Customer Segmentation and Customer Profiling for a Mobile Telecommunications Company Based on Usage Behavior

A Vodafone Case Study

S.M.H. Jansen

July 17, 2007
Acknowledgments

This Master thesis was written to complete the study Operations Research at the University of Maastricht (UM). The research took place at the Department of Mathematics of UM and at the Department of Information Management of Vodafone Maastricht. During this research, I had the privilege to work together with several people. I would like to express my gratitude to all those people for giving me the support to complete this thesis. I want to thank the Department of Information Management for giving me permission to commence this thesis in the first instance, to do the necessary research work and to use departmental data.

I am deeply indebted to my supervisor Dr. Ronald Westra, whose help, stimulating suggestions and encouragement helped me in all the time of research for and writing of this thesis. Furthermore, I would like to give my special thanks to my second supervisor Dr. Ralf Peeters, whose patience and enthusiasm enabled me to complete this work. I have also to thank my thesis instructor, Drs. Annette Schade, for her stimulating support and encouraging me to go ahead with my thesis.

My former colleagues from the Department of Information Management supported me in my research work. I want to thank them for all their help, support, interest and valuable hints. Especially I am obliged to Drs. Philippe Theunen and Laurens Alberts, MSc.

Finally, I would like to thank the people, who looked closely at the final version of the thesis for English style and grammar, correcting both and offering suggestions for improvement.
## Contents

1 Introduction ................................................. 8
   1.1 Customer segmentation and customer profiling ................. 9
       1.1.1 Customer segmentation ................................ 9
       1.1.2 Customer profiling .................................. 10
   1.2 Data mining ............................................. 11
   1.3 Structure of the report ................................... 13

2 Data collection and preparation ............................... 14
   2.1 Data warehouse ........................................... 14
       2.1.1 Selecting the customers .............................. 14
       2.1.2 Call detail data .................................... 15
       2.1.3 Customer data ...................................... 19
   2.2 Data preparation ......................................... 20

3 Clustering .................................................. 22
   3.1 Cluster analysis ......................................... 22
       3.1.1 The data ............................................ 23
       3.1.2 The clusters ........................................ 23
       3.1.3 Cluster partition .................................... 24
   3.2 Cluster algorithms ........................................ 27
       3.2.1 K-means ............................................. 27
       3.2.2 K-medoid ........................................... 28
       3.2.3 Fuzzy C-means ...................................... 28
       3.2.4 The Gustafson-Kessel algorithm ...................... 29
       3.2.5 The Gath Geva algorithm ............................ 30
   3.3 Validation ................................................. 31
   3.4 Visualization ............................................. 33
       3.4.1 Principal Component Analysis .......................... 33
       3.4.2 Sammon mapping .................................... 34
       3.4.3 Fuzzy Sammon mapping .............................. 35

4 Experiments and results of customer segmentation .......... 37
   4.1 Determining the optimal number of clusters ................... 37
   4.2 Comparing the clustering algorithms ........................ 42
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3</td>
<td>Designing the segments</td>
<td>45</td>
</tr>
<tr>
<td>5</td>
<td>Support Vector Machines</td>
<td>53</td>
</tr>
<tr>
<td>5.1</td>
<td>The separating hyperplane</td>
<td>53</td>
</tr>
<tr>
<td>5.2</td>
<td>The maximum-margin hyperplane</td>
<td>55</td>
</tr>
<tr>
<td>5.3</td>
<td>The soft margin</td>
<td>56</td>
</tr>
<tr>
<td>5.4</td>
<td>The kernel functions</td>
<td>56</td>
</tr>
<tr>
<td>5.5</td>
<td>Multi class classification</td>
<td>59</td>
</tr>
<tr>
<td>6</td>
<td>Experiments and results of classifying the customer segments</td>
<td>60</td>
</tr>
<tr>
<td>6.1</td>
<td>K-fold cross validation</td>
<td>60</td>
</tr>
<tr>
<td>6.2</td>
<td>Parameter setting</td>
<td>61</td>
</tr>
<tr>
<td>6.3</td>
<td>Feature Validation</td>
<td>65</td>
</tr>
<tr>
<td>7</td>
<td>Conclusions and discussion</td>
<td>66</td>
</tr>
<tr>
<td>7.1</td>
<td>Conclusions</td>
<td>66</td>
</tr>
<tr>
<td>7.2</td>
<td>Recommendations for future work</td>
<td>68</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
<td>68</td>
</tr>
<tr>
<td>A</td>
<td>Model of data warehouse</td>
<td>71</td>
</tr>
<tr>
<td>B</td>
<td>Extra results for optimal number of clusters</td>
<td>73</td>
</tr>
</tbody>
</table>
List of Figures

1.1 A taxonomy of data mining tasks .............................. 12

2.1 Structure of customers by Vodafone ......................... 15
2.2 Visualization of phone calls per hour ....................... 17
2.3 Histograms of feature values ................................. 18
2.4 Relation between originated and received calls .......... 18
2.5 Relation between daytime and weekday calls ............... 19

3.1 Example of clustering data ................................. 22
3.2 Different cluster shapes in $R^2$ .......................... 24
3.3 Hard and fuzzy clustering ................................. 25

4.1 Values of Partition Index, Separation Index and the Xie Beni Index 38
4.2 Values of Dunn’s Index and the Alternative Dunn Index ...... 39
4.3 Values of Partition coefficient and Classification Entropy with Gustafson-Kessel clustering ................................. 40
4.4 Values of Partition Index, Separation Index and the Xie Beni Index with Gustafson-Kessel clustering ................................. 41
4.5 Values of Dunn’s Index and Alternative Dunn Index with Gustafson-Kessel clustering ................................. 41
4.6 Result of K-means algorithm ............................... 43
4.7 Result of K-medoid algorithm ................................ 44
4.8 Result of Fuzzy C-means algorithm .......................... 44
4.9 Result of Gustafson-Kessel algorithm ........................ 44
4.10 Result of Gath-Geva algorithm ............................. 45
4.11 Distribution of distances from cluster centers within clusters for the Gath-Geva algorithm with $c = 4$ ................. 46
4.12 Distribution of distances from cluster centers within clusters for the Gustafson-Kessel algorithm with $c = 6$ ................ 46
4.13 Cluster profiles for $c = 4$ .................................. 47
4.14 Cluster profiles for $c = 6$ .................................. 48
4.15 Cluster profiles of centers for $c = 4$ ....................... 49
4.16 Cluster profiles of centers for $c = 6$ ....................... 50

5.1 Two-dimensional customer data of segment 1 and segment 2 .... 54
## List of Tables

2.1 Proportions within the different classification groups .......................... 20

4.1 The values of all the validation measures with K-means clustering 39
4.2 The values of all the validation measures with Gustafson-Kessel clustering .................................................. 42
4.3 The numerical values of validation measures for $c = 4$ ........ 42
4.4 The numerical values of validation measures for $c = 6$ ........ 43
4.5 Segmentation results ................................................................. 51

6.1 Linear Kernel, 4 segments ....................................................... 61
6.2 Linear Kernel, 6 segments ....................................................... 62
6.3 Average C-value for polynomial kernel, 4 segments ................. 62
6.4 Average C-value for polynomial kernel, 6 segments ................. 62
6.5 Polynomial kernel, 4 segments .................................................. 62
6.6 Polynomial kernel, 6 segments .................................................. 62
6.7 Radial basis function, 4 segments ............................................. 63
6.8 Radial basis function, 6 segments ............................................. 63
6.9 Sigmoid function, 4 segments .................................................. 63
6.10 Sigmoid function, 6 segments .................................................. 64
6.11 Confusion matrix, 4 segments .................................................. 64
6.12 Confusion matrix, 6 segments .................................................. 64
Abstract

Vodafone, an International mobile telecommunications company, has accumulated vast amounts of data on consumer mobile phone behavior in a data warehouse. The magnitude of this data is so huge that manual analysis of data is not feasible. However, this data holds valuable information that can be applied for operational and strategical purposes. Therefore, in order to extract such information from this data, automatic analysis is essential, by means of advanced data mining techniques. These data mining techniques search and analyze the data in order to find implicit and useful information, without direct knowledge of human experts. This research will address the question how to perform customer segmentation and customer profiling with data mining techniques. In our context, ‘customer segmentation’ is a term used to describe the process of dividing customers into homogeneous groups on the basis of shared or common attributes (habits, tastes, etc). ’Customer profiling’ is describing customers by their attributes, such as age, gender, income and lifestyles. Having these two components, managers can decide which marketing actions to take for each segment. In this research, the customer segmentation is based on usage call behavior, i.e. the behavior of a customer measured in the amounts of incoming or outgoing communication of whichever form. This thesis describes the process of selecting and preparing the accurate data from the data warehouse, in order to perform customer segmentation and to profile the customer. A number of advanced and state-of-the-art clustering algorithms are modified and applied for creating customer segments. An optimality criterion is constructed in order to measure their performance. The best i.e. most optimal in the sense of the optimality criterion clustering technique will be used to perform customer segmentation. Each segment will be described and analyzed. Customer profiling can be accomplished with information from the data warehouse, such as age, gender and residential area information. Finally, with a recent data mining technique, called Support Vector Machines, the segment of a customer will be estimated based on the customers profile. Different kernel functions with different parameters will be examined and analyzed. The customer segmentation will lead to two solutions. One solution with four segments and one solution with six segments. With the Support Vector Machine approach it is possible in 80.3% of the cases to classify the segment of a customer based on its profile for the situation with four segments. With six segments, a correct classification of 78.5% is obtained.
Chapter 1

Introduction

Vodafone is world’s leading mobile telecommunications company, with approximately 4.1 million customers in The Netherlands. From all these customers a tremendous amount of data is stored. These data include, among others, call detail data, network data and customer data. Call detail data gives a description of the calls that traverse the telecommunication networks, while the network data gives a description of the state of the hardware and software components in the network. The customer data contains information of the telecommunication customers. The amount of data is so great that manual analysis of data is difficult, if not impossible [22]. The need to handle such large volumes of data led to the development of knowledge-based expert systems [17, 22]. These automated systems perform important functions such as identifying network faults and detecting fraudulent phone calls. A disadvantage of this approach is that it is based on knowledge from human experts.

Obtaining knowledge from human experts is a time consuming process, and in many cases, the experts do not have the requisite knowledge [2]. Solutions to these problems were promised by data mining techniques. Data mining is the process of searching and analyzing data in order to find implicit, but potentially useful, information [12]. Within the telecommunication branch, many data mining tasks can be distinguished. Examples of main problems for marketing and sales departments of telecommunication operators are churn prediction, fraud detection, identifying trends in customer behavior and cross selling and up-selling.

Vodafone is interested in a complete different issue, namely customer segmentation and customer profiling and the relation between them. Customer segmentation is a term used to describe the process of dividing customers into homogeneous groups on the basis of shared or common attributes (habits, tastes, etc) [10]. Customer profiling is describing customers by their attributes, such as age, gender, income and lifestyles [1, 10]. Having these two components, marketers can decide which marketing actions to take for each segment and then allocate scarce resources to segments in order to meet specific business objectives.

A basic way to perform customer segmentation is to define segmentations in
advance with knowledge of an expert, and dividing the customers over these segmentations by their best fits. This research will deal with the problem of making customer segmentations without knowledge of an expert and without defining the segmentations in advance. The segmentations will be determined based on (call) usage behavior. To realize this, different data mining techniques, called clustering techniques, will be developed, tested, validated and compared to each other. In this report, the principals of the clustering techniques will be described and the process of determining the best technique will be discussed. Once the segmentations are obtained, for each customer a profile will be determined with the customer data. To find a relation between the profile and the segments, a data mining technique called Support Vector Machines (SVM) will be used. A Support Vector machine is able to estimate the segment of a customer by personal information, such as age, gender and lifestyle. Based on the combination of the personal information (the customer profile), the segment can be estimated and the usage behavior of the customer profile can be determined. In this research, different settings of the Support Vector Machines will be examined and the best working estimation model will be used.

1.1 Customer segmentation and customer profiling

To compete with other providers of mobile telecommunications it is important to know enough about your customers and to know the wants and needs of your customers [15]. To realize this, it is needed to divide customers in segments and to profile the customers. Another key benefit of utilizing the customer profile is making effective marketing strategies. Customer profiling is done by building a customer’s behavior model and estimating its parameters. Customer profiling is a way of applying external data to a population of possible customers. Depending on data available, it can be used to prospect new customers or to recognize existing bad customers. The goal is to predict behavior based on the information we have on each customer [18]. Profiling is performed after customer segmentation.

1.1.1 Customer segmentation

Segmentation is a way to have more targeted communication with the customers. The process of segmentation describes the characteristics of the customer groups (called segments or clusters) within the data. Segmenting means putting the population in to segments according to their affinity or similar characteristics. Customer segmentation is a preparation step for classifying each customer according to the customer groups that have been defined. Segmentation is essential to cope with today’s dynamically fragmenting consumer marketplace. By using segmentation, marketers are more effective in channeling resources and discovering opportunities. The construction of user
segmentations is not an easy task. Difficulties in making good segmentation are [18]:

- **Relevance and quality of data** are essential to develop meaningful segments. If the company has insufficient customer data, the meaning of a customer segmentation in unreliable and almost worthless. Alternatively, too much data can lead to complex and time-consuming analysis. Poorly organize data (different formats, different source systems) makes it also difficult to extract interesting information. Furthermore, the resulting segmentation can be too complicated for the organization to implement effectively. In particular, the use of too many segmentation variables can be confusing and result in segments which are unfit for management decision making. On the other hand, apparently effective variables may not be identifiable. Many of these problems are due to an inadequate customer database.

- **Intuition**: Although data can be highly informative, data analysts need to be continuously developing segmentation hypotheses in order to identify the ‘right’ data for analysis.

- **Continuous process**: Segmentation demands continuous development and updating as new customer data is acquired. In addition, effective segmentation strategies will influence the behavior of the customers affected by them; thereby necessitating revision and reclassification of customers. Moreover, in an e-commerce environment where feedback is almost immediate, segmentation would require almost a daily update.

- **Over-segmentation**: A segment can become too small and/or insufficiently distinct to justify treatment as separate segments.

One solution to construct segments can be provided by data mining methods that belong to the category of clustering algorithms. In this report, several clustering algorithms will be discussed and compared to each other.

### 1.1.2 Customer profiling

Customer profiling provides a basis for marketers to ‘communicate’ with existing customers in order to offer them better services and retaining them. This is done by assembling collected information on the customer such as demographic and personal data. Customer profiling is also used to prospect new customers using external sources, such as demographic data purchased from various sources. This data is used to find a relation with the customer segmentations that were constructed before. This makes it possible to estimate for each profile (the combination of demographic and personal information) the related segment and visa versa. More directly, for each profile, an estimation of the usage behavior can be obtained.

