SILESIAN UNIVERSITY OF TECHNOLOGY

FACULTY OF AUTOMATIC CONTROL, ELECTRONICS AND COMPUTER SCIENCE

PhD Thesis

Data clustering with mixtures of multidimensional distributions

Abstract

mgr Mateusz Kania

supervisor: prof. zw dr hab inż Andrzej Polański

The following thesis was financially supported by European Union funds project AIDA (Applied Integrative Data Analysis) POWR.03.02.00-00-I029.

This page intentionally left blank.

Contents

1 Introduction

Unsupervised clustering is a widely used technique in data analysis and machine learning. The goal is to group similar objects or observations into clusters without prior knowledge of the true class labels. This can be a challenging task, especially when dealing with high-dimensional and complex datasets. Many different algorithms and approaches can be used to perform unsupervised clustering. One way to categorize unsupervised clustering algorithms is based on their underlying models or assumptions.

One such group are model-based algorithms, which assume that the data is generated from a mixture of probability distributions. Those algorithms aim to estimate the parameters of these distributions to identify the clusters. An example of such an algorithm is the Expectation-Maximization (EM) algorithm.

1.1 GitHub Code availability

The code is available at the github: https://github.com/callimae/multivarEM

To install the package, R should be installed. It is free, open-source programming environment, available on the website: **https://www.r-project.org/**

Commands to execute after opening R:

```
install.packages("devtools")
install_github("callimae/multivarEM")
```

2 Model-based algorithms

The EM is a powerful computational technique used to estimate the parameters of a statistical model in the presence of missing or incomplete data. The algorithm consists of two alternating steps: the E-step and the M-step.

In the E-step, the algorithm calculates the expected value of the missing data given the observed data and the current estimate of the parameters. This step involves calculating the posterior distribution of the missing data using the current estimate of the parameters.

In the M-step, the algorithm updates the parameter estimates based on the expected values of the missing data calculated in the E-step. This step involves finding the parameters' maximum likelihood estimate given the missing data's expected values.

The EM algorithm is used in various fields, including machine learning, computer vision, natural language processing, and bioinformatics. It is effective when data is incomplete or missing and traditional maximum likelihood methods are not applicable.

One of the strengths of the EM algorithm is that it is guaranteed to converge to a local maximum of the likelihood function and often the global maximum. However, the algorithm can be sensitive to the choice of initial parameter values and can be computationally intensive, particularly for large datasets. It is especially true in multidimensional data. To minimize computationally heavy cases, two version of EM has been implemented.

2.1 Multivariate Gaussian Mixture EM

The Gaussian Mixture Model EM uses only diagonal variances of the covariance matrix instead of a full one. Derivation of the algorithm lead to condigtional distribution of hidden variables, q-function and the equation to update the parameters.

2.1.1 Conditional distribution of hidden variables

We accept some values of parameters as initial values for iterations. We call this parameter guess. We indicate those guessed parameters by the letter g in superscript. Using Bayes Theorem, we can calculate the conditional distribution of the hidden variable using parameter guess.

$$p(z_i = k|\mu_k^{\mathrm{g}}, (\Sigma^U)_k^{\mathrm{g}}, \alpha_k^{\mathrm{g}}) = p(k \mid i) = \frac{\alpha_k^{\mathrm{g}} f\left(x_i, \mu_k^{\mathrm{g}}, (\Sigma^U)_k^{\mathrm{g}}\right)}{\sum_{\chi=1}^{K} \alpha_\chi^{\mathrm{g}} f\left(x_i, \mu_\chi^{\mathrm{g}}, (\Sigma^U)_\chi^{\mathrm{g}}\right)} \tag{1}$$

The likelihood is multiplied by its prior probability (mixing proportion). Then the standardization is done by summing the likelihood multiplied by the prior of all other mixture components. The result is the posterior probability of z_i generated by k mixture component.

