Nicolas March

Building a Data Mining Framework for Target Marketing

Thesis

DOKTORAT DER SOZIAL- UND WIRTSCHAFTSWISSENSCHAFTEN

1. Beurteilerin/1. Beurteiler: **a. o. Prof. Dr. Thomas Reutterer**

2. Beurteilerin/2. Beurteiler: **o. Prof. Dr. Kurt Hornik**

Eingereicht am: ____

WIRTSCHAFTS
UNIVERSITÄT
WIEN VIENNA
UNIVERSITY OF
ECONOMICS
AND BUSINESS

EFMD
EQUIS
ACCREDITED

Titel der Dissertation:

Building a Data Mining Framework for Target Marketing

Dissertation zur Erlangung des akademischen Grades

einer Doktorin/eines Doktors

der Sozial- und Wirtschaftswissenschaften an der Wirtschaftsuniversität Wien

eingereicht bei

1. Beurteilerin/1. Beurteiler: **a. o. Prof. Dr. Thomas Reutterer**

2. Beurteilerin/2. Beurteiler: **o. Prof. Dr. Kurt Hornik**

von **Dipl. Kfm. Nicolas March**

Fachgebiet: **Betriebswirtschaftslehre**

Wien, im **Mai 2011**

WIRTSCHAFTS
UNIVERSITÄT
WIEN VIENNA
UNIVERSITY OF
ECONOMICS
AND BUSINESS

EFMD
EQUIS
ACCREDITED

Ich versichere:

1. dass ich die Dissertation selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfe bedient habe.

2. dass ich diese Dissertation bisher weder im In- noch im Ausland (einer Beurteilerin/ einem Beurteiler zur Begutachtung) in irgendeiner Form als Prüfungsarbeit vorgelegt habe.

3. dass dieses Exemplar mit der beurteilten Arbeit übereinstimmt.

Datum ____________________ Unterschrift ______________________________

Kurzfassung

Die Unterstützung der Kaufentscheidung einzelner Kunden oder Kundengruppen mit spezifischen Produktempfehlungen ist erfolgversprechend. Darüber sind sich sowohl Händler als auch Wissenschaftler einig. Zielgerichtete Werbemanahmen erweisen sich im Gegensatz zu einheitlichen Verfahren der Absatzförderung (z. B. in Form von Sonderpreisen oder Preisnachlässen) als profitabler. Dies ist insbesondere dann der Fall, wenn die beworbenen Produkte optimal auf die Präferenzen der Kunden bzw. Kundengruppen abgestimmt sind. Doch wie kann der Händler Kundengruppen bilden und welche Produkte soll er ihnen anbieten? Zur Klärung dieser Frage beschreibt die Dissertation ein algorithmisches Verfahren, das in aufgezeichneten Transaktionsdatenbanken Kundengruppen anhand ihrer Präferenzen für ähnliche Produktkombinationen identifiziert. Zudem werden für jede Kundengruppe zu empfehlende Produkte definiert, die bei entsprechender Bewerbung einen höheren Absatz durch Cross-Selling versprechen. Um die Anwendung des algorithmischen Verfahrens zu verdeutlichen, werden exemplarisch die Transaktionsdaten eines Supermarktes analysiert. Die dabei gefundenen Kundengruppen und Produktempfehlungen werden für eine Simulation verwendet. Diese zeigt, dass entsprechende Werbemaßnahmen unter Verwendung des algorithmischen Verfahrens im Gegensatz zu einheitlichen Preisnachlässen auf Bestseller Gewinnsteigerungen zwischen 15% und 191% ermöglichen können.

Abstract

Most retailers and scientists agree that supporting the buying decisions of individual customers or groups of customers with specific product recommendations holds great promise. Target-oriented promotional campaigns are more profitable in comparison to uniform methods of sale promotion such as discount pricing campaigns. This seems to be particulary true if the promoted products are well matched to the preferences of the customers or customer groups. But how can retailers identify customer groups and determine which products to offer them? To answer this question, this dissertation describes an algorithmic procedure which identifies customer groups with similar preferences for specific product combinations in recorded transaction data. In addition, for each customer group it recommends products which promise higher sales through cross-selling if appropriate promotion techniques are applied. To illustrate the application of this algorithmic approach, an analysis is performed on the transaction database of a supermarket. The identified customer groups are used for a simulation. The results show that appropriate promotional campaigns which implement this algorithmic approach can achieve an increase in profit from 15% to as much as 191% in contrast to uniform discounts on the purchase price of bestsellers.

Acknowledgements

First of all, I am indebted to my advisor, Prof. Thomas Reutterer, who invested a great deal of effort in guiding me through the last few years. In the same way I have to thank my co-advisor, Prof. Kurt Hornik, who spent a lot of his time answering my questions in numerous meetings, phone calls and emails. Additionally, I am grateful to Prof. Friedrich Leisch of the Ludwig-Maximilian University of Munich for his input into this thesis.

I want to thank the CEOs of Troostwijk Veilingen B.V. (Amsterdam), Mr. Wilhelmus Dieker and Mr. Tjade Dieker, as well as the company's IT-Manager, Mr. Peter den Hollander. By showing consideration for my desire to finish this thesis, my supervisors and colleagues gave me the flexibility I needed to combine my work and my research.

Furthermore, I am grateful to my friends Carsten Bonn, Christoph Cöln, Dr. Yassin Dimassi, Christoph Faber, and Dr. Christian Nettersheim. In addition, I want to thank my doctoral colleagues in Austria who made the time in Vienna unforgettable.

Finally, I want to thank my family, who gave me advice, support and also encouragement when I really needed it.

Contents

> "We are drowning in information but
> starved for knowledge."
>
> *(J. Naisbitt)*

1 Introduction

1.1 Motivation

Customers in every sector of the economy make thousands of choice decisions every day. The clients of insurance companies enter into different kinds of contracts, bank customers choose among several investment products and shoppers at supermarkets fill their baskets with only a few items from the broad assortment available. With the increasing power of computational systems, companies have started recording the choice decisions of their clientele in datasets with the aim of using the information to enhance their business strategies (cf. Rossi, McCulloch and Allenby 1996, Adomavicius and Tuzhilin 2001). Marketing researchers early assumed that different individuals share similar choice behavior, due to such factors as comparable life cycles, similar demographical ancestry or simply common interests (cf. Ainslie and Rossi 1998, Andrews and Currim 2002, Solomon, Bamossy, Askegaard and Hogg 2006). Today, companies are becoming more sensitive to the ways in which this information can be used to better fulfill the expectations of their customers and to act more according to their clientele's needs.

Particularly in retailing, this is an important objective (cf. Kahn and McAllister 1997). Many retailers operate in highly competitive markets since customers are able to choose among many different companies offering similar product ranges. Since those firms heavily canvass the same customer base, decision-makers focus on the essential requirement of communicating with consumers on a more personal level in order to collect knowledge about their expectations and to satisfy their needs in the most suitable way. The need to concentrate on customers is reflected in retailers' increasing efforts at customer relationship management (CRM), embracing all activities to intensify the business connection between sellers and buyers. Instead of focusing on the seller, CRM emphasizes the interests of the customers and proposes ways to adjust the retailer's offer to meet their expectations (cf. Leenheer 2004, Neckel and Knoblauch 2005).

Observing the behavioral purchase patterns of customers and deriving their potential interest in certain products depends on recording the contents of the shopping baskets. Whereas online shops can collect this data without much difficulty, stationary retailers have to install scanning devices at the point of sale (POS). These systems store every single transaction in the connected databases of a retailer's data warehouse systems (cf. Mallach 2000). With the implementation of registration and loyalty programs which link each transaction to the originator of the record, retailers nowadays are collecting huge amounts of personalized transaction data comprising detailed knowledge of individual behavioral purchase patterns. For each registered customer, retailers are able to summarize the records of the market basket in a customer-specific buying history. Such sequences of transactions reflect buyers' habits – particularly with regard to their commonly purchased categories – over a period of observation (cf. Rossi et al. 1996, Adomavicius and Tuzhilin 2001, Acquisti and Varian 2005). Although many firms are aware that they should be using this information to target their customers with more appropriate offers, little is known about how to derive strategic decisions from the collected transaction sequences in retailing.

Revealing hidden information in data warehouse systems is a complex challenge due to the high dimensionality of the growing datasets. In contrast to many other research streams in marketing, researchers analyzing buying behavior as reflected in transactions struggle under a weight of data. It is a simple fact that our limited cognitive abilities prevent us from identifying and comprehending the hidden information which might be useful for companies' future actions. This explains the growing interest of marketers in business intelligence (BI), which uses computational techniques to extract information from the data warehouse systems that can then be used to support a company's strategic management decisions (cf. Cody, Kreulen, Krishna and Spangler 2002, Turban, Aronson, Liang and Sharda 2008). The field of BI is separated into online analytical processing (OLAP) techniques and data mining. OLAP involves querying techniques that extract information from multidimensional datasets. In contrast to the field of data mining, the analyst has an idea about what he wants to verify. Data mining is used to reveal yet-unknown information in the data. Most of its techniques are derived from the field of knowledge discovery in databases (KDD). The statistical and computational algorithms of KDD try to mine repeated patterns in huge databases efficiently (cf. Fayyad, Piatetsky-Shapiro and Smyth 1996). Since there are as many explanations of data mining as there are authors writing about it, in this study we define data mining in accordance with Kuonen (2005) as *"a process of identifying valid, novel, potentially useful and comprehensible patterns or models in data to make crucial business decisions"*. The process of data mining is interdisciplinary. Knowledge from statistics, computer science and business administration is needed to develop successful management decisions when analyzing different data sources.

The main data source of the analytical data mining approach presented in this dissertation is the recorded and collected transactions of a retailer's clientele. Analysts conducting market basket analysis assume that each transaction of a customer (= shopping basket, market basket) can be interpreted as the output of a combined multicategory decision process made during a shopping trip (Manchanda, Ansari and Gupta 1999, Russell and Petersen 2000). Finding category co-occurrence patterns within the summarized transactions of a sufficiently large part of a retailer's clientele could allow the retailer to draw conclusions about customers' consumption behavior – especially about their typical product compositions. Knowledge about purchasing behavior could then help in adjusting the business to the customer's expectations, e.g. with customized promotions or optimized product ranges. The objective of market basket analysis is to reveal and explain the choice patterns which are needed to develop suitable marketing strategies.

Hence, the regular composition of specific items during one or several purchase occasions can be valuable information for marketers. If a sufficient proportion of customers compose the same products in their shopping baskets, retailers can reason that there is an implicit correlation or connection among the products. Researchers have proposed some explanations for why people buy products in combination. For example, it is thought that customers reduce their effort when buying all the items during a single-stop occasion (cf. Bell and Lattin 1998, Sheth and Mittal 2004). Moreover, the size of a household budget can have an impact on buying behavior and item combination (cf. Chintagunta and Song 2007).

Concerning the assumptions of the present work, we suggest that certain consumers share similar preferences, which are reflected in a comparable purchasing behavior and a similar combination of specific products. In consequence, these customers could react in the same way to target marketing activities such as customized promotional campaigns (cf. Ainslie and Rossi 1998, Seetharaman, Ainslie and Chintagunta 1999, Cottle 2000). To target the buyers, it is necessary to know which households are responsible for the occurrence of certain purchase correlations among the items of a retailer's assortments. Identifying the relationship between articles is also important for optimal category management. For example, deleting one of two often-combined categories would also effect the sales of the correlated product (cf. Müller-Hagedorn 2005). This highlights the linkage between the field of category management and market basket analysis (cf. Jiang, Klein and Pick 1998). Instead of developing category management decisions, we use identified purchase correlations to target similar customers with more appropriate offers (cf. Gosh 1997). This could prove important not only in retailing but also in other selling environments such as finance, insurance, service and many more sectors.

1.2 Objectives

In accordance with Ahn, Kim and Han (2003), we understand target marketing as a collection of strategies specifying "customer groups (in order) to solve the problems of mass marketing and to raise marketing efficiency". Although companies are often aware that they can use the transaction data gained from their data warehouse systems for strategic marketing, they rarely attempt to apply this data in making concrete customer relationship management decisions. This thesis will make this more feasible by introducing the building blocks of a data-driven target marketing approach. The approach uses collected transaction data to create better customized promotion. Since one-to-one marketing is still not practicable for most business cases in retailing, the first objective of the approach is to identify groups of customers which should be presented with more individual offers. This task refers to the field of market segmentation, which tries to arrange the clientele into homogeneous groups of addressable customers (Smith 1956, Dibb and Simkin 1996, Wedel and Kamakura 2000, Grapentine and Boomgaarden 2003). The customers are grouped according to their similar market basket compositions. This is the main characteristic defining the typical purchasing behavior of the members of a segment. Many different approaches exist using other dimensions, e.g. those derived from demographics or geography to distinguish the households (cf. Duchessi, Schaninger and Nowak 2004, Weinstein 2004). For example, the RFM (recency/frequency/monetary value) applications consider, in addition to the monetary value of the shopping trip, the recency and frequency of customers' store visits in order to separate buyers into addressable segments (Hughes 1996, Aaker, Kumar and George 1998). Customer Lifetime Value (CLV) approaches are predictive and try to determine the customers belonging to a specific group according to the expected equity. This helps focus on retailers' most valuable buyers (Mulhern 1999, Berry and Linoff 2004, Rust, Lemon and Zeithaml 2004). In addition, methods combining both approaches are also possible (Fader, Hardie and Lee 2005).

These efforts aim at identifying different customer segments, but they do not suggest a clear strategy to retailers for determining what kinds of products should be offered to the identified groups. In contrast, the approach in our study will also solve this "item selection problem" (Blattberg and Neslin 1990). The approach calculates a list of items whose sales in a segment should be supported by a suitable promotional campaign. The recommended items for these segment-specific campaigns are derived from products which are combined frequently and commonly enough in the transactions of a segment during customers' purchase occasions to imply a purchase connection. In other words, the idea is to use similar purchase correlations not only to identify customer segments but also to advertise the categories involved to the members of the segment. Retailers have been trying for a long time to comprehend the correlations between parts of their assortments by analyzing jointly-combined products. Several different techniques measuring category interdependencies have been developed (cf. Böcker 1978, Merkle 1981, Hruschka 1991). Knowing the relationships between items of retail assortments can help to enhance cross-selling effects (cf. Harding 2002). For example, if customers often purchase product *A* in combination with product *B*, it could be useful to promote one of the products to boost the sales volume of the other one, and vice versa.

In addition to combining the customer segmentation and the item selection problem, the approach extracts the derived items from the category correlations between items bought less frequently. In a typical retail assortment, the purchase frequencies of the offered items are not distributed equally. Many distributions of category purchase frequencies recorded by different retail firms tend to be skewed. This is the case with the one analyzed in our empirical application (see the black solid line on the left-hand side in Figure 4.7),and in the studies by Hui, Tan and Kumar (2006) and Anderson (2006). Usually, about 10% to 20% of an assortment's categories are bought disproportionally often, whereas the rest occurs rarely within the market baskets. Concerning grocers, these highly frequently bought categories (HFC) contain products needed every day, e.g. dairy products such as fresh milk. Some form of milk occurs in almost every second transaction at most common groceries or supermarkets. Hence, these top categories are often used for "one-for-all" promotion campaigns which are applied to the retailer's entire clientele (cf. Elberse 2008). However, we think that consumers can be better distinguished according to the purchases they make in the categories bought less frequently (LFC). For example, a young family and a senior citizen will both buy the highly frequent food categories such as milk and vegetables, but they will more likely differ in buying baby products and denture adhesive. Retailers have become aware of the benefits of using customers' interest in buying these "long-tail" categories of the assortment for making marketing decisions (cf. Anderson 2006). The customer segmentation component of the approach will implement the category correlations in the LFC as the characteristic which distinguishes the buyers, since these underlying purchase habits differentiate the customers more clearly than possible correlations in the HFC. Of course, combinations of categories from the HFC (such as *bread* and *milk*) could also be used for segmentation purposes. Nevertheless, considering the grouping structure subject to category combinations in the LFC seems to be more valuable from the target marketer's point of view.

Since the less frequently bought items contribute less to retailers' overall earnings, we have to validate the profitability of a segment-specific target marketing approach using the LFC compared to typical one-for-all promotion campaigns which usually involve the bestsellers (i.e. items of the HFC). A simulation is used to apply the approach to empirical data and estimates the profits generated when the selected items are promoted using successfully conducted target marketing techniques. Based on the extraction of the items from customer groups showing similar category compositions in the past, we assume that target marketing with the more customer-compliant items initiates cross-selling and increases customer loyalty. From a methodological point of view, the approach should not only take statistical correlations between the items into account, but also the monetary values of the defined categories.

The following sections of the thesis will introduce the development of the building blocks of the target marketing approach used to analyze personalized binary transactions gained from customer loyalty programs. The approach should increase retailers' earnings compared to a common customer-unspecific promotional heuristic. By introducing the approach and applying it to real-world data, we aim to achieve three objectives:

1. **CUSTOMER SEGMENTATION:** The approach should segment the clientele of a stationary re-

tailer into customer groups worth addressing with more customized offers. Since different partition techniques can be used, the most appropriate one should be determined.

2. **ITEM RECOMMENDATION:** The approach should name specific categories which might be suitable for customized promotional campaigns in each segment. The recommended categories should not be drawn from the bestsellers but from the “long-tail” of the assortment. The selected items should nevertheless have a sufficient impact on retailer’s earnings and show high purchase correlations to other items.

3. **PROFIT ESTIMATION:** A simulation of the approach should illustrate the expected return if the defined items are used for segment-specific marketing campaigns. Moreover, the expected growth should be compared to the growth initiated with a one-for-all campaign promoting only the bestsellers of a retailer.

The introduction of the approach, the discussion of its modules and its simulation using real-world data fulfills the above stated objectives. The results of this thesis are applicable not only to retail companies but to all firms who want to conduct more customer-compliant marketing campaigns by analyzing their recorded transaction data with the suggested framework.

1.3 Theoretical Background

In this section, the theoretical and methodological background of the thesis is explained and embedded into the general process of knowledge production in science. Moreover, we compare the methods used to the ones found in common scientific work of business administration.

Gaining knowledge is a stepwise process (cf. von Alemann 1984, Aamodt and Nygård 1995, Schulz and Nocke 2007). Usually, the researcher extracts from data relevant information, which produces knowledge if the data verifies the found information. If statistical techniques are used to analyze empirical data, knowledge corresponds to the ability to predict and possibly explain coherencies. In published work in scientific business administration, the majority of examinations apply statistical data models to enable prediction and explanation (cf. Breimann 2001, Homburg 2007). The authors set up hypotheses about causal effects and validate them by testing the models against the collected data. For instance, the analyst might assume a correlation between a response variable y and an input variable x. He may think that a linear regression model expresses the correlation between y and x best. By estimating the parameters, the analyst develops the model and hopes to explain the causal connection. The validation of the model is often done with goodness-of-fit tests (e.g. R^2). The quality of the explanation depends on how well the model fits to the real correlation between *y* and *x*. The left-hand side of Figure 1.1 illustrates this modeling culture in statistics.

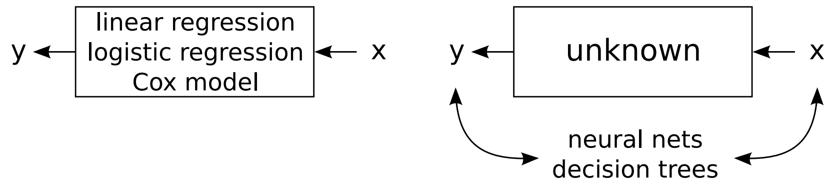

Figure 1.1: Illustration of the two statistical cultures (Breimann 2001): data modeling (left-hand side) and algorithmic modeling (right-hand side)

In contrast, the prevailing methods applied in this thesis belong to the much less common practice of algorithmic modeling. In contrast to data modeling, algorithmic modeling, illustrated by the right-hand side of Figure 1.1, comprises the development of a more or less complex function which predicts the reaction of *y* according to *x* (Breimann 2001). Instead of aiming at explaining a cause-effect relationship between *y* and *x*, the description and prediction of the correlation is the more important objective.

Concerning the dimensionality of the data, the fundamental problem arises from our limited human cognitive abilities, which do not allow us to apprehend the data in a useful way. To overcome human limitations, appropriate statistical and computational methods of data mining are used. The approach of this thesis is exploratory since its objective is to reveal unknown information instead of explaining the correlations. A positivistic view affects the approach due to its strong connection to statistics and computer science, whose algorithmic functions are used to achieve the above-stated aims (cf. Kanitscheider 1981). In Heinzl's (2001) terms, the approach postulates a "technique-challenge" relationship.

The question arises whether descriptive research is just a preparatory step that cannot be classified as the work of science. For von Alemann (1984), data exploration justifies descriptive research since the acquired information can be the source of new hypotheses leading to important conclusions and knowledge (cf. Kanitscheider 1981). The exploration of customer segments and unknown product correlations are the key objectives of our approach and legitimate its scientific nature (cf. Schulz and Nocke 2007).

1.4 Outline

The thesis is structured as follows: in Chapter 2 the reader is introduced to the field of market basket analysis (MBA). Since the proposed target marketing approach implements several exploratory techniques of market basket analysis, we give an overview of some of its relevant methods. We describe the development of the different techniques and their basic functionality. For illustration purposes, we refer to the Appendix, which includes the results of some short data examples. Chapter 3 explains the building blocks and the methodic concept of the approach. The main part concentrates on a partitioning step that identifies and builds the customer groups. Since it is a fundamental module, several different partition algorithms are discussed and analyzed in detail. Artificial binary transaction data is used to show the characteristic of each algorithm and to derive the most suitable partition algorithm for the given context. The second part of Chapter 3 addresses the question of how the approach determines and evaluates the specific items used for customized target marketing. Besides mining algorithms, filter and grouping techniques, a linear programming model is described which finally extracts the items to be featured.

The target marketing approach introduced here is applied to the transaction data of a loyalty program at a supermarket in Chapter 4. This chapter describes the output of each step of the framework and illustrates some modifications. The modifications refer mainly to the partition algorithm used to identify the customer segments. The second part of Chapter 4 includes the what-if-scenario which simulates the profit increase of a segment-specific promotional campaign compared to a standardized one-for-all promotion heuristic. The segment-specific promotional campaigns use the items determined with our data-driven approach. We will show if and in what way the marketer can increase his profits when a less analytical promotion heuristic is replaced.

Finally, Chapter 5 summarizes the key findings of the thesis, discusses the results of the empirical application and gives an outlook on future enhancements. Moreover, a short decision scheme is presented to help managers to decide whether the approach should be implemented in their business environment.

2 Techniques of Exploratory Market Basket Analysis

Since category combinations are the key characteristic for defining similar customer groups and selecting the items which should be featured in the corresponding segment, their identification is of central importance over the entire approach. To extract the category choice patterns and the variables that effect them, the field of market basket analysis uses various techniques. This chapter introduces common techniques of MBA. We refer to the methods that are correlated to the building blocks of our target marketing approach, as described in Chapter 3.

The research stream of market basket analysis is separated into two sets of approaches (cf. Reutterer and Mild 2003, Boztug and Silberhorn 2006):

1. *Explanatory, model-based approaches* try to explain observed choice patterns. The underlying models focus the variables of the marketing mix, such as price or a promotion affecting the category choice of the customers. In addition, some models take latent and random effects into account. Usually, logit, probit or tobit models are used to describe the correlations (cf. Manchanda et al. 1999). The models often differ in the usage of estimation procedures, e.g. the maximum likelihood or the Markov chain Monte Carlo methods (cf. Adiguzel, Wedel and Zhang 2007). Although growing computational capacities have extended the power of estimation procedures, model-based approaches usually include only a limited number of categories (cf. Boztug and Silberhorn 2006).

2. The objective of *exploratory, descriptive approaches* is to reveal purchase correlations by analyzing the recorded transaction data systematically (cf. Reutterer and Mild 2003, Berry and Linoff 2004). These approaches aim at so far unknown purchase patterns. Due to our limited cognition, we humans cannot recognize all of the patterns adequately. The challenge is to condense the hidden information in large datasets by using counting or cluster algorithms which support the user in finding interesting patterns (cf. Hahsler, Grün and Hornik 2005, Hahsler, Hornik and Reutterer 2006, Reutterer, Hahsler and Hornik 2007).

The combination of exploratory and explanatory methods has also become popular (cf. Boztug and Reutterer 2008). Hereby, the revealed category purchase patterns of the descriptive analysis are passed on to a subsequent model-based approach. The subsequent step can be used to validate the correlations of the first step, and extends the analysis of the category correlations.

This dissertation employs the exploratory methods of market basket analysis because the main objective is to find purchase correlations with data-driven, analytical algorithms scanning large numbers of collected transactions with many different items. In contrast to the model-based approaches, which use estimation procedures to explain the correlation among a limited number of items or variables, the approach described in Chapter 3 tries to uncover category correlations without claiming or explaining any causality.

The following section introduces some different techniques of exploratory MBA in more detail. Before the methods are described, Section 2.1 examines the collection and construction of the transaction data. Since the methods of exploratory MBA depend on the attributes of the data, we explain the most important of these attributes.

2.1 Characteristics of Transaction Data

The name "market basket analysis" implies that the object of investigation is the observed shopping or market baskets of customers. We define a market basket, shopping basket or transaction as the combination of goods, products or even services a customer has bought during a single purchase occasion. The analysis of customers' purchasing behavior benefits from observing conducted transactions. For example, in contrast to interviews, the examination of market basket compositions is much more unbiased. It comes closer to a research observation since customers usually do not think about the fact that their behavior is registered by the data warehouse systems. The aggregated transactions of the retailer's clientele hide latent purchase patterns which customers are probably not aware of themselves. The focus of this section is the collection and aggregation of market basket records and the consideration of some important computational issues in their analysis.

2.1.1 Data Collection

To analyze purchase patterns with special regard to purchase correlations between items in an assortment, the decision maker has to observe the transactions or shopping baskets composed by his clientele. A very simple way to do this is to collect the sales slips of the cash registers at the point of sale and mark all the items bought during the shopping trip. Of course, modern retailers have replaced this manual registration with suitable hardware at the point of sale. Optical scanning devices at the cash register recognize the bought items by their bar code labels automatically (Grünblatt 2001). In Europe the European Article Number (EAN) and in the United States of America the Universal Product Code (UPC) allow a computational and automated recognition of the corresponding items. Recently, some companies have started to implement Radio Frequency Identification (RFID) technology. Instead of barcodes, articles are labeled with a transponder. Frequency readers enable the automatic identification of all items in the market basket without scanning each individual article manually (cf. Jäger 2007). Usually, the electronic cash registers are connected to the firm's data warehouse systems. They collect all the data sources of a company for further data analysis in a central data storage (Mallach 2000, Gonzales 2003). The data warehouse systems can provide the different databases of the retailer to an analyst for OLAP or data mining techniques. In stationary retailing, these data sources are primarily used to calculate stock capacities or accounting information. Rarely, the datasets are connected to sophisticated programs analyzing customers' purchasing behavior.

Concerning transaction data, each purchase occasion is recorded in a relational database in the data warehouse system. Relational databases record the information in tables. Each row of the table represents an entity and each column represents an attribute of the entity. The content of the table's cells specifies the values of the entity's attributes. Transferred to the MBA context, each single shopping basket, i.e. transaction x_n, corresponds to the row of a table in the relational database (with $n = 1 \ldots N$). Since the columns define the items $j = 1 \ldots J$ of an assortment, an empty cell or a zero value means that the item has not been bought during the purchase occasion. It has been bought when the cell shows a positive value. Table 2.1 illustrates an example data table X_N in a database.

m	x_n	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$	$j=7$	$j=8$
5	x_1	0	1	1	0	1	0	1	1
2	x_2	0	0	1	0	1	1	1	1
4	x_3	0	0	0	0	1	0	1	0
5	x_4	1	0	0	0	0	1	0	1
5	x_5	0	0	0	0	1	0	0	1
2	x_6	0	0	0	0	1	0	1	0
1	x_7	$u_{7,1} = 0$	1	1	0	1	0	0	1
3	x_8	1	0	0	0	0	1	0	1
5	x_9	1	0	0	1	1	0	0	0
4	x_{10}	1	0	0	0	1	1	0	1
2	x_{11}	1	1	0	0	1	1	1	0
2	x_{12}	0	0	0	1	1	1	0	1
1	x_{13}	1	1	1	0	0	0	0	1
3	x_{14}	1	0	0	0	0	1	0	1
3	x_{15}	0	0	0	1	0	0	1	0
Customer Identification	Binary coded data table								

Table 2.1: Binary coded data table X_N with customer identification m (cf. Decker and Schimmelpfennig 2002 or Müller-Hagedorn 2005)

In the early stages of exploratory MBA, the objective was to find category correlations in the single onestop purchase occasions of aggregated customers. Hence, corresponding methods analyzed transaction data summaries without regard for who made the choice decision where and at what point in time. The analysts aimed at the right-hand side of the vertical divide in the example data table (see Table 2.1). When the temporal dimension of the customer's choice decision is ignored, the results of the underlying methods refer to the "connection of purchase". This defines all the items bought at a certain point in time (Böcker 1978, Poggenpohl 1993, Schnedlitz and Kleinberg 1994, Boztug and Silberhorn 2006). Although the individual customers are not tracked through time and space, identified purchase correlations within these simple two-dimensional transaction datasets give decision makers valuable information about the choice behavior of customers. Scanning anonymous transaction data can reveal important item correlations in retail assortments. For example, an identified purchase correlation between fish and white wine implies that the customers might respect the convention of serving seafood with white wine. As a consequence a retailer could place selected white wines beside the freezer containing fish. Despite the fact that assuming causality for every found item correlation carries risks (see Section 5.3), verified category purchase correlations might hold important information for category or promotion management.

The informational value of recorded shopping baskets has increased as many retailers have begun offering loyalty programs that provide registered members with plastic cards or user authentication data. When using these customer-identifying instruments for each purchase occasion, the corresponding transactions can be assigned to their originators. The programs make it possible to track each purchase occasion over an observational period. With this development, MBA methods are no longer limited to the "connection of purchase" but can be extended to the "connection of demand" which defines all the items bought by a household over a specific period of time (Böcker 1978, Poggenpohl 1993). Refering to Table 2.1, the column on the left-hand side of the vertical devide shows the extension of the database from the introduction of registration or loyalty programs tying each transaction to a customer identification number (ID) m. For example, the purchases x_1, x_3 and x_4 of Table 2.1 are assigned to customer $m = 1$. If the data collection systems of different shops are connected online to a central data warehouse, it does not even matter at which store the customer with the corresponding ID buys the items.

In the U.S., retail companies have been successfully issuing their own credit cards for many years, which facilitates recording the purchasing behavior of card holders. In Germany, the introduction of customer loyalty cards, bonus club cards and company credit cards has become possible due to the removal of legal barriers in 2001. Since this point in time, firms have built consortiums such as "Payback" or "DeutschlandCard", and now provide customers with many different programs. Although the loyalty programs can indeed increase retailers' profits in the long run, some authors advise keeping an eye on the program's profitability (cf. Reinartz and Kumar 2000, Reichheld 2001, Leenheer 2004). Many firms collect huge amounts of data with highly-sophisticated data warehouse systems but derive only small benefits from the information. Hence, it is crucial to develop methods for using the collected transaction data from loyalty programs and to discuss the question of whether the implementation of data-driven target marketing methods really increases the retailer's profit.

2.1.2 Computational Issues

Analyzing the recorded transaction data is usually done with computers. Since human cognition is limited, machines support decision makers in retrieving information from data. Even the modern capacities of computers are not infinite, however, which means that programs and methods of exploratory MBA must not waste the given computational resources. The different invented techniques vary in the effectiveness with which they extract the purchase correlation from the data set. To understand why some techniques are more effective than others, it is necessary to consider the components of a computer system and how the algorithms of MBA interact with these components to identify purchase correlations. An "algorithm" is a finite course of action to solve a stated calculatory problem (cf. Cormen 2007). In the MBA context, an algorithm represents a program code or a mathematical formulation identifying the purchase correlations within the data table.

Looking at the computational power of computer systems such as mainframes or personal computers in a simplified way, the following hardware devices are of interest: the central processing unit (CPU), the random access memory (RAM), and non-volatile storage devices such as hard disk drives (HDD) (cf. Zilahi-Szabó 1998). The storage devices gather the databases of the data warehouse system; these devices provide the high storage capacities that are needed. For the analysis of the transaction data, the programs load parts of the dataset sequentially into the smaller RAM and start the algorithm. The algorithmic program requires the power of the CPU to finally solve the calculatory problem. A comparison of the effectiveness of two programs can be done by comparing two complexity measures: that of computing time and that of storage space. For a given set of computational resources and an equal output, one algorithm performs better than another if it optimizes the computing time and minimizes the storage space. In practice, some MBA algorithms lower the computational load by reducing time-consuming input-output (I/O) processes between RAM and storage devices. I/O processes describe the exchange of data between the different parts of a computer system. For example, the transfer of data between the storage device (e.g. the hard disk) and the RAM is a typical I/O process. Some algorithms reduce the I/O processes by sampling techniques which pass on smaller parts from the slow storage devices to the processing unit. The algorithms themselves can be optimized, e.g. by pruning steps. They make it possible to process more data in the CPU or read more data into the fast RAM (cf. Reischuk 1990, Zilahi-Szabó 1998, Ottmann and Widmayer 2002). We will present some techniques for optimizing the algorithms in Section 2.2. To summarize, purchase correlations or patterns can be found in data with many different algorithms, but the efficiency of the algorithms in extracting the valuable information varies. Hence, computational issues are an important point when analyzing high dimensional transaction data.

Sparsity and Dimensionality of the Transaction Data

Increasing the effectiveness of the algorithms has become even more important as exploratory MBA has begun to face further computational challenges. Although the capacities of modern computers have grown enormously over the last forty years, the amount of collected data in retailing has become larger too. Scanning devices are collecting thousands of transactions in each retail store every day, hundreds of members have started taking part in loyalty programs, and retailers are extending their assortments to increase the probability of satisfying their customers. With reference to the growth of the data table X_N, the number of rows and columns is increasing and the analytical procedures used have to deal with high-dimensional data. This explains the jump in demand for efficient algorithms to inspect these datasets in a justifiable period of time (cf. Strehl 2002).

In addition to the increasing dimensions of the data table X_N, it is often very sparsely provided with values. Here, the sparsity of a data table defines the relation between non-zero and total cells (cf. Strehl and Ghosh 2003, Strehl 2002). For our example data table (see Table 2.1), the matrix shows a sparsity of $42.5\% = 51/120$ which is not a real value for most selling environments. Because customers usually buy only a small part of the product range in stationary retail environments, only a few non-zero values appear in a transaction. Most data tables of supermarkets show a sparsity of about 95% and higher. Some online markets (e.g. booksellers) construct even a much more sparse transaction data matrix since customers can sometimes choose among 10,000 to 100,000 articles. For the techniques of MBA it becomes more difficult to find similar, related transactions in sparse transaction data since the probability of individual product compositions increases. Finding similar market basket compositions in the data table is a key idea of many MBA techniques. In cases where data is very sparse, these techniques can lead to poor results.

To overcome the difficulties of high-dimensional, sparse data in the MBA context, the analyst can look for item correlations according to different degrees of item aggregation. Following Müller-Hagedorn (2005), Table 2.2 describes four different levels of aggregation relevant in the context of this thesis.

When retailers want to clarify the purchase correlations between single articles, the computational challenge is the highest since nearly the whole product range has to be considered for the analysis. This is not always the most effective course of action. Purchase correlations on the category level or on an even lower aggregation level can also bring out useful knowledge about customers' buying preferences. Hence, an analyst can reduce the number of columns in the data table by summarizing the sales of

Item	Description	Example	Level of aggregation
Article/SKU	The smallest unit of disposition with important economical differences (such as manufacturer, package size, sort etc.)	Whole milk chocolate from different manufacturers/ in different package sizes etc.	lowest
Category (of goods)	Groups of articles with different varieties and sorts.	Whole milk chocolate, white chocolate, chocolate with nuts	low
Class of goods	Categories satisfying a special need or wish.	Chocolate	high
Kind of goods	Similar to the sector	Candies	highest

Table 2.2: Levels of item aggregation in a typical retail assortment (cf. Müller-Hagedorn, 2005)

products in a category. The drawbacks of neglecting information are balanced by two possitive effects:

1. The number of observed attributes of the data table (i.e. the number of j-columns) becomes much smaller. This reduces at least one dimension of the matrix. For example, the 10,000 SKUs of a virtual supermarket can be assigned to 250 categories of 40 SKUs each.

2. At the same time, the sparsity of the data table decreases. If the retailer recognizes each purchase of the customer in one of the 40 articles as a sale in the corresponding superior category, the matrix will tend to have more filled cells.

Both issues can positively affect the output of the MBA algorithms. On the one hand, similarities between shopping baskets can be found more easily, and on the other hand the reduced number of items releases computational resources. For the purposes of this dissertation, we refer mainly to the categories of a retail company, although all studies can be transferred to subcategory levels (e.g. SKUs). Thresholds are only given by computational limitations. To avoid misunderstandings, in what follows, each of the expressions “item”, “article”, “category” and “product” represents the units of an offered assortment. In the same way, the expressions “category correlation”, “association” or “itemset” name the purchase relationship identified between two or more units of the analyzed dataset.

Scale of the Transaction Data

A further issue of some importance is the scale of the collected data. The cash register usually fills the row of each transaction with the amount spent in the corresponding categories. Hence, the data table contains metric values (e.g. sales volume or even gross margins of the corresponding item). Since retailers are more interested in knowing which items were bought during the purchase occasion than in how much money was spent on the items, typical exploratory MBA approaches assume binary coded data (Agrawal, Imielinski and Swami 1993, Schnedlitz, Reutterer and Joos 2001, Decker and

Schimmelpfennig 2002, Reutterer and Mild 2003, Müller-Hagedorn 2005). Binary data does not burden computational resources as much as do metric values. This speeds up most algorithms and helps even in the case of high-dimensional data tables (cf. Merkle 1981, Berry and Linoff 2004). The occurrence of a category in a market basket is denoted by a value of one and its absence by a zero value of $u_{n,j}$ (cf. Russell and Petersen 2000). On the other hand, binary data implies that these techniques exclude quantities and prices of the purchased products in the analysis. Neglecting this additional information might have a detrimental effect on the results (cf. Strehl and Ghosh 2003, Brijs, Swinnen, Vanhoof and Wets 2004). For example, a binary coded market basket with three bottles of champagne and 10 boxes of oysters has the same statistical strength as a market basket with a bottle of milk and a slice of cheese. Nevertheless, the metric values behind the binary variables are still present and can be used in a later stage of the mining procedure. For example, the target marketing approach described in Chapter 3 uses binary data in the first mining steps and re-imports the metric values when the number of considered items has been pruned.

The following sections give an overview of the existing techniques and algorithms used to analyze collected and recorded binary data. Section 2.2 starts with techniques employed to find purchase correlations in anonymous transaction data, i.e. data which does not allow the retailer to identify the customers behind the recorded transactions. This means that revealed purchase correlations are valid for the average customer only. Section 2.3 introduces techniques which consider the linkage to registered buyers.

2.2 Analysis of Anonymous Transaction Data

Since we do not address customer-identifying loyalty programs in this section, the following methods analyze the purchase connection between the items of an assortment without taking account of the individual behavior of the customers. Hence, the subsequent techniques analyze the binary data on the right-hand side of the vertical divide in Table 2.1 and neglect the customer ID in the first column. Starting with pairwise association analysis, the basic method involves comparing the purchase incidences of two single items in the transaction dataset using similarity or distance measures. Since partition techniques also need these measures to find similarities between two entities, Section 2.2.3 can refer to the basic methodology of pairwise association analysis. Association rule mining is the most popular technique for extracting purchase correlations from non-personalized datasets, and defines the content of Section 2.2.2.

2.2.1 Pairwise Association Analysis

Initial research looks at the pairwise purchase correlations between two items of an assortment, i.e. how often item i and j are bought in common. Hence, the interesting entities are the columns of the data table instead of the rows. The matrix X_N can be considered to be a transposed one. The purchase relationship between two items grows with their co-occurrence in a sufficiently large share of all market baskets.

Böcker (1978) and Merkle (1981) devised and tested different statistical measures which derive the similarity between two items by counting their common occurrence and absence within a dataset. Many authors have developed a wide variety of different similarity measures and discussed their characteristics extensively in the literature. To describe the measures of similarity for nominal scaled data, a table of contingency is often used (see Table 2.3 and Merkle 1981, Backhaus, Plinke, Erichson et al. 2006).

	j is bought	j is not bought	Σ
i is bought	a	b	a + b
i is not bought	c	d	c + d
Σ	a + c	b + d	a + b + c + d

Table 2.3: Table of contingency to explain measures of similarity (cf. Merkle 1981)

Different relative arrangements of the cells can express the similarity or dissimilarity between two entities. To identify purchase correlations in binary data, many authors propose the Tanimoto-similarity measure described in Equation 2.1 (Bordemann 1985).

$$\Psi_{Tan} = \frac{a}{a+b+c} \quad with\ \Psi_{Tan} \in [0,1] \tag{2.1}$$

The measure calculates how often i and j co-occur within the data compared to the sum of entities in which at least one of them is bought. For example, the pairwise Tanimoto-coefficient between item j_1 and j_2 of the transposed data matrix in Table 2.1 is $\frac{2}{5+2+2} \approx 0.22$.

The similarity between two entities is often expressed with a distance measure. If the values of the entities correspond to the coordinates of a representative point in the J-dimensional feature space, the distance between two points reflects the similarity between the corresponding entities. A high distance between two entities means a low similarity (or a high dissimilarity). In this thesis, the algorithms usually implement distance measures, e.g. the Jaccard distance (cf. Anderberg 1973). The Tanimoto similarity value can be transferred into the Jaccard distance as follows:

$$d_{Jaccard} = 1 - \Psi_{Tan} \tag{2.2}$$

Techniques calculating the distance between metric scaled entities often implement the Euclidean or the Manhattan distance (cf. Backhaus et al. 2006). The Euclidean distance in particular is quite important for partitioning since the well-known K-means algorithm described in Section 2.2.3 depends on it (Hartigan and Wong 1979). The Manhattan and Euclidean distances are derived from the Minkowski metric as shown in Equation 2.3.

$$D_{i,j} = [\sum_{n=1}^{N} \mid u_{i,n} - u_{j,n} \mid^{r}]^{\frac{1}{r}} \tag{2.3}$$

When the constant r is set to a value of one, Equation 2.3 reflects the Manhattan distance. For binary data the Manhattan distance leads to the same results as the output derived from the Hamming distance. Hence, the Hamming distance is often used to describe the dissimilarity in binary machine code (Hamming 1950). When $r = 2$, the Minkowski metric describes the Euclidean distance. Due to the exponent $r = 2$ in the brackets, bigger differences get a higher weight compared to $r = 1$. Raising to the power of two increases the sensitivity of the Euclidean distance when identifying dissimilarities in metric data.

Although the measures derived from Equation 2.3 are developed to calculate the dissimilarity between metric scaled entities, the Euclidean and Manhattan distances can also be applied to binary datasets. Nevertheless, in the case of sparse market basket data, the Jaccard-distance seems to be more appropriate. For explanation purposes, the following transposed table comprises four binary coded rows with the composition of two virtual products i and j.

	1	2	3	4	5
$j = 1$	0	0	0	0	1
$i = 2$	0	0	0	1	0
$j = 3$	1	1	0	0	1
$i = 4$	1	1	0	1	0

Table 2.4: Example to demonstrate the advantage of the Jaccard distance when examining binary market basket data

The first two rows represent the composition of $j = 1$ and $i = 2$ and show that the items have never been bought in common. Rows No. 1 and No. 2 are totally different when evaluating the co-occurrence of the items within the five baskets. The Manhattan distance leads to an absolute value of $|0-0| + |0-0| + |0-1| + |1-0| = 2$ and the Jaccard distance to a value of $1 - (0/0+1+1) = 1$. Now, let us compare the lower two rows No. 3 and No. 4. The item composition is the same as that in the case just discussed except for the first two baskets, which both include the items in this case. The Manhattan distance remains at a value of two and implies that there is no change compared to the first case. In contrast, the Jaccard distance decreases to a value of 0.5 and implies that rows No. 3 and No. 4 are now more similar to each other than rows No. 1 and No. 2. Instead of the Manhattan or Euclidean distance, the Jaccard distance weights the co-occurrence of one values more strongly. For sparse binary data, this is an important characteristic since we are more interested in finding the products which have been bought in combination than the ones which have not. We will return to this important issue in Section 3.1.4. It examines the dependencies between the selection of the distance measure and the results of some specific cluster algorithms.

The foregoing discussion shows that there is no universal measure that can be used for every application and context. The analyst always has to choose an appropriate measure according to the scenario and the general expectations about the achievable results. Hence, applying a suitable measure is key to getting a valuable outcome from analysis techniques.

For each pairwise item combination the user has to calculate the corresponding association value regardless of the implemented similarity or distance measure. The values are usually passed on to a distance or similarity matrix, as shown in the Appendix for the data example of Table 2.1 (see Section A.1.1, Table A.1). Thereby, the similarity or distance measure can express a symmetrical or asymmetrical direction of item correlation. A symmetrical association means that the purchase correlation between items i and j is of equal strength. Hence, the distance or similarity matrix does not need to include values above the diagonal.

