mismatch between p_{ij} and q_{ij}. T-SNE minimizes the sum of the Kullback-Leibler divergence over all datapoints using a standard gradient descent technique. The cost function of this minimization problem is given in equation 25.

$$C(Z) = \sum_i KL(P_i \mid\mid Q_i) = \sum_i \sum_j p_{ij} \frac{p_{ij}}{q_{ij}} \tag{25}$$

The gradient of the cost function with respect to z_i is given in equation 26. The derivation for this gradient makes use of the fact that the joint probability distribution of $\mathbf{Q}$ is student t-distributed and can be found in van der Maaten & Hinton (2008).

$$\frac{\partial C(Z)}{\partial z_i} = 4 \sum_j (p_{ij} - q_{ij})(z_i - z_j)(1 + \|z_i - z_j\|^2)^{-1} \tag{26}$$

This gradient is used for the updating step in the gradient descent algorithm. The t-SNE algorithm is summarised in algorithm 5 and depends on several hyperparameters.

Algorithm 5: t-Distributed Stochastic Neighbor Embedding (t-SNE) Algorithm

---

1: **Data:** $\mathbf{X} = \{x_1, ..., x_N\}$, cost function parameter: perplexity and optimization parameters: number of Iterations (T), learning rate (η) and momentum ($\alpha(t)$).
2: **Compute:** p_{ij} using equation 23 and the perplexity parameter.
3: **Sample:** $Z^{(0)} = \{z_1^{(0)}, ..., z_N^{(0)}\}$ from $\mathcal{N}(0, 10^{-4}I)$
4: **Update:** $Z^{(t)} = Z^{(t-1)} + \eta \frac{\partial C(Z)}{\partial Z} + \alpha(t)(Z^{(t-1)} - Z^{(t-2)})$ until convergence or if the maximum number of iterations is reached

---

In this thesis, the number of iterations is set at 1000, the momentum is equal to 0.5 for $t \leq 250$ and 0.8 otherwise and the learning rate is set to 100 and will be updated every iteration by means of an adaptive learning procedure (van der Maaten & Hinton, 2008). Now only the σ_i must be estimated

in order to finalise the algorithm. It is not likely that there exists a single value for σ_i that is an optimal solution for all datapoints. A binary search is done for σ_i in way such that a density P_i is produced that has a fixed perplexity which is specified by the user. The perplexity is a transformation of the Shannon entropy and it is notable to mention that the perplexity increases monotonically with the variance σ_i. The perplexity of a distribution can be calculated with $Perp(P_i) = 2^{H(P_i)}$ where $H(P_i)$ equals the Shannon entropy $H(P_i) = -\sum_j p_{j|i} log_2(p_{j|i})$. The full t-SNE algorithm is robust against changes in the perplexity and normal values for this parameter are between 5 and 50. The perplexity is a parameter which reflects the balance between the local and global aspects of the data. The parameter can be seen as a guess about the number of close neighbors each data point has. So when the perplexity parameter has a low value this corresponds that the algorithm is more focused on preserving the local structure and when the perplexity parameter has a high value this means that the algorithm is more focused on preserving the global structure of the data in the reduced subspace.

3.3.1 Supervised t-Distributed Stochastic Neighbor Embedding

It is also possible to extend the t-SNE framework such that it becomes suitable for supervised learning purposes. As stated in the literature review, the Euclidean distance which used in the t-SNE algorithm before is not suitable for supervised t-SNE as it does not has the desired properties. Xu et al. (2020) proposed a supervised t-SNE algorithm for classification problems which makes use of the Aitchison distance in the t-SNE algorithm. The Aitchison distance between two vectors $\mathbf{x} = \{x_1, ..., x_p\}$ and $\mathbf{y} = \{y_1, ..., y_p\}$ can be calculated using equation 27.

$$d_a(\mathbf{x}, \mathbf{y}) = [\sum_{i=1}^{p} (ln \frac{x_i}{g(\mathbf{x})} - ln \frac{y_i}{g(\mathbf{y})})^2]^{\frac{1}{2}} \tag{27}$$

Here $g(\mathbf{x})$ and $g(\mathbf{y})$ refer to the geometric mean of vectors $\mathbf{x}$ and $\mathbf{y}$ respectively. With the Aitchison distance measure the conditional probabilities which are necessary for the t-SNE algorithm have to be calculated in the following way: $p_{i|j} = \frac{exp\{-d_a(x_i,x_j)/2\sigma_i^2\}}{\sum_{i \neq k} exp\{-d_a(x_i,x_k)/2\sigma_i^2\}}$. The reduced dimensional data $\mathbf{Z} = \{z_1, ..., z_N\}$ can be used as input features for the classification problem. Now assume that a new sample x_0 is given and the goal is to obtain input features for the classification problem. Now the k-nearest neighbors have to be identified using the conditional probabilities as given in equation 28.

$$sp_{0,i} = \frac{exp\{-d_a(x_0, x_i)/2\sigma_i^2\}}{\sum_{h \neq i} exp\{-d_a(x_i, x_h)/2\sigma_i^2\}} \tag{28}$$

All these conditional probabilities can be order such that the following holds: $sp_{0,(1)} > ... > sp_{0,(N)}$. The k points with the highest conditional probability are identified as the k-nearest neighbors of x_0. In the reduced t-SNE subspace these k points are denoted as: $z' = \{z'_1, ..., z'_k\}$. Finally the input features for the classification problem can be calculated as follows: $\mathbf{Z}_0 = \sum_{i=1}^{k} w_i z'_i$ where $w_i = \frac{sp_{0,(i)}}{\sum_{i=1}^{k} sp_{0,(i)}}$. The amount of neighbors k has to be tuned using cross validation and all other hyperparameters have to selected as described in the general t-SNE algorithm above.

3.4 Multidimensional Scaling (MDS)

Multidimensional Scaling (MDS) is a dimension reduction technique where a lower dimensional subspace is created in a way such that the pairwise spatial distances of the original higher dimensional spaces are preserved. First classical MDS will be explained as this approach is very closely related to PCA and forms the basis of the metric MDS approach which will be discussed afterwards.

3.4.1 Classical MDS

As described above the idea of (classical) MDS relies on the preservation of the spatial distances of the higher dimensional space when it is compressed into a lower dimensional subspace (Tripathy & Ghela, 2022). Let $\mathbf{X} = \{x_1, ..., x_n\}$ be a dataset where every $x_i \in \mathbb{R}^d$. Now a n-dimensional square matrix $\mathbf{M}$ can be created which represent every pairwise distance between all n points such that M_{ij} represents the distance between point x_i and x_j. The (classical) MDS approach tries to find a subspace $\mathbf{Z} = \{z_1, ..., z_n\}$ where every $z_i \in \mathbb{R}^p$ and $p \leq d$ such that $\|z_i - z_j\|_2 \approx M_{ij}$. For further analysis it is convenient to introduce a new matrix $\mathbf{D}$ such that the following relation holds $D_{ij} = M_{ij}^2$. Now assume that there exist points $\{z_1, ..., z_n\}$ such that that their distances are exactly equal to the distances which are represented in the matrix $\mathbf{M}$. If this assumption holds this equation can be written as equation 29. Note that all mathematical notation is in line with the work of Tripathy & Ghela (2022).

$$D_{ij} = \|z_i - z_j\|_2^2 = \|z_i\|_2^2 + \|z_j\|_2^2 - 2z_i^T z_j \tag{29}$$

This can be rewritten in the following way: $z_i^T z_j = \frac{1}{2}(\|z_i\|_2^2 + \|z_j\|_2^2 - D_{ij})$. Note that if all n points are centered their pairwise distance remains the same. Therefore we may assume that the sum of all observations in the subspace equal to zero: $\sum_{i=1}^n z_i = 0$. As a result of this assumption and using equation 29, equation 30 must hold.

$$\frac{1}{n}\sum_{i=1}^n D_{ij} = \frac{1}{n}\sum_{i=1}^n \|z_i\|_2^2 + \|z_j\|_2^2 - \frac{2}{n}\sum_{i=1}^n z_i^T z_j = \frac{1}{n}\sum_{i=1}^n \|z_i\|_2^2 + \|z_j\|_2^2 \tag{30}$$

Note that the part $\|z_j\|_2^2$ is not part of any summation and that the second summation of the first equation equals to zero due to the centering assumption. The following statement also must hold as a result of the imposed zero mean assumption: $\frac{1}{n^2}\sum_{i=1}^n \sum_{j=1}^n D_{ij} = \frac{2}{n}\sum_{i=1}^n \|z_i\|_2^2$. Now we can express z_i and z_j in terms of known values of D_{ij} using equation 29 as is done in equation 31.

$$z_i^T z_j = \frac{1}{2}[\sum_{i=1}^n D_{ij} + \frac{1}{n}\sum_{j=1}^n D_{ij} - \frac{1}{n^2}\sum_{i=1}^n \sum_{j=1}^n D_{ij} - D_{ij}] \tag{31}$$

This result can be written in matrix notation such that it becomes clear how to reduce the high dimensional space into a lower dimensional subspace. This transformation into matrix notation is done in equation 32.

$$\mathbf{Z}\mathbf{Z}^T = \frac{1}{2}[\frac{1}{n}ee^T\mathbf{D} + \frac{1}{n}\mathbf{D}ee^T - \frac{1}{n^2}ee^T\mathbf{D}ee^T - \mathbf{D}] = -\frac{1}{2}\mathbf{HDH} \tag{32}$$

Here $\mathbf{H}$ is equivalent to the centering matrix which is defined earlier in section 3.1.2. $\mathbf{HDH}$ can be seen as a matrix of inner products of centered data points. As done before it is possible to obtain a lower dimensional representation of these n observations by performing an EVD on $\mathbf{HDH}$. From now on the matrix $\mathbf{HDH}$ is called matrix $\mathbf{B}$ for convenience. The obtained eigenvectors must be scaled with the square root of the corresponding eigenvalue in order to compute the lower dimensional subspace. The classical MDS algorithm is summarised in algorithm 6.

As only the first p eigenvectors and corresponding eigenvalues are selected to reduce the input space to a p-dimensional subspace, the equality sign of equation 29 will not hold anymore. Similar as in t-SNE, often $p = 2 \vee p = 3$ such that it can easily be used for visualization purposes. The distances in the subspace will approximate the dissimilarities of the higher dimensional inputspace. In a situation where all eigenvectors are selected, so $p = d$, the equality sign will hold again.

Algorithm 6: Classical Multidimensional Scaling Algorithm

1: **Compute:** matrix $\mathbf{D}$ which can be computed with the given distance matrix $\mathbf{M}$
2: **Calculate:** matrix $\mathbf{B} = \mathbf{HDH}$
3: **Compute:** $\{u_1, ..., u_p\}$ which are the top p eigenvectors of matrix $\mathbf{B}$ with their corresponding eigenvalues $\lambda_1, ..., \lambda_p$
4: **Compute:** the $n \times p$ matrix $\mathbf{Z}$ where the i'th column of $\mathbf{Z}$ equals to $\sqrt{\lambda_i} u_i$.

3.4.2 Metric MDS

Metric MDS can be seen as a generalization of classical MDS. In classical MDS analysis, the distances between two points in the created subspace approximate the dissimilarities (squared distances) between these two points in the original higher dimensional space. With metric MDS this assumption is relaxed such that the interpoint distances between two points in the created subspace approximate a function of the dissimiliraties between these points in the original space. In mathematical terms this can be expressed like equation 33. The mathematical notation is in line with the notation used by Williams (2002).

$$d_{ij} \approx f(\delta_{ij}) \tag{33}$$

Note that for classical MDS $f(\delta_{ij}) = D_{ij}$. The function f must be specified by the user before any metric MDS analysis. This does not mean that the function cannot have parameters which need to be estimated such as $f(\delta) = \alpha + \beta\delta$. f can be any analytical function as long as it is monotonically increasing in δ. The coordinates of the subspace, d_{ij}, can be obtained by minimizing the stress function with respect to d_{ij}. The stress function is given in equation 34.

$$S = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij}(d_{ij} - f(\delta_{ij}))^2}{\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij}^2} \tag{34}$$

Here w_{ij} is a weight function which also needs to be determined by the user. A commonly used weight function for these types of problems is $w_{ij} = \frac{1}{\delta_{ij}}$. In a perfect situation this stress function will equal zero which corresponds with the situation that the interpoint distances in the subspace equal the function of dissimilarities in the original space. The stress function can be minimized using a gradient descent algorithm.

3.5 Dimension Reduction Techniques Comparison

Despite that sometimes the different dimensionality reduction techniques seem to differ completely from each other, all the techniques described above do relate with each other in a particular way. Therefore this section will discuss the different relations between the dimensionality reduction techniques described earlier in this Section.

Relation between PCA and MDS

The first connection which will be discussed is the relation between Principal Component Analysis (PCA) and Multidimensional Scaling (MDS). It can be shown that the MDS solution is equivalent to the PCA solution of the sample covariance matrix multiplied by the amount of observations.

This transformation of the sample covariance can mathematically expressed as $n\mathbf{S} = \mathbf{X}^T\mathbf{H}\mathbf{X} = (\mathbf{H}\mathbf{X})^T(\mathbf{H}\mathbf{X})$. Here $\mathbf{H}$ is again the earlier defined centering matrix (see Section 3.1.1). To prove that the MDS and the transformed PCA solutions are equivalent it is convenient to write the EVD of the matrix $\mathbf{B}$ like equation 35. The matrix $\mathbf{B}$ is defined in the MDS section above and is equal to $(\mathbf{H}\mathbf{X})(\mathbf{H}\mathbf{X})^T$.

$$\mathbf{B}v_i = \lambda_i v_i \tag{35}$$

Both sides can now be premultiplied by $(\mathbf{H}\mathbf{X})^T$ such that the equation above becomes equivalent to $(\mathbf{H}\mathbf{X})^\mathbf{T}(\mathbf{H}\mathbf{X})(\mathbf{H}\mathbf{X})^\mathbf{T}v_i = \lambda_i(\mathbf{H}\mathbf{X})^\mathbf{T}v_i$. It is clear that λ_i is the i'th eigenvalue of $(\mathbf{H}\mathbf{X})^\mathbf{T}(\mathbf{H}\mathbf{X})$. Note that these are also a eigenvalue of $n\mathbf{S}$. The corresponding eigenvector is defined as $y_i = (\mathbf{H}\mathbf{X})^\mathbf{T}v_i$. The MDS solution can be obtained by projecting a unit eigenvector of n$\mathbf{S}$ ($\hat{y_i} = \lambda_i^{-\frac{1}{2}}y_i$) onto the centered data $\mathbf{H}\mathbf{X}$. The mathematical derivation can be found in equation 36.

$$\mathbf{H}\mathbf{X}\hat{y_i} = \lambda_i^{-\frac{1}{2}}\mathbf{H}\mathbf{X}(\mathbf{H}\mathbf{X})^\mathbf{T}v_i \tag{36}$$

Note that the matrix in the second part of the equation is identical to the matrix $\mathbf{B}$ and there can be rewritten as $\lambda_i^{-\frac{1}{2}}\mathbf{B}v_i$. With the use of equation 35 the transformed PCA solution can be showed to be identical to the MDS solution: $\mathbf{H}\mathbf{X}\hat{y_i} = \lambda_i^{\frac{1}{2}}v_i$.