2.1.2 Conditional expectation of the log likelihood function (Q-function)

Using 1 we can derive conditional expectation of the log-likelihood function. Following nomenclatrure often used in the literature we also call this conditional expectation a Q-function.

$$E\left(L^C \mid \text{ data, parameter guess }\right) = Q =$$

$$= \sum_{i=1}^{N}\sum_{k=1}^{K} (\ln \alpha_k)\, p(k \mid i) + \sum_{i=1}^{N}\sum_{k=1}^{K} \left[-\frac{M}{2}\ln(2\pi) - \frac{1}{2}\ln\left|\Sigma_k^U\right| - \frac{1}{2}\left(x_i - \mu_k\right)^T (\Sigma^U)_k^{-1} \left(x_i - \mu_k\right)\right] p(k \mid i) \tag{2}$$

In the equation above terms were already explained before. M is a number of dimensions.

Maximizing Q-function with respect to parameters α_k,μ_k,Σ_k we obtain

$$\hat{\alpha}_k = \frac{\sum_{i=1}^{N} p(k \mid i)}{N} \tag{3}$$

$$\hat{\mu}_k = \frac{\sum_{i=1}^{N} x_i p(k \mid i)}{\sum_{i=1}^{N} p(k \mid i)}, \quad k = 1, 2, \ldots, K \tag{4}$$

$$\hat{\Sigma}_k^U = \frac{\sum_{i=1}^{N} [\mathrm{diag}\left(x_i - \hat{\mu}_k\right)]^2 p(k \mid i)}{\sum_{i=1}^{N} p(k \mid i)} \tag{5}$$

where:

$\mathrm{diag}(y) = \begin{bmatrix} y_1 & 0 & \cdots & 0 \\ 0 & y_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & y_M \end{bmatrix}$ - is a diagonal matrix composed with elements of a vector y

$y = [y_1, y_2, ..., y_M]^T$

2.2 Multinomial Mixture EM

2.2.1 Conditional distribution of hidden variables

As before, guessed parameters are denoted with "g" in superscript. They are needed initialize the iterations and calculate the conditional distribution of the hidden variable through the Bayes Theorem.

$$p(k|d) = \frac{\alpha_k^{\mathbf{g}} \times p_{k_1}^{\mathbf{g},nd_1} \times p_{k_2}^{\mathbf{g},nd_2} \times ... \times p_{k_M}^{\mathbf{g},nd_M}}{\sum_{\kappa=1}^{K} \alpha_\kappa^{\mathbf{g}} p_{\kappa_1}^{\mathbf{g},nd_1} \times p_{\kappa_2}^{\mathbf{g},nd_2} \times ... \times p_{\kappa_M}^{\mathbf{g},nd_M}} = \frac{\alpha_k^{\mathbf{g}} \prod_{m=1}^{M} p_{k_m}^{\mathbf{g},nd_m}}{\sum_{\kappa=1}^{K} \alpha_\kappa^{\mathbf{g}} \prod_{m=1}^{M} p_{\kappa_M}^{\mathbf{g},nd_M}} \tag{6}$$

To obtain the posterior probability of the variable z_i generated by the k-th mixture component, the corresponding prior probability (mixing proportion) is first multiplied by the likelihood. The resulting value is then standardized by adding it to the product of the likelihood and prior probabilities of all the other mixture components.

2.2.2 Conditional expectation of the log likelihood function (Q-function)

Q-function for multinomial mixture models is as follows:

$$\begin{aligned} E\left(L^C \mid \text{ data, parameter guess }\right) = Q = \\ = C + \sum_{d=1}^{D} \left[\sum_{k=1}^{K} \ln\left(\alpha_k\right) p(k \mid d) + \sum_{m=1}^{M} \sum_{k=1}^{K} n_{dm} \ln\left(p_{km}\right) p(k \mid d) \right] \end{aligned} \tag{7}$$

Maximizing Q-function with respect to parameters α_k, p_k, we obtain:

$$\hat{\alpha} = \frac{\sum_{d=1}^{D} p(k|d)}{D} \tag{8}$$

where:
$\hat{\alpha}$ - is vector of new mixing proportions
D - is the count of observations

$$\hat{p}_{km} = \frac{\sum_{d=1}^{D} n_{dm} p(k|d)}{\sum_{n=1}^{M} \sum_{d=1}^{D} n_{dm} p(k|d)} \tag{9}$$

where:
n_{dm} - is an observation vector
$\hat{p}_{km}$ - is a vector of new probabilities proportions