Regarding asymmetrical purchase correlations, the direction of the interdependence between two items is important. For example, Hruschka's (1991) probabilistic model of purchase correlations also determines the reverse correlation of two items. Let us assume that two items i, j of an assortment are not correlated. The definition of the statistical independence determines that the common purchase probability is the product of the single purchase probabilities of the items:

$$P(i \cap j) = P(i) * P(j) \tag{2.4}$$

If the retailer assumes a purchase correlation between the items, the common purchase probability has to be higher than the product of the single purchase probabilities of the considered items:

$$P(i \cap j) > P(i) * P(j) \tag{2.5}$$

According to the definition of the conditional probability, the left side of Inequation 2.5 can be replaced with the expression in Equation 2.6:

$$P(i \mid j) = \frac{P(i \cap j)}{P(j)} \quad \Leftrightarrow \quad P(i \cap j) = P(i \mid j) * P(j) \tag{2.6}$$

$$P(i \mid j) * P(j) > P(i) * P(j) \quad \Leftrightarrow \quad P(i \mid j) > P(i) \tag{2.7}$$

The resulting Inequation 2.7 determines a complementary purchase probability between items i and j. It becomes true if the purchase probability of i, on the condition that j has been bought, is higher than the single purchase probability of i.

The deduced assumption justifies the usage of the conditional purchase probability for finding pairwise category purchase correlations in binary data. The cells of the contingency table can also be used to express the purchase correlation between i and j according to the probabilistic model (see Table 2.3).

$$P(i \mid j) = \frac{P(i \cap j)}{P(j)} \quad \Leftrightarrow \quad P(i \mid j) = \frac{a}{a+c} \tag{2.8}$$

$$P(j \mid i) = \frac{P(i \cap j)}{P(i)} \quad \Leftrightarrow \quad P(j \mid i) = \frac{a}{a+b} \tag{2.9}$$

For each pair of items, the conditional purchase probability is calculated and inserted into the similarity or distance matrix. Tables A.1, A.2 and A.3 in Section A.1.1 of the Appendix show the distance matrices of the conditional probability, the Jaccard- and the Euclidean distance that result when the data matrix of Table 2.1 is used. Since the conditional probability defines an asymmetric purchase correlation, the values above the diagonal have to be calculated as well. This is because $P(i \mid j)$ usually differs from $P(j \mid i)$. With reference to Table A.1, the lowest value of the Jaccard distance is calculated between item i_2 and i_3 (0.4) and implies the strongest purchase correlation. The next lowest figures are 0.44, 0.45 and 0.58 pointing to similarities between the pairs made out of i_1, i_6 and i_8.

From a methodological point of view, the asymmetry of the probabilistic approach can be advantageous and problematic at the same time. On the one hand, the asymmetry provides more information about the relationship between two items. On the other hand it might be more complicated to find a useful application for this information in practice. For example, the retailer does not know a priori which item has caused the purchase correlation when he gets two values for one pair of items. Nevertheless, we will show that the conditional purchase correlation plays an important role for counting algorithms in Section 2.2.2.

In addition to counting the co-occurrence of items within the collected market baskets, a second major technique of exploratory market basket analysis is to build groups of transactions that include similar products or items. The idea behind the segmentation methods of exploratory MBA is the assumption that shopping baskets with similar product compositions imply purchase correlations between the grouped categories. For example, if a segment of market baskets contains beer, fizzy drinks and water with a higher probability than average, analysts assume a correlation between these categories. The reason behind this category combination might be the heavy weight of the beverages. It could induce the buyers to come by car and transport the cans or bottles in one single shopping trip. If the rows of Table 2.1 are grouped according to their similarity, one partition will comprise the equal baskets x_4, x_8, x_{10} and x_{14} with the items i_1, i_6 and i_8. Both methods – counting the pairwise co-occurrence and grouping according to the similarity of the market baskets – define correlations between the same items (see Section 2.2.1). Regarding the procedure of pairwise association analysis and some partitioning approaches, there is a coincidence between finding groups of similar market baskets and counting co-occurrences to uncover category correlations. For both techniques, the output is often a similarity (or distance) matrix which comprises for each pair of units a value representing their level of agreement. The difference between the methods results mainly from the processing of the distance matrices in subsequent procedures.

Calculating the pairwise associations of items is an early method of MBA. However, several modern MBA techniques still use a similarity matrix set-up (cf. Strehl and Ghosh 2000, Strehl and Ghosh 2003). Some algorithms of cluster analysis in particular extract groups of items or market baskets from the similarity matrices (see Section 2.2.3). Although the matrices become enormous if the number of involved items is high (see Section 2.2.1), Strehl and Ghosh (2003) have introduced an approach which takes 762 entities into account. To deal with this number of entities, powerful computing methods are needed.

According to Strehl and Ghosh (2000), an advantage of similarity matrices for grouping market baskets is so-called “feature reduction”. Once the matrix (also called the similarity space) is set up, each value represents the agreement of a single pair of entities. In contrast to a $N \times J$ data table, the dimensionality of the $J \times J$ similarity matrix is much smaller, since the number of transactions is usually higher than the number of offered items ($N \gg J$). The complexity measure of storage space decreases and the substeps that follow are a minor computational problem. Nevertheless, the information within the similarity space needs to be aggregated and extracted using further methods, since a manual inspection simply overburdens an analyst. As the similarity matrices in Tables A.1 to A.3 illustrate, it is a challenge even to identify the maximum values – although the small example presented includes only eight categories.

Visualization of Interdependencies in Similarity or Distance Matrices

To reveal the hidden information within the data, an important part of exploratory market basket analysis is the development of visualization techniques. This section includes some selected methods used on similarity matrices derived from pairwise association analysis. We start with short examples of multidimensional scaling (MDS) and hierarchical clustering. Both techniques are suitable for matrices of a

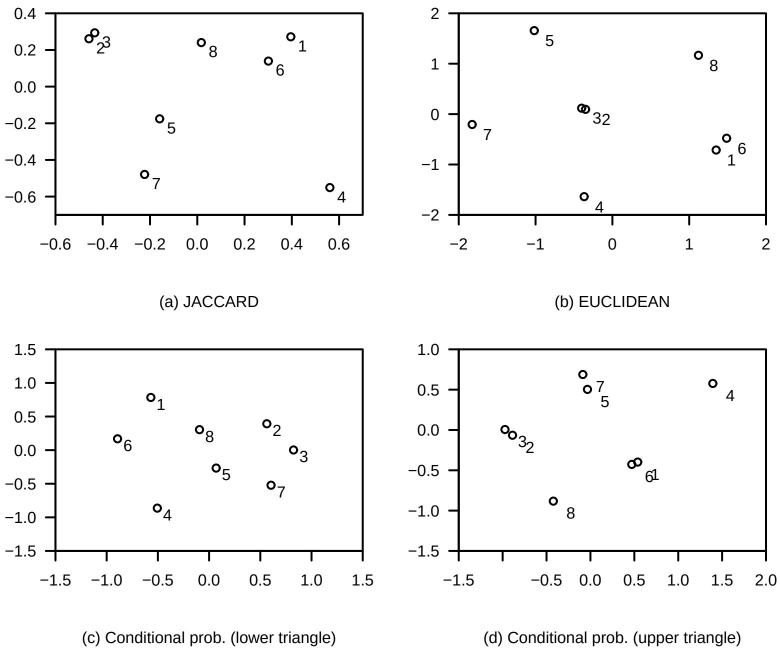

Figure 2.1: Visualization of four distance matrices with MDS

limited size. Concerning high-dimensional similarity matrices, Strehl and Ghosh's (2003) cluster visualization technique CLUSION is demonstrated as an example.

Non-Metric Multidimensional Scaling: Multidimensional scaling visualizes the similarity or dissimilarity between objects in a two- or three-dimensional plot. The iterative procedure determines for each pair of items an arrangement of representative points by using the values of the similarity or distance matrices. The arrangements of the points mediate the correlation among all entities in the feature space. For instance, points which are close to each other represent more similar items (i.e. products or categories) whereas points with a larger distance between them reflect the dissimilarity of the corresponding items. Figure 2.1 shows the visualization of the three distance matrices in the Appendix (see Section A.1.1, Tables A.1, A.2 and A.3). The lower two graphs are derived from the upper and lower triangle of the asymmetric matrices calculated with the conditional probability. Decker and Schimmelpfennig (2002) and Müller-Hagedorn (2005) have already shown the first and fourth plots slightly modified. The plot representing the MDS result of the Jaccard-distance matrix implies a purchase relation between item $j = 2$ and $j = 3$ or between the items $j = 1$, $j = 6$ and $j = 8$. The arrangement of the data points changes if the Euclidean distance or the conditional probability is used. Since the matrix derived from the probabilistic model is different below and above the diagonal, Figures 2.1(c) and 2.1(d) are also different. Notice that even for this small data sample, the three different distance measures lead to four different plots. This shows that the determined correlation between the items depends strongly on the selection of the distance measure.

Dendrograms from Hierarchical Clusteral Analysis: Hierarchical cluster analysis should group items into segments which are combined quite often in the market baskets. Some simple algorithms exist to merge similar entities. Usually the algorithms are separated into two steps (Backhaus et al. 2006):

1. The pairwise distances of the entities are calculated with an appropriate distance measure.
2. A fusion algorithm builds groups according to the distances between the entities.

With reference to the short data example in Table 2.1, the distance matrices are already known from the pairwise association analysis. The complete-, single-, average linkage alorithm or the Ward algorithm are typical fusion algorithms to aggregate the distance matrices. At the beginning, they consider each entity as a group (bottom-up) and merge the groups until all entities belong to one cluster. To find an appropriate grouping structure, the analyst considers the levels of fusion and decides at what point in time the best grouping is achieved (cf. Decker and Schimmelpfennig 2002). Some fusion algorithms can also group all entities in one big cluster and divide it into smaller clusters successively (top-down). All of the fusion algorithms are described extensively in the literature (Kaufman and Rousseeuw 2005, Backhaus et al. 2006). Figure A.1 represents the dendrograms built with the average linkage fusion algorithm if the distance values in Tables A.1, A.2 and A.3 are used. The branches of the dendrogram represent groups of similar items. Of course, it is not always clear where to cut off the branches and which items belong to a cluster. Nevertheless, some of the dendrograms represent the groups of items that were already recognized with the MDS.

CLUSION: The cluster visualization technique CLUSION introduced by Strehl and Ghosh (2003) visualizes groups of similar market baskets in high-dimensional data. It starts by defining a fixed number of clusters K. Strehl and Ghosh (2003) suggest a quality measure to approximate an appropriate value of K.[1] The graph partition algorithm embedded into CLUSION needs the similarity matrix as input and defines for each pairwise similarity value a cluster label. CLUSION itself reorders the cluster labels representing the (dis-)similarity of two entities of the matrix as long as the same labels are contiguous. For visualization purposes, a dot in a two-dimensional, rectangular grid represents the similarity value between two market baskets. Each dot takes on a gray-level shade. The shade ranges from white to black. Black corresponds to complete similarity and white to complete dissimilarity between two entities. The CLUSION algorithm groups similar entities into sub-squares along the diagonal of the grid. The average

[1]We return to the problem of determining K for our presented cluster approach in Section 4.2

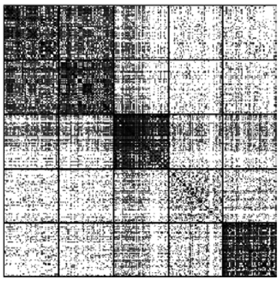

Figure 2.2: Example of the cluster visualization technique according to Strehl and Gosh (2002)

intensity of the gray-level of each square along the diagonal stands for the internal similarity of the entities of one cluster. Off-diagonal squares symbolize the separation level between the clusters. Figure 2.2 shows an example output of CLUSION.

After the reordering of the entities has been done, Figure 2.2 shows five different clusters separated by thin black lines. Beginning with the upper left corner of the diagonal, the first two clusters are not very distinct from each other since the off-diagonal rectangles are also dark-gray. In contrast, clusters three and five seem to be quite strong clusters. The light-gray shade of cluster four gives a clear indication that the averages of the included entities are relatively dissimilar. Taken together, the CLUSION plots do not only reflect the correlation between the clusters, but also give information about their quality.

The example has also shown that the visualization of the purchase interdependencies with graphical tools provides the observer with valuable information quickly. Hence, visualization techniques are a well-noted field of development in the data mining and KDD community and are often classified as "visual data mining". Since the approach introduced in Chapter 3 does not need visualization techniques to bring out useful results coercively, we abstain from giving a broad overview of existing techniques. For an introduction to visualization in data mining, Grinstein and Ward (2002) explain and compare some modern techniques (for additional applications and techniques see Fayyad, Grinstein and Wierse 2002).

Implications

The determination of pairwise purchase correlations with symmetric or asymmetric distance or similarity measures leads to two relevant issues for the following research. First, Section 2.2.1 implies that the choice of the distance measure affects the results of the techniques. Therefore, it is quite difficult to define the "right" distance measure as the suitability depends on the context of the problem. Because partition methods in particular rely on distance calculations, we have to expect varying solutions in the case of different measures. It is necessary to verify which distance measure brings out the most valuable and suitable outcome in the corresponding context. In regard to our approach, we address this issue in Section 3.1.4. Second, calculating a distance or similarity measure between each pair of entities is a complex computational problem since the number of similarity values increases exponentially with the number of items involved. Equation 2.10 and Equation 2.11 describe the correlation between the number of cells z in the matrices subject to the number of examined items J (or the number of baskets N) for a symmetric and an asymmetric measure. Calculating the full similarity or distance matrix for high-dimensional data can easily overburden the existing computational resources. Hence, the calculation of pairwise correlation coefficients seems to be suitable if retailers want to analyze purchase interdependencies on a higher item aggregation level. For example, examining correlations between classes of goods or categories could be done with these techniques (cf. Hamming 1950, Müller-Hagedorn 2005).

$$z_{symmetric} = \frac{J(J-1)}{2} \tag{2.10}$$

$$z_{asymmetric} = J^2 - J \tag{2.11}$$

Researchers recognized early that the calculation of full similarity or distance matrices limits the potential of MBA due to the resulting computational complexity. The growing data mass owing to, for example, the implementation of scanner technologies at retail cash registers, the increases in online purchases, the growing of assortments and the customer bases, has forced scientists to develop algorithms which can find purchase correlations without comparing all pairs of entities in a dataset. Taking into account only the statistically important item correlations or approximating a useful partition of the items boosts the efficiency of MBA algorithms.

Two principal kinds of MBA techniques are mainly applied to this task. First, association rule mining algorithms count the co-occurrence of items systematically within the transaction dataset. The corresponding algorithms are the topic of the following section. The second kind of technique comprises extended partitioning approaches. In addition to the already mentioned algorithms of hierarchical cluster analysis and CLUSION, partitioning cluster algorithms do not need a full similarity matrix to build similar groups of market baskets. Since we combine an algorithm of association rule mining and partitioning cluster analysis into one framework, the main method of both techniques is crucial to understanding the approach introduced in Chapter 3.

2.2.2 Association Rule Mining

For some time now, association rule mining (ARM) algorithms have been the most popular techniques for finding cross-category purchase co-incidences within binary transaction data (cf. Agrawal et al. 1993, Agrawal and Srikant 1994, Bayardo and Agrawal 1999). The objective of ARM is to identify all statistically important co-occurrences of items with efficient counting strategies. Let $J = \{j_1,\ j_2,\ j_3...\}$ be the set of items in dataset X_N. Each transaction x_n of the dataset contains a set of l-itemsets where l is the length of the itemset and defines the number of single items in the itemset. We name a subset of length l an l-subset. The *support* of an itemset A is defined as the relative share of the transactions in the dataset X_N which contain the itemset (see Equation 2.12).

$$supp(A) = \frac{|\{x_n \in X_N \mid A \subseteq x_n\}|}{|X_N|} \tag{2.12}$$

If the support value exceeds the user-defined minimal support value *minsupp*, the corresponding itemset is called a frequent itemset (FI) (Agrawal et al. 1993, Savasere, Omiecinski and Navathe 1995, Brin, Motwani, Ullman and Tsur 1997). For example, a frequent itemset $A = \{red\ wine,\ appetizers\}$ defines a correlation of statistically sufficient interest between the two categories since the included items co-occur often enough in the transactions of a dataset. According to Agrawal et al. (1993), an association rule is an expression such as "if item(set) A was chosen, then item(set) B was also bought". The symbol $A \rightarrow B$ with $A, B \subset J$ and $A \cap B = \varnothing$ describes the probability of the purchase of B, given the purchase of A. Equation 2.13 expresses the correlation between this conditional purchase probability and the support value:

$$conf(A \rightarrow B) = \frac{|\{x_n \in X_N \mid (A \cup B) \subseteq x_n\}|}{|\{x_n \in X_N \mid A \subseteq x_n\}|} = \frac{supp(A \cup B)}{supp(A)} = P(B \mid A) \tag{2.13}$$

ARM algorithms typically follow a two-step procedure (Brin, Motwani, Ullman and Tsur 1997, Hettich and Hippner 2001):

- In the first stage, the algorithm counts the frequent itemsets systematically over the dataset. This is frequent itemset mining.
- The rule-generation step builds and selects the association rules by calculating from all identified frequent itemsets the corresponding conditional probabilities of co-occurrence or a similar kind of correlation measure (cf. Mannila 1997).

The mining step is the computationally intensive part of ARM. Hence, computer scientists concentrate on the enhancement of the identification of frequent itemsets. The community has developed a large number of varying algorithms within the last few years. The algorithms differ mainly in the techniques used to decrease execution time and to reduce memory space. Compared to the mining step, building rules from the identified frequent itemsets is not a significant calculatory challenge. Nevertheless, some work points to the issue of generating and selecting only the statistically meaningful itemsets with different rule-generation techniques and measures of interest. In the next sections, we describe the concept of these algorithms in brief. Some of these algorithms and their variations are already widely used due to their implementation in several free and commercial data-mining software packages.

Frequent Itemset Mining

One of the most popular algorithms to efficiently scan large datasets for frequent itemsets is Agrawal et al.'s (1993) APRIORI algorithm. It counts the l-itemsets for each l-pass over the dataset and keeps those that exceed the minimum support. As an example, we mine all frequent itemsets with a length l of 1 and a minimum support level of 0.25 in the dataset shown in Table 2.1. Except for item j_4, every item occurs at least four times in the dataset. Hence, seven of the eight $l = 1$-items are marked as frequent. After each pass, the APRIORI algorithm includes the *join step* which combines the potential $l+1$-itemsets from all frequent items or itemsets. With regard to the short mining example, the join-step would combine 21 itemsets with a length of two items out of the seven itemsets. Again, the minimum support values from these itemsets are calculated and only seven itemsets are marked as frequent (see Table A.4). For $l \geq 2$, the APRIORI algorithm includes the *downward-closed*-implication to reduce the number of itemsets in the so-called *pruning step*. This implication considers that the support value of an $l+1$-itemset cannot exceed the support value of its l-subsets. Hence, after the join-step has composed the potential $l = 3$-itemsets from the $l = 2$-itemsets, the pruning step would drop all potential $l = 3$-itemsets which include infrequent $l = 2$-itemsets. Due to the downward-closed-implication, these itemsets cannot reach the minimum support value. Since non-frequent itemsets need not be considered in subsequent mining passes, the pruning step increases the speed of the APRIORI algorithm. For example, consider the 2-itemsets in Table A.4 of the Appendix: the join step would also create the potential 3-itemset $\{j_1, j_5, j_8\}$ from the itemsets $\{j_1, j_8\}$ and $\{j_5, j_8\}$. Since the 2-subset $\{j_1, j_5\}$ is not a frequent itemset, itemset $\{j_1, j_5, j_8\}$ can be dropped. Due to the downward-closed-implication, the APRIORI algorithm outperforms algorithms such as the AIS, which calculates the support value for all combinations of subsets (cf. Agrawal et al. 1993).

To reduce the amount of data that has to be scanned for frequent itemsets, some variations of the APRIORI algorithm have been developed. The *AprioriTID* replaces the transactions of the original dataset with compositions of found frequent itemsets (Agrawal, Mannila, Toivonen and Verkamo 1996). If one or more frequent itemsets cannot express a transaction, the transaction will be deleted. For higher passes (i.e. a higher l), the modified dataset becomes much smaller than the original dataset since many transactions do not contain long frequent itemsets. Dropping the transactions which do not include any of the found itemsets reduces memory space and search time in subsequent repetitions. This will not usually happen for small l since more itemsets are needed to express the transactions. In this case, the corresponding transactions consume more memory space and the modified dataset becomes larger. Since the performance of the APRIORI algorithm is superior to the AprioriTID modification for small l, the authors introduced the *Apriori Hybrid* algorithm (Agrawal and Srikant 1994). It combines the advantages of both algorithms by replacing the APRIORI with AprioriTID for higher l. The point at which the change from one to the other algorithm is made can be determined according to the following premises: first, the modified dataset is small enough to fit in the RAM of the system. This permits the subsequent counting steps to be done with reduced I/O-processes. Second, the modified dataset of pass $l+1$ is smaller than the one at length l. Other rules are also possible.

In addition to the pruning steps involved in the downward-closed implication, computer scientists have introduced some further methods to increase the performance of the existing algorithms. The main deficiency of the APRIORI algorithm and its variations is the need for long and repeated search passes over the data. Making several passes reduces performance due to the burdening I/O-processes. If a frequent itemset of length l_{max} exists, l_{max} search repetitions have to be made. Some algorithms implement one or more of the following general techniques to reduce long search passes or prevent frequent search passes. These methods are not bound exclusively to association rule mining but are also used in many other algorithms (e.g. those of cluster analysis) to increase computational efficiency:

1. Parallel computing

2. Dataset splitting

3. Sampling

Parallel computing means that the algorithms increase the performance of the itemset mining through a simultaneous calculation of subproblems. The dynamic itemset counting (DIC) algorithm of Brin, Motwani, Ullman and Tsur (1997) does not wait to count the $j+1$ itemsets until the l-pass has been finished and all l-itemsets have been mined. Instead, the algorithm determines the l, $l+1$, $l+2 \ldots l_{max}$-itemsets simultaneously. This is done by separating the data matrix into predefined intervals of equal size. If the end of an interval has been reached, the algorithm starts identifying the $l+1$ itemsets as well, although the frequent itemsets of length l are not yet identified for the whole dataset. For example, four intervals of 2,500 transactions divide a dataset with 10,000 transactions. When the algorithm has finished counting the 1-itemsets in the first 2,500 transactions, it starts recording the 2-itemsets additionally. After transaction no. 5,000 has been passed, the 3-itemsets will be added as well, and so on. During the counting step, each itemset is marked as an estimated frequent itemset, a confirmed frequent itemset, an estimated non-frequent itemset, or a confirmed non-frequent itemset. If a first full pass over the entire dataset has been done, estimated frequent and non-frequent itemsets can be confirmed as frequent or non-frequent itemsets. Moreover, the downward-closed-implication is used to drop the estimated frequent and non-frequent $l+1$-itemsets as well. If the estimated itemsets contain deleted subsets, they can be excluded too. This method reduces the number of itemsets considered during the counting step significantly. The gain in performance of the DIC algorithm decreases if the distribution of the frequent itemsets over the dataset is deficient. For example, estimated frequent itemsets become confirmed as non-frequent itemsets late when they occur mainly in the last intervals of the dataset. Therefore, the authors recommend a randomization of the dataset by an artificial reordering of the transactions.

Instead of identifying the itemsets successively, some algorithms process different parts of the data at the same time (cf. Cheung, Ng, Fu and Fu 1996). This can be done by passing the fragments to several parallel working hardware platforms. These nodes calculate local frequent itemsets using the APRIORI or another algorithm and summarize their outcome for further inspection (Sakthi, Hemalatha and Bhuvaneswaran 2008). Usually, the nodes are arranged in a grid and are able to communicate their locally found frequent itemsets to each other. By working in parallel, such environments can mine much larger datasets. Zaki (1999) presents an overview of parallel and distributed algorithms for association rule mining.

Dataset splitting is another technique to reduce the load on system resources. It means cutting the data into smaller pieces. Computer systems have much less trouble when processing smaller data fragments successively (cf. Jamshaid, Jalil, Khiyal and Saeed 2007). For example, the partition algorithm of Savasere et al. (1995) searches within non-overlapping subsets of the data for local frequent itemsets. Each of the parts fits in the RAM of the computer system where it can be processed more rapidly. The sum of all “local” frequent itemsets mined in the dataset partitions builds the set of potential “global” frequent itemsets. These global frequent itemsets are confirmed in the second phase of the algorithm. Since the sum of all local frequent itemsets includes the confirmed ones, the algorithm simply has to identify the false-positives. This course of action needs only two passes over the dataset and reduces the I/O processes significantly. Nevertheless, similarly to the DIC algorithm, a deficient, non-equal distribution of frequent itemsets over the transactions of the dataset reduces the algorithm’s advantages. Again, a previous step should randomly arrange the transactions in the dataset. The random partition algorithm of Razi, Kausar, Khiyal and Saeed (2008) directly integrates the randomization into the algorithm and extenuates periodic imbalances in the distribution of transactions.

For very large databases, Zaki, Parthasarathy, Li and Ogihara (1997) have presented a mining algorithm based on **sampling**. Instead of identifying the frequent itemsets in the complete database, the idea is to extract representative samples from X_N with fewer transactions. The algorithm does not scan the complete dataset, but rather counts the co-occurrence of category correlations within the sample and assumes that the support level of the found itemsets is characteristic for the whole data. Due to the smaller size of the samples, the mining process is much faster (cf. Luo and Chung 2004). Toivonen (1996) combines sampling and dataset splitting into one algorithm. The extracted samples with a predefined size fit into the main memory to assure high effectiveness, similar to the partition algorithm. Nevertheless, the sampling of datasets always means a trade-off between accuracy and efficiency. Small samples offer an efficient mining procedure but increase the chances of defining false-positive frequent itemsets. Therefore, association mining algorithms using sampling methods often try to determine the right sample size with precedent techniques. The sampling algorithm of Zaki, Parthasarathy, Li and Ogihara (1997) implements a sequential random sampling method to extract the transaction set from X_N (see also Vitter 1987). After determining the sample size and mining the frequent itemsets in the sample, the second objective is to assure that the sample and the found frequent itemsets actually reflect the condition in the overall dataset. The sampling algorithm assumes that the probability of an itemset A occurring in a transaction x_n is determined by a binominal random variable R_n ($R_n = 1$ if $A \subset x_n$; otherwise $R_n = 0$). The probability that the itemset A occurs in a basket x_n equals the support value $supp(A)$ of the itemset. The binomial distribution approximates the probability distribution of R_n since there are N trials with the option of $P(R_n = 1) = supp(A)$. To verify the accuracy between the found frequent itemsets in the sample and those found in the whole dataset, the Chernoff bounds are calculated (Hagerup and Rüb 1990). Zaki, Parthasarathy, Li and Ogihara (1997) present the evaluation of some known datasets with the sampling algorithm and show that frequent itemset mining in smaller samples produces itemsets which are quite similar to those that can be found with a complete mining sequence. Instead of the Chernoff bounds, Zhao, Zhang and Zhang (2006) recommend so-called Hybrid bounds.

Independent of the usage of the varying mining algorithms, the algorithms identify for each counting pass l a number of l-frequent itemsets and summarize them in the H_l solution sets. With reference to the data example in Table 2.1, Table A.4 of the Appendix contains all H_l with mined frequent itemsets subject to a minimum support threshold of 0.25.

Finding only frequent itemsets of the maximal length l_{max} is a very common but complex problem of ARM (cf. Yang 2004). A frequent itemset is denoted as *maximal* “if it is not a proper subset of any other frequent itemset” (Zaki, Parthasarathy, Ogihara and Li 1997). Since every subset of a maximal frequent itemset (maxFI) is also frequent, the number of maximal frequent itemsets is usually smaller than the total number of frequent itemsets in a dataset. Hence, knowing the maxFI can lower the computational effort for subsequent mining steps. Moreover, the informative value of maximal frequent itemsets is often higher in certain situations. Take for instance the three frequent subsets $\{cereal, milk\}$, $\{cereal, corn\ flakes\}$ and $\{corn\ flakes, milk\}$ of the maxFI $\{cereal, corn\ flakes, milk\}$. The long pattern illustrates compactly and easily the correlation between the ingredients of a breakfast without having to examine all the frequent subsets. Therefore, algorithms which mine long patterns or maximal frequent itemsets efficiently are of high interest for many ARM applications (e.g., Bayardo 1998, Zaki 1999, Yang 2004, Gouda and Zaki 2005). The approach presented in Chapter 3 also requires the identification of maximal frequent itemsets in a later step (cf. Section 3.3).

Rule Generation and Measures of Interest

The second step of ARM comprises the generation of association rules. Agrawal et al.'s (1993) rule-generation algorithm uses the confidence value (see Equation 2.13) to build the association rules from the mined frequent itemsets.

The solution set H_l comprises the found frequent itemsets h_l of length l. From the solution set H_l the rule generation algorithm combines all association rules which exceed a user-defined minimum confidence value, called the *minconf*. The algorithm uses the possibility to express the confidence value with the support values as described in Equation 2.14 (cf. Hettich and Hippner 2001):

$$conf[(h_l - A) \rightarrow A] = \frac{supp(h_l)}{supp(h_l - A)} \tag{2.14}$$

Since the support values have been calculated in the first mining step, a further exploration of the data pool is not necessary. The challenge of the rule generation is to find each rule $A \rightarrow h_l - A$ with $l > 1$ and $A \subset h_l$ exceeding the minimum confidence value. Similar to the mining step, the identification of the rules is efficiently done by considering two precepts: first, if the frequent itemset A exceeds the minimum support, the rule $A \rightarrow h_l - A$ exhibits this requirement too. Second, if a frequent itemset A' with $A' \subset A$ exists, the rule $A' \rightarrow h_l - A'$ does not exceed the minimum support of the rule $A \rightarrow h_l - A$. For the reverse case: if the rule $A \rightarrow h_l - A$ does not hold the minimum support condition, all rules $A' \rightarrow h_l - A'$ do not exceed the *minconf* and can be ignored for the rule generation as well. Considering these two postulations speeds up the combinatorial building of the rules (cf. Agrawal et al. 1993). Table A.5 of the Appendix lists all association rules which can be combined from the frequent itemsets of Table A.4. Similar to pairwise association analysis, the counting algorithm reveals the purchase correlations between $i_1,\ i_6,\ i_8$ and $i_5,\ i_7$.

Depending on the height of the minimum support or the height of the minimum confidence as well as the structure of the data, ARM often leads to an unmanageable number of frequent itemsets and association rules in real world situations. It is difficult for retailers to define in the mass of category correlations the ones which can be transferred to practical decisions for category management or promotion. In this case, analysts should conduct a subsequent filtering step to extract only the meaningful correlations of the sets (cf. Hettich and Hippner 2001, Hahsler, Hornik and Reutterer 2005). We discuss three approaches to determining the most interesting associations in the subsequent mining steps:

1. Sorting frequent itemsets with database queries

2. Grouping of frequent itemsets

3. Filtering frequent itemsets

First of all, analysts usually sort and reduce the number of mined itemset correlations using simple database queries, e.g. considering only the 100 rules with the highest confidence value. Moreover, analysts can also extract or drop correlations that include specific items. Using these less sophisticated methods compresses the information easily but increases the danger of ignoring important correlations.

A less common approach is to group association rules or frequent itemsets pursuant to their similarity (cf. Toivonen, Klemettinen, Ronkainen et al. 1995, Gupta, Strehl and Gosh 1999). Since the lists of generated association rules or frequent itemsets are often sorted according to the support or confidence values and not according to the included items, inspecting the lists is quite labor-intensive. Furthermore, if long maximal frequent itemsets are identified, the output lists include the quite similar subsets and rules containing these subsets. Sometimes it helps to group the found association of a specific customer segment into subgroups. In this way, the decision maker can easily inspect the segments of associations which catch his attention the most. The grouping of itemsets can reveal areas of similar itemsets within the segment-specific itemsets and increases the retailer's understanding of the corresponding customer segment. For example, in Chapter 4 we present a group of itemsets with hard-alcoholic beverages such as $\{brandy, \ whiskey\}$ in the segment of typical wine buyers. Chapter 4 shows the visualization of such partitioned frequent itemsets with dendrograms too.

In addition to data queries and association grouping, many measures of interest have been developed in recent decades (cf. Hettich and Hippner 2001, Hahsler, Hornik and Reutterer 2005). Filter measures evaluate the found correlations and suggest additional information about their statistical importance. Different assets and drawbacks characterize the measures. A universal filter measure does not exist and a successful application of the measure always depends on the context of the analysis. The following explanations introduce some common measures of interest which are often implemented in data mining applications.

The two most frequently used measures have been introduced already: the support and the confidence measures. A closer look at the characteristics of the confidence value will reveal some deficient characteristics. Given the confidence of two frequent itemsets A, B (see Equation 2.13), it neglects the support value of the counterpart itemset. This becomes a problem for correlations between frequent itemsets with very different support values, e.g. $\{champagne\}$ and $\{milk\}$. Due to a higher demand for dairy products, milk's support is usually higher than champagne's. A rule generation algorithm using the confidence value could identify the two rules $\{champagne\} \rightarrow \{milk\}$ and $\{milk\} \rightarrow \{champagne\}$. The confidence of the first rule would be much higher since the conditional probability of buying milk once the customer has chosen a bottle of champagne exceeds the second possibility by trend (i.e. $P(milk \mid champagne) > P(champagne \mid milk)$). Hence, the rule generation algorithm based on the confidence can produce rules, even though the confidence of the reverse rule does not exceed the *minconf*-threshold and the purchase correlation is weak. Here, we call such an itemset a weakly-related cross-support itemset (cf. Hui et al. 2006). The mining of weakly-related cross-support itemsets is a larger problem in transaction data. As we discussed in Chapter 1, typical retail assortments can be separated into the HFC and LFC (i.e. the long-tail). The different purchase frequencies of the corresponding products support the accrual of cross-support itemsets. To avoid such weakly-related cross-support itemsets, we introduce the all-confidence measure in Section 3.3.1 (cf. Omiecinski 2003, Hahsler, Grün and Hornik 2005).

The right-most column in Table A.5 of the Appendix shows a common measure of interest called *lift* (cf. Brin, Motwani, Ullman and Tsur 1997). Since the confidence measure neglects the support value of the consequence in the association rule, the lift value includes the occurrence of the right-hand side itemset as well (see Equation 2.15):

$$lift(A \rightarrow B) = \frac{supp(A \cup B)}{supp(A)supp(B)} \tag{2.15}$$

Assuming the purchase frequency of itemset A, B is stochastically independent, the probability of co-occurrence equals the denominator of Equation 2.15. Hence, if this value becomes low in relation to itemsets' common purchase probability, the lift is higher than 1 and describes a complementary correlation of the items. A value below 1 would imply substitutes, and a 1 value no correlation. In contrast to the confidence value, lift is a symmetrical measure (i.e. $lift(A \rightarrow B) = lift(B \rightarrow A)$).

Although the lift balances the deficiencies of the confidence value, Hahsler, Hornik and Reutterer (2005) have shown that it can over-evaluate rules when a low minimum support is chosen. The authors defined a small minimum support threshold ($minsupp = 0.001$) and analyzed a synthetic dataset which did not include any correlations. The results showed that the lift marked 130 rules falsely as very interesting ($lift \gg 1$) – even though no correlations were present in the data. This happens if the denominator of Equation 2.15 becomes nearly zero for two rarely appearing frequent itemsets, and the chance of co-occurrence represented by the numerator is relatively high. To avoid marking these spurious rules, Hahsler, Hornik and Reutterer (2005) introduced the *hyperlift* measure. Since the choice decision of buying independent items equals a Bernoulli trial, a hypergeometric distribution can approximate the chance of purchasing an item. In Equation 2.15, the denominator is interpreted as the expectation value of two independent items. The authors replace the expectation value with a quantile of the hypergeometric distribution to reduce inadequate high lift values. After the authors evaluated the found rules of the synthetic datasets with the hyperlift, only two rules were still marked as interesting.

Many other ratio figures exist which try to balance the specific deficiencies of the previously described measures. For example, due to the quotient, the confidence or the lift of a rule can become high even though the support value of the frequent itemsets involved is low (cf. Hettich and Hippner 2001). Instead of a relative expression, the *p-s-function* uses a difference as described in Equation 2.16 (Piatetsky-Shapiro 1991).

$$p - s(A \Rightarrow B) = supp(A \cup B) - supp(A)supp(B) \tag{2.16}$$

According to the p-s-value, a rule becomes more interesting the bigger the difference is between statistical independence (right-hand side term of Equation 2.16) and the probability of co-occurrence of the frequent itemsets (left-hand side term). In a similar way, the *gain measure* also uses a difference (cf. Fukuda, Morimoto, Morishita and Tokuyama 1996). Brin, Motwani and Silverstein's (1997) *conviction* measure tries to enhance the outcome of the lift value. Some conditional expressions are always true but the lift of the corresponding rules does not reflect this properly. For example, in census data, 5% of a country's population are veterans and 90% are older than five years. According to Equation 2.15, the lift value of the rule $\{veteran\} \rightarrow \{older\ than\ five\ years\}$ would be $0.05/0.05 * 0.9 = 1.1$ (Brin, Motwani, Ullman and Tsur 1997). This value barely exceeds the 1 value and describes only a weak correlation. In contrast, the conviction value becomes infinite (∞) showing that the conditional characteristic between the age of a person and his wartime experiences is highly correlated (cf. Hettich and Hippner 2001). Silverstein, Brin and Motwani (1998) implement the χ^2-test to reach a similar effect with their filter measure.

The choice of a specific filter measure always depends on the objectives of the analysis and the characteristics of the data. Scanning transaction datasets for specific item correlations, for example, is different to analyzing census data, which is by nature more dense (cf. Bayardo and Agrawal 1999). Some analysts recommend combining several measures to improve the outcome of the association rule mining analysis. This is a good recommendation as long as the analyst knows the characteristics of each measure used to extract the meaningful rules or frequent itemsets. Concerning the method of our data-driven approach in the next chapter, the support, confidence, all-confidence and to some extent the lift-value are of interest. For a detailed overview of the above-mentioned filter measures and other additional ones, we refer to the secondary literature, e.g. Hilderman and Hamilton (2001) who introduce a wide variety of different measures of interest.

Taxonomies and Disassociation Rules

To enhance the informative value of the found frequent itemsets or rules, the analyst can manipulate or extend the data. Looking at the taxonomy of the item arrangements reduces the dimensionality of the binary datasets. According to Srikant and Agrawal (1996), a taxonomy represents the systematic order of items in different levels of detail. With reference to market basket analysis, the levels of aggregation illustrated in Table 2.2 are suitable for building a taxonomy for a typical retail assortment. Some algorithms are able to take the taxonomy into account during the mining process (cf. Srikant and Agrawal 1996). Let us assume for example that the dairy category of a retailer includes milk, curd and cream. A typical rule which considers this taxonomy is “if $\{Bakerman's\ Best\ flour\}$ was bought, $\{dairy\ categories\}$ were bought also”. The algorithm includes one of the three items for the right-hand side of the rule during the mining process. Such mining algorithms can reveal valuable associations, particularly for sparse data with fewer co-occurrences of items.

Instead of finding rules between items usually bought in combination, it might also be interesting to determine statements such as “A does not co-occur with B” (cf. Hettich and Hippner 2001, Berry and Linoff 2004). This can be done by inverting items, i.e. making 1 values define the absence of an item instead of its occurrence. Before inverting several columns of the data matrix, the analyst should have a strong assumption about the effect of this technique. Instead of sparse data, the transactions become very dense when the analyst inverts too many attributes. As a result the mining step might find many useless rules as well.

Implications

Association rule mining is strongly connected to exploratory market basket analysis since many of the algorithms were early applied to scan transaction datasets for specific item correlations. In addition to the analysis of retail data, ARM can also be used for searching specific click-streams within the collected logfiles of online retailers. Reproducing online shoppers’ paths through the web-content might reveal an unknown interest in certain offers.

The popularity of ARM results from its easily interpretable output, gained by efficient, well-distributed algorithms. Even people with a weak data-mining background seem able to understand the resulting rules intuitively. In addition, users can weight the statistical importance of the found rules according to the calculated measures of interest such as support, confidence and lift. Implementing support thresholds combined with efficient counting strategies, instead of defining a full similarity matrix for each entity, makes these algorithms an important tool for analyzing the customer’s purchasing behavior.

Despite its advantages and widespread use in practice, however, some challenges and issues of ARM are important for the method of our approach (cf. Chapter 3):

- *Anonymity:* Association mining algorithms are usually applied to the aggregated data matrices that include the transactions of an entire clientele (cf. Hettich and Hippner 2001, Berry and Linoff 2004). Due to the missing connection to the originators of each transaction, the identified rules reflect the average buying behavior of a retailer's customer base. In real world situations, retailers often want to communicate with specific customers or customer groups on a more personal level. Even when registration or online programs are present, most ARM techniques are not able to meet this objective.

- *Relevance of the output:* Although a variety of measures of interest exists, in practice ARM algorithms can still produce a high number of frequent itemsets or rules. Since many of the found correlations are redundant, trivial or just meaningless, the analyst has to take care to mine only relevant correlations. This depends on the user's knowledge of the characteristics of the rule generation algorithms and measures of interest which have a strong effect on the results. In addition, the informative value of the rules is limited since ARM is based on binary data (cf. Section 2.1). It is very likely that the monetary value of frequent itemsets or association rules is of crucial interest for the majority of sellers.

- *Computational issues:* Since the customer base and the assortments of today's retailers are still expanding (cf. Section 2.1.2), the efficiency of ARM algorithms has to increase as well. To reduce the computational effort, sampling, partitioning or parallel computing should be used in conjunction with the mining step.

Although some user interaction is still required to get valuable output, ARM approximates an automated method of finding interesting patterns within large collections of data. Solving the above-mentioned challenges could increase the practicability of ARM in many target marketing environments. Hence, our approach presented in Chapter 3 helps to improve the applicability of ARM when taking these issues into account.

2.2.3 Partitioning Cluster Analysis

In addition to association rule mining, algorithms from partitioning cluster analysis also play a major role in exploratory market basket analysis. In general, the objective of cluster analysis is to build groups of entities such that entities in the same cluster are as similar to each other as possible and entities in different clusters are as different as possible (cf. Aldenderfer and Blashfield 1984, Strehl 2002). Transferred to the task of MBA, the objective is to build groups with similar transactions (cf. Ordonez 2003).

Market baskets with a comparable item composition imply a purchase correlation between the included items, as discussed in Section 2.2.1. It is assumed that the households which originated the transactions show a similar purchasing behavior and can be targeted with more appropriate offers.
Partitioning cluster analysis is a wide research field with algorithms differing according to a number of characteristics. In this thesis, we refer to selected algorithms which have been used to group binary transactions successfully and which affect the methodology of our target marketing approach as explained in Chapter 3 (cf. Dolnicar, Leisch, Steiner and Weingessel 1998). For a broader introduction to partitioning cluster analysis and its common algorithms, we refer to the extended standard literature of the research community (e.g. Han and Kamber 2000, Duda, Hart and Stork 2001, Kaufman and Rousseeuw 2005). Cluster analysis is commonly divided into hierarchical and partitioning algorithms, among others (cf. Kaufman and Rousseeuw 2005, Backhaus et al. 2006). Hierarchical cluster algorithms depend on calculating the similarity matrices. Hence, we already introduced their methodology in connection with the visualization of matrices gained from pairwise association (cf. Section 2.2.1). In contrast to hierarchical cluster algorithms, the iterative procedure of partitioning cluster algorithms avoids the computational complexity of building all pairwise distance values. The partitioning algorithms approximate an optimal grouping of the entities with a stepwise reordering of the objects according to a target function, such that the sum of distances between the entities of a cluster becomes minimal. Popular partition algorithms for market basket analysis are the algorithms of K-centroid cluster analysis (KCCA) with K expressing the number of groups. In contrast to hierarchical cluster analysis, the analyst has to predefine the K value with statistic or heuristic methods *before* the procedure starts. Partitioning approaches have been recently adapted to build segments of homogeneous transactions even in very high-dimensional datasets with numerous items. Some of these algorithms can even be applied to the article level. The following sections introduce a collection of partitioning techniques. Instead of giving a complete explanation, the main objective here is to describe the general modules of the algorithms. For more details concerning each algorithm, the corresponding literature is cited.

Algorithms of K-Centroid Cluster Analysis

Considering a market basket vector as denoted in Section 2.1.1, each basket defines a point in the J-dimensional feature space. The objective of cluster analysis is to find K homogeneous market basket classes $C_K = \{c_1, c_2, ...c_K\}$ which are heterogeneous among each other (cf. Gordon 1981). Regarding algorithms of K-centroid cluster analysis, the typical iterative procedure starts with a random selection of K points P_K (Leisch 2006). These points are called centroids or centers $p_k = \{w_1, w_2, ..., w_J\}$ with every value or weight $w \in \mathbb{R}$ referring to a category j. In the subsequent iterative steps, the generalized KCCA algorithm tries to find the centroids p_k for every transaction vector x_n and a predefined number of partitions K in such a way that the average distance D of each entity to the closest centroid is minimal (cf. Bock 1999). The distance between each entity (i.e. a market basket) x_n and the centroid p_k can be calculated with an optional distance measure d that the user considers to be suitable in the corresponding context (cf. Section 2.2.1). During the iterative process the entities are reordered as long as a defined criterion to abort the procedure is fulfilled (e.g. the number of maximal iterations) or the arrangement of the entities does not change anymore. After the procedure has stopped, the sum of entities which are assigned to one specific p_k builds a specific cluster k. The generalized target function of KCCA is denoted by Equation 2.17 (Leisch 2006):

$$D(X_N, P_K) = \frac{1}{N}\sum_{n=1}^{N} d(x_n, p(x_n)) \rightarrow \min_{C_K} \tag{2.17}$$

One of the most popular algorithms of K-centroid cluster analysis for solving the target function in Equation 2.17 is the standard K-means algorithm (cf. McQueen 1967, Hartigan and Wong 1979, Kaufman and Rousseeuw 2005). The following steps summarize its iterative procedure:

1. Choose a random initialization of P_K data points.
2. Assign each transaction vector $x_n \in X_N$ to the closest centroid p_k with respect to the distance measure $d(.)$.
3. Hold $p(x_n)$ fixed and determine the new centroids according to the following equation: $p_k := \underset{p \in P}{argmin} \sum_{n: p(x_n) = p_k} d(x_n, p)$.
4. Repeat Steps 2 and 3 until the system converges or the predefined number of iterations has been reached.

