Stockholms universitet

Statistical survey of clustering using message passing

Hiam Shaba

Masteruppsats i matematisk statistik
Master Thesis in Mathematical Statistics

Masteruppsats 2021:1
Matematisk statistik
Februari 2021

www.math.su.se

Matematisk statistik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm

Matematiska institutionen

Stockholm University

Mathematical Statistics
Stockholm University
Master Thesis **2021:1**
http://www.math.su.se

Statistical survey of clustering using message passing

Hiam Shaba*

February 2021

Abstract

Clustering analysis is an important part of machine learning due to the need of grouping data into segments. By handling the abundant amount of data we have today, these analyses have created new opportunities. Different fields such as social science and biology have benefited from cluster analysis and machine learning in general. However, clustering methods usually limit us to certain types of similarity measures and require to make assumptions on the structure of the data. *Affinity propagation* (AP) is a method that addresses these inconveniences. The algorithm can take nonmetric similarity graphs as input and do not require the number of clusters prespecified, which creates great opportunities in a variety of fields. This study aims to scrutinize AP, using the negative squared Euclidean distance as similarity measure, and its inputs. We will also compare it to one of the most common clustering methods, *k-means*. By investigating the method's statistical properties with different test examples, we conclude that the results from AP are similar to k-means. The results show similar clustering of imbalanced, noisy and arbitrarily shaped data. Both methods try to cluster imbalanced and arbitrarily shaped data into balanced spherically shaped clusters, and may find structure in noise when clustering noisy data. Moreover, AP is computationally expensive in computer time when dealing with large datasets, since it needs to be run with multiple self-similarities to find a suitable value. To find the right self-similarity the original authors of AP applied a root-finding method called the bisection method, which is slow. For further studies, we therefore suggest using a faster root-finding method than the bisection method, to increase the efficiency when searching for the right self-similarity.

*Postal address: Mathematical Statistics, Stockholm University, SE-106 91, Sweden. E-mail: hiam.shaba@hotmail.com. Supervisor: Chun-Biu Li.

Acknowledgements

I would like to thank my supervisor, Prof. Chun-Biu Li, for his encouragement, support and guidance through this thesis. It would not have been possible without his help. I would also like to show my gratitude and thank my dear friend Marina for proofreading and commenting the report.

Contents

1 Introduction

1.1 Background

Current abundance of data is one of the main reasons why machine learning (ML) has gained popularity lately. This requires methods that provide a possibility to locate and assess the data, and to control information overload. Hence, we avoid the problem of losing valuable data. ML is also used in great extent due to its ability to handle high dimensional data, such as medical records and financial transactions. Together with the increasing computer power and more elegant methods, we are able to create models and methods that enhance companies' productivity, such as face recognition in Apple and user recommendations in Netflix or Spotify. ML can be defined as a collection of statistical methods used for regression, classification, dimensionality reduction or clustering. ML approaches can be divided into two categories, namely, *supervised learning* and *unsupervised learning*. The main difference between supervised and unsupervised learning resides in how each method learns from the data. The data decides what category we apply and can consist of only independent variables or both independent and dependent variables. Hence, the data can be unlabeled or labeled. Supervised learning is based on a dependent approach, where the algorithms are trained on labeled data, while unsupervised learning is based on finding patterns in unlabeled data. The two main topics in unsupervised learning are *dimensionality reduction* and *clustering*, which will be the main focus of this thesis. Clustering analysis, together with a suitable similarity measure, which is a function that quantifies the similarity between two data points, aims to segment data points into groups that are more similar to each other than to observations in other groups. Therefore, the groups should be meaningful to capture the natural structure of the data, and useful for the study.

Clustering analysis has been used in a wide variety of fields, such as biology and social sciences [18]. Some basic clustering methods are k-means, hierarchical clustering and density-based clustering. K-means is a distance-based algorithm, known to be one of the simplest and most common methods for identifying clusters in unsupervised learning. Similarly to many other clustering methods, the performance of the k-means algorithm is influenced by the choice of the parameter k. This aspect of the method is considered to be one of its main artifacts, as a suitable choice of k will in most of the cases require some prior knowledge of the data, and this is not always the case. The AP method, unlike k-means and other simple clustering methods, does not need to make any assumption on the number of clusters. This creates great opportunities for the user by letting the algorithm decide a suitable number itself solely based on the information of the similarity matrix as input. Moreover, the method can take metric and nonmetric similarity measures which makes it applicable in many contexts.

1.2 Objectives

The main objective of this thesis is to scrutinize AP and its performances by explaining the intuition behind the algorithm. The *shared preference* is one of the inputs of AP and is the self-similarity, i.e., the similarity of an observation to itself, that is obtained from the information of the similarity matrix. This thesis also aims to explain how to evaluate the similarity matrix and the shared preference, and how AP differs from more common methods such as k-means. We also aim to investigate the method statistically with different datasets and evaluate possible limitations, artifacts, weaknesses and strengths. To complete the study we validate the results with different validation indices, which are methods used to validate the quality of the clustering results. The choice of validation index depends on the type of data, e.g., if the data is lying on the Euclidean or non-Euclidean space.

1.3 Outline of thesis

The structure of the thesis is as follows. Section 2 consists of the theory and methods essential for the thesis. We provide the important concepts and steps of the k-means algorithm and AP, where advantages and disadvantages of the methods are discussed. The theory behind *principal component analysis* (PCA), which is used for data visualization, is also presented in this section. Moreover, we define and apply the *silhouette coefficient*, which is a validation method for clusterings. We also present another validation method with a validation index based on nearest neighbours, which might also be used on clusterings of data lying on the non-Euclidean space. In Section 3 the methods are applied on different kind of test cases and the clustering results are analyzed and validated. Artifacts of AP are also presented in this section. Finally, discussions, conclusions and outlooks for further studies are presented in Section 4.

2 Theory and method

2.1 The k-means clustering

Clustering is a method to categorize data into groups where the observations with similar characteristics are grouped together. Their relation can be measured by a pairwise dissimilarity measure, and clustered into groups based on the information obtained from the dissimilarities. The k-means clustering algorithm is the oldest and most used clustering algorithm [18]. It is based on clustering data where a cluster's observations are closer to their centroid, which is the mean of the observations in the cluster and the cluster center in k-means, than to other clusters' centroids. This indicates that the pairwise dissimilarities between the observations in the same cluster tend to be smaller than to the other clusters. The centroids are usually applied to observations in the continuous D-dimensional space. Hence, the method is mostly used on quantitative data using the squared Euclidean distance as the dissimilarity measure, where

we sum over all the pairwise dissimilarities [8]. The pairwise dissimilarity is given by

$$d(i,k) = \sum_{\alpha=1}^{D} (x_{i\alpha} - x_{k\alpha})^2 = ||x_i - x_k||^2 \tag{1}$$

where x_i and x_k are D-dimensional data points in the dataset $X = (x_1, ..., x_N)$.

Generally, clustering methods assign every observation to one cluster only, meaning that the assignments have the task to map the ith observation to the lth cluster, based on the dissimilarity measure. The aim is to assign points close to each other to the same cluster in order to minimize the within-point scatter loss function, which tries to minimize the variance of the clusters. The within-point scatter is the following

$$W(C) = \frac{1}{2} \sum_{l=1}^{K} \sum_{x_i \in c_l} \sum_{x_k \in c_l} d(i,k) \tag{2}$$

where c_l is the lth cluster in the set $C = (c_1, ..., c_K)$. The within-point scatter measures how close observations within each cluster are to each other, which we aim to minimize. On the other hand, the between-point scatter measures how far apart observations in different clusters are from each other, which we aim to maximize. The total-point scatter is constant given the data and is the sum of the within-point and between-point scatter, and given by

$$T = \frac{1}{2} \sum_{l=1}^{K} \sum_{x_i \in c_l} \left(\sum_{x_k \in c_l} d(i,k) + \sum_{x_k \notin c_l} d(i,k) \right) = W(C) + B(C) \tag{3}$$

where the between-point scatter is given by

$$B(C) = \frac{1}{2} \sum_{l=1}^{K} \sum_{x_i \in c_l} \sum_{x_k \notin c_l} d(i,k) \tag{4}$$

The within-point, between-point and total-point scatter give the following relation

$$W(C) = T - B(C) \tag{5}$$

which reflects that minimizing $W(C)$ is equivalent to maximizing $B(C)$. In k-means, the within-point scatter is given by

$$W(C) = \sum_{l=1}^{L} N_l \sum_{x_i \in c_l} ||x_i - \mu_l||^2 \tag{6}$$

where the mean vector $\mu_l = (\mu_{1l}, ..., \mu_{Dl})$ is of the lth cluster for every dimension, and $N_l = \sum_{n=1}^{N} I(x_i \in c_l)$, which is the number of observations belonging to cluster c_l. How Equation 2 is equivalent to Equation 6 for k-means is proven

in Appendix A. The algorithm aims to minimize the within-point scatter by using the mean of the observations in each cluster as the centroid. The mean is obtained from the derivative of $W(C)$ with respect to the mean, which is set equal to zero to solve out the mean μ_l and obtain $\mu_l = \frac{1}{N_l} \sum_{x_i \in c_l} x_i$. Thus, the average of each cluster's data points should be used as the centroid to minimize the loss function.

2.1.1 The algorithm

The k-means algorithm is described in Algorithm 1. With the data points and the anticipated number of clusters as inputs, the cluster assignments can be computed. Firstly, it randomly assigns initial values of the centroids. Secondly, the within-point scatter is minimized by assigning each observation to its nearest centroid according to the squared Euclidean dissimilarity measure. Thirdly, the centroids are recomputed for each cluster as the mean of all the data points belonging to that cluster. This is repeated until convergence and lastly, the cluster assignments are obtained.

Algorithm 1: k-means algorithm

Input: Set of data points
Number of clusters K
Initialization: Randomly assign initial values for the centroids $\{m_1, ..., m_K\}$
Repeat: Update the following until convergence:
The within-point scatter

$$\sum_{l=1}^{K} N_l \sum_{x_i \in c_l} ||x_i - m_l||^2$$

is minimized by assigning each data point to the nearest centroid. Hence, this indicates that the cluster assignment of the point x_i is computed according to the following rule

$$\text{argmin}_{1 \leq l \leq K} ||x_i - m_l||^2$$

Recompute the centroid for each cluster as the mean of all data points in the respective cluster

$$m_l = \frac{1}{N_l} \sum_{x_i \in c_l} x_i$$

Output: Cluster assignments

Algorithm 1 does not guarantee to find a global minimum and there is risk of convergence on a local minimum. The problem may occur from poorly chosen centroids. The choice of the initial centroids is essential. In many cases the initializations are randomly sampled, which could lead to poor results. One way

to obtain a more reliable result is to perform multiple runs where the initializations are sampled, and the clustering result with the lowest sum of squared errors is chosen. The sum of squared errors evaluates the performance of clustering methods on unlabeled data, and measures how much the clusters vary by computing the sum of the squared differences between every observation in a cluster and its cluster's mean. However, this does not necessarily result in a global minimum despite the many iterations. Moreover, the method is an optimization greedy algorithm since it minimizes the within-point scatter and repeatedly updates the centroids in each iteration, which may give results on a local minimum.

The algorithm is a variant of the *Expectation-maximization algorithm* (EM) [2]. Recall that the EM algorithm consists of two phases, where the first phase is the E-step and the second phase is the M-step. The E-step in EM consists of assigning a data point to a cluster using a certain probability weight, which is from a chosen probability distribution. EM computes *soft clustering*, which is when a data point can belong to more than one cluster, and is assigned to the cluster with a certain probability weight. Moreover, in the M-step EM recomputes the parameters of the probability distribution, by maximizing the expected logarithm of the density function based on the current assignments. For k-means, the E-step consists of assigning each observation to the closest centroid. Hence, as opposed to EM, each data point in k-means can only belong to one cluster. Furthermore, the M-step consists of recomputing the centroid of each cluster by computing the mean of the observations in the cluster.

2.1.2 Advantages and disadvantages

The k-means is simple, easy to implement and a good choice of clustering method when the data consists of equally sized spherical clusters. Other methods, such as DBSCAN [4] and hierarchical clustering [8], tend to be more expensive in speed and memory which makes k-means a common choice. Nevertheless, k-means has several well-known limitations. When using the method we restrict ourselves to data with centroids that are randomly initialized in every run [18]. Therefore, every run could compute different results of both the centroids and clusterings. Some initial centroids can be poorly chosen. Consequently, we compute multiple runs to obtain the best result with the lowest sum of squared errors. However, the reached optimum may still not be global and a poor initialization can result in, e.g., a true cluster being split.

Another disadvantage is that k-means is centroid-based and restricted to the squared Euclidean distance, which prevents us from applying it to non-euclidean features and non-spherically shaped data. The method also has problem to handle imbalanced clusters, in both cases of imbalance in the number of data points and spatial extension, mostly the latter, due to the same reasons. Furthermore, the dissimilarity measure does not minimize the sum of squared distances of each cluster separately. The measure minimizes the sum of squared distances for all

clusters together by assigning observations belonging to clusters with more observations to clusters with fewer observations, meaning that the method tends to form clusters of similar size. This is illustrated in Figure 1 which demonstrates how the method gives rise to clusters with similar sizes. Moreover, k-means is not efficient for data with outliers or noise since it may find structure in them. Detecting and removing outliers can be essential for a good clustering result.

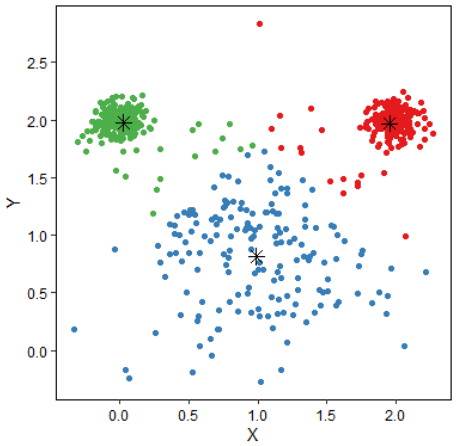


Figure 1: Example of k-means clustering of three Gaussian mixtures to illustrate unsuccessful clustering for data with imbalanced spatial extent. The three clusters identified by k-means are colored in green, blue and red, respectively. The means of the Gaussians are $\mu_{\text{green}} = (0, 2)$, $\mu_{\text{blue}} = (1, 1)$ and $\mu_{\text{red}} = (2, 2)$, and the standard deviations are $\sigma_{\text{green}} = \sigma_{\text{red}} = 0.1$ and $\sigma_{\text{blue}} = 0.5$. (X, Y) are the 2D features. The centers of the three Gaussians are indicated by the star-symbols.

Furthermore, an issue common for most clustering methods, such as k-means and EM, is the choice of the number of clusters. Some examples of clustering methods that do not require the number of clusters prespecified are, e.g., DBSCAN [4] and hierarchical clustering [8]. DBSCAN and hierarchical clustering do not need to prespecify the number of clusters since DBSCAN is based on the data points' density in a region, and hierarchical clustering considers every observation as a cluster and groups similar observations together. AP is not as common as the mentioned methods, so we will investigate it more thoroughly in this thesis [5].

2.2 Affinity propagation

AP is a clustering method introduced in 2007 by Frey and Dueck [5]. The authors aim to create a fast method that is easy to implement and can take metric and nonmetric data, i.e., negative, asymmetric or violating the triangle

inequality, as input. These are qualities that create broad opportunities for the user. AP uses *exemplars*, i.e., centers of the clusters obtained in the last stage of the algorithm, to identify clusters and considers all data points as potential exemplars in a deterministic way without the need of sampling the initializations as in the case of k-means. The deterministic exemplars are actual data points, which is a benefit dealing with data from, e.g., bioinformatics since the exemplars of DNA segments would not be computed as hypothetical averages. Furthermore, AP does not make assumptions on the structure of the data, but relies on the information obtained from the similarity matrix to decide the number of clusters.

AP has been applied in different areas. The authors of AP have applied the method to data of genes, images and airline travel times which demonstrates its diversity. The method is suitable for image segmentation since it does not require random selection of initial exemplars, which gives more stable clustering results [22], and for pairs of stereo images to find their global optimum [14]. Moreover, socioeconomic research has recently used AP where they have used zip-codes to cluster areas in the USA [9]. It has also been applied in identification of vulnerable lines in smart grid systems [6], i.e., electrical grids that consist of diverse operation and energy measures.

The algorithm consists of two quantities that compute the message passing, the *responsibility* and the *availability*, where messages are exchanged between data points [5]. The definitions and the intuitions behind the quantities will be described in the following sections.

2.2.1 The quantities of message-passing

2.2.1.1 The responsibility Considering other potential exemplars for data point x_i, the $N \times N$ responsibility matrix assesses how suitable data point x_k is to be an exemplar of data point x_i. The responsibilities $r(i,k)$ which are sent from point x_i to x_k are illustrated in Figure 2 and computed with the rule:

$$r(i,k) \leftarrow s(i,k) - \max_{k':k' \neq k} \left\{ a(i,k') + s(i,k') \right\} \tag{7}$$

where the availability $a(i,k')$ assesses how available $x_{k'}$ can serve as an exemplar for x_i, considering the other data points' support of having $x_{k'}$ as their exemplar. The responsibility can be interpreted as the relative similarity since it reflects how similar x_i is to x_k relative to $x_{k'}$, considering its availability. Furthermore, Equation 7 can be explained as the similarity between a data point and its potential exemplar, which will decrease with the largest similarity and availability the data point has to another potential exemplar. Thus, the similarity will decrease with the largest similarity and availability that the data point should choose another exemplar. The responsibility reflects how similar an observation and its potential exemplar are to each other, considering how that observation is related to other competing candidate exemplars. By subtracting

the similarity with the largest value of the availability and similarity the data point has to another exemplar, the relation between the investigated points x_i and x_k is weakened.

For $k = i$, the "self-responsibility" $r(k, k)$ is defined by

$$r(k,k) \leftarrow s(k,k) - \max_{k':k' \neq k} \left\{ s(k,k') \right\} \tag{8}$$

which is the shared preference minus the largest similarity the data point x_k has to other potential exemplars. In this case, the availability is not taken into consideration since we will not have the data point x_i to consider when computing the availability. This means that we will not be able to assess how suitable it would be for $x_{i=k}$ to choose x_k as an exemplar, since they are the same. Thus, the self-responsibility is deterministic that only depends on the similarity matrix.

