Die approbierte Originalversion dieser Diplom-/Masterarbeit ist an der Hauptbibliothek der Technischen Universität Wien aufgestellt (http://www.ub.tuwien.ac.at).

The approved original version of this diploma or master thesis is available at the main library of the Vienna University of Technology (http://www.ub.tuwien.ac.at/englweb/).

TECHNISCHE UNIVERSITÄT WIEN

VIENNA UNIVERSITY OF TECHNOLOGY

D I P L O M A R B E I T

Neue Methoden zur Marktsegmentierung durch Clustern von gruppierten Warenkorbdaten

ausgeführt am Institut für
Statistik und Wahrscheinlichkeitstheorie
der Technischen Universität Wien

unter Anleitung von
Privatdozent Dipl.-Ing. Dr.techn. Friedrich Leisch

durch
Kerstin Balka
Salitergasse 93
2380 Perchtoldsdorf

Datum

Unterschrift

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Danksagung

An dieser Stelle möchte ich mich herzlich bei Herrn Dipl.-Ing. Dr.techn. Friedrich Leisch für die Vergabe des Themas und die Betreuung meiner Diplomarbeit bedanken. Durch seine Ideen wurde die vorliegende Arbeit in dieser Form erst möglich.

Weiters danke ich Herrn a.o.Univ.Prof. Thomas Reutterer dafür, dass er mir die Echtdaten zur Verfügung stellte, und für seine Anregungen zur Gestaltung der Simulation.

Ausserdem möchte ich meinen Eltern danken, die mir das Studium und damit diese Arbeit ermöglichten und mich stets unterstützten. Insbesondere bedanke ich mich bei meiner Mutter für das Korrekturlesen der Arbeit.

Kapitel 1

Einführung

Umfangreiche Sammlungen von binären Daten tauchen in vielen Anwendungsbereichen auf, wie zum Beispiel auf medizinischen Gebieten, in Webanwendungen, im Telekommunikationsbereich und bei der Analyse von Warenkörben. Die Daten sind oft hochdimensional und die darin enthaltenen Objekte geben Auskunft über zahlreiche Variablen. Da diese Variablen selten unabhängig sind, ist es natürlich anzunehmen, dass sich die Objekte in Gruppen gliedern, das heißt in Ansammlungen, die in bestimmter Art und Weise einander ähnlich sind. Zum Beispiel könnten bei Warenkörben von Supermärkten diese Gruppen zusammenhängende Produktgruppen wie Getränke, Gemüse etc. sein oder auch Gruppen von Kunden mit ähnlichem Einkaufsverhalten.

Diese Gruppen, in denen sich die Objekte möglichst ähnlich sein sollen, in den Daten zu finden ist je nach Problemstellung oft keineswegs einfach und genau das ist die Aufgabe der Clusteranalyse. Die vorliegende Arbeit beschäftigt sich dabei mit Anwendungen im Bereich des Marketings und dem Ziel der Marktsegmentierung. Das Problem die Konsumenten in Gruppen zu klassifizieren mag einfach erscheinen und beschäftigt die Marktforscher dennoch seit Jahrzehnten (siehe z.B. Wind (1978) für eine ausführliche Schilderung des Problems der Marktsegmentierung wie es sich vor fast 30 Jahren stellte oder Baumgartner und Homburg (1996) für einen Überblick über die Anwendungen von 1977 bis 1994).

Zur konkreten Lösung des Clusterproblems gibt es eine Vielzahl an unterschiedlichen Algorithmen und Vorgehensweisen für die verschiedensten Anwendungsbereiche. In dieser Arbeit werden traditionelle Clusteralgorithmen wie zum Beispiel K-Means (MacQueen, 1967; Hartigan und Wong, 1979) und Algorithmen aus dem Bereich des maschinellen Lernens wie zum Beispiel das

neuronale Gas (Martinetz et al., 1993) betrachtet. Außerdem werden binäre Mischmodelle (siehe zB. Wedel und Kamakura, 1998) und Assoziationsregeln (siehe zB. Agrawal et al., 1993) vorgestellt.

Besonderes Augenmerk wird auf die Segmentierung von gruppierten Daten gelegt. Man denke dabei beispielsweise an eine Supermarktkette, die durch ein Kundenbindungsprogramm die einzelnen Einkäufe bestimmten Kunden zuordnen und so Objekte des Warenkorbes verbinden kann. Die Clusteranalyse soll dann fähig sein auf die zu Gruppen verknüpften Objekte Rücksicht zu nehmen und soll nicht die einzelnen Einkäufe, sondern die Kunden in Cluster teilen um ein zielgerichtetes Marketing zu ermöglichen.

Am Beginn dieser Arbeit (2. Kapitel) werden zunächst die Grundlagen der Clusteranalyse erläutert und unterschiedliche Vorgehensweisen zur Marktsegmentierung vorgestellt. Das 3. Kapitel beschäftigt sich mit der Segmentierung von gruppierten Daten, wozu am Beginn des 4. Kapitels ein Beispiel gebracht wird. Im Folgenden beschäftigt sich das 4. Kapitel mit der Simulation eines synthetischen Warenkorbes, der im 5. Kapitel analysiert wird. Das 6. Kapitel beschäftigt sich mit der Analyse und Segmentierung eines Datensatzes mit binären Umfragedaten zum Einkaufsverhalten. Die Barplots zu den verschiedenen Clusterlösungen befinden sich im Anhang.

Sämtliche Berechnungen und Analysen, die im Zuge dieser Arbeit benötigt wurden, wurden mit R Development Core Team (2005) durchgeführt.

Kapitel 2

Grundlagen zur Clusteranalyse

Die Clusteranalyse ist eine statistische Methode zur Klassifikation und Partitionierung von Daten bzw. zur Auffindung der Struktur in einem Datensatz. Ihren Ausgangspunkt bildet eine Menge von Objekten, zum Beispiel Datensätze von Messwerten, Bildpunkte, Wörter eines Dokumentes oder Produkte eines Warenkorbes. Diese Objekte sollen aufgrund bestehender Ähnlichkeitsbeziehungen in Gruppen (bzw. Cluster oder Klassen) eingeteilt werden, sodass die Objekte eines Clusters eine weitgehend verwandte Eigenschaftsstruktur aufweisen, d. h. sich möglichst ähnlich sind. Vor allem wenn das Hauptaugenmerk die Partitionierung der Daten ist, sollen zwischen den Gruppen (so gut wie) keine Ähnlichkeiten bestehen und das Ziel der Clusteranalyse ist homogene Teilmengen in einer heterogenen Gesamtheit von Objekten zu identifizieren.

Ihren Ablauf kann man in zwei grundlegende Schritte unterteilen:

1. Schritt: Wahl des Distanzmaßes
2. Schritt: Wahl des Clusteralgorithmus

2.1 Distanz- und Ähnlichkeitsfunktionen

Distanz- und Ähnlichkeitsfunktionen beschreiben den Grad der Übereinstimmung von Vektoren, sie bilden das Kernstück der klassischen clusteranalytischen Verfahren. Ein Distanzmaß ist definiert als

$$d(\mathbf{x}, \mathbf{y}) : \mathcal{X} \times \mathcal{X} \rightarrow \mathbf{R}^{+}$$

wobei $\mathcal{X}$ einen Raum deterministischer oder zufälliger Variablen bezeichnet.

Zur Partitionierung von Daten werden die Abstände zwischen den zu clusternden Objekten und den Clusterzentren benötigt um die Zuordnung zu bestimmen. Daher sollen im folgenden Distanzmaße der Form $d(\mathbf{x}, \mathbf{c}) : \mathcal{X} \times \mathcal{C} \rightarrow \mathbf{R}^+$ betrachtet werden, wobei $\mathcal{C}$ die Menge oder den Raum der zulässigen Zentren sowie $\mathbf{x}$ und $\mathbf{c}$ jeweils M-dimensionale Vektoren bezeichnen.

Bei metrischen Variablen wird zumeist die *euklidische Distanz* zur Abstandsberechnung herangezogen:

$$d(\mathbf{x}, \mathbf{c}) = ||\mathbf{x} - \mathbf{c}|| = \sqrt{\sum_{i=1}^{M} (x_i - c_i)^2}$$

Die Clusterzentren sind dann die Mittelwerte jedes Clusters. Manchmal ist es jedoch zweckmäßig den Median als Clusterzentrum und damit die *Absolutdistanz* (auch Manhattan Distanz) zu wählen:

$$d(\mathbf{x}, \mathbf{c}) = |\mathbf{x} - \mathbf{c}| = \sum_{i=1}^{M} |x_i - c_i|$$

Die Wahl des Distanzmaßes kann man eventuell anhand von Kenntnissen über die Daten treffen; sie bestimmt die Form der Cluster. Zur Partitionierung nominaler Daten verwenden zum Beispiel Chaturvedi et al. (2001) das Distanzmaß

$$d(\mathbf{x}, \mathbf{c}) = \frac{\#\{i | x_i \neq c_i\}}{M}$$

das zählt, in wie vielen Dimensionen die Beobachtung $\mathbf{x}$ und das Zentrum $\mathbf{c}$ nicht denselben Wert haben.

Distanz- und Ähnlichkeitsfunktionen bei binären Vektoren

Binäre Variablen sind Variablen, die nur zwei Ausprägungen $x = 1$ oder $x = 0$ annehmen können. Zum Beispiel kodiert man so männlich / weiblich oder in Warenkörben gekauft / nicht gekauft. Oft wird auch für binäre Vektoren die euklidische Distanz zur Abstandsberechnung herangezogen, zur Verbesserung der Vergleichsmöglichkeiten wurden jedoch spezielle Distanzmaße für binäre Vektoren entwickelt.