When speaking about K-means, $d(.)$ denotes the Euclidean distance (cf. Section 2.2.1). Since K-means was one of the first algorithms used to partition larger data matrices, it was applied early to group transaction data (cf. Berry and Linoff 2004). Referring to the individual steps of the described cluster process, some aspects affect the partitioning of transaction datasets to a significant extent.

The updating of the centroids in Step 3 of the iterative procedure leads to an agreement between the centroids p_k and the partition's class means of one-observations after the optimal segmentation has been reached (Bock 1999, Reutterer, Mild, Natter et al. 2006). Since each value of the centroid equals the class means of the corresponding item in the cluster, we can consider this value to be the categories' probability of occurrence in an average market basket. Hence, the resulting centroid vectors p_k act as market basket prototypes with the higher purchase probabilities implying a correlation between the corresponding categories. The right-hand side of Table 2.5 shows a centroid vector of a segment comprising four different categories. A typical customer assigned to this centroid (i.e. prototypical market basket) would buy red wine with a probability of 74 percent. The other higher probability values of white wines and appetizers imply a purchase correlation between these alcoholic beverages in the cluster. This explains the good fit of KCCA to exploratory MBA if the updating of the centroids comprises the cluster-wise means according to Step 3.

Usually, the majority of recorded market baskets include just a few items from a retailer's wide product range. This means that the transaction vectors consist of many zeros and fewer ones. The analyst has to expect a sparse data matrix X_N (cf. Section 2.1.2). As discussed in Section 2.2.1, the Jaccard distance seems to be more appropriate than the Euclidean distance in this case since it gives more weight to the co-occurrence of entities. Hence, some authors have introduced a vector version of the Jaccard distance as shown in Equation 2.18, with $x^\Gamma c$ equaling the scalar product (cf. Schnedlitz et al. 2001, Strehl and Ghosh 2003, Decker 2005):

$$d(x_n, p(x_n)) = 1 - \frac{x^\Gamma c}{|x| + |c| - x^\Gamma c} \tag{2.18}$$

Regarding the update process of the centroids, convergence of the algorithm is only guaranteed if canonical centroids are implemented in Step 2 of the iterative process. In addition to the much higher computational complexity of calculating canonical centroids, these centroids are binary coded. In contrast to prototypes representing class means, binary j-values would not provide analysts with the preferred category choice probabilities (cf. Reutterer et al. 2006, Leisch 2006).

For a practical example, take Table 2.5, which compares a binary centroid vector on the left side with an expectation-based centroid vector on the right side. Both centroids define the co-occurrence of three categories in a fictitious segment. The centroid vector on the left side would provide retailers with the information that segment members combine red with white wines and appetizers. The right centroid vector represents the purchase probabilities of the items. This information is much more precise compared to the binary centroid value which simply determines the probable existence of the corresponding category in the segment.

item	p_k
red wine	1
white wine	1
appetizers	1
chocolate	0

item	p_k
red wine	0.73
white wine	0.54
appetizers	0.53
chocolate	0

Table 2.5: Example of a binary centroid (left) and an expectation-based centroid (right)

Compared to the centroid vectors defined by the K-means algorithm, the left side of Table 2.5 equals the output of the partitioning around medoids (PAM) algorithm described by Kaufman and Rousseeuw

(2005). In contrast to K-means, the algorithm tries to find for the K groups a set of existing entities called medoids x_n^{medoid} in such a way that the average distance of each transaction to the chosen medoids is minimal. The result of the PAM algorithm is a set of K binary transaction vectors. Each one can be considered as an actually existing, typical market basket of the corresponding segment. Instead of purchase probabilities, the dichotomous value of each category defines whether or not the category is usually bought by the average customer of the segment. Although this might be a suitable outcome in some instances, the expection-based centroid vectors seem to be more practicable. A further problem is that PAM also requires the computationally complex calculation of a full similarity or distance matrix (cf. Balka 2005, Kaufman and Rousseeuw 2005).

Iterative algorithms of K-centroid cluster analysis usually do not define an optimal segmentation. The calculation of all possible arrangements of the entities would overburden even today's computational capabilities (cf. Aldenderfer and Blashfield 1984). The partition often represents a local optimum and not the global optimal solution of the objective function described in Equation 2.17. Hence, a partition depends heavily on the initialization of the algorithm. Several repetitions of the procedure with different starting partitions reduce the risk of finding a weak local optimum (cf. Hornik 2005*a*). The different partitions found with the same algorithm can be bundled into a cluster ensemble for further inspection (cf. Hornik 2005*b*). In addition to re-clustering, Hand and Krzanowski (2005) recommend moving some data points randomly to a centroid other than the closest one during the iterative process. By disturbing the normal scheme, the algorithm is stimulated to break out of the rut of a probable non-optimal segmentation. Finding a more appropriate initialization of a KCCA algorithm can also be done with a preliminary hierarchical cluster analysis (cf. Hand and Krzanowski 2005).

These issues show that the outcome of partitioning depends on various characteristics of the algorithm. To achieve a valuable segmentation, the analyst has to keep them in mind. Besides the selection of the distance measure and the initialization, the updating of the centroids affects the results of the clustering significantly. The next section deals with partitioning algorithms which differ mainly in the determination of the centers during the iterative process.

Online Algorithms and Competitive Learning

In the iterative process of the generalized K-means algorithm, the new centroids are calculated after all entities of a cluster have been rearranged. In contrast, Hartigan and Wong's (1979) K-means exchange algorithm updates the centroids every time the distance to a randomly-chosen, single entity has been calculated. This modification introduces the conceptual difference between online and offline (or "batch") algorithms. The difference is important for the adaptation of machine learning and neural network algorithms to exploratory MBA. Offline algorithms, such as generalized K-means, need the complete dataset to calculate the point-centroid mapping. When the data becomes very large, it presents a computationally challenging task. Online algorithms process only one entity per round and adjust the centroid mapping according to this single input. In other words, the task of calculating the centroid-point mapping is separated into many smaller sub-processes. Specifically, competitive learning and exchange algorithms group the entities according to the sequential consideration of the single inputs (cf. Fisher 1987, Dolnicar et al. 1998).

From a retailer's point of view, separating between batch and online algorithms becomes quite interesting for clustering shopping baskets. Since the market basket vectors appear constantly at the cash register, a connection to a processing online algorithm could assign the recorded transaction directly to the appropriate cluster. In contrast, batch algorithms partition an existing transaction dataset at a certain point in time. Online algorithms learn the potential grouping structure within the data steadily and will recognize a new upcoming arrangement of the market baskets automatically. Moreover, the sequential calculation of a single input vector unburdens computational resources – as opposed to the processing of all point-centroid assignments in the memory of a machine. Theoretically, online algorithms can handle unlimited streams of records, which is an important advantage in market basket analysis with its high-dimensional data.

The machine learning and neural network communities present a huge number of such online algorithms. Some of them have been adapted to MBA successfully (cf. e.g. Ainscough and Aronson 1999, Decker 2005). Regarding further issues of the centroid updating process, the algorithms differ in soft and hard competitive learning techniques. Hard competitive learning implies centroid updating as implemented in Hartigan and Wong's (1979) exchange algorithm: only the closest centroid is adjusted. Hence, this is dubbed the "winner-takes-all" principle in the literature. Comparable to the self-organizing vector quantization algorithms described by Ripley (1996) as well as Hastie, Tibshirani and Friedman (2001), an online algorithm with hard competitive learning developed by Reutterer et al. (2006) is used to find groups of similar transactions in the recorded dataset of a do-it-yourself market and the ZUMA dataset (cf. Papastefanou 2001, Schnedlitz et al. 2001, Boztug and Reutterer 2008).

Since online partition algorithms depend in the same way as batch algorithms on an appropriate initialization of the starting partition, the centroid updating can follow the soft competitive learning concept to avoid a weak local optimum and to increase the quality of the partition. Soft competitive learning means that a neighborhood function defines not only the winning centroid for the updating process, but also the centroids lying within a defined radius around it. Updating the neighbors of the assigned center increases the algorithm's ability to move away from a potential non-optimal arrangement of the entities. Regarding the movement of the centroids within the feature space, the intensity of the adaptation has

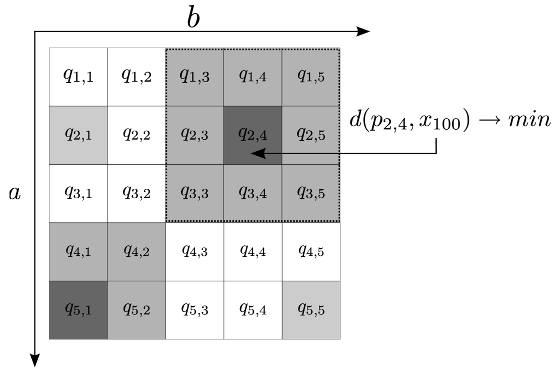

Figure 2.3: Example of a small Self-Organized Feature Map

to decrease constantly over time to approximate a local optimum. The learning rate, which typically follows a monotonously decaying function of time, ensures this for hard and soft competitive learning algorithms. In the case of soft competitive learning, the neighbors of the winning unit usually "learn" with less intensity the greater the distance to the winning unit becomes. The different algorithms vary in the conceptual construction of the neighborhood function or the learning rate. For instance, the functions can decrease exponentially or linearly (cf. Fritzke 1997).

Self-Organized Feature Maps: The self-organized feature maps (SOM) introduced by Kohonen (1982) are an often-cited example of online soft-competitive learning algorithms. SOMs are usually two-dimensional rectangular grids. Each a, b-cell of the grid is called a unit $q_{a,b}$ (cf. Kohonen 2000). Similar to a centroid vector of a KCCA algorithm, a reference vector $p_{a,b} = \{w_1, w_2, ..., w_J\}$ represents each unit of the grid. In accordance with the visualization preferences of the analyst, it is also possible to build a three-dimensional cube or to order the units in-line. After defining the number of units $a \times b$ and their dimensional arrangement, the algorithm starts with a random initialization of the reference vectors. Using SOM for exploratory MBA, each binary input vector of the transaction dataset is assigned to the unit $q_{a,b}$ whose corresponding reference vector is closest, as determined by a pre-defined distance (or similarity) measure $d(.)$. Again, since binary market baskets are considered, the analyst can replace the originally used Manhattan distance with a Jaccard distance (cf. Section 2.2.1). In the updating process of the reference vectors, the neighborhood function and learning rate define the participating units and the strength of their adaptation. The algorithm stops when a predefined maximum number of transaction vectors is reached or another criterion is fulfilled.

The key idea of the SOM algorithm can separated into two issues:

1. In the same way as other partitioning algorithms, SOM builds groups of similar transactions. Each reference vector $p_{a,b}$ acts as the prototypical market basket of a segment (cf. Decker 2005).

2. Additionally, the SOM arranges the resulting reference vectors according to their similarity in the two-dimensional grid. Hence, the resulting grid reflects an overall topology of the defined transaction segments.

The second issue is a characteristic feature of SOM, which is explained in simplified form in Figure 2.3. It shows one iteration with transaction x_{100} pointing to the winning unit $q_{2,4}$, since the distance $d(.)$ to the reference vector $p_{2,4}$ is minimal. The dotted line in the figure corresponds to the radius of the neighborhood function. In this example, the radius encloses all units directly adjacent to the winning unit. Besides the reference vector $p_{2,4}$, the eight corresponding vectors of the units next to $q_{2,4}$ will be updated as well – but to a lesser extent due to the decreasing learning rate. In contrast to the "winner-takes-all" principle, this reduces the danger of remaining stuck in a certain area of the grid.

Similarly to the graphical visualization used in many data mining software packages such as SPSS Clementine©, the shading of the grid's cells becomes darker the more transactions are assigned to the corresponding unit $q_{a,b}$. Light areas in the grid are units which do not attract many transactions. Due to the learning rate and the neighborhood function, similar prototypes $p_{a,b}$ determine adjacent units in the grid after the last iteration is complete. Hence, darkly shaded areas in the feature map visualize segments of numerous transactions with the represented vectors defining similar market basket prototypes. Decker and Monien (2003) present an 8×8 SOM that shows the segmentation of transactions including up to 25 different categories. After 100.000 iterations, the algorithm groups similar prototypes in different areas of the grid. These areas indicate that e.g. similar spots of clusters exist in the grid whose reference vectors reflect the combination of different hair care products.

The advantage of SOM for market basket analysis lies in the arrangement of the prototypes in the grid. By visual inspection of the shaded area, it is possible to explore groups of similar transaction segments if the items of the reference vectors are also listed for every unit. Decker and Monien (2003) replace the shaded cells of the SOM with the numbers of the categories with the highest purchase frequencies. The final map reflects purchase correlations in two ways. On the one hand the prototypes mediate purchase correlations between the included categories; on the other hand the shaded areas represent bundles of similar (but not equal) prototypes whose involved categories are also correlated.

When the analyst inspects the areas of similar prototypes in the grid, he gets an overview of the most important category combinations in his assortment. For the example Figure 2.3, two areas indicate a possible interesting correlation between the categories of the corresponding reference vectors. Moreover, defining K is less problematic compared to a K-means exchange algorithm since the darkly shaded areas appear automatically – as long as the number of units exceeds the number of expected segments. The number of shaded areas might be a proper recommendation for the K-value if further KCCA algorithms are applied to the dataset. Nevertheless, evaluating the outcome of SOM is not straightforward when the dimensions of the grid become high. Another problem is to define a priori the dimensions of the grid. To overcome this, the algorithm can include a condition which controls the value of a, b. If needed, the algorithm is able to increase or decrease a, b during the iterative learning process. This liberates the analyst from the need to determine a fixed grid (cf. Fritzke 1997).

Neural Gas Networks: The neural gas network (NGN) partition approach also stems from machine learning research and is a typical soft-competitive learning algorithm. Like the SOM approach, reference vectors represent a number of units called “nodes”. Again, a neighborhood function and a learning rate determine the units and the strength of the adaptation of the reference vectors. Since there is no two-dimensional grid, the participating units of the learning process are not determined according to a radius around the winning unit, but according to a ranking level. The outcome of the NGN approach is very similar to that of KCCA: the reference vectors act as the prototypical market baskets of a segment with the highest weights w_j matching the purchase probability of the corresponding category j.
Decker and Monien (2003) transferred the NGN approach to the market basket context and were able to find the same purchase correlations as were revealed in the dataset passed through the SOM algorithm. In contrast to the SOM approach, which arranges similar prototypes within the $a \times b$-grid, ten segments (or units) were generated. Determining an appropriate number of units equals the problem of defining K in KCCA algorithms. To lessen the problem of finding the proper number of units, Decker (2005) adapted growing NGN, which is similar to the extended “growing grid” approach of SOM (cf. Fritzke 1997). Instead of defining a fixed number of units, the growing NGN approach adds and removes them automatically according to the units’ ability to represent the incoming transactions.

Implications

Building groups of transactions with partitioning cluster algorithms seems to be a particularly well-adapted, powerful technique for revealing purchase correlations. The different applications such as self-organized feature maps, soft-competitive learning or KCCA algorithms enable analysts to uncover groups of similar transactions and give the opportunity to use the results for marketing decisions such as cross-promotion strategies. Above all, the ability of online algorithms to partition growing datasets steadily is a worthwhile feature since the transactions accumulated dynamically can be handled successively even with weaker computational resources.
In contrast, cluster algorithms, which require full similarity matrices such as hierarchical clustering or some graph partitioning algorithms (cf. Strehl and Ghosh 2000, Backhaus et al. 2006), can become a computational challenge. Hence, the application of each method also depends on the aggregation level

of the grouped items and the technical limitations. Nevertheless, the presented cluster algorithms deal only with anonymous transaction data and do not take the connection to the originators of the transactions into account. Since we aim at the building of a target marketing approach, and since we want to consider the customers behind the collected transactions, the cluster procedure has to be extended.

2.3 Analysis of Personalized Transaction Data

Reviewing the presented techniques of exploratory market basket analysis in the previous paragraphs, it must be noted that they all refer to customer-anonymous transaction datasets. This means that the retailer does not use any information about which customer has combined the different categories in the shopping basket. Since these techniques neglect the additional information of the customer ID in the first column of Table 2.1, the results of the techniques target the average customer and not individuals or groups of buyers. Thus the information gained about purchase correlations can be used for category management decisions (e.g. the listing or unlisting of complementary products) but not for targeting specific customers with appropriate promotional activities. Moreover, activities over time are also neglected. The corresponding analysis always reflects the buying behavior at the point in time when the individual transaction was recorded and does not consider customers' intertemporal purchase patterns. Owing to the increasing implementation of registration and loyalty programs, buyers are not anonymous anymore and establishing the chronological order of their conducted purchase occasions has become easy. This enables two important changes regarding the promotional strategies in MBA (cf. Passingham 1998, Liebermann 1999).:

1. Retailers can collect a personalized purchase history even if the customer uses different stores belonging to the same company. This facilitates discovering the specific customer's intertemporal purchase correlations.

2. Customer identification allows retailers to specify and to address similar individuals or groups of buyers combining comparable categories.

In the following sections, we describe several techniques of exploratory MBA using customer identification to increase the informative value of the analysis. We start with Section 2.3.1. It explains a simple extension of ARM algorithms to consider registration data. Afterwards, we introduce a modified ARM algorithm to find intertemporal purchase correlations.
Section 2.3.3 briefly describes recommendation systems based on collaborative filter algorithms. Recommendation systems usually support the present purchase occasion of an online visitor by comparing his current behavior to that of other customers. Section 2.2.3 comprises the extension of partitioning cluster

algorithms to non-anonymous customer data. The idea introduced there is crucial for the methodological framework explained in Chapter 3.

2.3.1 Virtual Items for Association Rule Mining

To enable the algorithms of ARM to take personal information about customers into account, it seems to be useful to integrate virtual items. Virtual items represent not a category or product but an attribute, e.g. gender or age in demographic census data. These attributes act as an artificial item and can be treated in the same way as regular ones. For example, if a loyalty program is applied (see Section 2.1.1), the personal information of the participating customer is known as a result of the registration procedure. By inserting these variables into additional columns of the data matrix, the mining algorithm could identify association rules such as $\{female,\ women's\ clothes\} \rightarrow \{skin\ care\}$. Nevertheless, including too many virtual items can lead to an inferior outcome (cf. Berry and Linoff 2004). Since these attributes usually show more 1 values (cf. Bayardo, Agrawal and Gunopulos 1999) than transaction data, mining algorithms with low minimum support thresholds might find many correlations between virtual items. This can distract the analyst from the interesting category correlations. Nevertheless, considering virtual items is a first simple method for exploiting customer information from registration or loyalty data (cf. Berry and Linoff 2004).

2.3.2 Mining Sequential Patterns

Due to the absence of a link to a specific customer, frequent itemset mining in anonymous transactions can only reveal behavioral patterns in aggregated, closed purchase occasions. Observations over time are impossible since the behavior of the single customer is not tracked. The implementation of the loyalty programs makes it possible to follow the transactions of a customer and to consider his consumption habits over the course of a period of time. Within the collected transactions of customers, retailers expect to find similar intertemporal patterns which represent a typical chronological sequence of purchases. An example of such an intertemporal purchase pattern in the field of market basket analysis is the customer who buys non-perishable baking ingredients (e.g. flour) in the months before Christmas and perishable goods (such as baker’s yeast and fresh milk) close to the point in time when he or she actually wants to bake Christmas cookies. Identifying specific behavioral purchase sequences is valuable for target marketing since it enables retailers to predict customers’ interest in certain products at a certain point in time. In addition to forecasting purchases made in the future, sequence mining is used to do logfile analysis in online shops. Tracking the virtual movement of a customer through a web content makes it possible to reconstruct the source of his interest in specific products (cf. Dong and Pei 2007).

One of the first sequence mining algorithms was introduced by Agrawal and Srikant (1995) who resembled the APRIORI algorithm introduced above. Since the problem of mining sequences is very complex, researchers tried early on to increase the effectiveness of the algorithm by implementing methods to reduce the computational load, such as parallel computing or data sampling (cf. Section 2.2.2). Zaki's (2001) SPADE algorithm is a very efficient algorithm and can deal with high-dimensional data. To improve the outcome, the algorithms can consider different constraints, e.g. a *minimum time period* in which a specific sequence has to be finished (cf. Srikant and Agrawal 1996). Regarding the example above, the purchase of baker's yeast should be finished in December of the same year and not in December of the following year. Unless the time slot of a few months is kept in mind, this sample sequential pattern is quite useless.

2.3.3 Recommendation and Decision Support Systems

In online environments, retailers can identify their customers by registration, login data or even by IP addresses; shoppers extend the transaction data set automatically with every purchase occasion. Tracking customers' paths through the webpages of an online shop or registering the items added to the virtual shopping cart is not as difficult as in stationary retailing. Most web browsers store small data files on the users' computers (so-called "cookies") to retrace users' movements through the virtual shops. Companies such as Amazon and eBay early recognized the potential of analyzing the online behavior of their visitors to set up selling strategies (cf. Ansari, Essagaier and Kohli 2000, Zhang and Krishnamurthi 2004). One strategy is to support the current visitor of a shop with purchase recommendations derived from the behavior of other customers with a similar buying intention. Such recommendation systems usually display products which might suit the interest of the current online visitor (cf. Reutterer and Mild 2003, Bodapati 2008). Here, we call this targeted online user the "active user" with his "active transaction". Since the active transaction at least is a personalized one (e.g. due to the IP-address or registration data) and the objective is to target an individual customer, we subordinate recommendation systems to techniques analyzing non-anonymous transactions.

Earlier work has shown that online shoppers appreciate being supported by such automated decision support systems since they help in reducing buyers' search costs. Due to unlimited virtual "shelf-space" and the "just-in-time" delivery of manufacturers, online assortments can comprise many more items compared to stationary assortments. The large number of products increases the effort needed by the customer to find suitable items. With recommendation systems it is possible to present the active user with products which might appeal to his interests. This can result in a more satisfying shopping experience for the customer. Hence, both retailer and clientele usually benefit from the implementation of these systems (cf. Hübl and Trifts 2000).

Most of the recommendation systems currently in use involve content-based or collaborative filtering (CF) techniques. Content-based systems suggest similar objects according to the previously visited information sources of the online user. For example, the system suggests to the user a text document with a content similar to his preferences. Collaborative filtering approaches try to predict useful recommendations according to an equivalent choice behavior of other customers (cf. Runte 2000). The system depends on a sufficiently large dataset of collected transactions or clickstreams of comparable visitors. When looking at collaborative filtering, two different classes exist: first, model-based approaches develop a descriptive model to derive the recommendations for a single user; for example, Ungar and Foster (1998) use a Bayesian clustering model. In contrast, memory-based approaches extract item recommendations by comparing the current virtual shopping cart with fulfilled transactions of other observed customers. If the item composition in the market basket of the current online visitor conforms with similar basket compositions, the system will recommend buying those items which have been bought by the other customers as well. For example, the online visitor has already clicked on J. K. Rowling's "Harry Potter", volumes one and two. If the data of the online bookstore includes a sufficiently large share of visitors who also bought or clicked on volume three, the system will probably recommend it to the active user. In accordance with the MBA techniques introduced above, the determination of similarities plays a major role for these techniques as well.

To show the similarities between the previous methods of exploratory MBA and memory-based recommendation systems with CF-techniques, we demonstrate an approach introduced by Ungar and Foster (1998). It deals with binary datasets and derives a TOP-J list of item recommendations for the observed active user α. Let us consider a data matrix as shown in Table 2.1, but without customer identification (i.e. the first column is excluded). In addition, let us define an active customer α who has combined item i_1, i_6 and i_8 in his virtual shopping basket x_α so far. Comparing this preliminary item combination with the example data matrix, it is exactly equivalent to transaction no. 4, 8 and 14. Since the three virtual customers did not buy any further items, they are not useful for recommending a product to the active user α. In contrast, transaction no. 10 contains item i_5 in addition to items i_1, i_6 and i_8. Since the corresponding user shows an equivalent purchasing behavior except for i_5, this item might be a good choice for the active user as well and should be recommended.

The recommendation system based on CF presented here solves this task in a two-step procedure. The first step calculates the corresponding purchase probability of every item for the active user. The second step then returns a TOP-J list of suitable item recommendations. According to these steps, Equation 2.19 determines the active user's purchase probability $\gamma_{\alpha,j}$ of item j. The probability depends on a similarity weight ω between the current transaction of the active user and the purchases made by all other users of the dataset. Since $\gamma_{\alpha,j}$ represents a probability value, the normalizing factor κ ensures that the single values add up to 1.

$$\gamma_{\alpha,j} = \kappa \sum_{n=1}^{N} \omega\left(x_{\alpha}, x_{n}\right) u_{n,j} \tag{2.19}$$

The similarity (or distance) measure ω determines the correspondence between x_α and each x_n. Again, the Tanimoto similarity measure (or Jaccard distance) seems to be a good choice for binary transaction datasets, although the Hamming distance or other measures can also be used for such systems (Hamming 1950, Mild and Reutterer 2001). Notice that we figure the similarity between the rows of the matrix instead of the columns, as was done for the pairwise association analysis described in Section 2.2.1. Table A.6 includes the values for the comparison between the example active shopping basket x_α and the transactions of the data matrix. The highest digits represent the identical transactions nos. 4, 8 and 14. According to Equation 2.19, multiplication with the binary variables $u_{n,j}$ builds for every item the purchase probability $p_{\alpha,j}$.

The second step comprises the building of the TOP-J recommendation list. Less sophisticated methods use static thresholds, e.g. they recommend the TOP-J items with $p_{n,j} \geq 0.5$. This method does not consider the unequal distribution of items' occurrences in the transaction data (cf. Chapter 1). Mild and Reutterer (2001) calculate the residuals between the median purchase probability over all transactions $median(\gamma_{.,j})$ and each single $\gamma_{n,j}$ (see Equation 2.20). The resulting values are inserted into Table A.8 (see Table A.7 for the median values).

$$diff(n,j) = \gamma_{n,j} - median(\gamma_{.,j}) \tag{2.20}$$

In our example, we want to generate the TOP-4 recommendations by determining the four highest values 0.20, 0.17, 0.15 and 0.12 in Table A.8. These figures highlight the expected four items (j_1, j_5, j_6 and j_8). Of course, the system would only recommend j_5. Suggesting the other three items is unnecessary since they are already included in the active transaction x_α. Summarizing the result, the item's probability of appearing in the recommended TOP-J list increases if the items occur in a large share of transactions which are comparable to the observed active transaction.

When looking at pairwise association analysis, association rule mining, partitioning approaches and recommendation systems based on binary transaction data, measuring the similarity between the transactions is an important technical core for all approaches. Typical challenges discussed in the foregoing sections, such as a skewed distribution of items' purchase frequencies and data sparsity are problematic for collaborative filtering approaches as well (cf. Mild and Reutterer 2001). In particular, the sparsity of data matrices is a major problem. To recommend interesting products to an active user, the usability of the system depends on transactions with similar item compositions. If the transactions contain only a few items on average, the probability of finding comparable market baskets tends to be lower. Moreover, high-dimensional matrices can decrease the time of response if the algorithm does not need sampling or other data reduction techniques.

The practical application of product recommendations with collaborative filtering approaches is very much limited to online environments. Due to the absence of a necessary web-interface which allows an immediate reaction to the customer's product compositions, this technique could not be applied in common supermarkets readily. Although some authors have introduced a similar decision support system in stationary retailing (cf. Lawrence, Almasi, Kotlyar, Viveros and Duri 2001), it depends on sophisticated technical equipment. For example, shopping carts and/or the customers have to be endowed with elaborate displays, RFID electronics or smartphones. In many business contexts, the high investment costs will hinder the implementation of these methods.

The collaborative filtering approach described here does not necessarily need personalized transaction data (cf. left column of Table 2.1) to determine the recommendations for the active user. Nevertheless, particularly in the case of very sparse data, it is useful to consider not just the single transactions but the complete buying history of a customer. For example, a user might buy the first volume of “Harry Potter” on Monday and the second volume on Wednesday. Since the ordinary system would not consider the time factor which is made possible by personalization, both transactions would be considered separately and include only one single item. Using information e.g. user registration in online shops, the buying history of a specific customer can be compressed to a representative transaction which comprises all items bought during a specific period of time. Take for example user $m = 1$ in Table 2.1, who made the transaction x_7 and later on x_{13}. By simply changing each $u_{n,j}$ to a value of 1 if the item was bought at least once, this short buying history could be compressed to $x^* = \{1,\ 1,\ 1,\ 0,\ 1,\ 0,\ 0,\ 1\}$. The advantages are quite obvious: the sparsity and dimensionality of the data table decreases, intertemporal purchase correlations are included and the item recommendation has a chance of becoming more appropriate to the active user's interest.

With regard to the objectives set out in Chapter 1, collaborative filtering approaches are less suitable for target marketing approaches in stationary retailing. Hence, they are not needed for the framework introduced in Chapter 3. Nevertheless, we would like to apply an approach that suggests suitable products as clearly as recommendation systems do. The next section describes constrained algorithms of cluster analysis for analyzing personalized transaction data. The constrained cluster algorithms play a major role in the target marketing framework we introduce, the outcome of which recommends certain products to specific customer groups in a way similar to decision support and recommendation systems.

2.3.4 Customer Segmentation with Constrained Cluster Analysis

Partition techniques used to build clusters of similar market baskets in anonymous transaction datasets have been explained in Section 2.2.3. The question is how to extend these algorithms to build clusters of customers with similar buying behavior. One idea is that the existing partitioning approaches should take the identification numbers (IDs) of every transaction into consideration and use them to build groups of customers with comparable purchasing behavior, as reflected in similar market basket compositions over a specific period of time. Algorithms of constrained clustering can handle the data extension of the customer identification for this purpose (cf. Basu, Banerjee and Mooney 2004, Basu, Davidson and Wagstaff 2008). The need for such algorithms has been known for as long as the problem has existed. In many real-world situations, additional information about the dataset's structure is available from the outset and forces the ordering of the entities subject to a specific constraint. For example, in soil science analysts have to construct sets of contiguous terrains characterized by a similar distribution of soil properties. Instead of just grouping the area's properties (i.e. the items), the segmentation has to consider the location of the mineral resources in a specific terrain (i.e. the constraints) when building the sets (cf. Aldenderfer and Blashfield 1984, Wagstaff, Cardie, Rogers and Schroedl 2001). The idea is to integrate the background information for clustering purposes. By including constraints conditioning the linkage between particular entities, analysts hope to enhance the output of the segmentation and come closer to the natural structure of the dataset (cf. Gordon 1981). Although verifying the "real" groups within empirical data is usually impossible, the constrained cluster algorithms seem to describe predefined grouping information within synthetic datasets better (cf. Aldenderfer and Blashfield 1984, Wagstaff et al. 2001). Basu et al. (2008) provide an introduction to constrained clustering and explain a number of available algorithms.

In the MBA context, the collected buying histories provide the background information. The m values define the bundle of transactions which have to belong to one specific customer. Instead of clustering all single, anonymous market baskets, the approaches link the transaction sequences (or buying histories) as background information to the cluster algorithm. In terms of the classification literature, the buying histories act as so-called must-link constraints, since all transactions of the whole sequence have to belong to a single group and must not spread onto different segments (Wagstaff et al. 2001). Hence, a resulting segment includes a number of customers who are characterized by making similar compositions of jointly purchased categories during their shopping trips. Since algorithms of K-centroid cluster analysis are a popular technique for exploratory MBA, in the next chapter we introduce extensions of these algorithms to deal with constraints and choose an appropriate one for our target marketing approach.

3 Methodology of the Target Marketing Approach

The previous chapter explained the different exploratory MBA techniques. To meet the demands stated in the introduction, we combine some of these techniques to build a target marketing framework. This chapter introduces the methodological concept of the framework's stepwise procedure in detail. The combined approach supports retailers in partitioning their customers, understanding the segment-characteristic category correlations and identifying those categories which should be featured in an appropriate target marketing campaign to optimize its profits.

Figure 3.1 illustrates the complete procedure of the framework, the steps of which are described in the subsequent sections. In keeping with the objective of segmenting the households, we first want to build customer groups showing similar category correlations among the items bought less frequently (i.e. LFC). Therefore, we test three modified algorithms of K-centroid cluster analysis dealing with constraints as introduced in Section 2.3.4. The presented algorithms take into account the fact that the individual buying history summarizes the transactions of a certain customer. Instead of dealing only with single transactions, the algorithm has to consider the complete transaction sequence of a customer as must-link constraints when building the customer groups. Since more than one alternative is present to implement the must-link constraints into the iterative procedures of the algorithm, we will compare three relevant variations of KCCA algorithms.

Taking the buying sequences into consideration is of crucial importance for our approach. Due to the lower purchase frequencies of products in the long-tail, there tend to be only a few similar basket compositions, and finding groups according to this feature becomes difficult. Using the additional information that the transactions of a buying sequence all have to belong to the same cluster should enhance the recognition of similar customers. For verification purposes, we analyze artificial transactions with a known grouping structure. The predefined structure of the artificial data is comparable to that of real-world transaction datasets. Hence, the algorithm which reveals the structure of the artificial data best might be more likely to find the groups in the dataset provided by the empirical application in Chapter 4.

The next two steps of the framework aim at deriving suitable item recommendations for each cluster of customers. If the cluster analysis identifies customers with similar basket structures, this implies members' comparable interest in particular product compositions. Recommendations derived from the

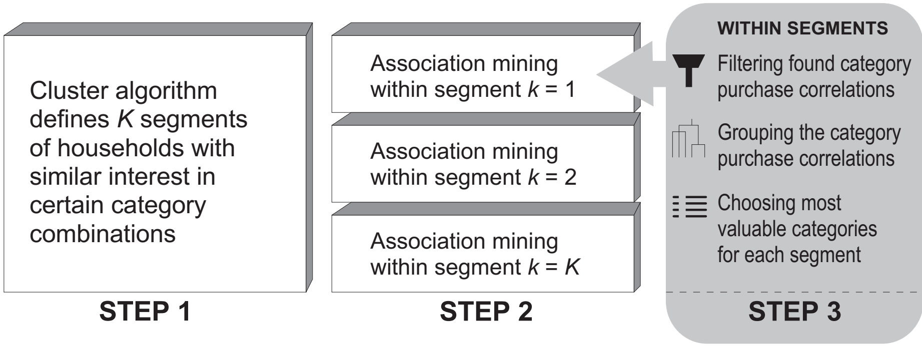

Figure 3.1: Stepwise procedure of the framework

found segment-specific category correlations promise to be better attuned to the expectations of the cluster members. Consequently, these items will also have a greater chance of stimulating cross-selling. An association rule mining algorithm specifies the category correlations after the partitioning has taken place. Therefore, the combined buying histories of a customer group are pooled into a segment-specific transaction dataset and are explored in the same way as the aggregated transactions. The resulting frequent itemsets include segment-specific purchase correlations which occur sufficiently frequently in the corresponding cluster. By combining ARM with partitioning cluster analysis, we balance the deficiency of most ARM algorithms in discovering product associations that only reflect the buying behavior of average clientele (cf. Section 2.2.2).

From the revealed segment-specific frequent itemsets of the foregoing step, the final step of the approach can extract single items used to build a recommendation list for target marketing purposes. To reduce the number of found associations, we select only the FI which contribute most to the retailer's utility. As introduced in Section 2.2.2, a filter measure is used to separate the interesting associations from the less-important ones. This reduces the number of considered category correlations and decreases the computational effort of the subsequent steps of the analysis. Moreover, a hierarchical clustering of discovered segment-specific associations can give retailers a better insight into the category-correlations of a specific cluster and makes it possible to group the segment-specific FI according to their similarity. The frequent itemsets are used as input for an optimization routine. This routine determines a predefined number of single categories which maximize the profit and take categories' purchase correlations and their gross profit margins into account.

The segmentation objective of the approach is the most complex module since the cluster building process depends heavily on an appropriate algorithm. If the algorithm is not able to build a useful customer partition for marketing purposes, the remaining steps of the approach will be pointless. In the next section, we discuss the partitioning problem in detail and compare three modified KCCA techniques.

3.1 STEP 1: Identifying the Customer Segments with Constrained Clustering

Following prior research and recalling the assumptions made in Chapter 2, the composition of a market basket corresponds to a 'pick-any/J' problem. Each transaction can be interpreted as a J-dimensional binary vector $x_n \in [1,0]^J$ with J representing the range of the retailer's categories and $n = 1...N$ the number of transactions (cf. Manchanda et al. 1999, Russell and Petersen 2000). Integrated into a matrix X_N, the rows correspond to transactions and the columns represent the items (cf. Table 2.1).

Due to the presence of a loyalty program, the transactions of customer m (with $m = 1 \dots M$) can be pooled into his buying sequence G_m. If the summarized transactions X_N can be linked to a data point classification $G_N = \{g_1, \dots, g_N\}$ with $g_n \in 1, \dots, M$, the customer's buying history (or transaction sequence) is denoted by the following equation (Leisch and Grün 2006):

$$G_m = \{x_n \in X_N \mid g_n = m\}, \qquad m = 1 \dots M, \qquad M \leq N \tag{3.1}$$

If buying histories are implemented, the linkage between the transactions and their initiator remains and extends the data organization. According to the terminology of data-theory, this format can be considered *three-mode* data (Coombs 1964). The category incidences J (first mode) provide the segmentation base. The customer mode $m = 1, \dots, M$ corresponds to the second mode and the aggregation of the customer's transactions into the personalized buying sequence G_M acts as the third mode. Notice that the number of repeated observations varies with m. Hence, each customer has a purchase history of an individual length. For example, the buying history of customer $m = 1$ in Table 2.1 includes only two transactions while the sequence of customer $m = 5$ comprises four market baskets. We expect from a methodological point of view that the length of the transaction sequence will correspond to the strength of the probability of assigning the related buyer to the appropriate cluster. In other words, observing more transactions will tend to uncover the customer's typical category purchase correlations better and will be more likely to assign him to the appropriate group. This assumption is examined and verified in Section 3.1.4. It seems to be practicable to define a minimum length of the buying history as a prerequisite for taking part in the partitioning step. For instance, customers who make just a single purchase occasion during a period of one year do not show a high loyalty to the retailer. Ignoring these customers for our target marketing approach can be useful if the retailer still provides them with standard promotional campaigns.

Concerning the customer-cluster assignment conducted with the constrained partitioning algorithms, each buyer belongs to one segment exclusively. In visual terms, the borders of each cluster do not overlap each other. Although the approaches could be extended to fuzzy clustering, which allows the assignment of one household to different groups, retailers often want to treat each customer with just a single marketing action (e.g. one brochure or email-newsletter per household). Owing to limited marketing budgets, this scenario is probably more common than directing several campaigns at the same customers. Hence, the presented cluster techniques do not consider fuzzy segmentation (for an example approach see Chaturvedi, Carroll, Green and Rotondo 1997).

The three contrary modifications of the generalized KCCA algorithms which are able to deal with background information are evaluated in the following subsections. The techniques differ mainly in the point in time at which they include the buying histories for the customer clustering (cf. Wagstaff et al. 2001, Leisch and Grün 2006). Here, we will look at three variations:

- *A priori*: Before clustering takes place, each customer sequence could be compressed to one synthetic market basket. The resulting transaction represents the average structure of all the customer's m transactions.
- *A posteriori:* After the cluster process has been conducted without considering the buying histories, the customer could be assigned to the segment that includes the majority of his transactions.
- *Simultaneous*: The buying sequences of the customer are implemented as must-link constraints to the iterative cluster process. In this case, the sequences are considered during the partitioning.

Judging from earlier studies, the methods will produce different partitions (cf. Balka 2005). Although Wagstaff et al. (2001) have shown the predominance of simultaneously contrained clustering to standard K-means, the authors applied the algorithms to small datasets with metric values. In addition, the GPS line data used does not usually show the specific characteristics of binary transaction data. High dimensionality, sparsity and correlations in the categories bought less frequently complicate the process of finding appropriate segments due to weak grouping information in the data. Hence, the question arises which of the three proposed techniques is able to deal with the stated challenges best. To answer this question, we explain the methodology of the three modified algorithms in the subsequent sections. In Section 3.1.4 the constrained cluster algorithms should reveal predefined grouping information in artificial transaction data with the data showing the expected characteristics of real-world market basket data gained from loyalty programs.

3.1.1 A Priori Consideration of Constraints

To reflect the past transactions of a buyer, the analyst can compress his or her buying history into a single, representative market basket. We introduced this method in Section 2.3.3 for recommendation systems whose datasets struggle with high sparsity. If the sparsity is hindering the production of appropriate purchase recommendations, building a representative, more dense transaction seems to be a good choice. For this technique, the must-link constraint would be considered *before* the cluster algorithm is conducted. Since the consolidation of the sequence brings out one representative transaction y_m for each customer m, this method reduces significantly the number of entities which a cluster algorithm has to partition. Moreover, the pruned dimensionality of the data saves computational capacities. A popular way to construct the compressed representative transaction y_m is to form a one value for every category that occurs at least once in the customer m's buying history.

Another way to construct y_m is the following method: a vector ρ_m denotes the line-by-line summarized transactions of a buying history G_m. The mean values of ρ_m are built for each J-value over the sequence G_m. According to a threshold o, the J vector values are transferred into binary data. The vector values turn into a 1 value if they exceed the threshold. In the other case, they denote a zero. The resulting vector is y_m. The height of the threshold o determines how often a specific category j has to occur within the buying history of a customer to turn the j-values into 1. If $o = 0.5$ is chosen, at least half of the buying history's transactions have to contain the contributed category j.

The compression of the buying history usually decreases the sparsity and dimensionality of the data. Nevertheless, this method ignores the information showing which items were bought together during a single purchase occasion and also the point in time at which the corresponding transaction was made. Moreover, if the compression method is not declared from the beginning, the output of this technique will depend on how the analyst proposes to build the representative baskets from the whole sequence. Choosing among different techniques or constructing one's own compression step could overburden decision makers. In addition, finding a suitable method to build representative market baskets requires assumptions and knowledge about the structure of customers' average buying histories. Since the practicability is quite doubtful and the advantage of contrained clustering should arise from its "unsupervised" character, a more automated course of action is actually preferable. Due to its inherent deficiencies, the a priori consideration of customer sequences seems to be of limited interest for our target marketing context. This is also supported by the results achieved with the analysis of the synthetic data (cf. Section 3.1.4).

3.1.2 A Posteriori Majority Voting

Grouping information can be considered *after* the cluster algorithm has been completed. This means that the standard KCCA algorithm (e.g. K-means) is applied to the transaction dataset X_N in the standard way. Hence, the iterative process does not differ from the one described above (cf. Section 2.2.3). A subsequent voting approach supports the consideration of the buying histories as must-link constraints. For example, Equation 3.2 denotes a measure s_k^m that defines how many of a customer's m transactions are assigned to partition c_k (cf. Reutterer et al. 2006):

$$s_k^m = \sum_{x_n \in G_m} 1_{\{x_n^m \in c_k\}} \tag{3.2}$$

The maximum value of s_k^m determines the cluster k of the considered household. According to this voting scheme, each buyer is mapped onto one of the segments k whose centroid vector p_k best reflects the majority of that customer's m past purchases.

In the marketing context, majority voting has been successfully applied to transaction data (Reutterer et al. 2006). Since the outcome depends on the foregoing KCCA algorithm, the suitability of this method has to be tested for our objectives. This is necessary since usual algorithms of KCCA are not always a good choice for high dimensional sparse data (cf. Wagstaff et al. 2001).

3.1.3 Simultaneous Consideration of Constraints

In contrast to the a posteriori voting approach, this variation of the generalized K-means cluster algorithm looks at the customer's transaction sequence *during* its iterative steps. It avoids majority voting by using function $f(G_m)$ which simultaneously assigns the combined buying history as a whole to the best matching center. In terms of method, the generalized K-means cluster process described above has to be altered between Steps 2 and 3 to consider the buying sequences within the procedure (see Section 2.2.3). Equation 3.3 denotes the modified objective function:

$$D(X_N, P_K) = \sum_{m=1}^{M} \sum_{n=1}^{N} d\big(x_n, p(G_m)\big) \rightarrow \min_{P_K} \tag{3.3}$$

Compared to the standard iterative process of K-means, the following enhanced steps have to be conducted:

1. Choose an initial set of K prototypes P_K and make sure to assign all x_n of each customer m to the same initial segment.

2. Assign each $x_n \in X_N$ to the cluster of the closest centroid in terms of the Jaccard distance measure $d(.)$.

3. Find the corresponding center $p(G_m)$ according to the function $f(G_m)$ for every G_m group of transactions.

4. Use cluster-wise means as new centroids instead of canonical ones.

5. Repeat steps 2 to 4 until convergence or until a predefined number of maximal iterations has been reached.

Step 3 is of primary interest since it determines the assignment of the transaction sequence to the corresponding centroid. Among others, two possible group mapping functions $f(G_m)$ can be compared. The first one assigns every single sequence G_m to the center having the minimal sum of distances to all $x_n \in G_m$. Here, it is called the ”minsum-distance” function. The other function links them to the center to which the majority of $x_n \in G_m$ is assigned. It is named the ”majority-clusters” function. We implement the first function since it considers the composition of the buying histories as a whole instead of conducting the assignment of the entities according to a majority voting technique.

3.1.4 Analyzing Artificial Transaction Data

The different methods by which the three-techniques include constraints will affect their suitability for our target marketing context. Due to the conceptual deficiencies of the a priori method, it will be of minor relevance compared to the a posteriori majority voting approach and the simultaneous implementation of the grouping information. Nevertheless, we analyze its results in the artificial data when the representative transaction y_m includes a 1 value for all items occurring at least once.