In every iteration of the AP algorithm the responsibilities and availabilities are updated. The AP algorithm is initialized by setting all availabilities to zero in the first iteration, meaning that the responsibility between x_i and x_k in Equation 7 will only decrease depending on the largest similarity the data point x_i has to other potential exemplars, and therefore, x_k will be less suited as an exemplar of x_i. The responsibility in later iterations will decrease as the availabilities increase, which indicates that the exemplar's fit to x_i will decrease when x_i is more suitable to choose another exemplar. In the case that a data point is assigned to another exemplar, the availability will be negative which will give a larger responsibility than before. Hence, the availabilities will decrease the similarities between data point x_i and other exemplars $x_{k'}$, and therefore remove the possible exemplars from competition, making data point x_k more suitable as an exemplar for data point x_i.

2.2.1.2 The availability The availability is defined as the smaller value between zero, and the sum of the self-responsibility of x_k and the summation of all positive responsibilities x_k receives from the other data points. Considering the support from other observations that x_k should be their exemplar, the $N \times N$ availability matrix assesses how suitable it would be for data point x_i to choose x_k as an exemplar. The off-diagonal availabilities, i.e., the availabilities computed for $i \neq k$, that are sent from point x_k to point x_i, are illustrated in Figure 3 and updated with the rule in the iteration:

$$a(i,k) \leftarrow \min \left\{ 0, r(k,k) + \sum_{i':i' \notin \{i,k\}} \max\{0, r(i',k)\} \right\} \tag{9}$$

The responsibilities we sum over in Equation 9 reflect how suitable it is for x_k to be an exemplar to x_i, and how suitable x_k is as an exemplar of other data points. The responsibilities of x_k in the summation over i' are positive since a

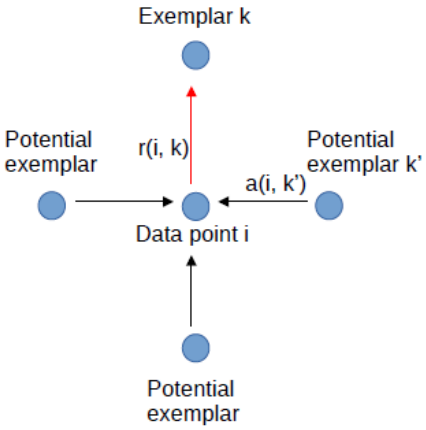


Figure 2: An illustration of sending responsibilities. The image reflects how the responsibility, i.e., the evidence of how well-suited x_k is as an exemplar of data point x_i, is sent from x_i to x_k considering other potential exemplars for x_i. Other potential exemplars send their availabilities to x_i, which in combination with the given information send its responsibility to x_k.

good exemplar should reflect how similar it is to some data points, regardless of how dissimilar it is to other data points. An exemplar's dissimilarities to data points are reflected by the negative responsibilities. As stated in the original article, it is mainly relevant for a suitable exemplar to explain some observations well regardless of how bad it explains other observations. The self-responsibility in Equation 8 reflects how suitable x_k is as an exemplar. By choosing the smaller value between zero and the sum of the responsibilities in Equation 9, we limit the impact positive responsibilities have on the availabilities and threshold it so it does not rise above zero. An availability of zero indicates that x_i should choose x_k as an exemplar since it is the highest possible availability value, and x_k would in general be fit as an exemplar. Note how the availability aims to gather evidence that the potential exemplar is an appropriate choice, rather than evaluating all possible exemplars of x_i, as computed by the responsibility.

If the self-responsibility $r(k, k)$ is negative, it can be noticed from Equation 8 that a candidate exemplar would be more suitable as a data point of another cluster than being an exemplar. This could result in a negative or small value of the availability since x_k will not be suitable as an exemplar for other data points either. In the case the self-responsibility $r(k, k)$ is zero, the availability will only depend on how suitable data point x_k is as an exemplar of other data points. The availability will then have the value zero, since we will only sum over positive responsibilities. Having a positive self-responsibility indicates that point x_k is well-suited as an exemplar. Adding more positive responsibilities would still give an availability of the value zero. Hence, it is suitable for x_i

to choose x_k as an exemplar. The update rule for self-availability $a(k,k)$ was proposed to be

$$a(k,k) \leftarrow \sum_{i':i' \neq k} \max\left\{0, r(i',k)\right\} \tag{10}$$

which reflects how suitable x_k is to be an exemplar, depending on positive responsibilities of x_k and other data points. As mentioned earlier, positive responsibilities indicate that an exemplar is more suitable for some data points regardless of how dissimilar it is to other data points. Only the positive responsibilities are included since it is relevant that an exemplar explains some data points well, no matter how bad it explains other data points. When this is the case, the diagonal of the availability matrix will consist of zeroes and positive values reflecting how suitable x_k is as an exemplar from the information obtained from the other observations.

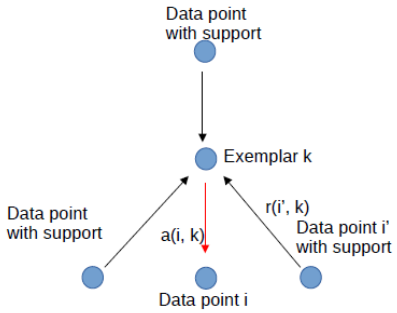


Figure 3: An illustration of sending availabilities. The image reflects how the availability, i.e., how suitable it is for data point x_i to choose x_k as an exemplar, is sent from x_k to x_i considering the support of other data points that x_k should be their exemplar. The data points send their responsibilities to x_k, which in combination with the given information send its availability to x_i.

2.2.2 Inputs and other quantities of message-passing

The message-passing algorithm involves several other quantities that will be discussed as follows. The similarity graph which is the input of the message-passing algorithm, the shared preference and the damping factor λ. The final step of the algorithm is to compute the clusterings, which are obtained with the criterion matrix presented in Section 2.2.2.4.

2.2.2.1 Similarity graphs The $N \times N$ similarity matrix represents the similarity graph constructed by a function that expresses the similarity between pairwise observations from the set $\mathbf{X} = (x_1, ..., x_N)$ [13]. Data points together

with their similarities can be represented by a weighted graph.

Generally, an undirected and weighted graph is defined as $G = (V, E, W)$, where V is the set of vertices, E is the set of edges and W is the set of weights obtained from the pairwise similarities, where each vertex v_n represents an observation x_n for $n = 1, ..., N$. Furthermore, the number of vertices is $N = |V|$, where $|.|$ is the cardinality. An undirected graph is characterized by having edges where the order of the vertices is not relevant, meaning that the similarity measure of two distinct vertices will be the same on both directions, i.e., $s(i, k) = s(k, i)$. By having symmetric distances between vertices, the graph will obtain symmetric edges. When the similarity $s(i, k)$ between x_i and x_k is positive or above some threshold, the vertices will be connected and the edge weighted by the similarity. One of these thresholds can be the ε-distance between data points, which creates the *ε-neighbourhood graph*. The data points with pairwise distances smaller than ε are connected by an edge. The edges are not weighted since the pairwise distances are mostly of the same scale and would therefore not give more information of the data. Hence, the graph does not require weighted edges, which makes the ε-neighbourhood graph an unweighted graph.

Another threshold is the k-nearest neighbours where vertex v_i is connected to v_k, if v_k is one of the k-nearest neighbours of v_i. This creates a directed graph due to the asymmetric relationships. Nevertheless, the graph can be undirected if we ignore the direction of the edges and follow the nearest neighbour concept to obtain a *k-nearest neighbour graph*. Another way of making the graph undirected is by having the requirement of both vertices being among each other's nearest neighbours, which will result in the *mutual k-nearest neighbour graph*. The edges in both cases will be decided by a weight obtained from the similarity of the investigated vertices. The weight can be obtained by, e.g., $\frac{1}{||x_i - x_k||^2}$ between x_i and x_k, and reflects how the connected data points or vertices that are further away from each other, will obtain a smaller weight relative to the other nearest neighbours. If a pair of vertices are not connected according to the concept used, the edge will be given the value zero.

Lastly, the third kind of similarity graph to mention is *the fully connected graph*, where all data points with positive similarities are connected with edges weighted by the similarities. The edge's weight between data point x_i and x_k can be constructed by the Gaussian similarity function

$$s(i, k) = \exp\left(\frac{-||x_i - x_k||^2}{2\sigma^2}\right) \tag{11}$$

where σ controls the width of the neighbourhoods. However, note that the similarity only captures the relationship between different data points and not their relationship to themselves. Thus, $s(i, k) = 1$ whenever $i = k$.

Choosing the right type of similarity graph is challenging. For AP we will construct a network with pairwise similarities. Hence, the graph will be represented by a $N \times N$ similarity matrix that provides the similarity between every two data points. As mentioned earlier, the similarity matrix can be nonmetric. The chosen similarity measure for this thesis will be, to enable a comparison between AP and k-means, the negative squared Euclidean distance given by

$$s(i,k) = -||x_i - x_k||^2 \tag{12}$$

This measure, which is used in the original article [5], denotes how the similarity becomes smaller as the distance grows larger. The diagonal would consist of zeroes in the first iteration, which would result in more clusters since every data point would be mapped to itself. Using other similarity measures, such as the inverse of the squared Euclidean distance $\frac{1}{||x_i - x_k||^2}$, would give undefined self-similarities and undefined similarities between data points with zero squared Euclidean distance. Thus, the negative squared Euclidean distance is an appropriate choice. To decide the values of the diagonal, we will use the off-diagonal elements of the similarity matrix to obtain the shared preference [5]. This subject is discussed in the following section.

2.2.2.2 Shared preference The shared preference is the similarity a data point has to itself. The shared preference is zero when first computing the similarity matrix, which can be easily understood from Equation 12. The shared preference is obtained from the off-diagonal elements of the similarity matrix and reflects the number of clusters, and such choice is a convention used by the original authors [5]. From their results, they draw the conclusion that lower shared preference values penalizes data points as exemplars and create fewer clusters, while higher shared preference values obtain more exemplars. Therefore, Frey and Dueck suggest the median or the first quantile of the input similarities to obtain a moderate number of clusters, which could result in a small number of clusters. Otherwise, it is suggested to use the smallest value of the similarities to obtain one or two clusters. A small value of the shared preference will compute fewer clusters and the data would be clustered into more distinct clusters. Frey and Dueck assumes that,*a priori*, all data points are equally suitable to be exemplars which means that the shared preference should be set to a common value.

In practice, the median and the first quantile of the similarities can sometimes give too many clusters. To find the appropriate value of the shared preference, the bisection method is a commonly used algorithm for root-finding, used by the authors of AP. The method bisects the intervals of the shared preference values and runs AP for each shared preference. Frey and Dueck used bisections, and by running the algorithm multiple times with different values, they tried to obtain the desired number of clusters. To investigate which shared preference that results in the right number of clusters, a plot of the shared preference against the number of clusters allows us to choose the number of clusters based

on the range of the shared preference values. For instance, if three clusters have the widest range of preference values then three is chosen as the right number of clusters. Nevertheless, it also depends on the plotted preference values since many smaller shared preference values can just result in one cluster. Furthermore, there is no linear relationship between the shared preference value and the number of clusters.

2.2.2.3 Damped factor and noise According to Frey and Dueck [5], there is a risk of numerical oscillations when updating the message-passing quantities, meaning that the data points alternate between being exemplars and non-exemplars. Degenerate situations can occur when AP struggles with choosing the right exemplar in a cluster, and lead to oscillations. For instance, when a cluster consists of two data points that are isolated from the other data points, AP may struggle with choosing the right data point as an exemplar. Therefore, it is essential to damp the message-passing quantities and add noise to facilitate and ensure convergence [5]. The dampening is computed by using the damping factor $\lambda \in (0, 1)$, which in each iteration is multiplied to the values of message-passing steps. To compute a new responsibility or availability value, the old value from the former iteration is multiplied by λ and the updated value is multiplied by $1 - \lambda$, as the following:

$$\begin{aligned} r_{\text{new}}(i,k) &\leftarrow \lambda r_{\text{old}}(i,k) + (1-\lambda) r_{\text{new}}(i,k) \\ a_{\text{new}}(i,k) &\leftarrow \lambda a_{\text{old}}(i,k) + (1-\lambda) a_{\text{new}}(i,k) \end{aligned} \tag{13}$$

The value of the dampening factor is chosen depending on if the clustering oscillates. Consequently, we have to increase the dampening factor. According to Dueck [3], $\lambda = 0.9$ is sufficient in most of his demonstrative examples, which is also the value used in this thesis. Moreover, by adding noise to the similarities, degenerate situations may be prevented. The noise added to a similarity should be, according to Frey and Dueck [5], a percentage of the value of a standard normal random variable multiplied with the difference between the largest and smallest similarity value, and is presented in Algorithm 2.

2.2.2.4 Criterion matrix The AP algorithm reaches convergence after the clustering remains the same after a number of iterations, or after a fixed number of iterations. According to Dueck [3], the maximum number of iterations are 1000 and the number of iterations the clustering results may stay the same are 100, and are the quantities used in this thesis. When AP reaches convergence it is time to determine the exemplars. The exemplars are obtained by firstly computing the $N \times N$ criterion matrix, which is the sum of the pairwise availabilities and responsibilities between data points such as the following:

$$c(i,k) \leftarrow a(i,k) + r(i,k) \tag{14}$$

Secondly, to identify the exemplars and assignments the largest values of the criterion matrix are identified by computing $\text{argmax}_k\{c(i,k)\}$. Hence, the columns

k with the largest matrix elements of the criterion matrix are the exemplars. The rows i with the same exemplars will share cluster.

2.2.3 The algorithm

AP is described in Algorithm 2. By taking the similarity matrix, with added noise, and the shared preference as input, the message-passing steps are computed. The availability matrix will in the first iteration be initialized to zero, which is with the responsibility matrix in later iterations updated according to Equations 7-10. In each iteration, the message-passing steps are damped, to avoid oscillations with $\lambda = 0.9$. Lastly, when the matrices have reached convergence or until the number of iterations exceed a threshold, the assignments of the exemplars are computed.

Algorithm 2: Affinity Propagation

Input: Similarities $\{s(i,k)\}_{(i,k)\in\{1,...,N\}^2, i\neq k}$
Shared preferences $s(k,k) = p \; \forall k \in \{1,...,N\}$
Noise[1] to similarity matrix
Damping factor of $\lambda = 0.9$
Initialization: Availabilities are set to zero, i.e., $\forall i,k \; a(i,k) = 0$
Repeat: Update responsibility and availability until convergence or until the number of iterations exceeds its threshold, which is given by

$$\forall i,k : r(i,k) = \begin{cases} s(k,k) - \max_{k':k'\neq k}\{s(k,k')\}, & \text{for } k = i \\ s(i,k) - \max_{k':k'\neq k}\{a(i,k') + s(i,k')\}, & \text{for } k \neq i \end{cases}$$

$$r(i,k) = \lambda r_{\text{old}}(i,k) + (1-\lambda) r(i,k)$$

$$\forall i,k : a(i,k) = \begin{cases} \sum_{i':i'\neq i} \max\{0, r(i',k)\}, & \text{for } k = i \\ \min\{0, r(k,k) + \sum_{i':i'\notin\{i,k\}} \max\{0, r(i',k)\}\}, & \text{for } k \neq i \end{cases}$$

$$a(i,k) = \lambda a_{\text{old}}(i,k) + (1-\lambda) a(i,k)$$

Output: Cluster assignments $\hat{\boldsymbol{c}} = (\hat{c}_1, ..., \hat{c}_N)$, where $\hat{c}_i = \text{argmax}_k\{c(i,k)\}$

2.2.3.1 An illustrative example To exemplify AP this section will demonstrate an example taken from reference [19]. This example is chosen due to its great simplicity, and the data is obtained from an experiential exercise where the participants are asked about their opinion of tax rate, a fee for services, interest rate, quantity limit and price limit on a five-point scale. Therefore, the

[1] Adding noise to similarities as

$$s(i,k) + 1e^{-12} \times Z \times \left(\max(\{s(i,k)\}_{(i,k)\in\{1,...,N\}^2}) - \min(\{s(i,k)\}_{(i,k)\in\{1,...,N\}^2})\right)$$

where $Z \sim N(0,1)$.

data variables are categorical of the ordinal type, since the participants rank their opinions about the different subjects. The data does not require normalization since the variables are of the same scale. The aim of this example is to illustrate how people with the same opinions should be in the same group. The data used is presented in Table 1.

Participant	Tax rate	Fee	Interest rate	Quantity limit	Price limit
Alice	3	4	3	2	1
Bob	4	3	5	1	1
Cary	3	5	3	3	3
Doug	2	1	3	3	2
Edna	1	1	3	2	3

Table 1: Data to exemplify AP, which consists of five categorical variables reflecting the participants opinions about each variable on a five-point scale.

Now, since the data consists of variables of the same scale it is possible to compute the similarity matrix using the negative squared Euclidean distance as similarity measure. The similarity between, e.g., Alice and Bob is

$$s(A, B) = -((3-4)^2 + (4-3)^2 + (3-5)^2 + (2-1)^2 + (1-1)^2) = -7$$

and between Alice and Cary is

$$s(A, C) = -((3-3)^2 + (4-5)^2 + (3-3)^2 + (2-3)^2 + (1-3)^2) = -6.$$

The same calculations are computed for all combinations and the smallest similarity value among the off-diagonal elements of the similarity matrix is -22, which will be used as the shared preference. The results are presented in Table 2.

	Alice	Bob	Cary	Doug	Edna
Alice	-22	-7	-6	-12	-17
Bob	-7	-22	-17	-17	-22
Cary	-6	-17	-22	-18	-21
Doug	-12	-17	-18	-22	-3
Edna	-17	-22	-21	-3	-22

Table 2: The similarity matrix of the data from Table 1.

The initial responsibility of, e.g., Alice and Bob using Equation 7 is

$$r(A, B) = -7 - \max(-6, -12, -17) = -7 - (-6) = -1$$

and, e.g., Alice and Cary is

$$r(A, C) = -6 - \max(-7, -12, -17) = -6 - (-7) = 1.$$

After computing the responsibilities for the other combinations, the initial responsibility matrix in Table 3 is obtained.