3 Distance-based algorithms

Hierarchical clustering is a clustering algorithm that groups data points into clusters based on similarity. The algorithm creates a tree-like structure of clusters called a dendrogram, where each node represents a cluster, and the leaves represent individual data points. There are two types of hierarchical clustering: agglomerative and divisive. In agglomerative hierarchical clustering, the algorithm starts with each data point as its cluster and merges the closest pairs of clusters until all points belong to a single cluster. In divisive

hierarchical clustering, the algorithm starts with all data points in a single cluster and recursively splits it into smaller clusters until each point is in its cluster.

K-means is a centroid-based clustering algorithm that partitions data into k clusters, where k is the number of predefined clusters. The algorithm starts by randomly assigning k centroids to the data points. Then, iteratively reassigns each data point to the nearest centroid and updates the centroids based on the newly formed clusters until the algorithm converges.

K-medoids is a variant of k-means that uses medoids, representative data points within each cluster, instead of centroids. The algorithm starts by randomly selecting k medoids from the data points. Then iteratively reassigns each data point to the nearest medoid and updates the medoids based on the newly formed clusters until convergence.

Fuzzy c-means is a soft clustering algorithm that assigns each data point a membership value for each cluster, indicating the degree to which the point belongs to each cluster. The algorithm randomly assigns membership values to each point and iteratively updates the membership values and cluster centroids based on the current assignments until convergence.

The significant difference between hierarchical clustering and the other three algorithms is that hierarchical clustering produces a hierarchy of clusters, while the others produce a fixed number of clusters. The main difference between k-means and k-medoids is how they represent each cluster, using centroids or medoids, respectively. Fuzzy c-means differs from the other algorithms in assigning membership values to each point rather than hard cluster assignments.

Those algorithms between these algorithms are that they use distance metrics to measure the similarity between data points and assign them to clusters.

4 Study pipeline

4.1 Data gathering

All datasets can be divided into two parts. The first one comprises artificially created data that can present a more controlled challenge to algorithm performance. The second subset includes real datasets that were selected with the criterion of multidimensionality in mind.

Simulated multivariate normal data Multivariate normal mixtures were generated using random vectors (μ), matrices (Σ), mixing proportions (α), and fixed dimensions (d). μ was derived from a uniform distribution between 0 and 10, while Σ assumed zero correlation between variables. Variances came from a uniform distribution between 0.1 and 1, and α was drawn from a uniform distribution between 0.1 and 1, with the vector of mixing ratios standardized to sum to 1. Over 15,000 files were created with 2 to 10 components and 5 to 1600 dimensions, using two PRNG packages: MASS and MultiRNG. Over 15,000 files were created.

Simulated multinomial data Multinomial mixtures were generated using random probability vectors (p), observation numbers (n), mixing proportions (α), and fixed dimensions (d). p was derived from a uniform distribution between 0 and 1 and standardized to sum to 1. n was generated from the interval [3, n], and α was calculated using a uniform

Data set	Source
Somatic Mutations Counts	TCGA
Gene Expressions	TCGA/cBioportal
Codons Frequency	UCI
Sport Activities	UCI
The Free Music Archive	GitHub
Arthythmia	UCI
NASA Kepplers	Kaggle

Table 1: Real data-set with their sources

distribution between 0.1 and 1, with values standardized to sum to 1. Over 8,000 files were created containing 2 to 10 clusters and 5 to 1600 dimensions.

4.1.1 Real data

The most important part of our comparative analysis of unsupervised clustering algorithms is computational experiments concerning some publicly available data sets. Extensive real-world datasets collections can be used, e.g., Kaggle, UCI, NCBI or NCI. When choosing data sets for clustering, the following criteria were used:

- high dimensionality
- numerical features (real or integer).
- variety of data types
- ground-truth availability

According to mentioned criterions, the data should have many dimensions, to begin with. The mixture should not be univariate or bivariate, and features cannot be categorical.

Various data types assume that data comes from different expertise fields or is measured differently. For example, somatic mutation count and gene expression came from the same TCGA project but were measured differently.