Although the concepts of the other two algorithms are quite similar to each other, the question arises as to *why* one should use a cluster algorithm which considers the present background information simultaneously as opposed to an a posteriori voting method. We assume that a simultaneous consideration of the background information will help to describe the hidden structure within the transaction datasets in a more accurate way compared to a technique that groups the households after clustering has taken place. To understand this assumption, consider the plots in Figure 3.2(a) to 3.2(c) of a generic example. Figure 3.2(a) depicts 250 randomly generated data points of two households within a two-dimensional grid. Obviously, the major part of the first household’s data points – symbolized by the black circles – lies above an imaginary horizontal line. This line separates the grid into two segments representing the *real* grouping structure. Most of the data points of household number two (gray triangles) are ordered below this line. Clustering these data points using the well-known K-means algorithm identifies two segments lying on either side of an imaginary vertical border, as shown in Figure 3.2(b). It represents a global optimum implying that no better grouping solution could be found with the algorithm for this random sample. Comparing the plot to the random data points in Figure 3.2(a), the solution achieved with K-means does not reflect the original groups very well. Nor does the situation improve if majority voting is applied to this outcome. For instance, a posteriori majority voting would check how many data points of each household belong to which cluster and – according to the data point’s majority – assign the corresponding household to the resulting segment. In this simple example, the majority of household no. 1’s data points (i.e. the black circles) – 66 of its 129 – belong to cluster no. 2 on the right-hand side of Figure 3.2(b). As a consequence, this household would be subordinated to cluster no. 2. In contrast, household no. 2 belongs to cluster no. 1 on the left-hand side of Figure 3.2(b) with most of its 121 data points. When compared to Figure 3.2(a), this resulting partition also appears different to the original grouping structure. Let us now assume that a must-link constraint is given which includes background information about the distribution of the datapoints. It takes into account (synthetically generated) information which determines the natural grouping of each datapoint. If the modified K-means cluster algorithm is used, which regards the must-link constraint during its iterative steps, the partition depicted in Figure 3.2(c) is achieved. Compared to the output of the a posteriori majority voting system, the resulting picture shows an arrangement that corresponds much more closely to the initial data distribution in Figure 3.2(a).

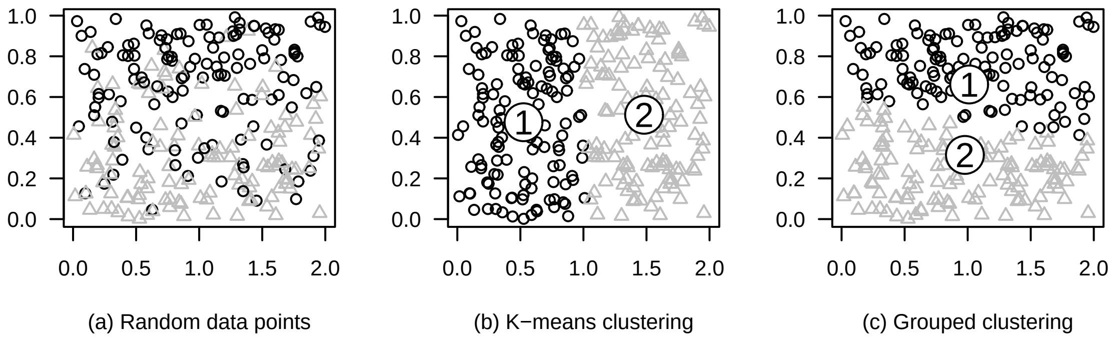

Figure 3.2: Example of finding grouping information within random data

This short example illustrates that similar algorithms can be more or less suitable to the predefined partition task. Notice that neither the K-means nor the a posteriori voting lead to "wrong" output. In the context of our stated problem, they both simply produce partitions which do not reflect the real grouping information in a sufficiently accurate way.

The implications of the given example can be transferred to our problem: we expect to find customer groups with a specific interest in certain category correlations made in the LFC of a transaction dataset. Now, the question arises as to which of the presented algorithms will be able to identify this expected structure if it actually exists. Since the grouping structure of real-world datasets could not be verified satisfactorily, we implement a predefined grouping structure in a synthetic transaction dataset. In this way, we try to give this data the typical characteristics of the real-world transaction data of a retail company. To determine which of the three methods for segmenting the customers in our target marketing context is best, all three algorithms are applied to the synthetic dataset. By comparing the cluster results, we can estimate which one might be the most appropriate technique for real-world transaction datasets.

The synthetic dataset contains 9,465 market baskets of 500 different customers who choose virtually

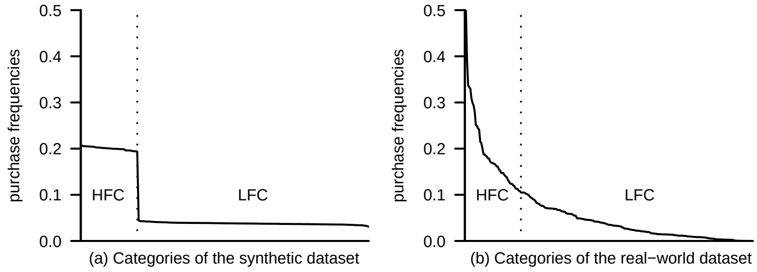

Figure 3.3: Purchase frequencies of the synthetic (a) and the real-world (b) transaction dataset of a supermarket

among 200 categories. Taking the customers' buying histories into consideration, we assume the distribution of Poisson and determine an average length of $\lambda = 19$ purchases made sequentially. According to our assumptions as laid down in Chapter 1 and Section 2.2.2, we design an artificial assortment with a skewed distribution of category purchase frequencies. Hence, the categories are divided into 40 HFC and 160 LFC.

Figure 3.3 shows a plot of the purchase frequencies in descending order within the synthetic and the real-world transaction data used for our empirical application. The vertical dotted line separates the HFC (left-hand side) from the LFC (right-hand side), e.g. according to a threshold. For the synthetic data, the determination of the two category groups is simple due to the stepped graph. In practice, the ordered purchase frequencies decrease monotonically, making it more difficult to determine a hard threshold separating the LFC from the HFC in real-world data (cf. Anderson 2006, Cavique 2007). We will return to this problem at the end of this section.

According to our assumptions about the common structure within retail transaction data, we compose a grouping structure in the HFC *and* the LFC. Customers probably vary in their habits when combining different HFC categories during their purchase occasion, e.g. customers who prefer to buy several kinds of beverages at the local store or who are more interested in compositions of dairy products. Moreover, the same customers will regularly but more rarely purchase items in the LFC, e.g. baby food and diapers. Hence, we first construct four major customer groups whose members chose among ten of the 40 categories with double the probability. The number of selected items from the 40 HFC are Poisson-distributed with $\lambda = 8$. All of the artificial customers can also belong to one of the 16 groups implemented in the LFC. The group members pick from ten defined LFC with a ratio of 1:3. Each market basket contains on average $\lambda = 6$ of the 160 LFC. Again, we assume a distribution of Poisson. Notice that the density of the purchases within the HFC is much higher compared to the LFC. If we try to reveal the grouping information within the LFC, the algorithms will have to deal with a sparsity of 97%. Concerning all categories (i.e. in HFC and LFC), the data is 93% sparse.

When analyzing the synthetic data with a constrained algorithm of K-centroid cluster analysis, we consider four modifications in order to take the buying histories as must-link constraints into account (see Section 3.1): no consideration of the buying history (= *standard*), before (= *a priori*), after (= *a posteriori*) and during the partitioning (= *simultaneous*). Apart from these variations, we implement the Euclidean and the Jaccard distance for each method to analyze whether the latter really does increase the quality of the output. Combining the two distance measures with the four techniques to include the constraints, we compare eight different cluster algorithms. When applying the algorithms to the artificial data, we have to consider the risk of finding a weak local optimum (cf. Chapter 2). Hence, we repeat each algorithm five times and store the one with the lowest sum of internal distances. Ten of these best solutions are stored for subsequent examination.

To evaluate the quality of the found cluster solutions, we have to define a criterion. A variety of different measures to compare cluster solutions exist (e.g. the Kappa measure, Cohen 1960). In this thesis, partitions are evaluated using the Rand or corrected Rand index (short: cRand, see Rand 1971, Hubert and Arabie 1985). A cluster solution determines for each item a cluster label which assigns the entity to the corresponding cluster. Given two cluster solutions with label vectors, the Rand index determines the agreement of the partitions by comparing all possible point-pairs of the vectors. For each point-pair, three different states can be distinguished: one in which both point-pairs are arranged in a common cluster, another in which they are separated into different clusters, and a third in which the pairs are mixed up. The mixed state means that one solution assigns both points to a common cluster while the other solution refers the points to different clusters. The Rand index increases with the share of point-pairs showing one of the first two states relative to all possible point-pairs.

Consider Table 3.1, which includes two example cluster labels of five points ordered into three clusters each. Table 3.2 shows the corresponding states of the point-pairs. Looking at point-pair (1,2), both cluster solutions assign these points to a common cluster, since point no. 1 and no. 2 have the same label in both vectors – label 1 for solution 1 and label 4 for solution 2. The point-pair (1,5) reflects the second alternative. The labels of both vectors are different (item 1 and 5 are assigned to cluster 1 and 3 for the first and to cluster 4 and 6 for the second solution). In contrast, point-pair (1,3) shows the mixed state because of the different cluster labels in both vectors (item 1 and 3 are both in cluster 1 while they are separated in cluster 4 and 5 for solution two). In summary, the two (virtual) algorithms clustered six of ten points identically and the Rand measure would define a value of 0.6 (cf. Table 3.2). Generally, the Rand measure determines values between zero and one, with a value of one meaning total cluster agreement and zero total disagreement.

The short example of Tables 3.1 and 3.2 implies an important advantage of comparing partitions with the Rand value. In contrast to many other measures, the Rand index does not depend on the identical

Item/point	1	2	3	4	5
Cluster labels / vector 1	1	1	1	2	3
Cluster labels / vector 2	4	4	5	4	6

Table 3.1: Two exemplary cluster labels of five points

Point-pairs	(1,2)	(1,3)	(1,4)	(1,5)	(2,3)
State	common	mixed	mixed	separate	mixed
Point-pairs	(2,4)	(2,5)	(3,4)	(3,5)	(4,5)
State	mixed	separate	separate	separate	separate

Table 3.2: Comparing the point-cluster assignment for the two cluster solutions

numbering of the cluster labels, which usually varies for each solution in practice. The Rand value can also be calculated if the number of found clusters is different in each of the two partitions. This issue becomes important in Chapter 4 when the Rand value is used to approximate an accurate value of K for the KCCA algorithm.

A problem of the Rand index is the dependence on randomness if a few very large clusters and several smaller ones are present. The more relatively big clusters are built by the algorithm, the more probable is the assignment of a point-pair to a common cluster. In contrast, the *corrected Rand* index considers the agreement of two partitions by chance and takes a value between -1 and 1. A value of -1 indicates full disagreement and a value of 1 full agreement between two partitions. Statistical independence between the partitions would be given by the zero value. For reasons of stability, we evaluate the different cluster solutions of the synthetic data using the corrected Rand index.

For each of the eight cluster algorithms we extract ten label vectors from the best of five solutions. The ten label vectors are compared with the real grouping vector built according to the above-stated grouping structures in the HFC and LFC. For each method and comparison, a Whisker plot illustrates the ten cRand values and shows how well the cluster assignment of the algorithm corresponds to the real grouping structure in the synthetic dataset (cf. Figure 3.4 and Figure 3.5).

Taking the varied cluster algorithms into consideration, we start by looking for the four groups within the HFC of the synthetic transactions. The resulting cRand-values are plotted in Figure 3.4 and indicate that the K-means algorithm is unable to identify the groups (cf. the standard method on the far left-hand side). The result does not change significantly if the Euclidean distance is replaced with the Jaccard distance. The best solution is gained from the KCCA algorithm that implements the constraints from the beginning. This supports previous research which has shown that the output's quality increases even with weak cluster algorithms if the grouping information is considered simultaneously as a must-link constraint during the iterative process (cf. Wagstaff et al. 2001). Although the *K*-means algorithm does not reveal the grouping structure, a posteriori majority voting improves the result if the Jaccard distance is implemented (cf. Reutterer et al. 2006).

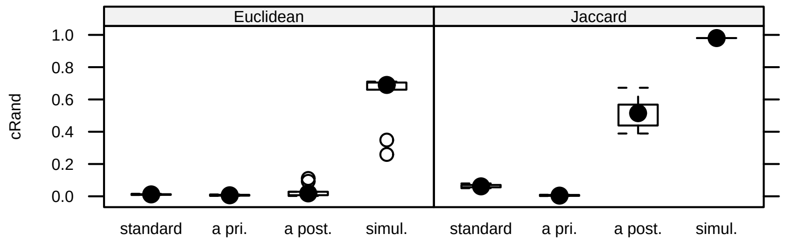

Figure 3.4: Identification of the four groups within the HFC using the eight cluster algorithms.

The situation changes if we want to find the 16 groups within the LFC of the synthetic dataset. With regard to the two upper arrangements of Whisker plots in Figure 3.5, the cRand values are low for all eight methods. Keeping this result in mind, we assume that the weak grouping information in the LFC is hidden by the dense data in the HFC. Since the purchase frequencies of the HFC are approximately four times higher compared to those of the LFC, the binary transactions contain much more 1 values within the 40 HFC. Similarly to data noise, this prevents the algorithms from finding groups in categories bought less frequently. Hence, the HFC are excluded from the dataset to enhance the results. The cRand-values of the plots in the middle of Figure 3.5 show that only the cluster method, which considers the buying sequences simultaneously, reveals the predefined structure. Compared to the implementation of the Euclidean distance, the Jaccard distance strongly supports the algorithm's ability to find the groups in the sparse LFC transactions. The higher cRand values in the middle right section of Figure 3.5 indicate this correlation.

The cRand values of the graphics in the top and middle of Figure 3.5 imply that the ability to find the grouping structure in the data probably depends on the modifications done to the algorithms' attributes. We want to verify how significant these effects are and which factors affect the quality of the output most strongly. To this end, we apply an ANOVA that includes the following factors: distance measure (Euclidean or Jaccard), method of considering the given grouping information (none, a priori, a posteriori, simultaneously) and exclusion of the HFC (yes or no). Conducting an ANOVA correctly depends on two conditions: first, standard distribution is assumed for the cRand values. A Kolmogoroff-Smirnov statistic has verified this successfully in this case. Second, the ANOVA requires homogeneity of variances. Even though a Bartlett test has produced a negative result, the ANOVA's output is usually unaffected if the variances are not very heterogeneous and the groups of observations are of similar size (cf. D'Agostino and Stephens 1986, Kanji 1993, Weerahandi 1995). All of the factors in Table 3.3 have a highly significant effect on the algorithm's output ($P < 0.001$). Taking the F-values into consideration, the exclusion of the HFC is mostly responsible for an increase in the cRand values. The next most decisive factors are the method of using the grouping information, followed by the type of distance measure.

	Df	Sum Sq.	Mean Sq.	F value
distance	1	0.51869	0.51869	657.95
method	3	2.26174	0.75391	956.32
exclusion	1	0.90228	0.90228	1144.52
distance:method	3	1.02500	0.34167	433.40
distance:exclusion	1	0.45870	0.45870	581.85
method:exclusion	3	2.15790	0.71930	912.42
distance:method:exclusion	3	0.88264	0.29421	373.20
Residuals	144	0.11352	0.00079	

Table 3.3: ANOVA output Table with response cRand subject to three factors

According to the ANOVA and the four graphics in the top and middle of Figure 3.5, decreasing the dominance of the HFC in the transaction data increases the cRand values. Although excluding the HFC enhances the ability of the algorithm to find the grouping information within the LFC, it is difficult for decision makers to define the HFC exactly. In other words, determining a threshold to cut off the "long tail" from the rest of the assortment depends on the retailer's personal experiences and the underlying characteristic of the dataset. For example, decision makers could drop categories occurring within approximately 10% of all transactions. This value is chosen heuristically and will not be optimal for all datasets. A solution is to weight the categories' frequency of occurrence. For this reason, we use the weighting value in Equation 3.4 and multiply it with each of its corresponding binary values in X_N. Again, κ is a normalizing factor to ensure $\sum_{j=1}^{J} w_j = 1$ with $w_j \in [0,1]$.

$$w_j = \kappa\big(1 - supp(j)\big) \tag{3.4}$$

The value considers the support (i.e. the purchase frequency) of every category in the dataset and lowers the factor of the HFC. After the eight cluster algorithms have been applied to the weighted dataset, the two graphics at the bottom of Figure 3.5 illustrate once again the predominance of the KCCA algorithm, which implements the must-link constraint simultaneously. The weighting seems to enhance the ability of the algorithm to find the grouping structure in the LFC even if the decision maker is not able to define a static threshold to drop the HFC. Again, the implementation of the Jaccard distance leads to higher cRand values compared to the Euclidean distance.

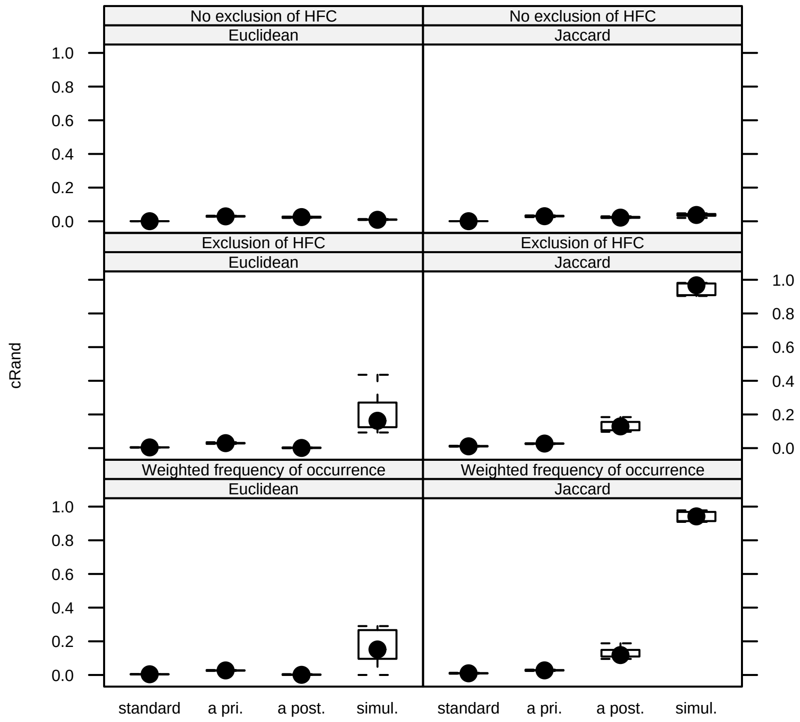

Figure 3.5: Identification of the 16 groups in the LFC using the eight cluster algorithms

Concerning the further analysis of the artificial data, we therefore consider only the output of the cluster algorithms using the Jaccard distance. Furthermore, due to the conceptual deficiencies of the a priori method and its weak results within the synthetic data, we exclude it from further examination as well and analyze just the three remaining methods – “standard”, “a posterori” and “simultaneous” – in attempting to find the groups in the data.

From a technical point of view, we want to identify the algorithm that produces the densest clusters. As stated above, the objective of partition algorithms is to gain homogeneous, dense groups which are heterogeneous relative to each other (cf. Aldenderfer and Blashfield 1984, Bacher 1996, Decker and Schimmelpfennig 2002). In the KCCA context, the closer each datapoint is arranged to its centroid, the lower is the total sum of distances between the datapoints of the clusters and their centers. An indicator for the overall heterogeneity of a found cluster solution is the total sum of squared distances. The squaring of distances gives more weight to larger distances. The higher the total sum of squared distances of all groups, the less dense are the resulting clusters of the corresponding algorithm.

After the total sum of squared distances for ten solutions of each method has been calculated, the average value of the "standard" method is 112,423.1, the value of the "a posteriori" method is 116,330.3 and the value of the "simultaneous" method is 114,549.3. Since the standard KCCA algorithm does not take constraints into account during point-centroid mapping, the resulting segments are usually more homogeneous and the corresponding total sum of distances becomes the lowest. Although the groups are quite dense, the output of the algorithm does not reflect the synthetic grouping structure as the low cRand values show. The group construction with the subsequent majority voting dissolves the dense, (local) optimal solution found with the standard method and generates less compact segments illustrated by a higher value of the total sum of squared distances. However, heterogeneous clusters imply that the centroids of the groups are less representative prototypical market baskets of an average cluster member. Nevertheless, compared to the standard KCCA algorithm, the a posteriori majority voting method approximates the real grouping structure of the synthetic scenario better. The value of the total sum of squared distances of the algorithm, which considers the customer sequences simultaneously, lies between the standard and the a posteriori majority voting algorithm. This method seems to produce more homogeneous segments compared to a posteriori majority voting. Its centroids represent the average market basket compositions of the associated customer group more accurately and reflect members' consumption habits more authentically. Besides the ability of this cluster algorithm to reveal the synthetic structure of the artificial data, this aspect also supports the simultaneous consideration of the constraints.

The output of all three cluster algorithms depends on the informative value of the customers' buying histories. It is very likely that the quality and stability of the cluster solutions will increase with the number of added transactions during the observation period because more information about customers' consumption behavior should enable a better segmentation of the buyers involved. To verify this assumption, we separate the synthetic dataset into eight subsamples growing cumulatively. We have chosen eight periods since the shortest sequence consists of eight synthetic transactions. In cases where the length of each buying history is not a multiple of the value of eight, the remaining transactions are forced to belong to the last period. Each subsample is built by adding the transactions of the following one to the existing transactions, and so on. Again, we cluster the eight resulting subsamples using the three algorithms and observe the allocation of the first subsample's entities over time. We compare the partition of each subsample with the previous one and calculate the corresponding cRand value. As shown in Figure 3.6, the cRand value of the simultaneous constrained clustering increases according to the length of the buying histories. This means that the entities of the first subsample point to a different centroid less frequently over time. In contrast, the stability of the partition gained from the a posteriori majority voting does not increase very much. Since the hard, rigid voting procedure can assign a customer to a different segment after each period, the customer-segment ordering often changes. The algorithm which considers the constraints simultaneously reaches quite a stable partition early on and is more robust to different lengths of the observation period than the other two algorithms. For retailers this is quite important since they have to rely on the clustering even when the datasets are collected over varying time intervals.

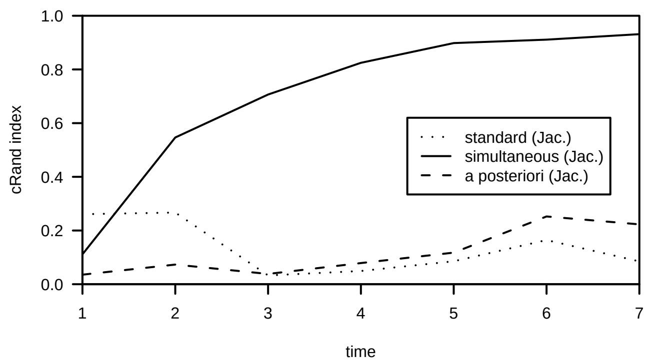

Figure 3.6: Stability of each cluster algorithm subject to growing buying histories

The analysis of the synthetic transaction data highlights the advantages of the modified cluster algorithm with simultaneously implemented must-link constraints in the described marketing context. It seems to partition customers' sparse binary transaction data from loyalty programs satisfactorily. In cases where decision makers would like to find grouping information within LFC products, it outperforms the KCCA algorithms usually applied (e.g. K-means). As shown, this depends on lowering the influence of the HFC products. Within the synthetic dataset, we have suggested deleting the HFC according to a static threshold or weighting the data matrix with the described measure (see Equation 3.4). In Chapter 4, both methods are applied to empirical data.

The selected constrained KCCA algorithm defines K centroid vectors p_k reflecting the market basket structure of the average segment's customer. These market basket prototypes provide the observer with quite detailed information about the purchase probabilities of category co-occurrences in the corresponding segment k. However, the centroid vectors do not give any information about exactly which categories are bought in combination within the segment's transactions. In addition, extracting product recommendations from these centroids is not straightforward since monetary values are not considered and the correlations between the single items are not specified in detail. Hence, after the segmentation step has been applied, the following steps of the application will determine valuable items for target marketing promotional activities.

3.2 STEP 2: Mining Segment-Specific Frequent Itemsets

According to Figure 3.1, the second step of the framework includes association rule mining to determine the exact purchase correlations in the transactions of a specific segment. Instead of the prototypical market basket vectors achieved with the KCCA algorithms, segment-specific frequent itemsets give a more precise view of the cross-category correlations. The retailer obtains itemsets containing the names of the included categories and a value returning the statistical strength of their correlation. In contrast, a table with the values for each category's single purchase frequency represents the market basket prototypes (cf. right-hand side of Table 2.5). Itemsets expose the found category correlations of a segment to the retailers in a more intuitive and understandable way. This contributes positively to the practical adaptability of the approach. Additionally, we are able to extract from the mined itemsets single category recommendations in a subsequent step.

Since the suggested KCCA algorithm includes buying histories as must-link constraints, all the customer transactions of a specific segment can be summarized into its transaction pool c_k. Within these market basket records, the APRIORI algorithm searches for frequent itemsets by defining the support value (see Equation 2.12) of each category. The segment-specific support of an itemset A in Equation 3.5 differs only in the denominator.

$$supp(A) = \frac{\mid \{x_n \in c_k \mid A \subseteq x_n\} \mid}{\mid c_k \mid} \tag{3.5}$$

As described in Section 2.2.2, the APRIORI algorithm usually combines association rules from the frequent itemsets by implementing another asymmetric measure of interestingness. Usually the confidence value denoted in Equation 2.13 is used. Due to the implementation of the confidence value, equal itemsets can denote different association rules since the conditional probability describes different correlation directions between the itemsets (i.e. $P(B \mid A) \neq P(A \mid B) \Rightarrow \{A\} \rightarrow \{B\} \neq \{B\} \rightarrow \{A\}$)). In this approach, we disregard association rules and focus on frequent itemsets since they are able to include a larger number of categories for grouping and valuating techniques. For example, the APRIORI's rule generation algorithm usually produces a lot of similar rules containing the same elements if real-world transaction datasets are used. The majority of these rules are permutations of the subsets, e.g. $\{wine\} \rightarrow \{water,\ beer\}$, $\{water\} \rightarrow \{wine,\ beer\}$ and $\{beer\} \rightarrow \{wine,\ water\}$. All these rules provide the same information on the question as to which categories build interesting purchase correlations. However, this information could also be obtained by determining the single frequent itemsets (e.g. $\{water,\ beer\}$, $\{wine,\ beer\}$, $\{water,\ wine\}$) or the maximal frequent itemset $\{water,\ wine,\ beer\}$ if the minimum support is exceeded. Since we are more interested in defining a high number of category correlations than in distinguishing them according to their correlation direction, frequent itemsets are mined.

Nevertheless, the approach considers the correlation direction derived from the confidence measure in a subsequent itemset filtering when it implements the all-confidence value introduced in Section 3.3.1.

3.3 STEP 3: Valuating Identified Cross-Category Purchases

The main objective of the third step is to separate the most interesting single categories from the found segment-specific frequent itemsets. In this context, we have to keep in mind that the number of mined associations depends on the pre-determined minimum support threshold. Usually, analysts prefer a low minimum support in order to detect less obvious associations within the transaction datasets as well (cf. Hui et al. 2006). On the other hand, the increasing number of mined frequent itemsets decreases the adaptability of the approach since not every itemset can be used for target marketing activities. To cope with a high number of frequent itemsets and to get a better understanding of the category correlations of the segment, filtering and grouping techniques are implemented. Finally, the remaining associations are transferred to the proposed optimization model, generating a list of single, segment-specific category recommendations.

3.3.1 Filtering Segment-Specific Itemsets

To separate the statistically valuable frequent itemsets from the non-valuable ones, we implement the all-confidence interest measure (cf. Omiecinski 2003, Hahsler, Grün and Hornik 2005). The all-confidence measure takes the asymmetric correlation direction of the confidence value into account (see Section 2.2.2). Unlike other filter measures, it is not applied in order to filter association rules but to filter frequent itemsets. The all-confidence determines for every itemset the minimum confidence of all rules which could be combined from the underlying subsets. For the frequent itemset C, it is described as follows (Omiecinski 2003):

$$allconf(C) = min\{conf(A \rightarrow B \mid \forall A, B \subset C, A \cup B = C, A \cap B = \emptyset)\} \tag{3.6}$$

Let us consider the example of the itemsets $\{champagne\}$ and $\{milk\}$ and assume that the frequent itemset $\{champagne, milk\}$ should be evaluated by the all-confidence measure (cf. Section 2.2.2). Since the all-confidence determines the confidence value of both possible association rules and chooses the minimum value, it would rate this frequent itemset as having minor importance due to the low confidence value of the rule $\{champagne\} \rightarrow \{milk\}$ (cf. Hui et al. 2006). The all-confidence incorporates the ability of the conditional probability to distinguish the direction of the purchase correlation.

As mentioned in Section 2.2.2, analysts have to be aware of finding many of such weakly-related cross-support patterns in transaction data with skewed support distributions. This becomes a particular problem in our approach. Since the partition step bundles users who purchase certain correlated categories disproportionately often, we also expect to find skewed support distributions in the pooled transaction data of a segment. Take for instance a segment containing the transactions of young parents. These customers will tend to buy a few baby products much more frequently than items in the other categories of the assortment. It is very likely that the baby-related items will build cross-support correlations with other items of the segment. Hence, the all-confidence measure seems to be a good choice to reduce the occurrence of such weakly-related cross-support associations in the transactions of our identified segments.

Filtering the segment-specific frequent itemsets supports the ability of the framework to focus on the category correlations that imply statistically interesting associations with the top products of a specific segment. Passing the non-filtered frequent itemsets on to the subsequent steps of the approach should enhance the quality of the output.

3.3.2 Grouping of Frequent Itemsets

As introduced in Section 2.2.2, grouping mined associations according to a similar frequent itemset composition is a useful descriptive technique for understanding and evaluating existing category correlations within the segment transactions. This approach conducts a distance-based clustering of found itemsets as originally introduced for association rules by Gupta et al. (1999). It is based on the distance metric shown in Equation 3.7. The metric equals the Jaccard distance described above (cf. Anderberg 1973, Müller-Hagedorn 2005):

$$D(A,B) = 1 - \frac{\mid m(A,B) \mid}{\mid m(A) \mid + \mid m(B) \mid - \mid m(A,B) \mid} \tag{3.7}$$

The expression $m(.)$ defines the absolute number of transactions that include the corresponding itemset. Hence, the distance between two disjoint itemsets A,B becomes smaller, the more often A,B occur in common transactions of the dataset. In order to examine the grouping structure, the distance measure is transferred to Ward's (1963) hierarchical cluster algorithm. The branches of a dendrogram visualize the grouping of the specific itemsets. In accordance with the experience of Toivonen et al. (1995), the hierarchical clustering is expected to group the subsets of itemsets as well as itemsets reflecting a comparable purpose of use, since the underlying products occur more often in a common part of the dataset's transactions.

3.3.3 Recommending Segment-Specific Categories

Finally, the approach should recommend a list of categories which promise to increase the earnings of the retailer if the items are used for appropriate target marketing campaigns in the segment. We have implemented a modified form of the PROFSET model designed by Brijs et al. (2004). It deals with two important aspects concerning a practical application in retailing (cf. Brijs, Goethals, Swinnen, Vanhoof and Wets 2000):

1. Taking the previous techniques into account, frequent itemsets are determined by statistical measures which compute the co-occurrence of categories. Due to the premise of binary-coded transactions, this is done without considering quantities – although there are very probable differences between the values of found associations because of their different profit margins (recall the example $\{champagne, oysters\}$ vs. $\{milk, cheese\}$ in Section 2.1.2). Hence, the proposed framework re-imports the quantities and prices of the categories for the valuation of itemsets in the last step of the framework. The re-import ensures a good balance between the increasing workload of the computational capacities on the one hand and the increasing explanatory power of the results on the other hand.

2. For retailers it is quite difficult to deduce concrete marketing decisions from frequent itemsets with a length $l \geq 2$ since it is unclear which of the included categories contributes in what way to the purchase incidence. Managers are more likely to be interested in knowing exactly which single category will maximize the utility of a planned campaign. In addition, retailers are only able to recommend a fixed number of categories to the segments' customers. For example, marketing leaflets or special display shelves can only present a predefined number of special offers, or the promotion budget is restricted. Therefore, the framework needs to reveal a *limited* list of single categories worth featuring with segment-specific target marketing.

Summarizing these thoughts, the challenge facing the PROFSET model is that it must deal with three requirements. First, the selected items have to build purchase correlations with other items of the assortment to stimulate cross-selling. Second, the items need to have a high monetary value. Third, the model has to define a limited number of items. The objective of optimizing the revenue of the promotional campaign depends on the optimal choice of featured categories. Since not all items can be used (third requirement), the model has to choose between the cross-selling attribute of an item (first requirement) or the height of its monetary value (second requirement). This is a decision problem in the sense described by Church (1936). For example, the items with the highest cross-selling potential do not necessarily show the highest monetary values in every case. The model has to define the items by considering all three requirements in the best way. A two-step procedure solves this problem.

In the first step, the model determines all frequent itemsets supporting the overall profit with high margins. Let $v(x_n)$ denote the profit margin of a transaction $x_n \in c_k$ expressed by Equation 3.8:

$$v(x_n) = \sum_{j \in x_n} U(j) \tag{3.8}$$

The function $U(j)$ shows in what way category j contributes to the retailer's utility. If the sales profit has to be optimized, the following equation denotes the utility function:

$$U(j) = \{PRICE(j) - COSTS(j)\} * QUANTITY(j) \tag{3.9}$$

Comparable to the profit margin of a transaction, $v(A)$ denotes the profit margin of a frequent itemset A achieved with a single transaction $x_n \in c_k$. The corresponding gross profit margin $V(A)$ is the sum of all $v(A)$ extracted from c_k's market baskets containing itemset A:

$$V(A) = \sum_{A \in c_k} v(A) \tag{3.10}$$

If a single transaction x_n equals a frequent itemset A, $v(A)$ corresponds to $v(x_n)$. With regard to transaction data, only a few market baskets usually exist which show the same category combination of a mined frequent itemset. In most instances, a transaction contains many different itemsets. Since we have to determine the profit margin of every itemset in order to be able to distinguish between their values, the approach needs to calculate the margin share of the frequent itemsets included in the corresponding market basket. This is done by identifying the maximal frequent itemsets of a transaction, selecting them subject to a probability distribution, and computing their profit margins. For instance, A, B are frequent itemsets of the itemset collection F with $A, B \in F$. A frequent itemset A_{max} is called maximal if it is not a subset of another itemset (Gouda and Zaki 2005). For illustration purposes, consider an example market basket x_{100}. It includes *white wine*, *red wine*, *appetizers* and *bread*. Moreover, Table 3.4 lists five fictitiously identified frequent itemsets (cf. Brijs et al. 2000):

No.	Frequent itemset (FI)	Support	maxFI	x_{100} = maxFI
1.	{white wine}	0.3	no	no
2.	{red wine}	0.2	no	no
3.	{appetizer}	0.8	no	no
4.	{white wine, red wine}	0.15	yes	no
5.	{red wine, appetizer}	0.05	yes	no

Table 3.4: Sample of five fictitious frequent itemsets (cf. Brijs et al. 2000)

Transaction x_{100} contains all frequent itemsets of Table 3.4. First, the procedure has to work out the two maximal frequent itemsets No. 4 and No. 5. Both do not equal x_{100} ($v(x_{100}) \neq v(\{white\ wine,\ red\ wine\}) \wedge v(x_{100}) \neq v(\{red\ wine,\ appetizer\})$). Since the two itemsets overlap each other in x_{100}, it is not clear which one should be assigned the corresponding share of the profit margin included in the transaction. The decision is based on the following assumption: the support value of itemset No. 4 is three times higher than that of itemset No. 5 (0.15 > 0.05). Assuming the support value corresponds to the itemset's probability of being picked by the customer, Equation 3.11 shows a possible probability choice function Θ (Brijs et al. 2000):

$$\Theta_{x_n}(A_{max}) = \frac{support(A_{max})}{\sum_{B_{max} \in x_n} support(B_{max})} \tag{3.11}$$

Since the probability of choice depends on Θ for every transaction x_n, itemset No. 4 will be assigned the proportionate margins more frequently on average. After the profit margin has been calculated, the corresponding maximal frequent itemset is deleted from x_{100} and the procedure is repeated until the transaction no longer contains any frequent itemsets. This procedure is carried out for all transactions of c_k until each frequent itemset's gross profit margin has been determined.

The second step is to derive the single categories from the itemsets by considering the decision problem of choosing the most profitable items from among the found category associations. For this, the slightly modified PROFSET model calculates the profit values of each category. It solves the optimization problem according to the retailer's overall utility function and restrictive constraints as formulated in Equations 3.12 to 3.14.

$$MAX \quad \{\sum_{A \in F} V(A) * S_A - \sum_{j \in F} COSTS(j) * Q_j\}, \ \ with \ \ Q_j, S_A \in \{0,1\} \tag{3.12}$$

$$subject \ \ to: \quad \forall A \in F, \forall j \in A : Q_j \geq S_A \tag{3.13}$$

$$\sum_{j \in F} Q_j = \Phi \tag{3.14}$$

The target function 3.12 is deduced from the utility function $U(j)$ in Equation 3.9. A frequent itemset A of the frequent itemset collection F contributes to the overall sales profit positively if its binary choice variable S_A equals a value of 1. In a negative sense, the corresponding costs of each category included in the frequent itemset reduce the profit. Supposing S_A is chosen, the constraint No. 3.13 ensures that the choice variables Q_j of the included categories will also become 1. Constraint No. 3.14 restricts the number of determined categories since the number of featured or recommended categories is limited due to space or budget restrictions. After a branch-and-bound algorithm solves this integer optimization problem, the solution determines Φ variables Q_j which point to the categories maximizing the target function (cf. Dakin 1965).

Sometimes retailers would like to include specific products in the list of recommendations, e.g. products that are characteristic for the store image. Although the optimization routine would not determine them, additional constraints can force the model to do so by setting the related decision variable of the corresponding items to a value of 1 (Brijs et al. 2000, Brijs et al. 2004).

Consequently, the three-step procedure of the approach examines possible cross-category purchase correlations at each stage of the process. Apart from customer segmentation, it suggests a limited list of single categories for segment-specific target marketing campaigns. The empirical application described in the following chapter simulates the results with real-world data if the items of the approach are used for such campaigns.

4 Empirical Application

To demonstrate the application of the framework, we analyze a transaction dataset obtained from a supermarket chain, which will remain anonymous. We aim at achieving two objectives: first, we wish to identify customer groups and the corresponding segment-specific categories worth being promoted by conducting the methodological steps described in Chapter 3 (cf. Section 4.2 to Section 4.4). Second, we want to analyze the potential profitability of the approach by describing how to use its outcome for a target marketing initiative. In addition, we try to estimate the profit generated by the target marketing campaign compared to a commonly conducted customer-unspecific promotion heuristic (cf. Section 4.5).

4.1 Data Description

The data contains more than 1.4 million transactions made over one year by 56,000 customers who chose among 268 major categories (e.g. ice cream, beef, white wine, red/rosé wine etc.). The supermarket's customers took part in a loyalty program which stored each transaction in the program members' transaction histories. Since every supermarket owned by the company is connected to the dataset, the transactions were linked to the customer even if each purchase was made at a different store. On average, customers carried out around 26 purchase occasions with a median market basket size of six categories over the one-year observation period. The prices and quantities of the included categories were recorded for every transaction. From another dataset, we were able to calculate for every category its approximate gross profit margin. Hence it was possible to determine the profit to the retailer achieved with a certain category.

To deal with the dimensionality of the data, we draw two samples with 3,000 customers each. The first sample is used to derive some preliminary information in order to be able to make some modifications to the samples. These modifications should increase the quality and stability of the analysis of the second sample. For instance, since we are interested only in customers who show a sufficiently long buying sequence which represents their interest in special categories, we exclude all customers with fewer than six transactions per year. This affects the lower 20 percent of all customers. In addition, the upper five percent with extremely long sequences are excluded from the data for stability reasons, e.g. customers who bought at the supermarket nearly every day. Of course, these customers are of great importance for every retailer and might be treated with even more personalized CRM activities. Here, however, we want to focus on promotional campaigns which target the majority of common buyers.

After the first sample has been reduced along these lines, 2,250 customers remain. We split these customers into three subsamples of equal size (i.e. $3 * 750$ customers) for further examination (see below). The 3,000 customers in the second sample are chosen taking the above-mentioned modifications into consideration from the beginning. Hence, no buyers from the second 3,000-customer sample have to be excluded.

The second data modification regards our assumptions about the grouping structure within the empirical data. As we have discussed above, the customers may differ more precisely in buying items from the long-tail of retailers' assortments (i.e. from the LFC). Hence, we try to reveal the category purchase correlations within this part of the assortment. In Chapter 3, we compared three segmentation approaches which seem to support the identification of this particular grouping structure. Section 3.1.4 has shown that a constrained KCCA algorithm leads to the most appropriate results in artificial data when it considers the constraints simultaneously. However, the quality of the partition depends on the methods used for lowering the influence of the HFC with data modification techniques. As shown in Section 3.1, we differ between excluding the HFC according to a threshold and according to a weight, taking categories' support into account.

In our empirical application of the approach, we apply the chosen constrained KCCA algorithm to both modifications of the dataset:

- Section 4.2.1 describes the partition gained from the second sample, in which HFC were deleted if the category occurred in more than 10% of all transactions (horizontal dotted line in Figure 4.1). This affected 52 categories (left-hand side of the vertical dotted line in Figure 4.1). Transactions were deleted which did not include any of the remaining 216 categories.

- In Section 4.2.2 the data matrix of the second sample is multiplied with the weighting value introduced in Section 3.1.4. According to this weight, the purchase frequencies of categories are lowered according to the height of their support value. Since no items are excluded, this method comprises all 268 categories.

Since we want to test the predictive characteristic of our approach in Section 4.5, we hold back the transactions of the second sample recorded in months eleven and twelve. With the first part of the sample (months one to ten), we determine the cluster membership of each customer and calculate the expected revenue in the following two months. Table 4.1 summarizes the generated samples and shows their usage in this chapter.

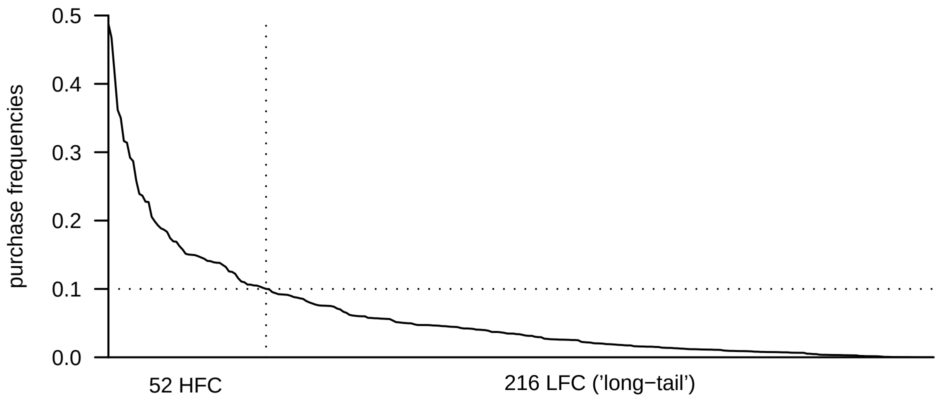

Figure 4.1: Purchase frequencies of 268 categories in descending order

	Split into	No. of customers	Application	Section
Sample 1	Subsample 1	750	Initialization of Subsample 2, determine K	4.2
	Subsample 2	750	Initialization of Subsample 3	4.2
	Subsample 3	750	Initialization of Sample 2	4.2
	Residual	750	Deleted	4.2
Sample 2	–	3000	Remaining analysis	4.2-4.5
	Sample (Jan-Oct)	–	Determination of segments and items	4.4
	Sample (Nov-Dec)	–	Simulation of profitability	4.5
		Total 6000		

Table 4.1: Overview of generated samples and their usage

While the preliminary data analysis (initialization, determining *K*) is done with the subsamples of the first sample, we conduct the full simulation with the second sample. The necessary analysis of the data is done with *R*, which is a programming language and environment for statistical computing (R Development Core Team 2005). Since the raw dataset is not compatible with the input format of the statistical software, some previous data preparation has to be done. For converting the raw ASCII-coded receipt data into *R* compliant input files, the program language PERL is used. In addition, dataset storing is done with the free relational database system MySQL whose query language provides a flexible tool to extract and sort the data. Most data processing was done on a server machine (Intel Pentium 4 CPU 3.0 Ghz and 2056 MB RAM) with the Linux operating system (Debian 4.0 Etch) installed.

4.2 Identifying the Customer Clusters

As shown in Figure 3.1, the first goal is to identify different customer groups within the empirical data. According to the modification of the data, we apply the constrained KCCA algorithm to the cropped and to the weighted dataset.

4.2.1 Excluding the HFC from the Dataset

Before starting the cluster analysis, the first modification excludes the HFC from the samples subject to a threshold ($supp(j) \geq 0.1$). If the remaining transactions do not include any further categories of the long-tail range, they can be deleted. Even if no transactions are deleted, the dimensionality of the dataset is reduced since only 216 instead of 268 columns of X_N are left over.

The quality of the results gained from exploratory segmentation algorithms often depends on a preliminary analysis of the data provided. Hence, we conduct some pre-tests with the first sample to collect information about the dataset's structure. This should help to partition the entities in the subsequent steps appropriately.