	Alice	Bob	Cary	Doug	Edna
Alice	-16	-1	1	-6	-11
Bob	10	-15	-10	-10	-15
Cary	11	-11	-16	-12	-15
Doug	-9	-14	-15	-19	9
Edna	-14	-19	-18	14	-19

Table 3: The initial responsibility matrix of the data from Table 1.

The initial availability of, e.g., Alice and Bob using Equation 9 is

$$a(A, B) = \min(0, -15 + \sum(\max(0, -11), \max(0, -14), \max(0, -19))) = -15$$

and, e.g., Alice and Cary is

$$a(A, C) = \min(0, -16 + \sum(\max(0, -10), \max(0, -15), \max(0, -18))) = -16.$$

Moreover, the self-availabilities are computed by using Equation 10. For instance, Alice's self-similarity is obtained from the following calculations:

$$a(A, A) = \sum(\max(0, 10) + \max(0, 11) + \max(0, -9) + \max(0, -14)) = 10 + 11 = 21$$

The initial availability matrix for all pairwise combinations is presented in Table 4.

	Alice	Bob	Cary	Doug	Edna
Alice	21	-15	-16	-5	-10
Bob	-5	0	-15	-5	-10
Cary	-6	-15	1	-5	-10
Doug	0	-15	-15	14	-19
Edna	0	-15	-15	-19	9

Table 4: The initial availability matrix of the data from Table 1.

The initial criterion matrix is the sum of these two message-passing matrices from Table 3 and 4, and presented in Table 5.

	Alice	Bob	Cary	Doug	Edna
Alice	**5**	-16	-15	-11	-21
Bob	**5**	-15	-25	-15	-25
Cary	**5**	-26	-15	-17	-25
Doug	-9	-29	-30	**-5**	-10
Edna	-14	-34	-33	**-5**	-10

Table 5: The initial criterion matrix of the responsibility matrix in Table 3 and the availability matrix in Table 4. The bold numbers are the largest criterion values.

The clustering result will not change with further iterations of the message-passing quantities, meaning that the data converges in the first iteration and the initial matrices obtained in Table 3-5 are also the converged matrices. The largest criterion value of each row reflects which exemplar each participant belongs to and is presented as a bold number in Table 5. The rows with the same criterion value of their exemplars are clustered together. The columns with these criterion values are the exemplars, which in Table 5 are Alice and Doug. Therefore, it is obtained that Alice, Bob and Cary are in one cluster with Alice as the exemplar, and Doug and Edna are in one cluster with Doug as the exemplar. The result can be explained from the original data in Table 1, where in the case of tax rate, Alice, Bob and Cary prefer higher tax rates in comparison to Doug and Edna that prefer lower tax rates. The same pattern can be observed in their opinions about a fee on services, where Alice's cluster prefers high fees and Doug's cluster prefers low fees. The remaining variables do not show a distinguishing pattern of the participants opinions. However, the clusters' opinions are more distinct in at least two variables which supports the obtained clustering result. From Table 2 it can be observed that Alice and Doug are the chosen exemplars since they have the most similar opinions to the participants in their clusters. Alice is closer in opinion to Bob and Cary but Bob and Cary are not close in opinion, which makes Alice the link between these two participants. Doug is the second exemplar since he is closer in opinion to Alice than Edna, which creates a possibility for communication between the two teams. By this example it is illustrated how teams consisting of people with similar opinions can be formed for, e.g., political parties.

2.2.4 Advantages and disadvantages

AP is a practical and versatile method, and is one of the few methods that can take nonmetric similarity matrices. This is a great advantage since it makes AP applicable to all kinds of similarity measures and creates great opportunities in fields that obtain nonmetric data, such as data of images. Additionally, the exemplars are actual data points and not hypothetical averages of the clusters' observations. Hence, AP is suitable for bioinformatics such as biological network

analyses where AP is used to decompose networks into connected modules [20]. Moreover, AP is deterministic and therefore not sensitive to initializations, in contrast to the case of k-means. This is very convenient since it could be quite time-consuming to run the same algorithm multiple times to obtain the best result. The authors of AP [5] mention this as a great advantage when comparing to other methods, which also results in a lower sum of squared errors meaning that AP obtains clusters with observations closer to their cluster's mean. The mentioned qualities make AP a fast algorithm easy to implement.

The disadvantages of AP are also relevant problems to consider. The similarity matrix is computed by taking pairwise similarities of the data, which creates a matrix with number of elements growing as $\mathcal{O}(N^2)$. Therefore, a large dataset can be very time and memory consuming. Another limitation is the choice of shared preference. The problem of prespecifying the number of clusters has been replaced by the problem of prespecifying the shared preference. For data with a known number of clusters, the procedure can be time-consuming. From Figure 1 and 4 we can draw the conclusion that the clustering results could be very similar for AP and k-means. However, for k-means we prespecified the number of clusters, while AP required a bisection method since both the choices of the median and the lowest quantile of the similarity values for the shared preference may result in too many clusters.

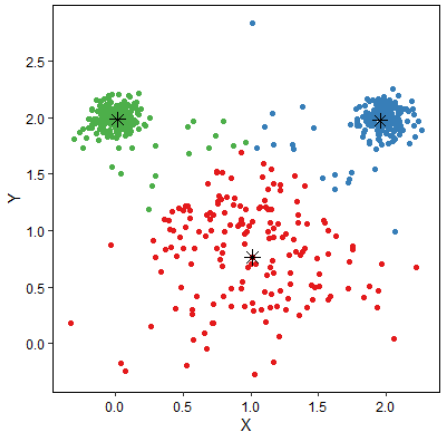


Figure 4: Example of AP clustering of three clusters of Gaussian mixtures to illustrate unsuccessful clustering for data with imbalanced spatial extent, identical to k-means. The three clusters identified by AP are colored in green, red and blue, respectively. The means of the Gaussians are $\mu_{\text{green}} = (0, 2)$, $\mu_{\text{red}} = (1, 1)$ and $\mu_{\text{blue}} = (2, 2)$, and the standard deviations are $\sigma_{\text{green}} = \sigma_{\text{blue}} = 0.1$ and $\sigma_{\text{red}} = 0.5$. (X, Y) are the 2D features. The centers of the three Gaussians are indicated by the star-symbols.

2.3 Principal component analysis

Clustering multidimensional data can be challenging since it is difficult to obtain prior knowledge about the underlying characteristics of the data. For further evaluation of the characteristics of high dimensional data, a dimensionality reduction method is required. The PCA is a method first introduced by Pearson in 1901 [15], and aims to transform multivariate data to a new coordinate system. It is an orthogonal linear transformation that projects data into a new basis, and is used for dimensionality reduction and data visualisation. When projecting the data, most of its variance is reflected in the first coordinate axis, i.e., the first principal component (PC), while the second largest variance is reflected in the second coordinate axis, i.e., the second PC, etc. PCA is suitable for multidimensional data since the sample variances along the coordinate axes are maximized and carry most of the variance of the data in the first few PCs. Hence, they can be used to visualize data, but the data is restricted on lying on the linear manifold. If the underlying data is non-linear, PCA might be misleading. The PCs are computed by finding the eigenvectors and eigenvalues of the covariance matrix, and projecting the data onto a subspace with eigenvectors that contain the largest eigenvalues as basis.

The aim of PCA is to transform the $N \times D$ data matrix $\mathbf{X}$, where $x_n \in \mathbb{R}^D$, to the matrix $\mathbf{Y}$ of dimension d, where $y_n \in \mathbb{R}^d$. The data matrix $\mathbf{X}$ consists of N row vectors with D columns, where row n consists of the vector $\mathbf{x}_n = (x_{n1}, ..., x_{nD})$ and column α consists of the vector $\mathbf{x}_\alpha = (x_{1\alpha}, ..., x_{N\alpha})^T$. PCA computes the new representation $\mathbf{Y}$ where $d < D$, such that

$$\mathbf{Y} = \mathbf{XT} \tag{15}$$

where the $D \times d$ matrix $\mathbf{T}$ consists of columns that represent the new basis' vectors, where $\mathbf{X}$ is projected.

The data matrix $\mathbf{X}$ is, without loss of generality, assumed to be centered. If it is not centered, the data matrix should be centered. Consequently, we compute the sample covariance as

$$\hat{C}_\mathbf{X} = \frac{1}{N-1}\mathbf{X}^T\mathbf{X}, \tag{16}$$

which is a square $D \times D$ matrix. Since the sample covariance matrix is symmetric, i.e., $\hat{C}_\mathbf{X} = \hat{C}_\mathbf{X}^T$, and has entries of real numbers (positive semidefinite), it can be diagonalized as

$$\hat{C}_\mathbf{X} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T, \tag{17}$$

where $\mathbf{Q}$ is the $D \times D$ orthonormal matrix of the eigenvectors of the covariance matrix, and $\mathbf{\Lambda}$ is the $D \times D$ diagonal matrix with eigenvalues λ_α. Recall that an orthonormal matrix is defined as a square matrix with rows and columns of orthonormal vectors, which means that $\mathbf{Q}^T\mathbf{Q} = \mathbf{Q}\mathbf{Q}^T = I$ and $\mathbf{Q}^T = \mathbf{Q}^{-1}$. To

obtain the sample covariance of the new basis in the $\mathbb{R}^d$ space, the covariance matrix of $\mathbf{Y}$ is

$$\hat{C}_{\mathbf{Y}} = \frac{1}{N-1}\mathbf{Y}^T\mathbf{Y} = \frac{1}{N-1}(\mathbf{XT})^T(\mathbf{XT}) = \mathbf{T}^T\frac{\mathbf{X}^T\mathbf{X}}{N-1}\mathbf{T} = \mathbf{T}^T\hat{C}_{\mathbf{X}}\mathbf{T} \qquad (18)$$

where $\hat{C}_{\mathbf{Y}}$ is a symmetric square $d \times d$ matrix. Furthermore, using the diagonalization of $\hat{C}_{\mathbf{X}}$ we obtain $\hat{C}_{\mathbf{Y}} = \mathbf{T}^T\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T\mathbf{T}$. The PCs are obtained from the eigenvectors and eigenvalues, which are computed from the sample covariance matrix projected on the new space.

2.4 Validation

2.4.1 Background

The challenge in validating the results lies in choosing an appropriate index. The validation indices we encounter can be categorized into three types, namely, unsupervised, supervised and relative. Unsupervised evaluation measures do not use external information as class labels and are divided into two kinds of measures, *cluster cohesion* and *cluster separation*, which are also called compactness and isolation, respectively. The cohesion measures how close observations in a cluster are to each other, and the separation measures how separated observations are from other clusters' observations. Supervised evaluation compares the clustering method's results to its class labels, i.e., it computes external validation. Relative evaluation can be either a supervised or an unsupervised measure that compares different clustering results. Since most of the data applied will be unsupervised, the focus will be on internal validation.

An external validation method could be to plot the true positive rate (TPR) against the false positive rate (FPR). The TPR is when the method correctly clusters the data points, while the FPR is when the method clusters dissimilar data points to the same cluster. The authors of AP validated the methods mainly based on external validation methods by evaluating the TPR and the FPR [5]. This thesis will mostly apply internal validation indices since unlabeled data will be used in most cases, and aim to measure the goodness of the clustering methods. It is essential to validate the clustering analyses since the algorithms can compute clusters even though the data does not possess a cluster structure [18]. The aim of cluster validation is to determine if there exists a non-random structure in the data. It is important to evaluate if the clustering results fit the data, decide the number of clusters, compare two types of clusterings and compare the results to class labels for the external validation. Therefore, validating results from clustering methods is as important as applying the actual methods. Most internal indices are based on either cohesion or separation but the silhouette coefficient uses both to evaluate the clustering performance, which motivates the choice of index in this thesis [17].

2.4.2 The silhouette coefficient

The silhouette coefficient is a validation method that uses the cohesion and separation to compute the coefficient. Hence, the silhouette coefficient measures how similar a data point is to its own cluster and how dissimilar it is to the other clusters. It is computed for each individual data point in three steps and therefore measures how well each observation has been classified. For each data point x_i, we compute the cohesion, separation and lastly, the silhouette value. The cohesion measures the investigated data point's distance to the other observations in its cluster. The cohesion for observation x_i is

$$a(i) = \frac{1}{|c_l| - 1} \sum_{x_k \in c_l; i \neq k} d(i, k) \tag{19}$$

where c_l is the cluster of observation x_i. It measures how well the data point x_i is assigned to its cluster. A small value indicates good results, which means that data point x_i is close to the data points in its own cluster.

The separation for observation x_i is assessed by the minimum average distance to all data points in a cluster that x_i does not belong to. The separation is given by

$$b(i) = \min_{m \neq l} \frac{1}{|c_m|} \sum_{x_k \in c_m} d(i, k) \tag{20}$$

where c_m is a cluster not containing x_i. The separation measure indicates good results when it has large values, since it measures how well a data point separates from other data points not in its cluster. By taking the minimum of the average distances of x_i to all the other points that are not in cluster c_l, only the closest cluster to x_i is considered. The goal is to acquire a large separation value, meaning that the clusters are well separated. The silhouette value of x_i is defined by

$$s(i) = \begin{cases} 1 - a(i)/b(i) & \text{if } a(i) < b(i) \\ 0 & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1 & \text{if } a(i) > b(i) \end{cases} \tag{21}$$

and can take values between -1 and 1. A negative silhouette value indicates that the cohesion is larger than the separation, meaning that the average distance to the points in the same cluster is larger than the minimum average distance to data points in other clusters, indicating a bad clustering result. It is desirable to have a cohesion $a(i)$ close to zero and a large separation $b(i)$ value, which indicates that the clusters are compact and well separated. A silhouette value close to zero indicates that the data points could belong to any of the clusters, consequently, the silhouette coefficient is computed by the average of silhouette values over all data points. The clustering method aims to obtain a silhouette coefficient close to one.

The advantage of the silhouette coefficient is its ability to evaluate each data point's clustering, since it is computed for each data point it may help evaluating which or how many data points are clustered badly. The desired values of the cohesion and separation measures are determined by their relation to each other. Their relationship follows the same notion as the within-point scatter and between-point scatter from Equation 2 and 4. The cohesion and separation measure consider all data points in the summation over distances, and not only a single pair of data points, when computing the measures for each individual data point, to represent entire clusters. Both, the cohesion and separation measure are based on the average pairwise distance which uses the Euclidean distance, meaning that they do not consider the geometrical shape of the cluster but only the observation's position in the cluster. Thus, the index may not be beneficial for arbitrarily shaped clusters nor for clusters close to each other, since the measures aim to find compact clusters that are relatively separated, meaning that other similarity measure that is able to capture the geometrical structure is needed.

2.4.3 Clustering validation index based on nearest neighbours

As a second option to a validation method, *clustering validation index based on nearest neighbours* (CVNN) is used since the original authors claim that it can handle non-spherically shaped clusters [11]. Experiments computed by the founders of the index show that CVNN outperforms other methods, such as the silhouette coefficient. However, as will be noticed later, we will discuss that CVNN may still not be able to evaluate arbitrarily shaped data. Similar to the silhouette coefficient, CVNN measures both the separation and cohesion. The introduction of the CVNN index is motivated by its separation measure, which is usually computed based on cluster centers that do not contain the information about the shape of the cluster and results in a measurement only suitable for well separated spherical clusters. On the other hand, CVNN uses multiple data points to represent a cluster, since a cluster center alone cannot reflect the cluster shape. The measure shares the same idea as the k-nearest neighbours consistency [11], namely, an observation at the center of a cluster does not contribute to the separation in contrast to an observation at the edge of a cluster, since an observation at the edge is surrounded by other clusters and connects to the observations in those clusters. The measure of separation aims to use different observations of the same cluster in different situations to reflect the geometrical shape. The separation measure is defined by

$$\mathrm{Sep}(K, k) = \max_{l=1,\dots,K} \Big(\frac{1}{N_l} \sum_{p=1,2,\dots,N_l} \frac{q_p}{k} \Big) \tag{22}$$

where K is the number of clusters, k is the number of nearest neighbours, N_l is the number of observations in cluster c_l, x_p is the p:th observation in c_l, and q_p is the number of k-nearest neighbours of x_p that are not in cluster c_l. A lower value of the measure reflects a better cluster separation. If the clusters are well

separated, and all k-nearest neighbours belong to the cluster of the investigated point, the separation measure will obtain the value zero. On the other hand, if the clusters are close to each other, the separation measure will obtain a higher value than zero. Note that the name of the measure is misleading since usually, a high separation value might indicate a better cluster separation. Moreover, the separation measure in the silhouette coefficient, given by Equation 20, is the minimum of the average pairwise distances, while the separation measure of CVNN in Equation 22 is the maximum of the average fraction of k-nearest neighbours that are not in the same cluster as the point under consideration. The separation measure of the silhouette coefficient will investigate every data point's distances to observations in other clusters, while the separation measure of CVNN will only investigate data points on the surface of the cluster since they may contribute the most to the shape of the cluster.

The second essential part of an internal validation index is the cohesion, or compactness. Instead of only using the centers as representatives of entire clusters, that lack of geometrical information, the compactness in CVNN is computed by including all distances to obtain the information for all data points. The cohesion measure of CVNN is given by [7]

$$\text{Com}(K) = \frac{\sum_l \sum_{x_i, x_k \in c_l} d(i,k)}{\sum_l N_l(N_l - 1)} \tag{23}$$

where x_i and x_k are two different observations in cluster c_l. Note that the original authors divided the sum of the within-cluster distances by $N_l(N_l - 1)$ within each cluster [11], which would not be suitable for data with different sized clusters since the clusters would be weighted equally. Equation 23 represents the average pairwise distances between observations in the same cluster, where a lower value reflects a better compactness, and the compactness usually monotonically decreases for higher number of clusters. Comparing the cohesion of CVNN in Equation 23 with the cohesion of the silhouette coefficient in Equation 19, it can be observed that both measure the average pairwise distances between data points in each cluster. Furthermore, comparing the cohesion of CVNN in Equation 23 with the loss function of k-means in Equation 6, it is easy to see that Equation 6 also measures the average pairwise distance, or the within-point scatter.