Depending on the goal, one has to select what is the profile that will be relevant to the project. A simple customer profile is a file that contains at least age and
gender. If one needs profiles for specific products, the file would contain product information and/or volume of money spent. Customer features one can use for profiling, are described in [2, 10, 19]:

- **Geographic.** Are they grouped regionally, nationally or globally
- **Cultural and ethnic.** What languages do they speak? Does ethnicity affect their tastes or buying behaviors?
- **Economic conditions, income and/or purchasing power.** What is the average household income or power of the customers? Do they have any payment difficulty? How much or how often does a customer spend on each product?
- **Age and gender.** What is the predominant age group of your target buyers? How many children and what age are in the family? Are more female or males using a certain service or product?
- **Values, attitudes and beliefs.** What is the customers’ attitude toward your kind of product or service?
- **Life cycle.** How long has the customer been regularly purchasing products?
- **Knowledge and awareness.** How much knowledge do customers have about a product, service, or industry? How much education is needed? How much brand building advertising is needed to make a pool of customers aware of offer?
- **Lifestyle.** How many lifestyle characteristics about purchasers are useful?
- **Recruitment method.** How was the customer recruited?

The choice of the features depends also on the availability of the data. With these features, an estimation model can be made. This can be realized by a data mining method called Support Vector Machines (SVM). This report gives an description of SVM’s and it will be researched under which circumstances and parameters a SVM works best in this case.

### 1.2 Data mining

In section 1.1, the term data mining was used. Data mining is the process of searching and analyzing data in order to find implicit, but potentially useful, information [12]. It involves selecting, exploring and modeling large amounts of data to uncover previously unknown patterns, and ultimately comprehensible information, from large databases. Data mining uses a broad family of computational methods that include statistical analysis, decision trees, neural networks, rule induction and refinement, and graphic visualization. Although, data mining tools have been available for a long time, the advances in computer hardware and software, particularly exploratory tools like data visualization and neural
networks, have made data mining more attractive and practical. The typical data mining process consists of the following steps [4]:

- problem formulation
- data preparation
- model building
- interpretation and evaluation of the results

Pattern extraction is an important component of any data mining activity and it deals with relationships between subsets of data. Formally, a pattern is defined as [4]:

*A statement S in L that describes relationships among a subsets of facts \( F_s \) of a given set of facts \( F \), with some certainty C, such that S is simpler than the enumeration of all facts in \( F_s \).*

Data mining tasks are used to extract patterns from large data sets. The various data mining tasks can be broadly divided into six categories as summarized in Figure 1.1. The taxonomy reflects the emerging role of data visualization as a separate data mining task, even as it is used to support other data mining tasks. Validation of the results is also a data mining task. By the fact that the validation supports the other data mining tasks and is always necessary within a research, this task was not mentioned as a separate one. Different data mining tasks are grouped into categories depending on the type of knowledge extracted by the tasks. The identification of patterns in a large data set is the first step to gaining useful marketing insights and marking critical marketing decisions. The data mining tasks generate an assortment of customer and market knowledge which form the core of knowledge management process. The specific tasks to be used in this research are Clustering (for the customer segmentation), Classification (for estimating the segment) and Data visualization.

*Clustering algorithms* produce classes that maximize similarity within clusters but minimize similarity between classes. A drawback of this method is that the number of clusters has to be given in advance. The advantage of clustering is that expert knowledge is not required. For example, based on user behavior data, clustering algorithms can classify the Vodafone customers into "call only" users, "international callers", "SMS only" users etc.

*Classification algorithms* groups customers in predefined classes. For example,
Vodafone can classify its customers based on their age, gender and type of subscription and then target its user behavior.

Data visualization allow data miners to view complex patterns in their customer data as visual objects complete in three or two dimensions and colors. In some cases it is needed to reduce high dimensional data into three or two dimensions. To realize this, algorithms as Principal Component Analysis and Sammon’s Mapping (discussed in Section 3.4) can be used. To provide varying levels of details of observed patterns, data miners use applications that provide advanced manipulation capabilities to slice, rotate or zoom the objects.

1.3 Structure of the report

The report comprises 6 chapters and several appendices. In addition to to this introductory chapter, Chapter 2 describes the process of selecting the right data from the data warehouse. It provides information about the structure of the data and the data warehouse. Furthermore, it gives an overview of the data that is used to perform customer segmentation and customer profiling. It ends with an explanation of the preprocessing techniques that were used to prepare the data for further usage.

In Chapter 3 the process of clustering is discussed. Clustering is a data mining technique, that in this research is used to determine the customer segmentations. The chapter starts with explaining the general process of clustering. Different cluster algorithms will be studied. It also focuses on validation methods, which can be used to determine the optimal number of clusters and to measure the performance of the different cluster algorithms. The chapter ends with a description of visualization methods. These methods are used to analyze the results of the clustering.

Chapter 4 analyzes the different cluster algorithms of Chapter 3. This will be tested with the prepared call detail data as described in Chapter 2. For each algorithm, the optimal numbers of cluster will be determined. Then, the cluster algorithms will be compared to each other and the best algorithm will be chosen to determine the segments. Multiple plots and figures will show the working of the different cluster methods and the meaning of each segment will be described.

Once the segments are determined, with the customer data of Chapter 2, a profile can be made. Chapter 5 delves into a data mining technique called Support Vector Machines. This technique will be used to classify the right segment for each customer profile. Different parameter settings of the Support Vector Machines will be researched and examined in Chapter 6 to find the best working model. Finally, in Chapter 7, the research will be discussed. Conclusions and recommendations are given and future work is proposed.
Chapter 2

Data collection and preparation

The first step (after the problem formulation) in the data mining process is to understand the data. Without such an understanding, useful applications cannot be developed. All data of Vodafone is stored in a data warehouse. In this chapter, the process of collecting the right data from this data warehouse, will be described. Furthermore, the process of preparing the data for customer segmentation and customer profiling will be explained.

2.1 Data warehouse

Vodafone has stored vast amounts of data in a Teradata data warehouse. This data warehouse exists off more than 200 tables. A simplified model of the data warehouse can be found in Appendix A.

2.1.1 Selecting the customers

Vodafone Maastricht is interested in customer segmentation and customer profiling for \textit{(postpaid) business customers}. In general, business customers can be seen as employees of a business that have a subscription by Vodafone in relation with that business. A more precisely view can be found in Figure 2.1. It is clear to see, that prepaid users are always consumers. In the postpaid group, there are captive and non captive users. A non-captive customer is using the Vodafone network but has not a Vodafone subscription or prepaid (called roaming). Vodafone has made an accomplishment with two other telecommunications companies, Debitel and InterCity Mobile Communications (ICMC), that their customers can use the Vodafone network. Debitel customers are always consumers and ICMC customers are always business customers. The ICMC customers will also be involved in this research. A captive customer has a business account if his telephone or subscription is bought in relation with the business
he works. These customers are called business users. In some cases, customers with a consumer account, can have a subscription that is under normal circumstances only available for business users. These customers also count as business users. The total number of (postpaid) business users at Vodafone is more than 800,000. The next sections describe which data of these customers is needed for customer segmentation and profiling.

2.1.2 Call detail data

Every time a call is placed on the telecommunications network of Vodafone, descriptive information about the call is saved as a call detail record. The number of call detail records that are generated and stored is huge. For example, Vodafone customers generate over 20 million call detail records per day. Given that 12 months of call detail data is typically kept on line, this means that hundreds of millions of call detail data will need to be stored at any time. Call detail records include sufficient information to describe the important characteristics of each call. At a minimum, each call detail record will include the originating and terminating phone numbers, the date and the time of the call and the duration of the call. Call detail records are generated in two or three days after the day the calls were made, and will be available almost immediately for data mining. This is in contrast with billing data, which is typically made available only once per month. Call detail records can not be used directly for data mining, since the goal of data applications is to extract knowledge at the customer level, not at the level of individual phone calls [7, 8]. Thus, the call detail records associated with a customer must be summarized into a single record that describes the customer’s calling behavior. The choice of summary variables (features) is critical in order to obtain a useful description of the customer []. To define the features, one can think of the smallest set of variables that describe the complete behavior of a customer. Keywords like what, when, where, how often, who, etc. can help with this process:
• **How?**: How can a customer cause a call detail record? By making a voice call, or sending an SMS (there are more possibilities, but their appearances are so rare that they were not used during this research). The customer can also receive an SMS or voice call.

• **Who?**: Who is the customer calling? Does he call to fixed lines? Does he call to Vodafone mobiles?

• **What?**: What is the location of the customer and the recipient? They can make international phone calls.

• **When?**: When does a customer call? A business customer can call during office daytime, or in private time in the evening or at night and during the weekend.

• **Where?**: Where is the customer calling? Is he calling abroad?

• **How long?**: How long is the customer calling?

• **How often?**: How often does a customer call or receive a call?

Based on these keywords and based on proposed features in the literature [1, 15, 19, 20], a list of features that can be used as a summary description of a customer based on the calls they originate and receive over some time period $P$ is obtained:

1. average call duration
2. average # calls received per day
3. average # calls originated per day
4. % daytime calls (9am - 6pm)
5. % of weekday calls (Monday - Friday)
6. % of calls to mobile phones
7. average # sms received per day
8. average # sms originated per day
9. % international calls
10. % of outgoing calls within the same operator
11. # unique area codes called during $P$
12. # different numbers called during $P$
These twelve features can be used to build customer segments. Such a segment describes a certain behavior of group of customers. For example, customers who use their telephone only at their office could be in a different segment then users that use their telephone also for private purposes. In that case, the segmentation was based on the percentage weekday and daytime calls. Most of the twelve features listed above can be generated in a straightforward manner from the underlying data of the data warehouse, but some features require a little more creativity and operations on the data.

It may be clear that generating useful features, including summary features, is a critical step within the data mining process. Should poor features be generated, data mining will not be successful. Although the construction of these features may be guided by common sense, it should include exploratory data analysis. For example, the use of the time period 9am-6pm in the fourth feature is not based on the commonsense knowledge that the typical workday on a office is from 9am to 5pm. More detailed exploratory data analysis, shown in Figure 2.2 indicates that the period from 9am to 6pm is actually more appropriate for this purpose. Furthermore, for each summary feature, there should be sufficient variance within the data, otherwise distinguish between customers is not possible and the feature is not useful. On the other hand, too much variance hampers the process of segmentation. For some features values, the variance is visible in the following histograms. Figure 2.3 shows that the average call duration, the number of weekday and daytime calls and the originated calls have sufficient variance. Note that the histograms resemble well known distributions. This also indicates that the chosen features are suited for the customer segmentation. Interesting to see is the relation between the number of calls originated and received. First of all, in general, customers originating more calls than receiving. Figure 2.4 demonstrates this, values above the blue line represent customers with more originating calls than receiving calls. In Figure 2.4 is also visible that the customers that originated more calls, receive also more calls in proportion. Another aspect that is simple to figure out is the fact that customer

Figure 2.2: Visualization of phone calls per hour

![Histogram of phone calls per hour](image)
Figure 2.3: Histograms of feature values

Figure 2.4: Relation between originated and received calls
that make more weekday calls also call more at daytime (in proportion). This is plotted in Figure 2.5. It is clear to see that the chosen features contain sufficient variance and that certain relations and different customer behavior are already visible. The chosen features appear to be well chosen and useful for customer segmentation.

![Figure 2.5: Relation between daytime and weekday calls](image)

2.1.3 Customer data

To profile the customer, customer data is needed. The proposed data in Section 1.1.2 is not completely available. Information about lifestyles and income is missing. However, with some creativity, some information can be subtracted from the data warehouse. The information that Vodafone stored in the data warehouse include name and address information and also include other information such as service plan, contract information and telephone equipment information. With this information, the following variables can be used to define a customers profile:

- **Age group**: <25, 25-40, 40-55, >55
- **Gender**: male, female
- **Type telephone**: simple, basic, advanced
- **Type subscription**: basic, advance, expanded
- **Company size**: small, intermediate, big
- **Living area**: (big) city, small city /town
Because a relative small difference in age between customers should show close relationships, the age of the customers has to be grouped. Otherwise, the result of the classification algorithm is too specific to the trainings data [14]. In general, the goal of grouping variables is to reduce the number of variables to a more manageable size and to remove the correlations between each variable. The composition of the groups should be chosen with care. It is of high importance that the sizes of the groups are almost equal (if this is possible) [22]. If there is one group with a sufficient higher amount of customers than other groups, this feature will not increase the performance of the classification. This is caused by the fact that from each segment a relative high number of customers is represented in this group. Based on this feature, the segment of a customer can not be determined. Table 2.1 shows the percentages of customers within the chosen groups. It is clear to see that sizes of the groups were chosen with care.

<table>
<thead>
<tr>
<th>Age:</th>
<th>&lt;25</th>
<th>25-40</th>
<th>40-55</th>
<th>&gt;55</th>
</tr>
</thead>
<tbody>
<tr>
<td>21.2%</td>
<td>29.5%</td>
<td>27.9%</td>
<td>21.4%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Gender:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Male 60.2%</td>
</tr>
<tr>
<td>Female 39.8%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Telephone type:</th>
</tr>
</thead>
<tbody>
<tr>
<td>simple 33.5%</td>
</tr>
<tr>
<td>basic 38.7%</td>
</tr>
<tr>
<td>advanced 27.8%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type of subscription:</th>
</tr>
</thead>
<tbody>
<tr>
<td>simple 34.9%</td>
</tr>
<tr>
<td>advanced 36.0%</td>
</tr>
<tr>
<td>expanded 29.1%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Company size:</th>
</tr>
</thead>
<tbody>
<tr>
<td>small 31.5%</td>
</tr>
<tr>
<td>intermediate 34.3%</td>
</tr>
<tr>
<td>big 34.2%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Living area:</th>
</tr>
</thead>
<tbody>
<tr>
<td>(big) city 42.0%</td>
</tr>
<tr>
<td>small city/town 58.0%</td>
</tr>
</tbody>
</table>

Table 2.1: Proportions within the different classification groups

and the values can be used for defining the customers profile. With this profile, a Support Vector Machine will be used to estimate the segment of the customer. Chapter 5 and Chapter 6 contain information and results of this method.

### 2.2 Data preparation

Before the data can be used for the actual data mining process, it need to cleaned and prepared in a required format. These tasks are [7]:

- Discovering and repairing inconsistent data formats and inconsistent data encoding, spelling errors, abbreviations and punctuation.
- Deleting unwanted data fields. Data may contain many meaningless fields from an analysis point of view, such as production keys and version numbers.
- Interpreting codes into text or replacing text into meaningful numbers.
Data may contain cryptic codes. These codes has to be augmented and replaced by recognizable and equivalent text.

- Combining data, for instance the customer data, from multiple tables into one common variable.
- Finding multiple used fields. A possible way to determine is to count or list all the distinct variables of a field.