Relation between ICA and PCA

With independent component analysis (ICA) a subspace is created such that the features become as independent as possible in the whitened feature space. On the contrary a PCA solution finds features which are orthogonal in the original feature space. To see the relation between the two different solution first two new concepts must be defined, namely PCA and ZCA whitening. PCA whitening can be applied to a dataset in order to decorrelate the data and setting the variance to unity using a PCA solution. This can be done in the following way: $\mathbf{W}_{PCA} = \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{U}^T$. Here $\mathbf{\Lambda}$ is the diagonal matrix with the eigenvalues as elements and $\mathbf{U}$ the matrix with eigenvectors as columns. ZCA whitening is another whitening procedure which is closely related to PCA whitening and it can be calculated by pre-multiplying the $\mathbf{W}_{PCA}$ with a rotation matrix. The ZCA whitening can be calculated using equation 37.

$$\mathbf{W}_{ZCA} = \mathbf{U}\mathbf{\Lambda}^{-\mathbf{\frac{1}{2}}}\mathbf{U}^\mathbf{T} = \mathbf{U}\mathbf{W}_{PCA} \tag{37}$$

Here the matrix with eigenvectors as columns $\mathbf{U}$ is used as rotation matrix. The ZCA whitening technique ensures that the average cross-covariance between each component of the whitened and the original vectors is maximized. Therefore ZCA whitening is often used to when it is desired that the whitened data is as close as possible to the original data. The ICA solution for estimating multiple independent components also includes a transformation of the weight matrix $\mathbf{W} = \frac{\mathbf{W}}{\sqrt{\mathbf{W}\mathbf{W}^\mathbf{T}}}$. This projection step can be seen as a ZCA whitening method where $(\mathbf{W}\mathbf{W}^\mathbf{T})^{-\frac{1}{2}}$ is used as rotation matrix instead of $\mathbf{U}$ (Stanford University, 2023). Note that the $\mathbf{W}$ matrices of the different solutions are not identical. Therefore it can be concluded that ICA is a special case of ZCA whitening. Instead of a matrix spanned by eigenvectors ($\mathbf{U}$) another rotation matrix is used. Namely a esimated rotation matrix which maximizes the negentropy meaning that an extra rotation step has to be done in order to obtain the independent components.

Relation between t-SNE and MDS

Now the connection between the MDS solution and the t-SNE solution. MDS is a technique where a subspace is created such that the distances between pairs of data points in the original feature space are preserved. On the other hand with t-SNE a subspcae is created such that neighborhood data points in the orginal space are maintained. This will result in a subspace where data points will be located close to each other when these data points are tight in the original space. Therefore t-SNE is often used when the user wants to preserve the local structures of the data. The MDS solution ensures that the global structure of the data is better preserved. Both methods use a certain measurement to identify the dissimilarities between data points. For the classical MDS method this is the distance matrix $\mathbf{M}$ and for the t-SNE method this is the conditional probability. In the t-SNE algorithm the Kullback-Leibler divergence measure is minimized which depend on the dissimilarity measures of this algorithm, namely the conditional probabilities. For the metric MDS the stress function S is minimized which depend on the dissimilarity measure chosen by the user. If we set the dissimilarity measure equal to the conditional probability such that: $f(\delta_{ij}) = p_{i|j}$ both methods minimizes a cost function which depends on the conditional probabilities between each pair of data points. For the the t-SNE this is the Kullback-Leibler divergence measure and for the (metric) MDS this is the stress function S.

3.6 Evaluation of the dimensionality reduction

The created subspaces for the dimension reduction techniques must be evaluated based on their clustering and classification performances. It is assumed that the amount of clusters in the unsupervised learning simulation study is known. The clusters in the reduced subspaces are created with a clustering algorithm known as the k-means algorithm. This algorithm minimizes the within cluster similarities and maximizes the between cluster similarities by iterative maximizing two different optimisation problems (Hartigan & Wong, 1979). For the supervised studies two different classifiers are used for making predictions. The first classifier is a standard logistic regression model which predicts a binary output based on the cumulative distribution function of the logistic density where the unknown parameters can be estimated using maximum likelihood estimation. The second classifier is a support vector machine (SVM). This is a machine learning algorithm which divides the input space with a linear hyperplane such that partition of the known datapoints are optimized. A kernel can be added to the SVM classifier such that the partition does not have to be linear. This classifier is used in combination with dimension reduction in already existing literature multiple times with significant effects (Anowar et al., 2021).

3.6.1 Evaluation Metrics for clustering

In the unsupervised learning studies, the data points are grouped into k different clusters. The advantage of a simulation study here is that for every data point it is known to which cluster it belongs. For the rest of this thesis it is assumed that the amount of clusters are known, this means that this value does not have to be estimated. The adjusted rand index (ARI) is a performance measure of the similarity between two different data clusterings while also taking cluster permutations into account Hubert & Arabie (1985). This measure also takes into account that some data points are clustered correctly by chance. The ARI is an extension of the normal rand index which is similar

to the accuracy in a classification problem, where the value is based on the relative amount of correct predictions. The hypergeometric distribution is added to the normal rand index such that is becomes the ARI in order to model the randomness. The ARI can be calculated using equation 38 and is in line with the mathematical notation of Yeung & Ruzzo (2001).

$$ARI = \frac{\sum_{i,j}\binom{n_{ij}}{2} - [\sum_i \binom{n_{i.}}{2}\sum_j \binom{n_{.j}}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{n_{i.}}{2} + \sum_j \binom{n_{.j}}{2}] - [\sum_i \binom{n_{i.}}{2}\sum_j \binom{n_{.j}}{2}]/\binom{n}{2}} \tag{38}$$

Here n_{ij} refers to the number of objects that are of class i and clustered in group j. $n_{i.}$ is the number of object which are of class i and $n_{.j}$ is the number of object which are clustered in group j. The ARI is a value between minus one and one. An ARI value of zero corresponds with a random agreement between the two clusterings, a value of minus one indicates that the two clusterings are completely different from each other and an ARI of one indicates a perfect agreement of the two cluster partitions.

3.6.2 Evaluation Metrics for classification

For the classification problem, the dataset will be divided into two parts: A training set and a test set. In this thesis approximately 67% of the observations are used for the training sample, The rest of the observations are used for the test sample. This means that the amount of observations in the test sample is relatively high. However this is due to the fact that for both datasets the amount of observations is around 180 which is relatively small. The described supervised methods will be estimated using the training set and the performances will be calculated with the test set. The accuracy measures how many times a model predicts the class correctly and can be calculated using equation 39.

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{39}$$

TP refers to the situation where both the prediction and actual value are equal to one. TN is the other way around where both the prediction and actual value are equal to zero. FN refers to the situation where the actual value is one but the model predicts zero and FP is when the actual value is zero but the model predicts one. A disadvantage of the accuracy measure is that it can give biased results when their is class imbalance within the dataset. Therefore other measures are more robust against class imbalance problems. An alternative measure for the accuracy which is far more robust against potential class imbalance is the F_1 score. The F_1 score combines two performance measures, the precision and the recall and it can be calculated using equation 40.

$$F_1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{40}$$

The abbreviations on the right hand side of the equation are identical to values in equation 39. A perfect model has a F_1 score of 1, in this case the accuracy is also 100%. The last evaluation metric which will be taken into consideration for the supervised study is the AUC-ROC (AUC). The AUC is a numerical value between zero and one and it indicates how much a certain classifier in capable of distinguishing the binary classes. An perfect classification of the two clusters will result in an AUC value of one. The AUC is closely related with the ROC probability curve, which represents the relation between the True Positive Rate (TPR) and the False Positive Rate (FPR) at various threshold values. The threshold value of interest here is the value which determines at which calculated probability the

classifier will label a certain data point as one instead of zero. The TPR and FPR can be calculated using the following equations:

$$TPR = \frac{TP}{TP + FN} \qquad FPR = 1 - \frac{TN}{TN + FP} \tag{41}$$

The abbreviations in equation 41 are again in line with the abbreviations defined in equation 39. The AUC can then be approximated with the summation of TPR values for different FPR values determined by a set of threshold values. This calculation converges towards the AUC when the cardinality of the threshold values set increases.

4 Data

The data section is divided in two parts: (1) the data section for the unsupervised where mainly the different data generating processes are discussed and (2) the data section for the supervised study where the data descriptions are given for the used real-world classification problems.

4.1 Simulation Study for Unsupervised Learning

The unsupervised study will be performed with a simulation study. This means that the class label for each observation is known before any dimensionality reduction and clustering. The class label information will be used in order to evaluate the clustering performances on the reduced dataset. For convenience the dimension of the dataset without any performed reduction is set to 25 and the amount of clusters can only have one of the following values: $k \in \{2, 3\}$. The 25-dimensional dataset will be compressed into a p-dimensional space where p can take the following values: $p \in \{2, 3, 5, 10\}$. At last it is of interest how the performances is influenced by the addition of noise ϵ. After the generation of the data noise can be added to each data point by adding a Gaussian distributed random variable with mean zero and variance of ϵ to each data point such that the final data point can be decomposed as: $x_{ij,\epsilon} = x_{ij} + \eta_{ij,\epsilon}$ where $\eta_{ij,\epsilon} \sim \mathcal{N}(0, \epsilon)$. For this study ϵ can only have one of the following values: $\epsilon \in \{0.5, 5, 25\}$. Moreover, every method is replicated 1000 times for each scenario, except for the RKM due to high computational times.

The addition of noise, extra dimensions in the subspaces and the amount of clusters can be seen as three different 'treatments' on the considered dimension reduction models. To estimate the interactions and effects of these treatment a repeated measures analysis of variance (ANOVA) is performed. Whenever the corresponding p-value of a treatment is smaller than 0.05, the effect of this treatment is considered as significant. The size of this effect or partial eta squared (η_p^2) can be calculated by dividing the variance explained with this specific treatment by the total variance. The sizes of these effects makes the analysis of the behavior of the considered dimension reduction method easier as the ANOVA tests point out which set of treatments mainly influence the clustering performances. The effect of the used method can also be calculated with the use of a repeated measure ANOVA test and shows whether or not the choice of dimension reduction technique matters significantly. The method or dimension reduction technique is called the within factor in the ANOVA tests and the other three treatments are called the between factors.

The data will be generated using different three scenario's. For every scenario the first i columns are used in order to determine the cluster structure. The rest of the columns are used in order to increase the dimensionality of the dataset and are (non-)linear combinations of the first columns. Now the cluster structure is contained or partially contained in every column in a (non-)linear way and conventional clustering methods on the whole dataset will not be efficient anymore Milligan et al. (1983). Now the different scenario's of generating the data columns are discussed:

Scenario 1: In the first scenario all the first i columns are all normal distributed. The last columns are all linear combinations of the first i columns. It is suspected that (kernel) PCA and t-SNE will perform strictly better compared to the ICA as the ICA algorithm assumes non-Gaussian variables. When the amount of noise increases it is suspected that the robust PCA will outperform all the other dimensionality reduction methods as this method is more resistant for added noise due to the fact that the location and scale parameters are robustly estimated.

Scenario 2: The second scenario is relatively similar to the first scenario. The difference is that the remaining columns are all non-linear combinations of the first i columns. Now it is suspected that conventional PCA and classical MDS will perform less compared to the first scenario as these methods will not handle these non-linearity relations well. On the other hand, the t-SNE and kernel PCA method might perform relatively better compared to the first scenario as these methods are better in identifying non-linear relationships in the data. It is also suspected that the ICA technique will work relatively well especially under the situation without added noise as most of the generated variables will not be Gaussian distributed. This is because non-linear combinations of Gaussian variables are non-normal distributed. However the addition of (Gaussian) noise will probably lead to a large declension of the clustering performance relative to the other considered dimension reduction techniques.

Scenario 3: In the third scenario the first i columns are all (non-)normal distributed and the rest of the columns are all (non-)linear combinations of the first i columns. It is suspected that ICA will perform better compared to the other scenario's as in this scenario, because the assumption of non-normality is not violated. Again, just as described in the previous scenario, the more classic dimension reduction techniques will probably perform less well due to many types of non-linearity's which will be represent in the data. It is suspected that the t-SNE method has less problems with these non-linearity's due to the fact that this method almost has no parametric assumptions.

4.2 Data for Supervised Learning

Breast Cancer Data

For the supervised learning study, real-world classification problems are used. This dataset contains information about different characteristics of individual cell nuclei and it used to classify to types of tumors in the corresponding breast. The tumor can be classified as malignant which means that the tumor is cancerous or it can be classified as benign which means that the tumor is non-cancerous [3]. For every observation three nuclei are taken and for every nucleus 10 different attributes about the tumor are known which are based on photographic images. In table 1 the data characteristics are

[3] 1 corresponds with a malignant tumor and 0 with a benign tumor

given for every first sample that is taken.

Table 1: Breast Cancer Data Characteristics

	Mean	St. dev	Min	Max
Diagnosis	**0.42**	**0.49**	**0**	**1**
Radius	14.27	3.54	6.98	28.11
Texture	18.95	4.12	9.71	39.28
Perimeter	92.95	24.43	43.79	188.5
Area	667.84	352.85	143.5	2499
Smoothness	0.10	0.01	0.06	0.14
Compactness	0.11	0.05	0.02	0.35
Concavity	0.09	0.08	0	0.43
Concave points	0.05	0.04	0	0.20
Symmetry	0.18	0.03	0.12	0.30
Fractal Dimension	0.06	0.01	0.05	0.10

The radius equals the mean of distances from center to points on the perimeter. The texture refers to the standard deviation of the gray-scale values. The smoothness corresponds with the local variation in radius length and the compactness can be calculated equals $\frac{perimeter^2}{area} - 1$. The concavitivity refers to severity of concave portions of the contour and the concave points equal the number of concave portions on the contour. At last the fractal dimension equals the "coastline approximation" - 1.

Alzheimer's Disease Data

In order to make a more general conclusion the performance of dimensional reduction models in a supervised setting, another real-world medical classification problem is used. The second data set includes handwriting data from 174 participants. Where some handwritings belong to diagnosed Alzheimer patients and others to healthy participants. For every individual 450 different handwriting features such as air time and extensions in x and y directions for 25 different writing tasks are measured which makes this problem potentially much harder for classification purposes as the original dimension is much larger compared to the previous empirical study. A more detailed overview of the Alzheimer's data can be found in the Appendix. Again, the data is retrieved from the University of California Machine Learning Database Fontanella (2022). In the sample 51% is diagnosed with the Alzheimer's disease. Therefore this data set can be considered as balanced.

5 Results

The results Section is also divided into two parts where one part focuses on the unsupervised study and the other part on the supervised study. The clustering performances for a two-dimensional subspace, the effect sizes for each ANOVA factor and numerical effects for the addition of noise and extra dimensions are shown for every considered scenario explained in Section 4. For the supervised study, the three different evaluation metrics are shown for all reduced subpaces and all models.