By combining the two measures the CVNN is given by

$$\text{CVNN}(K,k) = \text{Sep}_{\text{norm}}(K,k) + \text{Com}_{\text{norm}}(K) \tag{24}$$

where

$$\text{Sep}_{\text{norm}}(K,k) = \text{Sep}(K,k)/(\max_{K_{\min} \leq K \leq K_{\max}} \text{Sep}(K,k))$$

and

$$\text{Com}_{\text{norm}}(K) = \text{Com}(K)/(\max_{K_{\min} \leq K \leq K_{\max}} \text{Com}(K))$$

The separation and cohesion are normalized so they both have the same order of magnitude. A lower value of CVNN usually means a better clustering result. To obtain the minimum value of CVNN we have to compute the index for several values of k-nearest neighbours and identify the optimal number of clusters. This is, according to the authors of CVNN [11], possible since the relation between the index and k is shaped like a parabola. As k increases, the value of CVNN decreases and reaches its minimum value. When k keeps increasing after reaching its optimal value, the CVNN index will eventually increase and recommend the wrong number of clusters. The CVNN index at the curve's minimum will give the right number of clusters. This procedure will be used in the analysis. The reason why the relation between the CVNN index and k is shaped like a parabola is beyond the scope of this thesis and the interested reader can see reference [11].

The advantage of CVNN's separation measure is that it investigates nearest neighbours of all observations in a cluster, which will both illustrate the clusters and the observations' relationships to other clusters. The limitation of this method is in the choice of nearest neighbours since an assessment of k is required, which might be problematic in the case of arbitrarily shaped clusters where some data points with nearest neighbours can be further away from each other, than other points in their cluster to their nearest neighbours. Another disadvantage of CVNN concerns the cohesion measure, which is the average pairwise distance between objects. The pairwise distances are measured with the Euclidean distance, which again could not be sufficient dealing with arbitrarily shaped data. The consequence could be that CVNN is not suitable for non-spherically symmetric clusters, which will be investigated in the analysis.

3 Results

In this section, various test cases will be used to compare AP with k-means and validate their results. The data used is mostly simulated but a single case of real data will be considered with the aim of illustrating AP's properties and artifacts. The similarity measure chosen for AP is the negative squared Euclidean distance, for comparative reasons to the k-means method.

3.1 Test cases

3.1.1 Imbalanced data in number of data points

This simulated dataset consists of two clusters from Gaussian mixtures with means $\mu_1 = (0.3, 0.5)$ and $\mu_2 = (0.5, 0.3)$, and standard deviations $\sigma_1 = \sigma_2 = 0.06$. The data is imbalanced in the number of data points where the larger cluster has 1000 observations and the smaller cluster has 100 observations, which makes the ratio 10:1. In all the simulated test cases a plot of number of clusters versus shared preference for AP will be generated in order to illustrate the number of clusters for a given shared preference. The shared preference can be smaller than the smallest value of the similarity matrix. In order to find the right

shared preference, it is important to run the algorithm for several values of the shared preference to illustrate which number of clusters occurs more frequent for what shared preferences. This is illustrated in Figure 5 where it can be observed that the widest range of shared preferences occurs at two exemplars. As mentioned before, these values are computed based on a shared preference range where the smallest value is smaller than the smallest similarity. Subsequently, this range is split into 200 intervals and for each shared preference the algorithm is run, which makes the computation of this plot a computationally expensive procedure. As a convention, the first shared preference that gives the wanted number of clusters is used in AP. Figure 6a illustrates the identified clusters for the Gaussian mixture data using AP. Note in Figure 5 how it oscillates between three and four clusters despite the noise added to the similarity matrix and the damped factor of $\lambda = 0.9$, which indicates that the algorithm struggles to converge.

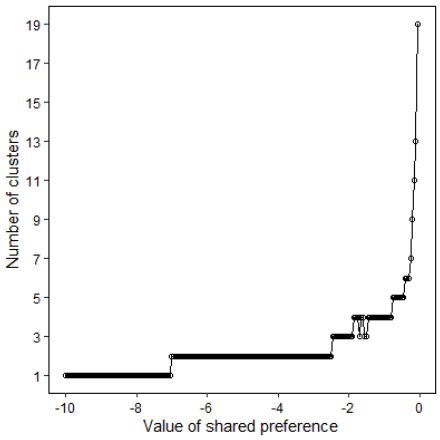
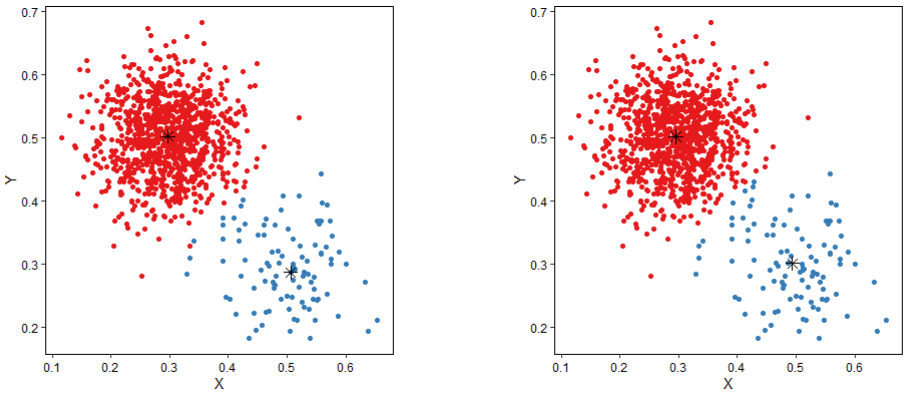


Figure 5: Shows effect of the shared preference on the number of exemplars of the imbalanced data in number of data points.

Nevertheless, after obtaining the right shared preference the application of AP is computationally fast in this case. Furthermore, the application of k-means does not take long time either even though we test multiple initial centroids and choose the one with the lowest sum of squared errors. As demonstrated in Figure 6, both methods give similar and reasonable results. Both methods apply the squared Euclidean distance as dissimilarity measure and have centroids or exemplars positioned in the middle of the clusters. Moreover, choosing the centroid as the mean of the cluster's data points or an actual data point does not affect the results. The spatial extension for both clusters is similar, meaning that the algorithms interpret the clusters as balanced in spatial size, which is reasonable since the clusters have almost the same radius. Furthermore, when overclustering, i.e., obtaining too many clusters such as three and four clusters, both methods split the larger cluster. This is due to their tendency of creating



(a) (b)

Figure 6: The two clusters of the imbalanced data in number of data points identified in (a) and (b) by AP and k-means, respectively, are coloured in red and blue. The means of the Gaussians are $\mu_1 = (0.3, 0.5)$ and $\mu_2 = (0.5, 0.3)$, and the standard deviations are $\sigma_1 = \sigma_2 = 0.06$. (X, Y) are 2D features. The centers of the two Gaussians are indicated by the star-symbols.

balanced clusters, which in this case will be in number of data points.

3.1.2 Imbalanced data in spatial extension

The imbalanced data in case of spatial extension consists of two clusters with 200 observations each from Gaussian mixtures with means $\mu_1 = (1, 1)$ and $\mu_2 = (1.7, 1.7)$, and standard deviations $\sigma_1 = 0.5$ and $\sigma_2 = 0.1$. The standard deviations give a spatial extension ratio of 5:1. The shared preference's effect on the number of clusters is illustrated in Figure 7, which shows that the suitable number of clusters is two since it has the widest range of shared preferences. This is not as computational expensive as with the previous case of imbalanced data, since the number of observations are only 400 compared to 1100. Consider the fact that oscillations seem to occur between four and five clusters, which means that AP struggles to converge.

The computational time of both AP and k-means is relatively fast and the methods compute very similar results. From Figure 8a and 8b it can be seen that both methods seek to create balanced spherical clusters, which yields bad clustering results, as the imbalance in spatial extent is not considered in the loss function of k-means in Equation 6, and not in the updating rules Equation 7 and 9 in Algorithm 2 of AP. The only visible difference is the choice of centroids and exemplars, where the centroid of the large cluster in Figure 8b is more centralized, while the exemplar of the same cluster in Figure 8a is more to

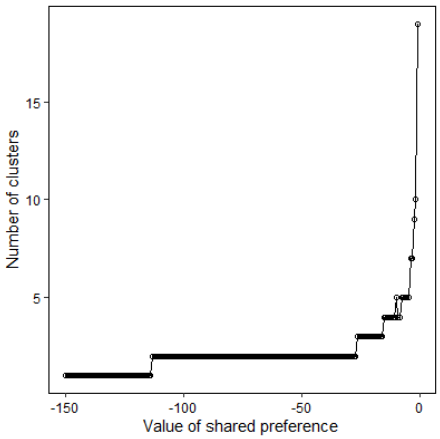
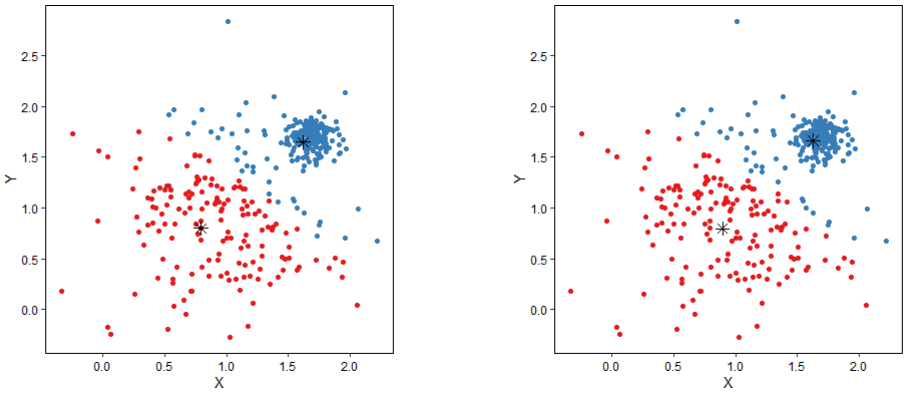


Figure 7: Shows effect of the shared preference on the number of exemplars of the imbalanced data in spatial extension.



(a) (b)

Figure 8: The two clusters of the imbalanced data in spatial extension identified in (a) and (b) by AP and k-means, respectively, are coloured in red and blue. The means of the Gaussian are $\mu_1 = (1, 1)$ and $\mu_2 = (1.7, 1.7)$, and the standard deviations are $\sigma_1 = 0.5$ and $\sigma_2 = 0.1$. (X, Y) are 2D features. The centers of the two Gaussians are indicated by the star-symbols.

the left aiming at a data point. This is a consequence of AP that uses real data points as centers while k-means uses the mean of the cluster. However, it does not create a great difference in the clustering. Despite the fact that AP is based on networks and message-passing quantities, it works similarly as k-means due to the similarity measure. The squared Euclidean distance in k-means does not aim to minimize the sum of squared distances of each cluster separately, but for all clusters together, meaning that it creates clusters of balanced sizes. Therefore, the Euclidean distance cannot distinguish imbalanced clusters in spatial extent and that is why the large cluster is almost splitted in half. In the case of AP, the same applies when using Euclidean distances since the message-passing quantities will as well choose the shortest distance between the data points to send their information. This concept is illustrated in Figure 9 where it can be observed that the shortest distance to the center of cluster B is not from its edge, but from the red line in the half way. Consequently, this red line determines the extent of cluster A since the splitting of the clusters is, according to the Euclidean distance, beneficial for the entire sum of squared distances by creating balanced clusters. In order to obtain better clustering results of the imbalanced data in spatial extent, we suggest using the mutual k-nearest neighbour graph from Section 2.2.2.1 in AP, since the data is Euclidean but imbalanced. The mutual k-nearest neighbours will connect data points that are each other nearest neighbours and may cluster the data better than the Euclidean distance.

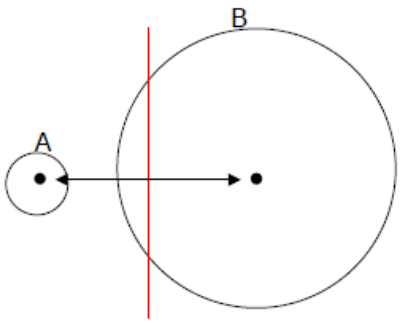


Figure 9: An illustrative figure of the Euclidean distance in case of imbalanced data in spatial extent. The clusters are named A and B, where B is the larger cluster in spatial extent. The centers of the clusters are indicated by the black dots. The red line going through cluster B is located at the half way between the two cluster centers. The distances from the red line to the centers are illustrated by the black arrows.

In case of overclustering with, e.g., three or four clusters for both methods, the large cluster is split into multiple clusters with equally sized spherical clusters. Depending on if k-means converges or not, it can sometimes split the smaller cluster. Hence, the importance lies in computing multiple runs to sample

different initial centroids and the global minimum of the loss function is reached.

3.1.3 Spherical data with noise

The dataset applied in this section consists of two spherical clusters with 70 observations each from Gaussian mixtures, with means $\mu_1 = (0.3, 0.3)$ and $\mu_2 = (0.7, 0.7)$, and standard deviations $\sigma_1 = \sigma_2 = 0.04$. Additionally, the data consists of 50 observations sampled from the uniform distribution, considered as noise. The shared preference's effect on the number of clusters is illustrated in Figure 10, which shows that AP interprets the dataset as two clusters and finds structure in the noise as illustrated in Figure 11a. On the other hand, as expected, the k-means finds structure in the noise and creates two large clusters that cut the data points in half as illustrated in Figure 11b, with the attempt to identify spherically shaped clusters.

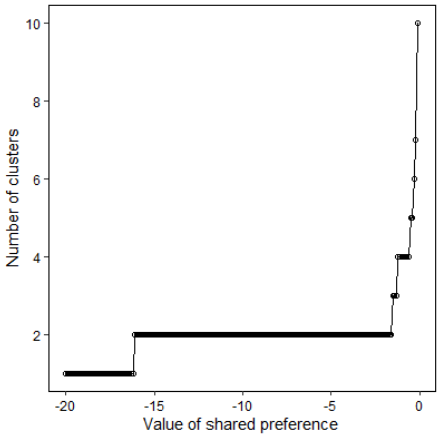
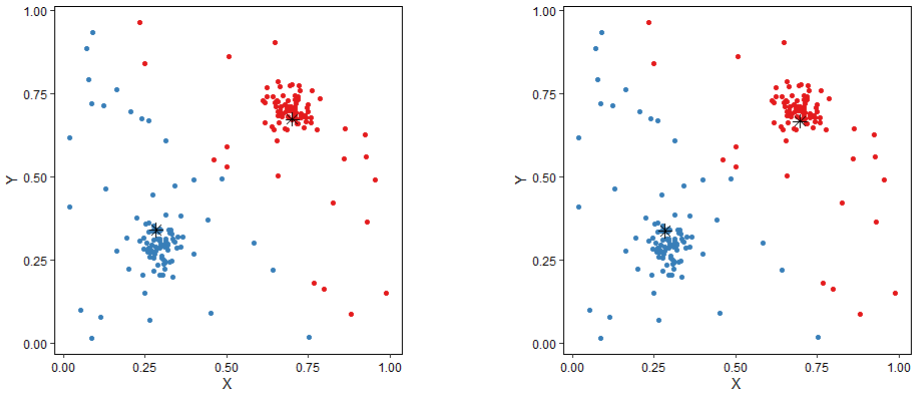


Figure 10: Shows effect of the shared preference on the number of exemplars of the data with noise.

Both AP and k-means cannot distinguish outliers or noise from clusters due to not having an outlier or noise identification procedure. However, we already know that k-means is sensitive to noise and outliers since extreme values influence the means. The influence is due to the centroid being the average of the data points close to the centroid. Hence, by having noise, the centroid will be pushed closer to the noise trying to create spherical clusters. AP considers every data point as a potential exemplar and does not have a noise identification procedure. Therefore, an introduction of density-based methods that take into account density information would be more suitable for data with noise. In the case of overclustering, the methods continue to find structure in the noise.



(a) (b)

Figure 11: The two clusters of the data with noise identified in (a) and (b) by AP and k-means, respectively, are coloured in blue and red. The means of the Gaussian are $\mu_1 = (0.3, 0.3)$ and $\mu_2 = (0.7, 0.7)$, and standard deviations are $\sigma_1 = \sigma_2 = 0.04$. The noise is from the standard uniform distribution. (X, Y) are 2D features. The centers of the two clusters are indicated by the star-symbols.

3.1.4 Flame-shaped data

The flame-shaped dataset is often used in clustering analyses to investigate how good clustering methods handle non-spherical clusters. It consists of 240 observations in two dimensions and is shaped like a flame. The data has two clusters and two outliers. Figure 12 shows that the appropriate number of clusters for AP is two clusters, and it can also be noticed that oscillations occur between two and three clusters, and also between three and four clusters. This means that despite our noise and damped factor the algorithm still fails to converge for some shared preferences.

Figure 13a and 13b show that AP and k-means again perform similarly. With the attempt to identify spherical clusters due to the use of Euclidean distances, they cannot cluster arbitrarily shaped data. This behaviour was expected from both algorithms since the similarity measure cannot distinguish the actual short distances between data points. An illustrative example of this problem is presented in Figure 14, where the Euclidean distance will not interpret the U-shaped cluster and its distances correctly. The distance from point C to A should be considered to be shorter than from C to B, which is not the case when using the Euclidean distance. Therefore, the algorithm assigns the points with shorter distance to the same cluster. The same applies to the flame-shaped dataset where the spherical cluster should be one cluster, and the moon-shaped cluster should be another. One suitable similarity measure could be the *commute time distance* (CTD), which is based on the random walk [12]. The discussion about the CTD is beyond the scope of this thesis, and the inter-

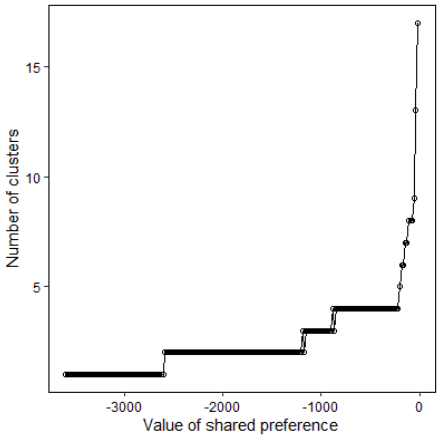
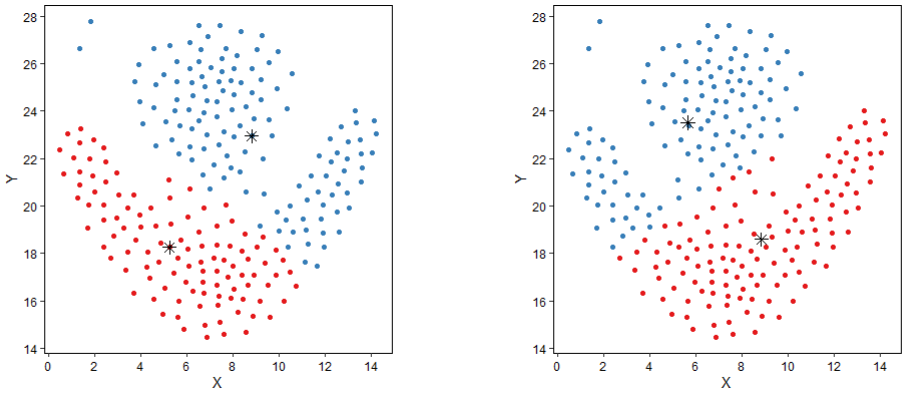


Figure 12: Shows effect of the shared preference on the number of exemplars of the flame dataset.