In keeping with the general objective of cluster analysis and our expectations as stated in Section 3.1.4, we want to validate the heterogeneity of the partitions gained from the different algorithms. We determine the heterogeneity in artificial binary transaction data with the total sum of distances of each cluster. The average value of the total sum of distances gained from several solutions indicates that the highest cluster heterogeneity is found for the *a posteriori majority voting* method and the lowest for the standard KCCA method. The average value of the suggested constrained cluster algorithm lies between these values. To verify this result in the empirical data, the three cluster algorithms are applied to the first subsample according to an increasing k. Again, we use the total sum of squared distances to highlight more clearly the differences between the algorithms' homogeneity in the corresponding graphs of Figure 4.2.

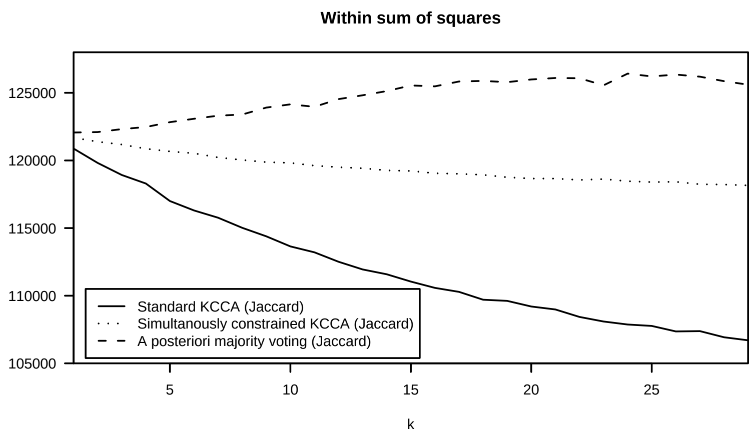

Figure 4.2: Comparing the total sum of squared distances for an increasing number of clusters K

Figure 4.2 shows that the values of the total sum of squared distances of the KCCA algorithm when the constraints are considered simultaneously is always lower for an increasing k compared to the values of the *a posteriori majority voting* technique. However, the values are higher compared to the standard KCCA algorithm with constraints ignored. This corresponds to the results extracted from the artificial data. As discussed in Section 3.1.4, the suggested constrained KCCA algorithm finds the given structure in the LFC of the artificial data quite well. At the same time, it includes the objective to compose relatively homogeneous clusters. Although we have confirmed our expectations about cluster homogeneity, we are not able to verify whether the corresponding solution represents the "natural" grouping information actually hidden in the data. Nevertheless, the subsequent found customer segments of the empirical data seem to be quite useful from a target marketer's point of view.

One important and still unsolved task of all KCCA algorithms is the determination of the number of groups K (Aldenderfer and Blashfield 1984, Milligan and Cooper 1985). In contrast to most hierarchical cluster algorithms, KCCA forces the analyst to predefine a priori the number of groups in the data being studied. This information is not present in most real-world situations. Nevertheless, several statistical methods support decision makers in estimating K. Common techniques propose the calculation of the total sum of distances subject to an increasing k, followed by the inspection of the ensuing graph for an 'elbow'-criterion (cf. Thorndike 1953, Bacher 1996, Backhaus et al. 2006). An abrupt low total sum of distances indicates that the density between the segment's entities does not become much higher after the corresponding k-value has been reached. All three subsamples are clustered up to 15 times for each $k \in [2,30]$ to reduce the risk of finding a weak local optimum (cf. Chapter 2). Again, the solutions with the lowest total sum of distances are kept. Since the outcome of KCCA depends on the initialization of the algorithms (see Section 2.2.3, Step 1 and Section 3.1.3, Step 1 of the iterative procedures), the quality of the partition can increase if the initialization is inherited from a former appropriate cluster solution (cf. Gordon and Vichi 1998). Hence, the initialization of each subsample and the second 3000-customer sample are the centroids of the previous sample.

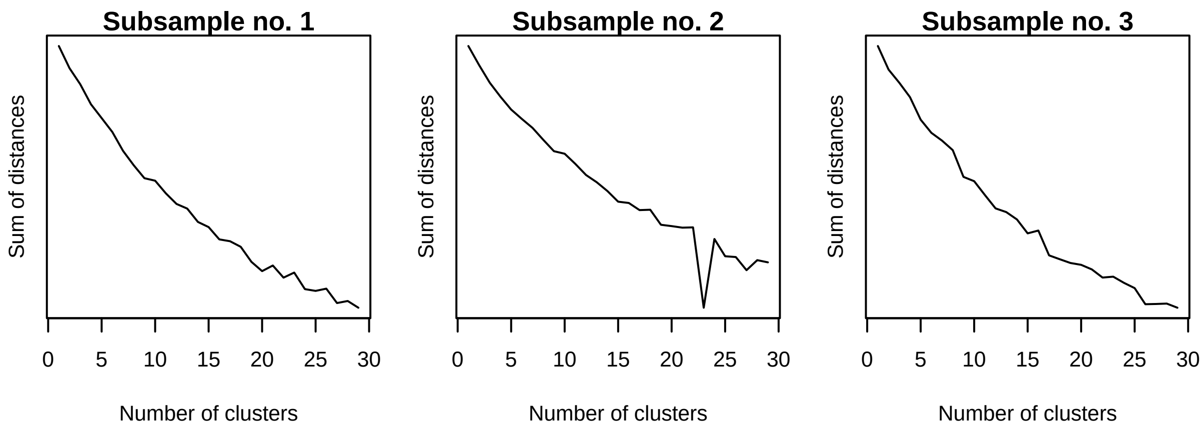

Figure 4.3: Total sum of distances for increasing number of clusters

Figure 4.3 shows the total sum of distances of each subsample's cluster solutions if k increases. Since none of the graphs indicates a clear elbow-like curve, determining the number of K by visual inspection seems unsuitable for our empirical data. Notice that the second graph of Figure 4.3 depicts a negative peak for a value of $k = 24$. This does not provide us with a appropriate indication of the number of groups since it is just an outlier concerning the sum of distances in this subsample. Directly after this local minimum, the total sum of distances increases for higher values of k, implying that cluster heterogeneity increases again.

Instead of considering the graphs visually, we would like to calculate more objective measures to determine the number of groups. For this purpose, a wide variety of cluster validation indexes have been developed recently (for an overview, see Milligan and Cooper 1985). Many of these indexes look at the within-cluster density and between-clusters separation subject to an increasing k. Dimitriadou, Dolnicar and Weingessel (2002) present an extensive analysis of several measures and rate them subject to their ability to recognize the given number of clusters in a synthetic binary dataset. In particular, the Davies-Bouldin index db_k is able to find the number of pre-determined groups within the artificial data of these authors (Davies and Bouldin 1979). We record for each $k = [2,\ 3, \ldots 30]$ the best cluster solutions out of ten iterations. This is repeated fifteen times which leads to fifteen sets of 29 Davies-Bouldin indices. In a second step, the “positive elbow” can be calculated for each of the fifteen sets according to $min_k((db_{k+1} - db_k) - (db_k - db_{k-1}))$. Since this value determines the recommended k-value, we choose the number of groups highlighted most often during the fifteen passes.

The upper-left graph in Figure 4.4 shows that the Davies-Bouldin index determines five times the solution with $k = 30$ groups, as shown by the highest bar in the plot. Indicating the maximal number of groups within the considered interval seems to produce a quite unwieldy number of customer segments from a managerial point of view. Hence, we tried the Xu and the SSI/SSIW indices also. Both indices are structured differently compared to the Davies Bouldin index (cf. Xu 1997, Mazanec 2001, Franke and Mazanec 2006). As shown by the other three plots in Figure 4.4, they all present quite vague and ambiguous values of K which demonstrates the difficulties of finding an appropriate K for binary transaction data.

Since the foregoing techniques do not solve the problem satisfactorily, we consider Rand's (1971) index for evaluating cluster solutions to approximate K. As described in Section 3.1.4, the Rand index is used to define the agreement of different cluster algorithms. Here, the index compares the solution of the same KCCA algorithm subject to different values of K within the first subsample. Although we know that the concurrence increases with the number of groups, it is not known for which k the cluster agreement ceases to change significantly. To determine when this takes place, the k-solution is compared to the $k+1$-partition. Figure 4.5 shows the corresponding Rand values.

As shown by the underproportionally increasing graph, the Rand value levels off after the cluster agreement between solution $k = 11$ and $k = 12$ has been calculated. In other words, the arrangement of the

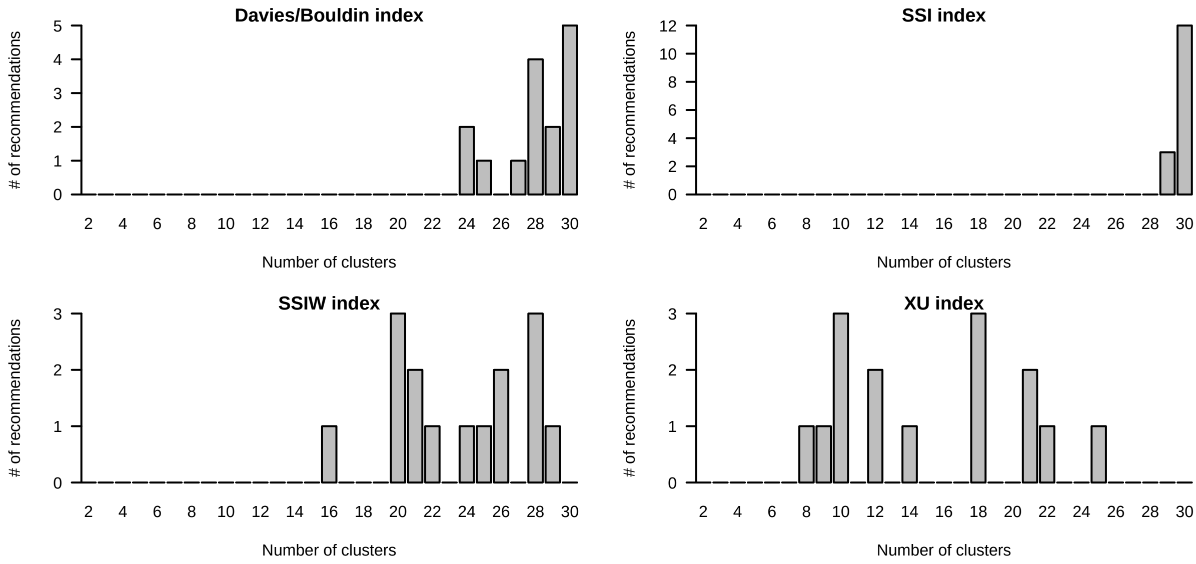

Figure 4.4: Recommended value of K according to index voting within the last subsample

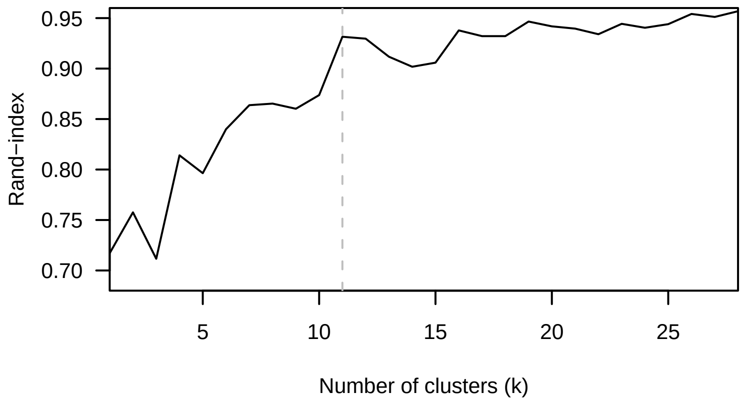

Figure 4.5: Comparing k- to $k+1$-cluster solutions with the Rand index

data points would not change radically if a further cluster were to be opened. When k reaches a value of eleven, the cluster agreement between both solutions is quite high, suggesting that $k = 11$ could be an adequate value for K. This also seems appropriate since lower numbers of segments seem to be more manageable for retail managers in most business cases and should be favored for practical applications (cf. Reutterer et al. 2006).

The application of the different techniques demonstrated that it is not straightforward to define K according to objective criteria such as indices. Different methods of determining K very often lead to different results in practice. Sometimes it seems more useful to determine K according to practical considerations, e.g. the minimum/maximum number of customer groups which can be treated with the available budget. Another method is to calculate solutions with different K values and rate the defined partition with regard to the structure or usability of the found clusters.

After K has been defined and the partition algorithm has been conducted, we consider the sizes of the generated groups and give them names according to the kinds of categories which occur most frequently within each cluster. The black bars of Figure 4.6 indicate the number of transactions of the second sample (months 1-10) that belong to each cluster after the algorithm has stopped. Although a bigger segment with about 20% of all transactions ($k = 6$, mix cluster) and a smaller one ($k = 10$, bar-products) exist, the rest of the entities are distributed quite equally among the other nine segments.

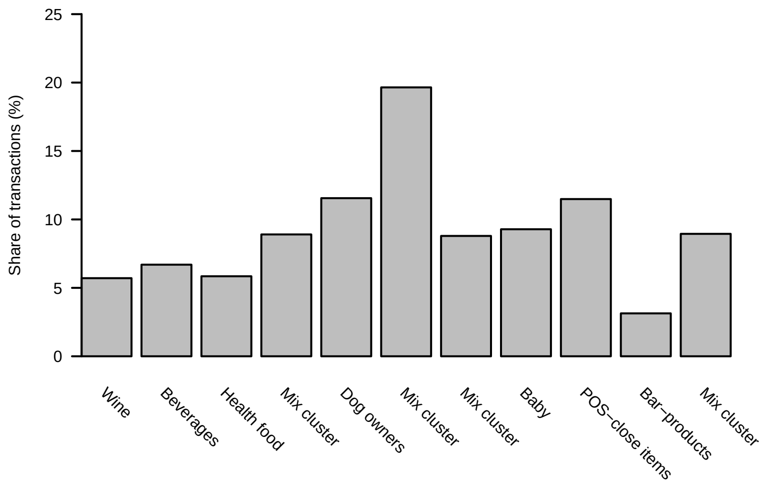

Figure 4.6: Transactions in each cluster (sample no. 2, months 1-10)

We examine four distinctive customer segments in more detail. Information concerning the rest of the customer segments can be found in the Appendix (see Section A.4.1). Figures 4.7 and Figure 4.8 show the graphical representations of the two market basket prototypes generated from segments $k = 8$ and $k = 1$. The plots give a first insight into the cross-category purchase relationships of the households in both segments. In contrast to the black solid line on the left-hand side representing the overall purchase frequencies within the whole sample, the 216 light-gray bars correspond to the category purchase frequencies within the segment being considered. The different arrangement of the high and low light-gray bars on the left side in Figure 4.7 and Figure 4.8 illustrate that the household members of both segments are interested in quite different category combinations. To identify the items belonging to the peaks, the ten most frequently purchased categories of each segment are shown on the right-hand side of both figures. The households in segment $k = 8$ seem to focus on baby food and care categories since these products are purchased at a higher than average rate. The households in the other segment combine different kinds of wine. Hence, we labeled the first segment the “baby” and the second segment the “wine” cluster.

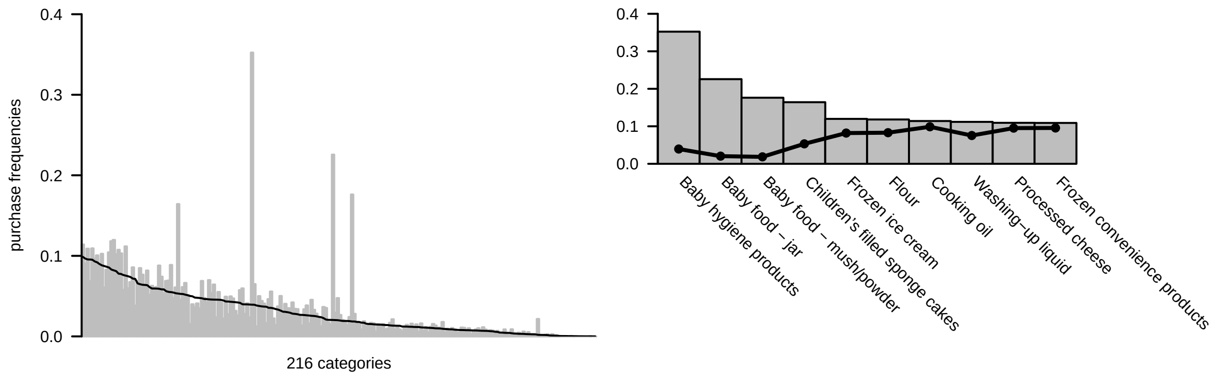

Figure 4.7: Graphical illustration of the prototypical market basket of segment $k = 8$ (baby cluster) with HFC excluded

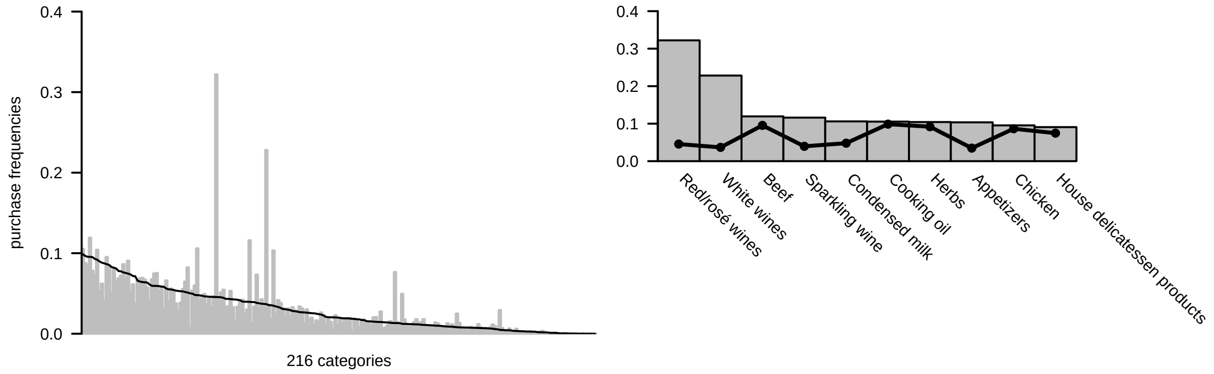

Figure 4.8: Graphical illustration of the prototypical market basket of segment $k = 1$ (wine cluster) with HFC excluded

For example, in contrast to the overall purchase probability of 3.86%, red/rosé wines occur at a rate about ten times higher than average in a basket from a wine-cluster household (red/rosé wine's segment-specific purchase frequency is 32.21%). Since the class means are used as new centers during the iterative cluster procedure, the prototype values are considered as conditional choice probabilities of the corresponding categories (compare Section 3.1). In other words, a typical household of the baby-cluster buys baby hygiene products with a probability of 35.24%, baby food in a jar with a probability of 22.82% and baby food mush/powder with a probability of 17.62%.

The information derived from these prototypes is quite helpful for target marketers, since the segmentation according to the purchasing behavior reveals the interest of segment members in certain categories (such as baby products or kinds of wine). In the same manner, the prototype of the "healthy" product segment shows the interest of its members in buying organic products. Besides organic or wholemeal products, these customers consume some kind of fruit more often on average. The graphical representation of the corresponding prototype is plotted in Figure 4.9.

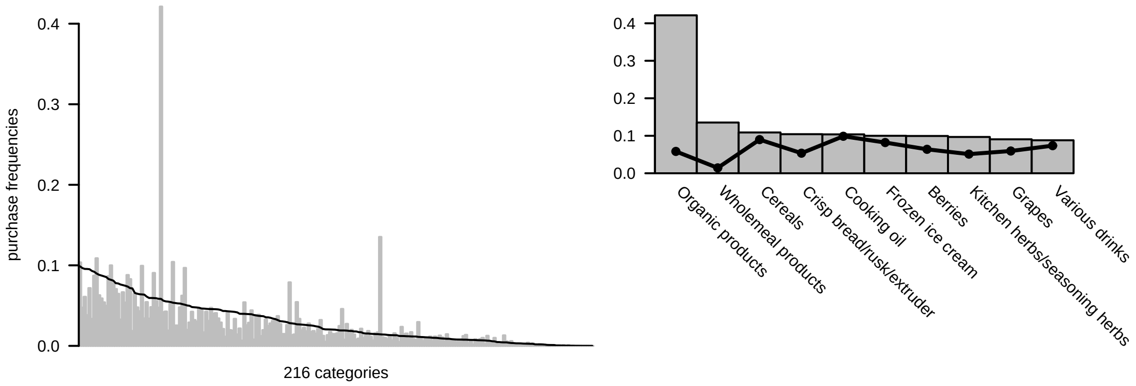

Figure 4.9: Graphical illustration of the prototypical market basket of segment $k = 3$ (healthy cluster) with HFC excluded

Figure 4.10 illustrates the prototype of another quite interesting segment. It includes the customers who buy newspapers and items which are usually arranged near the cash registers such as ice cream and candy. Hence, we call it the segment with "POS-close" items.

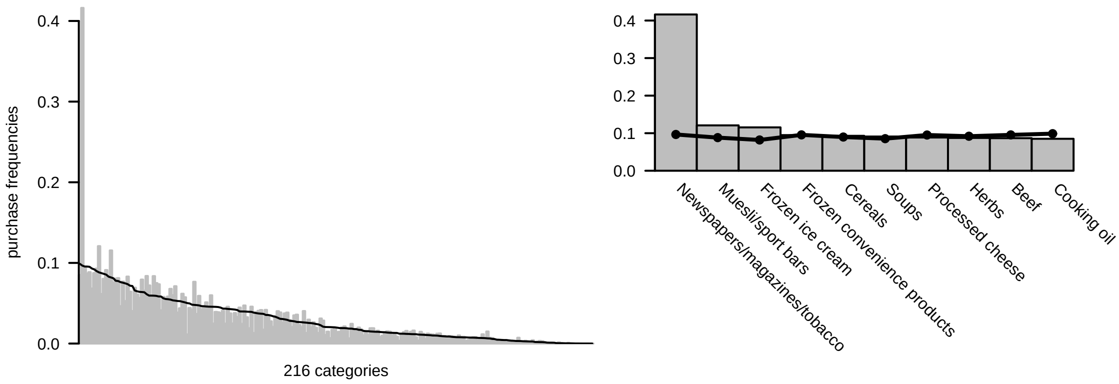

Figure 4.10: Graphical illustration of the prototypical market basket of segment $k = 9$ (POS-close items cluster) with HFC excluded

In addition to these four segments, other market basket prototypes contain category combinations related to meat (e.g. beef, chicken, other kinds of meat, etc.) or beverages (e.g. soda, lemonade, water, etc.) and are easy to label with a generic term. Naming the segments according to the type of category most often included is not always straightforward. For example, four of the eleven segments do not contain categories showing an obvious association. However, the corresponding market basket prototypes of these mix clusters may identify customer segments which buy unexpected, unusual category combinations.

This stresses the exploratory methodology of the proposed approach.

4.2.2 Weighting the Occurrence of HFC in the Dataset

In this section, we apply the suggested constrained KCCA algorithm to the weighted data matrix as discussed in Section 3.1.4 of Chapter 3. Multiplying every dichotomous value with the corresponding weight w_j (see Equation 3.4) will turn the binary values into metric ones. Since the distance measure according to Equation 2.2 can also deal with non-binary transaction vectors, the benefits of the Jaccard distance are retained. The weighting method lowers the impact of the HFC during the partition and should help to reveal the less-strong grouping structure within the LFC, similar to the exclusion of the HFC in Section 4.2.1. But compared to the exclusion, retailers do not need to define a support threshold which defines the categories to be dropped. From a practitioner's point of view, this facilitates the approach's application.

Regarding the determination of the number of clusters K, we use a technique comparable to that described in Section 4.2.1. Instead of the Rand-index, we implement the corrected Rand-index (cf. Section 3.1.4) since some preliminary tests have shown that the cluster sizes are quite different if the weighted data is partitioned (see right-hand side of Figure 4.11). When considering clusters' agreement by chance when several smaller and one huge cluster are present, the corrected Rand index might be more appropriate. Although the grouping structure should theoretically be the same as that in the foregoing section, and the analyst might again expect to find eleven customer segments, even slight modifications of the method or the data often lead to different values of K in practice.

The left-hand side of Figure 4.11 illustrates the graph which results from comparing the k and $k+1$ cluster solutions for an increasing $k \in [2,30]$. The two highest peaks determine $K = 3$ or $K = 9$ and define the highest correspondence to the subsequent cluster solution. Since three customer groups would not be a useful value for most real-world target-marketing applications, we decide to examine nine segments in more detail.

The right-hand side of Figure 4.11 depicts the cluster sizes and the cluster names derived from the categories with the highest support values. Comparing the bars of the segments to the ones achieved in Figure 4.6 of Section 4.2.1, the cluster sizes of this solution are distributed much less equally. Since it defines one very large, a medium sized and many tiny clusters, this is a first indication that the weighting value is not able to lower the impact of the HFC sufficiently. The disadvantageous cluster sizes persist even though we exclude five categories manually such as sellable shopping bags or baskets, as well as packaging materials at the cheese or the meat counters. Since these categories have high support values, we expect them to be responsible for the big mix cluster even though the weighting factor should reduce their impact on the definition of category correlations. These categories belong to the 52 categories of

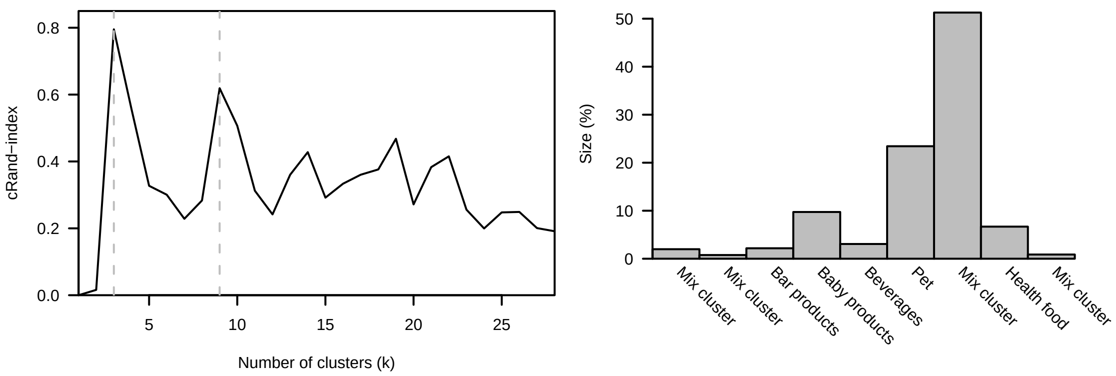

Figure 4.11: Determination of K and cluster sizes after the partitioning of the weighted data matrix

the HFC that we did not consider in the previous section either. In addition, they can be dropped due to their minor importance with respect to the analysis of purchase correlations. Anyhow, this does not lead to a more appropriate distribution of the cluster sizes.

Despite these problems, in Figures 4.12 to 4.14 we consider three prototypes of the resulting cluster solution that present some interesting category correlations. Similarly to the figures of the prototypes in the previous section, the solid black line represents the overall purchase probability. Again, the light-gray bars correspond to the segment-specific purchase frequencies of the categories. Notice that these figures also include the long-tail categories (except for the five manually dropped categories).

The right-hand sides of Figure 4.12, 4.13 and 4.14 are slightly different to the graphical parts of the previous section. In contrast to the inclusion of the black solid line, the light-gray bars represent the highest *deviations* between categories' overall and categories' segment-specific purchase frequencies. This method excludes categories from the graphical representation which occur with a similar probability in a common market basket of the supermarket. For example, consider Figure 4.8 of Section 4.2.1. The third highest gray bar on the right-hand side belongs to the beef category. The difference between the overall and the segment-specific purchase frequency is quite small. This could mean that the typical member of the wine segment is usually not interested in beef at a higher than average rate. Hence, the following graphical technique to present the prototypes would not list the beef category.

Considering the highest segment-specific deviations of purchase frequencies, the categories of the prototypes define the baby and the health food cluster already known from the previous section (cf. Figure 4.7 and 4.9). By comparing the two illustrations in Figure 4.12 and Figure 4.14 with the foregoing two, we can see that the composition of the categories within the prototypical market basket is similar but not equal to the ones in Figures 4.7 and 4.9. For example, the health food cluster contains, besides organic and wholemeal products, more fruits and vegetables. The differences in the segment-specific category compositions shows that the analyst has to be aware of finding varying solutions when he uses different cluster techniques.

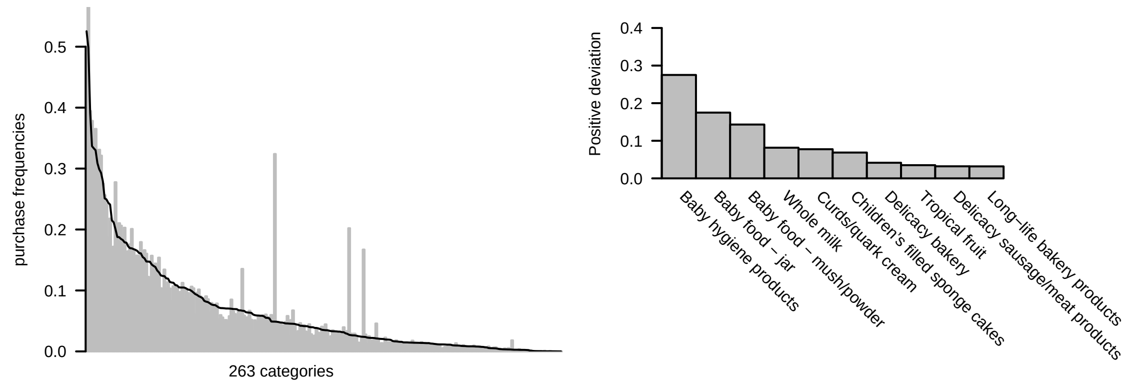

Figure 4.12: Graphical illustration of the prototypical market basket of the baby segment derived from the weighted data matrix

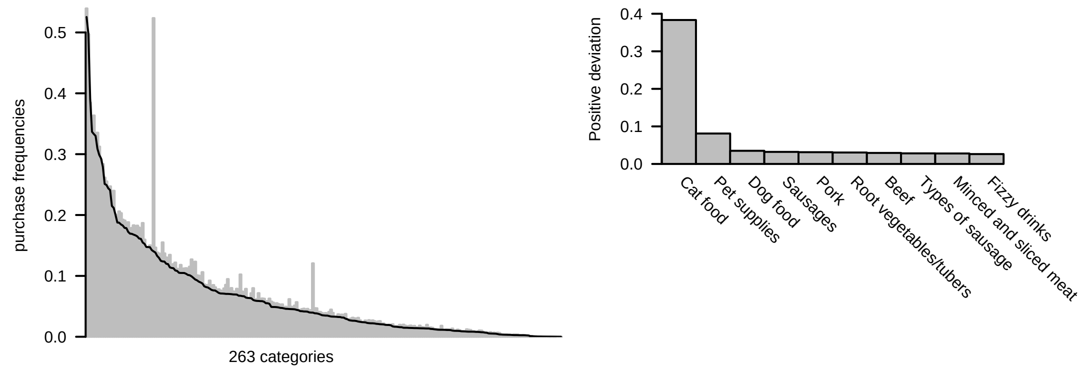

Figure 4.13: Graphical illustration of the prototypical market basket of the pet cluster derived from the weighted data matrix

Instead of considering the highest positive deviations from categories' overall purchase frequencies, it might be useful to look at the categories which are bought less often on average in a segment. We replace the right-hand part of Figures 4.13 and 4.14 with the two plots in Figure 4.15. The white bars represent the least interesting categories of each cluster, and the marketing efforts of the decision maker could consider such items separately. For instance, the highest white bars in the health food cluster mainly mark quite unhealthy products such as sweet fizzy drinks, varieties of meat or confectionery. These items do not seem to correspond to cluster members' usual purchase habits of buying healthy food. Similarly, the pet households are less interested in purchasing baby-related products – even though the small bars of the baby-related products do not indicate a high deviation from the average purchase frequencies.

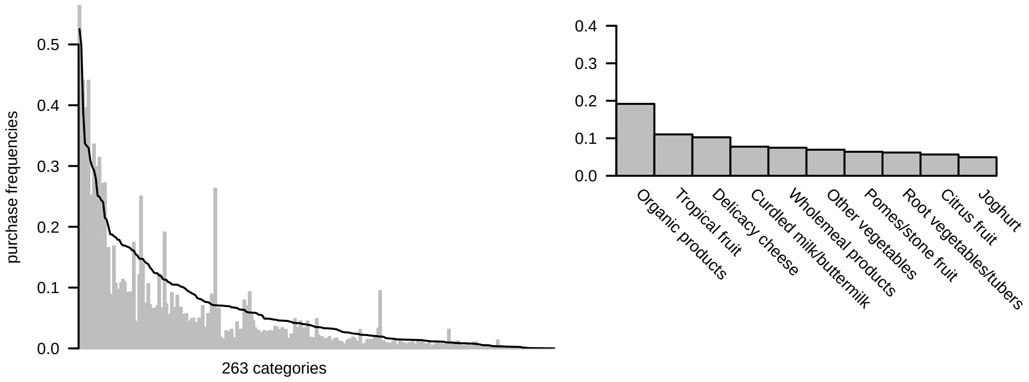

Figure 4.14: Graphical illustration of the prototypical market basket of the health food cluster derived from the weighted data matrix

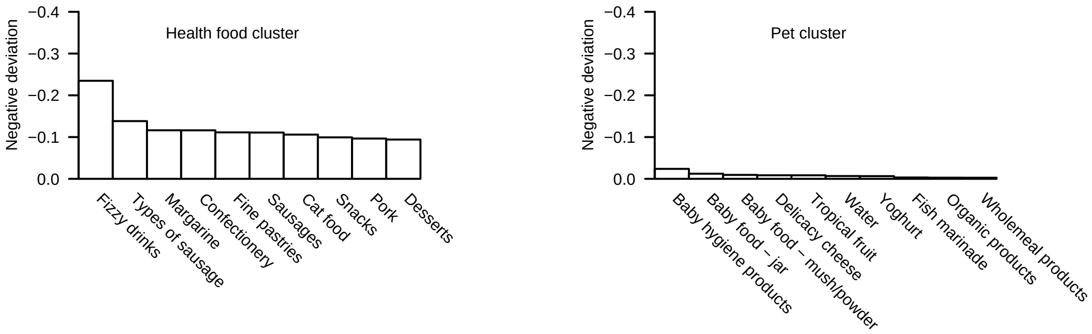

Figure 4.15: Highest negative deviations from overall categories' purchase frequencies in the health food cluster (left-hand side) and the pet cluster (right-hand side)

4.2.3 Summary of the Cluster Identification

To identify the customer groups, we applied the KCCA algorithm which considers the constraints simultaneously, as introduced in Chapter 3. Instead of deleting the HFC categories according to a predefined minimum support as in Section 4.2.1, the effect of the HFC categories on the clustering could also be limited by the weighting method (cf. Section 4.2.2). Due to a characteristic category composition, the algorithm identifies in both datasets customer groups which seem to be interesting from a target marketer's point of view. Comparing the outcome of the two applications, some clusters correspond to each other (e.g. the baby segments) whereas other clusters have not been revealed if the weighting is applied. This shows that the data modification can cause different cluster solutions.

In the following sections, we will use the reduced data matrix which does not include the HFC categories. Instead of the dataset weighting, the exclusion of the HFC ensures that the impact on category correlations is eliminated completely in this range of the assortment. Moreover, the cluster sizes are distributed more evenly compared to the cluster sizes of the solution that depends on the weighted data. From a target marketer's point of view, this is preferable since communicating with very small customer groups is expensive and communicating with extremely large groups contradicts the idea of target marketing. Although the weighting method has the advantage of not needing a static threshold to separate the HFC from the LFC, practitioners often know their HFC categories quite well and are able to exclude them effectively by defining a static threshold.

4.3 Mining Interesting Segment-Specific Associations

For demonstration purposes, we apply the next steps primarily to the segment-specific associations of the "wine" segment found in the dataset without the HFC (cf. the market basket prototype of segment $k = 1$ in Figure 4.8). About 6.13% (=184) of all the sample's households belong to the wine segment. After the 3456 transactions of the segment were pooled into $c_{k=1}$, the APRIORI algorithm mined 388 frequent itemsets with an heuristically predefined minimum support of 1%. Given that this number of frequent itemsets is too high and needs to be reduced to a more manageable figure, the 200 frequent itemsets with the highest all-confidence value and a minimum length of two categories are chosen for further examination. A separate cache stores all single frequent sub-itemsets since they are needed for the calculation of the optimization model in Section 4.4.

As predicted in Section 3.3.1, the distribution of category purchase frequencies within the generated transaction pools c_k of a segment is highly skewed, since the item combinations of only a few categories within the sparse transaction data characterize a segment. For example, different kinds of wines occur within the wine-segment disproportionately more often than other categories. Figure 4.16 illustrates this aspect in the "wine" and "baby" cluster. This justifies filtering the frequent itemsets with the all-confidence value because it reduces the risk of defining weakly-related cross-support patterns.

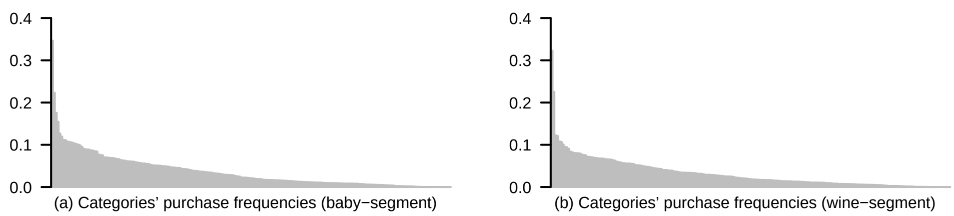

Figure 4.16: Skewed distribution of purchase frequencies within (a) the baby-segment and (b) the wine-segment

The analyst can examine the revealed frequent itemsets of the wine segment more easily if he applies the hierarchical cluster approach described in Section 3.3.2. It separates the frequent itemsets into smaller and more manageable association groups. Each branch of a dendrogram shows the decision maker the similar frequent itemsets according to the distance measure shown in Equation 3.7. Ward's (1963) algorithm compresses the distance matrix which consists of a value for each pair of the 200 frequent itemsets. Figure 4.18 plots the corresponding dendrogram.

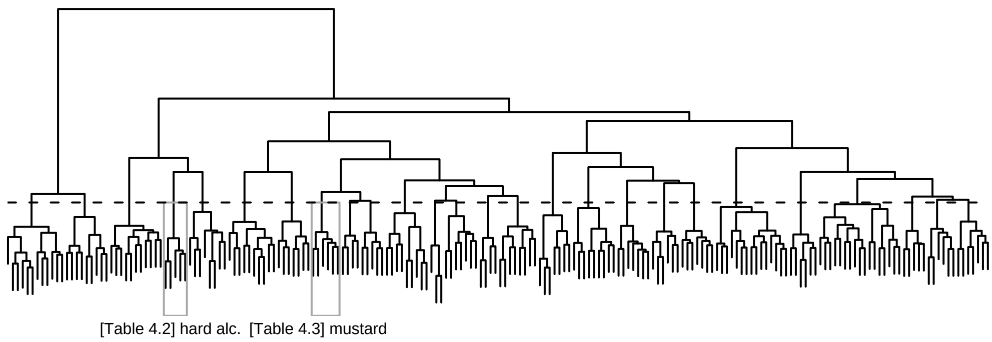

Figure 4.17: Dendrogram of frequent itemset grouping in the wine segment

Determining the "right" number of association groups l for this hierarchical cluster analysis is just as difficult as it is for the KCCA algorithms. To approximate an adequate l, the highest jumps within the fusion level are considered for the interval $l = [25, 35]$ (cf. Decker and Schimmelpfennig 2002). From a managerial point of view, a value between 25 and 35 seems to be acceptable since it defines a manageable number of association groups. Creating association groups splits the problem of exploring all the frequent itemsets of a segment into more easily solvable subproblems. This becomes even more useful if the itemsets of a group present a similar itemset composition according to which they can be sorted. Figure 4.17 depicts the dendrogram gained from the grouping of the 200 itemsets mined in the wine segment. The dashed horizontal gray line cuts off 26 association groups, with the gray rectangles marking two groups as examples. The one on the left-hand side includes five itemsets with relationships among categories of *hard alcoholic beverages* listed in Table 4.2. The association group on the right-hand side encloses itemsets containing *mustard* (see Table 4.3).

Usually, the groups do not only include frequent itemsets whose categories present a similarity reflecting, for example, a common characteristic (such as the hard alcoholic beverages). In most cases, they are simply heterogeneous and mix together itemsets containing different sorts of products, such as the second association group in Table 4.3. Nevertheless, arranging the found segment-specific associations into groups can help in determining special areas of similar itemsets, in sorting and separating the output lists into more manageable sublists, and in getting a better understanding of the segment's category purchase correlations (cf. Toivonen et al. 1995, Gupta et al. 1999).

The suggested KCCA cluster algorithm seems to build customer segments characterized by similar cat-

Itemset	Support	All-confidence
{brandy, whisky}	0.0113	0.2300
{brandy, fruit brandy}	0.0147	0.1839
{fruit brandy, appetizer}	0.0147	0.1839
{brandy, appetizer}	0.0158	0.1468
{whisky, fruit brandy}	0.0110	0.1379

Table 4.2: Group of itemsets in the wine segment that include hard alcoholic beverages

Itemset	Support	All-confidence
{mayonnaise, mustard}	0.0117	0.1734
{mustard, herbs}	0.0149	0.1480
{condensed milk, mustard}	0.0153	0.1276
{chicken, mustard}	0.0101	0.1205
{cooking oil, mustard}	0.0126	0.1185
{beef, mustard}	0.0101	0.1073

Table 4.3: Group of itemsets in the wine segment including mustard

egory correlations. To verify whether the generated associations specify the segments from which they have been derived, we use the itemset grouping. The idea is to partition the combined itemsets of two distinguished segments with the hierarchical cluster analysis previously introduced. The solution should reflect the existence of two association groups, with each group containing frequent itemsets that clearly arise from their corresponding segments. In addition, both segments are expected to have some common frequent itemsets.

Hence, 75 mined associations from two segments – here, the baby and the wine segment – are combined into one set of 150 associations. The corresponding transactions in c_8 and c_1 are pooled in a conjoint dataset. By means of Equation 3.7, the distance for each pair of itemsets is calculated and transferred to the distance matrix. After grouping the combined associations as proposed above, the branches of the resulting dendrogram in Figure 4.18 point out some smaller groups of associations on the left-hand side and two larger groups on the right-hand side of the plot. In fact, the small groups define itemsets which could be found in both segments (see the gray rectangles in Figure 4.18 and the corresponding itemsets listed in Table 4.4). However, the itemset partitioning rearranges the found segment-specific itemsets of the wine and baby clusters into two bigger association groups on the right-hand side. Consequently, the hierarchical clustering of the combined itemsets supports the cluster solution of the previous partition algorithm.

Source	Itemset	Support	All-confidence
Baby segment	{mayonnaise, mustard}	0.0143	0.1792
Wine segment	{mayonnaise, mustard}	0.1172	0.1465
Baby segment	{detergents, washing-up liquid}	0.0250	0.2618
Wine segment	{detergents, washing-up liquid}	0.0117	0.1226

Table 4.4: Groups of two identical itemsets found in both segments

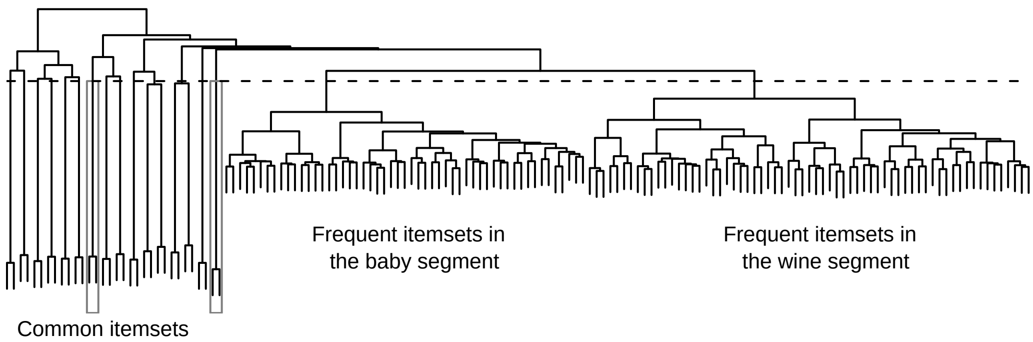

Figure 4.18: Dendrogram of frequent itemset grouping in the combined transactions of two segments

The grouping of the frequent itemsets identifies the different characteristics of the clusters with regard to the included category correlations. This verifies that our partitioning algorithm determines the purchasing behavior of the average households of a segment quite appropriately. Otherwise, the efficiency of the marketing efforts would be in danger since the targeted promotional campaigns depend on the segment-specific associations found in the data.

Generally, grouping the itemsets could provide the retailer with information about the existing category correlations of a specific segment. But since it is difficult to extract and to evaluate groups without visual inspection, an automated program can hardly be expected to implement this method. Although it is a useful extension for mining programs, we do not implement it for our fully-automated simulation in Section 4.5. The simulation comprises only the modules of the approach which can be computed without much user interaction, i.e. the KCCA, association rule mining, filtering of frequent itemsets and the optimization model.

4.4 Recommendation of Profitable Categories

As shown in Chapter 3, the framework ends in deriving the K limited lists of single categories from the segment-specific frequent itemsets identified previously. As discussed in Section 3.3.3, the optimization routine takes two important factors into account which effect the profitability of our target marketing framework: first, the profit generated with the recommended category and second, the cross-selling potential of the category. Considering only one factor could negatively affect the overall profitability of a framework which uses the items for promotional campaigns in the corresponding segment. For demonstration purposes, we generate the output list of the wine segment. Let us assume that a decision maker can only promote ten single categories from his long-tail (i.e. the LFC) in a leaflet for the wine households (i.e. $\Phi = 10$). To stress the characteristics of the category recommendations generated with the PROFSET model, we compare its output list with other alternatives available to the retailer to highlight specific categories.