The following data preparations were needed during this research:

- Checking abnormal, out of bounds or ambiguous values. Some of these outliers may be correct but this is highly unusual, thus almost impossible to explain.
- Checking missing data fields or fields that have been replaced by a default value.
- Adding computed fields as inputs or targets.
- Mapping continuous values into ranges, e.g. decision trees.
- Normalization of the variables. There are two types of normalization. The first type is to normalize the values between [0,1]. The second type is to normalize the variance to one.
- Converting nominal data (for example yes/no answers) to metric scales.
- Converting from textual to numeral or numeric data.

New fields can be generated through combinations of e.g. frequencies, averages and minimum/maximum values. The goal of this approach is to reduce the number of variables to a more manageable size while also the correlations between each variable will be removed. Techniques used for this purpose are often referred to as factor analysis, correspondence analysis and conjoint analysis [14]. When there is a large amount of data, it is also useful to apply data reduction techniques (data cube aggregation, dimension and numerosity reduction, discretization and concept hierarchy generation). Dimension reduction means that one has to select relevant feature to a minimum set of attributes such that the resulting probability distribution of data classes is a close as possible to the original distribution given the values of all features. For this additional tools may be needed, e.g. exhaustive, random or heuristic search, clustering, decision trees or associations rules.
Chapter 3

Clustering

In this chapter, the used techniques for the cluster segmentation will be explained.

3.1 Cluster analysis

The objective of cluster analysis is the organization of objects into groups, according to similarities among them [13]. Clustering can be considered the most important unsupervised learning method. As every other unsupervised method, it does not use prior class identifiers to detect the underlying structure in a collection of data. A cluster can be defined as a collection of objects which are "similar" between them and "dissimilar" to the objects belonging to other clusters. Figure 3.1 shows this with a simple graphical example. In this case the 3 clusters into which the data can be divided were easily identified. The similarity criterion that was used in this case is distance: two or more objects belong to the same cluster if they are "close" according to a given distance (in this case geometrical distance). This is called distance-based clustering. Another way of clustering is conceptual clustering. Within this method, two or more objects
belong to the same cluster if this one defines a concept common to all that objects. In other words, objects are grouped according to their fit to descriptive concepts, not according to simple similarity measures. In this research, only distance-based clustering algorithms were used.

3.1.1 The data

One can apply clustering techniques to quantitative (numerical) data, qualitative (categoric) data, or a mixture of both. In this research, the clustering of quantitative data is considered. The data, as described in Section 2.1.2, are typically summarized observations of a physical process (call behavior of a customer). Each observation of the customers calling behavior consists of \( n \) measured values, grouped into an \( n \)-dimensional row vector \( x_k = [x_{k1}, x_{k2}, ..., x_{kn}]^T \), where \( x_k \in \mathbb{R}^n \). A set of \( N \) observations is denoted by \( X = \{x_k | k = 1, 2, ..., N\} \), and is represented as an \( N \times n \) matrix:

\[
X = \begin{pmatrix}
x_{11} & x_{12} & \cdots & x_{1n} \\
x_{21} & x_{22} & \cdots & x_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
x_{N1} & x_{N2} & \cdots & x_{Nn}
\end{pmatrix}.
\] (3.1)

In pattern recognition terminology, the rows of \( X \) are called patterns or objects, the columns are called the features or attributes, and \( X \) is called the pattern matrix. In this research, \( X \) will be referred to the data matrix. The rows of \( X \) represent the customers, and the columns are the feature variables of their behavior as described in Section 2.1.2. As mentioned before, the purpose of clustering is to find relationships between independent system variables, called the regressors, and future values of dependent variables, called the regressands. However, one should realize, that the relations revealed by clustering are not more than associations among the data vectors. And therefore, they will not automatically constitute a prediction model of the given system. To obtain such a model, additional steps are needed.

3.1.2 The clusters

The definition of a cluster can be formulated in various ways, depending on the objective of the clustering. In general, one can accept the definition that a cluster is a group of objects that are more similar to another than to members of other clusters. The term "similarity" can be interpreted as mathematical similarity, measured in some well-defined sense. In metric spaces, similarity is often defined by means of a distance norm, or distance measure. Distance can be measured in different ways. The first possibility is to measure among the data vectors themselves. A second way is to measure the distance form the data vector to some prototypical object of the cluster. The cluster centers are usually (and also in this research) not known a priori, and will be calculated by the clustering algorithms simultaneously with the partitioning of the data.
The cluster centers may be vectors of the same dimensions as the data objects, but can also be defined as "higher-level" geometrical objects, such as linear or nonlinear subspaces or functions. Data can reveal clusters of different geometrical shapes, sizes and densities as demonstrated in Figure 3.2. Clusters can be spherical, elongated and also be hollow. Cluster can be found in any n-dimensional space. Clusters a, c and d can be characterized as linear and non linear subspaces of the data space ($R^2$ in this case). Clustering algorithms are able to detect subspaces of the data space, and therefore reliable for identification. The performance of most clustering algorithms is influenced not only by the geometrical shapes and densities of the individual clusters, but also by the spatial relations and distances among the clusters. Clusters can be well-separated, continuously connected to each other, or overlapping each other.

### 3.1.3 Cluster partition

Clusters can formally be seen as subsets of the data set. One can distinguish two possible outcomes of the classification of clustering methods. Subsets can
either be fuzzy or crisp (hard). Hard clustering methods are based on the classical set theory, which requires that an object either does or does not belong to a cluster. Hard clustering in a data set \( X \) means partitioning the data into a specified number of exclusive subsets of \( X \). The number of subsets (clusters) is denoted by \( c \). Fuzzy clustering methods allow objects to belong to several clusters simultaneously, with different degrees of membership. The data set \( X \) is thus partitioned into \( c \) fuzzy subsets. In many real situations, fuzzy clustering is more natural than hard clustering, as objects on the boundaries between several classes are not forced to fully belong to one of the classes, but rather are assigned membership degrees between 0 and 1 indicating their partial memberships (illustrated by Figure 3.3 The discrete nature of hard partitioning also causes analytical and algorithmic intractability of algorithms based on analytic functionals, since these functionals are not differentiable. The structure of the partition matrix \( U = [\mu_{ik}] \):

\[
U = \begin{pmatrix}
\mu_{1,1} & \mu_{1,2} & \cdots & \mu_{1,c} \\
\mu_{2,1} & \mu_{2,2} & \cdots & \mu_{2,c} \\
\vdots & \vdots & \ddots & \vdots \\
\mu_{N,1} & \mu_{N,2} & \cdots & \mu_{N,c}
\end{pmatrix}.
\]

(3.2)

**Hard partition**

The objective of clustering is to partition the data set \( X \) into \( c \) clusters. Assume that \( c \) is known, e.g. based on prior knowledge, or it is a trial value, of which partition results must be validated. Using classical sets, a hard partition can be seen as a family of subsets \( \{A_i | 1 \leq i \leq c \subset P(X) \} \), its properties can be defined as follows:

\[
\bigcup_{i=1}^{c} A_i = X, \quad (3.3)
\]

\[
A_i \cap A_j, 1 \leq i \neq j \leq c, \quad (3.4)
\]

\[
\varnothing \subset A_i \subset X, 1 \leq i \leq c. \quad (3.5)
\]
These conditions imply that the subsets $A_i$ contain all the data in $X$, they must be disjoint and none of them is empty nor contains all the data in $X$. Expressed in the terms of membership functions:

$$\bigvee_{i=1}^{c} \mu_{A_i} = 1, \quad (3.6)$$

$$\mu_{A_i} \lor \mu_{A_j}, 1 \leq i \neq j \leq c, \quad (3.7)$$

$$0 \leq \mu_{A_i} < 1, 1 \leq i \leq c. \quad (3.8)$$

Where $\mu_{A_i}$ represents the characteristic function of the subset $A_i$ which value is zero or one. To simplify these notations, $\mu_i$ will be used instead of $\mu_{A_i}$, and denoting $\mu_i(x_k)$ by $\mu_{ik}$, partitions can be represented in a matrix notation. $U = [\mu_{ik}]$, a $N \times c$ matrix, is a representation of the hard partition if and only if its elements satisfy:

$$\mu_{ij} \in \{0, 1\}, 1 \leq i \leq N, 1 \leq k \leq c, \quad (3.9)$$

$$\sum_{k=1}^{c} \mu_{ik} = 1, 1 \leq i \leq N, \quad (3.10)$$

$$0 < \sum_{i=1}^{N} \mu_{ik} < N, 1 \leq k \leq c. \quad (3.11)$$

A definition of a hard partitioning space can be defined as follows:

Let $X$ be a finite data set and the number of clusters $2 \leq c < N \in \mathbb{N}$. Then, the hard partitioning space for $X$ can be seen as the set:

$$M_{hc} = \{ U \in \mathbb{R}^{N \times c} | \mu_{ik} \in \{0, 1\}, \forall i, k; \sum_{k=1}^{c} \mu_{ik} = 1, \forall i; 0 < \sum_{i=1}^{N} \mu_{ik} < N, \forall k \}. \quad (3.12)$$

**Fuzzy partition**

Fuzzy partition can be defined as a generalization of hard partitioning, in this case $\mu_{ik}$ is allowed to acquire all real values between zero and 1. Consider the matrix $U = [\mu_{ik}]$, containing the fuzzy partitions, its conditions are given by:

$$\mu_{ij} \in [0, 1], 1 \leq i \leq N, 1 \leq k \leq c, \quad (3.13)$$

$$\sum_{k=1}^{c} \mu_{ik} = 1, 1 \leq i \leq N, \quad (3.14)$$

$$0 < \sum_{i=1}^{N} \mu_{ik} < N, 1 \leq k \leq c. \quad (3.15)$$

Note that there is only one difference with the conditions of the hard partitioning. Also the definition of the fuzzy partitioning space will not much differ with
the definition of the hard partitioning space. It can be defined as follows: Let $X$ be a finite data set and the number of clusters $2 \leq c < N \in \mathbb{N}$. Then, the fuzzy partitioning space for $X$ can be seen as the set:

$$M_{fc} = \{ U \in \mathbb{R}^{N \times c} | \mu_{ik} \in [0, 1], \forall i, k; \sum_{k=1}^{c} \mu_{ik} = 1, \forall i; 0 < \sum_{i=1}^{N} \mu_{ik} < N, \forall k \}.$$  (3.16)

The $i$-th column of $U$ contains values of the membership functions of the $i$-th fuzzy subset of $X$. Equation (1.14) implies that the sum of each column should be 1, which means that the total membership of each $x_k$ in $X$ equals one. There are no constraints on the distribution of memberships among the fuzzy clusters. This research will focus on hard partitioning. However, fuzzy cluster algorithms will be applied as well. To deal with the problem of fuzzy memberships, the cluster with the highest degree of membership will be chosen as the cluster where the object belongs to. This method will result into hard partitioned clusters. The possibilistic partition will not be used in this researched and will not be discussed here.

### 3.2 Cluster algorithms

This section gives an overview of the clustering algorithms that were used during the research.

#### 3.2.1 K-means

K-means is one of the simplest unsupervised learning algorithms that solves the clustering problem. However, the results of this hard partitioning method are not always reliable and this algorithm has numerical problems as well. The procedure follows an easy way to classify a given $N \times n$ data set through a certain numbers of $c$ clusters defined in advance. The K-means algorithm allocates each data point to one of the $c$ clusters to minimize the within sum of squares:

$$\sum_{i=1}^{c} \sum_{k \in A_i} ||x_k - v_i||^2.$$  (3.17)

$A_i$ represents a set of data points in the $i$-th cluster and $v_i$ is the average of the data points in cluster $i$. Note that $||x_k - v_i||^2$ is actually a chosen distance norm. Within the cluster algorithms, $v_i$ is the cluster center (also called prototype) of cluster $i$:

$$v_i = \frac{\sum_{k=1}^{N_{i}} x_k}{N_{i}}, x_k \in A_i,$$  (3.18)

where $N_i$ is the number of data points in $A_i$. 

27
3.2.2 K-medoid

K-medoid clustering, also a hard partitioning algorithm, uses the same equations as the K-means algorithm. The only difference is that in K-medoid the cluster centers are the nearest data points to the mean of the data in one cluster, \( V = \{ v_i \in X \mid 1 \leq i \leq c \} \). This can be useful when, for example, there is no continuity in the data space. This implies that a mean of the points in one cluster does actually not exist.

3.2.3 Fuzzy C-means

The Fuzzy C-means algorithm (FCM) minimizes an objective function, called C-means functional, to define the clusters. The C-means functional, invented by Dunn, is defined as follows:

\[
J(X; U, V) = \sum_{i=1}^{c} \sum_{k=1}^{N} (\mu_{ik})^m ||x_k - v_i||_A^2,
\]

with

\[V = [v_1, v_2, ..., v_c], v_i \in R^n.\]

\(V\) denotes the vector with the cluster centers that has to be determined. The distance norm \( ||x_k - v_i||^2_A \) is called a squared inner-product distance norm and is defined by:

\[
D_{ikA} = ||x_k - v_i||^2_A = (x_k - v_i)^T A (x_k - v_i).
\]

On a statistical point of view, equation 3.19 measures the total number of variance of \( x_k \) from \( v_i \). The minimization of the C-means functional can be seen as a non linear optimization problem, that can be solved by a variety of methods. Examples of methods that can solve non linear optimization problems are grouped coordinate minimization and genetic algorithms. The simplest method to solve this problem is utilizing the Picard iteration through the first-order conditions for the stationary points of equation 3.19. This method is called the fuzzy c-means algorithm. To find the stationary points of the c-means functional, one can adjoint the constrained in 3.14 to \( \bar{J} \) by means of Lagrange multipliers:

\[
\bar{J}(X; U, V, \lambda) = \sum_{i=1}^{c} \sum_{k=1}^{N} (\mu_{ik})^m D_{ikA}^2 + \sum_{k=1}^{N} \lambda_k \left( \sum_{i=1}^{c} \mu_{ik} - 1 \right),
\]

and by setting the gradients of \( \bar{J} \), with respect to \( U, V \) and \( \lambda \), to zero. When \( D_{ikA}^2 > 0, \forall i, k \) and \( m > 1 \), then the C-means functional may only be minimized by \((U, V) \in M_{fc}x R^{nxc}\) if

\[
\mu_{ik} = \frac{1}{\sum_{j=1}^{c} (D_{ikA}/D_{jkA})^{2/(m-1)}}, 1 \leq i \leq c, 1 \leq k \leq N,
\]

28
and

\[ v_i = \frac{\sum_{k=1}^{N} \mu_{ik}^m x_k}{\sum_{k=1}^{N} \mu_{ik}^m}, 1 \leq i \leq c. \tag{3.24} \]

The solution of these equations are satisfying the constraints that were given in equation (3.13) and (3.15). Remark that the \( v_i \) of equation (3.24) is the weighted average of the data points that belong to a cluster and the weights represents the membership degrees. This explains why the name of the algorithm is \( c \)-means. The Fuzzy C-means algorithm is actually an iteration between the equations (3.23) and (3.24). The FCM algorithm uses the standard Euclidean distance for its computations. Therefore, it is able to define hyper spherical clusters. Note that it can only detect clusters with the same shape, caused by the common choice of the norm inducing matrix \( A = I \). The norm inducing matrix can also be chosen as an \( n \times n \) diagonal matrix of the form:

\[
A_D = \begin{pmatrix}
(1/\sigma_1)^2 & 0 & \cdots & 0 \\
0 & (1/\sigma_2)^2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & (1/\sigma_n)^2
\end{pmatrix}. \tag{3.25}
\]

This matrix accounts for different variances in the directions of the coordinate axes of \( X \). Another possibility is to choose \( A \) as the inverse of the \( n \times n \) covariance matrix \( A = F^{-1} \), where

\[
F = \frac{1}{N} \sum_{k=1}^{N} (x_k - \hat{x})(x_k - \hat{x})^T \tag{3.26}
\]

and \( \hat{x} \) denotes the mean of the data. Hence that, in this case, matrix \( A \) is based on the Mahalanobis distance norm.