(a) (b)

Figure 13: The clusters of the flame dataset identified in (a) and (b) by AP and k-means, respectively, are coloured in blue and red. (X, Y) are 2D features. The centers of the two clusters are indicated by the star-symbols.

ested reader can see reference [12].

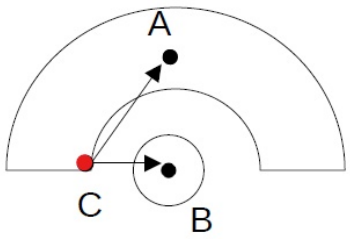


Figure 14: An illustrative figure of the problem of using the Euclidean distance in case of arbitrarily shaped data. The clusters consist of an arbitrarily shaped cluster and a spherical cluster. The black dots are named A and B. The distances from the red dot C to A and B, are illustrated by the black arrows.

In the case of overclustering, the moon-shaped cluster is splitted since its spatial extension is larger than the spherical cluster.

3.1.5 MNIST dataset

The MNIST dataset of handwritten digits will be used in this section and was first introduced by Yann LeCun [10]. The MNIST data is used since it is multidimensional real data, which is a type of data not investigated before in this thesis. The data consists of 60000 training images and 10000 test images with pixel values between 0 and 255, and their corresponding training and test labels. With the purpose of saving computational time, we only considered a sample of the first 6000 observations of the training data, which is balanced in the number of samples among the digits. Each image displays a digit, the label, between 0 and 9 with a $28 \times 28 = 784$ pixel resolution. From observing the data it can be noticed that some images are rotated or flipped, which will require a preprocessing of the images in order to rotate them in the correct orientation. The adjustment is performed by computing the Euclidean distance between each image and its corresponding label's mean. The comparison is made in eight ways where the mean to the rotated versions of the picture in 0, 90, 180, and 270 degrees, is first compared. Secondly, the image is flipped and for the same rotations the Euclidean distances to the digit's mean are computed. The orientation with the smallest Euclidean distance to the mean is assumed to be the "correct" image orientation, i.e., presented in a digit's natural form. Nevertheless, there still exists some ambiguity since the image with the smallest Euclidean distance to the mean is not always the correct orientation. In particular, since some images are inclined or unclear as illustrated in Figure 15.

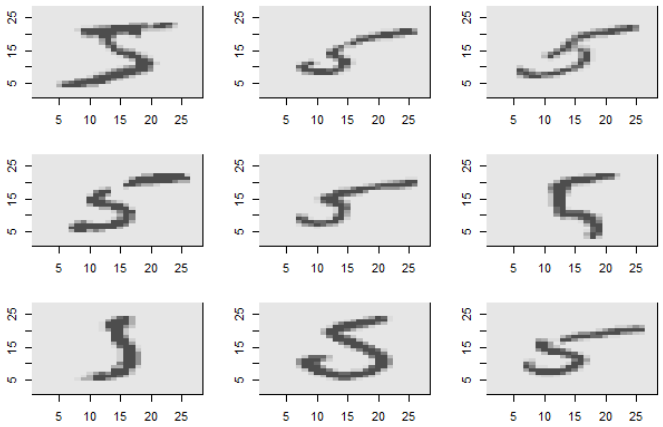


Figure 15: Images of differently handwritten 5s from the MNIST dataset. The axes show the pixel resolution of 28×28 pixels.

The MNIST dataset will be normalized to bring the values of all dimensions to a common scale, i.e., between 0 and 1. Furthermore, an exploratory analysis as described by David Robinson [16] will be performed, starting by changing the data representation. The aim is to create a data frame where each observation represents one pixel for a particular image. Two new variables, x and y, will be created to represent the frame of the images. These will be computed with some arithmetic as presented by Robinson [16]. Most of the pixel data consists of zeroes which represent the white background of each image, where only relatively few pixels are considered as black, i.e., with the pixel value 1, and the few pixels in between are in grey. Figure 16 shows how relatively few pixels are grey.

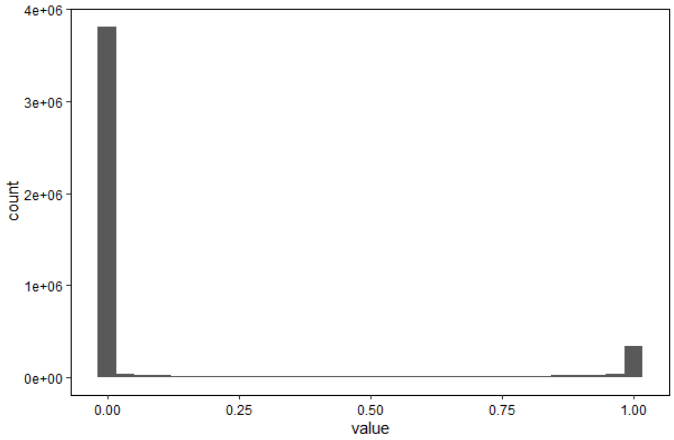


Figure 16: The frequency of pixel values of the MNIST data. The x-axis is the pixel values and the y-axis is the number of pixels with a particular pixel value.

Next, the mean values of each digit in each dimension are computed. The centroids are visualized in Figure 17 where the average value of each pixel is illustrated for each digit. This figure gives an intuitive idea on which digits will be harder to cluster. For instance, the digit zero and one might be easier to cluster since they will on average consist of digits that are white or dark in the middle, respectively, and the left and right edges will usually be dark for the zeroes and white for the ones. It could also happen that some digits can be confused by their similar appearance, which could be the case for the digits three and eight, four and nine, and seven and nine. Moreover, as a direct consequence of a wrong rotation or flipping of the digits six and nine, they could also be confused. However, this is an issue even for the human eye.

Figure 17: The centroids of each digit of the MNIST data where each pixel is the average pixel value of the particular digit. (X, Y) represent the frame of each image and consist of 28 pixels each. The legend shows the grey-scale of every average pixel value.

The problem in clustering is that all images with the same label do not look similar. To investigate this variation of appearance, a boxplot of the Euclidean distance of each image to its digit's centroid was generated, which is illustrated in Figure 18. It can be observed that digit one has relatively less variation to the centroid while zero and two have more variability. All of the digits have outliers far from their centroids. It can also be noticed from Figure 18 that the distribution of the Euclidean distances from the digits' centroids are skewed, especially for digit nine, which might come from the fact that the clusters have non-uniform density and arbitrarily shape. We illustrate this using the example in Figure 14, it is evident that the centroid of the U-shaped cluster falls in between point A and B in the figure and the Euclidean distance of a point in the U-shaped cluster to this centroid does not have a symmetric distribution

around its median. The most extreme case of outliers is the digit one, which is usually written as a straight vertical line. It could also be drawn with a inclined line on top of the vertical line, and a horizontal line under the vertical line. Moreover, except for these different types of writing styles, there are some abnormally written digits. This variation in writing could be the reason of the many and extreme cases of outliers. Nevertheless, the distance from the smallest value to the median of the boxplot is quite large in all cases, and the distance might be caused from the great variability among the images, which could be due to the variation in writing. There is also a great gap between the smallest value of each digit's boxplot and zero, which could happen when the centroid is outside the cluster. For instance, if the U-shaped cluster in Figure 14 had data point B as its centroid, it would obtain this gap from the distance between the cluster's closest point to the centroid and the centroid. This gap could also be illustrated in Figure 19 when plotting the Euclidean distances of each image from the centroid of the corresponding digit in histograms, where the first bar is far from zero. Hence, the data could be non-spherical symmetric with a misplaced centroid since the Euclidean distance is used. A misplaced centroid means that it is outside its cluster and inside another cluster. Figure 14 could illustrate this if we again consider the data point B as the U-shaped cluster's centroid. This centroid is misplaced since it is inside the spherical cluster, and the Euclidean distance will consider it closer to the spherical cluster than to the U-shaped cluster.

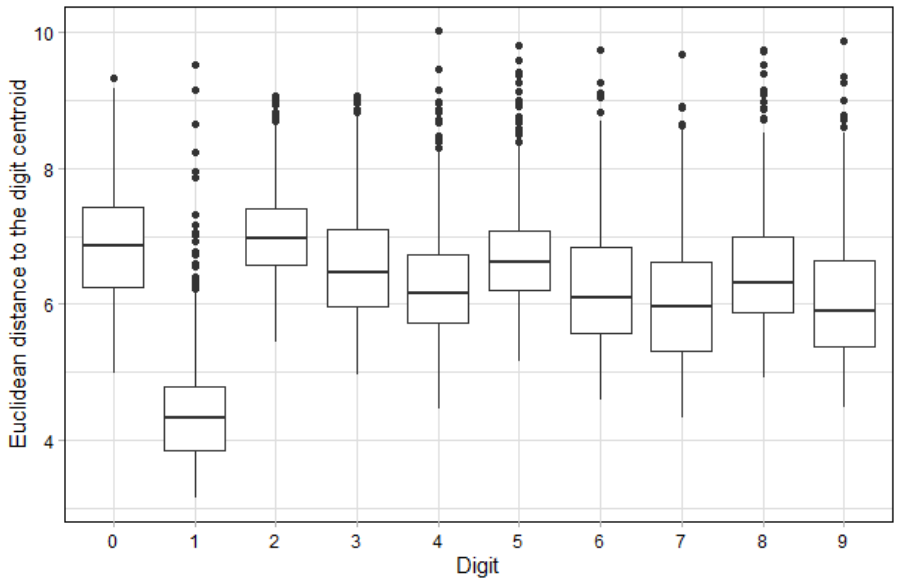


Figure 18: The Euclidean distance of each image of the MNIST data to its label's centroid presented in boxplots. The x-axis presents the digits and the y-axis presents the Euclidean distance to each digit's centroid.

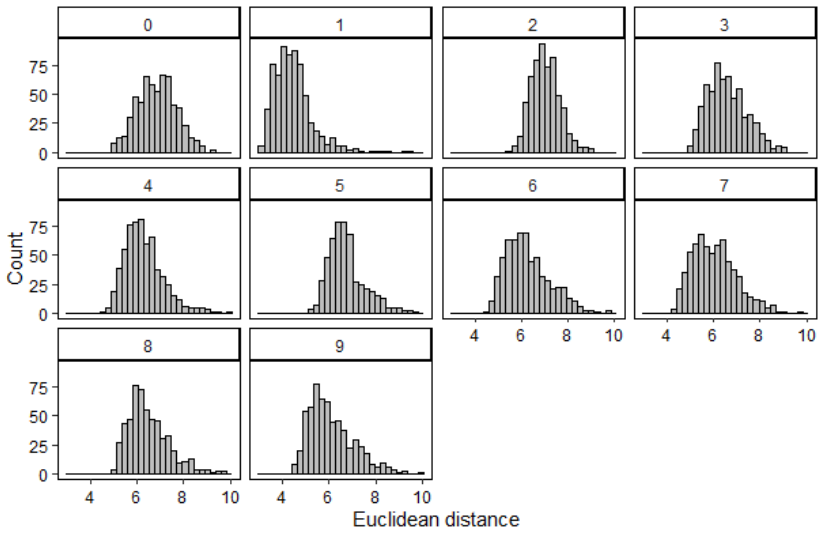


Figure 19: The Euclidean distance of each image of the MNIST data to its label's centroid presented in histograms. The x-axis presents the Euclidean distance and the y-axis presents the number of images with that distance.

The reason for the large gap between the smallest values of the Euclidean distances and zero could be investigated by dimensionality reduction. To investigate if the multidimensional dataset is non-spherical symmetric, PCA could be used. In case of PCA, the first few eigenvalues carry most of the variance, which means that they reflect the importance of the eigenvectors. The larger the eigenvalue the more important is that dimension. From Figure 20 we decided to look at the first three PCs since they capture a big portion of the variation, and for the simplicity of displaying the results. Notice that there is a slight variation in eigenvalues for each digit. Therefore, we will investigate the first three PCs for each digit separately.

PCA is applied to images belonging to each digit separately since each digit obtains differently shaped and varied PCs. For instance, Figure 32 of digit one and Figure 38 of digit four in Appendix B illustrate that the variation of the PCs of each digit is distributed differently, and the PCs try to capture the organisation of the clusters, making it more suitable to apply PCA to images of each digit separately. In Appendix B, Figure 30-49 illustrate both, the spatial extensions of the first three PCs of each digit and their distributions. Figure 30 illustrates how the PCs for digit zero could be spherical symmetric compared to, e.g., digit one in Figure 32. Figure 31 of digit zero illustrates how the distributions of the PCs are non-skewed. Furthermore, from the other figures it can be clearly seen that the PCs for digit one, two, four, five, six, seven and nine are non-spherical symmetric and their histograms show quite skewed results. Hence, the spatial extension is not the same among the clusters. Figure 36-37 and 46-47

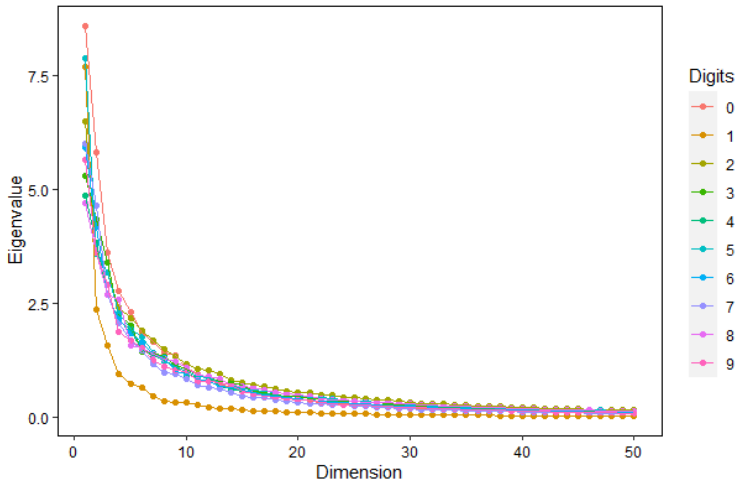


Figure 20: The eigenvalues of each digit of the MNIST data in each dimension. The x-axis presents the dimensions and the y-axis presents the eigenvalues of the covariance matrix of each digit. The legend gives the colour of each digit's line.

show relatively spherical results for digit three and eight. However, the results show outliers and have a higher density in some areas than others. Moreover, take into account that in this example only three out of the 784 dimensions of the dataset are considered, and that a cluster being spherical symmetric in three dimensional projected PC space does not necessarily mean that it is spherical symmetric. The histograms of Figure 37 and 47 also indicate skewed results to some extent. To summarize, the PCA mostly shows that the digits' PCs are non-spherical symmetric. Generally, the Euclidean distance imply that the data points fall into a linear manifold, which is usually good for local regions. For data points separated by large distances, the Euclidean distance fails to capture the global non-linearity in the data structure. Examples when the large pairwise distances appear are outliers and unevenly distributed data in large spatial extent, which seems to appear in the MNIST dataset.

Next, focus will be put on investigating AP, with the negative squared Euclidean distance as similarity measure, and k-means applied to the MNIST data as well as the possible drawbacks of applying the methods. Both methods are applied to the fully normalized, and not standardized data. Figure 21 presents a contingency table produced using the k-means clustering on the MNIST data, with the purpose of investigating if the method clusters correctly with the squared Euclidean distance as dissimilarity measure. From the contingency table it can be observed that the algorithm has split digit one and nine into two clusters, respectively. Digit seven can also be interpreted as split into two clusters. Thus, despite the comparisons of rotated and flipped images,

k-means could not distinguish some images with the same label yielding the conclusion that the method could be inappropriate. Figure 21 also shows that there are ten clusters with centroids of almost all digits except for digit four and seven. From the ninth cluster in Figure 21, also note that k-means has not been able to cluster four, seven and nine. These are digits that could be confused with each other due to their similar appearances. The previous results obtained by applying PCA, suggested that the data is non-spherical in the projected space which makes it harder to cluster with an Euclidean measure. Figure 21 highlights, when comparing to the rest of the clusters, that the method shows a poor performance on the fourth and fifth cluster. We also notice that the fourth and ninth clusters are mostly spread among four, seven and nine. However, most of the observations in cluster four consist of fours. The fifth cluster has most observations from digit one, but also a large amount of observations labeled as a five. These are two digits very dissimilar both in appearance and in boxplots, as visualized in Figure 18. Their bad clustering could be due to overlapping data, which is a subject not covered in this thesis but discussed in later section since it requires another kind of clustering.

The data used in this experiment consists of approximately 600 observations per digit and 6000 observations in total. Notice in Figure 21 that the largest cluster is given by the fourth cluster, while the smallest by the seventh, with a number of observations of 966 and 365 respectively, which results in a large difference from the expected cluster size of 600 data points. This is a problem possibly caused by the data's underlying characteristics, e.g., digits that are similar in appearance, digits with non-spherical clusters and the imbalance of spatial extent of the true clusters, making it difficult for k-means to cluster the images.