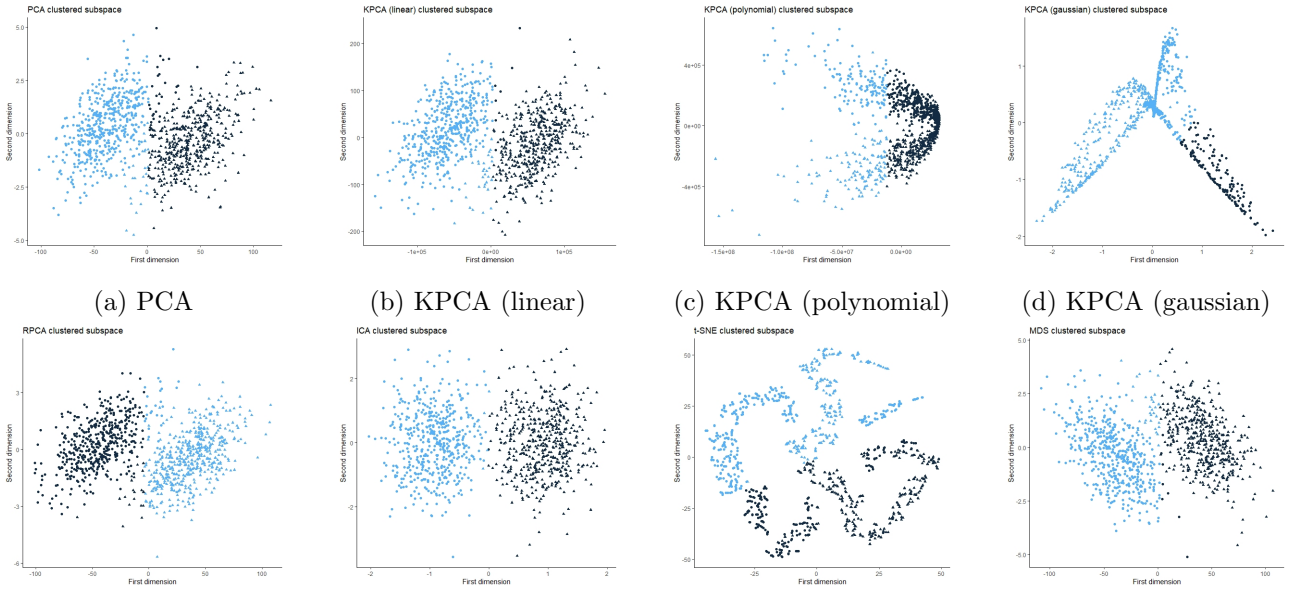
5.1 Unsupervised Study Results

The results for the unsupervised study are given per scenario. For each scenario the formed clusters for the reduced two-dimensional subspace are plotted for each dimension reduction technique except for the RKM as this method differs significantly from all other methods due to the fact that it is a simultaneous dimension reduction clustering method, instead of a tandem-like procedure. Therefore the RKM results will be discussed separately per scenario. Then for every scenario a repeated measures ANOVA test will be performed in order to estimate the effect of each 'treatment' and dimension reduction method as described in Section 4. The different treatments taken into considerations for this unsupervised study are: (1) the addition of noise to the data, (2) the number of dimensions in the reduced subspace and (3) the amount of clusters. Based on the outcome of the repeated measures ANOVA test for each scenario the treatments which cause the largest effects on the adjusted rand indices will be analyzed with the use of tables where the values represent the ARI for each model and the values between brackets the corresponding standard error. The formed clusters under the treatment of noise and the RKM based clusters can be found in the Appendix.

Scenario 1

In Figure 2 the formed clusters in the two-dimensional subspace are given for each dimensional reduction technique for the first scenario without the addition of any noise to the data. As expected the more conventional techniques like PCA, KPCA (linear), RPCA and MDS perform well and the within-cluster linear dependence is clearly visible here for these methods. However this contradicts the results of Chang (1983) where it was concluded that often the cluster structure was hard to detect when using only the first couple of principal components when the data was a mixture of two multivariate normal distributions. On the other hand, in this simulation study the two clusters are located relatively far away from each other (without any noise). This could lead to the over-performing clustering performance. The ICA method also performs significantly well for a two-dimensional subspace even though the non-normality ICA assumption is violated for all generated columns. This may be a result of the fact that bimodality of the marginal distributions is potentially very high when the data contains two clusters. The bimodality will be discussed more extensively later in the result section of this first scenario. When the reduced dimension increases, the variance of the ICA performance increases significantly which is due to the fact that for this situation too many independent components are extracted which probably do not contain cluster information. The t-SNE and other two KPCA methods (KPCA with a gaussian and polynomial kernel function) do not perform well in this scenario, because for both methods the kernel function is misspecified resulting in an incorrect expansion of the inputspace before any dimension reduction procedure. The t-SNE represents the high dimensional data in an unique way, however it clusters points in the subspace together which are also closely together in the original space. This particular way of mapping does not have to capture the correct clustering structure in the input data which may result in a lack of performance while clustering the data points in the subspace. Note that the performance of the t-SNE model is also highly sensitive to the perplexity hyperparameter which is not tuned in this simulation study. In the Appendix results for the formed clusters are shown in a situation where the clusters in the original space are located further from each other. This leads to a clearer distinction for all dimension reduction methods except the KPCA Gaussian model also for the methods which had much more clustering difficulties in the

situation depicted in Figure 2. The formed clusters under the treatment of many noise can also be found in the Appendix. This addition of noise directly leads to a situation where the distinction between the two generated clusters are less visible for all methods. Based on the figures it is not possible to point out which method is most suitable for this specific scenario. Therefore further analyses must be done in order to determine the preferred dimension reduction technique.



(a) PCA (b) KPCA (linear) (c) KPCA (polynomial) (d) KPCA (gaussian)

(e) RPCA (f) ICA (g) t-SNE (h) MDS

Figure 2: Clustered two-dimensional sub-spaces for scenario 1 without noise

Table 2 shows the results of the repeated measures ANOVA test for the first scenario. This test points out that the effect of the addition of noise to the data has an extremely significant effect on the ARI for the different dimension reduction techniques. The other two 'treatments' have a much lower effect to the ARI's and will therefore not be discussed extensively in this study. The effect of the addition of noise on the ARI is numerically represented in Table 3. The numerical values for the other effects can be found in the Appendix. As expected the used method also has large effect size. All significant higher order interactions can be found in the Appendix in Table 10. It is not surprising that for every higher order interaction the method factor is included, because the chosen method has a significant influence on the clustering performance which can be seen in Table 3.

Table 2: Effects for within and between factors in first scenario

	F-statistic	p-value	effect size
Reduced Dimension	1.054	0.320	0.012
Noise	312.48	0.000	0.781
Clusters	8.186	0.011	0.086
Method	281.17	0.000	0.935

As expected the overall performances decreases when noise is added to the data for all techniques taken into consideration except the t-SNE method. Again this can be a result of the proportion of importance of the local or global structure of the data which is influenced by the perplexity parameter which is

not tuned during this study. Another choice for the perplexity value potentially leads to significantly different clustering results. It is also plausible that the performances of most dimension reduction techniques are not highly influenced by a relatively small addition of noise, because the clusters which are generated are located far away from each other relative to the added noise. However, the ICA performances are already substantially influenced by this addition of noise as the ARI index decreases from 0.93 to 0.80 when the small noise is added to the data consisting of two clusters. A potential reason for this relatively large drop in clustering performance is that the data becomes less bimodal when noise is added to the data which can violate the normality assumption to a certain extend leading to lower ARI values. This hypothesis is graphically confirmed when looking at the bimodality of the independent components and original features which are shown in the Appendix. As bimodal variables are per definition less gaussian it is expected and observed that the ICA will perform better when there only exists two clusters despite that the clusters itself are simulated with gaussian components. Another conclusion that can be made is that the combination of the amount of clusters and addition of noise is quite stable and does not have a large effect on the calculated ARI's. This holds for all techniques except for the ICA method where the performance decreases overall slightly when the reduced dimension increases. This stable pattern for the other two treatments is in line with the high corresponding p-values of the repeated measures ANOVA.

Table 3: ARI and standard errors for the effect of Noise for two-dimensional subspaces in the first scenario.

Noise level:		0.5		5		25	
#Clusters		2	3	2	3	2	3
PCA		0.95 (0.02)	0.95 (0.02)	0.93 (0.03)	0.92 (0.02)	0.52 (0.03)	0.39 (0.02)
KPCA	gaussian	0.10 (0.04)	0.16 (0.06)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
	linear	0.95 (0.02)	0.95 (0.02)	0.93 (0.03)	0.92 (0.02)	0.52 (0.03)	0.40 (0.02)
	polynomial	0.00 (0.00)	0.30 (0.03)	0.00 (0.00)	0.29 (0.02)	0.00 (0.00)	0.11 (0.02)
RPCA		0.95 (0.02)	0.95 (0.02)	0.92 (0.03)	0.92 (0.02)	0.53 (0.03)	0.40 (0.03)
MDS		0.95 (0.02)	0.94 (0.02)	0.93 (0.03)	0.92 (0.02)	0.52 (0.03)	0.40 (0.02)
ICA		0.93 (0.13)	0.39 (0.03)	0.80 (0.02)	0.34 (0.02)	0.57 (0.03)	0.28 (0.01)
t-SNE		0.16 (0.20)	0.33 (0.14)	0.76 (0.04)	0.75 (0.03)	0.44 (0.03)	0.32 (0.02)

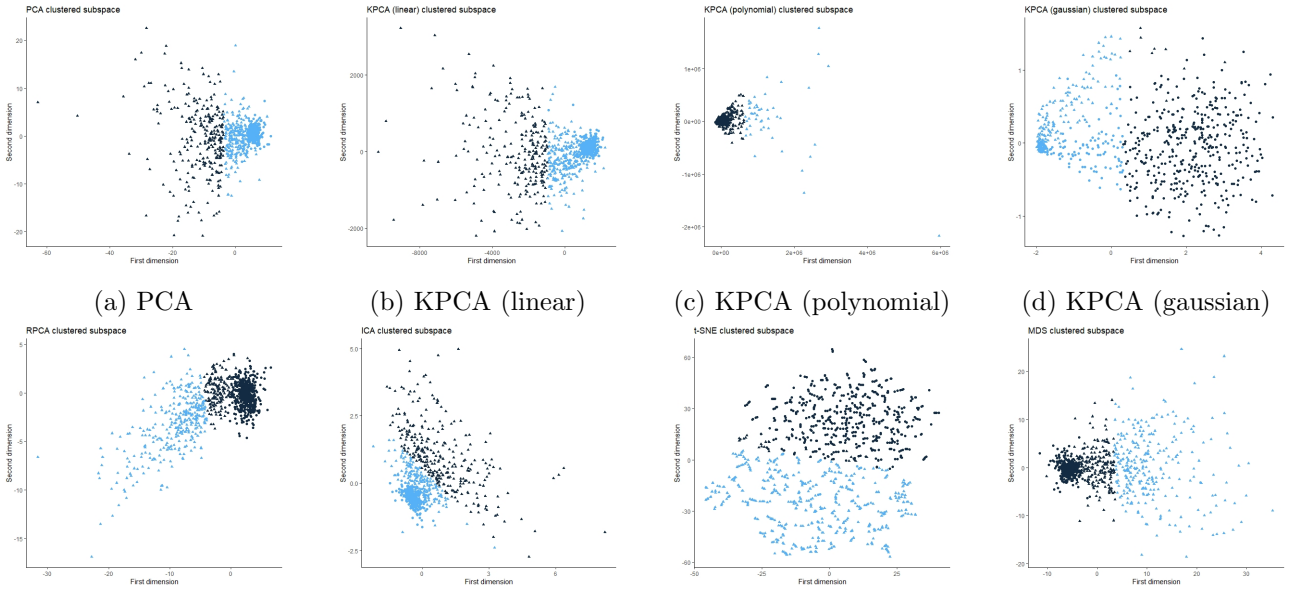
As discussed the effect of the addition of more dimensions in the reduced subspace does not have a significant overall effect on the ARI's of the different models. However for the ICA model this addition of extra dimensions does have an significant negative effect on the performance. This can be caused due to the fact that it is not clear what the optimal amount of independent components is for any ICA analysis this results in a pattern where first the performance increases when adding independent components until a certain point. From that point onward the performance will decline rapidly. This declension of the performance are also accompanied by the increase of the variance. The numerical values of the effect of the addition of extra dimensions in the reduced subspaces for the ICA and all other methods are given in the Appendix. The RKM model also performs significantly well in this first scenario resulting in high ARI values for all reduced dimensions. The results are much alike the results of the PCA method. A small difference is that a small declension of the clustering performance is observed for the RKM technique as the dimensions of the reduced subspace

increases. However this difference is relatively small and further research must be done in order to confirm this observed pattern. Still the more conventional dimension reduction methods are preferred over the RKM technique as the more classic methods are much faster in computing the reduced spaces.

To conclude which technique is preferred in this scenario where the cluster information and DGP is characterised by (linear combinations of) gaussian distributions it can be helpful to look at the running time for each technique which can be found in the Appendix. Looking at both the ARI's and the running time the PCA model seems to be the most preferred in this situation, because it outperforms similar performing methods based on the running time and it remains relatively consistent when adding noise or extra dimensions in the subspace which is not the case for ICA model which has similar running times. So based on the running times and clustering performances it may be concluded that for this scenario the PCA and ICA techniques are the preferred dimension reduction techniques.

Scenario 2

As described in the section 4, the second scenario reflects a situation where the remaining columns are non-linear combinations of the normal distributed columns determining the initial cluster structure of the data. Figure 3 shows the formed clusters for each method for a reduced two-dimensional subspace. It is clearly visible that for this scenario all considered methods perform less compared to the more standard and linear first scenario. Already without adding any noise, conventional (linear) dimension reduction techniques (PCA, KPCA linear and MDS) seem to have problems with clustering the data in a proper way, which is to be expected. In contrast with the first scenario this results in a situation where the clusters cannot be separated well. The ICA model also struggles with clustering the data in the two-dimensional subspace as the two independent components do not seem to be completely independent. It is noticeable that the RPCA strictly outperforms the non-robust alternatives and is able to cluster the data points much better in the reduced subspace compared to the other considered models. This can come due to the fact that the probability of an extreme value in the original input space is relatively high due to the fact that the input space consist mostly of non-linear combinations which can expand this space significantly more compared to a situation where the remaining columns are linear combinations. The robust estimation techniques for the scale and location parameters therefore ensure that the extreme values are more closely clustered in a lower-dimensional subspace which is observed in Figure 3. For this two-dimensional subspace the t-SNE clusters the data points best without any addition of noise.



(a) PCA (b) KPCA (linear) (c) KPCA (polynomial) (d) KPCA (gaussian)

(e) RPCA (f) ICA (g) t-SNE (h) MDS

Figure 3: Clustered two-dimensional sub-spaces for scenario 2 without noise

Table 4 shows the results of the repeated measures ANOVA test for this second scenario. Again similar to the previous scenario this statistical test points out that only the addition of noise to the data has on extremely significant effect on the ARI's. The effect of this addition is numerically shown in Table 5. The numerical values for the effect of the other treatments can be found in the Appendix and will not be analyzed extensively in this thesis. Again as expected the within factor which are the considered models for this ANOVA test are extremely significant. All significant higher order interactions can be found in the Appendix in Table 10. For this scenario there exist only one significant higher order interaction, namely the interaction between the noise and the method. This interaction is clearly visible in Table 5 as some of the used methods have much more difficulties with the clustering after only adding a small amount of noise (t-SNE and RPCA) while the clustering performances of other techniques remain quite constant (PCA, MDS, ICA).