Real data consist of various datasets from a few different fields, although majority have some genetic and medical background. As such following data was chosen:

Somatic Mutation Counts This study focuses on somatic mutations of DNA in cancer patients, a critical topic in cancer research. Somatic mutations can be caused by exposure to mutagens, replication errors, or other factors. They can be classified into two major types: driver mutations, which are related to cancer development, and passenger mutations, which have little or no impact. Distinguishing between these mutations is complex and involves various factors, including their positions in DNA and their effects on transcripts.

This study aims to cluster patients diagnosed with different types of cancer based on counts of somatic mutations in genes. To achieve this, the counts of mutations in genes were used as observational vector data. The hypothesis is that this information can be used to distinguish between various types of cancer.

The data used in the study were obtained from The Cancer Genomic Atlas (TCGA) and annotated using the somatic variant caller Mutect2. The processed data included

information on the sample number, the name of the gene in which the mutation occurred, and the frequency of mutations. Full data consisted over 10 000 observation and about 35 000 of features. The study also aimed to determine which clustering algorithms performed best in the clustering task. Overall, the study contributes to understanding the complex nature of somatic mutations in cancer.

Gene Expressions Gene expression is the process by which the cell utilises information stored in the genome to create functional proteins, leading to the manifestation of observable traits or phenotypes from genetic data. This process requires the participation of enzymes and proteins to convert the information encoded in DNA into RNA particles, collectively known as the transcriptome. Messenger RNA (mRNA) is an RNA molecule that plays a critical role in translating genetic information into proteins. During translation, the mRNA sequence is decoded into amino acids, which are the building blocks of proteins. Any changes in the expression level of mRNA can affect protein synthesis and lead to the development of various diseases, including cancer. cBioPortal is an online platform that provides access to molecular and clinical data from various cancer genomic studies. The data available on the platform include DNA methylation, mRNA expression, microRNA expression, and protein level data.

Presented analysis focused on mRNA expression data to investigate whether this type of data can distinguish between different kinds of cancer based on their expression patterns. The gene expression data on cBioPortal is multidimensional, containing information on numerous genes across multiple cancer types. A matrix transposition technique was used to reformat the data, enabling us to present the sample numbers as observations and genes as variables. This allowed us to explore the relationships between different cancer types based on their gene expression patterns, providing valuable insights into cancer biology. Full data consisted over 9 000 observation and about 35 000 of features, similarly as in the case of Somatic Mutation Counts.

Codons frequency The genetic code translating DNA sequences into amino acids was long thought universal across all species. However, recent discoveries have shown that there are exceptions to this rule, particularly in the non-standard codons used by mitochondria genomes. Understanding the frequency of codon usage in different organisms can provide insights into their genetic composition and taxonomic identity. A dataset of codon frequency usage in diverse organisms, sourced from the Codon Usage Tabulated from GenBank (CUTG) database, was analysed using unsupervised learning algorithms to investigate whether the species' structure could be restored by dividing them into their respective kingdoms based on codon frequency. The dataset was provided by Bohdan Khomtchouk of the University of Chicago and included frequencies of different codons used by species across several kingdoms. The results suggested that codon usage frequencies can be a helpful tool for distinguishing between species and identifying their taxonomic affiliations. However, it was also found that the effectiveness of this approach varies depending on the specific algorithm used.

The study highlights the importance of considering different methods when analysing codon usage data and underscores the value of machine learning techniques for understanding biological phenomena.

Sports activities Popularność smartwatchów znacznie wzrosła w ostatnich latach, a wiele z nich oferuje szereg funkcji wykraczających poza standardowe funkcje, takie jak

połączenia telefoniczne i wiadomości. Niektóre modele mogą śledzić tętno, spalanie kalorii i różne statystyki związane z aktywnością sportową. Bardziej zaawansowane smartwatche mogą automatycznie wykrywać działania użytkownika i rejestrować odpowiednie metryki. W badaniu tym przeanalizowano dane udostępnione przez profesora z Yale, Billura Barshana, na platformie UCI. Dane zawierały informacje o 19 aktywnościach sportowych wykonywanych przez ośmiu uczestników, czterech mężczyzn i cztery kobiety, w wieku od 20 do 30 lat, którzy nie otrzymali żadnych instrukcji dotyczących wykonywania ćwiczeń. Dane zostały zebrane za pomocą czujników umieszczonych na różnych częściach ciała, z częstotliwością próbkowania 25 Hz, co dało 45 atrybutów i 1140000 obserwacji.