The clustering result from AP on the MNIST dataset is presented in Figure 22. Since the AP algorithm has exemplars that are real data points, they can be found by using the exemplars' indices, where the found exemplars are the digits one, zero, four, six, one, four, three, eight, zero and seven, respectively. Alike k-means, AP has also split digits into two clusters, which are the digits zero, one and four. In the case of digit seven, it is clear from Figure 22 that almost all images corresponding to this class are split between cluster two and nine, meaning that AP cannot distinguish this digit from others. From the previous results on the MNIST dataset, it can be concluded that AP performed as bad as k-means in this aspect which is probably due to the fact that the same similarity measure is used. The zeroth cluster has digit one as an exemplar but digits such as eight also belong to it, which is counterintuitive to the reasoning that the digits in the same cluster should be similar in appearance to each other. Referring to cluster four, it also has digit one as exemplar but is clustered quite bad considering the simplicity and uniqueness in the appearance of the digit. The clusters with the exemplar four are the second and fifth cluster, where the latter is slightly better clustered. The k-means also has problems with clustering the digit nine, possibly caused by its similar appearance to digit four.

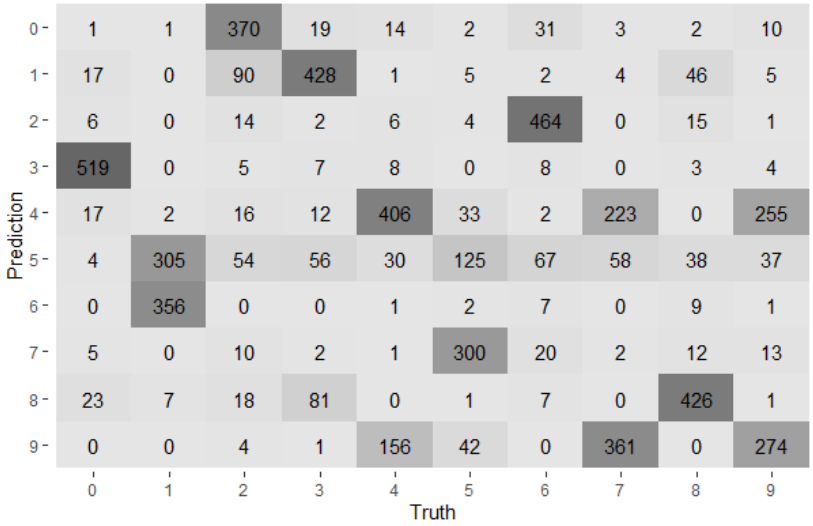


Figure 21: The contingency table of the clusters and labels after applying k-means on the MNIST data. The x-axis presents the distribution of each digit and the y-axis presents the clustering results of ten clusters.

However, the problem remained between digit four, seven and nine as opposed to cluster two with AP that cannot distinguish between almost any digit. This may be caused by the processed dataset where every image has been compared to its corresponding mean, which is the centroid of k-means. When the centroid and exemplar is six, cluster three of AP in Figure 22 is clustered worse than cluster two of k-means in Figure 21. By having three as centroid and exemplar, the results show similarly bad clusters in the sixth cluster of AP and the first cluster of k-means. Results show that the similarity between digits can yield to confusion when clustering the data, e.g., digit three can be confused with digit two and digit seven can be confused with digit nine, as shown in Figure 22.

Compared to k-means, the uneven cluster sizes resulted from AP are even larger since the largest cluster consists of 1115 observations while the smallest cluster consists of 302 observations, resulting in the largest difference from the expected cluster size with 515 observations. Hence, AP does not seem to balance the clusters which could be due to the data's underlying characteristics, e.g., digits that are similar in appearance, digits with non-spherical clusters and the imbalance of spatial extent of the true clusters. Another problem when using both k-means and AP is the similarity measure. The relatively bad clustering results could be due to the data being non-spherical symmetric as shown by the PCA. Using the CTD as similarity measure in AP would probably improve the results since it can learn about the shape of the clusters, making it work for non-spherical symmetric data.

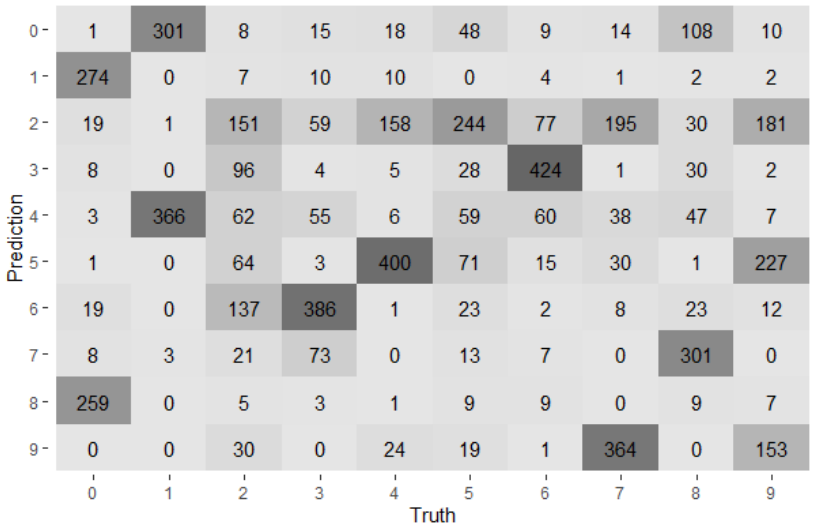


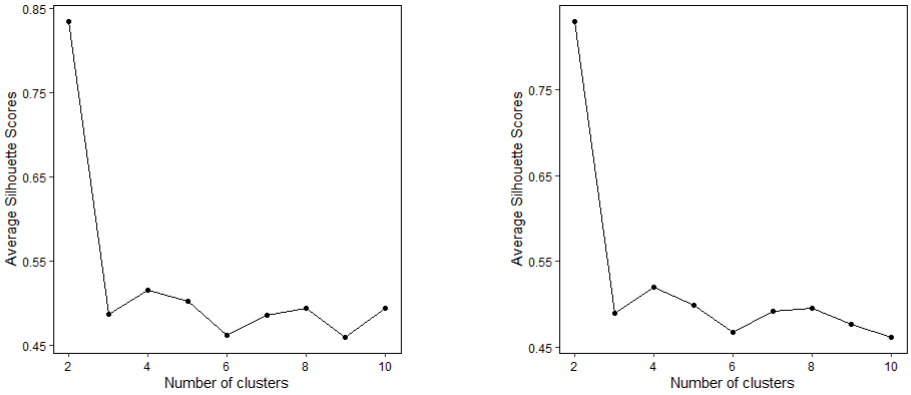
Figure 22: The contingency table of the clusters and labels after applying AP on the MNIST data. The x-axis presents the distribution of each digit and the y-axis presents the clustering results of ten clusters.

The computation time AP and k-means require relative to each other depends on how many runs the k-means method is computed. In this case, k-means is run 20 times and each run gives slightly different results due to the trapping in local minimum. AP requires to find the right shared preference to obtain ten clusters, which from prior knowledge of the MNIST dataset is known to be ten. Hence, it is computationally expensive to compute both the similarity matrix and testing different shared preferences to obtain the right number of clusters. However, this experiment compares the AP method with the k-means method for only 20 iterations, if 1000 iterations were considered instead, the results of the k-means algorithm will probably take much longer time. Finally, from the exploratory analysis presented in Figure 30-49 in Appendix B, the MNIST data is non-Euclidean and non-spherical which means that the squared Euclidean distance is an unsuitable similarity measure. Therefore, the algorithms cannot distinguish digits from each other and split the same digits into two clusters.

3.2 Validation

In this section the results of the test cases previously performed will be validated by applying different validation methods depending on the simulated dataset. In the case of the noisy and imbalanced data, in both the number of data point and spatial extent, the results will be validated using the silhouette coefficient, while in the case of the arbitrarily shaped data the CVNN validation will be

applied instead, since CVNN is claimed to be able to validate non-spherical data better. From Figure 6a and 6b it is clear that the methods have been able to cluster the imbalanced data in number of data points quite well. This is also reflected in the silhouette coefficients of both methods in Figure 23 where it can be observed that the highest values are obtained for two clusters. The silhouette coefficient for AP is 0.835 while k-means obtains the coefficient value 0.830. From the results in Figure 23 it can be concluded that AP performs slightly better than k-means but the difference is so small that it is not relevant considering the similar clustering results.

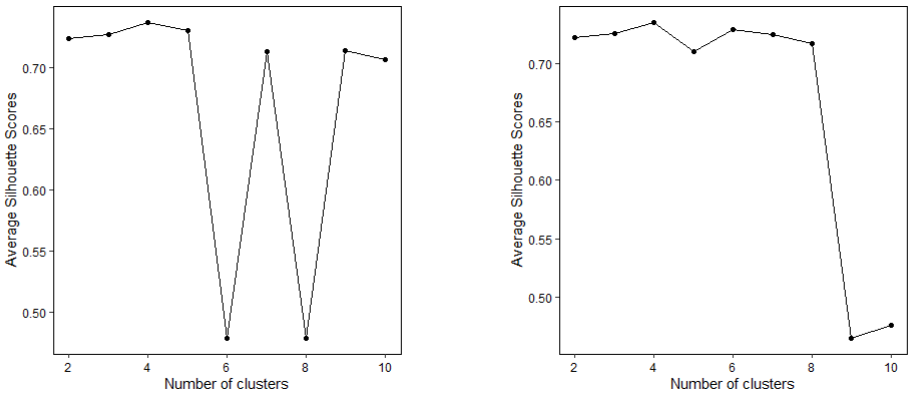


(a) (b)

Figure 23: The silhouette coefficients for different number of clusters in (a) and (b) using AP and k-means clusterings, respectively, applied on the imbalanced data in number of data points.

The imbalanced data in case of spatial extension is, as observed in Figure 8a and 8b and discussed before, not clustered well. From Figure 24 it can be noticed that the highest silhouette coefficient for both methods is reached in the case of four clusters, which is a sign of overclustering due to the bad clustering results. Considering that the Euclidean distance is not suitable for the data when clustering, the silhouette coefficient may not be suitable for the data when validating since it works for spherically shaped clusters when the Euclidean distance is used as the dissimilarity measure. Silhouette plots are useful as a mean for evaluating the method's performance as well as for visualizing the amount of points that were incorrectly clustered, these plots are presented for both methods in Figure 25. A value close to one indicates that the data point is clustered well, and a negative value means that the data point is wrongly clustered. Figure 25b illustrates the poor performance of the k-means clustering method, as many data points are not clustered correctly, which is also the case for AP illustrated in Figure 25a. The silhouette plots show that the smaller cluster contains wrongly clustered data points, these either have a small

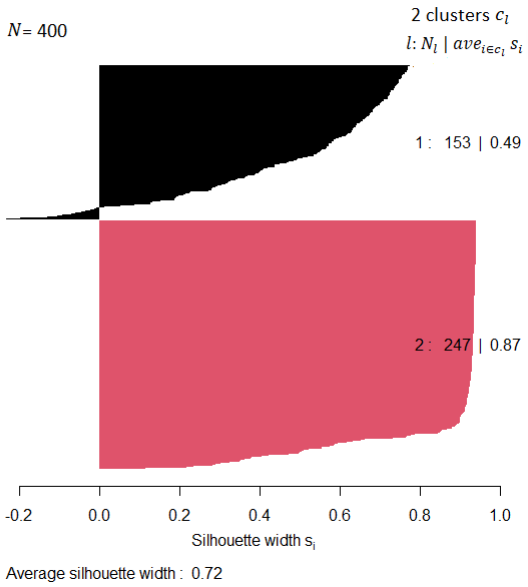
silhouette value, or even worse, a negative value, which means that the within-cluster distances are larger than the between-cluster distances. As previously mentioned, the original clusters consist of 200 observations each, so the cluster with wrongly clustered data points is actually the larger one (in quantity), which according to Figure 25 has data points with small silhouette values. The reason for the negative silhouette values in the smaller cluster (in quantity) in Figure 8, as shown in Figure 25, is due to it having data points that are further away from the points within the cluster than to the points in the other cluster, which may be caused by the bad clustering results and may also be the reason of the wrongly suggested number of clusters from the index.



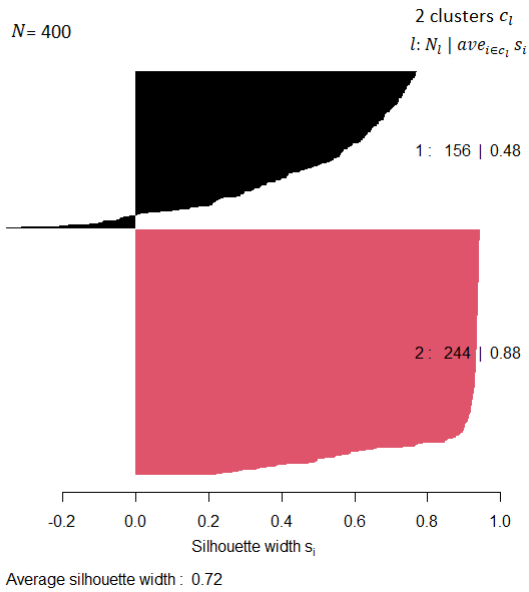
(a) (b)

Figure 24: The silhouette coefficients for different number of clusters in (a) and (b) using AP and k-means clusterings, respectively, applied on the imbalanced data in spatial extension.

Next, the noisy data with two spherical clusters is as presented in Figure 11 not clustered well, since the methods have found structure in random noise. Figure 26 illustrates how the silhouette coefficient fails to identify the right number of clusters in the data, where instead of four, the right number of clusters is two. AP and k-means are, as mentioned, not density-based methods so the methods fail to distinguish different densities, which is also reflected in the results of the validation index. From Figure 27 it can be concluded that both methods present similar clustering and validation results. The silhouette plots also show that some of the data points seem to be close to zero, which may indicate that they are noise, as it is uncertain which cluster these points should belong to. The noise is evenly spread among the data, so according to the methods that aim to create balanced spherical clusters, they can belong to either one.

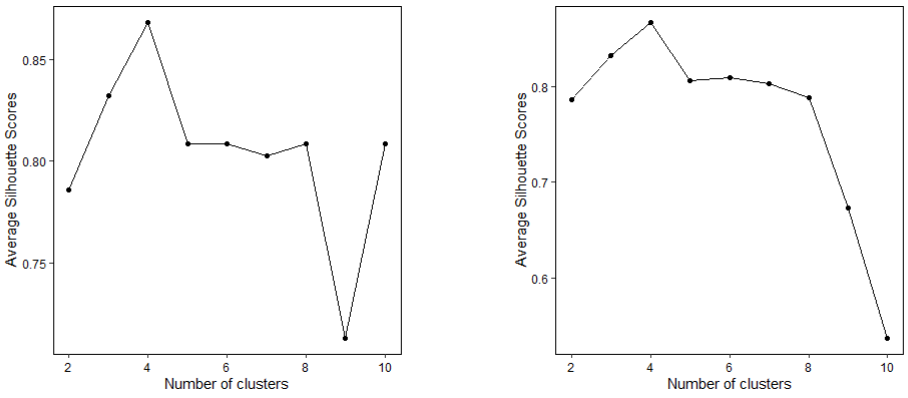


(a)



(b)

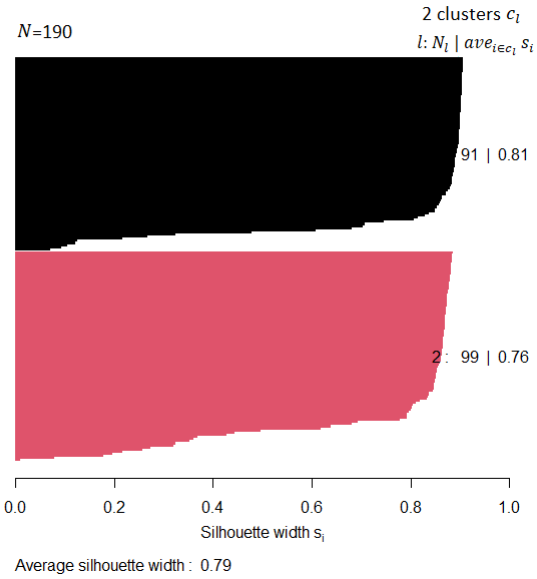
Figure 25: The silhouette plots (a) and (b) for two clusters using AP and k-means, respectively, applied on the imbalanced data in spatial extension. The number of observations in the data is N and N_l is the number of observations in cluster c_l. The silhouette value of data point i is s_i. The x-axis presents the silhouette values of each data point, obtained from Equation 21.



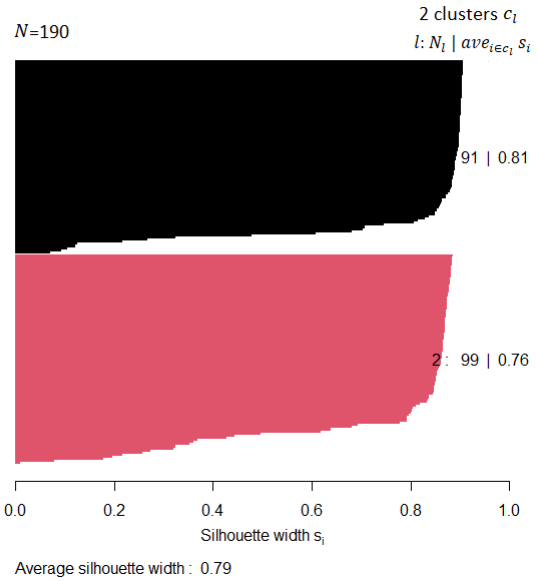
(a) (b)

Figure 26: The silhouette coefficients for different number of clusters in (a) and (b) using AP and k-means clusterings, respectively, applied on the noisy data with two spherical clusters.