Table 4: Effects for within and between factors in the second scenario

	F-statistic	p-value	effect size
Reduced Dimension	0.035	0.854	0.001
Noise	79.49	0.000	0.597
Clusters	0.003	0.956	0.000
Method	12.07	0.000	0.346

It is clearly visible and plausible that the addition of any noise to the data points has a negative effect on the clustering performances of all methods. In fact the ARI's converge towards zero when adding to much noise which means that the data points are clustered completely at random. Only a small addition of noise already deteriorates the results of most dimension reduction techniques significantly especially for the t-SNE method. This can come due to the fact that this small deterioration can cause a situation where two data points which are originally closely located to each other now will be located

in different 'neighborhoods'. These different neighborhoods may lie relatively far from each other in the reduced subspace which leads to misclassification of the two original data points. This problem can be resolved by tuning the perplexity parameter. However for this study this hyperparameter is not tuned and set equal to 30. The RPCA technique also suffers from the small addition of noise as the ARI is almost halved for both amount of clusters. The results of the RPCA method will converge towards the conventional methods (PCA, MDS and KPCA linear) as the addition of noise increases.

Table 5: ARI and standard errors for the effect of noise for two-dimensional subspaces in the second scenario

Noise:		0.5		5		25	
#Clusters		2	3	2	3	2	3
PCA		0.28 (0.05)	0.36 (0.03)	0.28 (0.04)	0.28 (0.02)	0.01 (0.01)	0.02 (0.01)
KPCA	gaussian	0.63 (0.03)	0.33 (0.01)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
	linear	0.28 (0.05)	0.36 (0.04)	0.25 (0.03)	0.29 (0.01)	0.00 (0.00)	0.02 (0.01)
	polynomial	0.02 (0.01)	0.03 (0.01)	0.01 (0.01)	0.03 (0.01)	0.00 (0.00)	0.00 (0.00)
RPCA		0.47 (0.08)	0.44 (0.03)	0.29 (0.04)	0.29 (0.02)	0.00 (0.01)	0.02 (0.01)
MDS		0.28 (0.05)	0.36 (0.03)	0.27 (0.03)	0.28 (0.02)	0.00 (0.01)	0.02 (0.01)
ICA		0.26 (0.08)	0.27 (0.05)	0.26 (0.04)	0.17 (0.02)	0.00 (0.01)	0.02 (0.01)
t-SNE		0.86 (0.03)	0.85 (0.03)	0.13 (0.07)	0.19 (0.04)	0.00 (0.00)	0.00 (0.00)

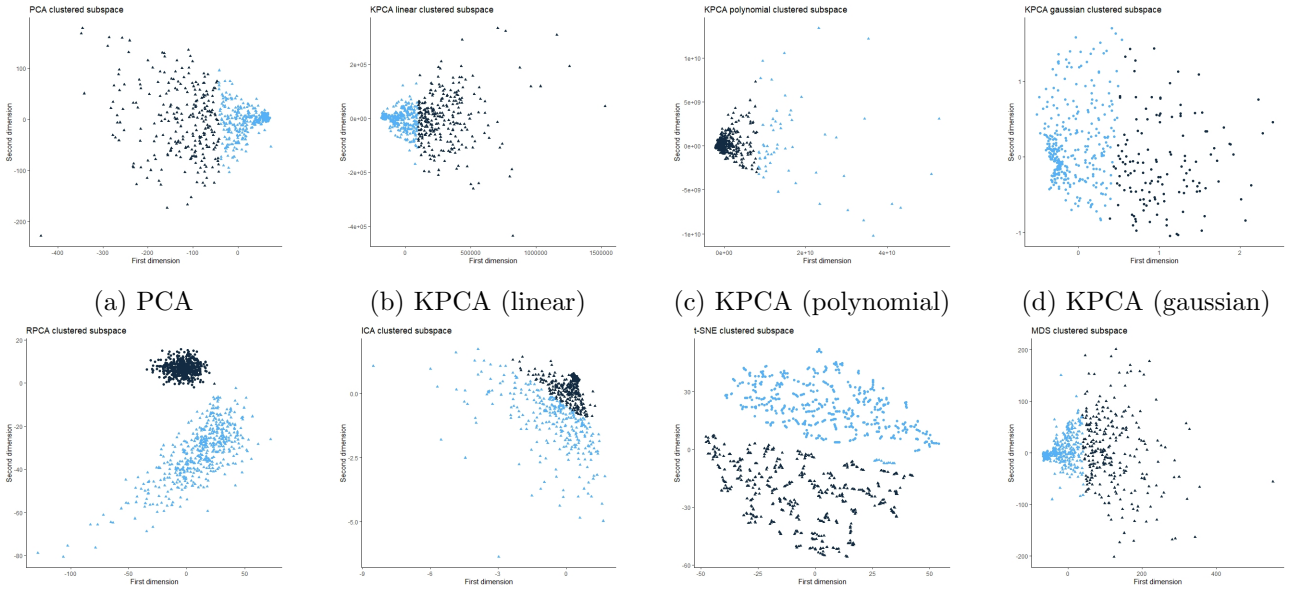
Although the overall effect of the addition of extra dimensions to the reduced subspaces has a insignificant effect, this does not hold for the ICA technique again. Moreover the ICA technique performs strictly better when the dimension of the reduced space is expanded (see Appendix). Again, this can be explained due to the fact that the amount of independent components used for clustering are unknown beforehand. This increasing performance is yet only observed under the treatment of no added noise. If noise is added to the data the ARI's corresponding to the ICA method will converge towards zero. The numerical values of the effects of adding extra dimensions to the subspaces can be found in the Appendix. The RKM method performs also relatively well and is preferred over most of the other dimension reduction techniques. In contrast to the other methods the RKM performance does not decline when the amount of clusters increases and it outperforms every other method if only a bit of noise is added to the data (see Appendix for the results). It also performs significantly better compared to most of the models without any addition of noise. This can potentially come due to the fact that the RKM model aims to preserve the discriminative information in the reduced subspace which is helpful for clustering tasks as is described by Timmerman & Vichi (2010) which is in contrast with the aim of for example PCA and ICA where the variance is maximised or statistical independent components are calculated respectively in the reduced space. Therefore the RKM is a serious candidate for choosing the optimal dimension reduction technique in this scenario. The numerical results of the RKM method can be found in the Appendix. On the other hand the running time of the RKM is much higher compared to any other method.

Based on both the running time and clustering performances for all methods it is not completely clear which technique is preferred for clustering in this scenario. However the RPCA seems to be the most appropriate method as it strictly outperforms the conventional methods in almost all cases.

On the other hand, the ICA seems to perform well when the dimension of the reduced subspace is relatively high, but this does not hold when there are three clusters represented in the data. Also when adding noise the data points the ICA method performs slightly less compared to other fast running methods (PCA, RPCA) even when the reduced dimension is high. The t-SNE and RKM methods perform significantly well without the addition of noise for all reduced subspace dimensions. However the RKM model still performs well when a small amount of noise is added which is not the case for the other models. Therefore when taking all considered model into account the RKM model is the preferred model in this second scenario even if a small amount of noise in the data is expected. When only looking at the tandem clustering methods, the t-SNE technique is preferred for this second scenario. However one could also argue to prefer the RPCA over the t-SNE for this situation.

Scenario 3

The third and last scenario which is taken into consideration for this unsupervised study is the situation where the cluster structure is determined by i (non)-normal distributed variables. The rest of the columns are all (non-)linear combinations of these first columns. Figure 4 shows the formed clusters for a two-dimensional subspace for each method. As showed most of the methods have many difficulties to clearly distinguish the two clusters in the reduced subspace. However the RPCA and t-SNE technique show both promising results as for both methods a clear and correct distinction between the two clusters is observed in the subspace. These high cluster performances is accompanied with a high variability of the performance variable. For the t-SNE method the performance is highly sensitive to small value changes in the data as this may cause a completely different location in the the reduced subspace as a result of the choice for the level of importance of the global or local structure of the original input space which is determined by the perplexity parameter. For the RPCA method the clustering performance is sensitive to which points are used for the robust estimation of the location and scale parameter. Again, just as concluded in the other two scenario's, a misspecified kernel function for the KPCA models leads to bad clustering performances due to the fact the in the first step of these methods the input space is expanded in an incorrect way. Although most of the columns are non-normally distributed the ICA technique also has serious difficulties with recognising the cluster structure in the two-dimensional reduced subspace which is potentially caused by too much loss of information as a result of selecting too few independent components. This will be discussed more extensively later in this Section.



(a) PCA (b) KPCA (linear) (c) KPCA (polynomial) (d) KPCA (gaussian)

(e) RPCA (f) ICA (g) t-SNE (h) MDS

Figure 4: Clustered two-dimensional sub-spaces for scenario 3 without noise

Table 6 shows the results of the repeated measures ANOVA test for the third scenario. Similar to the previous results from the ANOVA tests in the other scenario's the addition of noise to the data has a significant effect on the clustering performance as expected . Again the effect of the amount of dimensions in the reduced subspace and the amount of clusters do not have a significant effect on the ARI. Table 7 shows the numerical values for the effect of both the addition of extra dimensions in the reduced subspace as the addition of noise to the data points. Again the considered method and noise turn out to have a significant impact on the ARI values which correspond with high effect sizes in table 6. All significant higher order interactions can be found in the Appendix in Table 10. It is not surprising that for every higher order interaction the method factor is included, because the chosen method has a significant influence on the clustering performance which can be seen in Table 3.

Table 6: Effects for within and between factors in the third scenario

	F-statistic	p-value	effect size
Reduced Dimension	0.691	0.418	0.003
Noise	83.94	0.000	0.290
Clusters	205.85	0.000	0.500
Method	141.94	0.000	0.891

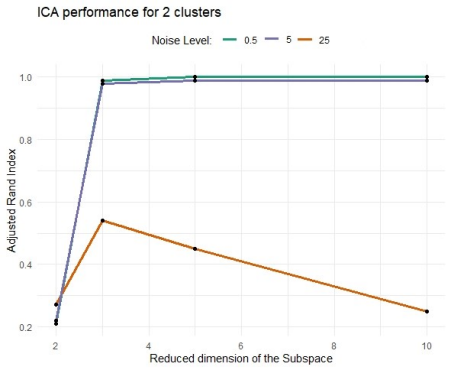
The numerical values confirm the difficulties almost all methods have with distinguishing two clusters without any addition of noise which was also concluded based on the results of Figure 4. As suspected the effect of adding noise to each data point has a negative effect on the clustering performances for all methods. Only the results of the ICA technique are almost not influenced by the addition of noise for both two- as three-clustered data. The t-SNE method outperforms every other technique taken into consideration for all levels of noise and the two different amounts of clusters which can be represent in the data for the two-dimensional subspace, but also this method is sensitive to the

addition of noise as the ARI's immediately decreases when noise is added. It is noticeable that the RPCA performs extremely well for the situation without noise relative to other more conventional dimension reduction techniques. And also when a small amount of noise is added to the data, the RPCA strictly outperforms the other conventional dimension reduction techniques like PCA, KPCA and MDS. However the performance of the RPCA slowly converges towards the performances of these other methods as the addition of noise increases for each data point.

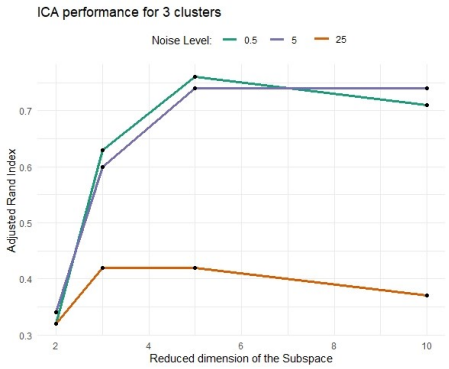
Table 7: ARI and standard errors for the effect of noise for two-dimensional subspaces in the third scenario

Noise level:		0.5		5		25	
#Clusters:		2	3	2	3	2	3
PCA		0.24 (0.05)	0.61 (0.04)	0.23 (0.04)	0.59 (0.04)	0.24 (0.05)	0.57 (0.03)
KPCA	gaussian	0.15 (0.05)	0.40 (0.01)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
	linear	0.23 (0.06)	0.59 (0.03)	0.22 (0.04)	0.60 (0.04)	0.22 (0.04)	0.57 (0.04)
	polynomial	0.01 (0.01)	0.05 (0.03)	0.01 (0.01)	0.04 (0.02)	0.01 (0.01)	0.05 (0.03)
RPCA		0.84 (0.17)	0.59 (0.04)	0.64 (0.30)	0.59 (0.03)	0.29 (0.07)	0.57 (0.03)
MDS		0.23 (0.06)	0.59 (0.04)	0.22 (0.05)	0.59 (0.04)	0.23 (0.06)	0.57 (0.03)
ICA		0.21 (0.08)	0.32 (0.08)	0.22 (0.08)	0.34 (0.08)	0.27 (0.06)	0.32 (0.05)
t-SNE		0.87 (0.26)	0.83 (0.20)	0.86 (0.11)	0.91 (0.10)	0.53 (0.04)	0.69 (0.03)

Until now the analysis is mainly focused on two-dimensional created reduced subspaces as the ANOVA test pointed out that the effect of extra dimensions in the subspaces do not have a significant effect on the clustering performances. And for most of the methods this pattern is confirmed by looking at the numerical values of the effect of adding extra dimensions in the corresponding subspace which can be found in the Appendix. However as mentioned before for the ICA technique the performance significantly increases as the amount of dimensions in the subspace increases. Figure 6 graphically shows this increase in performance when extra dimensions are added for each different noise level. Note that the ARI-axis are not scaled the same in the two figures. The maximum ARI for two clusters equals one while the maximum ARI for three clusters is around 0.75. It is observed that when this added noise becomes too big also the performance of the ICA model is influenced resulting in lower ARI values which is graphically shown in Figure 6 with the orange lines. This declension is potentially caused due to the fact that the columns are becoming too Gaussian when too much normal-distributed noise is added to each data point which will result in a situation where all input columns tend towards a Gaussian distribution which violates one of the assumption within the ICA framework. This same reasoning also holds for the other two scenario's taken into consideration for this study. However in this third scenarios most original generated columns are non-normally distributed. Therefore relatively much noise is allowed to be added as the input columns stay non-normally distributed after this addition. This outperformance of the ICA method is not seen for a two-dimensional subspace as apparently too much information is lost when only two independent components are used for clustering. On the other hand selecting too many independent components also lead to a non-optimal situation which may result in overfitted data problems meaning that one independent component may consist of only a single peak at multiple locations in the reduced subspace.



(a) Two Clusters in the Data



(b) Three Clusters in the Data

Figure 5: Effect of adding reduced dimensions for ICA. The different colours represent the different levels of Noise. Green corresponds with a noise level of 0.5, blue with a level of 5 and orange a noise level of 25. *Note: As described in the text above, the ARI-axis are not scaled the same for the two figures.*

It is clear that the more conventional dimension reduction techniques are not suitable for this more complex scenario where the columns are mostly non-normally distributed and non-linear combinations of these generated columns. Except for the RPCA method which performs well without any addition of noise. Based on the running time and performances of the considered models and there behaviour on the different treatments the ICA, RKM and t-SNE models stand out in this third scenario. And given that the ICA starts to perform significantly better when adding dimensions to the reduced space while the performance of the t-SNE methods remains stable it can be concluded that the ICA method is the most suitable method for this scenario. Also the RKM method shows promising and comparable results to the ICA method in this third scenario as the corresponding ARI values are extremely high even after a small addition of noise. The main difference regarding the performance compared to the ICA technique is that the results are stable under the addition of extra dimensions in the reduced spaces. However the running time of the RKM method is much higher compared to the running time of the ICA and therefore the ICA is preferred in this scenario.