The Free Music Archive Free Music Archive to kompleksowy projekt analizy muzyki, który oferuje ogromną bibliotekę 106 577 utworów pokrytych przez 16 341 artystów na 14 854 albumach. Dane, które zostały udostępnione publicznie pod koniec 2016 roku, obejmują: Wstępnie obliczone cechy audio. Metadane na poziomie użytkownika i utworu. Pełnowymiarowe, wysokiej jakości audio dla wybranych gatunków. Cechy audio zostały obliczone przy użyciu librosa, bogatej biblioteki Pythona do analizy audio i muzyki, która umożliwia niskopoziomową ekstrakcję cech, takich jak chromagramy, spektrogramy Mel, Mel Frequency Cepstral Coefficient (MFCC) oraz inne cechy spektralne i rytmiczne. Wszystkie utwory w archiwum są zorganizowane w hierarchiczną taksonomię 161 gatunków, w tym rock, jazz, pop, czy muzyka klasyczna.

W badaniu z wykorzystaniem danych pochodzących z Archiwum wolnej muzyki, dane zostały zredukowane do dwunastu najczęściej występujących gatunków. Ostateczny zbiór danych składał się z 49 598 obserwacji i 518 cech.

Arrhythmia Arrhythmias are a type of heart condition where the heartbeat is irregular, too fast, or too slow. Atrial fibrillation, a specific type of arrhythmia, is known to increase the risk of stroke by up to five times. Atrial fibrillation is responsible for almost 20% to 30% of strokes, and these strokes are often more severe and fatal than those caused by other factors. Shockingly, strokes caused by atrial fibrillation lead to death much more frequently than strokes due to other causes. According to a study conducted in 2016, nearly 7.6 million people in the European Union had atrial fibrillation. This number is predicted to increase by 89% to 14.4 million by 2060, with the current prevalence rising by 22% from 7.8% to 9.5%. The treatment of arrhythmia consumes a considerable amount of European funds spent on healthcare, with the yearly treatment cost ranging from 0.28% to 2.6% of the funds. The data analysed in the thesis comes from a study that aimed to differentiate between the presence and absence of cardiac arrhythmia. The observations were grouped into sixteen categories, with the first category representing regular ECG classes and the remaining categories representing various types of arrhythmia. Although a computer program currently classifies the data, there are differences between the grouping done by the program and the classification done by a cardiologist.

NASA Kepplers NASA's Kepler Space Telescope was launched on March 6, 2009, to identify other habitable planets beyond our solar system. Focusing on an area with approximately 150,000 stars like the sun, Kepler has made significant discoveries, including hundreds of star systems hosting multiple planets and identifying planets that orbit in so-called habitable areas, where the surface temperature may be fit for life-giving liquid water. Numerical data measured by the telescope was analyzed with unsupervised algorithms to check similarities between the planets.

4.2 Data preprocessing

After the data was downloaded and cleaned, it was parsed into a fixed format. A matrix of size $n \times d$ was created for each dataset, with observations n placed in rows and variables d in columns to ensure consistency across all analyzed datasets.

Except for the NASA Kepler dataset, the remaining datasets typically contained ten or more classes. The permutation method was employed to generate 10 datasets for 2, 3, 4, 5, and 6 component mixtures to create multiple datasets for analysis. In this way, each set of six component mixtures always contained at least one or more different classes. Consequently, 50 datasets with classes in various configurations were created from a single real dataset.

This methodology was applied to all datasets.

4.3 Data filtration and scaling

No filtration

In this case, the datasets containing all the original variables were left unchanged. They were used in the scaling step and in the clustering stage of the analysis.

Variance decomposition

The respective variance for each variable was calculated, resulting in a vector of n variances with varying values of n, dependent on the dataset used. The resulting vector was assumed to represent a one-dimensional mixture comprising essential variables and additional noise. A mixture decomposition technique was employed using the mclust package to differentiate between these elements. The mixtures were aligned to contain two to twenty-five components, and the final number of groups was determined using the Bayesian Information Criterion.