Referring to the flame-shaped dataset, the validation is done by applying the CVNN method instead of the silhouette coefficient, motivated by its claimed ability of handling non-spherical data. As mentioned in previous sections, one of the main shortcomings of the CVNN method relates to choosing the "correct" number of nearest neighbours. In [11] the authors of CVNN suggest that a CVNN-index plot, i.e., plotting the CVNN-index for different number of nearest neighbours, is a suitable approach for finding this value. According to the authors' findings, the CVNN-index plot creates a parabola where the correct number of nearest neighbours corresponds to the smallest CVNN-index value. As mentioned before, the reason is beyond the scope of this thesis. Figure 28 illustrates the CVNN-index plot corresponding to the flame-shaped dataset, where the CVNN-indices are obtained for the clustering results of two clusters. Notice that the results from the plots are more shaped like hooks than parabolas, which could be due to the wrongly clustered data. Accordingly, the correct number of nearest neighbours should be one for the AP method, and 22 for k-means. The resulting number of nearest neighbours for AP is not reasonable, so we compute the nearest neighbours for the second smallest index, which gives 8 nearest neighbours. Table 6 presents the indices of two, three, four and five clusters for both methods, with their respective suitable number of nearest neighbours, where the smallest index is obtained for four clusters for both methods. Note that the chosen number of nearest neighbours is taken for the right number of clusters, i.e., two clusters. However, according to the approach suggested in [11] the right number of clusters should be four, which could be due to the used similarity measure when clustering. The index has chosen smaller clusters which could be due to the cohesion measure that aims to minimize the distances between observations within the clusters. Both the compactness and the separation of CVNN are measures based on the Euclidean



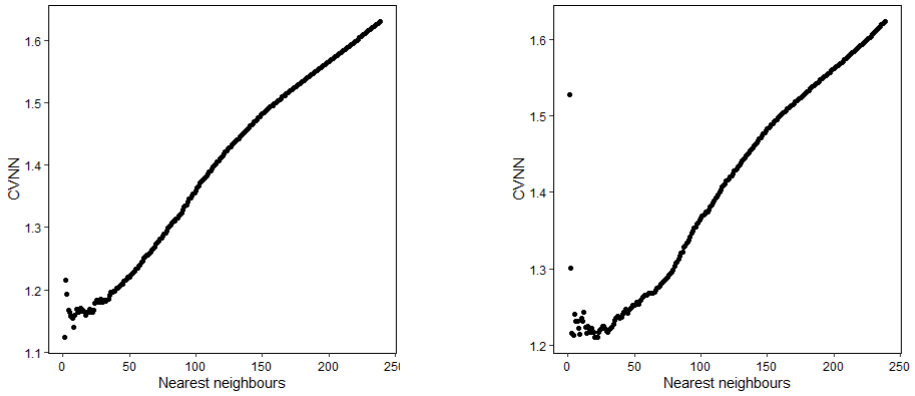
(a)



(b)

Figure 27: The silhouette plots (a) and (b) for two clusters using AP and k-means, respectively, applied on the noisy data with two spherical clusters. The number of observations in the data is N and N_l is the number of observations in cluster c_l. The silhouette value of data point i is s_i. The x-axis presents the silhouette values of each data point, obtained from Equation 21.

distance. Therefore, this method is questionable but might obtain more reasonable results if, e.g., AP is applied in terms of a non-Euclidean similarity measure and computed a reasonable clustering.



(a) (b)

Figure 28: The CVNN index on the flame-shaped dataset obtained from two clusters, for different number of nearest neighbours in (a) and (b) using AP and k-means clusterings, respectively.

	Two	Three	Four	Five
CVNN AP (8 nn)	1.140	1.180	0.950	1.334
CVNN k-means (22 nn)	1.210	1.084	0.917	1.334

Table 6: The CVNN index on the flame-shaped dataset for four different clusterings for AP and k-means, respectively, with their chosen nearest neighbours. The chosen numbers of nearest neighbours are presented in the parentheses.

3.3 Artifacts

The aim of this section is to discuss some artifacts that appeared in the test cases when applying the AP method with the negative squared Euclidean distance as similarity measure.

3.3.1 Confuses clusters for imbalance in spatial extension

The AP method with negative squared Euclidean distance as similarity measure performs poorly when clustering imbalanced data in spatial extension, as using the Euclidean distance as similarity measure forces the algorithm to search for the shortest distance for message-passing, meaning that it minimizes the

sum of squared distances for all clusters together creating balanced clusters as illustrated in Figure 9.

3.3.2 Finds structure in noise

An additional artifact of the AP algorithm is that it may find structure in noise, as the method cannot distinguish different densities, meaning that a density-based clustering method would be better suited in this case.

3.3.3 Fails to cluster non-spherical clusters

Arbitrarily shaped data tends to be incorrectly clustered when using the negative squared Euclidean distance as similarity measure. AP with the Euclidean distance as similarity measure may compute bad clustering results to data that does not consist of spherically shaped clusters without noise. Figure 14 illustrates this issue where the distance from C to A should be considered shorter than the distance from C to B. An alternative to circumvent this drawback is to use the CTD as similarity measure instead.

3.3.4 Computationally expensive to find the shared preference

Finding the right shared preference can be a difficult task. Some of the alternatives previously proposed in [5] are either using the smallest similarity or using the bisection method. Although both approaches can lead to satisfactory results, there are still some computational aspects that need to be considered. For choosing the appropriate shared preference, the bisection method is applied, which could lead to a missing of some shared preference values between the bisected intervals. For the test case consisting of seven arbitrarily shaped clusters shown in Figure 29, we cannot obtain the right number of clusters. The right number of clusters is seven but we can only obtain, as closest, six or eight clusters with the bisection method. However, this is computed with the negative squared Euclidean similarity measure which is unsuitable for this kind of data, but since the algorithm was able to obtain six and eight clusters, it should be able to obtain seven clusters. Another aspect that makes the task of finding the right shared preference computationally expensive in speed and time is large datasets. By searching for the right shared preference value of, e.g., the data with imbalance in data points, it is computationally expensive to compute the plot of shared preference values versus the number of clusters, e.g., see Figure 5, since we are dealing with 1100 observations. In case of the MNIST dataset with almost six times more data, this kind of figure is not computed at all. Instead we test different shared preference values based on the preference range and the results we obtain.

3.3.5 Large memory storage for the similarity matrix

Another limitation of the AP method arise from the size of the dataset. The AP method is a network-based method containing pairwise distances depending on the similarity measure. The similarity matrix of a dataset composed by N observations will be of $N \times N$ dimension, meaning that for high values of N it will be storing a lot of memory. This is an issue discussed already in the theory section. However, by applying the test cases, experience is gained of this problem as for, e.g., the MNIST dataset, where the computer could not store the full similarity matrix being 60000×60000, after hours of running.

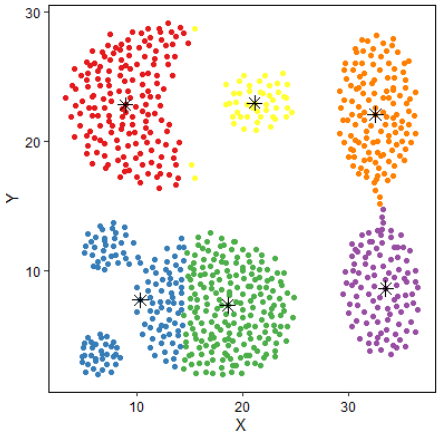


Figure 29: AP with the negative squared Euclidean measure applied on data of seven differently shaped clusters with 788 observations, and only acquiring six clusters. The clusters identified have different colours with centers indicated by the star-symbols. (X, Y) are 2D features.

4 Discussion and conclusions

This thesis is a statistical survey of the unsupervised learning method, AP, where the main objective is to investigate the statistical properties of the method and compare it with the widely used clustering method k-means. The AP method is a network-based method, whose main differences with k-means are that AP does not need to make any prior assumption about the number of clusters in the data, produces exemplars that are real data points and can use nonmetric similarity matrices as input.

In Section 2 the theory behind k-means, AP, PCA and the validation methods, the silhouette coefficient and CVNN, is presented. The methods are applied on the data presented in Section 3. The datasets consist of imbalanced data in number of data points and spatial extension, spherical data with noise, the flame-shaped data and the well known MNIST dataset of handwritten digits.

The results from the analysis show that the weaknesses and artifacts of AP, when using the negative squared Euclidean distance as similarity measure, are that it is unable to distinguish clusters with imbalanced spatial extent, finds structure in noise and fails to cluster non-spherical clusters. Another limitation of AP is that it is computationally expensive to find the right shared preference as input. Lastly, AP requires a similarity matrix as input, which can consume large memory storage when dealing with large datasets. However, we acknowledge that the most remarkable advantage from the test cases when using AP is that it is faster than k-means, when computing multiple runs for k-means to obtain the optimal result.

4.1 Outlooks

Many applications of AP have been done during the years. A similarity graph that has been applied to the method is, e.g., ISOMAP based metrics to handle data with manifold structure [1]. Another application is the negative generalized likelihood ratio in combination with Gaussian models, to cluster speakers from audio data [21].

For further studies, it would be valuable to implement AP using similarity measures such as the geodesic distance used in ISOMAP and CTD, which would be suitable for all datasets used in this thesis. By using CTD as similarity measure, it is expected that AP can cluster imbalanced, arbitrarily shaped and multidimensional data better, as these similarity measures learn about the clusters shape, also called shape-aware metrics. Additionally, to evaluate the statistical properties of the AP method using the CTD, it could be compared to other methods that use the CTD, such as spectral clustering [8]. It would also be interesting to use the mutual k-nearest neighbour graph and density-based methods, for comparison. Furthermore, to deal with overlapping data, i.e., data points belonging to more than one cluster with a membership weight between zero and one, it would be valuable to apply fuzzy or soft clustering [18]. This problem seems to appear in the MNIST dataset, and using fuzzy clustering might improve the results.

However, finding a suitable shared preference is still a tedious issue. A natural continuation of this thesis would be to investigate this more thoroughly. Moreover, we suggest using a faster root-finding method than the bisection method, to increase the efficiency when searching for the right shared preference.

References

[1] Baya, A.E. and Granitto, P.M. (2008) "ISOMAP based metrics for clustering", *Revista Iberoamericana de Inteligencia Artificial*, 12(37), p. 15–23

[2] Bishop, C. M. (2006). *Pattern recognition and machine learning*, New York, NY: Springer. p. 423-444

[3] Dueck, D. (2009) *Affinity propagation: Clustering data by passing messages*, PhD thesis, University of Toronto, Toronto

[4] Ester, M., Kriegel, H.-P., Sander, J. and Xu, X. (1996) "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", *in Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD, p. 226-231, AAAI Press

[5] Frey, B. J. and Dueck, D. (2007) "Clustering by passing messages between data points", *Science*, 315(5814), p. 972–976

[6] Gao, Q., Wang Y., Cheng X., Yu J., Chen X. and Jing T. (2019) "Identification of Vulnerable Lines in Smart Grid Systems Based on Affinity Propagation Clustering", *IEEE Internet of Things Journal*, 6(3), p. 5163-5171

[7] Halkidi, M., Vazirgiannis, M. and Hennig, C. (2015) "Method-independent indices for cluster validation", In C. Hennig, M. Meila, F. Murtagh, R. Rocci (eds.) *Handbook of Cluster Analysis*, CRC Press/Taylor and Francis, Boca Raton, p. 607-608

[8] Hastie, T., Tibshirani, R. and Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction.* 2. ed New York: Springer. p. 501-528, 534-541

[9] Heumann B. W., Liesch M. E., Bogen N. R., Meier R. A., Graziano M. (2020) "The contiguous United States in eleven zip codes: identifying and mapping socio-economic census data clusters and exemplars using affinity propagation", *Journal of Maps*, 16(1), p. 57-67

[10] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. (1998) "Gradient-based learning applied to document recognition", *Proceedings of the IEEE* 86(11), p. 2278-2324.

[11] Liu, Y., Li, Z., Xiong, H., Gao, X., Wu, J. and Wu, S. (2013) "Understanding and enhancement of internal clustering validation measures", *IEEE Transactions on Cybernetics 43*, p. 982-994.

[12] Lovász L. (1993) "Random Walks on Graphs: A Survey", *Combinatorics, Paul Erdos is Eighty*, 2, p. 1-46

[13] Luxburg, U. (2007) "A Tutorial on Spectral Clustering", *Statistics and Computing*, 17(4)

[14] Meltzer T., Yanover C., Weiss Y., (2005) "Globally optimal solutions for energy minimization in stereo vision using reweighted belief propagation", *Proc. 10th Conf. ICCV, IEEE Computer Society Press*, p. 428–435

[15] 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) p. 559-572

[16] Robinson, D. (2018) "Exploring handwritten digit classification: a tidy analysis of the MNIST dataset", viewed 15 October 2020, `http://varianceexplained.org/r/digit-eda/`

[17] Rousseeuw, P.J. (1987) "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis", *Journal of Computational and Applied Mathematics*, 20, p. 53-65

[18] Tan, P-N., Steinbach, M., Karpatne, A. and Kumar, V. (2019). *Introduction to data mining.* Second edition Harlow: Pearson. p. 487-514, 525-543, 571-597

[19] Thavikulwat, P. (2008) "Affinity Propagation: A clustering algorithm for computer-assisted business simulations and experiential exercises", *Developments in Business Simulation and Experiential Learning*, 35, p. 220-224.

[20] Wozniak, M., Tiuryn, J. and Dutkowski, J. (2010) "MODEVO: exploring modularity and evolution of protein interaction networks", *Bioinformatics*, 26(14), p. 1790–1791.

[21] Zhang, X., Gao, J., Lu, P. and Yan, Y. (2008) "A novel speaker clustering algorithm via supervised affinity propagation", *IEEE International Conference on Acoustics, Speech and Signal Processing*, p. 4369–4372

[22] Zhou, S. and Xu, Z. (2019) "Automatic grayscale image segmentation based on Affinity Propagation clustering", *Pattern Analysis and Applications*, 23(1), p. 331-348

A Proof of the within-point scatter

The definition of the within-point scatter for the squared Euclidean distance is

$$W(C) = \frac{1}{2}\sum_{l=1}^{K}\sum_{x_i \in c_l}\sum_{x_k \in c_l} ||x_i - x_k||^2$$

and the aim is to show that in the case of k-means, it is equivalent to

$$\sum_{l=1}^{K} N_l \sum_{x_i \in c_l} ||x_i - \mu_l||^2$$

To show this equivalence, $||x_i - x_k||^2$ is expanded to sum over the points x_k such that $x_k \in c_l$ according to:

$$\begin{aligned}\sum_{x_k \in c_l} ||x_i - x_k||^2 &= \sum_{x_k \in c_l} (x_i^T x_i - 2x_i^T x_k + x_k^T x_k) \\ &= N_l x_i^T x_i - 2x_i^T (N_l \mu_l) + \sum_{x_k \in c_l} x_k^T x_k\end{aligned} \tag{25}$$

The resulting expression in Equation 25 is summed over the points x_i such that $x_i \in c_l$ according to:

$$\begin{aligned}&\sum_{x_i \in c_l} (N_l x_i^T x_i - 2x_i^T (N_l \mu_l) + \sum_{x_k \in c_l} x_k^T x_k) \\ &= N_l \sum_{x_i \in c_l} x_i^T x_i - 2(N_l \mu_l)^T (N_l \mu_l) + N_l \sum_{x_k \in c_l} x_k^T x_k \\ &= 2N_l \left(\sum_{x_i \in c_l} x_i^T x_i - N_l \mu_l^T \mu_l \right)\end{aligned} \tag{26}$$

Lastly, the expression within the parenthesis is proven to be equivalent to $\sum_{x_i \in c_l} ||x_i - \mu_l||^2$ by using the following expansion:

$$\begin{aligned}\sum_{x_i \in c_l} ||x_i - \mu_l||^2 &= \sum_{x_i \in c_l} (x_i^T x_i - 2x_i^T \mu_l + \mu_l^T \mu_l) \\ &= \sum_{x_i \in c_l} (x_i^T x_i) - 2(N_l \mu_l)^T \mu_l + N_l \mu_l^T \mu_l \\ &= \sum_{x_i \in c_l} x_i^T x_i - N_l \mu_l^T \mu_l\end{aligned} \tag{27}$$

Hence, it is proven that $\sum_{x_i \in c_l} ||x_i - \mu_l||^2 = \sum_{x_i \in c_l} x_i^T x_i - N_l \mu_l^T \mu_l$. By using the obtained results it can be concluded that the within-point scatter for the

k-means algorithm is

$$\begin{aligned} W(C) &= \frac{1}{2}\sum_{l=1}^{K} 2N_l \sum_{x_i \in c_l} ||x_i - \mu_l||^2 \\ &= \sum_{l=1}^{K} N_l \sum_{x_i \in c_l} ||x_i - \mu_l||^2 \end{aligned} \tag{28}$$

B Additional figures

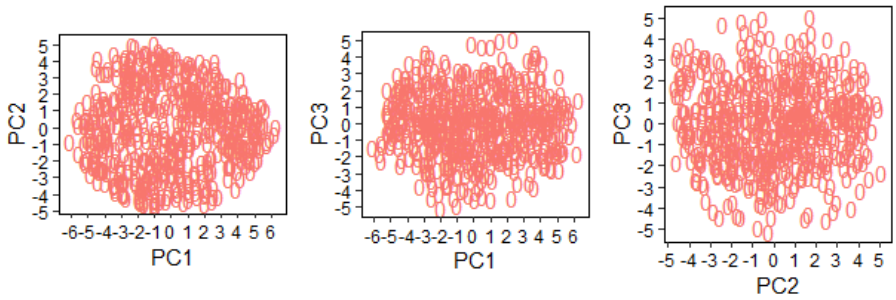


Figure 30: The first three PCs of digit zero from the MNIST dataset. Each figure presents a plotted combination of the PCs.

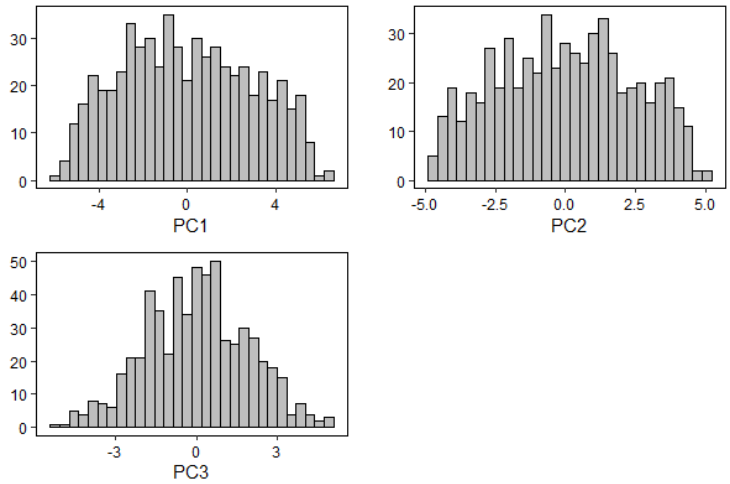


Figure 31: The distribution of the first three PCs of digit zero from the MNIST dataset. The x-axis presents a particular PC and the y-axis its count.