Man betrachte dazu zwei binäre Vektoren $\mathbf{x}$ und $\mathbf{y}$ und definiere die 2×2 Kontingenztabelle

		x		
		1	0	
y	1	α	β	$\alpha + \beta$
	0	γ	δ	$\gamma + \delta$
		$\alpha + \gamma$	$\beta + \delta$	M

Dabei zählt α die Anzahl der Dimensionen, in denen $\mathbf{x}$ und $\mathbf{y}$ gleich 1 sind, β die Anzahl der Dimensionen, in denen $\mathbf{x}$ gleich 0 und $\mathbf{y}$ gleich 1 ist, usw. Daraus haben sich eine Vielzahl verschiedener Distanz- und Ähnlichkeitsfunktionen entwickelt, zum Beispiel die *Hamming Distanz*, die durch

$$d(\mathbf{x}, \mathbf{y}) = \beta + \gamma$$

definiert ist und dividiert durch die Gesamtzahl $M = \alpha + \beta + \gamma + \delta$ den Prozentsatz ungleicher Einträge in $\mathbf{x}$ und $\mathbf{y}$ angibt. Behandelt man binäre Variablen wie metrische Variablen und berechnet die Absolutdistanz, so erhält man bis auf einen konstanten Faktor die Hamming Distanz; ebenso entspricht sie bis auf die Quadratwurzel der euklidischen Distanz.

In Anwendungen, zum Beispiel im Marketingbereich, ist es aber oft nicht ideal, wenn das Distanzmaß die Einträge eins und null gleich behandelt, da zwei Kunden, die ein Produkt beide gekauft haben, vermutlich deutlich mehr gemein haben, als zwei Kunden, die dieses Produkt beide nicht gekauft haben. Die Matrix eines Warenkorbes enthält typischerweise sehr viele Nullen und sehr wenige Einser. Die alleinige Betrachtung der Übereinstimmungen, also der Hamming-Distanz, würde zu dem Schluss führen, dass die meisten Einkäufe sehr ähnlich sind.

Zum Vergleich asymmetrischer binärer Variablen werden also andere Distanzmaße benötigt. Der bekannteste Koeffizient dazu ist der *Jaccard Koeffizient* (siehe zB. Kaufman und Rousseeuw, 1990)

$$d(\mathbf{x}, \mathbf{y}) = \frac{\beta + \gamma}{\alpha + \beta + \gamma}$$

Er bewertet gemeinsame Einser (in beiden Warenkörben enthaltene Produkte) stärker als gemeinsame Nullen, indem er die Dimensionen, in denen beide Vektoren den Wert Null annehmen, nicht in die Berechnung mit einbezieht.

2.2 Hierarchische Clusteralgorithmen

Zur Lösung des Clusterproblems wurde eine Vielzahl an Algorithmen entwickelt, die sich oft nur in kleinen Details unterscheiden. Man unterteilt sie zunächst in hierarchische und partitionierende Verfahren, wobei man bei hierarchischen Verfahren zwischen agglomerativen und divisiven Verfahren unterscheidet:

- **Agglomerative Verfahren** starten mit den kleinstmöglichen Clustern und versuchen durch sukzessives Zusammenlegen zweier Cluster immer gröbere Partitionen zu erzeugen.
- **Divisive Verfahren** beginnen umgekehrt mit der gröbsten Partition und erzeugen die Cluster durch sukzessive Teilungen.

2.3 Partitionierende Clusteralgorithmen

Den Ausgangspunkt der Clusteranalyse stellt eine Menge von N verschiedenen Objekten, von denen M verschiedene Variablen bekannt sind, dar. Ziel des Problems ist es Clusterzentren C_K zum Datensatz $X_N = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$, $\mathbf{x}_n \in \mathcal{X}$ für vorgegebenes K so zu finden, dass die durchschnittliche Distanz jedes Punktes zum nächstgelegenen Zentrum minimal wird:

$$D(X_N, C_K) = \frac{1}{N} \sum_{n=1}^{N} d(\mathbf{x}_n, c(\mathbf{x}_n)) \rightarrow \min_{C_K}$$

Partitionierende Verfahren gehen von einer fixen Anzahl der Cluster aus, die im Vorhinein geeignet gewählt werden muss. Das Verfahren beginnt mit einer gewählten oder zufälligen Startpartition und versucht dann mit Hilfe eines Austauschalgorithmus die Objekte zwischen den Clustern solange umzuordnen, bis die gewählte Zielfunktion ein Optimum erreicht hat.

2.3.1 Verallgemeinertes K-Means

Der K-Means Clusteralgorithmus ist ein iteratives Verfahren zur Optimierung der lokalen Partitionierung. Ziel des Algorithmus ist es N Datenpunkte in K Cluster so einzuteilen, dass die Abstände der Punkte zu den Clusterzentren minimiert werden. MacQueen (1967) formuliert den Algorithmus für

beliebige Distanzen und berechnet nach jedem K-Means Zyklus neue Clusterzentren. Zur Segmentierung wird eine K-Partitionierung mit lokal minimalen Distanzen gesucht, indem Punkte von einem Cluster in einen anderen verschoben werden.

1. Wähle eine initiale Partition mit K Zentren C_K.
2. Ordne jeden Punkt $\mathbf{x}_n \in X_N$ dem nächsten Zentrum zu.
3. Berechne die neuen Zentren, nachdem alle Punkte zugeordnet wurden:

$$\mathbf{c}_k := \arg\min_{\mathbf{c}\in C} \sum_{n:c(\mathbf{x}_n)=\mathbf{c}_k} d(\mathbf{x}_n, \mathbf{c}), \qquad k = 1, ..., K$$

4. Wiederholung ab Punkt 2, bis sich die Zuordnung nicht mehr ändert und ein lokales Minimum gefunden wurde.

2.3.2 K-Means nach Hartigan und Wong (1979)

$D(X_N, C_K)$ bezeichnet hier die euklidische Distanz zwischen dem Punkt X und Cluster C. Im Gegensatz zur Formulierung von MacQueen wird in jedem Iterationsschritt nur ein Punkt dem nächstgelegenen Zentrum neu zugeordnet.

Der Algorithmus arbeitet wie folgt:

1. Initialisierung: K Clusterzentren werden vorgegeben oder zufällig gewählt.
2. Zuordnung: Finde für jeden Punkt $\mathbf{x}_n \in X_N$, $n = 1, ..., N$ das nächstgelegene Zentrum C_j und ordne $\mathbf{x}_n$ diesem zu.
3. Wähle ein $\mathbf{x}_n \in C_i$ $(n = 1, ..., N; i = 1, ..., K)$, das neu zugeordnet wird.
   - Berechne $D(\mathbf{x}_n, C_j)$ für alle $j = 1, 2..., K$.
   - Bestimme Zentrum C_j, sodass $D(\mathbf{x}_n, C_j)$ minimal ist.
   - Ordne $\mathbf{x}_n$ dem Cluster C_j zu.
   - Berechne die neuen Zentren für die Cluster C_i und C_j.
4. Wiederholung von Punkt 3, bis sich die Zuordnung nicht mehr ändert und ein lokales Minimum gefunden wurde.

Die Berechnung der beiden neuen Zentren in Punkt 3 ist, da hier von euklidischer Distanz ausgegangen wird, leicht durchzuführen. Falls sich die Zuordnung von $\mathbf{x}_n$ verändert hat, wird einem Mittelwert ein Punkt hinzugefügt und vom anderen Mittelwert ein Punkt entfernt.

Konvergenz: Die Zielfunktion

$$D(X_N, C_K) = \frac{1}{N} \sum_{n=1}^{N} d(\mathbf{x}_n, c(\mathbf{x}_n)) \rightarrow \min_{C_K}$$

ist für beide vorgestellten Varianten des K-Means Algorithmus eine monoton nicht steigende Funktion. Da es nur endlich viele mögliche Partitionen für X_N gibt, ist gesichert, dass der Algorithmus nach endlich vielen Iterationen zu einem lokalen Optimum konvergiert.

Der **Nachteil** dieser und anderer partitionierender Verfahren ist, dass verschiedene Anfangspartitionen häufig zu unterschiedlichen lokalen Optima führen.

2.3.3 Hard Competitive Learning

Im Gegensatz zu den bisher vorgestellten Verfahren der klassischen Statistik wurde das Competitive Learning im Bereich des maschinellen Lernens entwickelt. Das Hard Competitive Learning ist davon der einfachste Clusteralgorithmus, wobei die Clusterzentren $C = \{c_1, ..., c_N\}$ als Ausgabeeinheit eines neuralen Netzes betrachtet werden. In jedem Iterationsschritt wird eine gewinnende Einheit bestimmt und entlang des Gradientenvektors in Richtung des gegebenen Musters verschoben. Das HCL oder Winner-take-all umfaßt alle Methoden, die jeweils nur eine Einheit - die gewinnende Einheit oder den „Gewinner“ - adaptieren.

Das Verfahren läuft dazu folgendermaßen (vgl. Dolnicar et al., 1998):

1. Initialisierung der Clusterzentren $C = \{c_1, ..., c_N\}$ mit Referenzvektoren w_{c_i}.
2. Wahl eines Vektors $x_i \in \mathcal{X}$.
3. Bestimmung des Gewinners $s(x_i) = \arg\min_{c \in C} ||x_i - w_c||$.
4. Adaptierung des Referenzvektor des Gewinners unter Benutzung der (abnehmenden) Lernrate ϵ.
5. Wiederholung ab Punkt 2 bis eine gewählte Lernrate erreicht ist.

Die Lernrate ϵ bestimmt dabei das Ausmaß, in dem der Gewinner adaptiert wird.

Im K-Means Ansatz wird die Lernrate über die Zeit kontinuierlich verringert, indem sie für jedes Objekt eine harmonische Reihe über i annimmt, wobei i die bereits stattgefundenen Adaptierungen bezeichnet: $\epsilon(i) = 1/i$. Der Nachteil von K-Means besteht darin, dass das Verfahren wegen der divergierenden harmonischen Reihe der Lernrate zumeist nicht gegen das globale Optimum konvergiert.