5.2 Supervised Study Results

Breast Cancer Data Results

The results of the binary classification problem of the Breast Cancer data based on a logit classifier can be found in Table 8. The values which are based on the SVM classifier can be found in the Appendix. The benchmark model corresponds with a situation where the original input columns are used in the respective classifier.

Table 8: Classification performances based on a logistic regression classifier

Performance Measure:		Accuracy			F1 Score			AUC		
Reduced Dimension:		2	5	10	2	5	10	2	5	10
Benchmark		**0.94**			**0.87**			**1.00**		
SPCA	linear	0.91	0.91	0.91	0.83	0.83	0.83	0.98	0.98	0.98
	delta	0.91	0.91	0.91	0.83	0.83	0.83	0.98	0.98	0.98
SKPCA	gaussian	0.77	0.77	0.77	0.29	0.29	0.29	0.93	0.93	0.93
	linear	0.91	0.91	0.91	0.83	0.83	0.83	0.98	0.98	0.98
	polynomial	0.91	0.91	0.91	0.83	0.83	0.83	0.98	0.98	0.98
ICA-FX		0.88	0.88	0.93	0.79	0.79	0.86	0.96	0.99	1.00
t-SNE		0.92	0.91	0.91	0.85	0.83	0.84	0.98	0.98	0.98

For all reduced feature spaces there are promising results for multiple dimension reduction techniques. On the other hand none of the used dimensional reduction techniques leads to a better performance measure compared to the benchmark model. However this is plausible as the original dimension of the input features are not of an extremely high order and the performance of this benchmark mark model is already quite high. Therefore this classification problem can be considered as easy where the reduction of the dimensions does not add any additional value to the performance of the considered classifiers. The difference between the logit and the SVM model also seems to be not that big except for all supervised kernel PCA (SKPCA) models. The numerical results of the SVM classifier can be found in the Appendix.

One of the reasons for the weak performances of KPCA models can be a misspecified kernel function which may lead to an incorrect expansion of the feature space. As the data generating process of this particular dataset is unknown it is not possible to conclude whether a specific kernel function is correct or not. Now the cause of the weak performances of the SKPCA models taken into consideration for this study will be discussed for this particular chosen dataset. The reason the SKPCA models did not perform as hoped is mainly because of the explosion or implosion of multiple matrices in the estimation procedure. For the gaussian SKPCA the kernel matrix $\mathbf{K}$ converged towards a zero-matrix for large amount of values of the standard deviation σ which is determined by the user. It is therefore plausible that the gaussian SKPCA model results in the same performance measures for all reduced dimensions as the matrix $\mathbf{K}$ does not depend on any reduced dimension. The imploded $\mathbf{K}$ matrix ensures that almost all predictions for the logit and SVM models are zero as a results of that the eigenvectors which create the reduced space are determined by the matrix $\mathbf{Q}$ which is highly influenced by $\mathbf{K}$. See the Methodology Section for the mathematical definition of these matrices. For the obtained results a standard deviation of 0.05 is used as this is a standard value of the *kernlab* package in R. The same applies for the other two SKPCA models as the kernel matrix $\mathbf{K}$ is a highly influenced by the used scale parameter. If this scale parameter becomes too big the kernel matrix explodes and this will lead to a situation where none of the other matrices can be estimated. On the other hand when the scale parameter becomes too small, the kernel matrix $\mathbf{K}$ converges again towards a zero matrix which was also the case in the Gaussian SKPCA model. With the same reasoning as before, it is therefore plausible that the results for all reduced dimensions are the same as the estimation of the

kernel matrix does not depend on any reduced dimension. For the obtained results in this study the scale parameters are set equal to 10^{-8}. The other used models perform much better and sometimes do come close to the performance of the benchmark model. For the relatively higher reduced dimensions the ICA-FX model outperforms any other used model and the performance measures almost equal the measures of the benchmark model. For the lower dimensional models t-SNE and SPCA are slightly better and almost identical.

Alzheimer's Disease Handwriting Data Results

Table 9 shows the numerical results for the Alzheimer's handwriting data when a logistic classifier is used. The results of the SVM classifier can be found in the Appendix. Since the dimension of this data set is much higher compared to the previous empirical study the benchmark values are significantly lower as expected which makes this a much harder classification problem.

Table 9: Alzheimer's Disease Handwriting Data Performance with logistic regression

Performance Measure		Accuracy				F1 Score				AUC			
Reduced Dimension		2	5	10	50	2	5	10	50	2	5	10	50
Benchmark		**0.67**				**0.65**				**0.85**			
SPCA	linear	0.53	0.61	0.80	0.65	0.68	0.69	0.76	0.67	0.48	0.72	0.84	0.81
	delta	0.51	0.71	0.76	0.92	0.63	0.73	0.80	0.91	0.66	0.82	0.79	0.92
SKPCA	gaussian	0.53	0.57	0.47	0.51	0.64	0.60	0.69	0.66	0.50	0.50	0.50	0.50
	linear	0.45	0.47	0.39	0.61	0.00	0.69	0.00	0.56	0.50	0.50	0.50	0.50
	polynomial	0.57	0.47	0.49	0.49	0.60	0.69	0.68	0.65	0.50	0.50	0.50	0.52
ICA-FX		0.57	0.61	0.59	0.78	0.59	0.63	0.63	0.78	0.61	0.64	0.59	0.88
t-SNE		0.83	0.79	0.75	0.67	0.75	0.74	0.73	0.64	0.83	0.95	0.88	0.79

Again, all the classical dimension reduction techniques perform quite similar. However all models do have much more difficulties with the classification task leading to lower performance measure values. For most of the classical techniques the accuracy, F_1-score and AUC tend to increase when the dimension of the reduced space increases. Nevertheless for most of these classical models the performance is significantly lower than the benchmark where no dimensional reduction technique is performed. Just as was observed with the Breast Cancer Data, multiple instabilities are observed for the gaussian and polynomial SKPCA models. The performance of both models are highly influenced by the respective hyperparameters of the models. It is noticeable that for the SPCA model with a delta kernel the performance is extremely high especially for a high reduced dimensional subspace. However this out-performance is not observed with the SVM classifier. Therefore an overall conclusion of this out-performance cannot be made.

The more complex models (t-SNE and ICA) perform much better and are able to outperform the benchmark models. For the ICA-FX, the performance increases when the reduced dimension increases. On the other hand, for the t-SNE model this is not the case. This model already has promising results for a reduced dimension of two. However the performance (slightly) decreases overall when an increase

in the reduced dimension is observed. Overall, the t-SNE model can be considered as the best model which is taken into consideration for harder classification problem. This conclusion is also in line with the previous empirical study about the breast cancer data as the t-SNE model there also outperformed all other models taken into consideration.

6 Conclusion

This thesis tried to give multiple insights into the behavior of dimension reduction techniques including an extensive analysis of the reaction to various effects and different DGPs in an unsupervised learning setting. To investigate this, a simulation study is performed in order to compare more sophisticated dimension reduction methods like Independent Component Analysis (ICA) and t-Distributed Stochastic Neighbor Embedding (t-SNE) with more standard reduction which are closely linked to Principal Component Analysis (PCA). For the simulation study three different data generated processes (DGPs) are taken into consideration and for each DGP it is analyzed how every considered dimension reduction method reacts on the addition of (1) Gaussian noise, (2) extra dimensions in the compressed subpsaces and (3) the number of clusters which are represent in the data. The DGPs all differ in the amount of linearity of the spaces and Gaussianity of the input columns as these characteristics seemed to be crucial regarding the tandem clustering performances of the considered models. As all performances of the considered dimension reduction methods are based on tandem clustering where first the inputspace is reduced and thereafter a clustering algorithm is used in order to cluster the data points in the reduced space it is of interest how these performances differ compared to a method which simultaneously reduced its inputspace and clusters the data points in the reduced space like reduced k-means (RKM).

6.1 Summarised Results

The first scenario reflects a situation where the cluster structure is based on all normally distributed variables and all other columns are all linear combinations of these Gaussian variables. The more conventional dimension reduction methods (PCA, KPCA linear, RPCA and MDS) all perform well and are able to extract the cluster structure in already a two-dimensional subspace. However the extraction of this structure becomes harder as Gaussian noise is added to each data point. The more sophisticated techniques perform strictly less in particular under the treatment where no noise is added and when three different clusters are represent in the data. For all methods in this scenario it holds that the addition of extra dimension in the compressed subspace has either no effect or for the ICA method a negative effect on the clustering performance. The second scenario reflects a situation where again the cluster structure is determined by Gaussian variables. However all other input columns are non-linear combinations of the normally distributed columns. This introduction of non-linearity and non-Gaussianity in the inputspace leads to a significant declension in the clustering performance for all methods. Again the addition of noise has a strong negative effect on the performances where the ARIs converge towards zero relatively fast. Without any noise (or little noise) the more sophisticated models (ICA, t-SNE and RKM) do outperform every other considered model and it can be concluded that those models are able to extract the cluster structure out of there unique compressed subspace. The third scenario reflects a situation where the cluster structure is also partially based on

non-normally distributed variables which introduces more non-linearity and non-Gaussianity in the inputspace. Again the more sophisticated models are better in extracting the cluster structure in the reduced spaces. It is interesting to analyze the ICA clustering performances as this model behaves significantly different from every other tandem clustering method, because it is clearly observable that for the ICA method the addition of extra dimension in the reduced space can both have a significant positive and negative effect on the clustering performance. This effect is caused as it is not clear how many independent components are needed for identifying the cluster structure.

Besides the unsupervised learning simulation study, all methods are either generalized or adjusted such that they become suitable for supervised learning problems in order to analyze their classification performances. In this thesis Breast Cancer Data and Alzheimer's disease data from the UCI Machine Learning Database is used for the comparison of the different methods. It can be concluded that there exists almost no difference in each classifier for relatively easy classification problems. The different compressed spaces of the different techniques do all differ and most of the methods ensure a high classification performance. However for this 'easy' problem none of the considered models do outperform the classification prediction which are based on the original input features. When we compare the models with each other for a much harder classification problem, larger differences between the models taken into consideration are observed. The more classical methods overall have much more difficulties with classifying the binary response. The non-parametrical t-SNE model is able to outperform the benchmark and all other models significantly which makes the t-SNE model the best suitable model.

6.2 Discussion

For further research it may be interesting to dive deeper into some technical details with the aim to exploit the behavior of certain model even better under different situations. It can for example be interesting to look at the behavior of other types of noise which should effect the performances of multiple considered models and especially the ICA if the non-Gaussianity of the noise is of high order. Durieux et al. (2022) showed for example that ICA-like models can handle autoregressive noise structure to a much further extend compared to gaussian noise. Other tandem clustering procedures could also be interesting lo look into with the aim to understand the relations between these methods and the considered methods in this thesis. As discussed earlier Archimbaud et al. (2018) uses Invariant Coordinate Selection (ICS) to remove outliers in higly dimensional data. Moreover this ICS procedure also shows promising results in the field of tandem clustering and could therefore be of interet to dive deeper in the behavior of this method for different scenarios Archimbaud et al. (2022). Even though it is tried to explain (mathematically) the connection between the t-SNE method and more conventional methods, this method still differs significantly from the other considered models. However for this research the t-SNE is included as for many medical applications the t-SNE method is successfully used and is often preferred over other dimensional reduction techniques because it has always performed decently Kobak & Berens (2019). However there do exist similar methods which could be looked into in order to understand the behavior of the t-SNE methods better under different situations. Uniform manifold approximation and projection (UMAP) for example is such a method where the global structure is preserved better and more consistent outcomes for different runs are observed Kobak &

Linderman (2021). Preserving the global structure of the data is also a desirable aspect for tandem-like clustering procedures which is not discussed extensively in this thesis as none of the considered models focus much on this aspect. Also the influence of the perplexity hyperparameter is something that could be looked into in further research as it is expected that the clustering performances are significantly influenced by the choice of this parameter. At last further research could dive deeper into the behavior of the clustering performances for a significantly higher input dimensions. It is expected that some considered models will perform relatively better or worse when this dimension becomes significantly large. It would then also be possible to increase the reduced dimension to a significantly higher dimension which can result in different choices for optimal or preferred techniques for multiple scenarios.

References

Anowar, F., Sadaoui, S., & Selim, B. (2021). Conceptual and empirical comparison of dimensionality reduction algorithms (pca, kpca, lda, mds, svd, lle, isomap, le, ica, t-sne). *Computer Science Review*, *40*. doi: 10.1016/j.cosrev.2021.100378

Archimbaud, A., Alfons, A., Nordhausen, K., & Ruiz-Gazen, A. (2022). Tandem clustering with invariant coordinate selection.

Archimbaud, A., Nordhausen, K., & Ruiz-Gazen, A. (2018). Ics for multivariate outlier detection with application to quality control. *Computational Statistics Data Analysis*, *128*, 184-199. doi: 10.1016/j.csda.2018.06.011

Aremu, O., Hyland-Wood, D., & McAree, P. (2020). A machine learning approach to circumventing the curse of dimensionality in discontinuous time series machine data. *Elsevier*, *195*. doi: 10.1016/j.ress.2019.106706

Banks, D. L., & Fienberg, S. E. (2003). Data mining, statistics. *Encyclopedia of Physical Science and Technology (Third Edition)*, 247-261. doi: 10.1016/B0-12-227410-5/00164-2

Barshan, E., Ghodsi, A., Azimifar, Z., & Jahromi, M. Z. (2011). Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds. *Pattern Recognition*, *44*(7), 1357-1371. doi: 10.1016/j.patcog.2010.12.015

Beckmann, C. (2012). Modelling with independent components. *Elsevier*, *62*(2), 891-901. doi: 10.1016/j.neuroimage.2012.02.020

Bellmann, R. (1961). *Adaptive control processes : a guided tour*. doi: 10.1515/9781400874668

Buja, A., Swayne, D., Littman, M., Dean, N., Hofmann, H., & Chen, L. (2012). Data visualization with multidimensional scaling. *Journal of Computational and Graphical Statistics*, *17*, 444-472. doi: 10.1198/106186008X318440

Calhoun, V. D., & Adali, T. (2012). Multisubject independent component analysis of fmri: A decade of intrinsic networks, default mode, and neurodiagnostic discovery. *IEEE Reviews in Biomedical Engineering*, *5*, 60-73. doi: 10.1109/RBME.2012.2211076

Chang, W.-C. (1983). On using principal components before separating a mixture of two multivariate normal distributions. *Journal of the Royal Statistics Society: Applied Statistics*, *32*(3), 267-275. doi: 10.2307/2347949

Chu, D., & Herskowitz. (1998). The transcriptional program of sporulation in budding yeast. , *282*(5389), 699-705. doi: 10.1126/science.282.5389.699

Comon, P. (1994). Independent component analysis, a new concept. *Signal Processing*, *36*(3), 287-314. doi: 10.1016/0165-1684(94)90029-9

Davison, M. (1983). Introduction to multidimensional scaling and its applications. *Sage Journals*, *7*(4). doi: 10.1177/014662168300700401

Devassy, B., George, S., & Nussbaum, P. (2020). Unsupervised clustering of hyperspectral paper data using t-sne. *Journal of Imaging*, *6*(5). doi: 10.3390/jimaging6050029

Dolnicar. (2003). Using cluster analysis for market segmentation - typical misconceptions, established methodological weaknesses and some recommendations for improvement. *Australasian Journal of Market Research*, *11*(2), 5-12.