### 3.2.4 The Gustafson-Kessel algorithm

The Gustafson and Kessel (GK) algorithm is a variation on the Fuzzy c-means algorithm [11]. It employs a different and adaptive distance norm to recognize geometrical shapes in the data. Each cluster will have its own norm-inducing matrix \( A_i \), satisfying the following inner-product norm:

\[
D_{ik}^2 = (x_k - v_i)^T A_i (x_k - v_i), \text{ where } 1 \leq i \leq c \text{ and } 1 \leq k \leq N. \tag{3.27}
\]

The matrices \( A_i \) are used as optimization variables in the c-means functional. This implies that each cluster is allowed to adapt the distance norm to the local topological structure of the data. A \( c \)-tuple of the norm-inducing matrices is defined by \( A \), where \( A = (A_1, A_2, ..., A_c) \). The objective functional of the GK algorithm can be calculated by:

\[
J(X; U, V, A) = \sum_{i=1}^{c} \sum_{k=1}^{N} (u_{ik})^m D_{ik}^2 A_i. \tag{3.28}
\]
If $A$ is fixed, the conditions under (3.13), (3.14) and (3.15) can be applied without any problems. Unfortunately, equation (3.28) cannot be minimized in a straightforward manner, since it is linear in $A_i$. This implies that $J$ can be made as small as desired by making $A_i$ less positive definite. To avoid this, $A_i$ has to be constrained to obtain a feasible solution. A general way to this is by constraining the determinant of the matrix. A varying $A_i$ with a fixed determinant relates to the optimization of the cluster with a fixed volume:

$$||A_i|| = \rho_i, \rho > 0.$$  \hspace{1cm} (3.29)

Here $\rho$ is a remaining constant for each cluster. In combination with the Lagrange multiplier, $A_i$ can be expressed in the following way:

$$A_i = [\rho_i \det(F_i)]^{1/n} F^{-1}_i,$$  \hspace{1cm} (3.30)

with

$$F_i = \sum_{k=1}^{N} (\mu_{ik})^w (x_k - v_i)(x_k - v_i)^T \sum_{k=1}^{N} (\mu_{ik})^w,$$  \hspace{1cm} (3.31)

$F_i$ is also called the fuzzy covariance matrix. Hence that this equation in combination with equation (3.30) can be substituted into equation (3.27). The outcome of the inner-product norm of (3.27) is a generalized squared Mahalanobis norm between the data points and the cluster center. The covariance is weighted by the membership degrees of $U$.

3.2.5 The Gath Geva algorithm

Bezdek and Dunn [5] proposed a fuzzy maximum likelihood estimation (FMLE) algorithm with a corresponding distance norm:

$$D_{ik}(x_k, v_i) = \left( \frac{\sqrt{\det(F_{wi})}}{\alpha_i} \right) \left( \frac{1}{2} (x_k - v_i^{(l)})^T F_{wi}^{-1} (x_k - v_i^{(l)}) \right),$$  \hspace{1cm} (3.32)

Comparing this with the Gustafson-Kessel algorithm, the distance norm includes an exponential term. This implies that this distance norm will decrease faster than the inner-product norm. In this case, the fuzzy covariance matrix $F_i$ is defined by:

$$F_{wi} = \sum_{k=1}^{N} (\mu_{ik})^w (x_k - v_i)(x_k - v_i)^T \sum_{k=1}^{N} (\mu_{ik})^w, 1 \leq i \leq c.$$  \hspace{1cm} (3.33)

The reason for using the $w$ variable is to generalize this expression. In the original FMLE algorithm, $w = 1$. In this research, $w$ will be set to 2, to compensate the exponential term and obtain clusters that are more fuzzy. Because of the generalization, two weighted covariance matrices arise. The variable $\alpha_i$ in equation (3.32) is the prior probability of selecting cluster $i$. $\alpha_i$ can be defined as follows:

$$\alpha_i = \frac{1}{N} \sum_{k=1}^{N} \mu_{ik}.$$  \hspace{1cm} (3.34)
Gath and Geva [9] discovered that the FMLE algorithm is able to detect clusters of different shapes, sizes and densities and that the clusters are not constrained in volume. The main drawback of this algorithm is the robustness, since the exponential distance norm can converge to a local optimum. Furthermore, it is not know how reliable the results of this algorithm are.

3.3 Validation

Cluster validation refers to the problem whether a found partition is correct and how to measure the correctness of a partition. A clustering algorithm is designed to parameterize clusters in a way that it gives the best fit. However, this does not apply that the best fit is meaningful at all. The number of clusters might not be correct or the cluster shapes do not correspond to the actual groups in the data. In the worst case, the data can not be grouped in a meaningful way at all. One can distinguish two main approaches to determine the correct number of clusters in the data:

- Start with a sufficiently large number of clusters, and successively reducing this number by combining clusters that have the same properties.

- Cluster the data for different values of $c$ and validate the correctness of the obtained clusters with validation measures.

To be able to perform the second approach, validation measures has to be designed. Different validation methods have been proposed in the literature, however, none of them is perfect by oneself. Therefore, in this research are used several indexes, which are described below:

- **Partition Coefficient (PC)**: measures the amount of ”overlapping” between clusters. It is defined by Bezdek [5] as follows:

$$PC(c) = \frac{1}{N} \sum_{i=1}^{c} \sum_{j=1}^{N} (u_{ij})^2$$

where $u_{ij}$ is the membership of data point $j$ in cluster $i$. The main drawback of this validity measure is the lack of direct connection to the data itself. The optimal number of clusters can be found by the maximum value.

- **Classification Entropy (CE)**: measures only the fuzziness of the cluster, which is a slightly variation on the Partition Coefficient.

$$CE(c) = -\frac{1}{N} \sum_{i=1}^{c} \sum_{j=1}^{N} u_{ij} \log(u_{ij})$$
• **Partition Index (PI):** expresses the ratio of the sum of compactness and separation of the clusters. Each individual cluster is measured with the cluster validation method. This value is normalized by dividing it by the fuzzy cardinality of the cluster. To receive the Partition index, the sum of the value for each individual cluster is used.

\[ PI(c) = \frac{\sum_{i=1}^{c} \sum_{j=1}^{N} (u_{ij})^m ||x_j - v_i||^2}{N_i \sum_{k=1}^{N} ||v_k - v_i||^2} \]  

(3.37)

*PI* is mainly used for the comparing of different partitions with the same number of clusters. A minor value of a SC means a better partitioning.

• **Separation Index (SI):** in contrast with the partition index (PI), the separation index uses a minimum-distance separation to validate the partitioning.

\[ SI(c) = \frac{\sum_{i=1}^{c} \sum_{j=1}^{N} (u_{ij})^2 ||x_j - v_i||^2}{N \min_{i,k} ||v_k - v_i||^2} \]  

(3.38)

• **Xie and Beni’s Index (XB):** is a method to quantify the ratio of the total variation within the clusters and the separations of the clusters [3].

\[ XB(c) = \frac{\sum_{i=1}^{c} \sum_{j=1}^{N} (u_{ij})^m ||x_j - v_i||^2}{N \min_{i,j} ||x_j - v_i||^2} \]  

(3.39)

The lowest value of the XB index should indicate the optimal number of clusters.

• **Dunn’s Index (DI):** this index was originally designed for the identification of hard partitioning clustering. Therefore, the result of the clustering has to be recalculated.

\[ DI(c) = \min_{i \in c} \\{ \min_{j \in c, i \neq j} \{ \frac{\min_{x \in C_i, y \in C_j} d(x, y)}{\max_{k \in C} \{ \max_{x, y \in C} d(x, y) \}} \} \} \]  

(3.40)

The main disadvantage of the Dunn’s index is the very expansive computational complexity as *c* and *N* increase.

• **Alternative Dunn Index (ADI):** To simplify the calculation of the Dunn index, the Alternative Dunn Index was designed. This will be the case when the dissimilarity between two clusters, measured with \( \min_{x \in C_i, y \in C_j} d(x, y) \), is rated in under bound by the triangle-inequality:

\[ d(x, y) \geq |d(y, v_j) - d(x, v_j)| \]  

(3.41)

were \( v_j \) represents the cluster center of the \( j \)-th cluster.

\[ ADI(c) = \min_{i \in c} \\{ \min_{j \in c, i \neq j} \{ \frac{\min_{x \in C_i, y \in C_j} |d(y, v_j) - d(x, v_j)|}{\max_{k \in C} \{ \max_{x, y \in C} d(x, y) \}} \} \} \]  

(3.42)
Note, that the Partition Coefficient and the Classification Entropy are only useful for fuzzy partitioned clustering. In case of fuzzy clusters the values of the Dunn’s Index and the Alternative Dunn Index are not reliable. This is caused by the repartitioning of the results with the hard partition method.

3.4 Visualization

To understand the data and the results of the clustering methods, it is useful to visualize the data and the results. However, the used data set is a high-dimensional data set, which cannot be plotted and visualized directly. This section describes three methods that can map the data points into a lower dimensional space.

In this research, the three mapping methods will be used for the visualization of the clustering results. The first method is the Principal Component Analysis (PCA), a standard and a most widely method to map high-dimensional data into a lower dimensional space. Then, this report will focus on the Sammon mapping method. The advantage of the Sammon mapping is the ability to preserve inter pattern distances. This kind of mapping of distances is much closer related to the proposition of clustering than saving the variances (which will be done by PCA). However, the Sammon mapping application has two main drawbacks:

- Sammon mapping is a projection method, which is based on the preservation of the Euclidean inter point distance norm. This implies that the Sammon mapping only can be applied on clustering algorithms that use the Euclidean distance norm during the calculations of the clusters.

- The Sammon mapping method aims to find in a high \( n \)-dimensional space \( N \) points in a lower \( q \)-dimensional subspace, such in a way the inter point distances correspond to the distances measured in the \( n \)-dimensional space. To achieve this, a computational expensive algorithm is needed, because in every iteration step a computation of \( N(N - 1)/2 \) distances is required.

To avoid these problems of the Sammon mapping method, a modified algorithm, called the Fuzzy Sammon mapping, is used during this research. A drawback of this Fuzzy Sammon mapping is the loose of precision in distance, since only the distance between the data points and the cluster centers considered to be important.

The three visualization methods will be explained in more detail in the following subsections.

3.4.1 Principal Component Analysis

Principal component analysis (PCA) include a mathematical procedure that maps a number of correlated variables into a smaller set of uncorrelated variables, called the principal components. The first principal component represents
as much of the variability in the data as possible. The succeeding components describe the remaining variability. The main goals of the PCA method are:

- Identifying new meaningful underlying variables.
- Discovering and/or reducing the dimensionality of a data set.

In a mathematical way, the principal components will be achieved by analyzing the eigenvectors and eigenvalues. The direction of the first principal component is diverted from the eigenvector with the largest eigenvalue. The eigenvalue associated with the second largest eigenvalue correspond to the second principal component, etc. In this research, the second objective is used. In this case, the covariance matrix of the data set can be described by:

$$F = \frac{1}{N}(x_k - v)(x_k - v)^T,$$

(3.43)

where $v = \bar{x}_k$. Principal Component Analysis is based on the projection of correlated high-dimensional data onto a hyperplane [3]. This methods uses only the first $q$ nonzero eigenvalues and the corresponding eigenvectors of the covariance matrix:

$$F_i = U_i\Lambda_i U_i^T.$$

(3.44)

With $\Lambda_i$ as a matrix that contains the eigenvalues $\lambda_{i,j}$ of $F_i$ in its diagonal in decreasing order and $U_i$ is a matrix containing the eigenvectors corresponding to the eigenvalues in its columns. Furthermore, there is a $q$-dimensional reduced vector that represents the vector $x_k$ of $X$, which can be defined as follows:

$$y_{i,k} = W_i^{-1}(x_k) = W_i^T(x_k).$$

(3.45)

The weight matrix $W_i$ contains the $q$ principal orthonormal axes in its column:

$$W_i = U_{i,q} \Lambda_{i,q}^{\frac{1}{2}}.$$

(3.46)

### 3.4.2 Sammon mapping

As mentioned before, the Sammon mapping uses inter point distance measures to find $N$ points in a $q$-dimensional data space, which are representative for a higher $n$-dimensional data set. The inter point distance measure of the $n$-dimensional space, defined by $d_{ij} = d(x_i, x_j)$ correspond to the inter point distances in the $q$-dimensional space, given by $d_{ij}^* = d^*(y_i, y_j)$. This is achieved by Sammon’s stress, a minimization criterion of the error:

$$E = \frac{1}{\lambda} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \frac{(d_{ij} - d_{ij}^*)^2}{d_{ij}},$$

(3.47)

where $\lambda$ is a constant:

$$\lambda = \sum_{i<j} d_{ij} = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d_{ij}.$$

(3.48)
Note that there is no need to maintain $\lambda$, since a constant does not change the result of the optimization process. The minimization of the error $E$ is an optimization problem in the $N \times q$ variables $y_{il}$, with $i \in \{1, 2, ..., N\}$ and $l \in \{1, 2, ..., q\}$ which implies that $y_i = [y_{i1}, ..., y_{iq}]^T$. The rating of $y_{il}$ at the $t$-th iteration can be defined by:

$$y_{il}(t + 1) = y_{il}(t) - \alpha \left[ \frac{\partial E(t)}{\partial y_{il}(t)} \right], \quad (3.49)$$

where $\alpha$ is a nonnegative scalar constant, with a recommended value $\alpha \approx 0.3 - 0.4$. This scalar constant represents the step size for gradient search in the direction of

$$\frac{\partial E(t)}{\partial y_{il}(t)} = -\frac{2}{\lambda} \sum_{k=1, k \neq i}^N \left[ \frac{d_{ki} - d_{ki}^*}{d_{ki}d_{ki}^*} \right] (y_{il} - y_{ki}) \quad (3.50)$$

$$\frac{\partial^2 E(t)}{\partial y_{il}^2(t)} = -\frac{2}{\lambda} \sum_{k=1, k \neq i}^N \frac{1}{d_{ki}d_{ki}^*} \left[ (d_{ki} - d_{ki}^*) - \left( \frac{(y_{il} - y_{ki})^2}{d_{ki}^*} \right) \left( 1 + \frac{d_{ki} - d_{ki}^*}{d_{ki}} \right) \right] \quad (3.51)$$

With this gradient-descent method, it is not possible to reach a local minimum in the error surface, while searching for the minimum of $E$. This is a disadvantage, because multiple experiments with different random initializations are necessary to find the minimum. However, it is possible to estimate the correct initialization based on the information which is obtained from the data.