In diesem Verfahren soll die Lernrate

$$\epsilon(i) = \epsilon_{ini}(\frac{\epsilon_{fin}}{\epsilon_{ini}})^{\frac{i}{i_{\max}}}$$

zum Einsatz kommen, wobei ϵ_{ini} und ϵ_{fin} die initiale und die finale Lernrate bezeichnen, $i_{\max}$ ist die Gesamtzahl der Adaptionsschritte des Verfahrens. Diese Lernrate ist unempfindlicher gegenüber einer schlechten Initialisierung der Referenzvektoren und so findet der Algorithmus zu Beginn des Lernens eher aus einem lokalen Minimum heraus als es bei K-Means der Fall ist.

2.3.4 Neuronales Gas (Soft Competitive Learning)

Das Neuronale Gas von Martinetz et al. (1993) unterscheidet sich von Hard Competitive Learning dahingehend, dass nicht nur der Gewinner adaptiert wird, sondern auch einige bis alle weiteren Units. Dazu werden die Referenzvektoren nach jeder erzeugten Eingabe gereiht und nacheinander adaptiert, wobei die Adaptionsrate und die Anzahl der zu adaptierenden Neuronen mit der Zeit abnimmt. So wird die Wahrscheinlichkeit nur ein lokales Optimum zu erreichen erneut verringert.

2.4 Binäre Mischmodelle

Einfache binäre Mischmodelle, auch bekannt durch die *Latent Class Analyse*, bieten eine modell-basierte Alternative zu traditionellen Clusterverfahren wie beispielsweise K-Means. Das heißt, dass vorausgesetzt wird, dass die Daten bzw. das Verhalten der Kunden durch ein passendes statistisches Modell beschrieben werden können und dass davon Abstand genommen wird keine a priori Annahmen über die Struktur der Daten zu haben. Dadurch sollen Mischmodelle der historisch bekannteren Clusteranalyse überlegen sein (Wedel und Kamakura, 1998). Einen direkten Vergleich zwischen einem Latent class Modell und K-Means zeigen zum Beispiel Magidson und Vermunt (2002).

Es wird angenommen, dass die Daten von einer Mischung zugrundeliegender Verteilungsfunktionen erzeugt werden.

$$\mathbf{P}(\xi = x \mid X) = \prod_{j=1}^{M} \left(\pi_j^{x_j} \cdot (1-\pi_j)^{(1-x_j)}\right)$$

X ist die Datenmatrix bestehend aus den Zeilenvektoren $x_{i\cdot}$, wobei jede Zeile einem Einkauf entspricht. Dabei nimmt man an, dass die Komponenten von $x_{i\cdot}$, gegeben eine Mischkomponente, unabhängig und binomialverteilt sind.

Die posteriori Wahrscheinlichkeit für ein Objekt $x_{i\cdot}$ im Cluster k zu sein ist durch

$$p_{ik} = \frac{w_k \cdot \prod_{j=1}^{M}(\pi_{kj}^{x_{ij}} \cdot (1-\pi_{kj})^{(1-x_{ij})})}{\sum_{r=1}^{K} w_r \cdot \prod_{j=1}^{M}(\pi_{rj}^{x_{ij}} \cdot (1-\pi_{rj})^{(1-x_{ij})})}$$

definiert.

$\hat{w}_k$ - die a priori Wahrscheinlichkeit für Klasse k - entspricht dabei dem Schätzer für die Wahrscheinlichkeit w_k eines Erfolges der Komponente k und berechnet sich durch

$$\hat{w}_k = \frac{1}{N}\sum_{i=1}^{N} p_{ik} \qquad \text{für} \qquad k = 1, ..., K$$

Die Likelihoodfunktion für den Parameter $\Phi = (\Pi_1, ..., \Pi_K; w_1, ..., w_K)$ mit $\Pi_j = (\pi_{k1}, ...\pi_{kM})$ bezüglich der Datenmenge X hat die Form

$$L(\Phi \mid X) = \prod_{i=1}^{N}\left(\sum_{k=1}^{K} w_k \prod_{j=1}^{M}(\pi_{kj}^{x_{ij}} \cdot (1-\pi_{kj})^{(1-x_{ij})})\right)$$

Ziel der Maximum Likelihood Schätzung ist es aus der Datenmenge die Parameter zu schätzen, dh. man versucht durch Maximierung der Likelihoodfunktion Werte für Φ so zu finden, dass die Auftrittswahrscheinlichkeit von X möglichst groß ist. Gelöst wird dieses Problem mit Hilfe des EM Algorithmus, einem iterativen Verfahren zur Maximum Likelihood Schätzung.

2.5 Assoziationsregeln

Die Assoziationsanalyse ist eine weitere Möglichkeit Zusammenhänge in Daten, zum Beispiel in Warenkorbdaten, zu finden. Eine Assoziationsregel gibt an, welche Produkte gemeinsam gekauft werden, und wird durch $A \Rightarrow B$ angegeben.

Dazu definiert man:

- $\text{support}(A \Rightarrow B) = P(A \cup B)$
- $\text{confidence}(A \Rightarrow B) = P(B|A) = \frac{P(A \cup B)}{P(A)}$

Eine Assoziationsregel wird als *stark* bezeichnet, wenn sie einen Mindestsupport und eine Mindestconfidence erfüllt, die Assoziationsanalyse bezeichnet die Suche nach starken Regeln. Ein mögliches Vorgehen dazu ist der Apriori Algorithmus. Er sucht zunächst *häufige Itemsets*, das heißt Teilmengen, die den minimalen Support erfüllen, und bildet dann aus ihnen starke Regeln. Dadurch werden Zusammenhänge in den Daten gezeigt, die vor allem bei Warenkorbdaten eine Alternative zu den Ergebnissen der Clusteranalyse darstellen bzw. oft auch mit ihnen vergleichbar sind.

In der Literatur werden zahlreiche Varianten dieses Algorithmus präsentiert (zB: Agrawal et al. (1993); Agrawal und Srikant (1994); Park et al. (1995); etc.), die Anwendung auf hochdimensionale binäre Daten wird jedoch wenig betrachtet. Ein Verfahren, das für binäre Daten Assoziationsregeln mit Mischmodellen verbindet, wird zum Beispiel von Seppaenen et al. (2003) beschrieben. Hahsler et al. (2005) präsentieren das R-Paket `arules`, das es ermöglicht häufige Itemsets und Assoziationsregeln mit R zu analysieren.

2.6 Clustern mit Hintergrundinformation

Im Gegensatz zu den beschriebenen Mischmodellen werden Clusteralgorithmen üblicherweise ohne a priori Informationen über mögliche Clusterzuordnungen oder Gruppierungen auf die Daten angewandt. Nur über die Form der Cluster werden durch die Wahl des Distanzmaßes Annahmen getroffen. In der Realität ist es jedoch oft so, dass Hintergrundinformation in Form von Labels, bekannte Gruppierungen oder Assoziationen in den Daten, etc. vorhanden ist, die von den traditionellen Algorithmen nicht genutzt wird. Beispielsweise könnte ein Supermarkt, der die durch Scannerkassen leicht verfügbaren Daten der Einkäufe speichert, zusätzlich durch ein Kundenbindungsprogramm fähig sein, die Einkäufe den Kunden zuzuordnen. Somit gibt es bereits vor Beginn der Segmentierung Information darüber, welche Objekte im selben Cluster sein müssen.

Diese a priori Informationen werden in Form von Bedingungen verarbeitet:

- **Must-link** constraints zwischen Ojekten, die demselben Cluster angehören müssen
- **Cannot-link** constraints zwischen Objekten, die nicht im selben Cluster liegen dürfen

Wagstaff et al. (2001) präsentieren dazu eine Variante des K-Means Algorithmus, der bei jeder Neuzuordnung darauf achtet die Bedingungen nicht zu verletzen. Alternativ zeigen zum Beispiel Law et al. (2004) eine Vorgangsweise, bei der sie von Mischmodellen ausgehen und neben den „fixen“ Bedingungen sogenannte „soft constraints“ zulassen.

2.7 Indizes zum Vergleich zweier Partitionen

Für den Vergleich zweier Clusterzuteilungen berechnet man geeignete Maßzahlen, die den Grad deren Übereinstimmung widerspiegeln, wie z.B. der *Diagonal Index*, der *Kappa Index*, der *Rand Index* oder der *Corrected Rand Index*. Dazu betrachtet man zuerst die Kontingenztafel der beiden Partitionen $T = [t_{kr}]$, für $k, r = 1, ..., s$. t_{kr} beschreibt dabei die Anzahl der Datenpunkte, die in der ersten Partition in Cluster k und in der zweiten Partition in Cluster r sind.

		Partition 2			
		1		s	$\sum$
Partition 1	1	t_{11}	...	t_{1s}	$t_{1.}$
	$\vdots$	$\vdots$	$\ddots$	$\vdots$	
	s	t_{s1}	...	t_{ss}	$t_{s.}$
	$\sum$	$t_{.1}$	...	$t_{.s}$	n

Unter der Annahme, dass die Partitionen dieselben Label verwenden, stimmen sie überein, falls nur in der Hauptdiagonale von Null verschiedene Einträge stehen. Je größer die Einträge außerhalb der Diagonale sind, umso verschiedener sind die Klassifikationen.

Falls die Partitionen nicht dieselben Label verwenden oder falls gleiche Label unterschiedliche Bedeutungen haben können, ist es ein wenig schwieriger Auskunft über den Grad der Übereinstimmung zu geben. Im ersten Fall ist es zum Beispiel hilfreich, die Spalten oder die Zeilen von T so zu vertauschen, dass die Diagonale $\sum_{k=1}^{s} t_{kk}$ maximal wird. Es gibt jedoch nicht immer eine einfache Lösung für das Problem der korrekten Zuordnung.