Durieux, J., Rombouts, S., de Vos, F., Koini, M., & Wilderjans, T. (2022). Clusterwise independent component analysis (c-ica): Using fmri resting state networks to cluster subjects and find neuro-functional subtypes. *Journal of Neuroscience Methods*, *382*. doi: 10.1016/j.jneumeth.2022.109718

Durieux, J., & Wilderjans, T. (2019). Partitioning subject bason on high-dimensional fmri data: comparison of several clustering methods and studying the influence of ica data reduction in big data. *Behaviormetrika*, *46*, 271-311.

Filzmoser, P., Hron, K., & Reimann, C. (2009). Principal component analysis for compositional data with outliers. *Environmetrics*, *20*, 621–632. doi: 10.1002/env.966

Fontanella, F. (2022). *DARWIN.* UCI Machine Learning Repository. (DOI: https://doi.org/10.24432/C55D0K)

Ghojogh, B., & Crowley, M. (2022). Unsupervised and supervised principal component analysis: Tutorial.

Gretton, A., Herbrich, R., Smola, O., Alexander Bousquet, & Scholkopf, B. (2005). Kernel methods for measuring independence. *Journal of Machine Learning Research*, *6*, 2075–2129.

Hajderanj, L., Weheliye, I., & Chen, D. (2019). A new supervised t-sne with dissimilarity measure for effective data visualization and classification. *International Conference of Software and Information Engineering*, 232-236. doi: 10.1145/3328833.3328853

Hartigan, J., & Wong, M. (1979). A k-means clustering algorithm. *journal of royal statistical society (applied statistics)*, *28*(1), 100-108. doi: 10.2307/2346830

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning data mining, inference, and prediction* (Vol. 2). Springer.

Hofmann, T., Scholkopf, B., & Smola, A. (2008). The annals of statistics. , 1171-1220.

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

Huber, P. (1985). Projection pursuit. *The annals of Statistics*, *13*, 435-475.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, *2*, 193-218.

Hyvarinen, A. (1999). Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, *10*(3), 626-634. doi: 10.1109/72.761722

Hyvarinen, A., & Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural Networks*, 411–430.

Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *374*. doi: 10.1098/rsta.2015.0202

Kaiser, H. (1958). The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, *23*, 187-200.

Kaur, A., & Datta, A. (2019). Detecting and ranking outliers in high-dimensional data. *International Journal of Advances in Engineering Sciences and Applied Mathematics*, *11*, 75–87. doi: 10.1007/s12572-018-0240-y

Kenkel, N., & Orloci, L. (1986). Applying metric and nonmetric multidimensional scaling to ecological studies: Some new results. *Ecology*, *67*(4), 919-928.

Kobak, D., & Berens, P. (2019). The art of using t-sne for single-cell transcriptomics. *Nature Communications*, *10*. doi: 10.1038/s41467-019-13056-x

Kobak, D., & Linderman, G. (2021). Initialization is critical for preserving global data structure in both t-sne as umap. *Nature Biotechnology*, *39*, 156-157.

Kodak, D., & Berens, P. (2019). The art of using t-sne for single-cell transcriptomics. *Nature Communications*, *10*. doi: 10.1038/s41467-019-13056-x

Kwak, N., & Choi, C.-H. (2003). Feature extraction based on ica for binary classification problems. *IEEE Transactions on Knowledge and Data Engineering*, *15*(6), 1374-1388. doi: 10.1109/TKDE.2003.1245279.

Kwak, N., & Kim, C. (2006). Dimensionality reduction based on ica for regression problems. *Artifical Neural Networks*, 1-10.

Linderman, G. C., Rachh, M., Hoskins, J. G., Steinerberger, S., & Kluger, Y. (2017). Efficient algorithms for t-distributed stochastic neighborhood embedding. *ArXiv*, *1712.09005*.

Liu, W., Payne, S., Ma, S., & Fenyo, D. (2019). Extracting pathway-level signatures from proteogenomic data in breast cancer using independent component analysis.
doi: 10.1074/mcp.TIR119.001442

Lopuhaa, H., & Rousseeuw, P. (1991). Breakdown points of affine equivariant estimators of multivariate location and covariance matrices. *The Annals of Statistics*, *19*(1), 229-248. doi: 10.1214/aos/1176347978

Ma, S., & Dai, Y. (2011). Principal component analysis based methods in bioinformatics studies. *Briefings in Bioinformatics*, *12*, 714–722. doi: 10.1093/bib/bbq090

Malhi, A., & Gao, R. X. (2004). Pca-based feature selection scheme for machine defect classification. *IEEE Transactions on Intelligent Transportation Systems*, *53*, 1517 - 1525. doi: 10.1109/TIM.2004.834070

Meinecke, F., Harmeling, S., & Müller, K. (2004). Robust ica for super-gaussian sources. , *3195*, 217–224. doi: 10.1007/978-3-540-30110-3-28

Milligan, G. W., Soon, S. C., & Sokol, L. M. (1983). The effect of cluster size, dimensionality, and the number of clusters on recovery of true cluster structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *5*(1), 40-47. doi: 10.1109/TPAMI.1983.4767342

Pearson, K. (1901). 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. doi: 10.1080/14786440109462720

Ray, P., Reddy, S., & Banerjee, T. (2021). Various dimension reduction techniques for high dimensional data analysis: a review. , *54*, 3473–3515. doi: 10.1007/s10462-020-09928-0

Stanford University. (2023). *Unsupervised feature learning and deep learning.* Retrieved from http://ufldl.stanford.edu/tutorial/unsupervised/ICA/ (Stanford University, UFLDL Tutorial)

Terada, Y. (2014). Strong consistency of reduced k-means clustering. , 1-3.

Timmerman, K., Ceulemans, & Vichi. (2010). Factorial and reduced k-means reconsidered. *Computational Statistics and Data Analysis*, *54*, 1858-1871.

Tripathy, A., & Ghela, S. (2022). *Unsupervised learning approaches for dimensionality reduction and data visualization* (T. . Francis, Ed.). CRC Press.

van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, *9*, 2579-2605.

Vichi, M., & Kiers, H. (2001). Factorial k-means analysis for two-way data. *Computational Statistics Data Analysis*, *37*, 49-64. doi: 10.1016/S0167-9473(00)00064-5

Westad, & Kermit. (2009). Comprehensive chemometrics chemical and biochemical data analysis. *Elsevier*, *2*, 227-248. doi: 10.1016/B978-044452701-1.00045-4

Williams, C. (2002). On a connection between kernel pca and metric multidimensional scaling. *Machine Learning*, *46*, 11-19.

Wolberg, W., Street, W., & Mangasarian, O. (1995). *Breast Cancer Wisconsin (Diagnostic).* UCI Machine Learning Repository. (DOI: 10.24432/C5DW2B)

Xu, X., Xie, Z., Yang, Z., Li, D., & Xu, X. (2020). A t-sne based classification approach to compositional microbiome data. *Sec. Computational Genomics*, *11*.

Yang, J., Gao, X., Zhang, D., & Yang, J. (2005). Kernel ica: An alternative formulation and its application to face recognition. *Pattern Recognition*, *38*(10), 1784-1787. doi: 10.1016/j.patcog.2005.01.023

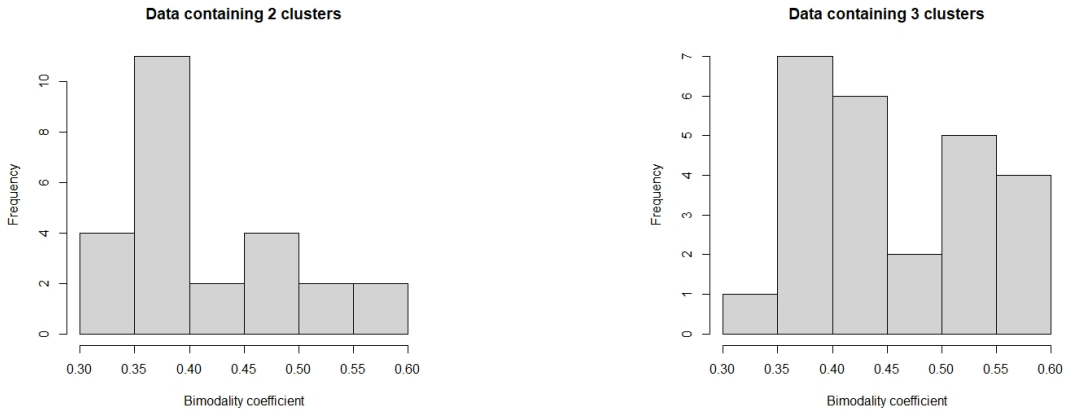
Yeung, K., & Ruzzo, W. (2001). Philosophical transactions of the royal society a: Mathematical, physical and engineering sciences. *Bioinformatics*, *17*(9), 763-774. doi: 10.1093/bioinformatics/17.9.763

Zhao, Q., Meng, D., Xu, Z., Zuo, W., & Zhang, L. (2014). Robust principal component analysis with complex noise. In E. P. Xing & T. Jebara (Eds.), *Proceedings of the 31st international conference on machine learning* (Vol. 32, pp. 55–63). PMLR.

7 Appendix

7.1 Extra Results Unsupervised Study

7.1.1 Bimodality Graphs



(a) Two Clusters in the Data

(b) Three Clusters in the Data

Figure 6: Historgram of bimodality of the generated columns for two- and three-clustered data

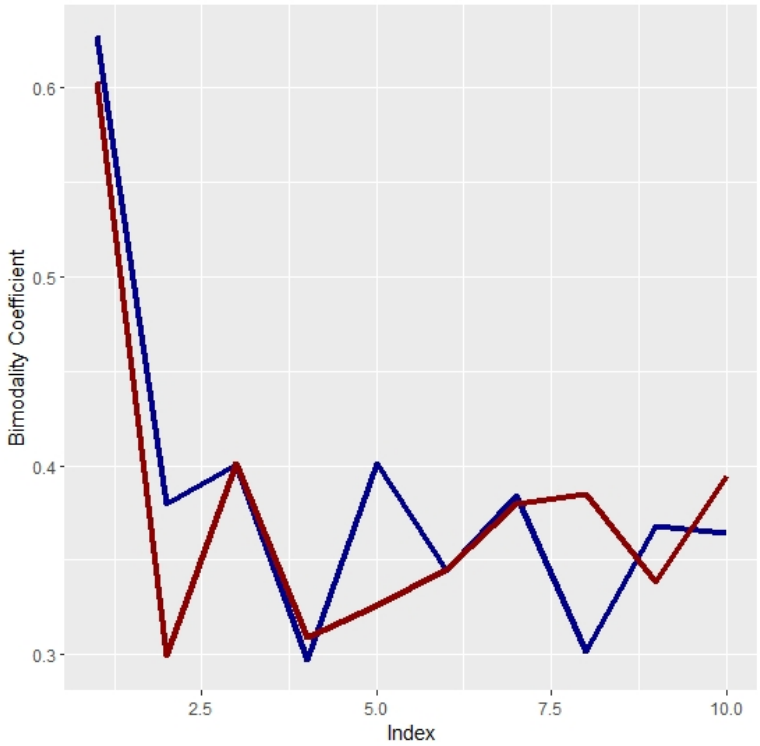


Figure 7: Bimodality of the independent components for two- (blue) and three-clustered (red) data

Table 10: Significant Higher order interaction of Anova test per scenarios

		F statistic	p-value	effect size
Scenario 1	epsilon × method	26.07	0.000	0.571
	clusters × method	22.59	0.000	0.536
	epsilon × cluster × method	4.01	0.006	0.170
Scenario 2	epsilon × method	8.09	0.000	0.262
Scenario 3	red dim × method	4.10	0.001	0.191
	epsilon × method	8.16	0.00	0.320
	clusters × method	19.36	0.000	0.527
	red dim × epsilon × method	2.84	0.009	0.141
	epsilon × clusters × method	2.36	0.027	0.120

Table 11: Running time (minutes) per method for each scenario

Time in minutes		Scenario 1	Scenario 2	Scenario 3*
PCA		0.08	0.07	0.68
KPCA	gaussian	2.94	2.90	2.08
	linear	2.40	2.29	1.77
	polynomial	2.37	2.40	1.78
RPCA		0.55	0.51	0.90
MDS		1.00	1.04	1.15
ICA		0.09	0.09	0.67
t-SNE		1.59	1.93	1.40

Table 12: Benchmark ARIs for unsupervised study

Scenario	Noise	#Clusters	ARI	Scenario	Noise	#Clusters	ARI	Scenario	Noise	#Clusters	ARI
1	0.5	2	0.95 (0.01)	2	0.5	2	0.29 (0.06)	3	0.5	2	0.23 (0.05)
1	5	2	0.93 (0.02)	2	5	2	0.27 (0.04)	3	5	2	0.23 (0.05)
1	25	2	0.52 (0.03)	2	25	2	0.00 (0.00)	3	25	2	0.24 (0.06)
1	0.5	3	0.95 (0.01)	2	0.5	3	0.38 (0.04)	3	0.5	3	0.60 (0.05)
1	5	3	0.92 (0.02)	2	5	3	0.29 (0.02)	3	5	3	0.06 (0.06)
1	25	3	0.39 (0.03)	2	25	3	0.01 (0.01)	3	25	3	0.57 (0.03)

Table 13: ARI values for RKM in unsupervised study

(a) Scenario 1

Noise:	0.5		5		25	
Clusters	2	3	2	3	2	3
2 dim.	0.95	0.95	0.86	0.86	0.50	0.28
3 dim	0.94	0.94	0.91	0.87	0.45	0.33
5 dim	0.95	0.94	0.93	0.92	0.49	0.36
10 dim	0.96	0.95	0.89	0.87	0.45	0.29