Scaling

Scaling was performed separately on the complete and reduced datasets to preserve the variance. The variables were scaled by subtracting the mean value from the vector of observations and dividing the resulting values by the standard deviation. After scaling, two additional files were created for each dataset.

4.4 Data clustering

All algorithms mentioned in the table were employed to cluster the data. Each algorithm generated a separate file, which was saved to disk for future processing.

4.5 Clusters evaluation

In this study, clusters were evaluated based on their correct assignment. For this purpose, the Hungarian algorithm was employed to assign results to the labels. Then, a few metrics were implemented to compare algorithms' performance from various perspectives. Finally, the results were visualized using different plot types.

No	Abbreviation	Full name
1	CMN	C-means
2	MANCMN	C-means with Manhattan distance
3	GMM	Gaussian Mixture EM
4	GMMK	Gaussian Mixture Models with k-means
5	HC	Hierarchical Clustering
6	MANHC	Hierarchical Clustering with Manhattan distance
7	KMN	K-means
8	KMN++	K-means++
9	KMD	K-medoids
10	MMM	Multinomial Mixture EM
11	MMMK	Multinomial Mixture EM with k-means

Table 2: Algorithms used in the clustering

4.5.1 Cluster assigment - Hungarian algorithm

The Hungarian algorithm is an efficient method to solve the operations research assignment problem. A combinatorial optimisation algorithm aims to find the optimal matching between two sets of equal size. Here, the algorithm was used to relabel clustered data.

4.5.2 Clusters validation metrics

The following metrics were used to measure quality of clustering:

- Adjusted Rand Index
- Jaccard and Weigted Jaccard Index
- Balanced Accuracy Index
- Simple Matching Coefficient
- Weighted Simple Matching Coefficient
- Beta-binomial conjugate distribution

4.5.3 Visualization

Dimensionality reduction Dimensionality reduction methods allow the representation of observations in space with fewer dimensions than in the original data. It makes visualization easier. In this study following techniques were used:

- PCA
- tSVD
- tSNE
- Random projection

No	Algorithm	Complete	Reduced	Scaled	Scaled And Reduced
1	GMM	1.00	0.79	0.96	0.81
2	KMN	0.81	0.57	0.84	0.58
3	KMN++	0.73	0.51	0.75	0.52
4	GMMK	0.74	0.48	0.71	0.41
5	MANHC	0.67	0.42	0.67	0.42
6	HC	0.63	0.40	0.63	0.40
7	CMN	0.54	0.39	0.56	0.40
8	MMMK	0.59	0.37	0.04	0.00
9	MMM	0.53	0.32	0.04	0.00
10	MANCMN	0.16	0.10	0.15	0.10
11	KMD	0.13	0.05	0.13	0.06

Table 3: Entries represent numbers of successes (correctly assigned clusters) normalized to all assigments in Simulated Multivariate Normal Mixtures

Metrics report Various plots were used to report the metrics.

- The ARI Index - was shown by using boxplots.
- Beta binomial distribution - was drawn using the beta probability density function.
- A median of Balanced Accuracy - was drawn on a polar coordinate plane.
- WSMC, SMC and WJACC - the three metrics were drawn similarly to violin plots. In addition, plots also contain means and single score points.
- Correlation plot – correlation coefficient was calculated between every two metrics to measure the linear relationship's strength and direction of clustering quality.

5 Summary and conclusions

5.1 Aggregated results

There are plethora numbers of metrics. Depending on the case, some might be better than others, especially in unbalanced data. However, the simplest one, a rough number of successes, was used in the aggregation. The scores have been normalized using the min-max method, where the lowest score is set to 0, the highest score is set to 1, and all other scores are scaled proportionally. The scores in the table represent the proportion of correctly assigned clusters, with a higher score indicating better performance. Lastly, the data were sorted based on the highest sum across different data types.