### 3.4.3 Fuzzy Sammon mapping

As mentioned in the introduction of this section, Sammon’s mapping has several drawbacks. To avoid these drawbacks, a modified mapping method is designed which takes into account the basic properties of fuzzy clustering algorithms where only the distance between the data points and the clustering centers are considered to be important [3]. The modified algorithm, called Fuzzy Sammon mapping, uses only $N \times c$ distances, weighted by the membership values similarly to equation (3.19):

$$E_{fuzz} = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ki})^m (d(x_k, v_i) - d_{ki}^*)^2, \quad (3.52)$$

with $d(x_k, v_i)$ representing the distance between data point $x_k$ and the cluster center $v_i$ in the original $n$-dimensional space. The Euclidean distance between the cluster center $z_i$ and the data $y_k$ of the projected $q$-dimensional space is represented by $d^*(y_k, z_i)$. According to this information, in a projected two dimensional space every cluster is represented by a single point, independently to the shape of the original cluster. The Fuzzy Sammon mapping algorithm is similar to the original Sammon mapping, but in this case the projected cluster.
The center will be recalculated in every iteration after the adaptation of the projected data points. The recalculation will be based on the weighted mean formula of the fuzzy clustering algorithms, described in Section 3.2.3 (equation 3.19). The membership values of the projected data can be plotted based on the standard equation for the calculation of the membership values:

$$\mu_{ki}^* = \frac{1}{\sum_{j=1}^{c} \left( \frac{d^*(x_k, \eta_j)}{d^*(x_k, v_j)} \right)^{2^{-1}}},$$

(3.53)

where $U^* = [\mu_{ki}^*]$ is the partition matrix with the recalculated memberships.

The plot will only give an approximation of the high dimensional clustering in a two dimensional space. To measure the quality of this rating, an evaluation function that determines the mean square error between the original and the recalculated error can be defined as follows:

$$P = ||U - U^*||.$$  

(3.54)

In the next chapter, the cluster algorithms will be tested and evaluated. The PCA and the (Fuzzy) Sammon mapping methods will be used to visualize the data and the clusters.
Chapter 4

Experiments and results of customer segmentation

In this chapter, the cluster algorithms will be tested and their performance will be measured with the proposed validation methods of the previous chapter. The best working cluster method will be used to determine the segments. The chapter ends with an evaluation of the segments.

4.1 Determining the optimal number of clusters

The disadvantage of the proposed cluster algorithms is the number of clusters that has to be given in advance. In this research the number of clusters is not known. Therefore, the optimal number of clusters has to be searched with the given validation methods of Section 3.3. For each algorithm, calculations for each cluster, $c \in [215]$, were executed. To find the optimal number of clusters, a process called Elbow Criterion is used. The elbow criterion is a common rule of thumb to determine what number of clusters should be chosen. The elbow criterion says that one should choose a number of clusters so that adding another cluster does not add sufficient information. More precisely, by graphing a validation measure explained by the clusters against the number of clusters, the first clusters will add much information (explain a lot of variance), but at some point the marginal gain will drop, giving an angle in the graph (the elbow). Unfortunately, this elbow can not always be unambiguously identified. To demonstrate the working of the elbow criterion, the feature values that represent the call behavior of the customers, as described in Section 2.1.2, are used as input for the cluster algorithms. From the 800,000 business customers of Vodafone, 25,000 customers were randomly selected for the experiments. More customers would lead to computational problems. First, the K-means algorithm will be evaluated. The values of the validation methods depending on the number of clusters will be plotted. The value of the Partition Coefficient is for all
clusters 1, and the classification entropy is always 'NaN'. This is caused by the fact that these 2 measures were designed for fuzzy partitioning methods, and in this case the hard partitioning algorithm K-means is used. In Figure 4.1, the values of the Partition Index, Separation Index and Xie and Beni’s Index are shown. Mention again, that no validation index is reliable only by itself.

Figure 4.1: Values of Partition Index, Separation Index and the Xie Beni Index

Therefor, all the validation indexes are shown. The optimum could differ by using different validation methods. This means that the optimum only could be detected by the comparison of all the results. To find the optimal number of cluster, partitions with less clusters are considered better, when the difference between the values of the validation measure are small. Figure 4.1 shows that for the PI and SI, the number of clusters easily could be rated to 4. For the Xie and Beni index, this is much harder. The elbow can be found at \( c = 3, c = 6, c = 9 \) or \( c = 13 \), depending on the definition and parameters of an elbow. In Figure 4.2 there are more informative plots shown. The Dunn’s index and the Alternative Dunn’s index confirm that the optimal number of clusters for the K-means algorithm should be chosen to 4. The values of all the validation measures for the K-means algorithm, are embraced in table 4.1
Table 4.1: The values of all the validation measures with K-means clustering

<table>
<thead>
<tr>
<th>c</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>CE</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>PI</td>
<td>3.8318</td>
<td>1.9109</td>
<td>1.1571</td>
<td>1.0443</td>
<td>1.2907</td>
<td>0.9386</td>
<td>0.8828</td>
</tr>
<tr>
<td>SI</td>
<td>0.0005</td>
<td>0.0003</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
</tr>
<tr>
<td>XBI</td>
<td>5.4626</td>
<td>4.9519</td>
<td>5.0034</td>
<td>4.3353</td>
<td>3.9253</td>
<td>4.2214</td>
<td>3.9079</td>
</tr>
<tr>
<td>DI</td>
<td>0.0082</td>
<td>0.0041</td>
<td>0.0034</td>
<td>0.0065</td>
<td>0.0063</td>
<td>0.0072</td>
<td>0.0071</td>
</tr>
<tr>
<td>ADI</td>
<td>0.0018</td>
<td>0.0013</td>
<td>0.0002</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>c</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>CE</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>PI</td>
<td>0.8362</td>
<td>0.8261</td>
<td>0.8384</td>
<td>0.7783</td>
<td>0.7696</td>
<td>0.7557</td>
<td>0.7489</td>
</tr>
<tr>
<td>SI</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
</tr>
<tr>
<td>DI</td>
<td>0.0071</td>
<td>0.0052</td>
<td>0.0061</td>
<td>0.0070</td>
<td>0.0061</td>
<td>0.0061</td>
<td>0.0061</td>
</tr>
<tr>
<td>ADI</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

Figure 4.2: Values of Dunn’s Index and the Alternative Dunn Index
It is also possible to define the optimal numbers of clusters for fuzzy clustering algorithms with this method. To illustrate this, the results of the Gustafson-Kessel algorithm will be shown. In Figure 4.3 the results of the Partition Index and the Classification Entropy are plotted. Compared to the hard clustering methods, the validation methods can be used now for the fuzzy clustering. However, the main drawback of PC is the monotonic decreasing with $c$, which makes it hardly to detect the optimal number of cluster. The same problem holds for CE: monotonic increasing, caused by the lack of direct connection to the data. The optimal number of cluster can not be rated based on those two validation methods. Figure 4.4 gives more information about the optimal number of clusters. For the PI and the SI, the local minimum is reached at $c = 6$. Again, for the XBI, it is difficult to find the optimal number of clusters. The points at $c = 3$, $c = 6$ and $c = 11$, can be seen as an elbow. In Figure 4.5, the Dunn index also indicates that the optimal number of clusters should be at $c = 6$. On the other hand, the Alternative Dunn index, has an elbow at the point $c = 3$. However, for the Alternative Dunn Index is not known how reliable its results are, so the optimal number of clusters for the Gustafson-Kessel algorithm will be six. The results of the validation measures for the Gustafson-Kessel algorithm are written in table 4.2. This process can be repeated for all other cluster algorithms. The results can be found in Appendix B. For the K-means, K-medoid and the Gath-Geva, the optimal number of clusters is chosen at $c = 4$. For the other algorithms, the optimal number of clusters is located at $c = 6$. 

Figure 4.3: Values of Partition coefficient and Classification Entropy with Gustafson-Kessel clustering
Figure 4.4: Values of Partition Index, Separation Index and the Xie Beni Index with Gustafson-Kessel clustering

Figure 4.5: Values of Dunn’s Index and Alternative Dunn Index with Gustafson-Kessel clustering
4.2 Comparing the clustering algorithms

The optimal number of cluster can be determined with the validation methods, as mentioned in the previous section. The validation measures can also be used to compare the different cluster methods. As examined in the previous section, the optimal number of clusters was found at $c = 4$ or $c = 6$, depending on the clustering algorithm. The validation measures for $c = 4$ and $c = 6$ of all the clustering methods are collected in the tables 4.3 and 4.4. Table 4.3

<table>
<thead>
<tr>
<th></th>
<th>PC</th>
<th>CE</th>
<th>PI</th>
<th>SI</th>
<th>XBI</th>
<th>DI</th>
<th>ADI</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-means</td>
<td>1</td>
<td>NaN</td>
<td>1.1571</td>
<td>0.0002</td>
<td>5.0034</td>
<td>0.0034</td>
<td>0.0002</td>
</tr>
<tr>
<td>K-medoid</td>
<td>1</td>
<td>NaN</td>
<td>0.2366</td>
<td>0.0001</td>
<td>Inf</td>
<td>0.0084</td>
<td>0.0002</td>
</tr>
<tr>
<td>FCM</td>
<td>0.2800</td>
<td>1.3863</td>
<td>0.0002</td>
<td>42.2737</td>
<td>1.0867</td>
<td>0.0102</td>
<td>0.0063</td>
</tr>
<tr>
<td>GK</td>
<td>0.3983</td>
<td>1.0009</td>
<td>1.5930</td>
<td>0.0007</td>
<td>1.4183</td>
<td>0.0083</td>
<td>0.0039</td>
</tr>
<tr>
<td>GG</td>
<td>0.4982</td>
<td>1.5034</td>
<td>0.0001</td>
<td>0.0001</td>
<td>1.0644</td>
<td>0.0029</td>
<td>0.0030</td>
</tr>
</tbody>
</table>

Table 4.3: The numerical values of validation measures for $c = 4$

and 4.4 show that the PC and CE are useless for the hard clustering methods K-means and K-medoid. On the score of the values of the three most used indexes, Separation index, Xie and Beni’s index and Dunn’s index, one can conclude that for $c = 4$ the Gath-Geva algorithm has the best results and for $c = 6$ the Gustafson-Kessel algorithm. To visualize the clustering results, the validation methods that are described in Section 3.4 can be used. With these visualization methods, the dataset can be reduced to a 2-dimensional space. To avoid visibility problems (plotting too much values will cause one
Table 4.4: The numerical values of validation measures for $c = 6$

<table>
<thead>
<tr>
<th></th>
<th>PC</th>
<th>CE</th>
<th>PI</th>
<th>SI</th>
<th>XBI</th>
<th>DI</th>
<th>ADI</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-means</td>
<td>1</td>
<td>NaN</td>
<td>1.2907</td>
<td>0.0002</td>
<td>3.9253</td>
<td>0.0063</td>
<td>0.0001</td>
</tr>
<tr>
<td>K-medoid</td>
<td>1</td>
<td>NaN</td>
<td>0.1238</td>
<td>0.0001</td>
<td>Inf</td>
<td>0.0070</td>
<td>0.0008</td>
</tr>
<tr>
<td>FCM</td>
<td>0.1667</td>
<td>1.7918</td>
<td>0.0001</td>
<td>19.4613</td>
<td>0.9245</td>
<td>0.0102</td>
<td>0.0008</td>
</tr>
<tr>
<td>GK</td>
<td>0.3044</td>
<td>1.4293</td>
<td>0.8903</td>
<td>0.0001</td>
<td>0.9203</td>
<td>0.0029</td>
<td>0.0007</td>
</tr>
<tr>
<td>GG</td>
<td>0.3773</td>
<td>1.6490</td>
<td>0.1043</td>
<td>0.0008</td>
<td>1.0457</td>
<td>0.0099</td>
<td>0.0009</td>
</tr>
</tbody>
</table>

big cloud of data points), only 500 values (representing 500 customers) from this 2-dimensional dataset will be randomly picked. For the K-means and the K-medoid algorithm, the Sammon’s mapping gives the best visualization of the results. For the other cluster algorithms, the Fuzzy Sammon’s mapping visualization gives the best projection with respect to the partitions of the data set. These visualization methods are used for the following plots. Figures 4.x-4.x show the different clustering results for $c = 4$ and $c = 6$ on the data set.

Figure 4.6 and 4.7 show that hard clustering methods can find a solution for the clustering problem. None of the clusters contain sufficient more or less customers than other clusters. The plot of the Fuzzy C-means algorithm, in Figure 4.8, shows unexpected results. For the situation with 4 clusters, there are only 2 clusters clearly visible. By a detailed look at the plot, one can see that there are actually 4 cluster centers, but the cluster centers are almost situated on the same location. In the situation with 6 clusters, one can see three big cluster, with one small cluster in one of the big clusters. The other two cluster centers are nearly invisible. This implies that the Fuzzy C-means algorithm is not able to find good clusters for this data set. In Figure 4.9, the results of the Gustafon-Kessel algorithm are plotted. For both situations, the clusters are well separated. Note that the cluster in the left bottom corner and the cluster in the
Figure 4.7: Result of K-medoid algorithm

Figure 4.8: Result of Fuzzy C-means algorithm

Figure 4.9: Result of Gustafson-Kessel algorithm
top right corner in Figure 4.9 are also maintained in the situation with 6 clusters. This may indicate that the data points in these clusters represents customers that differ on multiple fields with the other customers of Vodafone. The results of the Gath-Geva algorithm, visualized in Figure 4.10, for the situation $c = 4$ look similar to the result of the Gustafson-Kessel algorithm. The result for the $c = 6$ situation is remarkable. Here are also appearing clusters in other clusters. In the real high-dimensional situation, the clusters are not a subset of each other, but are separated. The fact that this is the case in the two-dimensional plot, indicates that a clustering with six clusters with the Gustafson-Kessel algorithms not a good solution. With the results of the validation methods and the visualization of the clustering, one can conclude that there are two possible best solutions: The Gath-Geva algorithm for $c = 4$ and the Gustafson-Kessel algorithm for $c = 6$. To determine which partitioning will be used to define the segments, a closer look to the meaning of the clusters will be needed. In the next section, the two different partitions will be closely compared with each other.