In diesem Fall sind der *Diagonal Index* und der *Kappa Index* wenig geeignet. Der *Rand Index* und der *Corrected Rand Index* sind invariant gegenüber Permutationen der Zeilen oder Spalten von T und liefern somit auch in diesem Fall brauchbare Ergebnisse.

Diagonal Index

Der Diagonal Index gibt den Anteil der Objekte, die in beiden Partitionen demselben Cluster angehören, an und ist definiert durch

$$\text{diag}(T) = \frac{\sum_{k=1}^{s} t_{kk}}{n}$$

Kappa Index

Der Kappa Index

$$\text{kappa}(T) = \frac{\frac{1}{n}\sum_{k=1}^{s} t_{kk} - \frac{1}{n}\sum_{k=1}^{s} t_{k.}t_{.k}}{1 - \frac{1}{n}\sum_{k=1}^{s} t_{k.}t_{.k}} \in [-1; 1]$$

entspricht dem Diagonal Index korrigiert um die Übereinstimmung durch Zufall. Diese Übereinstimmung muss im Fall unterschiedlich großer Cluster berücksichtigt werden, da zwei größere Cluster bedingt durch ihre Größe zufällige Übereinstimmungen haben können.

Rand Index

Zur Berechnung des Rand Index vergleicht man die Zuordnung aller Paare von Objekten. A bezeichnet die Anzahl aller Paare von Objekten, die entweder bei beiden Partitionen demselben Cluster oder die bei beiden Partitionen jeweils unterschiedlichen Clustern angehören. D zählt die Paare von Objekten, die bei einer Partition demselben Cluster, bei der anderen Partition aber jeweils unterschiedlichen Clustern angehören. Es gilt: $A+D = \binom{n}{2}$. Der Rand Index ist dann gegeben durch

$$\text{rand}(T) = \frac{A}{\binom{n}{2}}$$

Corrected Rand Index

Der Corrected Rand Index ist der um die Übereinstimmung durch Zufall korrigierte Rand Index; er kann wie der Kappa Index Werte im Intervall [-1,1] annehmen.

Die Berechnung erfolgt zum Beispiel durch

$$\text{crand}(T) = \frac{\sum_{k,r=1}^{s}\binom{t_{kr}}{2} - \frac{1}{\binom{n}{2}}\sum_{k=1}^{s}\binom{t_{k.}}{2}\sum_{r=1}^{s}\binom{t_{.r}}{2}}{\frac{1}{2}[\sum_{k=1}^{s}\binom{t_{k.}}{2} + \sum_{r=1}^{s}\binom{t_{.r}}{2}] - \frac{1}{\binom{n}{2}}\sum_{k=1}^{s}\binom{t_{k.}}{2}\sum_{r=1}^{s}\binom{t_{.r}}{2}}$$

Für den Kappa Index und den Corrected Rand Index gilt, dass sie den Wert 1 annehmen, falls die beiden Partitionen exakt übereinstimmen; bei einem Wert von -1 sind diese so unterschiedlich wie möglich. Sind die Partitionen unabhängig, werden der Kappa Index und der Corrected Rand Index einen Wert in der Gegend von 0 annehmen.

Kapitel 3

Partitionierung von gruppierten Daten

Im folgenden Kapitel soll näher darauf eingegangen werden, wie Hintergrundinformation in Form von bekannten Gruppen in den Daten verarbeitet werden kann. Diese Gruppen bilden die im voranstehenden Abschnitt beschriebenen Must-link constraints; Objekte, die derselben Gruppe angehören, sollen stets demselben Cluster zugeordnet werden.

Man betrachte eine Datenmatrix X bestehend aus N Objekten, kodiert als binäre Vektoren. Zum Beispiel könnte diese Datenmatrix einen Warenkorb bestehend aus N Einkäufen bilden. Jeder dieser Zeilenvektoren gibt für M Variablen - zum Beispiel zur Wahl stehende Produkte oder Produktgruppen - an, ob sie vorhanden oder nicht vorhanden sind, also ob ein Produkt dieser Gruppe für diesen Einkauf gewählt wurde. Es wird angenommen, dass mehrere Objekte zu einer Gruppe gehören und dass sich mehrere Gruppen zu einem Typ (Cluster) zusammenfassen lassen. Diese Gruppen von Einkäufen könnten zum Beispiel jeweils Personen, die an einem Kundenbindungsprogramm teilnehmen, zuzuordnen sein. Mehrere Personen mit ähnlichen Einkaufsgewohnheiten bilden dann einen Einkaufstyp. Ziel der Clusteranalyse ist es nun diese Typen zu identifizieren und die Objekte bzw. im konkreten Fall die ihnen übergeordneten Gruppen den Clustern möglichst korrekt zuzuordnen.

Im Folgenden werden dazu vier verschiedene Vorgehensweisen betrachtet. Als Ausgangspunkt dient jeweils der K-Means Algorithmus, wobei durch Modifikationen unterschiedliche Behandlungen der Gruppen ermöglicht werden.

3.1 A posteriori Gruppierung

Will man die Information, welche Objekte derselben Gruppe angehören, zunächst beiseite lassen, kann man K-Means direkt auf die Datenmatrix X anwenden und die bereits bekannte Zielfunktion

$$D(X_N, C_K) = \frac{1}{N} \sum_{n=1}^{N} d(\mathbf{x}_n, c(\mathbf{x}_n)) \to \min_{C_K}$$

betrachten. Nach der so erhaltenen Segmentierung kann man die einzelnen Gruppen mit Hilfe einer Mehrheitsabstimmung den ermittelten Zentren zuordnen:

1. Wähle eine initiale Partition mit K Zentren C_K.
2. Ordne jeden Punkt $\mathbf{x}_n \in X_N$ dem nächsten Zentrum zu.
3. Berechne die neuen Zentren, nachdem alle Punkte zugeordnet wurden:

$$\mathbf{c}_k := \arg\min_{\mathbf{c} \in C} \sum_{n:c(\mathbf{x}_n)=\mathbf{c}_k} d(\mathbf{x}_n, \mathbf{c}), \qquad k = 1, ..., K$$

4. Wiederholung der Punkte 2 und 3, bis sich die Zuordnung nicht mehr ändert und ein lokales Minimum gefunden wurde.
5. Ordne jede Gruppe von Objekten durch eine Mehrheitsabstimmung über die Zuteilung der einzelnen Objekte dieser Gruppe einem gemeinsamen Zentrum zu.

Dieses Vorgehen macht unter Umständen dann Sinn, wenn die Partitionierung ohne Beachtung der Hintergrundinformation bereits vorhanden ist. Für einen sinnvollen Vergleich mit Segmentierungen mit Bedingungen sollten Objekte, die derselben Gruppe angehören, auch im selben Cluster sein.

Zur alleinigen Anwendung kann die Segmentierung mit a posteriori Gruppierung jedoch nicht empfohlen werden, weil durch die Zuteilung der Gruppen zu den Zentren nach der Beendung des Clusterverfahrens die Partitionierung verändert wird, wodurch am Ende keine (lokal) optimale Partitionierung im Sinne der Minimierung der Zielfunktion mehr besteht.

3.2 A priori Gruppierung

Eine Möglichkeit, die Hintergrundinformation bereits vor der Segmentierung zu verarbeiten, ist nicht direkt die Matrix X zu clustern, sondern zuerst für jede Gruppe einen „Durchschnitt“ zu ermitteln. Hierfür kann man zum Beispiel bei Warenkorbdaten entweder die echten Mittelwerte pro Gruppe und Variable binarisieren oder man wählt Vektoren, in denen eine Eins bei jeder Variable steht, die zumindest einmal gewählt wurde. Bei verhältnismäßig kleinen Gruppen führen beide Methoden zur Gruppierung zum selben Ergebnis; bei Daten, die in einem anderen Zusammenhang stehen, ergeben eventuell andere Gruppierungen mehr Sinn. Durch diese Zusammenfassung verkleinert sich die Datenmatrix, die neuen Einträge beinhalten dafür eine generellere Information über das Verhalten der einzelnen Gruppen und die Matrix ist wesentlich dichter besetzt.

Die Zielfunktion bekommt hier die Form

$$D(Y_P, C_K) = \frac{1}{P} \sum_{p=1}^{P} d(\mathbf{y}_p, c(\mathbf{y}_p)) \rightarrow \min_{C_K}$$

wobei $\mathbf{y}_p$ die neuen Einträge bezeichnet und P der Anzahl der Gruppen, dh. der Anzahl neuer Einträge, entspricht. Die Größe der einzelnen Gruppen wird durch N_p $(p = 1, ..., P)$ gegeben, wobei $N = \sum_{p=1}^{P} N_p$ gelten muss.

Zur leichteren Berechnung definiert man eine Indexmenge I_p, $p = 1, ..., P$, die die Objekte den Gruppen zuordnet. Sie beinhaltet die Must-link constraints aus der Hintergrundinformation, die bei der Segmentierung berücksichtigt werden müssen, dh. Objekte, die durch die Indexmenge zusammengehalten werden, müssen stets demselben Cluster zugeordnet werden.

Das Verfahren läuft dazu folgendermaßen:

1. Gruppierung der Daten zum Beispiel durch Berechnung von

$$y_{pj} = \frac{1}{N_p} \sum_{i \in I_p} x_{ij} \qquad \text{für} \qquad j = 1, ..., M; p = 1, ..., P$$

   und anschließende Binarisierung anhand der zeilenweise gebildeten Mittelwerte $\bar{x}_j = \frac{1}{N} \sum_{i=1}^{N} x_{ij}$ für $j = 1, ..., M$.