For instance, if the retailer's objective is to generate as much cross-selling as possible with the promotion of Φ categories, the decision maker might use the Φ categories defined by the segment-specific, prototypical market basket of the chosen KCCA algorithm. The values of the centroids correspond to categories' purchase probabilities in the cluster and are determined according to their co-occurrence with other items. Hence, these items seem to be a good source for product recommendations in the segment. The left-hand side of Table 4.5 represents the prototypical market basket with the ten highest category purchase frequencies in the LFC of the wine cluster generated with the constrained KCCA algorithm. Although these categories have a high cross-selling potential, retailers would neglect their monetary values if they were chosen for promotional campaigns.

Another very simple method would be to select the Φ items generating the highest profit in the LFC of the segment (see the right side of Table 4.5). In contrast to the usage of the prototypical market basket of the KCCA algorithm, the retailer pays more attention to the monetary value of the segment-specific categories than to their purchase correlations with other categories. Neglecting these relationships in favor of a higher profit in single categories might increase the overall revenue.

The PROFSET model provides the middle course since it takes both factors into account. Consider the output list of the PROFSET model listed in Table 4.6. The "total sales profit" value reflects the total amount achieved with the corresponding category in the segment. Since the categories are embedded into frequent itemsets, the PROFSET model calculates the gross profit share of the segment-specific frequent itemsets and extracts the single categories by solving the decision problem, as explained in Section 3.3.3. The more often a category is included in a frequent itemset with a high support value and a high gross profit margin, the higher the "cross-selling profit" (CSP) value of the category grows. The residual of the "total sales profit" and the "cross-selling profit" is the "own sales profit". It reflects the rest of the profit earned with the single category, e.g. when it is sold separately. The list is sorted according to the CSP in descending order to show the Φ categories with the highest values. A higher CSP implies that the category contributes more to the overall profit with potential cross-selling and a higher monetary value.

Compare the recommended categories in Table 4.6 with the prototypical market basket (Table 4.5, left side) and the categories selected according to the highest profit (Table 4.5, right side). Obviously, the optimization model determines a list with a different category composition. For example, let us consider herbs. The centroid of the wine segment includes herbs due to their higher co-occurrence with other items in the market baskets of the wine segment. Nevertheless, herbs do not contribute to the overall profit very much. Hence, they do not appear on the right-hand side of Table 4.5. But as we can see, the PROFSET model lists them because of their higher CSP value. In contrast, condensed milk's correlation to other items does not compensate for its low monetary value. Despite its higher cross-selling potential, the lower profit achieved with condensed milk is the reason for its absence in Table 4.6. In a similar way, whiskey does not appear in the output of the PROFSET Table 4.6 due to its low CSP – despite the fact that whiskey generates an acceptable amount of profit (cf. right-hand side of Table 4.5). To summarize, Table 4.6 contains itmes of Tables 4.5 and 4.6. This indicates that the PROFSET model optimizes the selection of the Φ supported categories by including the category's profit *and* its cross-selling potential.

Item/category	$c_{j\|k=3}$
Red/rosé wines	0.32
White wines	0.23
Beef	0.12
Sparkling wine	0.12
Condensed milk	0.11
Cooking oil	0.11
Herbs	0.10
Appetizer	0.10
Chicken	0.10
House delicatessen products	0.09

Itemset	Total sales profit (€)
Red/rosé wines	1903.10
White wines	1109.37
Sparkling wine	662.60
Beef	547.07
Chicken	397.76
Brandy	368.81
Appetizer	359.17
Meat	311.58
Fruit brandy	299.44
Whisky	293.11

Table 4.5: Ten highest purchase probabilities extracted from the prototypical basket (left-hand side) and ten categories with the highest profit share in the LFC of the wine segment (right-hand side)

Item(set)	Own sales profit (€)	Total sales profit (€)	CSP (€)
{Red/rosé wines}	133.63	1903.10	1769.47
{White wines}	213.63	1109.37	895.75
{Sparkling wine}	436.75	662.60	225.85
{Beef}	358.08	547.07	188.99
{Chicken}	244.89	397.76	152.87
{Appetizer}	223.35	359.17	135.81
{Cooking oil}	118.44	239.85	121.41
{Herbs}	141.98	242.07	100.09
{House delicatessen products}	170.51	259.57	89.06
{Fruit brandy}	264.29	299.44	35.15

Table 4.6: Ten categories defined with the PROFSET model in the LFC of the wine cluster

It might be objected that implementing the PROFSET model is an inappropriate, overly-sophisticated effort since the categories of the output list often correspond to the most frequently purchased items in a segment. For example, the comparison of Table 4.5 and Table 4.6 shows that e.g. white wines, red/rosé and sparkling wine are on all three lists. The purchase frequencies of these products affect the earnings strongly and are mainly responsible for the highest share of the retailer's profit. However, this is not true in every case and depends on the number of defined categories, the distribution of purchase frequencies and the differences between the profit margins of the categories. Using the PROFSET model ensures that the retailer does not miss opportunities by neglecting cross-selling or by supporting valueless items.

4.5 Profitability Simulation

So far we have described the outcome of our combined approach in detail, but some of the most interesting questions still remain to be answered: How can the selected items from the framework be used for concrete marketing campaigns and what gain can be expected compared to a standardized, customer-unspecific price-promotional campaign? And returning to our initial question: Does the profit growth achieved by using a data-driven technique for targeted promotion exceed the effort spent to implement it? Although we did not conduct a field experiment under real-world circumstances, we can estimate the related profit of our proposed approach using the empirical transaction data.

4.5.1 General Assumptions

We can start to answer the questions posed above by considering the major objectives of promotional campaigns in general. In retailing, price is the most common variable used to distinguish a retailer from his competitors. Most marketing campaigns refer to the prices of featured categories or products (cf. Blattberg and Neslin 1990). Retailers usually have two main reasons for lowering the prices of specific goods:

- First, they often assume price-promotional elasticities of demand with a value lower than -1 concerning the featured products. This means that price reductions incite customers to buy more of these products, with earnings compensating or exceeding the losses (cf. Blattberg and Neslin 1990, van Heerde, Leeflang and Wittink 2004).
- Second, lowering the prices of certain categories should boost the visiting frequency of customers. If potential customers enter the store, they usually buy not only the promoted items but also additional products from the entire assortment. Managers expect from this strategy that the sales made by new or regular customers in the other categories will compensate for the loss involved in reducing the price. An example of this strategy is the selling of loss-leaders (cf. Walters and Rinne 1986, Blattberg and Neslin 1990, Fox and Hoch 2005).

Consider recent promotional campaigns conducted to achieve these goals by featuring specific products or categories. One of the most popular techniques is to highlight the “bestsellers” (top or hit products) of a company (cf. Brijs et al. 2000, Bodapati 2008, Elberse 2008). The question arises, how do retail managers select the items which are to be used for those campaigns? A wide variety of methods for defining the items exists, depending on e.g. the calculation of specific operating figures concerning the profitability of the categories. Nevertheless, we assume that most marketers choose the promoted categories simply according to personal experience. A common technique to mark the interesting categories is to use selection queries in the transaction or sales data. Here, we consider three simple queries determining the Φ items heuristically. The query can select the Φ items in the entire sales data (i.e. HFC + LFC) which generate the highest values in the following areas:

- profit margins (**option 1**)
- revenue (**option 2**)
- purchase frequencies/support values (**option 3**)

Featuring the most profitable categories seems intuitively to be the appropriate method (option 1): the retailer wants to boost sales in items which contribute the largest share to his profit. However, according to our stated objectives of promotional campaigns, the frequency of customer visits will probably increase if the retailer features the categories occurring in the highest share of all transactions (option 3). Promoting the items with a high support value might attract more customers and could initiate cross-selling most effectively. Concerning the first option, the retailer has to consider that the most profitable items do not necessarily also show a high support value. In practice, the decision maker could also feature the items generating the highest revenue since this is a common target figure in retailing (option 2). Table 4.7 lists for each of the three options the $\Phi = 4$ categories extracted from months one to ten of the second sample. The highest profit is achieved with sausages and the highest revenue with bottled beer. Various vegetables are found most often in single market baskets. All three options define categories which belong to the HFC.

	Option 1 (profit margins)	Option 2 (revenue)	Option 3 (support)
Cat. 1	Types of sausage	Bottled beer	Other vegetables
Cat. 2	Del. sausage/meat products	Del. sausage/meat products	Whole milk
Cat. 3	Other vegetables	Fizzy drinks	Delicacy bakery
Cat. 4	Delicacy bakery	Other vegetables	Del. sausage/meat products

Table 4.7: Result of the three options to select $\Phi = 4$ categories heuristically

Table 4.8 shows the number of transactions including at least one of the corresponding items of Table 4.7 as well as the profit which is achieved with these categories in months one to ten of the second sample. The first value might be a useful indicator for the reach of the marketing campaign. The more purchase occasions are affected by the corresponding promotion, the more customers will probably be attracted to the store in the future. Notice that the calculation of the profit values does not include any assumptions about cross-selling or correlations with other items. As expected, the categories of option 1 achieve the highest profit but show the lowest support values. The opposite is true for the categories of option 3. Promoting the four categories with the highest revenue leads to values which lie in between.

Each option takes the stated objectives of a promotional campaign into account with a different emphasis: promoting the Φ most profitable items from months one to ten will probably produce the highest increase in profit in the corresponding items. On the other hand, these items will not occur in the highest share of the transactions, as shown by Table 4.8. Concerning the stated objectives of typical price promotion, even retailers who implement customer-unspecific heuristics want to increase cross-selling and the frequency of visits. Featuring the items with the highest support values means that these items appear in more transactions than do other items. Hence, more households would be reached by the corresponding marketing campaign and the chance of cross-selling or unplanned purchases would rise. Nevertheless, our calculated profit achieved with these four items in the first ten months is very low (€ 41,005.35). If the company uses the Φ items with the highest revenue, the categories generate a slightly lower profit compared to those in option 1. However, the number of transactions including one of the four items is higher.

Heuristic query	Option 1 (profit margins)	Option 2 (revenue)	Option 3 (support values)
Number of transactions	47,356	48,511	51,028
Profit over all K in €	75,904.48	63,858.69	41,005.35

Table 4.8: Number of transactions containing at least one of the Φ categories, and profit achieved with the different categories of the heuristic promotion method

To show the differences between standard promotion heuristics and personalized target marketing, we compare the simulated profit gain produced by the customer-unspecific promotion method when the Φ categories of the HFC are determined using the three query options, and a segment-specific promotion campaign implementing long-tail categories (i.e. LFC) from our data-driven framework. If the Φ items are determined according to the three simple queries (options 1-3), all customers are presented with a price reduction in the same promoted categories. In other words, this method does not single out customers according to their individual characteristics. Here, we call such aggregated advertising campaigns “customer-unspecific promotion heuristics”, e.g. sending leaflets informing all households about special prices in the Φ major categories.

The customer-unspecific promotion heuristic will be compared to a campaign which uses the Φ categories derived from our data mining approach for two successfully implemented advertising techniques designed by Drèze and Hoch (1998): category destination programs (CDP) and cross-merchandising (CM). Category destination programs award the participants a price reduction for all purchases made within their Φ determined categories. For example, Drèze and Hoch (1998) implemented a “baby club” whose members get a coupon for a price reduction of 10 percent if they spend at least $100 in the related categories. These programs correspond to the results of our approach perfectly since we show in detail how to identify the members of the program and which categories have to be promoted. Category destination programs can be combined with cross-merchandising techniques easily. The cross-merchandising technique uses the customers’ affinity for specific goods to direct them through the store to other items. For example, Drèze and Hoch (1998) suggest combining the best-selling categories with less frequently purchased items, e.g. by placing a sign near the bottled beer recommending certain snacks which are arranged on another shelf at the store. It can be shown that sales within the targeted categories and the rest of the assortment grew in the observed period. The authors of the survey attribute this to the increase in the distance covered by customers walking through the store to reach the promoted items, and the corresponding increase in their exposure to the assortment.

4.5.2 Setup of the Simulation

Now, let us assume the owner of our empirical dataset wants to conduct a price-promotional campaign within $\Phi = 4$ of his 268 categories after the tenth month in order to increase sales in months eleven and twelve. To promote the four categories, the decision maker is allowed to reduce the average price of the categories by 10%. No further promotional campaigns are conducted during the first ten months of the year and seasonal and stock-buying effects are ignored.

The data from months one to ten is used to predict the customers’ cluster membership within the following two months. Figure 4.19 allows us to confirm whether the cluster size (left, dark-gray bars) corresponds to the number of customers (middle, light-gray bar) and customers’ generated percental profit (right, gray bar) in months eleven and twelve. Although some small deviations are present, the three values generally correspond to each other. Regarding the company’s dataset, we can reason that the profit generated by the customers of the corresponding group correlates approximately to the size of the cluster.

Of the 3,000 customers, 348 did not visit the company’s stores in months eleven and twelve. Nevertheless, our estimation of the overall profitability per person includes these non-buying customers to provide a more realistic picture. Generally, the retailer has to expect that his marketing campaigns will not touch every single customer.

For each segment, the decision maker can choose between implementing the standard promotional campaign, or carrying out segment-specific category destination programs combined with cross-merchandising techniques. To compare the expected profit gain of both alternatives, we first have to calculate for each

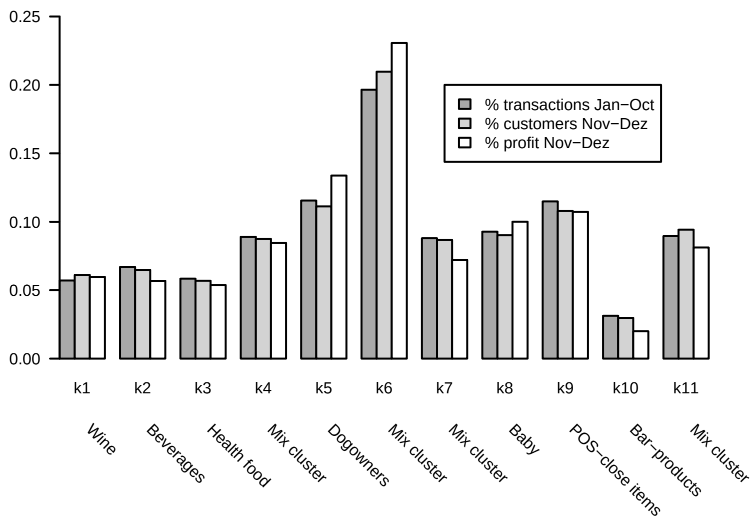

Figure 4.19: Transactions in each cluster (months 1-10); predicted number of customers and the actual profit generated by them in months 11/12

segment the output list with $\Phi = 4$ categories. Table 4.9 summarizes the Φ categories which should be featured according to the proposed segment-specific approach for four selected segments of the empirical dataset. The Φ categories of the other segments can be found in Tables A.9 and A.10 of the Appendix. Refering back to Chapter 1, the $\Phi = 4$ categories are derived from the itemsets found in the 216 categories of the long-tail with a minimum support of 2% for all segments. The 70 itemsets with the highest all-confidence value in each segment are passed on to the optimization model.

Segment	k=1	k=3	k=5	k=8
Titling	Wine	Health food	Dog owner	Baby
Cat. 1	Red/rosé wines	Organic products	Dog food	Baby hygiene prod.
Cat. 2	White wines	Wholemeal products	Beef	Baby food jars
Cat. 3	Sparkling wine	Organic beef	Chicken	Baby food powder
Cat. 4	Beef	Frozen ice cream	Meat	Frozen ice cream

Table 4.9: Four categories determined with the segment-specific approach in $k = 1$, $k = 3$, $k = 5$ and $k = 8$

Table 4.9 shows that the output lists vary in their composition for each segment. Notice that the three options in Table 4.7 determine completely different items compared to the ones in the customer groups. Since the heuristic queries tend to define the bestsellers, all three lists contain categories of the 52 HFC. Now, the goal is to demonstrate that managers will not run the risk of losing profits if they promote the "long-tail" categories (i.e. the LFC) identified with the segment-specific approach. Moreover, we want to show that the chances of increasing retailers' profits are higher if the advertising campaigns consider the characteristic purchase patterns of the different customer groups in particular.

To estimate which of the two campaigns – the customer-unspecific or the segment-specific one – results in a higher growth of profit in the segment, we have to determine two components. First, we calculate the profit achieved by selling the predicted Φ categories of a campaign in months eleven and twelve. We call this value Γ, reflecting how much profit is affected by the corresponding technique when it features the Φ items. Second, we estimate the percentage growth that will be probably be achieved with the promotion method. The multiplication of Γ with the estimated percentage return reflects the estimated profit of the corresponding campaign.

Let us take the wine segment as an example and assume that both methods would increase the retailer's profit by 10%. If the retailer achieves a higher sum of cumulated profit margins by selling the Φ categories derived from our segment-specific approach (left-hand side of Table 4.9) compared to the Φ categories of the promotion heuristic (one option of Table 4.7) in the eleventh and twelfth months, the segment-specific approach implies a higher profit increase when it is implemented after the tenth month. Before calculating the profit values, we re-merge the previously eliminated part of the transactions containing the 52 filtered HFC with the rest of the market basket records (i.e. the parts of the transactions containing the 216 long-tail categories). This makes the two alternatives comparable since they both refer to the same set of complete transactions.

Defining the affected profit (i.e. Γ) is not straightforward. Due to category correlations, we have to consider the purchase relationships between the featured categories and the rest of the assortment. Therefore, we must calculate not only the cumulated profit margins achieved with the categories directly, but also the profit yielded potentially by cross-selling with these items. Looking at the empirical data, we calculate the values as follows:

1. We add up all of the profit margins which are generated in the eleventh and twelfth months in the featured Φ categories of the corresponding segment.

2. To take into account the cross-category correlations potentially initiated by a method, we calculate all association rules with a minimum support of 2% for whichever one of the Φ categories builds the antecedent on the left-hand side of the rule. Since only the major correlations are of interest, we consider rules with a length of two items. Circular correlations are also ignored (e.g. $\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}, \{C\} \rightarrow \{A\}$). We identify all transactions in months eleven and twelve that contain the mined rules and add up the profit margins which are achieved with the categories on the right-hand side of the rule. By multiplying these values with the confidence value of the corresponding rule, we get the expected, indirectly affected cumulated profit margins generated by selling the Φ categories. Of course, two or more of the Φ categories could build a rule with

the same right-hand side category, e.g. $\{vegetables\} \rightarrow \{water\}$, $supp(0.02)$, $conf(0.5)$ and $\{bottled\ beer\} \rightarrow \{water\}$, $supp(0.07)$, $conf(0.8)$. For every transaction containing the two rules, a random selection, similar to the one used by the PROFSET model (cf. Section 3.3.3), is weighted with the support values and determines which the rule to choose. Here, the second rule $\{bottled\ beer\} \rightarrow \{water\}$ would be picked with a higher probability compared to the first one $(0.07 > 0.02)$.

The sum of both calculations is Γ. This profit value shows how much profit would be achieved in months eleven and twelve if the items were sold with no promotion. When estimating the percentage gains made by each promotion method added to Γ, we consider two scenarios determining a lower- and an upper boundary of the profit growth. The first scenario determines optimistic values for the segment-specific campaign and pessimistic values for the customer-unspecific heuristic, while in the second scenario the opposite is true. In reality, we think that the predicted values oscillate between these boundaries.

If category destination programs are implemented, Drèze and Hoch (1998) calculate a 25% increase in the featured categories after the program has been running for six months and taking costs into account. Our simulation comprises only two months. The profitability of the program depends on the length of the time period, since sales in the promoted categories usually increase over time. We expect a value of 10% with scenario no. 1. In contrast to our approach, Drèze and Hoch (1998) select the items of the original CDP without including any purchase relationships. The promoted categories of our approach are matched to the purchasing behavior of the targeted customers. Hence, we estimate a profit increase of as much as 15% in the Φ categories for the second scenario.

Turning to cross-merchandising, we assume that the retailer is able to link the Φ categories to the items on the right-hand side of the determined association rules with appropriate techniques (e.g. signs at the wine shelves recommending that participating club members buy a companion appetizer). A second study by Drèze and Hoch (1998) determines an increase in the correlated right-hand categories of 6% to 10%. We estimate a 6% increase for cross-merchandising methods as the lower boundary, and assume a 10% growth if the second scenario comes to pass.

For the customer-unspecific promotion heuristic, we refer to the meta-analysis by Tellis (1988) and Bijmolt, van Heerde and Pieters (2005), who summarized empirical research related to price elasticities. According to these studies, an appropriate value for the price elasticity of an average category is about $E_{x,p} = -1.5$. The price elasticity in the cited research refers to the percentage change in the sales x subject to the corresponding price reduction. If we want to look at the change in the profit G, we have to adjust this equation. Assuming that a given price elasticity $E_{p,x}$ exists, we denote $G = x * (p - c)$ with $x = sales$, $c = costs = 0$, $p = selling\ price$ and $x = x(p)$. According to the following equations, we can approximate the price elasticity subject to the revenue, according to $E_{G,p} = E_{x,p} + 1$.

$$E_{G,p} = \frac{\frac{dG}{G}}{\frac{dp}{p}} = \frac{dG}{dp} * \frac{p}{G} \tag{4.1}$$

$$\frac{\partial G}{\partial p} = \frac{dx}{dp} p + x \tag{4.2}$$

$$E_{G,p} = (\frac{dx}{dp} p + x)\frac{p}{G} = (\frac{dx}{dp} p + x)\frac{p}{p * x} \tag{4.3}$$

$$E_{G,p} = \frac{dx}{dp}\frac{p}{x} + 1 = E_{x,p} + 1 \tag{4.4}$$

With regard to the promotion heuristic, $E_{G,p} = -0.5$ means that the price reduction of 10% will cause a 5% growth of profit in the corresponding categories. For the sake of simplicity, we ignore costs whose consideration would decrease $E_{G,p}$. If costs are present (i.e. $c > 0$), the resulting profit increase of the customer-unspecific promotion heuristic is not as high as 5% in the case of a price reduction of 10% and $E_{p,x} = -1,5$. The profit estimation of the promotion heuristic benefits from $c = 0$ since it ensures a more conservative calculation of the output of our segment-specific method.
If retailers want to boost their sales with reductions on purchase prices, they should know the expected price elasticity of the corresponding products. Empirical research shows that a growth in sales does not always compensate for the losses caused by lower prices in many categories (cf. Günter and Klapper 2007). For example, analysts expect price elasticities below a value of 1 for many food categories (cf. Tellis 1988). Since the categories determined by the promotion heuristic all stem from the food domain (cf. right-hand side column of Table 4.7), the projected profit increase of 5% is very optimistic and the actual increase will probably be much lower or even negative. Nevertheless, to prevent ourselves from overestimating the results from the segment-specific framework, we still estimate a growth of 3% for the cumulated profit margins of the Φ categories when the standard promotion heuristic is applied.

The academic literature does not appear to include any further research which examines the generated profit growth of a standard promotion heurisitic in the correlated categories of the featured items. It is possible that the sales of the correlated categories could rise as much as the sales of the promoted categories (5% for the first scenario), but in fact, the gain will likely be much smaller. We assume that the profit in the correlated categories will increase by 1%. Putting all of the values together, Table 4.10 summarizes the expected percentage growth for each method and scenario.

4.5.3 Result of the Simulation

Comparing the three options of the promotion heuristic (options 1-3) to the segment-specific promotional campaign, and considering the two scenarios nos. 1 and 2, six cases of different profit calculations are

Method	Scenario no. 1 Estimated growth	Scenario no. 2 Estimated growth	Categories to which expected profit is added
Category destination program	10.00%	15.00%	Φ categories
Cross-merchandising	6.00%	10.00%	correlated categories
Standard promotion heuristic	5.00%	3.00%	Φ categories
Standard promotion heuristic	5.00%	1.00%	correlated categories

Table 4.10: Estimated percentage profit growth for each method and the two scenarios

possible. For demonstration purposes, we calculate the values of the customer-unspecific heuristic, which uses the four categories showing the highest profit in months one to ten (option 1). Suggesting the most profitable items for the standard promotion heuristic seems to be the obvious preference from a retailer's point of view. The results for the other two options are likewise determined.

In Figure 4.20, the bars represent the expected profit growth within each customer group derived from the values of Table 4.10. The gray bars illustrate the additional earnings if the customer-unspecific promotion heuristic is applied. The white bars symbolize the corresponding values of the segment-specific campaign in the same way.

With regard to the first scenario, Figure 4.20(a) shows that the white bars exceed the gray ones in four of the eleven segments – specifically in the wine ($k = 1$), the health food ($k = 3$), the dog owner ($k = 5$) and the baby segments ($k = 8$). In other words, if the retailer uses the four categories of the LFC determined using our approach for the customized promotional campaign in these segments, he can expect a higher gain compared to his standardized method which implements the four categories that have generated the highest profit in the past (option 1). Where the gray bars of a segment exceed the white bars, the retailer should go on applying the usual promotion heuristic since the use of the segment-specific approach would not be profitable. The situation changes if we expect the more optimistic scenario no. 2. Figure 4.20(b) depicts larger white bars for all segments. Hence, the segment-specific target marketing campaign would be much more profitable in every customer group. Figures A.15 and A.16 of the Appendix show a similar result if options two and three are used to define the categories for the promotion heuristic.

To highlight the differences between the two methods in terms of numbers, we summarize the results of all six cases in Table 4.11 and Table 4.12. For instance, by presenting the four customer segments $k = 1$, $k = 3$, $k = 5$ and $k = 8$ of the sample with the customized target marketing campaigns, the retailer would earn €2,176 more compared to his standard method, which promotes just the four items generating the highest profit. This is a gain of 15% in scenario no. 1. The customized marketing campaign would be profitable for 847 households (=32.2% of the 2652 targeted households – see Table 4.13) in months eleven and twelve.

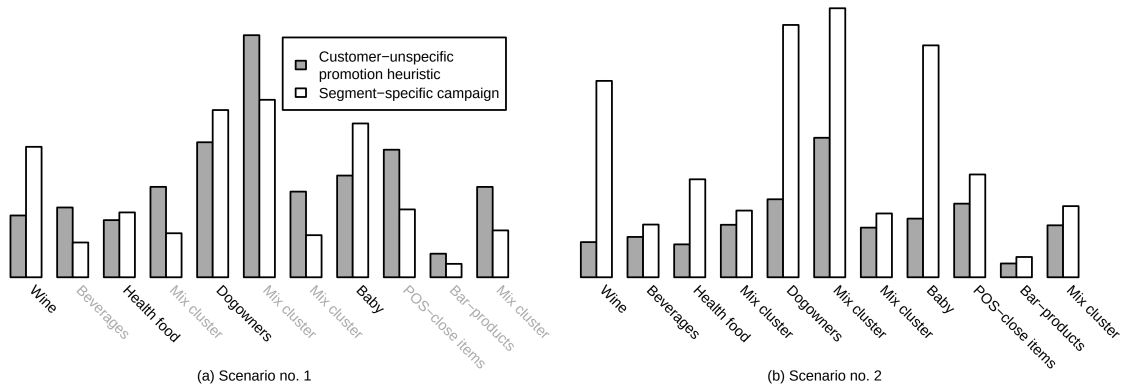

Figure 4.20: Comparing the expected profit growth in the Φ categories for (a) scenario no. 1 and (b) scenario no. 2, if the heuristic implements the categories generating the highest profit (option 1)

	Option 1 (profit margins)	Option 2 (revenue)	Option 3 (support)
Targeted segments k	1, 3, 5, 8	1, 3, 5, 8	1, 3, 5, 8
Relative surplus vs. standard promotion	15%	27%	22%
Absolute surplus	€ 2,176	€ 3,185	€ 2,802

Table 4.11: Surplus generated by customized campaigns compared to standard promotion heuristics for scenario no. 1

If the standard promotion heuristic defines the categories according to options 2 and 3 (i.e. featuring the Φ categories with highest revenue or purchase frequencies), the suitable segment-specific promotion programs with LFC seem to generate a profit gain of up to 27%, or 22% when scenario no. 1 is assumed. Nevertheless, about two thirds of all customers still need to be addressed with the standard promotion heuristic. The segment-specific promotion seems to be successful for only up to one third of the remaining customers (cf. Table 4.13). For the segments $k = \{2,4,6,7,9,10,11\}$ in Figure 4.20(a), almost all the white bars are only half as high as the gray bars. This means that the profit increase achieved with the standard promotion heuristic will be twice as high as that achieved in the segment-specific

	Option 1 (profit margins)	Option 2 (revenue)	Option 3 (support)
Targeted segments k	1-11	1-11	1-11
Relative surplus vs. standard promotion	128%	191%	167%
Absolute surplus	€ 10,825	€ 12,658	€ 12,078

Table 4.12: Surplus generated by customized campaigns compared to standard promotion heuristics for scenario no. 2

Segment(s) k	Number of households (abs./rel.)	Number of transactions (abs./rel.)
1, 3, 5, 8	847 (32.2%)	4263 (32.19%)

Table 4.13: Number of transactions and households targeted with the segment-specific promotional campaign (scenario no. 1)

case. For these segments, the expected gain in profit generated with category destination programs and cross-merchandising methods will probably not compensate for the losses incurred by not supporting the bestselling products with standard promotion techniques.

Applying category destination programs for target marketing is not useful if the standard customer-unspecific promotion heuristic seems to initiate a higher growth, or if the calculated Φ categories are not applicable for this technique. Nevertheless, interesting and useful category correlations can still be used for cross-merchandising, even if the standard promotion heuristic is carried out. For example, the itemset mining done previously within the transactions of the large mix-cluster $k = 6$ shows that it contains the frequent itemset $\{skin\ care,\ women's\ wear\}$, which does not occur as often in other segments. Cross-merchandising which includes this correlation could raise the number of purchases, even though a category destination program does not seem to be a suitable choice for the cluster.

If we consider the more optimistic scenario no. 2, we can see that supporting the segment-specific LFC categories of every customer group generates higher profit growth in every segment compared to the standardized promotion of the bestsellers. Depending on the method used to select the Φ categories for the standard promotion heuristic (options 1-3), the additional profit generated with segment-specific target marketing exceeds that gained using the standard method by up to 191%.

A consideration of the absolute difference of €2,176 for scenario no. 1 might make the method appear not very profitable in real-world applications. Since the sample comprises only a fraction of the retailer's overall 56,000 customers, however, this amount can be multiplied by a factor of $56000/3000 \approx 18.67$ to get the expected overall gain in profit for this scenario. In accordance with Drèze and Hoch (1998), the earnings are still likely to be higher than the additional costs of providing the identified segments with specific promotional activities (e.g. a wine-club program). Certainly, the height of these costs depends on the design of the marketing campaign in the end. A club program run by an online retailer who is able to communicate with his customers via cheap e-mails might be more cost-efficient than a program offered by a stationary retailer who depends on letter post. Concerning the relation between expenditures and earnings, it must be noted that our virtual simulation runs for only two months. In practice the category destination programs will last for much longer periods (e.g. half a year) and retailers' revenues will likely increase over time.

Although the calculated profitability of the target marketing campaign is much higher if the values of scenario no. 2 become reality, not every segment-specific list of Φ categories can be used to implement a CDP. Tables A.9 and A.10 show quite similar categories especially for the mix clusters, such as chicken and beef products. These categories do not characterize the purchasing behavior defined by the clustering. For example, the households of cluster $k = 7$ are characterized by their interest in biscuits. The PROFSET model does not include biscuits in the recommendation list (see Table A.9) because of their low gross profit margins. In these instances, it might be appropriate to summarize the customers of clusters with similar PROFSET items into one CDP, e.g. the "barbecue club" supporting sales of different kinds of meat.

Summarizing the results of the simulation, the transaction data of the Austrian supermarket chain includes several interesting customer groups which seem to be appropriate for targeting with specific promotional campaigns. In contrast to a uniform one-for-all special price campaign, the suggested promotional techniques combined with the outcome of our data-driven approach have a greater chance of initiating an increase in profit and should therefore be considered by the management.

5 Summary

5.1 Contribution and Results

Returning to the research objectives stated in the introduction (cf. Section 1.2), the first challenge was to show how customers for target marketing can be selected using loyalty program data. To do this, we created a data mining approach which identifies groups of customers with a similar interest in certain category combinations. The approach then extracts the most valuable items from the data to promote them in the revealed segments using suitable target marketing campaigns. The complete framework solves the retailer's item selection problem when planning promotional campaigns (cf. Blattberg and Neslin 1990).

The approach groups customers according to the category combinations found in the so-called "long tail". The long-tail range of an assortment comprises the items which are bought infrequently. In contrast to category correlations found between products with disproportionately high purchase frequencies, we argue that these item combinations distinguish the customers in a more expedient manner, and therefore enable retailers to adapt appropriate marketing strategies to the characteristic purchasing behavior of the group members more effectively.

In terms of method, customer segmentation is one of the most challenging tasks of the stepwise procedure. Owing to the infrequency of purchases in the long-tail categories, a partition algorithm has to deal with very sparse binary data. We have shown that common KCCA algorithms such as K-means are not reliable in finding predefined grouping information in our artificial binary data. Hence, our approach implements a constrained KCCA algorithm that shows good results compared to other algorithms (cf. Section 3.1.4). The chosen algorithm considers not only the single transactions but the complete buying histories of the customers during its iterative procedure. This additional information helps to reveal the predefined groups even in very sparse datasets.

When the cluster algorithm is applied to the real-world dataset of a chain of supermarkets, it identifies a number of customer segments which seem to be interesting from the marketer's point of view. For example, some members of the loyalty program combine child-related products or different kinds of wines in their market baskets. The characteristic purchasing behavior of young families and wine connoisseurs in particular can have a high value for retailers who wish to plan appropriate target marketing activities.

The items extracted at the end of the mining process seem to be a useful source of recommendations that can be used for promotional campaigns in each of the identified customer segments. Since category correlations are considered in all steps of the approach, the determined categories show a high cross-selling potential with other parts of the assortment. Moreover, the approach takes not only purchase frequencies into account, but also the monetary value of the recommended items. This affects the profit of the retailer positively if the items are used for customized promotional campaigns.

The next objective of the thesis was to demonstrate the usage of the extracted items for customized promotion and to estimate the profitability of their implementation. To achieve this, we simulated the implementation of the determined items from the medium-sized supermarket in two promotional models developed by Drèze and Hoch (1998): segment-specific category destination programs, and cross-merchandising activities. For a pessimistic and an optimistic scenario as well as three options of the standard promotion heuristic for selecting the featured items, we compared the profit gain generated by the customer-unspecific promotion heuristic and the segment-specific target marketing campaign within each single segment. The segment-specific advertising approach led to an overall profit growth of 15% to 191% when the standard heuristic was replaced with target marketing campaigns in the suitable customer groups.

If the parameters of the more pessimistic scenario turn out to be closer to reality, only about one third of all customers in months eleven and twelve will belong to groups which could be targeted with the customized campaigns. The rest of the customers should be addressed with the more profitable mass-marketing techniques featuring the bestsellers. If the expected values of the second, more optimistic scenario become reality, the segment-specific target marketing should be applied to all customer groups. This contradicts the findings of Elberse (2008), who maintains that taking the bestselling products into consideration leads to higher profits in most instances. From a calculatory point of view, Figure 4.20 as well as Figures A.15 and A.16 show higher gains for every segment compared to the standard promotional heuristic.

Concerning the methodology of the simulation, we have to stress some issues affecting the calculatory results. For instance, the expected profit growth is derived from the real-world experiments of Drèze and Hoch (1998). Since we do not have the ability to validate these figures, we have to rely on the survey. This has to be borne in mind when looking at our estimated values: although we were quite cautious in choosing the expected growth for each scenario, further empirical applications might refute our predictions. Moreover, the results depend on the preparation of the data which is used for the simulation. For example, the length of the purchase histories depends on the length of the observation period. However, longer or shorter histories affect the results – especially the stability of the cluster solution, as shown in Section 3.1.4. The results of the simulation cannot be generalized for every business environment. Nevertheless, the simulation demonstrates the application of the approach to a real-world dataset and shows the opportunities for retailers if they consider using more customized target marketing compared to heuristic promotional campaigns built on mass marketing. To guide managers through the decision process of choosing between segment-specific target marketing and the standard methods, the next section presents a short decision scheme.

5.2 Managerial Implications

The managerial implications of the general use of our data-driven approach in a segment-specific promotional campaign can be illustrated with a compact decision scheme (cf. Figure 5.1). The marketer has to decide whether to stick to his standard method or to switch to a segment-specific target marketing approach according to the decision paths for each segment individually. Similarly to the procedure described in the empirical application, we separate the decision task into two major steps: data exploration and estimation of profitability. Each decision path is numbered according to the following listing (cf. numbering scheme in Figure 5.1):

1. The data exploration step should clarify whether the approach identifies valuable customer segments in the data. The usability of the extracted items of the PROFSET list for customized promotional campaigns (e.g category destination programs) in the segment needs to be determined. If they are not usable, the retailer should stick to his standard mass marketing approach.

2. If suitable customer segments are present, the decision maker has to estimate for each segment individually whether and how much more profit can be achieved with segment-specific advertising compared to his standard promotion. The profit gain depends on the expected sales of the recommended items in each segment, as well as the estimated profitability of the planned segment-specific promotional campaigns.

3. If it appears that the segment-specific campaigns might not lead to a higher gain compared to the standard promotional heuristic, the corresponding customer groups should be addressed using the standard method. However, identified category correlations can be used to support traditional mass advertising (e.g. cross merchandising with segment-specific category correlations).

4. If it appears that the segment-specific campaigns could initiate a higher growth in a corresponding segment, the retailer should implement a target marketing campaign (e.g. a category destination programs).

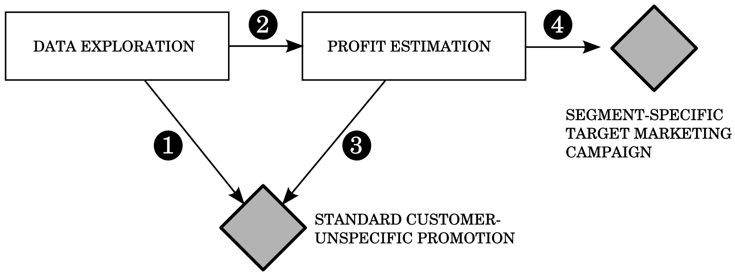

Figure 5.1: Decision scheme for choosing the promising promotional strategy

Using this decision scheme, the manager can evaluate whether the adoption of the approach is suitable for his own business environment. The profitability of adopting it also depends on several other factors – in particular, the costs of the customized promotional campaigns or the general costs involved in data collection (costs of the loyalty programs, costs of data processing etc.).

5.3 Discussion

Despite its promise of increased profitability in most customer segments, the concept of the exploratory approach is not unassailable. For instance, not all the customers of a retailer can be provided with more individual offers: it should be recalled that a quarter of the sampled buyers were dropped during the data preparation process. In addition, some customers might not come into contact with the campaigns at all (e.g. the 11% who did not visit the shop during the virtual application of the approach in Section 4.5). In some segments, moreover, the approach defines categories which might not be suited to target marketing campaigns (e.g. category destination programs).
The usability and profitability of target marketing campaigns based on our approach also depends on further methodological and conceptual issues. Some of these issues are worth discussing with regard to the applicability of the approach.

Although our approach relies on data-driven techniques, this does not mean that its results can be achieved without user interaction. For instance, the cluster algorithm usually does not reveal a grouping structure after its first application in an unknown dataset. Very often, the analyst has to conduct preliminary data analysis to form an idea about a possible arrangement of the entities. This is also the case for the association mining parameters such as the minimum support. The analyst has to find a good balance between revealing interesting item correlations and overburdening his computational capacities. Since the calculation time of the overall procedure increases with a low minimum support, the analyst is often forced to adjust the parameters following the "trial-and-error" principle. Hence, the approach does not provide a fully automated program but a framework which needs a well-informed and experienced user to create valuable target marketing campaigns.

Another issue of method which arises with our approach is the dissregard for monetary values during the clustering step. Although binary data speeds up the calculation of the outcome, the amount of money spent within each cluster is an important piece of information for target marketers. For instance, building clusters which all show an equal share of revenue would allow marketers to give the same attention or budget to each customer group (cf. Strehl and Ghosh 2003). It might be useful to consider prices already during the clustering stage instead of re-importing them for the subsequent optimization model.

From the point of view of method, the user always has to keep in mind that the intermediate and final result of each step of the procedure depends on the outcome of the foregoing step. If the analyst makes any conceptual mistakes in the first steps of the chain, these errors affect the entire outcome of the approach. To avoid false conclusions, the analyst has to check and verify the results at each step in the approach. For example, repeated clustering as described above helps to confirm the characteristics of the found groups.

The problems and deficiencies of our approach give justification to the widely-used application of simple mass marketing. Selecting items for customer-unspecific price promotions according to less-sophisticated heuristics (e.g. taking the Φ categories which have generated the highest profit in the past) has some simple and compelling advantages: the selection can be applied effortlessly and the chosen items often touch the majority of the clientele. Additionally, the heuristic can easily include further business-related circumstances such as inventory levels or the personal experience of the retailer. In contrast, the segment-specific approach depends on a cost-intensive loyalty program and the user needs to have much more experience in data processing techniques. The simulation shows that it is quite difficult to estimate the return of investment on the customized promotion using our framework without testing it under real conditions.

Although our simulation calculated a quantitiative enhancement when segment-specific target marketing was applied, some conceptual and qualitative issues might impact the approach's profitability indirectly or over the course of time. For example, researchers advise that common price promotion does not affect the purchasing behavior of customers significantly in the long run. Shortly after the price has been normalized, additional sales decrease (cf. van Heerde et al. 2004). In contrast, clustering buyers and selecting the featured items according to purchases made in the "long-tail" clearly separates specific groups and supports them with more customized offers. Hence, promotional campaigns which consider the special expectations of the clientele will likely hold customer loyalty longer (cf. van Heerde et al. 2004). This should enhance overall profitability since the majority of marketers agree that loyal customers spend more money and are less price-sensitive (cf. Kumara and Shahb 2004).

Some retailers might wish to implement club programs using a less-sophisticated customer selection method, e.g. offering customers membership in a baby-club program if they have made purchases in baby-related categories. This strategy depends on the a priori assumptions of retailers about the grouping structure of their customers. For growing assortments and changing consumption trends, defining customer segments ad hoc becomes difficult since decision makers are not able to identify every possible group of customers showing similar purchase patterns. Our exploratory segmentation approach is able to reveal these groups – even the ones that have not been anticipated by marketers.

Retailers are aware that there are some customers who will visit different stores to buy only the products whose prices have been lowered by a promotional campaign (cf. Lal and Bell 2003). Although these "cherry-picking" customers threaten the profitability of every promotion, we think that our approach is able to exclude them more effectively than normal mass advertising. Buying sequences reflect the consumption behavior of customers quite precisely and are a good source from which to evaluate customers' loyalty. The grouping of customers with a sufficient long purchase history enables the retailer to determine the exclusively rewarded customers on his own. For example, instead of providing a special price on diapers to all visitors, the retailer can send coupons to the targeted households. This lowers the risk of incurring losses by attracting cherry-pickers.

A field experiment should verify under real conditions whether the conceptual and qualitative advantages of the approach might lead to higher profits compared to standardized promotion heuristics in the long run.

Since our approach implements several data mining techniques, some general issues should be considered from a methodological point of view. First of all, the outcome of data mining techniques depends on the data, which has to reflect the business correctly. In other words: if garbage goes in, garbage comes out (cf. Kuonen 2005). The approach does not enable an enhancement of the business if the recording of the transactions is inaccurrate or incomplete. With regard to the results produced, users have to be aware of interpreting found data patterns incorrectly. There is always a risk of assuming a causality of certain category correlations extracted from transaction data. For instance, the rumor of a linkage between beer and diapers is quoted in many works about ARM or Data Mining (cf. Kelley 1996, Kleinberg, Papadimitriou and Raghavan 1998). To explain the appearance of this correlation, some suggest that young fathers reward themselves with beer when buying baby-related products. Management decisions based on such hypotheses about causality between items are quite risky without empirical validation and verification. Nevertheless, data mining is able to highlight such correlations, which can then be proved with surveys or other methods. It has to be stressed that data mining is not a fully-automated black box that extracts business information from data with the press of a button. Since it is a process, the user has to control the retrieval of information with his intuition, cognition and feedback (cf. Schulz and Nocke 2007). This starts with the data-preprocessing and ends with the implementation of the extracted knowledge in business decisions.

5.4 Outlook and Future Work

Although our data-driven approach provides a complete course of action, some modifications might enhance the results gained in further fields of application. For instance, it would be interesting to see how the results differ for varying datasets. A do-it-yourself store might show more equally distributed purchase frequencies compared to a grocery or supermarket. As a consequence, weakly-related cross-support patterns would not be a major problem and it might be helpful to replace the all-confidence measure with another filtering measure.

Concerning the online cluster algorithms dealing with anonymous transaction data, a very important advantage is the dynamic partitioning of the entities, as described in Section 2.2.3 of Chapter 2. Online algorithms are able to partition the constantly-accumulating data from market baskets progressively. This decreases the computational effort needed compared to batch algorithms. Partitioning the transactions as soon as they appear makes possible a real-time exploration of the clusters. The analyst can see in what ways the clusters change over time, and a retailer will be able to adjust his marketing decisions as early as possible. The (batch) cluster algorithm of our approach requires the complete dataset from the beginning. Hence, the partitioning reflects a static view of the customer groups which might have changed in the meantime. Since customers usually wander between different segments over time (e.g., due to the birth of a child or the acquisition of a pet), an extension of the approach should combine the must-link condition with the dynamic building of the transaction sequences. Concerning the dynamic growth of the buying histories, the approach has also to consider that recent transactions have more weight than older ones.