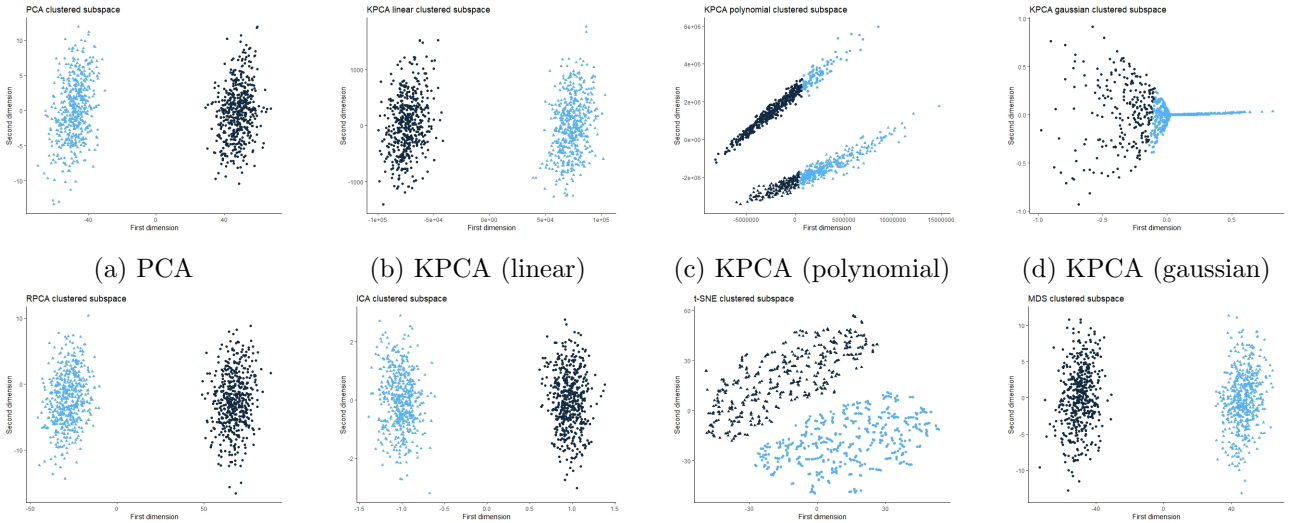
(b) Scenario 2

Noise:	0.5		5		25	
Cluster	2	3	2	3	2	3
2 dim.	0.76	0.80	0.30	0.30	0.00	0.02
3 dim	0.76	0.78	0.28	0.32	0.00	0.02
5 dim	0.76	0.82	0.30	0.37	0.02	0.01
10 dim	0.76	0.81	0.28	0.31	0.00	0.01

(c) Scenario 3

Noise:	0.5		5		25	
Clusters	2	3	2	3	2	3
2 dim.	0.98	0.95	0.89	0.87	0.57	0.55
3 dim	0.97	0.95	0.91	0.86	0.48	0.51
5 dim	0.97	0.94	0.91	0.85	0.46	0.56
10 dim	0.97	0.93	0.88	0.85	0.47	0.57

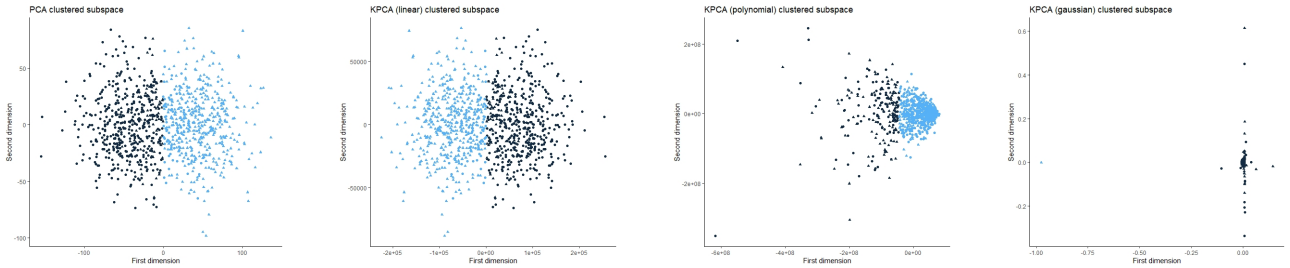
Scenario 1 extra results



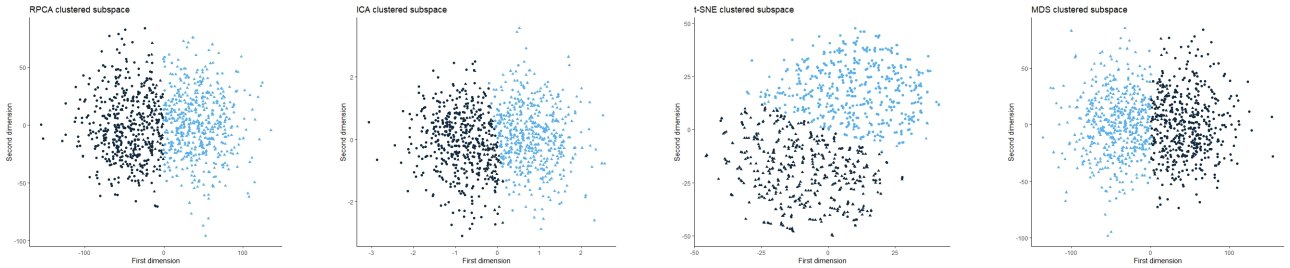
(a) PCA (b) KPCA (linear) (c) KPCA (polynomial) (d) KPCA (gaussian)

(e) RPCA (f) ICA (g) t-SNE (h) MDS

Figure 8: Formed clusters when initial clusters are located further from each other in scenario 1



(a) PCA (b) KPCA (linear) (c) KPCA (polynomial) (d) KPCA (gaussian)



(e) RPCA (f) ICA (g) t-SNE (h) MDS

Figure 9: Formed clusters scenario 1 under treatment of much noise

Table 14: ARI values and standard errors for 2 clustered data in scenario 1

Noise	Red. Dim.	PCA	KPCA			RPCA	MDS	ICA	t-SNE
			Gaussian	Polynomial	Linear				
0.5	2	0.95 (0.02)	0.10 (0.04)	0.00 (0.00)	0.95 (0.02)	0.95 (0.02)	0.95 (0.02)	0.93 (0.13)	0.16 (0.20)
	3	0.95 (0.02)	0.09 (0.03)	0.00 (0.00)	0.94 (0.02)	0.96 (0.02)	0.95 (0.02)	0.95 (0.01)	0.27 (0.28)
	5	0.95 (0.02)	0.10 (0.04)	0.00 (0.00)	0.94 (0.02)	0.95 (0.02)	0.95 (0.02)	0.93 (0.13)	0.33 (0.28)
	10	0.96 (0.02)	0.11 (0.050	0.00 (0.00)	0.95 (0.03)	0.95 (0.03)	0.94 (0.03)	0.84 (0.29)	0.31 (0.28)
5	2	0.93 (0.03)	0.00 (0.00)	0.00 (0.00)	0.93 (0.03)	0.94 (0.03)	0.94 (0.03)	0.80 (0.02)	0.76 (0.04)
	3	0.94 (0.03)	0.00 (0.00)	0.00 (0.00)	0.93 (0.03)	0.93 (0.03)	0.93 (0.02)	0.80 (0.03)	0.76 (0.05)
	5	0.93 (0.02)	0.00 (0.00)	0.00 (0.000	0.93 (0.03)	0.93 (0.02)	0.93 (0.03)	0.77 (0.15)	0.75 (0.04)
	10	0.94 (0.03)	0.00 (0.00)	0.00 (0.00)	0.93 (0.01)	0.93 (0.03)	0.92 (0.02)	0.63 (0.31)	0.76 (0.04)
25	2	0.52 (0.03)	0.00 (0.00)	0.00 (0.00)	0.52 (0.03)	0.52 (0.03)	0.52 (0.03)	0.57 (0.06)	0.44 (0.03)
	3	0.52 (0.03)	0.00 (0.00)	0.00 (0.00)	0.52 (0.03)	0.52 (0.03)	0.52 (0.03)	0.52 (0.06)	0.44 (0.03)
	5	0.52 (0.03)	0.00 (0.00)	0.00 (0.00)	0.52 (0.03)	0.52 (0.03)	0.52 (0.03)	0.52 (0.14)	0.44 (0.03)
	10	0.52 (0.03)	0.00 (0.00)	0.00 (0.00)	0.52 (0.03)	0.52 (0.03)	0.52 (0.03)	0.16 (0.20)	0.44 (0.03)

Table 15: ARI values and standard errors for 3 clustered data in scenario 1

Noise	Red. Dim.	PCA	KPCA			RPCA	MDS	ICA	t-SNE
			Gaussian	Polynomial	Linear				
0.5	2	0.95 (0.02)	0.16 (0.06)	0.30 (0.03)	0.95 (0.02)	0.95 (0.02)	0.95 (0.02)	0.39 (0.03)	0.33 (0.14)
	3	0.95 (0.02)	0.14 (0.05)	0.30 (0.03)	0.95 (0.020	0.95 (0.02)	0.95 (0.05)	0.38 (0.04)	0.33 (0.14)
	5	0.95 (0.02)	0.11 (0.03)	0.31 (0.02)	0.96 (0.02)	0.95 (0.02)	0.95 (0.02)	0.36 (0.03)	0.33 (0.15)
	10	0.95 (0.02)	0.13 (0.05)	0.30 (0.02)	0.95 (0.02)	0.95 (0.02)	0.95 (0.02)	0.30 (0.10)	0.32 (0.16)
5	2	0.92 (0.02)	0.00 (0.00)	0.29 (0.02)	0.92 (0.02)	0.92 (0.02)	0.92 (0.03)	0.34 (0.02)	0.75 (0.03)
	3	0.93 (0.02)	0.00 (0.00)	0.29 (0.03)	0.92 (0.02)	0.92 (0.02)	0.92 (0.02)	0.33 (0.03)	0.76 (0.02)
	5	0.92 (0.03)	0.00 (0.00)	0.33 (0.02)	0.92 (0.02)	0.92 (0.03)	0.92 (0.02)	0.33 (0.03)	0.75 (0.03)
	10	0.92 (0.02)	0.00 (0.00)	0.28 (0.02)	0.91 (0.02)	0.90 (0.03)	0.89 (0.02)	0.25 (0.11)	0.76 (0.03)
25	2	0.39 (0.02)	0.00 (0.00)	0.11 (0.02)	0.39 (0.02)	0.39 (0.03)	0.39 (0.02)	0.28 (0.01)	0.32 (0.02)
	3	0.39 (0.02)	0.00 (0.00)	0.11 (0.02)	0.39 (0.02)	0.39 (0.02)	0.39 (0.02)	0.27 (0.03)	0.32 (0.03)
	5	0.39 (0.03)	0.00 (0.00)	0.11 (0.02)	0.39 (0.02)	0.38 (0.03)	0.39 (0.02)	0.23 (0.06)	0.38 (0.02)
	10	0.38 (0.02)	0.00 (0.00)	0.11 (0.02)	0.39 (0.02)	0.38 (0.02)	0.39 (0.02)	0.13 (0.09)	0.32 (0.02)

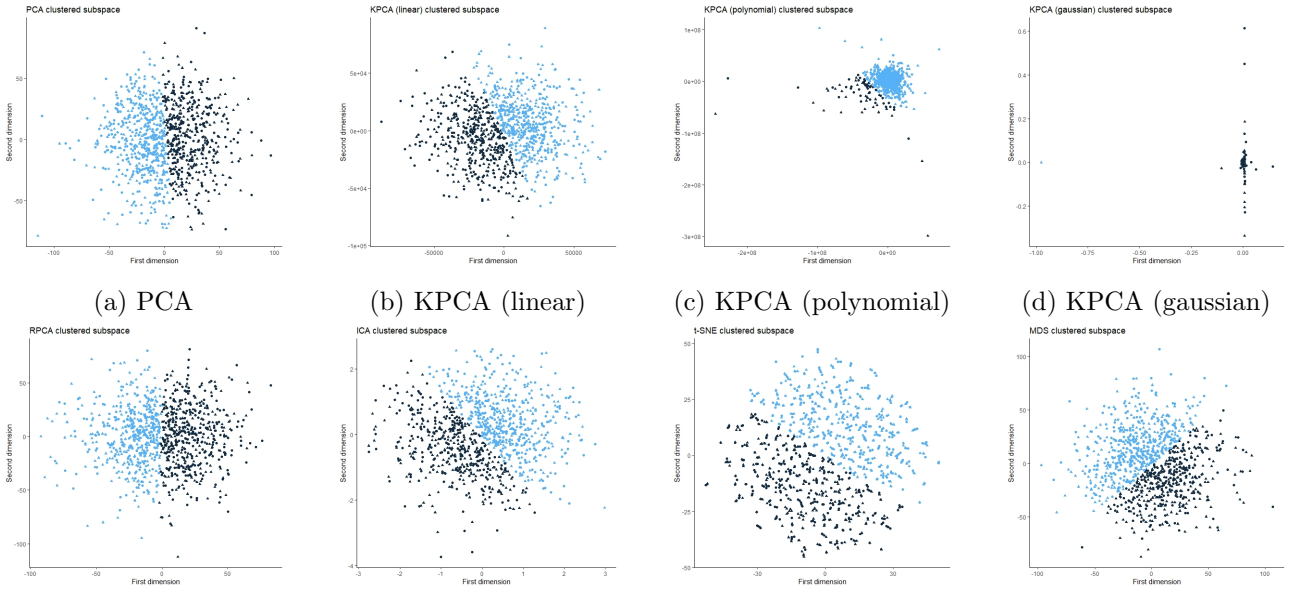
Scenario 2 extra results

Table 16: ARI values and standard errors for 2 clustered data in scenario 2

Noise	Red. Dim.	PCA	KPCA			RPCA	MDS	ICA	t-SNE
			Gaussian	Polynomial	Linear				
0.5	2	0.28 (0.05)	0.63 (0.03)	0.02 (0.01)	0.28 (0.05)	0.47 (0.08)	0.28 (0.05)	0.26 (0.08)	0.86 (0.03)
	3	0.28 (0.05)	0.62 (0.03)	0.02 (0.01)	0.28 (0.05)	0.45 (0.08)	0.28 (0.05)	0.24 (0.07)	0.85 (0.03)
	5	0.28 (0.05)	0.62 (0.03)	0.02 (0.01)	0.28 (0.06)	0.44 (0.08)	0.28 (0.05)	0.22 (0.06)	0.86 (0.03)
	10	0.29 (0.05)	0.62 (0.03)	0.02 (0.01)	0.27 (0.05)	0.36 (0.07)	0.28 (0.05)	0.81 (0.24)	0.86 (0.03)
5	2	0.28 (0.04)	0.00 (0.00)	0.01 (0.01)	0.25 (0.03)	0.29 (0.04)	0.27 (0.03)	0.26 (0.04)	0.13 (0.07)
	3	0.26 (0.04)	0.00 (0.00)	0.01 (0.01)	0.25 (0.02)	0.29 (0.04)	0.27 (0.04)	0.24 (0.07)	0.11 (0.06)
	5	0.27 (0.04)	0.00 (0.00)	0.01 (0.01)	0.27 (0.04)	0.28 (0.04)	0.27 (0.04)	0.14 (0.11)	0.11 (0.07)
	10	0.27 (0.04)	0.00 (0.00)	0.01 (0.01)	0.29 (0.06)	0.29 (0.040	0.27 (0.03)	0.05 (0.07)	0.10 (0.08)
25	2	0.01 (0.01)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.01)	0.00 (0.01)	0.00 (0.01)	0.00 (0.00)
	3	0.01 (0.01)	0.00 (0.00)	0.00 (0.00)	0.00 (0.01)	0.00 (0.00)	0.01 (0.01)	0.00 (0.01)	0.00 (0.00)
	5	0.01 (0.01)	0.00 (0.00)	0.00 (0.00)	0.01 (0.01)	0.00 (0.00)	0.00 (0.01)	0.00 (0.00)	0.00 (0.00)
	10	0.00 (0.01)	0.00 (0.00)	0.00 (0.00)	0.01 (0.01)	0.00 (0.01)	0.01 (0.01)	0.00 (0.00)	0.00 (0.00)