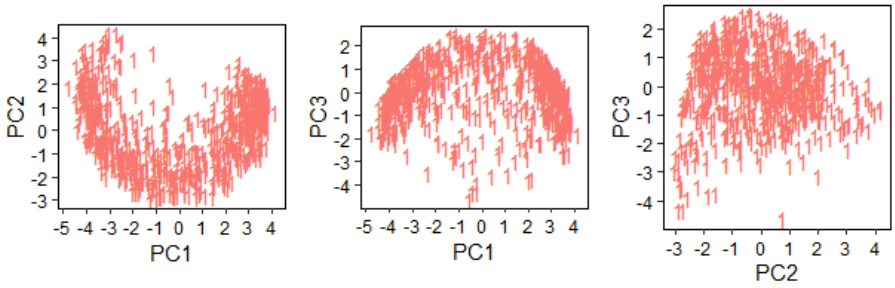


Figure 32: The first three PCs of digit one from the MNIST dataset. Each figure presents a plotted combination of the PCs.

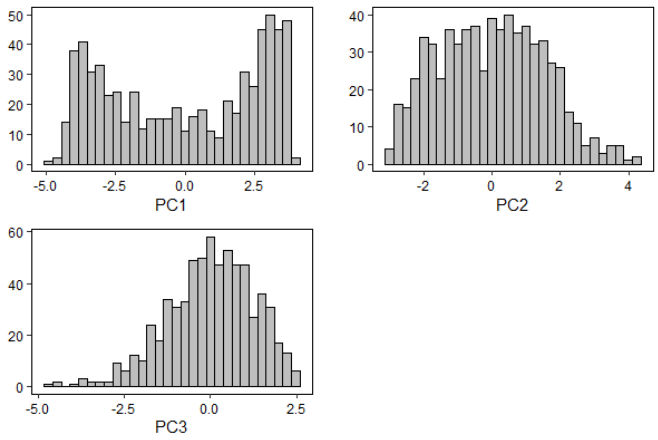


Figure 33: The distribution of the first three PCs of digit one from the MNIST dataset. The x-axis presents a particular PC and the y-axis its count.

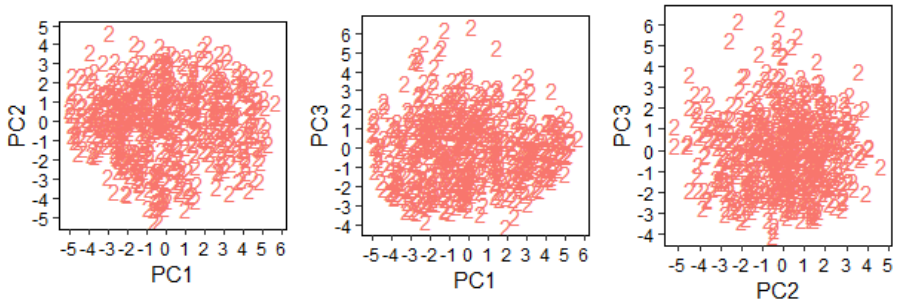


Figure 34: The first three PCs of digit two from the MNIST dataset. Each figure presents a plotted combination of the PCs.

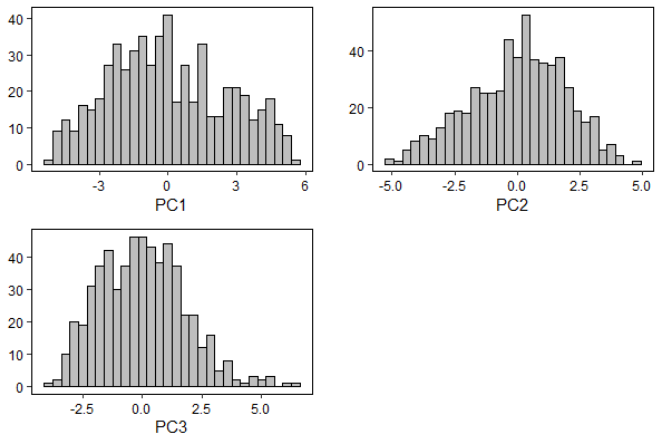


Figure 35: The distribution of the first three PCs of digit two from the MNIST dataset. The x-axis presents a particular PC and the y-axis its count.

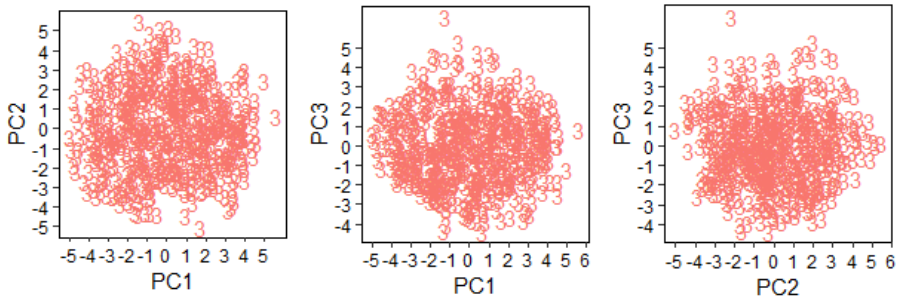


Figure 36: The first three PCs of digit three from the MNIST dataset. Each figure presents a plotted combination of the PCs.

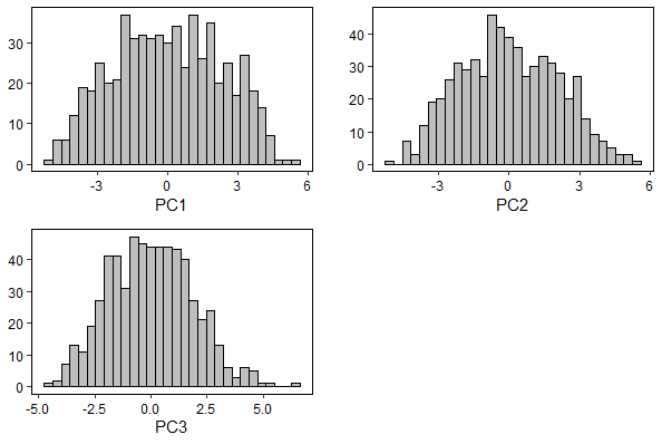


Figure 37: The distribution of the first three PCs of digit three from the MNIST dataset. The x-axis presents a particular PC and the y-axis its count.

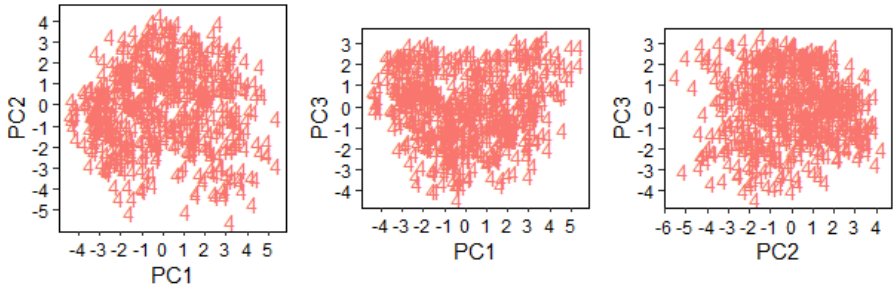


Figure 38: The first three PCs of digit four from the MNIST dataset. Each figure presents a plotted combination of the PCs.

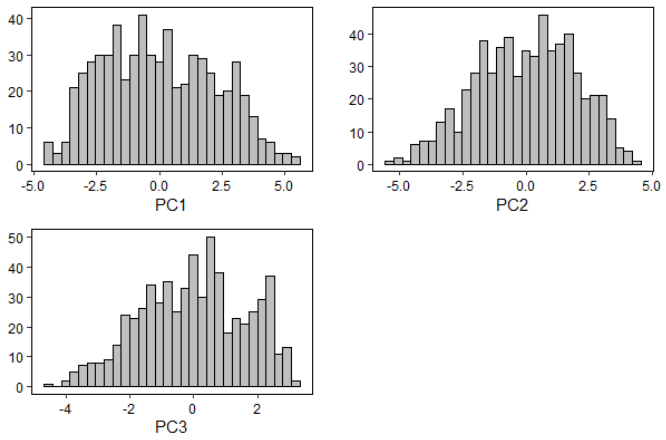


Figure 39: The distribution of the first three PCs of digit four from the MNIST dataset. The x-axis presents a particular PC and the y-axis its count.

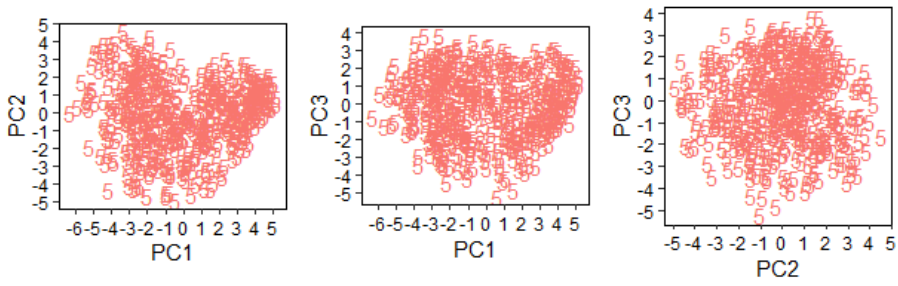


Figure 40: The first three PCs of digit five from the MNIST dataset. Each figure presents a plotted combination of the PCs.

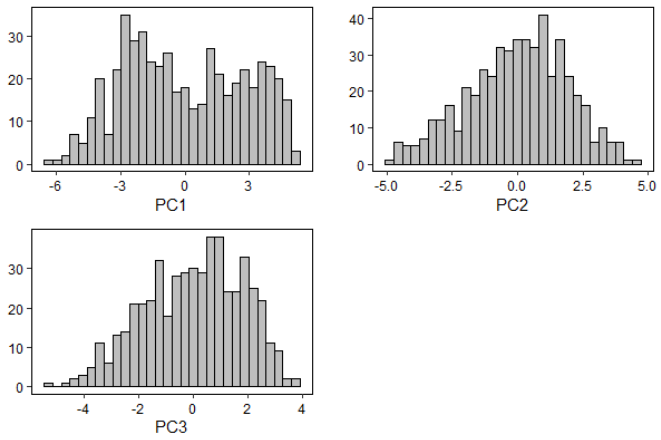


Figure 41: The distribution of the first three PCs of digit five from the MNIST dataset. The x-axis presents a particular PC and the y-axis its count.

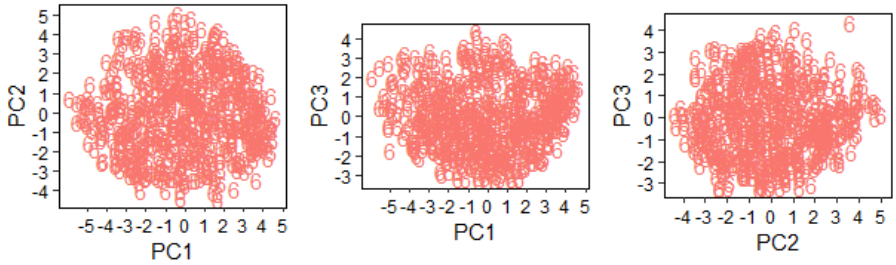


Figure 42: The first three PCs of digit six from the MNIST dataset. Each figure presents a plotted combination of the PCs.

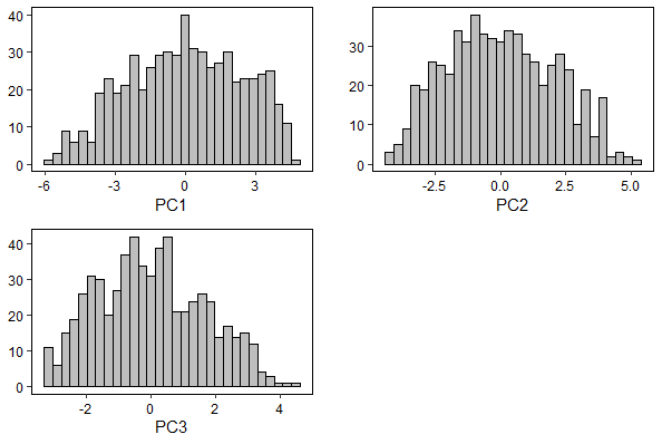


Figure 43: The distribution of the first three PCs of digit six from the MNIST dataset. The x-axis presents a particular PC and the y-axis its count.

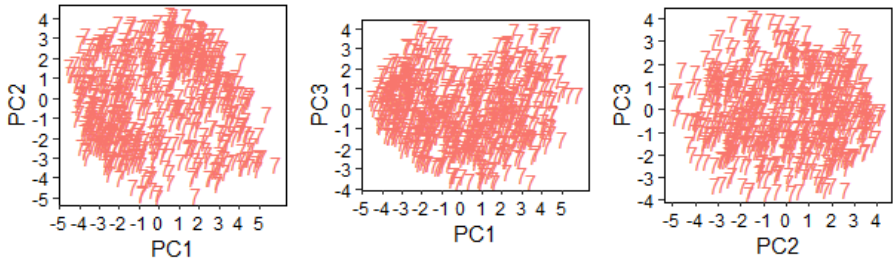


Figure 44: The first three PCs of digit seven from the MNIST dataset. Each figure presents a plotted combination of the PCs.

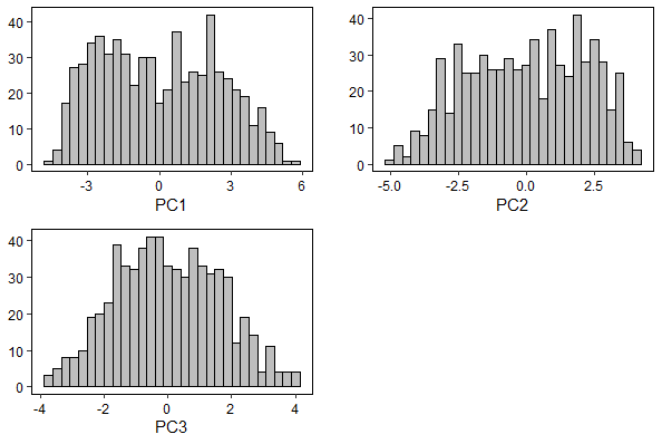


Figure 45: The distribution of the first three PCs of digit seven from the MNIST dataset. The x-axis presents a particular PC and the y-axis its count.

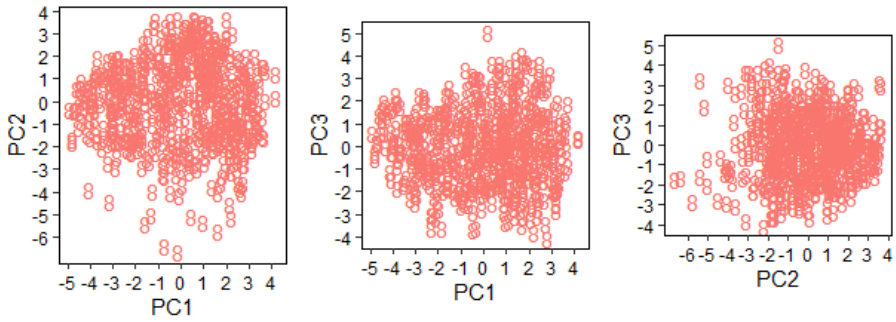


Figure 46: The first three PCs of digit eight from the MNIST dataset. Each figure presents a plotted combination of the PCs.

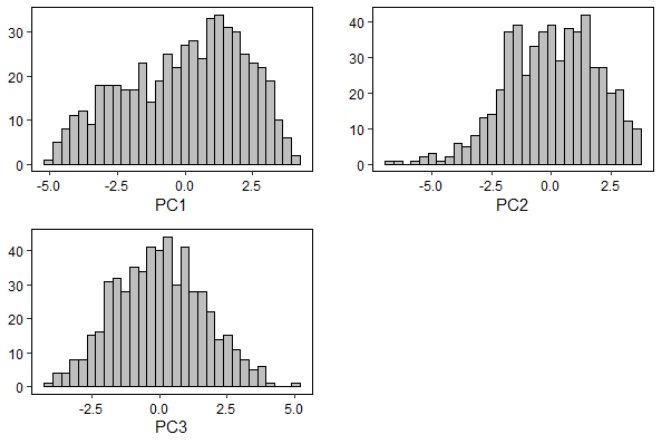


Figure 47: The distribution of the first three PCs of digit eight from the MNIST dataset. The x-axis presents a particular PC and the y-axis its count.

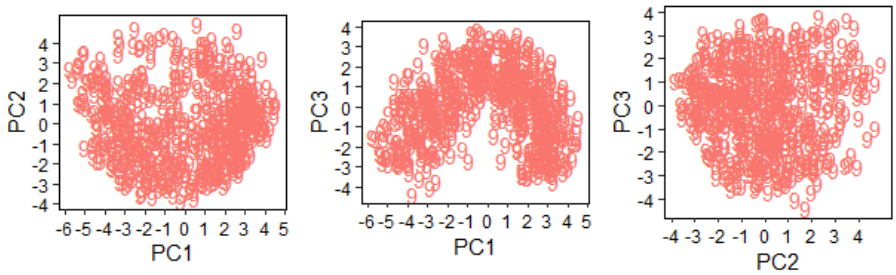


Figure 48: The first three PCs of digit nine from the MNIST dataset. Each figure presents a plotted combination of the PCs.

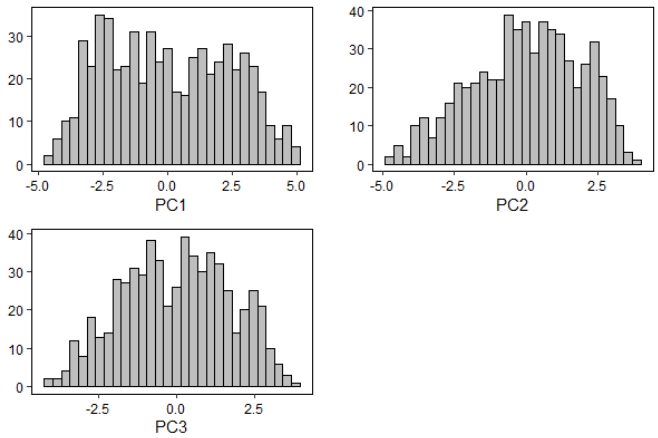


Figure 49: The distribution of the first three PCs of digit nine from the MNIST dataset. The x-axis presents a particular PC and the y-axis its count.