2. Wähle eine initiale Partition der $\mathbf{y}_p$ mit K Zentren C_K.

3. Ordne jeden Punkt $\mathbf{y}_p \in Y_P$ dem nächsten Zentrum zu.

4. Berechne die neuen Zentren, nachdem alle Punkte zugeordnet wurden:

$$\mathbf{c}_k := \arg\min_{\mathbf{c}\in C} \sum_{p:c(\mathbf{y}_p)=\mathbf{c}_k} d(\mathbf{y}_p, \mathbf{c}), \qquad k = 1, ..., K$$

5. Wiederholung der Punkte 3 und 4, bis sich die Zuordnung nicht mehr ändert und ein lokales Minimum gefunden wurde.

3.3 Simultanes Clustern der Gruppen

Um die Information über zusammengehörende Gruppen nicht nur vor oder nach der Partitionierung nutzen zu können, soll eine Modifikation des K-Means Algorithmus betrachtet werden, bei der die Bedingungen durch die Hintergrundinformation unmittelbar bei der Segmentierung beachtet werden. Dazu werden in jedem Iterationsschritt zuerst die einzelnen Objekte den Clusterzentren zugeordnet. Anhand einer Mehrheitsabstimmung über die Zuordnungen der Objekte einer Gruppe wird dann die jeweilige Gruppe einem Zentrum zugeordnet. Ist sichergestellt, dass keine Bedingung verletzt wurde, werden die Zentren neu bestimmt.

Es ergibt sich die Zielfunktion

$$D(X_N, C_K) = \sum_{p=1}^{P} \sum_{i\in I_p} d(\mathbf{x}_i, c(I_p)) \to \min_{C_K}$$

Hierfür ist folgende Vorgehensweise erforderlich:

1. Wähle eine initiale Partition mit K Zentren C_K, wobei sichergestellt sein muss, dass Objekte, die derselben Gruppe angehören, im selben Cluster liegen.

2. Ordne jeden Punkt $\mathbf{x}_n \in X_N$ dem nächsten Zentrum zu.

3. Finde für $p = 1, ..., P$ jeweils das Zentrum $c(I_p)$, dem die Mehrheit der $\mathbf{x}_i$ mit $i \in I_p$ zugordnet wurde, und weise alle $\{\mathbf{x}_i | i \in I_p\}$ diesem Zentrum zu.

4. Berechne die neuen Zentren, nachdem alle Punkte bzw. Gruppen zugeordnet wurden:

$$\mathbf{c}_k := \arg\min_{\mathbf{c}\in C} \sum_{n:c(\mathbf{x}_n)=\mathbf{c}_k} d(\mathbf{x}_n, \mathbf{c}), \qquad k = 1, ..., K$$

5. Wiederholung ab Punkt 2, bis sich die Zuordnung nicht mehr ändert und ein lokales Minimum gefunden wurde.

Alle drei bisher vorgestellten Verfahren beruhen auf dem in Abschnitt 2.3.1 dargestellten verallgemeinerten K-Means Algorithmus. Sie unterscheiden sich voneinander vor allem durch den Zeitpunkt zu dem die Bedingungen aus der Hintergrundinformation in die Partitionierung eingehen. Die Gruppen werden vor oder nach der Segmentierung oder in jedem einzelnen Iterationsschritt berücksichtigt. Die Lösungen, die sich dadurch ergeben, weichen bei den betrachteten Distanzmaßen deutlich voneinander ab, weil ohne eine Veränderung des Ergebnisses die Berechnung der Distanzen nicht mit der Summation vertauscht werden kann (Dreiecksungleichung). Bei der Betrachtung von linearen Distanzmaßen tritt dieses Problem nicht auf.

3.4 Clustern der Gruppen im Austauschverfahren

Alternativ soll eine weitere Modifikation des K-Means Algorithmus betrachtet werden, bei der in jedem Iterationsschritt nur eine Gruppe neu zugeteilt wird. Das heißt, bei jeder Wiederholung wird eine Gruppe zufällig ausgewählt, dem nächstliegenden Zentrum zugeordnet und die betroffenen Zentren werden neu berechnet. Dieses Vorgehen entspricht dem in Abschnitt 2.3.2 beschriebenen K-Means nach Hartigan insofern, dass die Zentren in jedem Iterationsschritt mit *einer* veränderten Zuordnung neu berechnet werden, jedoch wird nicht nur ein einzelnes Objekt neu zugeteilt, sondern eine Gruppe von Objekten, die aufgrund der vorhandenen Hintergrundinformation demselben Cluster zugeordnet werden muss.

Der Algorithmus arbeitet wie folgt:

1. Initialisierung:
   - Zufällige oder vorgegebene Einteilung der P Gruppen von Objekten $\mathbf{x}_n$ in K Cluster.
   - Zugehörige Clusterzentren werden berechnet.
2. Wähle eine Gruppe $G_p \in C_i$ $(p = 1, ..., P; i = 1, ..., K)$, die neu zugeordnet wird.
   - Berechne $D(\mathbf{x}_n, C_j)$ für alle $\mathbf{x}_n \in G_p$ und für alle $j = 1, ..., K$.

- Bestimme Zentrum C_j so, dass die Mehrheit der $D(\mathbf{x}_n, C_j)$ für $\mathbf{x}_n \in G_p$ minimal ist.
- Ordne alle $\mathbf{x}_n \in G_p$ dem Cluster C_j zu.
- Berechne die neuen Zentren für die Cluster C_i und C_j falls $i \neq j$.

3. Wiederholung von Punkt 2, bis sich die Zuordnung nicht mehr ändert und ein lokales Minimum gefunden wurde.

Die Zuordnung zu einem Cluster in Punkt 2 kann auch aufgrund der minimalen Gesamtdistanz

$$\arg \min_{j=1,\ldots,K} \sum_{\mathbf{x}_n \in G_p} D(\mathbf{x}_n, C_j)$$

der Punkte der betrachteten Gruppe zu diesem Cluster erfolgen.

Dieser Algorithmus erfordert natürlich deutlich mehr Iterationen als K-Means Varianten, bei denen simultan alle Punkte dem nächsten Clusterzentrum zugeordnet werden. Da die Gruppierung wie beim simultanen Clustern in jedem Iterationsschritt stattfindet, wird die Dreiecksungleichung bei der Berechnung der Distanzen beachtet und die beiden Vorgehensweisen liefern ähnliche Lösungen.

Kapitel 4

Synthetische Warenkorbdaten

Zur Anwendung der Verfahren sollen hier Daten, wie man sie von Warenkörben erwarten könnte, betrachtet werden. Durch den Einsatz von Computern und Scannerkassen hat mittlerweile fast jeder, der etwas verkauft, auch Aufzeichnungen darüber in digitaler Form. Häufig stehen zusätzlich noch Informationen über die Kunden zur Verfügung und die einzelnen Einkäufe können bestimmten Personen oder Firmen zugeordnet werden. Oft bleibt diese Fülle an Information jedoch unbeachtet gespeichert.

Ziel der Marktsegmentierung (Definition nach Smith, 1956) ist es aus dem heterogenen Markt kleinere homogene Märkte zu identifzieren. Innerhalb dieser kleinen Märkte sollen die Kunden ähnliche Präferenzen haben und daher gemeinsam adressierbar sein. So soll ein zielgerichtetes Marketing ermöglicht werden, das effizienter ist als Massenmarketing, bei dem z.B. alle Kunden diesselbe Zuschrift erhalten, und das günstiger und leichter zu realiseren ist als direktes Marketing, bei dem jeder Kunde persönlich kontaktiert wird.

Das Interesse die vorhandenen Daten zu Marketingzwecken zu nutzen ist gerade in letzter Zeit sehr gestiegen, wodurch es mittlerweile auch eine Fülle an Literatur, die sich dem Thema der Marketsegmentierung und der Partitionierung von Marketing Daten widmet, gibt. Allerdings wird nach meinem Wissensstand noch nicht auf Clustern mit Hintergrundinformation und Segmentierung von gruppierten Daten eingegangen, worauf wie eingangs erwähnt besonderer Wert gelegt werden soll.

Zum Test und Vergleich der verschiedenen Verfahren und Vorgehensweisen zur Segmentierung von Daten soll ein passender synthetischer Warenkorb erzeugt werden. Der Vorteil von simulierten Daten gegebenüber Echtdaten besteht darin, dass die wahren Clusterzugehörgkeiten bekannt sind und mit den Ergebnissen der Verfahren verglichen werden können. Andernfalls könn-

ten nur die Verfahren untereinander verglichen werden, was oft wenig Rückschlüsse darauf zulässt, welche Partitionierung die tatsächliche Struktur der Daten am besten widerspiegelt.

4.1 Ein Beispiel zur Anwendung der Verfahren an synthetischen Warenkorbdaten

Zur Veranschaulichung soll folgendes Beispiel betrachtet werden: Angenommen X besteht aus 18 Einkäufen $x_n \in \{0,1\}^6, n = 1, ..., 18$ von denen jeweils drei von derselben Person getätigt wurden und jeweils zwei Personen einem Typ angehören. So ergeben sich sechs verschiedene Personen P1 bis P6 und drei Typen α, β und γ. Die Einträge von X stehen in Tabelle 4.1 unter „einzelne Einkäufe“. Die Daten stammen aus dem HFC-Teil des im Folgenden vorgestellten synthetischen Warenkorbes.