4.3 Designing the segments

To define which clustering method will be used for the segmentation, one can look at the distances from the points to each cluster. In Figure 4.11 and 4.12, two box plots of the distances from the data points to the cluster are plotted. The box indicates the upper and lower quartiles. In both situations, the results show that the clusters are homogeneous. This indicates that, based on the distances to the cluster, one can not distinguish between the two cluster algorithms. Another way to view the differences between the cluster methods is to profile the clusters. For each cluster, a profile can be made by drawing a line between all normalized feature values (each feature value is represented at the x-as) of the customers within this cluster. The result is visible for the Gath-Geva algorithm for $c = 4$ and for the Gustafson-Kessel algorithm with six clusters.
Figure 4.11: Distribution of distances from cluster centers within clusters for the Gath-Geva algorithm with \( c = 4 \)

Figure 4.12: Distribution of distances from cluster centers within clusters for the Gustafson-Kessel algorithm with \( c = 6 \)
The profiles of the different clusters do not differ much in shape. However, in each cluster, at least one value differs sufficient from the values of the other cluster. This confirms the assumption that customers of different clusters have indeed a different usage behavior. Most of the lines in one profile are drawn closely together. This means that the customers in one profile contain similar values of the feature values.

Figure 4.13: Cluster profiles for $c = 4$
Figure 4.14: Cluster profiles for $c = 6$
More relevant plots are shown in Figure 4.15 and ???. The mean of all the lines (equivalent to the cluster center) was calculated and a line between all the (normalized) feature values was drawn. The difference between the clusters are visible by some feature values. For instance, in the situation with four clusters, Cluster 1 has customers, compared with other cluster, have a high feature value at feature 8. Cluster 2 has high values at position 6 and 9, while Cluster 3 contains peaks at features 2 and 12. The 4th and final cluster has high values at feature 8 and 9.

Figure 4.15: Cluster profiles of centers for $c = 4$
Figure 4.16: Cluster profiles of centers for $c = 6$
With the previous clustering results, validation measures and plots, it is not possible to decide which of the two clustering methods gives a better result. Therefore, both results will be used as a solution for the customer segmentation. For the Gath-Geva algorithm with \(c = 4\) and the Gustafson-Kessel algorithm with \(c = 6\), table 4.5 shows the result of the customer segmentation. The feature numbers correspond to the feature numbers of Section 2.1.2. (Feature 1 is the call duration, feature 2 the received voices calls and feature 3 the originated calls, feature 4 the daytime calls, feature 5 the weekday calls, 6 are calls to mobile phones, 7 received sms, 8 originated sms, feature 9 the international calls, feature 10 the calls to Vodafone mobiles, 11 the unique are codes and feature 12 the number of different numbers called). In words, the segments can be described as follows: For the situation with 4 segments:

- **Segment 1**: In this segment are customers with a relative low number of voice calls. This customers call more in the evening (in proportion) and to fixed lines then other customers. Their sms usage is higher then normal. The number of international calls is low.

- **Segment 2**: This segment contains customers with an average voice call

<table>
<thead>
<tr>
<th>Feature</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Average</strong></td>
<td>119.5</td>
<td>1.7</td>
<td>3.9</td>
<td>65.8</td>
<td>87.0</td>
</tr>
<tr>
<td>(c = 4)</td>
<td>Segment 1 (27.2%)</td>
<td>91.3</td>
<td>0.9</td>
<td>2.9</td>
<td>54.8</td>
<td>86.6</td>
</tr>
<tr>
<td></td>
<td>Segment 2 (28.7%)</td>
<td>120.1</td>
<td>1.8</td>
<td>3.6</td>
<td>73.6</td>
<td>87.1</td>
</tr>
<tr>
<td></td>
<td>Segment 3 (23.9%)</td>
<td>132.8</td>
<td>2.4</td>
<td>4.4</td>
<td>60.1</td>
<td>86.7</td>
</tr>
<tr>
<td></td>
<td>Segment 4 (20.2%)</td>
<td>133.8</td>
<td>1.7</td>
<td>4.7</td>
<td>74.7</td>
<td>87.6</td>
</tr>
<tr>
<td>(c = 6)</td>
<td>Segment 1 (18.1%)</td>
<td>94.7</td>
<td>1.2</td>
<td>2.8</td>
<td>66.3</td>
<td>88.0</td>
</tr>
<tr>
<td></td>
<td>Segment 2 (14.4%)</td>
<td>121.8</td>
<td>1.7</td>
<td>4.1</td>
<td>65.9</td>
<td>86.4</td>
</tr>
<tr>
<td></td>
<td>Segment 3 (18.3%)</td>
<td>121.6</td>
<td>2.5</td>
<td>4.9</td>
<td>66.0</td>
<td>84.3</td>
</tr>
<tr>
<td></td>
<td>Segment 4 (17.6%)</td>
<td>126.8</td>
<td>1.6</td>
<td>4.0</td>
<td>65.7</td>
<td>87.3</td>
</tr>
<tr>
<td></td>
<td>Segment 5 (14.8%)</td>
<td>96.8</td>
<td>1.1</td>
<td>3.5</td>
<td>65.2</td>
<td>88.6</td>
</tr>
<tr>
<td></td>
<td>Segment 6 (16.8%)</td>
<td>155.3</td>
<td>2.1</td>
<td>4.1</td>
<td>65.7</td>
<td>87.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Feature</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Average</strong></td>
<td>1.6</td>
<td>3.7</td>
<td>2.2</td>
<td>14.4</td>
<td>6.9</td>
</tr>
<tr>
<td>(c = 4)</td>
<td>Segment 1 (27.2%)</td>
<td>1.7</td>
<td>4.0</td>
<td>1.6</td>
<td>12.3</td>
<td>6.2</td>
</tr>
<tr>
<td></td>
<td>Segment 2 (28.7%)</td>
<td>1.2</td>
<td>3.1</td>
<td>2.1</td>
<td>12.8</td>
<td>6.6</td>
</tr>
<tr>
<td></td>
<td>Segment 3 (23.9%)</td>
<td>1.4</td>
<td>3.4</td>
<td>2.1</td>
<td>22.4</td>
<td>9.4</td>
</tr>
<tr>
<td></td>
<td>Segment 4 (20.2%)</td>
<td>2.1</td>
<td>4.3</td>
<td>3.0</td>
<td>10.1</td>
<td>5.4</td>
</tr>
<tr>
<td>(c = 6)</td>
<td>Segment 1 (18.1%)</td>
<td>2.3</td>
<td>4.5</td>
<td>1.8</td>
<td>11.3</td>
<td>6.1</td>
</tr>
<tr>
<td></td>
<td>Segment 2 (14.4%)</td>
<td>1.6</td>
<td>3.7</td>
<td>1.9</td>
<td>17.8</td>
<td>9.5</td>
</tr>
<tr>
<td></td>
<td>Segment 3 (18.3%)</td>
<td>1.0</td>
<td>2.9</td>
<td>2.9</td>
<td>15.1</td>
<td>6.6</td>
</tr>
<tr>
<td></td>
<td>Segment 4 (17.6%)</td>
<td>1.5</td>
<td>3.6</td>
<td>1.9</td>
<td>15.0</td>
<td>6.2</td>
</tr>
<tr>
<td></td>
<td>Segment 5 (14.8%)</td>
<td>0.8</td>
<td>2.9</td>
<td>1.8</td>
<td>12.4</td>
<td>6.1</td>
</tr>
<tr>
<td></td>
<td>Segment 6 (16.8%)</td>
<td>2.4</td>
<td>4.6</td>
<td>2.9</td>
<td>14.8</td>
<td>6.9</td>
</tr>
</tbody>
</table>
usage. They call often to mobile phones during day time. They do not send and receive many sms messages.

- **Segment 3**: The customers in this segment make relative many voice calls. These customers call to many different numbers and have a lot of contacts which are Vodafone customers.

- **Segment 4**: These customers originate many voice calls. They also send and receive many sms messages. They call often during daytime and call more then average to international numbers. Their call duration is high. Remarkable is the fact that they don not have so many contacts as the number of calls do suspect. They have a relative small number of contacts.

For the situation with 6 segments, the customers in this segments can be described as follows:

- **Segment 1**: In this segment are customers with a relative low number of voice calls. Their average call duration is also lower than average. However, their sms usage is relative high. These customers do not call to many different numbers.

- **Segment 2**: This segment contains customers with a relative high number of contacts. They also call to many different areas. They have also more contacts with a Vodafone mobile.

- **Segment 3**: The customers in this segment make relative many voice calls. Their sms usage is low. In proportion, they make more international phone calls than other customers.

- **Segment 4**: These customers are the average customers. None of the feature values is high or low.

- **Segment 5**: These customers do not receive many voice calls. The average call duration is low. They also receive and originate a low number of sms messages.

- **Segment 6**: These customers originate and receive many voice calls. They also send and receive many sms messages. The duration of their voice calls is longer than average. The percentage of international calls is high.

In the next session the classification method Support Vector Machine will be explained. This technique will be used to classify/estimate the segment of a customer by personal information as age, gender and lifestyle (the customer data of Section 2.1.3).
Chapter 5

Support Vector Machines

A Support Vector Machine is a algorithm that learns by example to assign labels to objects [16]. In this research a Support Vector machine will be used to recognize the segment of a customer by examining thousands of customers (e.g. the customer data features of Section 2.1.3) of each segment. In general, a Support Vector Machine is a mathematical entity; an algorithm for maximizing a particular mathematical function with respect to a given collection of data. However, the basic ideas of Support Vector Machines can be explained without any equations. The next few sections will describe the four basic concepts:

- The separating hyper plane
- The maximum-margin hyperplane
- The soft margin
- The kernel function

For now, to allow an easy, geometric interpretation of the data, imagine that there exists only two segments. In this case the customer data consist of 2 feature values, age and income, which can be easily plotted. The green dots represent the customers that are in segment 1 and the red dots are customers that are in segment 2. The goal of the SVM is learn to tell the difference between the groups and, given an unlabeled customer, such as the one labeled 'Unknown' in Figure 5.1, predict whether it corresponds to segment 1 or segment 2.

5.1 The separating hyperplane

A human being is very good at pattern recognition. Even a quick glance at Figure 5.1a shows that the green dots form a group and the red dots form another group that can easily be separated by drawing a line between the two groups (Figure 5.1b). Subsequently, predicting the label of an unknown customer is simple: one simply needs to ask whether the new customer falls on the segment
1 or the segment 2 side of the separating line. Now, to define the notion of a separating hyperplane, consider the situation where there are not just two feature values to describe the customer. For example, if there was just 1 feature value to describe the customer, then the space in which the corresponding one-dimensional feature resides is a one-dimensional line. This line can be divided in half by using a single point (see Figure 5.2a). In two dimensions, a straight line divides the space in half (remember Figure 5.1b). In a three-dimensional space, a plane is needed to divide the space, illustrated in Figure 5.2b. This procedure can be extrapolated mathematically in higher dimensions. The term for a straight line in a high-dimensional space is a hyperplane. So the term separating hyperplane is, essentially, the line that separates the segments.
5.2 The maximum-margin hyperplane

The concept of treating objects as points in a high-dimensional space and finding a line that separates them, is a common way of classification, and therefore not unique to the SVM. However, the SVM differs from all other classifier methods by virtue of how the hyperplane should be selected. Consider again the classification problem of Figure 5.1a. The goal of SVM is to find a line that separates the segment 1 customers from the segment 2 customers. However, there are an infinite number of possible lines, as portrayed in Figure 5.2. The question is which line should be chosen as the optimal classifier and how should the optimal line be defined. A logical way of selecting the optimal line, is selecting the line that is, roughly speaking, ‘in the middle’. In other words, the line that separates the two segments and adopts the maximal distance from any of the given customers (see Figure 5.2). It is not surprising that a theorem of the statistical learning theory is supporting this choice [6]. By defining the distance from the hyperplane to the nearest customer (in general an expression vector) as the margin of the hyperplane, the SVM selects the maximum separating hyperplane. By selecting this hyperplane, the SVM is able to predict the unknown segment of the customer in Figure 5.1a. The vectors (points) that constrain the width of the margin are the support vectors. This theorem, is in many ways, the key

![Figure 5.3: Demonstration of the maximum-margin hyperplane](image)

(a) Many possibilities  
(b) The maximum-margin hyperplane

Figure 5.3: Demonstration of the maximum-margin hyperplane

to the success of Support Vector Machines. However, there are a some remarks and caveats to deal with. First at all, the theorem is based on the fact that the data on which the SVM is trained are drawn from the same distribution as the data it has to classify. This is of course logical, since it is not reasonable that a Support Vector machine trained on customer data is able to classify different car types. More relevantly, it is not reasonable to expect that the SVM can classify well if the training data set is prepared with a different protocol then the test data set. On the other hand, the theorem of a SVM indicates that the two data sets has to be drawn from the same distribution. For example, a SVM
does not assume that the data is drawn from a normal distribution.

5.3 The soft margin

So far, the theory assumed that the data can be separated by a straight line. However, many real data sets are not cleanly separable by a straight line, for example the data of Figure 5.4a. In this figure, the data contains an error object. An intuitively way to deal with the problems of errors is designing the SVM in such a way that it allows a few anomalous customers to fall on the 'wrong side' of separation line. This can be achieved by adding a 'soft margin' to the SVM. The soft margin allows a small percentage of the data points to push their way through the margin of the separating hyperplane without affecting the final result. With the soft margin, the data set of Figure 5.4a will be separated in the way it is illustrated in Figure 5.3 The customer can be seen as an outlier and resides on the same side of the line with customers of segment 1. Of course, a SVM should not allow too many misclassification. Note, that with the introduction of the soft margin, a user-specified parameter is involved that controls the soft margin and, roughly, controls the number of customers that is allowed to violate the separation line and determines how far across the line they are allowed. Setting this parameter is a complicated process, by the fact that a large margin will be achieved with respect to the number of correct classifications. In other words, the soft margin specifies a trade-off between hyper plane violations and the size of the margin.