In addition, the computational advantages of online algorithms might make it possible to assign a customer ad hoc to a certain customer segment. If our approach could calculate the output list within a reasonable response time (e.g. a few milliseconds), the system might support the purchase decisions of online-visitors by recommending the Φ most suitable products. Transferring our approach to an online environment requires the development of a fully automated program with default parameters and the ability to react to varying circumstances (e.g. different distribution of purchase frequencies, higher or lower sparsity, etc.).

A further enhancement of the approach would be a more appropriate method to lower the effect of the HFC on the cluster results. We introduced a simple weighting measure in Section 3.1.4 and Section 4.2.2 that leads to a very unequal distribution of cluster sizes. The impact of highly frequently bought categories, or of low-frequency categories which do not contribute usefully to an expected customer segmentation, can also be removed by variable-selection methods (cf. Carmone Jr., Kara and Maxwell 1999, Brusco and Cradit 2001). These techniques might improve the results of the approach and should be tested in the future.

The constrained KCCA algorithm of our approach assigns each customer to a single centroid exclusively. As a consequence, a household is presented with one specific target marketing campaign. Sometimes, however, customers can belong to more than one group if they show ambiguous purchasing behavior. If budgets are not restricted and if the decision makers want to increase the chances of success of an appropriate targeting, the households could be presented with a second (or third) marketing campaign. From a methodological point of view, fuzzy clustering approaches could assign the transactions of the households to more than one centroid according to a probability value (cf. Chaturvedi et al. 1997, Dimitriadou, Weingessel and Hornik 2002). Merging constrained clustering with fuzzy methods poses an interesting challenge.

Another interesting modification of the approach would be the combination of the constrained partitioning of personalized transaction data with sequential pattern mining, as described in Section 2.3.2. After the KCCA algorithm has identified segments of customers with similar purchase histories, the sequential pattern mining could reveal segment-specific intertemporal purchase patterns. Such patterns might make possible a more appropriate timing of promotional activities. For example, applying the sequential-pattern mining algorithm of Agrawal and Srikant (1995) to the aggregated transactions of the wine segment might identify specific market basket combinations made before or after the purchase occasion in which the wines occurred. With this extension, the approach would not only suggest which customers to target with which offers, but would also isolate a point in time at which the promotional activity should be carried out (cf. Zhang and Krishnamurthi 2004).

It is to be hoped that other researchers will take up the above mentioned issues and help to develop more appropriate data-driven target marketing methods in the future.

Appendix

A Explanatory Appendix

A.1 Example: Pairwise Purchase Correlations

A.1.1 Similarity and Distance Matrices

	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$	$j=7$	$j=8$
$j=1$	0,00							
$j=2$	0,78	0,00						
$j=3$	0,90	0,40	0,00					
$j=4$	0,89	1,00	1,00	0,00				
$j=5$	0,79	0,73	0,73	0,82	0,00			
$j=6$	0,44	0,90	0,90	0,89	0,69	0,00		
$j=7$	0,92	0,75	0,75	0,88	0,55	0,82	0,00	
$j=8$	0,58	0,73	0,60	0,92	0,57	0,45	0,86	0,00

Table A.1: Distance matrix for pairwise association with the Jaccard distance measure

	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$	$j=7$	$j=8$
$j=1$	0.00							
$j=2$	2.65	0.00						
$j=3$	3.00	1.41	0.00					
$j=4$	2.83	2.65	2.65	0.00				
$j=5$	3.32	2.83	2.83	3.00	0.00			
$j=6$	2.00	3.00	3.00	2.83	3.00	0.00		
$j=7$	3.32	2.45	2.45	2.65	2.45	3.00	0.00	
$j=8$	2.65	2.83	2.45	3.32	2.83	2.24	3,46	0,00

Table A.2: Distance matrix for pairwise association with the Euclidean distance measure

	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$	$j=7$	$j=8$
$j=1$	0.00	0.50	0.75	0.67	0.70	0.29	0.83	0.50
$j=2$	0.71	0.00	0.25	1.00	0.70	0.86	0.67	0.70
$j=3$	0.86	0.25	0.00	1.00	0.70	0.86	0.67	0.60
$j=4$	0.86	1.00	1.00	0.00	0.80	0.86	0.83	0.90
$j=5$	0.57	0.25	0.25	0.33	0.00	0.43	0.17	0.40
$j=6$	0.29	0.75	0.75	0.67	0.60	0.00	0.67	0.40
$j=7$	0.86	0.50	0.50	0.67	0.50	0.71	0.00	0.80
$j=8$	0.29	0.25	0.00	0.67	0.40	0.14	0.67	0.00

Table A.3: Distance matrix for pairwise association with conditional probability

A.1.2 Graphical Visualization of Similarity and Distance Matrices

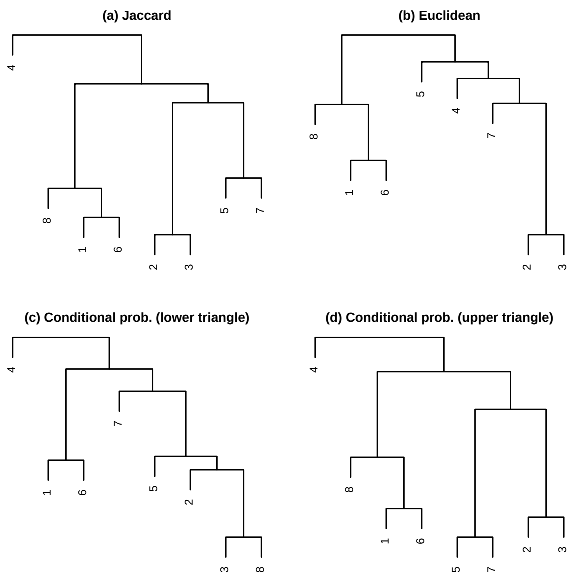

Figure A.1: Visualization of four distance matrices with a dendrogram derived from average linkage fusion algorithm of hierarchical clustering

A.2 Example: Association Rule Mining

To demonstrate the outcome of the APRIORI algorithm, the following two tables show the mined frequent itemsets and association rules for a minimum support value of 0.25 and a minimum confidence value of 0.7 in the example dataset of Table 2.1.

H_l	No.	Frequent itemsets.	Support
H_1	1	$\{j_2\}$	0.27
	2	$\{j_3\}$	0.27
	3	$\{j_7\}$	0.40
	4	$\{j_1\}$	0.47
	5	$\{j_6\}$	0.47
	6	$\{j_5\}$	0.67
	7	$\{j_8\}$	0.67
H_2	8	$\{j_3, j_8\}$	0.27
	9	$\{j_5, j_7\}$	0.33
	10	$\{j_1, j_6\}$	0.33
	11	$\{j_1, j_8\}$	0.33
	12	$\{j_5, j_6\}$	0.27
	13	$\{j_6, j_8\}$	0.40
	14	$\{j_5, j_8\}$	0.40
H_3	15	$\{j_1, j_6, j_8\}$	0.27

Table A.4: Frequent itemsets of the data sample mined with a $minsup = 0.25$

No.	Rules.	Support	Confidence	Lift
1	$\{j_3\} \Rightarrow \{j_8\}$	0.27	1.00	1.50
2	$\{j_7\} \Rightarrow \{j_5\}$	0.33	0.83	1.25
3	$\{j_1\} \Rightarrow \{j_6\}$	0.33	0.71	1.53
4	$\{j_6\} \Rightarrow \{j_1\}$	0.33	0.71	1.53
5	$\{j_1\} \Rightarrow \{j_8\}$	0.33	0.71	1.07
6	$\{j_6\} \Rightarrow \{j_8\}$	0.40	0.86	1.29
7	$\{j_1, j_6\} \Rightarrow \{j_8\}$	0.27	0.80	1.20
8	$\{j_1, j_8\} \Rightarrow \{j_6\}$	0.27	0.80	1.71

Table A.5: Association rules mined in the data sample with a $minsup = 0.25$ and a $minconf = 0.7$

A.3 Example: Recommender System with Memory-Based Collaborative Filtering

n	1	2	3	4	5	6	7	8
ω	0.14	0.33	0.00	1.00	0.25	0.00	0.17	1.00
ω^κ	0.03	0.07	0.00	0.20	0.05	0.00	0.03	0.20
n	9	10	11	12	13	14	15	
ω	0.20	0.75	0.33	0.40	0.40	1.00	0.00	
ω^κ	0.04	0.15	0.07	0.08	0.08	0.20	0.00	

Table A.6: Non-normalized and κ-normalized Tanimoto similarity values between the active shopping basket x_α and all transactions of X_N

Item	j_1	j_2	j_3	j_4	j_5	j_6	j_7	j_8
Median of $\gamma_{\cdot j}$	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.03

Table A.7: Median of the purchase probability value $\gamma_{\cdot j}$

n	j_1	j_2	j_3	j_4	j_5	j_6	j_7	j_8
1	0.00	0.03	0.03	0.00	0.00	0.00	0.03	-0.00
2	0.00	0.00	0.07	0.00	0.04	0.07	0.07	0.03
3	0.00	0.00	0.00	0.00	-0.03	0.00	0.00	-0.03
4	**0.20**	0.00	0.00	0.00	-0.03	**0.20**	0.00	**0.17**
5	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.02
6	0.00	0.00	0.00	0.00	-0.03	0.00	0.00	-0.03
7	0.00	0.03	0.03	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	-0.03	0.00	0.00	-0.03
9	0.04	0.00	0.00	0.04	0.01	0.00	0.00	-0.03
10	**0.15**	0.00	0.00	0.00	**0.12**	**0.15**	0.00	**0.12**
11	0.07	0.07	0.00	0.00	0.04	0.07	0.07	-0.03
12	0.00	0.00	0.00	0.08	0.05	0.08	0.00	0.05
13	0.08	0.08	0.08	0.00	-0.03	0.00	0.00	0.05
14	**0.20**	0.00	0.00	0.00	-0.03	**0.20**	0.00	**0.17**
15	0.00	0.00	0.00	0.00	-0.03	0.00	0.00	-0.03

Table A.8: $diff$-values according to Equation 2.20 with the four bolt maximal values.

A.4 Simulation: Prototypes of the Residual Household Segments

In addition to the market basket prototypes presented in Section 4.2.1, the following graphical pictograms describe the purchase frequencies within the segments.

A.4.1 Excluding the HFC from the Dataset

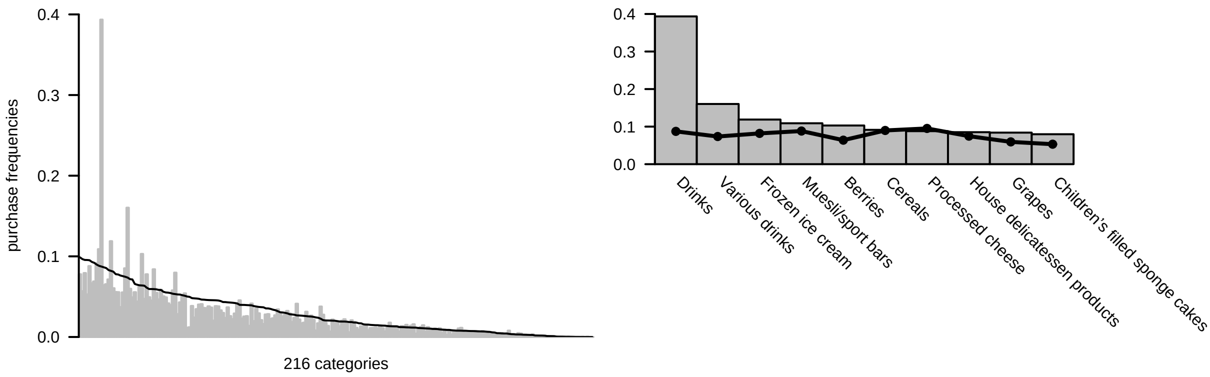

Figure A.2: Graphical illustration of the prototypical market basket of segment $k = 2$ (beverages cluster) with HFC excluded

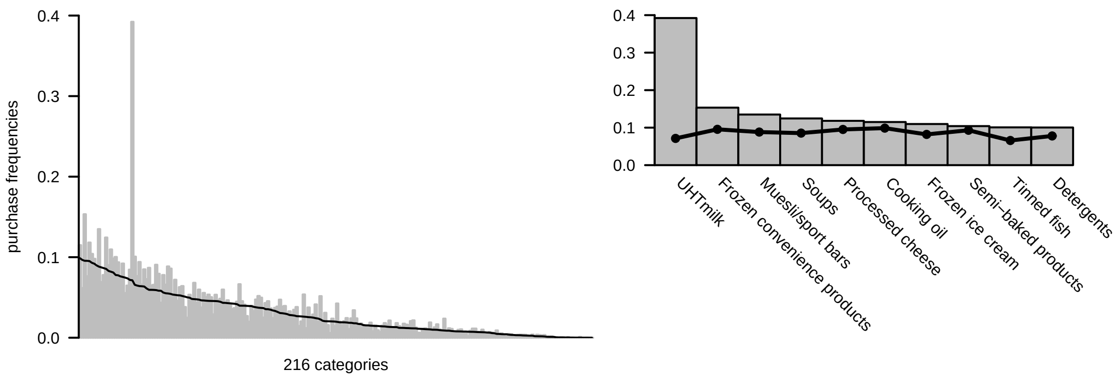

Figure A.3: Graphical illustration of the prototypical market basket of segment $k = 4$ (mix cluster) with HFC excluded

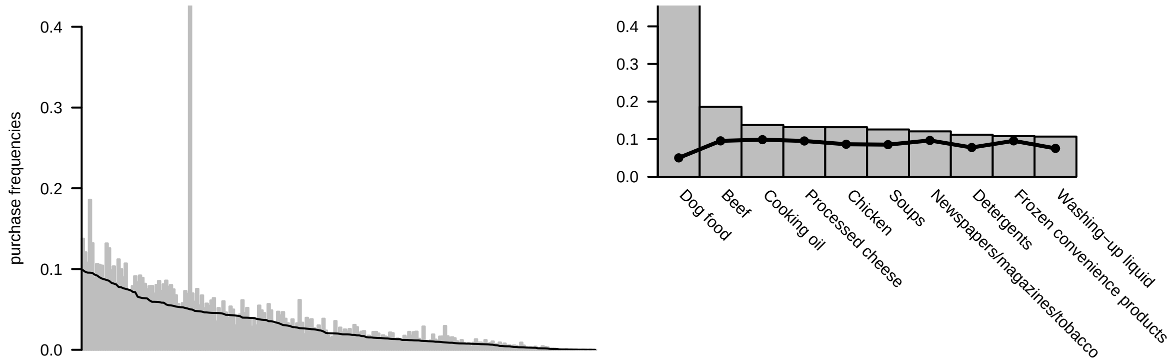

Figure A.4: Graphical illustration of the prototypical market basket of segment $k = 5$ (dog owners cluster) with HFC excluded

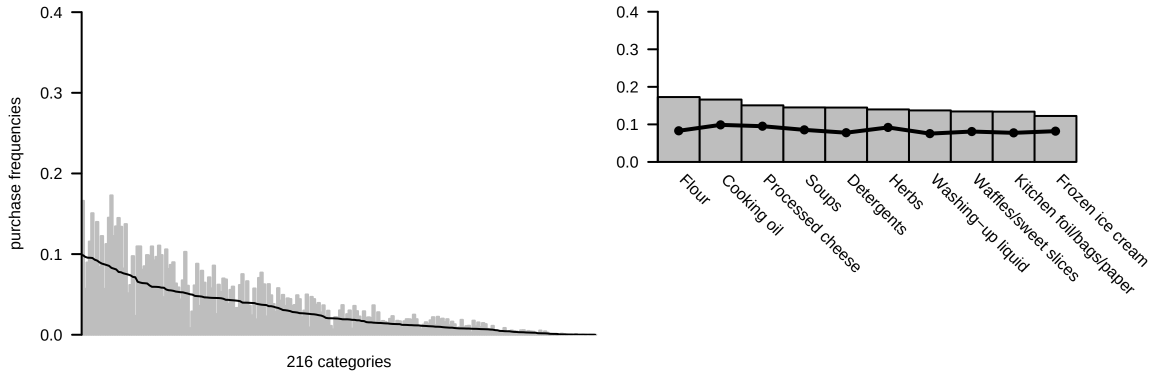

Figure A.5: Graphical illustration of the prototypical market basket of segment $k = 6$ (mix cluster) with HFC excluded

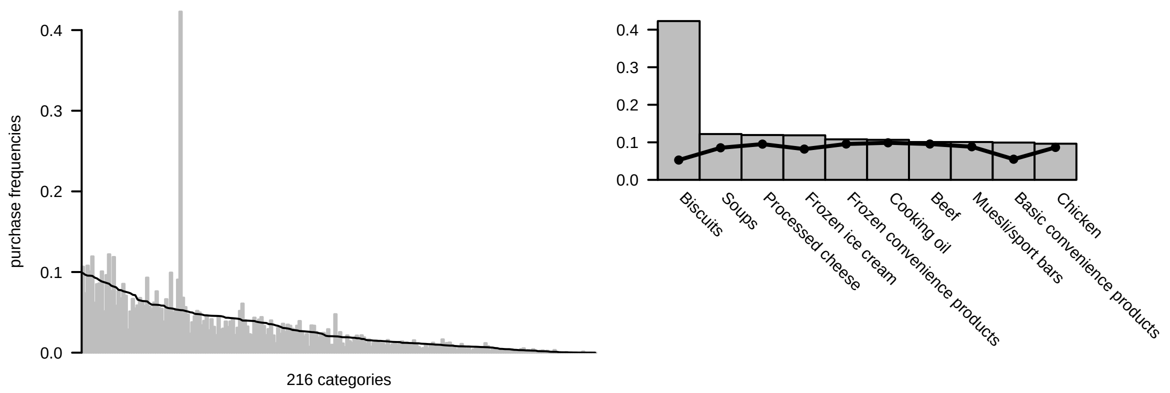

Figure A.6: Graphical illustration of the prototypical market basket of segment $k = 7$ (mix cluster) with HFC excluded

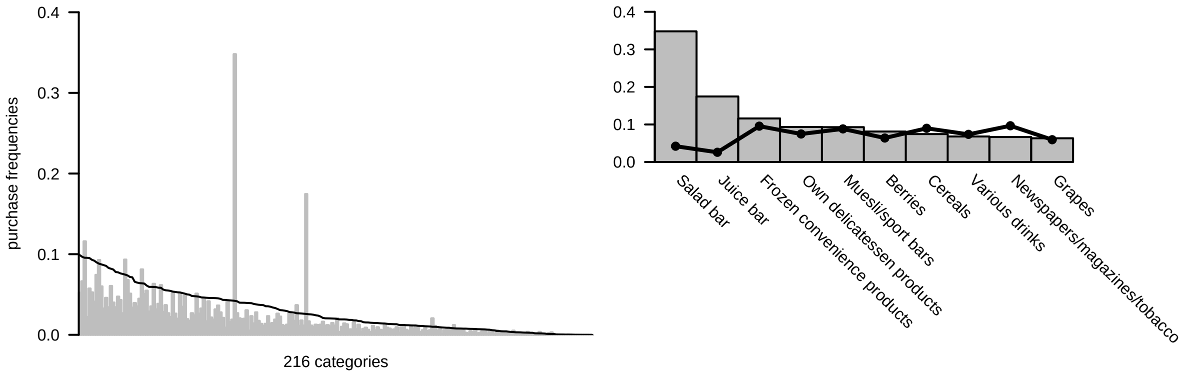

Figure A.7: Graphical illustration of the prototypical market basket of segment $k = 10$ (bar products cluster) with HFC excluded

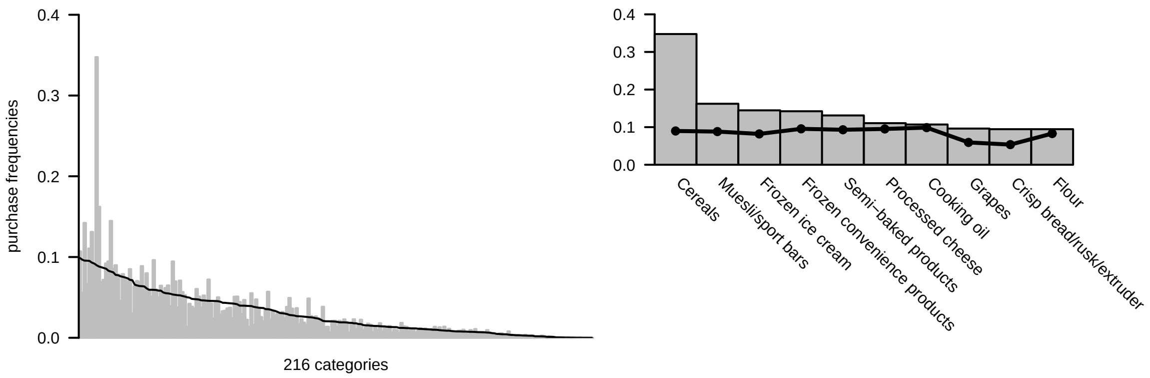

Figure A.8: Graphical illustration of the prototypical market basket of segment $k = 11$ (mix cluster) with HFC excluded

A.4.2 Weighting the Occurrence of HFC in the Dataset

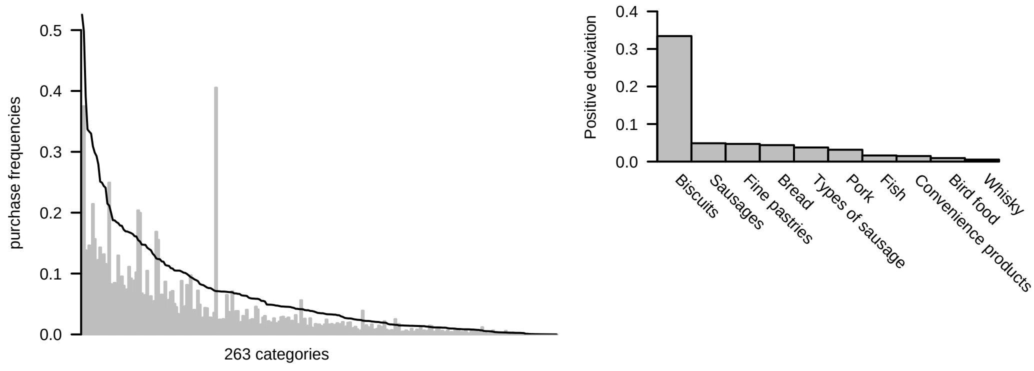

Figure A.9: Graphical illustration of the prototypical market basket of the mix segment no. 1 derived from the weighted data matrix

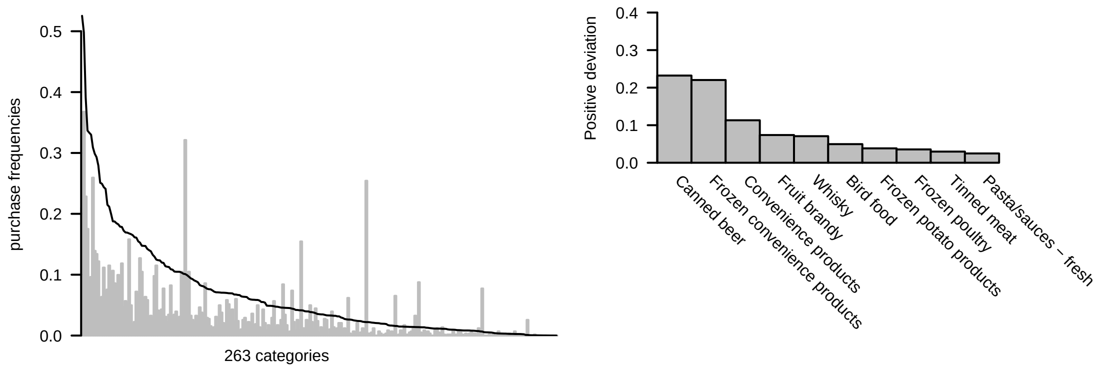

Figure A.10: Graphical illustration of the prototypical market basket of the mix segment no. 2 derived from the weighted data matrix

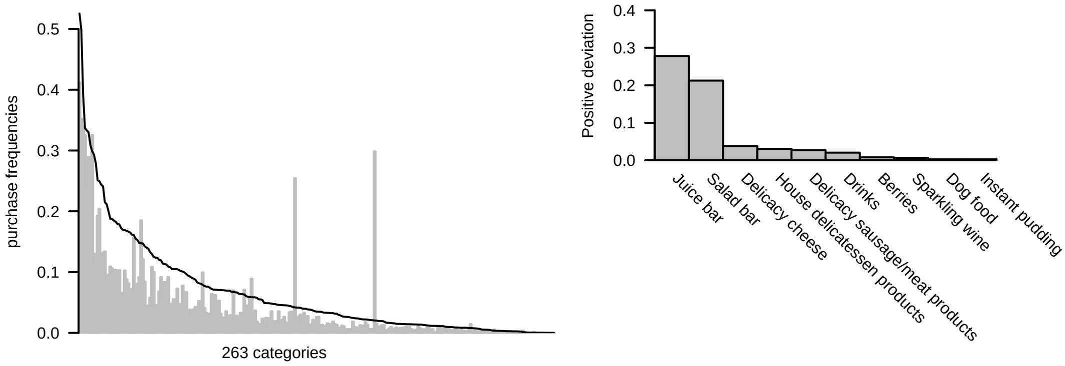

Figure A.11: Graphical illustration of the prototypical market basket of the bar products segment derived from the weighted data matrix

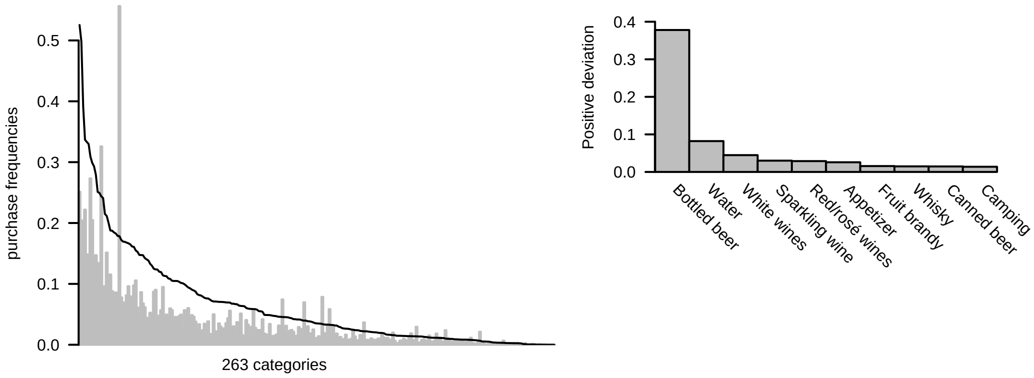

Figure A.12: Graphical illustration of the prototypical market basket of the beverages segment derived from the weighted data matrix

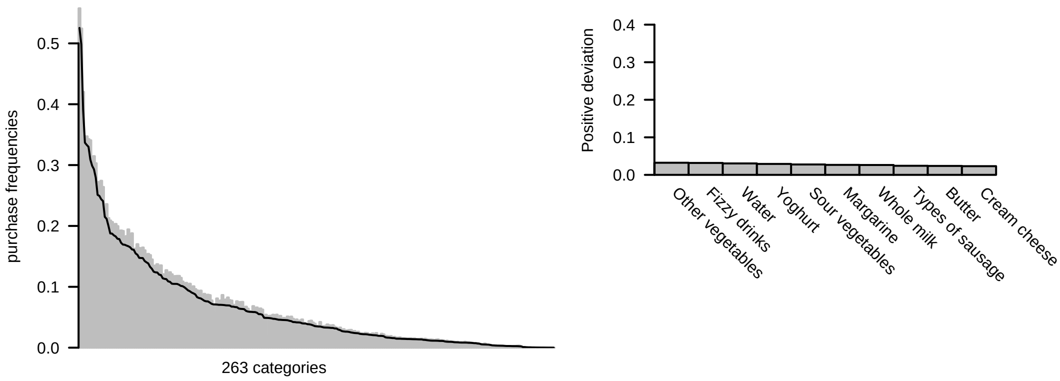

Figure A.13: Graphical illustration of the prototypical market basket of the mix segment no. 7 derived from the weighted data matrix

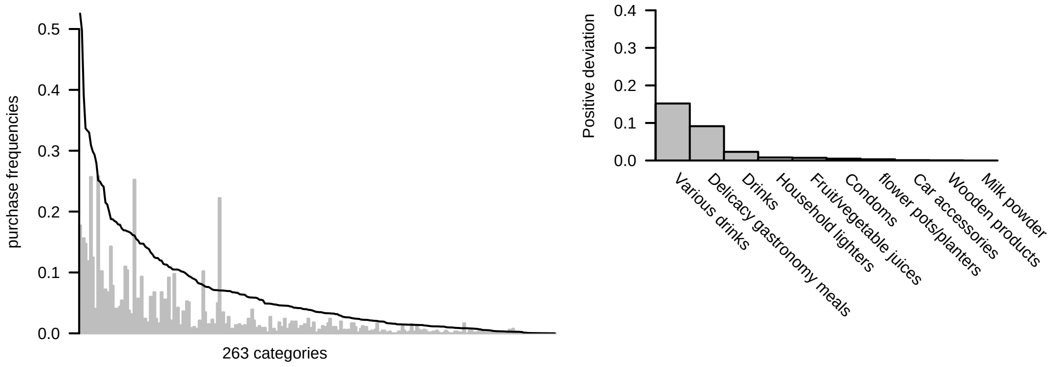

Figure A.14: Graphical illustration of the prototypical market basket of the mix segment no. 9 derived from the weighted data matrix

A.5 Simulation: Results

Segment	$k = 2$	$k = 4$	$k = 6$	$k = 7$
Cat no. 1	Drinks	Frozen ice cream	Herbs	Beef
Cat no. 2	Various drinks	Herbs	Cooking oil	Chicken
Cat no. 3	Frozen ice cream	Chicken	Chicken	Frozen ice cream
Cat no. 4	Chicken	Beef	Beef	Turkey

Table A.9: $\Phi = 4$ categories determined by the segment-specific approach in the segments $k = 2$, $k = 4$, $k = 6$ and $k = 7$

Segment	$k = 9$	$k = 10$	$k = 11$
Cat no. 1	Beef	Salad bar	Herbs
Cat no. 2	Frozen ice cream	Juice bar	Chicken
Cat no. 3	Chicken	House delicatessen products	Beef
Cat no. 4	Herbs	Frozen convenience products	House delicatessen products

Table A.10: $\Phi = 4$ categories determined by the segment-specific approach in the segments $k = 9$, $k = 10$ and $k = 11$

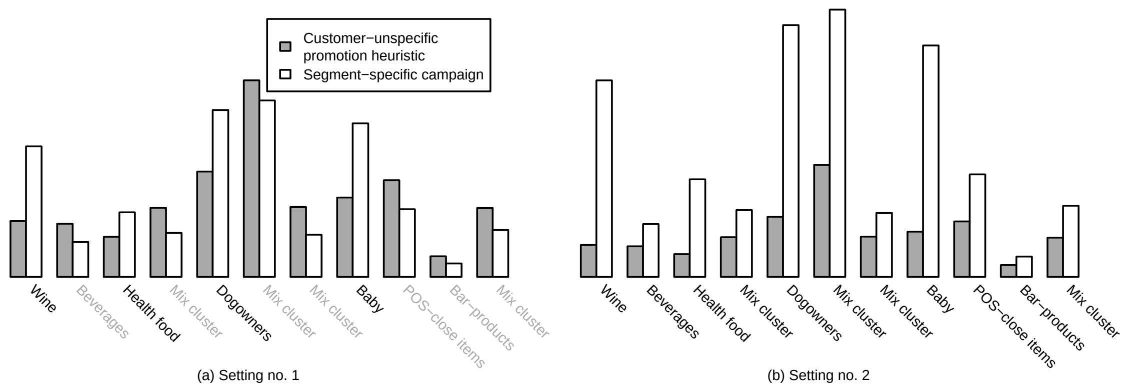

Figure A.15: Comparing the expected profit of the campaigns for (a) setting no. 1 and (b) setting no. 2 if the heurisitic implements the categories generating the highest revenue (option 2)

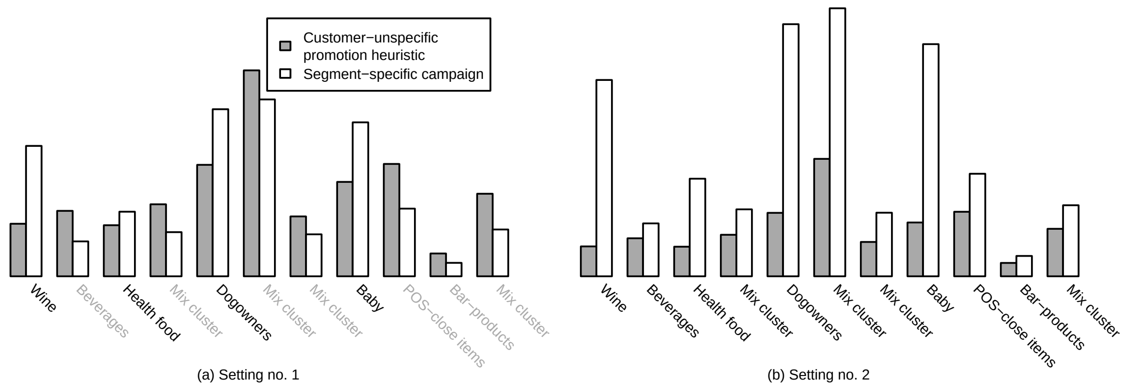

Figure A.16: Comparing the expected profit of the campaigns for (a) setting no. 1 and (b) setting no. 2 if the heurisitic implements the categories generating the highest support values (option 3)

B Technical Appendix

This technical appendix lists the main scripts and functions required to reproduce the results of the thesis. Since about 200 functions have been developed, only the most interesting programs and scripts are described. The full data analysis naturally comprises a great deal more programming effort.

The appendix shows examples of how to apply the functions in *R*. It should be noted that the functions are used to find a quick solution for the research problem instead of building an R-compliant program code. Hence, the naming of the functions (*f.*.R*) as well as the structure of the code itself does not follow the guidelines of the R-developement team. Moreover, most functions will have to be adjusted to the actual environment if an enhancement of the code is desired. For example, the paths to the included functions will have to be adjusted.

B.1 Data Preparation and Data Simulation

As described in Section 2.1.1, the transaction data is arranged in a table with each row representing a single transaction (i.e. market basket) and each column a category. The original receipt data of the supermarket chain, in contrast, was received in the following format (see file *receipt.txt*):.

```
14 24.03.1997 09:49      2394 099     48.70
14 24.03.1997 09:49      2394 211     11.50
14 24.03.1997 09:49      2394 212     19.90
14 24.03.1997 09:49      2394 216     21.80
14 24.03.1997 09:49      2394 217      7.90
14 24.03.1997 09:49      2394 234     14.90
```

The columns include the user ID (the customers), as well as the date, time, location, category and price. Additional information about the customers (e.g. sex, age) is stored in a second file called *passvar*.

To convert the receipt data into a matrix with rows and columns, the script *convert.pl* can be used. It requires the programming language PERL (see http://www.perl.org/). The default parameters are listed in the config file *settings.conf*. In addition, the file *itemclass* is needed, which contains the number of the given categories.

```
linux:~$ perl convert.pl receipt.txt
```

The output consists of two ASCII files: *outcontent_receipt.txt* and *outident_receipt.txt*. The first file includes the transactions and the second one the additional information, such as store ID and time of purchase. Both files can be imported easily into *R*. Header lines are added for both files by typing the following at the linux prompt:

```
linux:~$ cat header_content outcontent_receipt.txt > content
linux:~$ cat header_ident outident_receipt.txt > ident
```

After starting the *R* prompt, a data table can be created in the following manner:

```
> dm.test <- read.delim(file = "content", header = TRUE)
```

Since most functions and cluster algorithms require binary data, the function *f.dmasbin.R* converts the value of the matrix into one and zero values. The function needs the *bindata R* package.

```
> source("f.dmasbin.R")           # loading the function
> dm.test <- as.matrix(dm.test)   # converting data table into matrix object
> dm.test.b <- f.dmasbin(dm.test) # making binary data
```

To build a dataset with predefined grouping information for testing and simulation purposes (see Section 3.1.4), the *f.basket.R* is present. Its arguments are the number of buyers and the number of categories.

```
> dm.synthetic <- f.basket(buyers = 500, cat = 200)
```

Instead of generating *R* objects from the ASCII files, the author can provide interested readers with the actually used objects upon request. The objects are listed in Table B.1.

Object	**Description**
dm31.train.R	Training data sample (months 1-10)
dm31.hold.R	Hold-out data sample (months 11-12)
cl01.R	Flexclust cluster object from dm31.train.R
kv.dm31.train.R	Customer IDs of data sample
pdalle	Names of ategories (English)
pdall	Names of categories (German)

Table B.1: R objects of the main data analysis

B.2 Partitioning the Data

A major focus of the thesis is cluster algorithms. The R-package *flexclust* includes the algorithms needed to partition the data. A cluster process can be started using the following commands (for further details, please see the manual pages of the *flexclust* package):

```
> load("dm31.train.R")
> mycont <- new("flexclustControl", iter.max = 30, verbose = 1)
> kmfam <- kccaFamily(which = "ejaccard", groupFun = "minSumClusters")
> clsolution <- kcca(dm31.train, k = 11, family = kmfam,
+ control = mycont, group = kv.dm31.train)
```

To simplify the application of the *kcca*-function, the function *f.stepclust.R* shortens the listing above. It builds a list of K cluster solutions for $k = 1....K$ (*range*-value) with the smallest internal sum of distances of *nrep* cluster repetitions.

```
> cll <- f.stepclust(dm31.train, kv.vec = kv.dm31.train,
+ range = c(2:20), nrep = 5, which = "ejaccard")
```

For a *flexclust* cluster object, the function *f.result.R* prints the centroids with the names of the categories. The value $hv = 10$ means that the ten categories of the centroids with the highest mean values of the columns are displayed. The language of categories' names can be controlled by the *lang*-value. A value of 1 gives the German notation, while a value of 2 gives the English one.

```
> load("pdalle.R")
> load("cl01.R") # flexclust cluster object
> source("f.result.R")
> outl <- f.result(cl01, hv = 10, lang = 2)
> outl[[1]]
                                  [,1]
Red/rose' wines             0.32219020
White wines                 0.22824207
Beef                        0.11930836
Sparkling wine              0.11613833
Condensed milk              0.10605187
Cooking oil                 0.10547550
Herbs                       0.10432277
Appetizers                  0.10345821
Chicken                     0.09538905
House delicatessen products 0.09077810
```

The parameter $centers = TRUE$ means that the calculated centroids are used. If the original binary data matrix of the partitioning is present and the value is set to $FALSE$, the real mean values of the columns are calculated. With this option, it is possible to check whether the centroid values are different from the class means.

Similarly to *f.result.R*, the function *f.clsoldiff.R* prints the centroids of a cluster solution. Instead of *f.result.R*, it presents the highest positive and negative deviations between categories' overall purchase frequencies and class means (see Section 4.2.2).

The function *f.kopt.R* builds a list of several cluster solutions. Moreover, it implements voting indices as described by Dimitriadou, Dolnicar and Weingessel (2002). The program code of the indexes is extracted from the *clustIndex*-function of the *cclust* package. Among others, the *clustIndex* function includes the Davies-Bouldin index, the Xu index, the SSI and SSIW index, the Ratkowsky index and the Calinski index. To use the code with flexclust objects, the following functions were coded by slightly modifying the original code: *f.db.R*, *f.xu.R*, *f.ssi.R*, *f.ssiw.R*, *f.ratkowsky.R* and *f.calinski.R*.

```
> elist <- f.kopt(dm.sample, maxk = 20, nrep = 5,
+ gvec = kv.dm.b, kinit = mat.init)
```

The output of the *f.kopt.R* functions are several matrices which include the above-mentioned indices for an increasing K. To calculate the “minimum value of the second differences”, the *f.sediff.R* function can be used.

In addition to finding an appropriate value for K with indexes, a method with the corrected Rand index is shown. The package *e1071* includes the possibility of calculating the class agreement between two cluster solutions. The function *f.crandvec.R* implements a simple calculation for several cluster solutions which are summarized in a list object (e.g. made with *f.stepclust.R*). The option $cRand = TRUE$ applies the corrected Rand index. If it is set to $FALSE$, the normal Rand index is used.

```
> library(e1071)
> crandvec <- f.crandvec(cll, cRand = TRUE)
```

As described in Section 3.1.1, the a priori consideration of the buying histories can lead to representative transactions for each customer. A representative transaction of this kind includes a value of one for every category which was bought at least once. The function *f.apriori.R* extracts the representative transaction from a binary matrix and a corresponding vector with the customer IDs.

```
> dm.apriori <- f.apriori(dm31.train, kv.dm31.train)
```

The function *f.mvv.R* simulates the majority voting approach (see Section 3.1.2) if a cluster object and the corresponding customer IDs are present. Option $opt = 1$ means that a matrix is made. The first column of the matrix includes the ID of the customer. The second column shows the calculated class membership of the corresponding customer. If $opt = 2$ is set, the function returns a vector with the class membership value of each transaction.

```
> outmat <- f.mvv(cl01, kv.dm31.train, opt = 1)
> outmat[1:2,] # printing the first two rows

          kid cls
226640 226640   3
226671 226671   1
```

B.3 Rule Mining

The Comprehensive *R* Archive Network (CRAN) provides the package *arules*, which is a flexible and fast rule mining package for R. It includes all the functions which are needed for the analysis described in this thesis. Nevertheless, some functions were developed to reduce typing of commands. The following commands are used to start a simple mining process with the APRIORI algorithm (see Section 3.2):

```
> library(arules)
> library(bindata)
> source("f.dmasbin.R")
> trans.dm31.train <- as(f.dmasbin(dm31.train), "transactions")
> rules.dm31.train <- apriori(trans.dm31.train,
+ parameter = list(supp = 0.02, conf = 0.2, target = "rules"))
> inspect(rules.dm31.train)
```

The function *f.fimake.R* generates a list object with 70 frequent itemsets. These itemsets show the highest all-confidence values and a minimum length of at least two items. The data matrix *dm* is transferred into binary data if it contains metric values.

```
> dm.wine <- dm31.train[cl01 == 1,]
> fi.wine <- f.fimake(supp = 0.02, dm.wein, finumber = 70)
> fi.wine[[1]] # 70 itemsets with highest all-confidence
set of 70 itemsets
> fi.wine[[2]] # includes also the single items for PROFSET
set of 115 itemsets
```

The second object in the list includes also the single items of the frequent itemsets ($minlen = 1$). These items are needed in a subsequent step (see next section, *f.brijs.R*). In some situations, it can be useful to know how many customers support a specific itemset in the dataset. The function *f.setsupport.R* reveals this information.

B.4 Hierachical Clustering and PROFSET

Sections 4.3 and 4.4 describe the valuating and recommending of the mined itemsets. Hierachical clustering can be used to partition the items of a transaction data matrix (see Section 2.2.1). A simple way to cluster the matrix of Table 2.1 with *R* is shown below:

```
> library(cluster)
> load("dm01.R") # The data sample
> hcl.jac <- hclust(dist(t(dm01), method = "binary"), method = "average")
> plot(hcl.jac)
```

The function *f.condprob.R* calculates a distance matrix by using the conditional probability (see Section 2.2.1).

```
> dist.cb <- f.condprob(dm01)
> dist.cbl <- as.dist(1 - dist.cb, upper = FALSE)
> hcl.cbl <- hclust(dist.cbl, method = "average")
> plot(hcl.cbl)
```

Hierarchical cluster analysis can also be used to partition mined itemsets of a customer cluster as shown in Section 4.3. To partition itemsets, a distance measure is used. The function *f.gupta.R* builds the distance measure for frequent itemsets which have been mined with the package *arules*.

```
> trans.dm.wine <- as(dm.wine, "transactions")
> dist.wine <- f.gupta(fi.wine[[1]], trans.dm.wine)
[1]
[2]
...
[70]
> hcl.wine <- hclust(dist.wine, method = "ward")
> plot(hcl.wine)
```

An extended version of the function *f.gupta.R* is *f.guptae.R*, which can use the “tidList” information of the ECLAT mining algorithm (see the documentation of the *arules*-package). The function *f.toivonen.R* is a similar distance measure for frequent itemsets.

The calculation of the PROFSET model is separated into two stages (see Section 3.3.3). The function *f.brijs.R* comprises the first stage since it defines the profit margin for each frequent itemset. The output is used for the second function *f.brijssolver.R*, which solves the Mixed Integer Problem (MIP). .

```
> brijsvec.wine <- f.brijs(dm.wine, fi.wine[[2]])
[1] 1
[2] 2
...
> solv.brijs <- f.brijssolver(brijsvec.wine,
+ fi.wine[[2]], maxl = 4, dm.wine, realnames = TRUE)
> solv.brijs[[2]]
[[2]]
                 own_margin    total       csp
{Red/rose wines}   1838.850 104749.1 102910.25
{Sparkling wine}   6009.847  65893.4  59883.55
{White wines}      2939.575  61061.2  58121.62
{Beef}             4927.305  33954.9  29027.60
```

If one wishes to use a FMPS-file (Free Mathematical Programming System) to solve the MIP with another program, it is possible to use the function *f.prebrijs.R*.