Type	Person		einzelne Einkäufe						Gesamteinkauf					
			1	2	3	4	5	6	1	2	3	4	5	6
α	P1	1	1	0	0	0	0	0	1	1	0	0	0	1
		2	0	1	0	0	0	0						
		3	0	1	0	0	0	1						
	P2	4	1	0	0	0	0	0	1	1	1	0	0	1
		5	0	0	0	0	0	1						
		6	1	1	1	0	0	0						
β	P3	7	0	0	1	1	0	0	0	1	1	1	0	0
		8	0	0	1	0	0	0						
		9	0	1	0	0	0	0						
	P4	10	0	0	1	0	1	0	0	1	1	1	1	0
		11	0	1	0	1	0	0						
		12	0	1	1	0	0	0						
γ	P5	13	0	0	0	1	0	1	0	0	1	1	1	1
		14	0	0	0	1	0	0						
		15	0	0	1	1	1	1						
	P6	16	0	0	1	1	0	0	1	0	1	1	1	1
		17	0	0	0	0	1	0						
		18	1	0	0	1	0	1						

Tabelle 4.1: Beispiel zur Segmentierung von Warenkörben

4.1.1 Clustern in euklidischer Distanz

Segmentierung mit a posteriori Gruppierung

Zielfunktion:

$$D(X_N, C_K) = \frac{1}{18} \sum_{n=1}^{18} ||\mathbf{x}_n, c(\mathbf{x}_n)|| \rightarrow \min_{C_K}$$

Zur Initialisierung werden die tatsächlichen Zentren (siehe Tabelle 4.2) gewählt um mit möglichst wenig Iterationen zu einem möglichst brauchbaren Ergebnis zu kommen.

Im ersten Schritt berechnet man

$$\arg \min_{k \in \{1,2,3\}} \sum_{i=1}^{6} (x_{ni} - c_{ki})^2 \qquad \text{für} \qquad n = 1, ..., 18$$

um für jeden Einkauf x_n das - in euklidischer Distanz - nächste Zentrum zu kennen und erhält so die Zuordnung für die 18 Warenkörbe in Tabelle 4.2. Anschließend werden anhand dieser Zuteilung die neuen Zentren berechnet.

	Initialisierung						Zuordnung	neue Zentren					
c_1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{6}$	0	0	$\frac{1}{3}$	1, 2, 3, 4, 5, 6, 9	$\frac{3}{7}$	$\frac{4}{7}$	$\frac{1}{7}$	0	0	$\frac{2}{7}$
c_2	0	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{1}{6}$	0	7, 8, 10, 11, 12, 16, 17	0	$\frac{2}{7}$	$\frac{5}{7}$	$\frac{3}{7}$	$\frac{2}{7}$	0
c_3	$\frac{1}{6}$	0	$\frac{1}{3}$	$\frac{5}{6}$	$\frac{1}{3}$	$\frac{1}{2}$	13, 14, 15, 18	$\frac{1}{4}$	0	$\frac{1}{4}$	1	$\frac{1}{4}$	$\frac{3}{4}$

Tabelle 4.2: Ergebnisse der Partitionierung mit euklidischer Distanz

Im nächsten Iterationsschritt werden die Einkäufe den neuen Zentren nach demselben Prinzip zugeteilt, die Zuordnung verändert sich dadurch aber nicht. Teilt man nun die Personen nach Mehrheitsabstimmung den Zentren zu, wird Person 6 fälschlicherweise dem zweiten Cluster zugeordnet.

Segmentierung mit a priori Gruppierung

Die Indexmenge I_p, $p = 1, ..., 6$, die die Einkäufe den Personen zuordnet, hat die Form:

- Type A: $I_1 = \{1, 2, 3\}$, $I_2 = \{4, 5, 6\}$

- Type B: $I_3 = \{7, 8, 9\}$, $I_4 = \{10, 11, 12\}$
- Type C: $I_5 = \{13, 14, 15\}$, $I_6 = \{16, 17, 18\}$

Die gruppierten Daten y_p, $p = 1, ..., 6$ stehen in Tabelle 4.1 in der rechten Spalte unter „Gesamteinkauf“ und werden mittels

$$y_{pi} = \begin{cases} 1 & \text{falls } \exists\ l \in I_p \text{ mit } x_{li} = 1 \\ 0 & \text{falls } x_{li} = 0\ \forall l \in I_p \end{cases}$$

bestimmt.

Es ergibt sich die Zielfunktion

$$D(Y_P, C_K) = \frac{1}{6} \sum_{p=1}^{6} ||\mathbf{y}_p, c(\mathbf{y}_p)|| \rightarrow \min_{C_K}$$

Zur Initialisierung werden ebenfalls die tatsächlichen Zentren der Einzeleinkäufe (Tabelle 4.2 links) herangezogen. Möglich wäre auch die initiale Partition anhand der zugrundeliegenden Typen von Einkäufern der gruppierten Daten zu bestimmen; die zugehörigen Zentren weisen jedoch diesselbe Struktur wie die Zentren der Einzeleinkäufe auf.

Zur Lösung des Clusterproblems berechnet man die Zuordnung durch

$$\arg \min_{k \in \{1,2,3\}} \sum_{i=1}^{6} (y_{pi} - c_{ki})^2 \qquad \text{für} \qquad p = 1, ..., 6$$

Ordnet man diese Werte jeweils dem nächsten Clusterzentrum zu, werden alle sechs Personen korrekt klassifiert. Das heißt, die ermittelten Zentren stimmen mit den oben erwähnten tatsächlichen Zentren der gruppierten Daten überein.

c_1	1	1	$\frac{1}{2}$	0	0	1
c_2	0	1	1	1	$\frac{1}{2}$	0
c_3	$\frac{1}{2}$	0	1	1	1	1

Tabelle 4.3: Zentren der gruppierten Daten

Simultanes Clustern der Gruppen

Zielfunktion:

$$D(X_N, C_K) = \sum_{p=1}^{6} \sum_{i \in I_p} ||\mathbf{x}_i, c(I_p)|| \rightarrow \min_{C_K}$$

Konkret berechnet man zunächst wieder

$$\arg \min_{k \in \{1,2,3\}} \sum_{i=1}^{6} (x_{ni} - c_{ki})^2 \qquad \text{für} \qquad n = 1, ..., 18$$

und wählt dann für jede Person die häufigste Zuteilung.

In diesem Beispiel bedeutet das, dass die Personen 1 und 2 in Cluster 1 kommen, Cluster 2 besteht aus den Personen 3, 4 und 6 und Cluster 3 besteht aus Person 5.

c_1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{6}$	0	0	$\frac{1}{3}$
c_2	$\frac{1}{9}$	$\frac{1}{3}$	$\frac{5}{9}$	$\frac{4}{9}$	$\frac{2}{9}$	$\frac{1}{9}$
c_3	0	0	$\frac{1}{3}$	1	$\frac{1}{3}$	$\frac{2}{3}$

Tabelle 4.4: Ermittelte Zentren durch simultanes Clustern der Gruppen

Ordnet man die Personen jedoch nicht nach der Mehrheit sondern nach dem niedrigsten Gesamtabstand zu und berechnet

$$\arg \min_{k \in \{1,2,3\}} \sum_{i \in I_p} ||x_{ki} - c_{ki}|| \qquad \text{für} \qquad p = 1, ..., 6$$

so erhält man eine korrekte Partitionierung wie im Fall der gruppierten Daten.

Clustern der Gruppen im Austauschverfahren

Ordnet man in jedem Iterationsschritt jeweils eine Person dem nächstgelegenen Zentrum zu, erhält man dasselbe Ergebnis wie durch die im voranstehenden Abschnitt beschriebene Vorgehensweise, dh. Person 6 wird zu Cluster 2 zugeteilt.

4.1.2 Clustern mit Jaccard- Distanz

Segmentierung mit a posteriori Gruppierung

Die Zuteilung der Einkäufe zu den Zentren soll nun anhand des Jaccard-Distanzmaßes erfolgen, wobei zu beachten ist, dass zur korrekten Berechnung dieser Distanz binäre Zentren erforderlich sind. Tabelle 4.5 zeigt die ermittelte Partitonierung sowie die Zentren, die sich nach dem zweiten Iterationsschritt nicht mehr verändern.

	Initialisierung						Zuordnung	neue Zentren					
c1	1	1	0	0	0	1	1, 3, 4, 5, 18	1	0	0	0	0	1
c2	0	1	1	0	0	0	2, 6, 7, 8, 9, 10, 11, 12, 16	0	1	1	0	0	0
c3	0	0	0	1	1	1	13, 14, 15, 17	0	0	0	1	1	1

Tabelle 4.5: Ergebnisse der Segmentierung mit Jaccard- Distanz

Bei der a posteriori Gruppierung werden die Personen 1 und 2 Cluster 1 zugewiesen, Cluster 2 enthält die Personen 3 und 4, und Person 5 wird dem 3. Cluster zugewiesen. Person 6 kann nur einem beliebigen Cluster zugeordnet werden.

Segmentierung mit a priori Gruppierung

Auch mit der Jaccard-Distanz werden die sechs Gesamteinkäufe korrekt in drei Gruppen geteilt.

Simultanes Clustern der Gruppen

Wie schon die Zuteilung in obiger Tabelle zeigt, werden die ersten fünf Personen bei Mehrheitsabstimmung richtig klassifiziert. Die 6. Person kann nur zufällig einem Cluster zugeordnet werden, wobei die Zentren erhalten bleiben, falls man Cluster 1 oder 3 wählt. Ordnet man Person 6 zu Cluster 2 zu, wird bereits im nächsten Iterationsschritt Cluster 3 leer.

Die Zuteilung aufgrund des minimalen Gesamtabstands der Personen zu den Clusterzentren ergibt wieder eine korrekte Segmentierung.

Clustern der Gruppen im Austauschverfahren

Die Segmentierung erfolgt eindeutig und korrekt für die Personen 1-5. Für Person 6 wird ein Cluster zufällig gewählt.