5.4 The kernel functions

To understand the notion of a kernel function, the example data will be simplified even further. Assume that, instead of a two-dimensional data set, there
is a one-dimensional data set, as seen before in Figure 5.1. In that case, the separating hyperplane was a single point. Now, consider the situation of Figure 5.4, which illustrates an non separable data set. No single point can separate the two segments and introducing a soft margin would not help. A kernel function provides a solution to this problem. The kernel function adds an extra dimension to the data, in this case by squaring the one dimensional data set. The result is plotted in Figure 5.4. Within the new higher dimensional space, as shown in the figure, the SVM can separate the data in two segments by one straight line. In general, the kernel function can be seen as a mathematical trick for the SVM to project data from a low-dimensional space to a space of higher dimensions. If one chooses a good kernel function, the data will become separable in the corresponding higher dimension. To understand kernels better, some extra examples will be given. In Figure 5.4 is plotted a two-dimensional data set. With a relative simple kernel function, this data can be projected to a four-dimensional space. It is not possible to draw the data in the 4 dimensional space, but with a projection of the SVM hyperplane in the four-dimensional space back down to the original two-dimensional space, the result is shown as the curved line in Figure 5.4. it is possible to prove that for any data set exists a kernel function that allows the SVM to separate the data linearly in a higher dimension. Of course, the data set must contain consistent labels, which means that two identical data points may not have different labels. So, in theory, the SVM should be a perfect classifier. However, there are some drawbacks of projecting data in a very high-dimensional space to find the separating hyperplane. the first problem is the so called curse of dimensionality: as the numbers of variables under consideration increases, the number of possible solutions also increases, but exponentially. Consequently, it becomes harder for any algorithm to find a correct solution. In Figure 5.4 the situation is drawn when the data is project into a space with too many dimensions. The figure contains the same data as Figure 5.4, but the projected hyperplane is found by a very high dimen-

Figure 5.5: Demonstration of kernels
sional kernel. This results in boundaries which are too specific to the examples of the data set. This phenomenon is called over fitting. The SVM will not function well on new unseen unlabeled data. There exists another large practi-

\[ K(x_i, x_j) = \Phi(x_i)^T \Phi(x_j), \quad (5.1) \]

where \( x_i \) are the training vectors. The vectors are mapped into a higher dimensional space by the function \( \Phi \). Many kernel mapping functions can be used, probably an infinite number, but a few kernel functions have been found to work well in for a wide variety of applications [16]. The default and recommended kernel functions were used during this research and will be discussed now.

- **Linear**: which function is defined by:
  \[ K(x_i, x_j) = x_i^T x_j. \quad (5.2) \]

- **Polynomial**: the polynomial kernel of degree \( d \) is of the form
  \[ K(x_i, x_j) = (\gamma x_i^T x_j + c_0)^d. \quad (5.3) \]
• **Radial basis function**: also known as the Gaussian kernel is of the form

\[ K(x_i, x_j) = \exp(-\gamma||x_i - x_j||^2). \]  \hspace{1cm} (5.4)

• **Sigmoid**: the sigmoid function, which is also used in neural networks, is defined by

\[ K(x_i, x_j) = \tanh(\gamma x_i^T x_j + c_0). \]  \hspace{1cm} (5.5)

When the sigmoid function is used, one can regard it with a as a two-layer neural network.

In this research the constant \( c_0 \) is set to 1. The concept of a kernel mapping function is very powerful. It allows a SVM to perform separations even with very complex boundaries as shown in Figure 5.7.

![Figure 5.7: A separation of classes with complex boundaries](image)

5.5 **Multi class classification**

So far, the idea of using a hyperplane to separate the feature vectors into two groups was described, but only for two target categories. How does a SVM discriminate between a large variety of classes, as in our case 4 or 6 segments? There are several approaches proposed, but two methods are the most popular and most used [16]. The first approach is to train multiple, one-versus-all classifiers. For example, if the SVM has to recognize three classes, A, B and C, one can simply train three separate SVM to answer the binary questions, "Is it A?", "Is it B?" and "Is it C?". Another simple approach is the one-versus-one where \( k(k-1)/2 \) models are constructed, where \( k \) is the number of classes. In this research the one-versus-one technique will be used.
Chapter 6

Experiments and results of classifying the customer segments

6.1 K-fold cross validation

To avoid over fitting, cross-validation is used to evaluate the fitting provided by each parameter value set tried during the experiments. Figure 6.1 demonstrates how important the training process is. Different parameter values may cause under or over fitting. By K-fold cross validation the training dataset will be divided into two groups, the training set, the test set and the validation set. The training set will be used to train the SVM. The test set will be used to estimate the error during the training of the SVM. With the validation set, the actual performance of the SVM will be measured after the SVM is trained. The training of the SVM will be stopped when the test error reached a local

[Figure 6.1: Under fitting and over fitting]
minimum, see Figure 6.2. By K-fold cross validation, a k-fold partition of the
data set is created. For each of K experiments, K-1 folds will be used for training and the remaining one for testing. Figure 6.3 illustrates this process. In this research, K is set to 10. The advantage of K-fold cross validation is that all the examples in the dataset are eventually used for both training and testing. The error is calculated by taking the average off all K experiments.

### 6.2 Parameter setting

In this section, the optimal parameters for the Support Vector Machine will be researched and examined. Each kernel function with its parameters will be tested on their performance. The linear Kernel function itself has no parameters. The only parameter that can be researched is the soft margin value of the Support Vector Machine, denoted by $C$. In table 6.1 and table 6.2 the results for the different C-values are summarized. For the situation with 4 clusters, the

<table>
<thead>
<tr>
<th>C</th>
<th>1</th>
<th>2</th>
<th>5</th>
<th>10</th>
<th>20</th>
<th>50</th>
<th>100</th>
<th>200</th>
<th>500</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>42.1%</td>
<td>42.6%</td>
<td>43.0%</td>
<td><strong>43.2%</strong></td>
<td>43.0%</td>
<td>42.4%</td>
<td>41.7%</td>
<td>40.8%</td>
<td>36.1%</td>
</tr>
</tbody>
</table>

Table 6.1: Linear Kernel, 4 segments
optimal value for the soft margin is $C = 10$ and by using the 6 segments $C = 50$. The correct number of classifications are respectively, 43.2% and 32.0%. For the polynomial kernel function, there are two parameters. The number of degrees, denoted by $d$ and the width $\gamma$. Therefore, the optimal number for the maximal margin will be determined. This is done by multiple test runs with random values for $d$ and $\gamma$. The average value for each soft margin $C$ can be found in the tables 6.3 and 6.4. These $C$-values are used to find out which $d$ and $\gamma$ give the best results. The results are shown in tables 6.5 and 6.6. For the situation with

<table>
<thead>
<tr>
<th>$C$</th>
<th>1</th>
<th>2</th>
<th>5</th>
<th>10</th>
<th>20</th>
<th>50</th>
<th>100</th>
<th>200</th>
<th>500</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>28.9%</td>
<td>29.4%</td>
<td>30.9%</td>
<td>31.3%</td>
<td>31.4%</td>
<td>32.0%</td>
<td>27.6%</td>
<td>27.6%</td>
<td>21.8%</td>
</tr>
</tbody>
</table>

Table 6.2: Linear Kernel, 6 segments

<table>
<thead>
<tr>
<th>$C$</th>
<th>1</th>
<th>2</th>
<th>5</th>
<th>10</th>
<th>20</th>
<th>50</th>
<th>100</th>
<th>200</th>
<th>500</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>27.6%</td>
<td>27.6%</td>
<td>21.8%</td>
<td>32.0%</td>
<td>31.4%</td>
<td>31.3%</td>
<td>30.9%</td>
<td>29.4%</td>
<td>28.9%</td>
</tr>
</tbody>
</table>

Table 6.3: Average C-value for polynomial kernel, 4 segments

<table>
<thead>
<tr>
<th>$C$</th>
<th>1</th>
<th>2</th>
<th>5</th>
<th>10</th>
<th>20</th>
<th>50</th>
<th>100</th>
<th>200</th>
<th>500</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>73.8%</td>
<td>77.4%</td>
<td>76.6%</td>
<td>74.6%</td>
<td>73.5%</td>
<td>72.8%</td>
<td>70.6%</td>
<td>63.2%</td>
<td>53.7%</td>
</tr>
</tbody>
</table>

Table 6.4: Average C-value for polynomial kernel, 6 segments

<table>
<thead>
<tr>
<th>$d$</th>
<th>$\gamma = 0.4$</th>
<th>$\gamma = 0.6$</th>
<th>$\gamma = 0.8$</th>
<th>$\gamma = 1.0$</th>
<th>$\gamma = 1.2$</th>
<th>$\gamma = 1.4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>76.1% 76.3% 75.9%</td>
<td>76.0% 76.3% 77.6%</td>
<td>75.8% 76.3% 77.2%</td>
<td>76.2% 76.4% 78.0%</td>
<td>76.0% 76.2% 78.1%</td>
<td>75.2% 76.2% 78.1%</td>
</tr>
<tr>
<td>2</td>
<td>73.2% 74.8% 76.0%</td>
<td>74.1% 74.5% 75.4%</td>
<td>74.0% 74.4% 77.1%</td>
<td>75.0% 75.2% 75.6%</td>
<td>74.6% 75.1% 76.0%</td>
<td>74.9% 75.5% 76.3%</td>
</tr>
<tr>
<td>3</td>
<td>74.8% 76.0% 75.0%</td>
<td>74.5% 75.4% 75.8%</td>
<td>74.4% 77.1% 75.2%</td>
<td>75.2% 75.6% 75.8%</td>
<td>77.1% 75.1% 76.0%</td>
<td>74.9% 75.5% 76.3%</td>
</tr>
<tr>
<td>4</td>
<td>76.0% 76.2% 78.1%</td>
<td>75.4% 75.8% 75.0%</td>
<td>75.0% 75.2% 75.6%</td>
<td>75.6% 75.8% 75.8%</td>
<td>76.0% 76.2% 78.1%</td>
<td>75.5% 76.3% 74.9%</td>
</tr>
<tr>
<td>5</td>
<td>74.3% 75.3% 76.2%</td>
<td>74.8% 75.0% 75.0%</td>
<td>74.4% 77.1% 75.2%</td>
<td>75.2% 75.6% 75.8%</td>
<td>74.6% 75.1% 76.0%</td>
<td>74.9% 75.5% 76.3%</td>
</tr>
<tr>
<td>6</td>
<td>76.0% 76.2% 78.1%</td>
<td>75.4% 75.8% 75.0%</td>
<td>75.0% 75.2% 75.6%</td>
<td>75.6% 75.8% 75.8%</td>
<td>76.0% 76.2% 78.1%</td>
<td>75.5% 76.3% 74.9%</td>
</tr>
<tr>
<td>7</td>
<td>75.0% 75.0% 76.0%</td>
<td>74.3% 74.3% 74.3%</td>
<td>75.8% 75.8% 75.8%</td>
<td>75.2% 75.6% 75.8%</td>
<td>74.6% 75.1% 76.0%</td>
<td>74.9% 75.5% 76.3%</td>
</tr>
</tbody>
</table>

Table 6.5: Polynomial kernel, 4 segments

<table>
<thead>
<tr>
<th>$d$</th>
<th>$\gamma = 0.4$</th>
<th>$\gamma = 0.6$</th>
<th>$\gamma = 0.8$</th>
<th>$\gamma = 1.0$</th>
<th>$\gamma = 1.2$</th>
<th>$\gamma = 1.4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>75.0% 74.6% 75.9%</td>
<td>74.2% 75.1% 74.9%</td>
<td>73.8% 74.7% 74.3%</td>
<td>74.1% 75.0% 73.6%</td>
<td>72.1% 74.1% 75.5%</td>
<td>73.6% 74.3% 72.2%</td>
</tr>
<tr>
<td>2</td>
<td>76.0% 76.2% 76.0%</td>
<td>75.4% 75.4% 75.4%</td>
<td>75.0% 75.4% 75.4%</td>
<td>75.0% 75.4% 75.4%</td>
<td>74.1% 74.1% 75.5%</td>
<td>76.0% 76.2% 76.0%</td>
</tr>
<tr>
<td>3</td>
<td>76.3% 76.9% 76.0%</td>
<td>76.2% 76.2% 76.2%</td>
<td>76.0% 76.0% 76.0%</td>
<td>75.9% 75.9% 75.9%</td>
<td>75.4% 75.4% 75.4%</td>
<td>74.4% 74.4% 74.4%</td>
</tr>
<tr>
<td>4</td>
<td>75.8% 75.8% 76.0%</td>
<td>75.4% 75.4% 75.4%</td>
<td>75.0% 75.0% 75.0%</td>
<td>75.0% 75.0% 75.0%</td>
<td>75.4% 75.4% 75.4%</td>
<td>74.3% 74.3% 74.3%</td>
</tr>
<tr>
<td>5</td>
<td>74.3% 74.3% 74.3%</td>
<td>74.3% 74.3% 74.3%</td>
<td>74.3% 74.3% 74.3%</td>
<td>74.3% 74.3% 74.3%</td>
<td>74.3% 74.3% 74.3%</td>
<td>72.9% 72.9% 72.9%</td>
</tr>
<tr>
<td>6</td>
<td>75.0% 75.0% 75.0%</td>
<td>74.6% 74.6% 74.6%</td>
<td>75.8% 75.8% 75.8%</td>
<td>76.2% 76.2% 76.2%</td>
<td>75.4% 75.4% 75.4%</td>
<td>74.3% 74.3% 74.3%</td>
</tr>
<tr>
<td>7</td>
<td>76.0% 76.0% 76.0%</td>
<td>75.4% 75.4% 75.4%</td>
<td>76.3% 76.3% 76.3%</td>
<td>76.9% 76.9% 76.9%</td>
<td>75.4% 75.4% 75.4%</td>
<td>74.3% 74.3% 74.3%</td>
</tr>
</tbody>
</table>

Table 6.6: Polynomial kernel, 6 segments

62
4 segments, the optimal score is 78.1% and for 6 segments 76.2%. The following kernel function, the radial basis function has only one variable, namely $\gamma$. The results of the Radial Basis function are given in table 6.7 and table 6.8. The best result with 4 segments is 80.3%, with 6 segments the best score is 78.5%. Remarkable is the fact that the difference is small between the two situations, while there are two extra clusters. The confusion matrix for both situations, table 6.11 and 6.12, show that there are two clusters which can easily be classified with the customer profile. This corresponds to the cluster in the top right corner and the cluster in the bottom of Figures 4.9 and 4.10.