B.5 Miscellaneous

This section includes some functions which were developed to support the analysis of the data. For example, since the categories are coded with numbers, the function *f.names.R* replaces the numbers with the corresponding names of the categories. The default language is German but it can be changed into English with the option $lang = 2$.

```
> string <- c("X211", "X040")
> f.names(string, lang = 2)
[1] "Whole milk" "Beef"
```

To quickly generate a random binary data table for testing purposes, the function *f.randomdm.R* can be used. The parameter *i* defines the items, *w* the number of transactions.

```
> f.randomdm(i = 8, w = 4)
  X1 X2 X3 X4 X5 X6 X7 X8
1  1  0  0  1  1  0  0  1
2  0  0  0  0  1  1  1  0
3  0  1  1  0  0  0  0  0
4  0  0  0  1  0  0  0  0
```

B.6 Examples of PERL and *R* Program Code

B.6.1 PERL script *convert.pl*

```
#!/usr/bin/perl
#
#
# UMWANDELN der BON-DATEN / CONVERTING the RECEIPT-DATA

# Einbinden der Konfigurations-Datei
do 'settings.conf';

print "Geben Sie den Namen der Input-Datei ein:\n";
$datei = <STDIN>;

print "Geben Sie den Wert der ersten Warenkorb-ID ein:\n";
chomp($wkstart = <STDIN>);

print "Wieviele unterschiedliche Kunden sollen generiert werden\?\n";
chomp($kidmax = <STDIN>);

# Initialisierungen

# Die Datei "itemclass" beinhaltet alle Nummerncodes der Waren

open(IN2, "itemclass");
chop(@itemclass = <IN2>);
close(IN2);
```

```
$numclass = scalar(@itemclass)-1; # ohne Kategorie X000

# Kopfzeile der Ausgabedatei

# Abfrage, ob eine Kopfzeile gem Konfig-Datei gewuenscht ist

if ( $head == 1 )
{

@itemclass_head = @itemclass;

# Loeschen der X000-Leerwarengruppe
shift(@itemclass_head);

open(KOPF1, ">>out_c_datei");
printf(KOPF1 "WK ");
printf(KOPF1 "ID ");
printf(KOPF1 "@itemclass_head\n");
close(KOPF1);

open(KOPF2, ">>out_i_datei");
printf(KOPF2 "WK ");
printf(KOPF2 "ID ");
printf(KOPF2 "DATUM ");
printf(KOPF2 "ZEIT ");
printf(KOPF2 "FILIALE\n");
close(KOPF2);
}

# Initialisierung Warenkorb-ID
# Wenn mehrere Input-Files angegeben werden,
# muss dem folgenden WK-Datensatz, eine neue
# WK-ID zugewiesen werden (Die letzte WK-ID
# des vorangehenden Datensatzes zzgl. eins)

if ($wkstart > 1)
{$wk = $wkstart;}
else {$wk = 1;}

$az = 0;
$kidit = 0;
#BEGINN der sequentiellen Dateneinlesung

open (IN, "$datei");
while (<IN>)
        {
        chomp($z = $_);

        if ( $nullwerte == 1)
        {
        # Durchsuchen der Zeile nach Betraegen mit Dezimalwerten ohne
        # fuehrende Null und Hinzufuegen der Null - falls in Konfig-
        # Datei gewuenscht.

        $z =˜ s/\040\056/0\056/g;
        }
```

```
($leer, $kundenid, $datum, $uhrzeit, $filiale, $item, $betrag) = split (/\040+/, $z);

        # Pruefung, ob eingelesener Betrag mit Komma- oder mit
        # Punktdezimaltrennern dargestellt werden soll

        if ( $komma == 1 )
        {
        $betrag =~ s/\056/\054/g;
        }

        # Pruefung, ob Negative Geldwerte (Auszahlungen, Retouren, Leergut etc)
        # in Null-Werte umgewandelt werden sollen

        if ( $nullnegativ == 1 )
        {
                if ( $betrag < 0 )
                {
                $betrag = 0;
                }
        }

        # Iteration zur Pruefung des Abbruchs nach kidit-Kunden

        if ( $kundenid != $lastid )
        {
        $kidit = $kidit + 1;
        }

        # Pruefung, ob ein neuer WK mit einer neuen WK-ID angelegt werden muss

        if ( ($kundenid == $lastid) && ($datum == $lastdat) && ($uhrzeit == $lastuhr) )
        {

                #ITEM-TESTSCHLEIFE

                &itemtest;

                sub itemtest
                {
                        while ($az != $numclass)
                        {
                        $az = $az+1;

                                if (@itemclass[$az] == $item)
                                {
                                push(@wkcontent,$betrag);
                                last;
                                }
                                else
                                {
                                push(@wkcontent,0);
                                }
                        }
                }
        }
        else
        {
```

```
                # AUSFUEHRUNG
                # Output des vorangehenden WKs, sofern ein neuer WK erstellt werden muss

                if ( $kidit == $kidmax )
                {
                last;
                }

                &output;

                # NEUER WARENKORB

                $az = 0;
                $wk = $wk+1;
                undef(@wkcontent);

                &itemtest;

                }

        $lastid = $kundenid;
        $lastfil = $filiale;
        $lastdat = $datum;
        $lastuhr = $uhrzeit;

        }

close (IN);
&output;

sub output
{

        # 0-Wert-Auffuellung - Durch die "last" Anweisung in der
        # Schleife, werden die letzten Produktklassen nach er-
        # folgreicher Zuweisung des letzten passenden Betrags
        # nicht mehr mit 0-Werten aufgefuellt. Die folgende Schleife
        # erledigt dies.

        while ($az != $numclass)
        {
        $az = $az+1;
        push(@wkcontent,0);
        }

        # AUSGABE in Datei

        open(OUT, ">>out_c_datei");
        printf(OUT "$wk ");
        printf(OUT "$lastid ");
        printf(OUT "@wkcontent\n");
        close(OUT);

        open(OUT2, ">>out_i_datei");
        printf(OUT2 "$wk ");
        printf(OUT2 "$lastid ");
```

```
        printf(OUT2 "$lastdat ");
        printf(OUT2 "$lastuhr ");
        printf(OUT2 "$lastfil\n");
        close(OUT2);
}

# Austausch der Leerzeichen durch Tabulatoren

system "sed -e 's/ /\t/g' out_c_datei > outcontent_$datei";
system "sed -e 's/ /\t/g' out_i_datei > outident_$datei";
system "rm out_i_datei";
system "rm out_c_datei";
exit
```

B.6.2 R-function *f.gupta.R*

```
# Function which builds a distance matrix from mined
# frequent itemsets according to Gupta/Strehl/Gosh (1999)

f.gupta <- function(rulz, trans){

z <- dim(trans)[1]
S <- quality(rulz)[,1]*z

# Transformation of the rules into a matrix (rule matrix)

rmx <- as(items(rulz), "matrix")
j <- dim(rmx)[1]
i <- j

# Initialisation of the distance matrix
distm.t <- matrix(0, ncol = j, nrow = i)

# k, j, y refer to columns / m, i, z refer to rows

        # The following function adds the j-row to all rows of the rule matrix
        f.sum <- function(x, j){ x + rmx[j, ] }

        for(k in 1:j)
        {
        # Prints the number of each rule for which the function
        # is calculating the distance measure

        print(k)

        # Combined item matrix of two rules: for every row of rmx the
        # following expression adds with f.sum each j-rule to each
        # j-rule of the rule matrix. Since the rows of the resulting
        # matrix represent the items, "t()" transposes it and "as()"
        # transforms it into an item matrix.

        rmxar <- as(t(apply(rmx, 1, f.sum, k)), "itemMatrix")
```

```
                # Calculating the distance measure between every rule (rows of rmx):
                # Due to the presence of item matrix (rmxar), the support of every
                # combined rule (scr) can be calculated within the transactions.

                for(m in k:i)
                {
                scr <- support(rmxar[m], trans, "absolute")
                distm.t[m, k] <- (1 - ( scr / (S[k] + S[m] - scr)))
                }
        }
distm.t <- as.dist(distm.t)
return(distm.t)
}
```

Bibliography

Aaker, D. A., Kumar, V. and George, S. (1998), *Marketing Research*, Wiley & Sons, New York, NY.

Aamodt, A. and Nygård, M. (1995), 'Different roles and mutual dependencies of data, information, and knowledge - an AI perspective on their integration.', *Data Knowl. Eng.* **16**(3), 191–222.

Acquisti, A. and Varian, H. R. (2005), 'Conditioning prices on purchase history', *Marketing Science* **24**(3), 367–381.

Adiguzel, F., Wedel, M. and Zhang, J. (2007), Customizing promotions across multiple categories, PhD thesis, Rijksuniversiteit Groningen. PhD Dissertation: Essays on Customization Applications in Marketing.

Adomavicius, G. and Tuzhilin, A. (2001), 'Using data mining methods to build customer profiles', *Computer* **34**(2), 74–82.

Agrawal, R., Imielinski, T. and Swami, A. (1993), Mining association rules between sets of items in large databases, *in* 'Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data', Washington, DC, pp. 207–216.

Agrawal, R., Mannila, H., Toivonen, H. and Verkamo, A. I. (1996), Fast discovery of association rules, *in* U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy, eds, 'Advances in Knowledge Discovery and Data Mining', AAAI Press, Menlo Rark, USA, pp. 307–328.

Agrawal, R. and Srikant, R. (1994), Fast algorithms for mining association rules, *in* 'Proceedings of the 20th International Conference on Very Large Databases', Santiago, Chile, pp. 487–499.

Agrawal, R. and Srikant, R. (1995), Mining sequential patterns, *in* 'Proceedings of the 11th International Conference on Data Engineering', Taipei, Taiwan, pp. 3–14.

Ahn, J. Y., Kim, S. K. and Han, K. S. (2003), 'On the design concepts for CRM system', *Industrial Management and Data Systems* **103**(5), 324–331.

Ainscough, T. and Aronson, J. (1999), 'An empirical investigation and comparison of neural networks and regression for scanner data analysis', *Journal of Retailing and Consumer Services* **6**(13), 205–217.

Ainslie, A. and Rossi, P. E. (1998), 'Similarities in choice behavior across product categories', *Marketing Science* **17**(2), 91–106.

Aldenderfer, M. S. and Blashfield, R. K. (1984), *Cluster Analysis*, Vol. 44 of *Quantitative Applications in the Social Sciences*, Sage University Paper, Beverly Hills.

Anderberg, M. R. (1973), *Cluster Analysis for Applications*, Academic Press, New York.

Anderson, C. (2006), *The Long Tail: how endless choice is creating unlimited demand*, RH Business Books, London.

Andrews, R. L. and Currim, I. S. (2002), 'Identifying segments with identical choice behaviors across product categories: An intercategory logit mixture model', *International Journal of Research in Marketing* **19**(1), 65–79.

Ansari, A., Essagaier, S. and Kohli, R. (2000), 'Internet recommendation systems', *Marketing Research* **37**(3), 363–375.

Bacher, J. (1996), *Clusteranalyse - Anwendungsorientierte Einführung*, 2 edn, Oldenbourg Verlag, München, Wien.

Backhaus, K., Plinke, W., Erichson, B. et al. (2006), *Multivariate Analysemethoden - Eine anwendungsorientierte Einführung*, 11 edn, Springer, Berlin.

Balka, K. (2005), Neue Methoden zur Marktsegmentierung durch Clustern von gruppierten Warenkorbdaten, Master thesis, Technische Universität Wien, Vienna, Austria. Friedrich Leisch, advisor.

Basu, S., Banerjee, A. and Mooney, R. J. (2004), Active semi-supervision for pairwise constrained clustering, *in* 'In Proceedings of the 2004 SIAM International Conference on Data Mining (SDM-04)', pp. 333–344.

Basu, S., Davidson, I. and Wagstaff, K. (2008), *Constrained Clustering: Advances in Algorithms, Theory, and Applications*, Chapman & Hall.

Bayardo, R. J. (1998), Efficiently mining long patterns from databases, *in* 'Proceedings of the 1998 ACM-SIGMOD International Conference on Managment of Data', pp. 85–93.

Bayardo, R. J., Agrawal, R. and Gunopulos, D. (1999), Constraint-based rule mining in large, dense databases, *in* ‘Proceedings of the 15th International Conference on Data Engineering’, pp. 188–197.

Bayardo, R. J. and Agrawal, S. (1999), Mining the most interesting rules, *in* ‘Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining’, pp. 145–154.

Bell, D. R. and Lattin, J. M. (1998), ‘Shopping behavior and consumer preference for store price format: why “large basket” shoppers prefer EDLP’, *Marketing Science* **17**(1), 66–88.

Berry, M. J. and Linoff, G. S. (2004), *Data mining techniques*, 2 edn, Wiley, Indianapolis.

Bijmolt, T. H. A., van Heerde, H. J. and Pieters, R. G. M. (2005), ‘New empirical generalizations on the determinants of price elasticity’, *JMR, Journal of Marketing Research* **42**(2), 141–156.

Blattberg, R. C. and Neslin, S. A. (1990), *Sales promotion - concepts, methods, and strategies*, Prentice Hall, Englewood Cliffs, NJ.

Bock, H.-H. (1999), Clustering and neural network approaches, *in* W. Gaul and H. Locarek-Junge, eds, ‘Classification in the Information Age, Proceedings of the 22nd Annual Conference of the Gesellschaft für Klassifikation e.V.’, Springer, Heidelberg, Germany, pp. 42–57.

Böcker, F. (1978), *Die Bestimmung der Kaufverbundenheit von Produkten*, Vol. 57 of *Schriften zum Marketing*, Duncker and Humblot.

Bodapati, A. V. (2008), ‘Recommendation systems with purchase data’, *Journal of Marketing Research* **45**(1), 77–93.

Bordemann, H.-G. (1985), Analyse von Verbundbeziehungen zwischen Sortimentsteilen im Einzelhandel, PhD thesis, Universiät Duisburg GH, Duisburg.

Boztug, Y. and Reutterer, T. (2008), ‘A combined approach for segment-specific analysis of market basket data’, *EJOR European Journal of Operational Research* **187**(1), 294–312.

Boztug, Y. and Silberhorn, N. (2006), ‘Modellierungsansätze in der Warenkorbanalyse im Überblick’, *Journal für Betriebswirtschaft* **56**(2), 105–128.

Breimann, L. (2001), ‘Statistical modeling: The two cultures (with comments and a rejoinder by the author)’, *Statistical science* **16**(3), 199–231.

Brijs, T., Goethals, B., Swinnen, G., Vanhoof, K. and Wets, G. (2000), A data mining framework for optimal product selection in retail supermarket data: the generalized PROFSET model, *in* 'KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining', ACM Press, New York, NY, USA, pp. 300–304.

Brijs, T., Swinnen, G., Vanhoof, K. and Wets, G. (2004), 'Building an association rules framework to improve product assortment decisions', *Data Mining and Knowledge Discovery* **8**(1), 7–23.

Brin, S., Motwani, R. and Silverstein, C. (1997), Beyond market baskets: Generalizing association rules to correlations, *in* J. Peckham, ed., 'SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, May 13-15, 1997, Tucson, Arizona, USA', ACM Press, pp. 265–276.

Brin, S., Motwani, R., Ullman, J. D. and Tsur, S. (1997), Dynamic itemset counting and implication rules for market basket data, *in* J. Peckham, ed., 'SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, May 13-15, 1997, Tucson, Arizona, USA', ACM Press, pp. 255–264.

Brusco, M. J. and Cradit, D. J. (2001), 'A variable-selection heuristic for k-means clustering', *Psychometrika* **66**(2), 249–270.

Carmone Jr., F., Kara, A. and Maxwell, S. (1999), 'A new model to improve market segment definition by identifying noisy variables', *Journal of Marketing Research* **36**(4), 501–509.

Cavique, L. (2007), 'A scalable algorithm for the market basket analysis', *Journal of Retailing and Consumer Services* **14**(6), 400–407.

Chaturvedi, A., Carroll, J. D., Green, P. E. and Rotondo, J. A. (1997), 'A feature-based approach to market segmentation via overlapping k-centroids clustering', *Journal of Marketing Research* **34**(3), 370–377.

Cheung, D. W., Ng, V. T., Fu, A. W. and Fu, Y. (1996), 'Efficient mining of association rules in distributed databases', *IEEE Transactions On Knowledge And Data Engineering* **8**(6), 911–922.

Chintagunta, P. and Song, I. (2007), 'A discrete/continuous model for multi-category purchase behavior of households', *Journal of Marketing Research* **44**(4), 595–612.

Church, A. (1936), 'A note on Entscheidungsproblem', *Journal of Symbolic Logic* **1**(1), 40–41. Correction 1 (1936): pp. 101-102.

Cody, W. F., Kreulen, J. T., Krishna, V. and Spangler, W. S. (2002), 'The integration of business intelligence and knowledge management', *IBM Systems Journal* **41**(4), 697–713.

Cohen, J. (1960), 'A coefficient of agreement for nominal scales', *Educational and Psychological Measurement* **20**(1), 37–46.

Coombs, C. H. (1964), *A Theory of Data*, Wiley, New York.

Cormen, T. H. (2007), *Algorithmen - Eine Einführung*, Oldenbourg, München.

Cottle, D. (2000), *Professional's guide to target marketing*, Harcourt Prof. Publishing.

D'Agostino, R. B. and Stephens, M. A. (1986), *Goodness-of-fit techniques*, Marcel Dekker Inc., New York.

Dakin, R. J. (1965), 'A tree-search algorithm for mixed integer programming problems', *The Computer Journal* **8**(3), 250–255.

Davies, D. L. and Bouldin, D. W. (1979), 'A cluster separation measure', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **1**(2), 224–227.

Decker, R. (2005), 'Market basket analysis by means of a growing neural network', *The International Review of Retail, Distribution and Consumer Research* **15**(2), 151–169.

Decker, R. and Monien, K. (2003), 'Market basket analysis with neural gas networks and self-organising maps', *Journal of Targeting, Measurement and Analysis for Marketing* **11**(4), 373–387.

Decker, R. and Schimmelpfennig, H. (2002), Alternative Ansätze zur datengestützten Verbundmessung im Electronic Retailing, *in* D. Ahlert, R. Olbricht and H. Schröder, eds, 'Jahrbuch Handelsmanagement 2002', Frankfurt am Main, pp. 193–212.

Dibb, S. and Simkin, L. (1996), *The market segmentation workbook - target marketing for marketing managers*, 1. edn, Routledge, London.

Dimitriadou, E., Dolnicar, S. and Weingessel, A. (2002), 'An examination of indexes for determining the number of clusters in binary data sets', *Psychometrika* **67**(1), 137–160.

Dimitriadou, E., Weingessel, A. and Hornik, K. (2002), 'A combination scheme for fuzzy clustering', *International Journal of Pattern Recognition and Artificial Intelligence* **16**(7), 901–912.

Dolnicar, S., Leisch, F., Steiner, G. and Weingessel, A. (1998), A comparison of several cluster algorithms on artificial binary data scenarios from tourism marketing, Working Paper 19, SFB "Adaptive Information Systems and Modeling in Economics and Management Science".
URL: *http://www.wu-wien.ac.at/am/workpap.htm*

Dong, G. and Pei, J. (2007), *Sequence Data Mining - Advances in Database Systems*, Springer, Berlin.

Drèze, X. and Hoch, S. J. (1998), 'Exploiting the installed base using cross-merchandising and category destination programs', *International Journal of Research in Marketing* **15**(5), 459–471.

Duchessi, P., Schaninger, C. M. and Nowak, T. (2004), 'Creating cluster-specific purchase profiles from point-of-sale scanner data and geodemographic clusters: improving category management at a major us grocery chain', *Journal of Consumer Behaviour* **4**(2), 97–117.

Duda, R. O., Hart, P. E. and Stork, D. G. (2001), *Pattern Classification*, 2 edn, Wiley, New York.

Elberse, A. (2008), 'Should you invest in the long tail?', *Harvard Business Review* **86**(7/8), 88–96. HBS Centennial Issue.

Fader, P. S., Hardie, B. G. S. and Lee, K. L. (2005), 'RFM and CLV: Using iso-value curves for customer base analysis', *Journal of Marketing Research* **42**(4), 415–430.

Fayyad, U., Grinstein, G. G. and Wierse, A. (2002), *Information visualization in data mining and knowledge discovery*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Fayyad, U. M., Piatetsky-Shapiro, G. and Smyth, P. (1996), 'From data mining to knowledge discovery in databases.', *Artificial Intelligence Magazine* **17**(3), 37–54.

Fisher, D. H. (1987), 'Knowledge acquisition via incremental conceptual clustering', *Machine Learning* **2**, 139–172.

Fox, E. J. and Hoch, S. J. (2005), 'Cherry-picking', *Journal of Marketing* **69**(1), 46–62.

Franke, N. and Mazanec, J. (2006), 'The six scientific identities of marketing: A vector quantization of research approaches', *European Journal of Marketing* **5/6**(40), 634–661.

Fritzke, B. (1997), Some competitive learning methods.
URL: *URL: http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/JavaPaper/*

Fukuda, T., Morimoto, Y., Morishita, S. and Tokuyama, T. (1996), Data mining using two-dimensional optimized association rules: Scheme, algorithms, and visualization, *in* 'Proc. of the 1996 ACM-SIGMOD International Conference on the Management of Data', pp. 13–23.

Gonzales, M. L. (2003), *IBM Data Warehousing*, Wiley.

Gordon, A. D. (1981), *Classification*, Monographs on Applied Probability and Statistics, Chapman and Hall, London.

Gordon, A. D. and Vichi, M. (1998), 'Partitions of partitions', *Journal of Classification* **15**(2), 265–285.

Gosh, A. K. (1997), 'Targeted promotions using scanner panel data', *Journal of Product and Brand Management* **6**(6), 405–416.

Gouda, K. and Zaki, M. J. (2005), 'GenMax: An Efficient Algorithm for Mining Maximal Frequent Itemsets', *Data Mining and Knowledge Discovery* **11**(3), 223–242.

Grapentine, T. and Boomgaarden, R. (2003), 'Maladies of market segmentation - the promise of this technique still remains untapped', *Marketing Research* **15**(1), 27–31.

Grinstein, G. G. and Ward, M. O. (2002), Introduction to data visualization, *in* U. Fayyad, G. G. Grinstein and A. Wierse, eds, 'Information visualization in data mining and knowledge discovery', Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 21–45.

Grünblatt, M. (2001), Verfahren zur Analyse von Scanningdaten - Nutzenpotentiale, praktische Probleme und Entwicklungsperspektiven, *in* R. Olbricht, ed., 'Berichte aus dem Lehrstuhl für Betriebswirtschaftslehre, insb. Marketing', Vol. 5 of *Forschungsbericht*, Fernuniversität Hagen - Fachbereich Wirtschaftswissenschaften, Hagen.

Günter, T. M. and Klapper, D. (2007), 'Do the long-run category demand effects of retailer promotions vary across different store types?', *Marketing - Journal of Research and Management* **3**(1), 17–33.

Gupta, G. K., Strehl, A. and Gosh, J. (1999), Distance based clustering of association rules, *in* 'Intelligent Engineering Systems Through Artificial Neural Networks', Vol. 9, ANNIE, ASME Press, pp. 759–764.

Hagerup, T. and Rüb, C. (1990), 'A guided tour of chernoff bounds', *Information Processing Letters* **33**(6), 305–308.

Hahsler, M., Grün, B. and Hornik, K. (2005), 'A computational environment for mining association rules and frequent item sets', *Journal of Statistical Software* **14**(15), 1–25.

Hahsler, M., Hornik, K. and Reutterer, T. (2005), Implications of probabilistic data modeling for rule mining, Report 14, Research Report Series, Department of Statistics and Mathematics, Wirtschaftsuniversität Wien, Augasse 2–6, 1090 Wien, Austria.

Hahsler, M., Hornik, K. and Reutterer, T. (2006), Warenkorbanalyse mit hilfe der statistik-software r, *in* P. Schnedlitz, R. Buber, T. Reutterer, A. Schuh and C. Teller, eds, 'In Innovationen in Marketing und Handel', Linde, Wien, pp. 144–163.

Hamming, R. W. (1950), 'Error detecting and error correcting codes', *Bell System Technical Journal* **26**(2), 147–160.

Han, J. and Kamber, M. (2000), *Data Mining: Concepts and Techniques*, Morgan Kaufmann.

Hand, D. J. and Krzanowski, W. J. (2005), 'Optimising k-means clustering results with standard software packages', *Computational Statistics and Data Analysis* **49**(4), 969–973.

Harding, F. (2002), *Cross selling success*, Adams Media, Avon, Mass.

Hartigan, J. A. and Wong, M. A. (1979), 'A K-means clustering algorithm', *Applied Statistics* **28**(1), 100–108.

Hastie, T., Tibshirani, R. and Friedman, J. H. (2001), *The Elements of Statistical Learning*, Springer, New York, NY.

Heinzl, A. (2001), Zum Aktivitätsniveau empirischer Forschung in der Wirtschaftsinformatik - Erklärung und Handlungsoptionen, *in* H. Böhler and J. Sigloch, eds, 'Unternehmensführung und empirische Forschung: Festschrift zum 65. Geburtstag von Peter Rötger Wossidlo', Arbeitspapier 7/2001, Universiät Bayreuth, Lehrstuhl für Betriebswirtschaftslehre VII - Wirtschaftsinformatik, Bayreuth.

Hettich, S. and Hippner, H. (2001), Assoziationsanalyse, *in* H. Hippner, U. L. Küsters, M. Meyer and K. Wilde, eds, 'Handbuch Data Mining im Marketing - Knowledge Discovering in Marketing Databases', Viewag, Wiesbaden, pp. 427–463.

Hilderman, R. J. and Hamilton, H. J. (2001), *Knowledge discovery and measures of interest*, Kluwer, Boston, Mass.

Homburg, C. (2007), 'Betriebswirtschaftslehre als empirische Wissenschaft - Bestandsaufnahme und Empfehlung', *ZfbF - Zeitschrift für betriebswirtschaftliche Forschung* **56**(7), 27–60.

Hornik, K. (2005*a*), 'A CLUE for CLUster Ensembles', *Journal of Statistical Software* **14**(12), 1–25.

Hornik, K. (2005*b*), Cluster ensembles, *in* C. Weihs and W. Gaul, eds, 'Classification — The Ubiquitous Challenge', Springer-Verlag, Heidelberg, pp. 65–72. Proceedings of the 28th Annual Conference of the Gesellschaft für Klassifikation e.V., University of Dortmund, March 9–11, 2004.

Hruschka, H. (1991), 'Bestimmung der Kaufverbundenheit mit Hilfe eines probalistischen Messmodells', *Zeitschrift für betriebswirtschaftliche Forschung* **43**(5), 418–434.

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

Hübl, G. and Trifts, V. (2000), 'Consumer decision making in online shopping environments: The effects of interactive decision aids', *Marketing Science* **19**(1), 4–21.

Hughes, A. M. (1996), *The Complete Database Marketer*, McGraw Hill, New York, NY.

Hui, X., Tan, P.-N. and Kumar, V. (2006), 'Hyperclique pattern discovery', *Data Mining and Knowledge Discovery* **13**(2), 219–242.

Jäger, A. (2007), *RFID als Marketingstrategie : Einsatzpotentiale im Einzelhandel*, VDM Verlag Müller, Saarbrücken.

Jamshaid, S., Jalil, Z., Khiyal, M. S. H. and Saeed, M. I. (2007), 'Association rule mining in centralized databases', *Information Technology Journal* **6**(2), 174–181.

Jiang, J. J., Klein, G. and Pick, R. A. (1998), 'A marketing category management system: a decision support system using scanner data', *Decision Support Systems* **23**(3), 259–271.

Kahn, B. E. and McAllister, L. (1997), *Grocery Revolution: The New Focus on the Consumer*, Addision-Wesley, Reading, MA.

Kanitscheider, B. (1981), *Wissenschaftsthoerie der Naturwissenschaft*, de Gruyter.

Kanji, G. K. (1993), *100 Statistical Tests*, SAGE Publications, Newbury Park, Cal.

Kaufman, L. and Rousseeuw, P. J. (2005), *Finding Groups in Data: An Introduction to Cluster Analysis.*, John Wiley.

Kelley, S. (1996), *Data Warehousing: The Route to Mass Customization*, John Wiley & Sons, New York.

Kleinberg, J. M., Papadimitriou, C. H. and Raghavan, P. (1998), 'A microeconomic view of data mining.', *Data Mining Knowledge Discovering* **2**(4), 311–324.

Kohonen, T. (1982), 'Self-organized formation of topologically correct feature maps', *Biological Cybernetics* **43**, 59–69.

Kohonen, T. (2000), *Self-Organizing Maps*, 3 edn, Springer.

Kumara, V. and Shahb, D. (2004), 'Building and sustaining profitable customer loyalty for the 21st century', *Journal of Retailing* **80**(4), 317–329.

Kuonen, D. (2005), ‘Is data mining for gold ’statistical déjà vu’?’, *CRM Zine* **53**. Published by CRM Today.

Lal, R. and Bell, D. (2003), ‘The impact of frequent shopper programs in grocery retailing’, *Quantitative Marketing and Economics* **1**(2), 179–202.

Lawrence, R. D., Almasi, G. S., Kotlyar, V., Viveros, M. S. and Duri, S. S. (2001), ‘Personalization of supermarket product recommendations’, *Data Mining and Knowledge Discovering* **5**(1-2), 11–32.

Leenheer, J. (2004), The Adoption and Effectiveness of Loyalty Programs in Retailing, PhD thesis, Tilburg University, Netherlands.

Leisch, F. (2006), ‘A toolbox for k-centroids cluster analysis’, *Computational Statistics and Data Analysis* **51**(2), 526–544.

Leisch, F. and Grün, B. (2006), Extending standard cluster algorithms to allow for group constraints, *in* A. Rizzi and M. Vichi, eds, ‘Compstat 2006, Proceedings in Computational Statistics’, Physica-Verlag, Heidelberg, pp. 885–892.

Liebermann, Y. (1999), ‘Membership clubs as a tool for enhancing buyers patronage’, *Journal of Business Resarch* **45**(3), 291–297.

Luo, C. and Chung, S. M. (2004), A scalable algorithm for mining maximal frequent sequences using sampling, *in* ‘Tools with Artificial Intelligence’, ICTAI, 16th IEEE International Conference, pp. 156–165.

Mallach, E. G. (2000), *Decision Support and Data Warehouse Systems*, Irwin McGraw-Hill.

Manchanda, P., Ansari, A. and Gupta, S. (1999), ‘The ”shopping basket”: a model for multicategory purchase incidence decisions’, *Marketing Science* **18**(2), 95–114.

Mannila, H. (1997), Methods and problems in data mining, *in* F. N. Afrati and P. G. Kolaitis, eds, ‘Database Theory - ICDT ’97, 6th International Conference, Delphi, Greece, January 8-10, 1997, Proceedings’, Vol. 1186 of *Lecture Notes in Computer Science*, Springer, pp. 41–55.

Mazanec, J. (2001), ‘Neural market structure analysis: Novel topology-sensitive methodology’, *European Journal of Marketing* **35**(7-8), 894–916.

McQueen, J. (1967), Some methods for classification and analysis of multivariate observations, *in* ‘Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability’, University of California Press, Berkeley, CA, USA, pp. 281–297.

Merkle, E. (1981), *Die Erfassung und Nutzung von Informationen über den Sortimentsverbund in Handelsbetrieben*, Vol. 11 of *Schriften zum Marketing*, Duncker and Humblot, Berlin.

Mild, A. and Reutterer, T. (2001), Collaborative filtering methods for binary market basket data analysis, *in* 'AMT '01: Proceedings of the 6th International Computer Science Conference on Active Media Technology', Springer-Verlag, London, UK, pp. 302–313.

Milligan, G. W. and Cooper, M. C. (1985), 'An examination of procedures for determining the number of clusters in a data set', *Psychometrika* **50**(2), 159–179.

Mulhern, F. J. (1999), 'Customer profitability analysis. measurement, concentration, and future directions', *Journal of Interactive Marketing* **13**(1), 25–40.

Müller-Hagedorn, L. (2005), *Handelsmarketing*, 4 edn, Kohlhammer, Stuttgart.

Neckel, P. and Knoblauch, B. (2005), *Customer relationship analytics - Praktische Anwendung des Data Mining im CRM*, Dpunkt-Verlag, Heidelberg.

Omiecinski, E. (2003), 'Alternative interest measures for mining associations in databases.', *IEEE Transactions on Knowledgle and Data Engineering* **15**(1), 57–69.

Ordonez, C. (2003), Clustering binary data streams with k-means, *in* 'DMKD '03: Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery', ACM, New York, NY, USA, pp. 12–19.

Ottmann, T. and Widmayer, P. (2002), *Algorithmen und Datenstrukturen*, 4 edn, Spektrum, Akademischer Verlag.

Papastefanou, G. (2001), The ZUMA data file version of the GfK ConsumerScan Household Panel, *in* G. Papastefanou, P. Schmidt, A. Börsch-Supan, H. Lüdke and U. Oltersdorf, eds, 'Social and Economic Analyses of Consumer Panel Data, Zentrum für Umfragen, Meinungen und Analysen (ZUMA)', Mannheim, pp. 206–212.

Passingham, J. (1998), 'Grocery retailing and the loyalty card', *Journal of Market Research Society* **40**(1), 55–63.

Piatetsky-Shapiro, G. (1991), *Knowledge Discovery in Databases*, AAAI/MIT Press, Cambridge, MA.

Poggenpohl, M. (1993), Verbundanalyse im Einzelhandel auf der Grundlage von Kundenkarteninformationen – Eine empirische Untersuchung von Verbundbeziehungen zwischen Abteilungen, PhD thesis, Universität Münster.

R Development Core Team (2005), *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.
URL: *http://www.R-project.org*

Rand, W. M. (1971), 'Objective criteria for the evaluation of clustering methods', *Journal of the American statistical association* **66**(336), 846–850.

Razi, F., Kausar, S., Khiyal, M. S. H. and Saeed, M. I. (2008), 'Comparative study of randomized partition algorithm with CMA algorithm', *European Journal of Scientific Research* **19**(3), 482–490.

Reichheld, F. F. (2001), *The loyalty effect - the hidden force behind growth, profits, and lasting value*, Harvard Business School, Boston, Mass.

Reinartz, W. J. and Kumar, V. (2000), 'On the profitability of long-life customers in a noncontractual setting: An empirical investigation and implications for marketing', *Journal of Marketing* **64**(4), 17–35.

Reischuk, R. (1990), *Einführung in die Komplexitätstheorie*, 2 edn, Teubner, Stuttgart.

Reutterer, T., Hahsler, M. and Hornik, K. (2007), 'Data mining und marketing am beispiel der explorativen warenkorbanalyse', *Marketing - Zeitschrift fr Forschung und Praxis (ZFP)* **29**(3), 163–179.

Reutterer, T. and Mild, A. (2003), 'An improved collaborative filtering approach for predicting cross-category purchases based on binary market basket data', *Journal of Retailing and Consumer Services* **10**(3), 123–133.

Reutterer, T., Mild, A., Natter, M. et al. (2006), 'A dynamic segmentation approach for targeting and customizing direct marketing campaigns', *Journal of Interactive Marketing* **20**(3), 43–57.

Ripley, B. D. (1996), *Pattern recognition and neural networks*, Cambridge University Press, Cambridge, UK.

Rossi, P. E., McCulloch, R. E. and Allenby, G. M. (1996), 'The value of purchase history data in target marketing', *Marketing Science* **15**(4), 321–340.

Runte, M. (2000), Individualisierte Angebote mit Collaborative Filtering, PhD thesis, Christian-Albrechts-Universität Kiel, Wiesbaden. DUV.

Russell, G. J. and Petersen, A. (2000), 'Analysis of cross category dependence in market basket selection', *Journal of Retailing* **76**(3), 367–392.

Rust, R. T., Lemon, K. and Zeithaml, V. A. (2004), ‘Return on marketing: Using customer equity to focus marketing strategy’, *Journal of Marketing* **68**(1), 109–127.

Sakthi, U., Hemalatha, R. and Bhuvaneswaran, R. S. (2008), ‘Parallel and distributed mining of association rule on knowledge grid.’, *Proceedings of World Academy of Science: Engineering & Technology* **32**, 346 – 350.

Savasere, A., Omiecinski, E. and Navathe, S. B. (1995), An efficient algorithm for mining association rules in large databases, *in* ‘VLDB ’95: Proceedings of the 21th International Conference on Very Large Data Bases’, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 432–444.

Schnedlitz, P. and Kleinberg, M. (1994), ‘Verbundanalyse im Lebensmittelhandel’, *Der Markt* **33**(1), 31–39.

Schnedlitz, P., Reutterer, T. and Joos, W. (2001), Data-Mining und Sortimentsverbundanalyse im Einzelhandel, *in* H. Hippner, U. L. Küsters, M. Meyer and K. Wilde, eds, ‘Handbuch Data Mining im Marketing - Knowledge Discovering in Marketing Databases’, Viewag, Wiesbaden, pp. 951–970.

Schulz, H.-J. and Nocke, T. (2007), Maschinelle Datenanalyse im Informationszeitalter - Können oder müssen wir ihr vertrauen?, *in* R. Langner, T. Luks, A. Schlimm, G. Straube and D. Thomaschke, eds, ‘Ordungen des Denkens: Debatten um Wissenschaftstheorie und Erkenntniskritik’, 2 edn, Verhandlungen mit der Gegenwart, LIT-Verlag, Münster.

Seetharaman, P. B., Ainslie, A. and Chintagunta, P. K. (1999), ‘Investigating household state dependence effects across categories’, *Journal of Marketing Research* **36**(4), 488–500.

Sheth, J. N. and Mittal, B. (2004), *Customer Behavior: A Managerial Perspective*, 2 edn, South Western.

Silverstein, C., Brin, S. and Motwani, R. (1998), ‘Beyond market baskets: Generalizing association rules to dependence rules’, *Data Mining and Knowledge Discovery* **2**(1), 39–68.

Smith, W. (1956), ‘Product differentiation and market segmentation as alternative marketing strategies’, *Journal of Marketing* **21**(1), 3–8.

Solomon, M., Bamossy, G., Askegaard, S. and Hogg, M. K. (2006), *Comsumer Behaviour - A European Perspective*, 3 edn, Pearson Education, Essex, England.

Srikant, R. and Agrawal, R. (1996), Mining sequential patterns: Generalizations and performance improvements, *in* ‘Advances in Database Technology – EDBT ’96’, Vol. 1057 of *Lecture Notes in Computer Science*, Springer, Berlin, pp. 1–17.

Strehl, A. (2002), Relationship-based clustering and cluster ensembles for high-dimensional data mining, PhD thesis, The University of Texas at Austin. Supervisor-Joydeep Ghosh.

Strehl, A. and Ghosh, J. (2000), A scalable approach to balanced, high-dimensional clustering of market-baskets, *in* 'HiPC '00: Proceedings of the 7th International Conference on High Performance Computing', Springer-Verlag, London, UK, pp. 525–536.

Strehl, A. and Ghosh, J. (2003), 'Relationship-based clustering and visualization for high-dimensional data mining', *INFORMS Journal on Computing* pp. 208–230.

Tellis, G. J. (1988), 'The price elasticity of selective demand: A meta-analysis of econometric models of sales', *JMR, Journal of Marketing Research* **24**(4), 331–341.

Thorndike, R. L. (1953), 'Who belongs in the family?', *Psychometrika* **18**(4), 267–276.

Toivonen, H. (1996), Sampling large databases for association rules, *in* 'Proceedings of the 22nd VLDB Conference', Morgan Kaufmann, Bombay, India, pp. 134–145.

Toivonen, H., Klemettinen, M., Ronkainen, P. et al. (1995), Pruning and grouping discovered association rules, *in* 'ECML-95 Workshop on Statistics, Machine Learning, and Knowledge Discovery in Databases', Heraklion, Greece, pp. 47–52.

Turban, E., Aronson, J., Liang, T.-P. and Sharda, R. (2008), *Decision Support and Business Intelligence Systems*, 8 edn, Prentice Hall.

Ungar, L. and Foster, D. (1998), Clustering methods for collaborative filtering, *in* 'Workshop on Recommendation Systems', AAAI Press.

van Heerde, H. J., Leeflang, P. S. H. and Wittink, D. R. (2004), 'Decomposing the sales promotion bump with store data', *Marketing Science* **23**(3), 317–334.

Vitter, J. S. (1987), 'An efficient algorithm for sequential random sampling', *ACM Transactions on Mathematical Software* **13**(1), 58–67.

von Alemann, H. (1984), *Der Forschungsprozess - Einführung in die Praxis der empirischen Sozialforschung*, 2 edn, Teubner, Stuttgart.

Wagstaff, K., Cardie, C., Rogers, S. and Schroedl, S. (2001), Constrained k-means clustering with background knowledge, *in* 'Proceedings of the International Conference on Machine Learning (ICML)', pp. 577–584.

Walters, R. G. and Rinne, H. J. (1986), 'An empirical investigation into the impact of price promotions on retail store performance', *Journal of Retailing* **62**(3), 237–266.

Ward, J. H. (1963), 'Hierarchical grouping to optimize an objective function', *Journal of American Statistical Association* **58**(301), 236–244.

Wedel, M. and Kamakura, W. A. (2000), *Market Segmentation: Conceptual and Methodological Foundations*, 2 edn, Kluwer, Boston.

Weerahandi, S. (1995), 'Anova under unequal error variances', *Biometrics* **51**(2), 589–599.

Weinstein, A. (2004), *Handbook of Market Segmentation*, 3 edn, The Haworth Press.

Xu, L. (1997), 'Bayesian ying-yang machine, clustering and number of clusters', *Pattern Recognition Letters* **18**(11-13), 1167–1178.

Yang, G. (2004), The complexity of mining maximal frequent itemsets and maximal frequent patterns, *in* 'KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, New York, NY, USA, pp. 344–353.

Zaki, M. J. (1999), 'Parallel and distributed association mining: A survey', *IEEE Concurrency* **7**(4), 14–25. Special issue on Parallel Mechanisms for Data Mining.

Zaki, M. J. (2001), 'SPADE: An efficient algorithm for mining frequent sequences', *Machine Learning Journal* **42**(1/2), 31–60. special issue on Unsupervised Learning.

Zaki, M. J., Parthasarathy, S., Li, W. and Ogihara, M. (1997), Evaluation of sampling for data mining of association rules, *in* '7th Intl. Wkshp. Research Issues in Data Engg', pp. 42–50.

Zaki, M. J., Parthasarathy, S., Ogihara, M. and Li, W. (1997), New algorithms for fast discovery of association rules., *in* 'KDD', pp. 283–286.

Zhang, J. and Krishnamurthi, L. (2004), 'Customizing promotions in online stores', *Marketing Science* **23**(4), 561–578.

Zhao, Y., Zhang, C. and Zhang, S. (2006), Efficient frequent itemsets mining by sampling, *in* 'Proceedings of the fourth International Conference on Active Media Technology', AMT06, Brisbane, Australia, pp. 112–117.

Zilahi-Szabó, M. G. (1998), *Informatik*, 3 edn, Oldenbourg, Munic, Vienna.

List of Figures

List of Tables

List of Abbreviations and Symbols

Abbreviations

ANOVA	Analysis of variances
ARM	Association rule mining
ASCII	American standard code for information interchange
BI	Business intelligence
CF	Collaborative filtering
CLV	Customer lifetime value
CPU	Central processing unit
cRand	Corrected Rand index
CRM	Customer relationship management
CSP	Cross selling profit
DIC	Dynamic itemset counting algorithm
EAN	European article number
FI	Frequent itemset
FMPS	Free Mathematical Programming System
HDD	Hard disk drive
HFC	Categories bought highly frequently
I/0	Input/Output
ID	Identification number
IP	Internet protocol
KCCA	*K*-centroid cluster analysis
KDD	Knowledge discovery in databases
LFC	Categories bought less frequently
maxFI	Maximal frequent itemset
MB	Megabyte
MBA	Market basket analysis
MDS	Multidimensional scaling

MIP	Mixed Integer Problem
NGN	Neural gas network
OLAP	Online analytical processing
PAM	Partitioning around medoids
POS	Point of sale
RAM	Random access memory
RFID	Radio frequency identification
RFM	Recency, frequency and monetary value
SKU	Stock keeping unit
SOM	Self-organized feature maps
UPC	Universal product code
Symbols	
α	Active user observed by a recommendation system
$\gamma_{\alpha,j}$	Active user's purchase probability of item j
κ	Normalizing factor
λ	Expectation value or mean
ω	Similarity or distance measure used in the CF approach
Φ	Number of categories recommended in the specific segment
Ψ_{Tan}	Tanimoto similarity measure
Θ	Probability value
$\{expression\}$	Itemset/item
a,b	Coordinates defining the cells of a two-dimensional SOM/cells of a table of contingency
A,B,C	Frequent itemset/frequent items
C_K	Set of K partitions
c_k	Single partition of transactions
D	Distance
d	Distance measure
db_k	Davies-Bouldin index for the cluster solution with k predefined segments
G_m	Customer's m buying sequence or purchase history
H_l	Solution set of frequent itemsets with the length l derived from the counting step l
h_l	Collection of frequent itemsets with length l of the solution set H_l
j,i	Item/category/product with $j,i = 1...J$
K	Number of clusters

k	Cluster index
l	Number of items in an itemset or the index of a counting pass of an algorithm
m	Customer ID/index of a buying sequence G_m of customer m
n	Index of the transaction x from $n = 1...N$
o	Threshold, limiting value
$P(.)$	Probability of occurrence
P_K	Set of cluster centroids or prototypical market baskets
p_k	Cluster centroid or prototypic market basket
$p_{a,b}$	Reference vector of a SOM
Q_j	Binary decision variables regarding category j
$q_{a,b}$	Unit of a SOM with the two-dimensional a,b coordinates
r	Minkowski metric
R_n	Random variable $R \in [0,1]$ of an itemset A in the transaction with the index n
S_A	Binary decision variables regarding itemset A
s_k^m	Majority voting measure
$U(j)$	Utility/Profit of category j
$u_{n,j}$	Binary value of the cell at row n and column j in X_N
$V(A)$	Profit margin of the frequent itemset A
$v(x_n)$	Profit margin of a transaction x_n
w_j	Weight or weighting factor concerning category j
x	Transaction(-vector)/market basket/shopping basket
x^*	Representative transaction of a compressed buying history
X_N	Data table of all transaction vectors x_n with $n = 1...N$
x_n^{medoid}	Medoid
y_m, ρ_m	Representative transaction of the buying history of customer m
z	Number of cells in a distance matrix subject to the number of items J