4.1.3 Zusammenfassung des Anwendungsbeispiels

Das vorgestellte Beispiel zeigt, dass die Segmentierung mit a posteriori Gruppierung, wobei der gesamte Datensatz geclustert wird, und die Segmentierung mit a priori Gruppierung, bei der der deutlich kleinere Datensatz der gruppierten Daten geclustert wird, sowie das simultane Clustern der Gruppen jeweils unterschiedliche Ergebnisse liefert. Auch die Ergebnisse der Partitionierung mit euklidischer Distanz und mit der Jaccard Distanz sind bereits bei diesem kleinen Datensatz zum Teil deutlich voneinander verschieden.

4.2 Die Simulation der Daten

Die R-Funktion `basket` simuliert einen Warenkorb bei vorgegebener Verteilung. Das zugehörige Design wurde in Kooperation mit dem Institut für Handel und Marketing der Wirtschaftsuniversität Wien erstellt. Die Daten werden als Binärdaten generiert, wobei 1 bedeutet, dass dieses Produkt bzw. ein Produkt dieser Kategorie bei diesem Einkauf gekauft wird, und 0 bedeutet, dass kein Produkt gekauft wird. Die Anzahl der gekauften Produkte ist dabei nicht relevant. Der Funktionsaufruf erfolgt mit `basket(cat,n)`, wobei `cat` für die Anzahl der Produktkategorien und `n` für die Anzahl der Warenkörbe steht.

Ausgangspunkt für die gewählte Verteilung ist ein typisches Handelssortiment eines Lebensmittelsupermarktes mittlerer Größe, die Produkte werden deterministisch in vier Kategorien eingeteilt:

- 70% LFC: Low-Frequency-Categories mit Support $< 5\%$
- 20% MFC: Medium-Frequency-Categories mit Support $< 10\%$, aber $> 5\%$
- 5% HFC: High-Frequency-Categories mit Support $< 20\%$, aber $> 10\%$
- 5% VHFC: Very High-Frequency-Categories mit Support $> 20\%$

Bei den Einkäufen, dh. bei den einzelnen Warenkörben werden zwei Gruppen unterschieden:

- (A) Major Shopping Trips: Der klassische Wochenendeinkauf mit relativ großen Warenkörben mit Produkten aus allen Kategorien
- (B) Fill-In-Trips: Kleinere Warenkörbe, die unter der Woche bzw. von kleineren Haushalten realisiert werden

Zur Simulation werden die Warenkörbe in die Untergruppen A1 - A3 mit jeweils 10% und B1 - B3 mit jeweils 15% der Einkäufe unterteilt. Zusätzlich gibt es noch die Gruppen Rest1 (10%) und Rest2 (15%) als Störfaktoren. Die Anzahl der Produkte einer Kategorie wird poissonverteilt angenommen mit den in Tabelle 4.6 zusammengefassten Erwartungswerten.

Weiters werden die Warenkörbe in drei Kategorien (α, β, γ), die die Produktvorlieben symbolisieren, unterteilt. Für die Simulation bedeutet das, dass bei jedem Einkauf 80% der Produkte der MFC- und HFC-Kategorie aus einem

Type	Größe	Gesamt	LFC	MFC	HFC	VHFC
A1	10%	20	3	6	4	7
A2	10%	12	1	5	3	4
A3	10%	8	1	4	2	3
B1	15%	6	-	2	2	3
B2	15%	4	-	1	1	2
B3	15%	3	-	-	1	2
Rest1	10%	2				
Rest2	15%	1				

Tabelle 4.6: Vorgaben zur Simulation des synthetischen Warenkorbes

festgelegten Teil gewählt werden. Dabei wählt die Gruppe α bevorzugt die ersten $\frac{4}{10}$ der zur Auswahl stehenden Produktgruppen, die Gruppe β kauft eher die mittleren $\frac{5}{10}$, und die Gruppe γ wählt vorrangig aus den letzten $\frac{6}{10}$.

Außerdem ist die Simulation so gestaltet, dass bei Einkäufen des Types (B) (Fill-In-Trips) jede fünfte Produktkategorie nur mit einer Wahrscheinlichkeit von 1% gewählt wird.

4.3 Der Datensatz

Nach diesen Vorgaben wurde ein Datensatz mit 200 Produktkategorien und 15000 Warenkörben generiert. Dabei entstanden die in Tabelle B.1 zusammengefassten Gruppenzugehörigkeiten.

Type	(A)			(B)			Rest	Gesamt
	A1	A2	A3	B1	B2	B3		
α	528	485	476	716	737	717		3659
β	502	537	505	755	789	701		3789
γ	519	478	478	743	771	777		3766
Rest							3786	3786
Gesamt	1549	1500	1459	2214	2297	2195	3786	15000

Tabelle 4.7: Gruppenzugehörigkeiten des synthetischen Warenkorbes

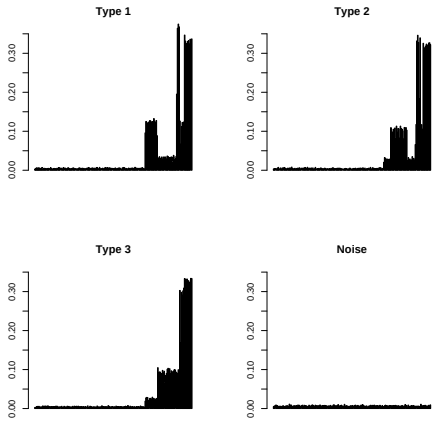

Abbildung 4.1: Barplot des gesamten Datensatz

Abbildung 4.1 zeigt die Barplots der Warenkörbe aufgeteilt nach den drei Kategorien, die die Produktvorlieben symbolisieren sollen, sowie einen Plot der Gruppe „Rest“. Abbildung 4.2 zeigt die Barplots der 4 verschiedenen Typen (α, β, γ und Rauschen) einzeln für die Produktkategorien LFC (links oben), MFC (rechts oben), HFC (links unten) und VHFC (rechts unten). Es zeigt sich, dass aus den LFC Produkten durchschnittlich sogar weniger gekauft wird, als das Rauschen ausmacht, die VHFC Produkte werden im Gegensatz dazu gleichmäßig sehr viel gekauft, diese Kategorien sind daher vermutlich wenig hilfreich für die Partitionierung. Die MFC und HFC Produktkategorien zeigen deutlich die zugrundeliegende Struktur.

4.4 Gruppieren von Warenkörben

Wie bereits beschrieben, wäre es in der Praxis des Marketings denkbar, dass zum Beispiel über ein Kundenbindungsprogramm Zusammenhänge der einzelnen Warenkörbe untereinander bekannt sind. Ein Kunde soll dabei immer demselben Typ angehören, kauft aber dennoch jedesmal teilweise unterschiedliche Produkte. Um dies anhand des simulierten Datensatzes nachzuvollziehen, wird eine gewählte Anzahl verschiedener Warenkörbe desselben Typs zu einer Person zusammengefasst, indem die Daten pro Person und pro Produktkategorie gemittelt und anschließend wieder binarisiert werden. Alternativ soll auch ein Datensatz betrachtet werden, der beschreibt, welche Produkte zumindest einmalig gekauft wurden. Für jeden Kunden bleibt da-

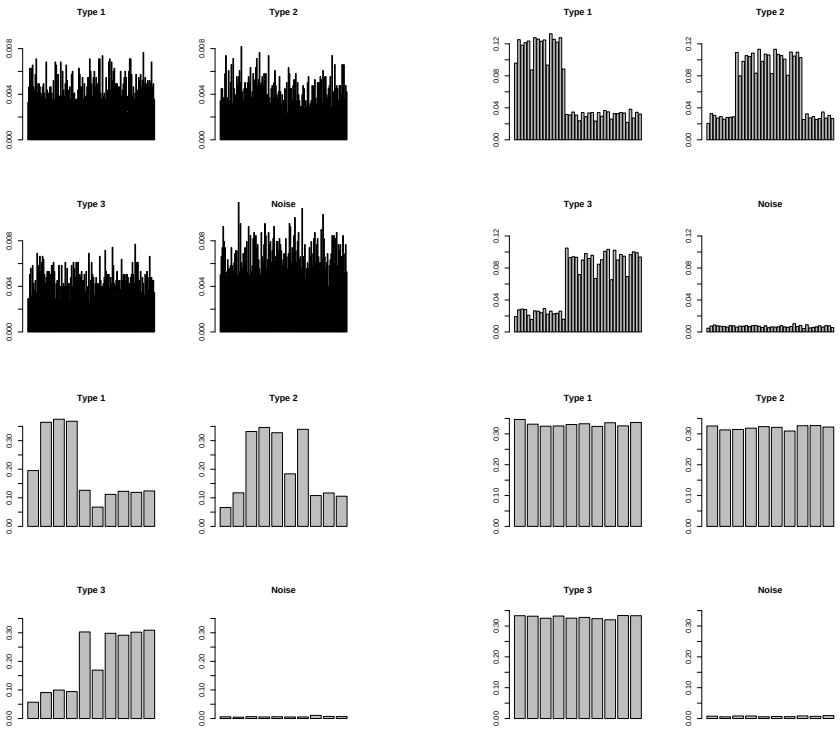

Abbildung 4.2: Barplots der Produktkategorien LFC, MFC, HFC und VHFC

mit ein Warenkorb im Datensatz. Dieser Schritt erleichtert die Segmentierung bedeutend, da er die drei verschiedenen Typen, die der Datensatz enthält, für den Clusteralgorithmus sichtbarer macht und gleichzeitig die Anzahl der zu clusternden Objekte deutlich reduziert. Dabei gilt, dass die Ergebnisse, zu denen die Clusterverfahren gelangen, umso mehr mit den zugrundeliegenden Gruppen übereinstimmen, je mehr Warenkörbe jeweils zusammengefasst werden.

Weiters soll zum Vergleich der Segmentierung der gruppierten Daten mit dem bereits in Abschnitt 3.3 vorgestellten modifizierten K-Means Algorithmus zur simultanen Segmentierung der Gruppen ein Vektor zur Verfügung stehen, der Auskunft über die zusammengefassten Objekte des gesamten Datensatzes gibt.