<table>
<thead>
<tr>
<th>C</th>
<th>1</th>
<th>2</th>
<th>5</th>
<th>10</th>
<th>20</th>
<th>50</th>
<th>100</th>
<th>200</th>
<th>500</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\gamma = 0.4$</td>
<td>80.0</td>
<td>79.0</td>
<td>76.6</td>
<td>78.3</td>
<td>76.4</td>
<td>73.3</td>
<td>60.2</td>
<td>52.4</td>
<td>37.5</td>
</tr>
<tr>
<td>$\gamma = 0.6$</td>
<td>80.1</td>
<td>80.3</td>
<td>77.7</td>
<td>79.0</td>
<td>79.9</td>
<td>72.8</td>
<td>63.6</td>
<td>59.6</td>
<td>27.5</td>
</tr>
<tr>
<td>$\gamma = 0.8$</td>
<td>79.3</td>
<td>79.5</td>
<td>78.2</td>
<td>80.2</td>
<td>78.4</td>
<td>69.3</td>
<td>59.3</td>
<td>51.4</td>
<td>29.7</td>
</tr>
<tr>
<td>$\gamma = 1.0$</td>
<td>78.4</td>
<td>78.2</td>
<td>80.3</td>
<td>78.5</td>
<td>76.9</td>
<td>66.2</td>
<td>61.7</td>
<td>47.9</td>
<td>30.6</td>
</tr>
<tr>
<td>$\gamma = 1.2$</td>
<td>79.6</td>
<td>79.9</td>
<td>79.8</td>
<td>80.2</td>
<td>80.1</td>
<td>69.0</td>
<td>61.3</td>
<td>45.5</td>
<td>26.3</td>
</tr>
<tr>
<td>$\gamma = 1.4$</td>
<td>77.4</td>
<td>76.9</td>
<td>76.5</td>
<td>79.4</td>
<td>77.7</td>
<td>71.4</td>
<td>61.3</td>
<td>41.2</td>
<td>26.0</td>
</tr>
</tbody>
</table>

Table 6.7: Radial basis function, 4 segments

<table>
<thead>
<tr>
<th>C</th>
<th>1</th>
<th>2</th>
<th>5</th>
<th>10</th>
<th>20</th>
<th>50</th>
<th>100</th>
<th>200</th>
<th>500</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\gamma = 0.4$</td>
<td>73.6</td>
<td>77.4</td>
<td>72.6</td>
<td>70.9</td>
<td>68.0</td>
<td>65.1</td>
<td>52.7</td>
<td>51.8</td>
<td>40.0</td>
</tr>
<tr>
<td>$\gamma = 0.6$</td>
<td>72.5</td>
<td>74.8</td>
<td>74.8</td>
<td>72.7</td>
<td>73.0</td>
<td>70.4</td>
<td>54.0</td>
<td>49.3</td>
<td>39.1</td>
</tr>
<tr>
<td>$\gamma = 0.8$</td>
<td>74.1</td>
<td>76.6</td>
<td>80.3</td>
<td>80.0</td>
<td>68.4</td>
<td>60.5</td>
<td>55.5</td>
<td>54.1</td>
<td>40.9</td>
</tr>
<tr>
<td>$\gamma = 1.0$</td>
<td>70.7</td>
<td>72.9</td>
<td>73.8</td>
<td>70.9</td>
<td>66.1</td>
<td>64.7</td>
<td>52.2</td>
<td>48.5</td>
<td>34.2</td>
</tr>
<tr>
<td>$\gamma = 1.2$</td>
<td>72.6</td>
<td>73.5</td>
<td>73.4</td>
<td>73.1</td>
<td>71.9</td>
<td>74.6</td>
<td>64.8</td>
<td>60.0</td>
<td>38.3</td>
</tr>
<tr>
<td>$\gamma = 1.4$</td>
<td>69.4</td>
<td>68.5</td>
<td>70.7</td>
<td>69.1</td>
<td>68.0</td>
<td>68.5</td>
<td>54.4</td>
<td>52.4</td>
<td>31.0</td>
</tr>
</tbody>
</table>

Table 6.8: Radial basis function, 6 segments

correct, with respectively 4 and 6 segmentents, by the Sigmoid function. This means that the Radial basis function has the best score for both situations, with 80.3% and 78.5%. Remarkable is the fact that the difference is small between the two situations, while there are two extra clusters. The confusion matrix for both situations, table 6.11 and 6.12, show that there are two clusters which can easily be classified with the customer profile. This corresponds to the cluster in the top right corner and the cluster in the bottom of Figures 4.9 and 4.10.
Table 6.10: Sigmoid function, 6 segments

<table>
<thead>
<tr>
<th>C</th>
<th>1</th>
<th>2</th>
<th>5</th>
<th>10</th>
<th>20</th>
<th>50</th>
<th>100</th>
<th>200</th>
<th>500</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\gamma = 0.4$</td>
<td>33.8</td>
<td>34.0</td>
<td>34.7</td>
<td>33.1</td>
<td>34.6</td>
<td>30.0</td>
<td>32.6</td>
<td>28.8</td>
<td>28.8</td>
</tr>
<tr>
<td>$\gamma = 0.6$</td>
<td>29.6</td>
<td>27.4</td>
<td>28.5</td>
<td>29.7</td>
<td>21.4</td>
<td>20.8</td>
<td>20.0</td>
<td>18.8</td>
<td>18.1</td>
</tr>
<tr>
<td>$\gamma = 0.8$</td>
<td>39.1</td>
<td>36.4</td>
<td>33.6</td>
<td>35.7</td>
<td>38.9</td>
<td>32.0</td>
<td>26.4</td>
<td>24.6</td>
<td>22.9</td>
</tr>
<tr>
<td>$\gamma = 1.0$</td>
<td>40.0</td>
<td>42.5</td>
<td>39.8</td>
<td>40.7</td>
<td>39.9</td>
<td>39.8</td>
<td>30.4</td>
<td>31.1</td>
<td>28.0</td>
</tr>
<tr>
<td>$\gamma = 1.2$</td>
<td>41.9</td>
<td>40.6</td>
<td>43.6</td>
<td>43.2</td>
<td>44.1</td>
<td>43.2</td>
<td>44.6</td>
<td>40.6</td>
<td>41.7</td>
</tr>
<tr>
<td>$\gamma = 1.4$</td>
<td>38.6</td>
<td>34.5</td>
<td>32.1</td>
<td>30.6</td>
<td>30.2</td>
<td>27.5</td>
<td>24.3</td>
<td>26.3</td>
<td>27.9</td>
</tr>
</tbody>
</table>

Table 6.11: Confusion matrix, 4 segments

<table>
<thead>
<tr>
<th>Predicted →</th>
<th>Segment 1</th>
<th>Segment 2</th>
<th>Segment 3</th>
<th>Segment 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual ↓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Segment 1</td>
<td>97.1%</td>
<td>0.5%</td>
<td>1.9%</td>
<td>0.5%</td>
</tr>
<tr>
<td>Segment 2</td>
<td>3.6%</td>
<td>76.6%</td>
<td>7.8%</td>
<td>12.0%</td>
</tr>
<tr>
<td>Segment 3</td>
<td>2.2%</td>
<td>0.8%</td>
<td>96.3%</td>
<td>0.7%</td>
</tr>
<tr>
<td>Segment 4</td>
<td>7.1%</td>
<td>13.0%</td>
<td>6.9%</td>
<td>73.0%</td>
</tr>
</tbody>
</table>

Table 6.12: Confusion matrix, 6 segments

<table>
<thead>
<tr>
<th>Predicted →</th>
<th>Segm. 1</th>
<th>Segm. 2</th>
<th>Segm. 3</th>
<th>Segm. 4</th>
<th>Segm. 5</th>
<th>Segm. 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual ↓</td>
<td>Segm. 1</td>
<td>Segm. 2</td>
<td>Segm. 3</td>
<td>Segm. 4</td>
<td>Segm. 5</td>
<td>Segm. 6</td>
</tr>
<tr>
<td>Segment 1</td>
<td>74.1%</td>
<td>1.1%</td>
<td>10.1%</td>
<td>8.4%</td>
<td>0.6%</td>
<td>5.7%</td>
</tr>
<tr>
<td>Segment 2</td>
<td>0.2%</td>
<td>94.5%</td>
<td>0.6%</td>
<td>1.4%</td>
<td>1.2%</td>
<td>2.1%</td>
</tr>
<tr>
<td>Segment 3</td>
<td>5.6%</td>
<td>4.7%</td>
<td>71.2%</td>
<td>9.1%</td>
<td>2.1%</td>
<td>7.3%</td>
</tr>
<tr>
<td>Segment 4</td>
<td>12.3%</td>
<td>4.1%</td>
<td>3.9%</td>
<td>68.9%</td>
<td>6.8%</td>
<td>4.0%</td>
</tr>
<tr>
<td>Segment 5</td>
<td>2.0%</td>
<td>0.6%</td>
<td>0.7%</td>
<td>1.3%</td>
<td>92.6%</td>
<td>2.8%</td>
</tr>
<tr>
<td>Segment 6</td>
<td>12.5%</td>
<td>2.4%</td>
<td>3.7%</td>
<td>10.4%</td>
<td>1.3%</td>
<td>69.7%</td>
</tr>
</tbody>
</table>

64
6.3 Feature Validation

In this section, the features will be validated. The importance of each feature will be measured. This will be done, by leaving one feature out of the feature vector and train the SVM without this feature. The results of both situations, are shown in Figure 6.4 and 6.5. The result show that Age is an important feature for classifying the right segment. This is in contrast with the type of telephone, which increase the result with only tenths of percents. Each feature increases the result and therefore each feature is useful for the classification.

![Figure 6.4: Results while leaving out one of the features with 4 segments](image1)

![Figure 6.5: Results while leaving out one of the features with 6 segments](image2)
Chapter 7

Conclusions and discussion

This section concludes the research and the corresponding results and will give some recommendations for future work.

7.1 Conclusions

The first objective of our research was to perform automatic customer segmentation based on usage behavior, without the direct intervention of a human specialist. The second part of the research was focused on profiling customers and finding a relation between the profile and the segments. The customer segments were constructed by applying several clustering algorithms. The clustering algorithms used selected and preprocessed data from the Vodafone data warehouse. This led to solutions for the customer segmentation with respectively four segments and six segments. The customer’s profile was based on personal information of the customers. A novel data mining technique, called Support Vector Machines was used to estimate the segment of a customer based on his profile.

There are various ways for selecting suitable feature values for the clustering algorithms. This selection is vital for the resulting quality of the clustering. One different feature value will result in different segments. The result of the clustering can therefore not be regarded as universally valid, but merely as one possible outcome. In this research, the feature values were selected in such a way that it would describe the customer’s behavior as complete as possible. However, it is not possible to include all possible combinations of usage behavior characteristics within the scope of this research. To find the optimal number of clusters, the so-called elbow criterion was applied. Unfortunately, this criterion could not always be unambiguously identified. An other problem was that the location of the elbow could differ between the validation measures for the same algorithm. For some algorithms, the elbow was located at $c = 4$ and for other algorithms, the location was $c = 6$. To identify the best algorithm, several validation measures were used. Not every validation method marked the same
algorithm as the best algorithm. Therefore, some widely established validation measures were employed to determine the most optimal algorithm. It was however not possible to determine one algorithm that was optimal for \( c = 4 \) and \( c = 6 \). For the situation with four clusters, the Gath-Geva algorithm appears to be the best algorithm and the Gustafson-Kessel algorithm gives the best results by six clusters. To determine which customer segmentation algorithm is best suited for a particular data set and a specific parameter setting, the clustering results were interpreted in a profiling format. The results show, that in both situations the clusters were well separated and clearly distinguished from each other. It is hard to compare the two clustering results, because of the different number of clusters. Therefore, both clustering results were used as a starting point for the segmentation algorithm. The corresponding segments differ on features as number of voice calls, sms usage, call duration, international calls, different numbers called and percentage of weekday and daytime calls. A short characterization of each cluster was made.

A Support Vector Machine algorithm was used to classify the segment of a customer, based on the customer’s profile. The profile exists of the age, gender, telephone type, subscription type, company size, and residential area of the customer. As a comparison, four different kernel functions with different parameters were tested on their performance. It was found that the radial basis function gives the best result with a classification of 80.3% for the situation with four segments and 78.5% for the situation with six segments. It appeared that the resulting percentage of correctly classified segments was not as high as expected. A possible explanation could be that the features of the customer are not adequate for making a customer’s profile. This is caused by the frequently missing data in the Vodafone data warehouse about lifestyle, habits and income of the customers. A second reason for the low number of correct classification is the fact that the usage behavior in the database corresponds to a telephone number and this telephone number corresponds to a person. In real life, however, this telephone is maybe not used exclusively by the person (and the corresponding customer’s profile) as stored in the database. Customers may lend their telephone to relatives, and companies may exchange telephones among their employees. In such cases, the usage behavior does not correspond to a single customer’s profile and this impairs the classification process.

The last part of the research involves the relative importance of each individual feature of the customer’s profile. By leaving out one feature value during classification, the effect of each feature value became visible. It was found that without the concept of ‘customer age’, the resulting quality of the classification was significantly decreased. On the other hand, leaving out a feature such as the ‘telephone type’ barely decreased the classification result. However, this and some other features did well increase the performance of classification. This implies, that this feature bears some importance for the customer profiling and the classification of the customer’s segment.
7.2 Recommendations for future work

Based on our research and experiments, it is possible to formulate some recommendations for obtaining more suitable customer profiling and segmentation. The first recommendation is to use different feature values for the customer segmentation. This can lead to different clusters and thus different segments. To know the influence of the feature values on the outcome of the clustering, a complete data analysis research is required. Also, a detailed data analysis of the meaning of the cluster is recommended. In this research, the results are given by a short description of each segment. Extrapolating this approach, a more detailed view of the clusters and their boundaries can be obtained. Another way to validate the resulting clusters is to offer them to a human expert, and use his feedback for improving the clustering criteria.

To improve on determining the actual number of clusters present in the data set, the application of more specialized methods than the elbow criterion could be applied. An interesting alternative is, for instance, the application of evolutionary algorithms, as proposed by Wei Lu [21]. Another way of improving this research is to extend the number of cluster algorithms like main shift clustering, hierarchical clustering or mixture of Gaussians. To estimate the segment of the customer, also, other classification methods can be used. For instance, neural networks, genetic algorithms or Bayesian algorithms. Of specific interest is, within the framework of Support Vector Machines, cluster analysis of the application of miscellaneous (non-linear) kernel functions.

Furthermore, it should be noted that the most obvious and best way to improve the classification is to come to a more accurate and precise definition of the customer profiles. The customer profile used in this research is not sufficient detailed enough to describe the wide spectrum of customers. One reason for this is the missing data in the Vodafone data warehouse. Consequently, an enhanced and more precise analysis of the data warehouse will lead to improved features and, thus, to an improved classification.

Finally, we note that the study would improve noticeably by involving multiple criteria to evaluate the user behavior, rather than mere phone usage as employed here. Similarly, it is challenging to classify the profile of the customer based on the corresponding segment alone. However, this is a complex course and it essentially requires the availability of high-quality features.
Bibliography


Appendix A

Model of data warehouse

In this Appendix a simplified model of the data warehouse can be found. The white rectangles correspond to the tables that were used for this research. The most important data fields of these tables are written in the table. The colored boxes group the tables in a category. To connect the tables with each other, the relation tables (the red tables in the middle) are needed.
Figure A.1: Model of the Vodafone data warehouse
Appendix B

Extra results for optimal number of clusters

In this Appendix, the plots of the validation measures, for the algorithms that not were discussed in Section 4.1, are given.

The K-medoid algorithm:

Figure B.1: Partition index and Separation index of K-medoid
The Fuzzy-C-means algorithm:

Figure B.2: Dunn’s index and Alternative Dunn’s index of K-medoid

Figure B.3: Partition coefficient and Classification Entropy of Fuzzy C-means
Figure B.4: Partition index, Separation index and Xie Beni index of Fuzzy C-means

Figure B.5: Dunn’s index and Alternative Dunn’s index of Fuzzy C-means