Table 17: ARI values and standard errors for 3 clustered data in scenario 2

Noise	Red. Dim.	PCA	KPCA			RPCA	MDS	ICA	t-SNE
			Gaussian	Polynomial	Linear				
0.5	2	0.36 (0.03)	0.33 (0.01)	0.03 (0.01)	0.36 (0.04)	0.44 (0.03)	0.36 (0.03)	0.27 (0.05)	0.85 (0.03)
	3	0.36 (0.03)	0.33 (0.01)	0.02 (0.01)	0.36 (0.03)	0.44 (0.04)	0.36 (0.03)	0.25 (0.05)	0.85 (0.03)
	5	0.36 (0.03)	0.33 (0.02)	0.03 (0.01)	0.36 (0.04)	0.43 (0.04)	0.36 (0.03)	0.21 (0.07)	0.85 (0.03)
	10	0.37 (0.04)	0.33 (0.02)	0.02 (0.01)	0.37 (0.04)	0.39 (0.04)	0.36 (0.04)	0.85 (0.03)	0.02 (0.01)
5	2	0.28 (0.02)	0.00 (0.00)	0.03 (0.01)	0.29 (0.01)	0.29 (0.02)	0.28 (0.02)	0.17 (0.02)	0.19 (0.04)
	3	0.28 (0.02)	0.00 (0.00)	0.02 (0.01)	0.27 (0.02)	0.29 (0.02)	0.28 (0.02)	0.03 (0.04)	0.19 (0.03)
	5	0.28 (0.02)	0.00 (0.00)	0.03 (0.01)	0.29 (0.01)	0.29 (0.02)	0.28 (0.02)	0.02 (0.02)	0.19 (0.04)
	10	0.28 (0.02)	0.00 (0.00)	0.03 (0.01)	0.27 (0.03)	0.29 (0.02)	0.28 (0.02)	0.03 (0.02)	0.19 (0.04)
25	2	0.02 (0.01)	0.00 (0.00)	0.00 (0.00)	0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.00 (0.00)
	3	0.02 (0.01)	0.00 (0.00)	0.00 (0.00)	0.02 (0.01)	0.01 (0.01)	0.02 (0.01)	0.01 (0.01)	0.00 (0.00)
	5	0.02 (0.01)	0.00 (0.00)	0.00 (0.00)	0.02 (0.01)	0.01 (0.01)	0.02 (0.01)	0.01 (0.01)	0.01 (0.02)
	10	0.02 (0.01)	0.00 (0.00)	0.00 (0.00)	0.02 (0.01)	0.01 (0.01)	0.02 (0.01)	0.01 (0.01)	0.01 (0.01)



(a) PCA (b) KPCA (linear) (c) KPCA (polynomial) (d) KPCA (gaussian)

(e) RPCA (f) ICA (g) t-SNE (h) MDS

Figure 10: Formed clusters under addition of noise scenario 2

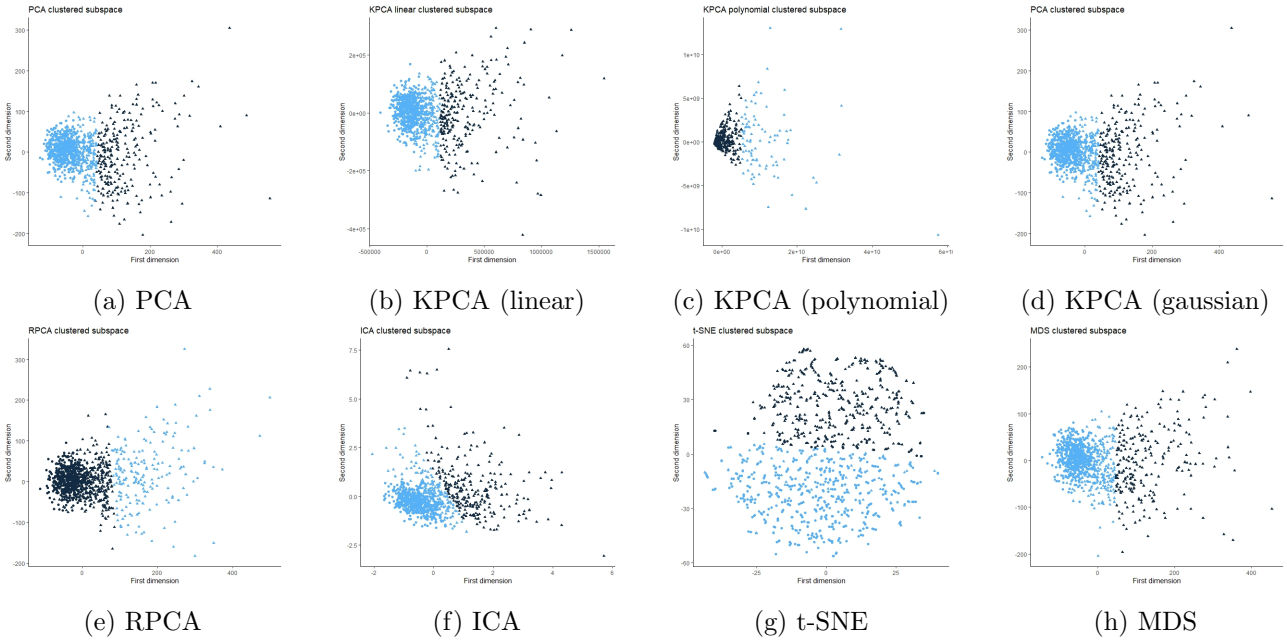
Scenario 3 extra results

Table 18: ARI values and standard errors for 2 clustered data in scenario 3

Noise	Red. Dim.	PCA	KPCA			RPCA	MDS	ICA	t-SNE
			Gaussian	Polynomial	Linear				
0.5	2	0.24 (0.05)	0.15 (0.05)	0.01 (0.01)	0.23 (0.06)	0.84 (0.17)	0.23 (0.06)	0.21 (0.08)	0.87 (0.26)
	3	0.22 (0.05)	0.15 (0.05)	0.01 (0.01)	0.25 (0.06)	0.77 (0.30)	0.22 (0.05)	0.99 (0.02)	0.91 (0.11)
	5	0.22 (0.05)	0.16 (0.04)	0.01 (0.01)	0.22 (0.05)	0.27 (0.11)	0.23 (0.05)	1.00 (0.00)	0.90 (0.13)
	10	0.22 (0.03)	0.16 (0.05)	0.01 (0.02)	0.23 (0.05)	0.21 (0.02)	0.22 (0.03)	1.00 (0.01)	0.89 (0.10)
5	2	0.23 (0.04)	0.00 (0.00)	0.01 (0.01)	0.22 (0.05)	0.64 (0.09)	0.22 (0.04)	0.22 (0.03)	0.86 (0.10)
	3	0.21 (0.05)	0.00 (0.00)	0.01 (0.00)	0.23 (0.04)	0.49 (0.07)	0.24 (0.05)	0.98 (0.02)	0.86 (0.09)
	5	0.22 (0.05)	0.00 (0.01)	0.01 (0.01)	0.22 (0.05)	0.33 (0.06)	0.22 (0.05)	0.99 (0.00)	0.86 (0.10)
	10	0.23 (0.04)	0.00 (0.00)	0.00 (0.01)	0.23 (0.05)	0.26 (0.04)	0.24 (0.05)	0.99 (0.01)	0.87 (0.11)
25	2	0.24 (0.05)	0.00 (0.00)	0.01 (0.01)	0.22 (0.04)	0.29 (0.07)	0.23 (0.06)	0.27 (0.06)	0.53 (0.04)
	3	0.25 (0.04)	0.00 (0.00)	0.01 (0.01)	0.25 (0.04)	0.23 (0.06)	0.22 (0.05)	0.54 (0.11)	0.54 (0.04)
	5	0.24 (0.05)	0.00 (0.00)	0.00 (0.00)	0.25 (0.03)	0.27 (0.06)	0.23 (0.05)	0.45 (0.08)	0.55 (0.09)
	10	0.25 (0.04)	0.00 (0.00)	0.01 (0.01)	0.23 (0.04)	0.25 (0.07)	0.24 (0.05)	0.25 (0.06)	0.53 (0.06)

Table 19: ARI values and standard errors for 3 clustered data in scenario 3

Noise	Red. Dim.	PCA	KPCA			RPCA	MDS	ICA	t-SNE
			Gaussian	Polynomial	Linear				
0.5	2	0.61 (0.04)	0.40 (0.07)	0.05 (0.02)	0.59 (0.03)	0.59 (0.03)	0.59 (0.03)	0.32 (0.06)	0.83 (0.08)
	3	0.60 (0.04)	0.40 (0.06)	0.05 (0.01)	0.59 (0.02)	0.59 (0.03)	0.59 (0.03)	0.63 (0.05)	0.80 (0.07)
	5	0.60 (0.03)	0.40 (0.07)	0.05 (0.02)	0.60 (0.02)	0.60 (0.02)	0.60 (0.03)	0.76 (0.07)	0.82 (0.07)
	10	0.59 (0.04)	0.40 (0.06)	0.04 (0.02)	0.59 (0.03)	0.60 (0.00)	0.60 (0.02)	0.71 (0.05)	0.83 (0.06)
5	2	0.59 (0.03)	0.00 (0.01)	0.04 (0.02)	0.60 (0.02)	0.59 (0.03)	0.59 (0.04)	0.34 (0.09)	0.91 (0.06)
	3	0.60 (0.03)	0.00 (0.00)	0.04 (0.01)	0.59 (0.03)	0.59 (0.020	0.59 (0.02)	0.60 (0.10)	0.89 (0.07)
	5	0.59 (0.02)	0.00 (0.00)	0.05 (0.01)	0.59 (0.04)	0.61 (0.03)	0.60 (0.02)	0.74 (0.08)	0.92 (0.06)
	10	0.60 (0.03)	0.00 (0.00)	0.04 (0.01)	0.60 (0.03)	0.60 (0.03)	0.59 (0.03)	0.74 (0.07)	0.94 (0.06)
25	2	0.57 (0.03)	0.00 (0.00)	0.05 (0.02)	0.57 (0.03)	0.57 (0.02)	0.57 (0.03)	0.32 (0.06)	0.69 (0.10)
	3	0.58 (0.02)	0.00 (0.00)	0.04 (0.02)	0.58 (0.03)	0.58 (0.03)	0.58 (0.03)	0.42 (0.08)	0.70 (0.08)
	5	0.58 (0.02)	0.00 (0.00)	0.05 (0.02)	0.58 (0.02)	0.58 (0.03)	0.57 (0.01)	0.42 (0.06)	0.71 (0.08)
	10	0.58 (0.03)	0.00 (0.00)	0.05 (0.01)	0.57 (0.02)	0.58 (0.03)	0.57 (0.02)	0.37 (0.05)	0.69 (0.07)



(a) PCA (b) KPCA (linear) (c) KPCA (polynomial) (d) KPCA (gaussian)

(e) RPCA (f) ICA (g) t-SNE (h) MDS

Figure 11: Formed clusters under addition of noise scenario 3

7.2 Extra results Breast Cancer Data

Table 20: Classification performances based on a support vector machine classifier

Performance Measure:		Accuracy			F1 Score			AUC		
Dimension:		2	5	10	2	5	10	2	5	10
Benchmark		**0.94**			**0.87**			**1.00**		
SPCA	linear	0.92	0.93	0.93	0.84	0.86	0.86	0.97	0.98	0.99
	delta	0.91	0.93	0.93	0.83	0.86	0.88	0.98	0.98	0.98
SKPCA	gaussian	0.76	0.76	0.76	0.18	0.18	0.18	0.93	0.93	0.93
	linear	0.77	0.77	0.77	0.00	0.00	0.00	0.98	0.98	0.98
	polynomial	0.77	0.77	0.77	0.00	0.00	0.00	0.98	0.98	0.98
ICA-FX		0.87	0.88	0.93	0.76	0.79	0.86	0.96	0.96	0.99
t-SNE		0.92	0.92	0.92	0.85	0.59	0.60	0.97	0.98	0.99

7.2.1 Extra Alzheimer's Handwriting Data Results with SVM classifier

Table 21: SVM Classifier for Alzheimer's Handwriting Data

Performance Measure		Accuracy				F1 Score				AUC			
Reduced Dimension		2	5	10	50	2	5	10	50	2	5	10	50
Benchmark		0.43				0.00				0.50			
SPCA	linear	0.51	0.51	0.55	0.57	0.66	0.66	0.62	0.60	0.50	0.50	0.50	0.50
	delta	0.57	0.49	0.47	0.53	0.60	0.68	0.69	0.64	0.46	0.46	0.50	0.50
SKPCA	gaussian	0.53	0.57	0.47	0.51	0.00	0.00	0.00	0.66	0.50	0.50	0.50	0.50
	linear	0.45	0.47	0.39	0.61	0.71	0.69	0.76	0.54	0.50	0.50	0.50	0.50
	polynomial	0.57	0.47	0.49	0.53	0.00	0.00	0.68	0.00	0.50	0.50	0.50	0.50
ICA-FX		0.69	0.67	0.63	0.80	0.68	0.67	0.63	0.81	0.80	0.69	0.40	0.90
t-SNE		0.92	0.80	0.75	0.75	0.80	0.89	0.86	0.96	0.79	0.11	0.09	0.13

7.3 Kernel functions for KPCA

For the Kernel PCA analysis in this thesis three different kernel functions are taken into consideration. The first one is a linear kernel which can be written as: $k(x_1, x_2) = x_1^T x_2 + c_1$. The second kernel taken into consideration is the polynomial kernel which can be written as: $k(x_1, x_2) = (c_1 x_1^T x_2 + c_2)^{c_3}$. The last kernel function which is taken into consideration is the Gaussian kernel which can be written as: $k(x_1, x_2) = exp(-\frac{\|x_1 - x_2\|_2^2}{2\sigma^2})$.

7.4 G_i functions for ICA

For the ICA clustering procedure two different G_i functions will be taken into consideration. The first equals: $G_1(u) = \frac{1}{a_1} log(cosh(a_1 u))$ where $1 \leq a_1 \leq 2$. The derivative of this function equals: $g_1(u) = tanh(a_1 u)$. The second function which is taken into consideration can be written as $G_2(u) = -exp(-\frac{u^2}{2})$ and the corresponding derivative equals $g_2(u) = u * exp(-\frac{u^2}{2})$.

7.5 Connection SVD and EVD

Let $\mathbf{X}$ be a numerical matrix where the SVD will be performed on. So with SVD it is possible to decompose $\mathbf{X} = \mathbf{Q\Sigma V^T}$. Here $\mathbf{\Sigma}$ is diagonal matrix which means that $\mathbf{\Sigma^T} = \mathbf{\Sigma}$, $\mathbf{Q}$ is a column orthonormal matrix of dimension N x r and $\mathbf{V}$ is also a column orthonormal matrix of dimension d x r. d equals the dimension of the data, r is the rank of the data and N is the amount of observations. Now it is possible to derive the following:

$$\mathbf{X^T X} = \mathbf{V\Sigma Q^T Q\Sigma V^T} = \mathbf{V\Sigma^2 V^T} \tag{42}$$

Note that the sample covariance matrix of $\mathbf{X}$ equals $\mathbf{S} = \frac{1}{N-1}\mathbf{X^T X}$. From the EVD it is known that this sample covariance matrix can be decomposed into $\mathbf{S} = \mathbf{V\Lambda V^T}$. Now it is clearly visible that the matrix $\mathbf{V}$ of the SVD contains the eigenvectors of the covariance matrix of $\mathbf{X}$. Another result which can be obtained here is that $\lambda_i = \frac{\sigma_i^2}{N-1}$.