5.2 Simulated data

The results of the study showed that GMM algorithm performed the best overall, with a score of 1 for complete data. KMN algorithm also performed well, with scores ranging from 0.57 to 0.84 depending on the type of data used. The KMN++, GMMK, MANHC, and HC algorithms had moderate performance, while CMN, MMMK, MMM, MANCMN, and KMD algorithms had poor to inferior performance. Overall, GMM and KMN algorithms are good choices for clustering in this dataset, especially when using complete and scaled data.

No	Algorithm	Complete	Reduced	Scaled	Scaled and Reduced
1	MANHC	0.92	0.52	0.94	0.51
2	HC	0.92	0.5	0.92	0.5
3	KMN	0.89	0.52	0.91	0.52
4	KMN++	0.86	0.5	0.89	0.5
5	GMM	0.23	0.58	0.79	0.59
6	GMMK	0.84	0.35	0.58	0.33
7	CMN	0.59	0.32	0.61	0.34
8	KMD	0.6	0.32	0.53	0.33
9	MMM	1	0.54	0	0.18
10	MMMK	0.97	0.54	0.01	0.08
11	MANCMN	0.05	0.13	0.02	0.12

Table 4: Entries represent numbers of successes (correctly assigned clusters) normalized to all assigments in Simulated Multinomial Mixtures

No	Algorithm	Complete	Reduced	Scaled	Scaled and Reduced
1	GMM	0.78	0.49	0.51	0.53
2	GMMK	0.76	0.36	0.64	0.5
3	HC	0.42	0.38	0.77	0.51
4	MMMK	0.91	0.82	0.12	0.2
5	KMN++	0.4	0.44	0.66	0.56
6	KMN	0.41	0.43	0.6	0.55
7	MANHC	0.45	0.41	0.64	0.46
8	MMM	1	0.67	0	0.27
9	KMD	0.38	0.39	0.49	0.44
10	CMN	0.12	0.1	0.72	0.32
11	MANCMN	0.15	0.08	0.39	0.31

Table 5: Entries represent aggregated successes (correctly assigned clusters), normalized to all assignments in Real Data

Based on the results, the MMM algorithm had the highest score across algorithms, with a score of 1 on the complete dataset, while the MMMK algorithm also performed well with a score of 0.97 on the complete dataset. HC, KMN, and KMN++ also performed well, with scores ranging from 0.86 to 0.92 on the complete dataset, and consistent performance across all datasets. GMMK had better performance than GMM on the complete dataset but performed worse on the reduced, scaled, and scaled and reduced datasets. CMN and KMD had lower scores ranging from 0.32 to 0.61 on the complete dataset, while GMMK scored 0.58.

Overall, the study suggests that MMM and MMMK are effective algorithms for clustering, particularly on complete datasets.

5.3 Real data

In the study, GMM and GMMK had the highest sum scores, with GMM achieving the highest scores in the Scaled dataset and GMMK achieving the highest scores in the Reduced dataset. However, the highest score achieved was MMM when the data was

complete. HC, KMN, and KMN++ performed similarly across all four datasets, while MANHC performed slightly better than HC. KMD and CMN had the lowest scores among all algorithms, while MANCMN performed marginally better than KMD and CMN.

In summary, while GMM and GMMK had the highest sum scores, it's important to note that MMM achieved the highest score in the complete dataset.

5.4 Conclusions

This PhD project aimed to compare the performance of distance and model-based algorithms on various datasets. In particular, in the study, two model-based algorithms were implemented (Gaussian Mixture EM and Multinomial Mixture EM). Those two algolrithms and four distance-based (agglomerative hierarchical clustering, k-means, k-medoids, and fuzzy c-means) were applied to both simulated and actual datasets. The study used several metrics to evaluate the clustering results, including Adjusted Rand Index, Simple Matching Coefficient, Weighted Jaccard Index, Balanced Accuracy, and metrics based on Beta-Binomial conjugate distribution. These metrics helped to quantify the quality of the clustering results and compare the performance of different algorithms on different datasets. The study results were presented graphically, along with a brief description of the findings.

Intuitively, we know, that performance of the different algorithms varies depending on the type and complexity of the dataset. While distance-based algorithms are very powerful, the model-based algorithms presented in this thesis are highly competitive and can be considered as potent tools in unsupervised clustering methods.