Die Gruppierung der Warenkörbe erfolgt aufgrund der Typen α, β und γ. Dadurch fällt die Gruppe „Rest“, deren Einträge keinem dieser Typen zuzuordnen sind, zunächst weg. Es ergibt sich ein Datensatz, der vor der Gruppierung 11214 Warenkörbe enthält. Auf die Betrachtung dieses Datensatzes

soll jedoch im Folgenden nicht näher eingegangen werden.

Fügt man die Daten der Gruppe „Rest“ dem bereits gruppierten Datensatz hinzu, entstehen überproportional viele Warenkörbe, die keinem Typ zuzuordnen sind. Daher wird für die Gruppierung des Datensatzes inklusive Rauschen dieses als eigener Typ betrachtet und ebenfalls reduziert, als könnte man auch dort Warenkörbe von Kunden zusammenfassen.

Um ausreichend Vergleichsmöglichkeiten zu erhalten, werden aus dem gesamten synthetischen Warenkorb sechs verschiedene Datensätze mit gruppierten Daten unterschiedlicher Gruppengröße generiert. Tabelle 4.8 zeigt die Größen der betrachteten Datensätze sowie den Unterschied der beiden Wege zur Binarisierung der Daten. Da der Mittelwert über alle Einkäufe zumeist unter dem Wert liegt, der entsteht, wenn ein Produkt von einem Kunden nur einmal gekauft wurde, unterscheiden sich die beiden Datensätze kaum. Je größer die Kundengruppen werden, umso größer ist aber der Unterschied.

# versch Kunden	Größe des Datensatzes	Gruppen-größe	# ungleicher Einträge	Unterschied in %
1500	2000	7-8	5552	1.39
2001	2668	5-6	8275	1.55
2502	3336	4-5	2056	0.31
3000	4000	3-4	0	0
3738	4984	2-4	0	0
5460	7280	2-3	0	0

Tabelle 4.8: Vergleich der gruppierten Datensätze

Mit der R-Funktion `groupData(data, types, n)` erhält man also für einen Datensatz (`data`) mit gegebenen Gruppenzugehörigkeiten (`types`) und für eine bestimmte Anzahl verschiedener Kunden (`n`) einen gruppierten binarisierten Datensatz wie beschrieben. Dieser ist aufgrund des Zufügen des Rauschens um $\frac{1}{3}$ größer als die gewählte Kundenanzahl. Die Funktion gibt auch die tatsächlichen Zentren und Clusterzugehörigkeiten der gruppierten Daten zurück, damit kann beispielsweise ein Clusteralgorithmus initialisiert werden. Weiters erhält man einen Vektor der Länge des ursprünglichen Datensatzes, der die Gruppenzugehörigkeiten - also die simulierten Personen - wiedergibt.

Alternativ liefert die Funktion `groupDataGekauft1` einen Datensatz, der darüber Auskunft gibt, welche Produkte zumindest einmalig gekauft wurden.

Kapitel 5

Analyse des synthetischen Warenkorbes

Ausgehend von den simulierten Basket-Daten sollen jetzt verschiedene Verfahren zur Segmentierung getestet werden. Interessant ist nicht nur, wie weit die Ergebnisse mit den den Daten zugrundeliegenden Gruppen übereinstimmen, sondern vor allem die Struktur der Cluster sowie die Stabilität der verschiedenen Verfahren. Abbildung A.1 zeigt anhand von Barplots die Struktur der zugrundeliegenden Gruppen für den späteren Vergleich mit den Ergebnissen der verschiedenen Clusterlösungen.

5.1 Segmentierung der simulierten Daten

Am Beginn sollen der gesamte simulierte Datensatz und die Teile, die durch die Wahl des Typs speziell beeinflusst wurden, das heißt die Teile, die die MFC- und HFC- Produktgruppen enthalten, ohne eine Gruppierung der Warenkörbe mit verschiedenen Clusterverfahren mit euklidischem und Jaccard Distanzmaß analysiert werden.

5.1.1 Das verallgemeinerte K-Means Verfahren mit euklidischem Distanzmaß

Den Anfang bildet der in Abschnitt 2.3.1 vorgestellte Algorithmus. Für verschiedene Clusterzahlen wird das Verfahren jeweils mehrfach auf die Daten angewandt um ein möglichst „gutes“ lokales Minimum zu finden. Die Auswahl erfolgt dabei aufgrund der minimalen Distanz der Punkte innerhalb

jedes Clusters nach dem euklidischen Distanzmaß. Zur Berechnung diente die R-Funktion `cclust` mit der Methode `"kmeans"`.

Abbildung 5.1 zeigt die minimal erreichten Summen der Abstände der Punkte innerhalb jedes Clusters für unterschiedliche Clusterzahlen bei einer Segmentierung des gesamten Datensatzes (links) und der speziell interessanten Teile mit den MFC (mitte) und HFC (rechts) Produktgruppen. Je nach Struktur der Daten kann man manchmal anhand dieses Graphen feststellen, welche Anzahl an Clustern sinnvoll ist. Mit zunehmender Clusterzahl werden die Abstände zwischen den Punkten eines Cluster und seinem Zentrum geringer. Zeigt der Plot, dass ab einer bestimmten Anzahl kaum mehr eine Verbesserung eintritt, so ist das ein deutlicher Hinweis diese Zahl an Clustern zu wählen. Im vorliegenden Fall kann man jedoch anhand der Graphen kaum eine Aussage treffen.

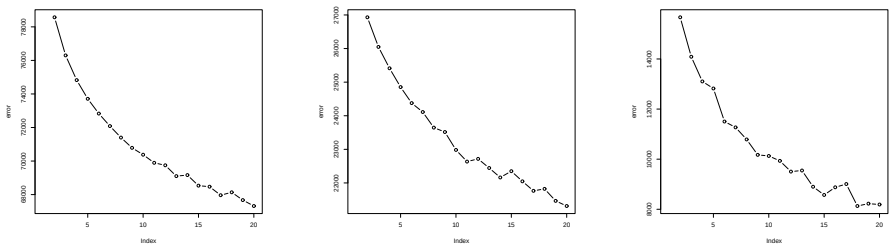

Abbildung 5.1: Distanzen der Punkte innerhalb von 2 bis 20 Clustern mit K-Means Partitionierung des gesamten Datensatzes, der 40 MFC und der 10 HFC Produktgruppen

Abbildung A.2 zeigt konkrete Clusterlösungen für 3, 4, 5 und 6 Cluster. Man erkennt, dass die Kleineinkäufe und damit große Teile des Rauschens in einem Cluster sind. Zum Teil stimmt die Struktur der anderen Cluster mit der Struktur der tatsächlichen Gruppen überein, insgesamt gibt es jedoch wenig Übereinstimmung mit den zugrundeliegenden Gruppen, wie der Vergleich in Abschnitt 5.1.4 zeigt.

5.1.2 Die Neural Gas Methode

Dieses in Abschnitt 2.3.4 vorgestellte Verfahren wird durch die Auswahl der Methode `"neuralgas"` der R-Funktion `cclust` auf die Daten angewandt. Es erreicht im allgemeinen niedrigere „Fehler“ als K-Means, doch auch hier lässt sich anhand der minimal erreichten Summen der Distanzen der Punkte

innerhalb desselben Clusters keine Aussage über eine vernünftige Anzahl von Clustern ableiten (Abbildung 5.2 links für den gesamten Datensatz und mitte für die 40 MFC Produktkategorien).

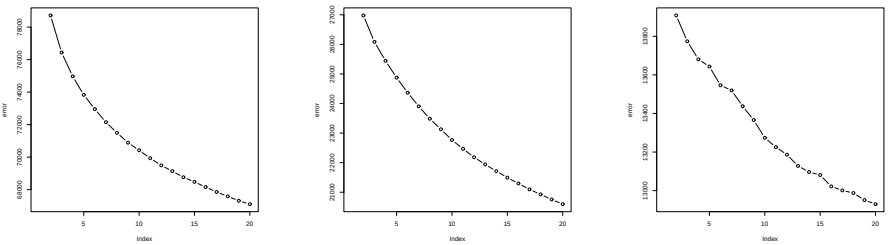

Abbildung 5.2: Distanzen der Punkte innerhalb von 2 bis 20 Clustern mit neuralgas Partitionierung des Datensatzes und der 40 MFC Produkte sowie bei Clustern mit Jaccard-Distanzmaß

5.1.3 Clustern mit Jaccard-Distanzmaß

Da die Segementierung der Daten mit euklidischer Distanz nicht die erhoffte Klarheit über die Anzahl der Cluster gebracht hat, liegt nach den Ausführungen in Abschnitt 2.1 der Gedanke nahe, Partitionierungen mit Jaccard-Distanzmaß zu betrachten. Die R-Funktion `kcca` aus dem Paket `flexclust` bietet dazu zwei unterschiedliche Vorgehensweisen. Mit dem Argument `kccaFamily("jaccard")` löst die Funktion das komplexe Optimierungsproblem zur Berechnung der Zentren, was deutlich mehr Rechenzeit in Anspruch nimmt als die Berechnung mit `kccaFamily("ejaccard")`, die den Mittelwert als Zentrum wählt (siehe Leisch, 2005b). Wie Abbildung 5.2 (rechts, die Berechnung erfolgte mit `kccaFamily("ejaccard")`) zeigt, liefert jedoch auch diese Analyse keine Aussage über eine sinnvolle Clusterzahl.

5.1.4 Vergleich der Ergebnisse der Clusteranalyse mit den zugrundeliegenden Gruppen

Bei der Simulation der Daten werden die Warenkörbe aufgrund der Zugehörigkeit zu verschiedenen Gruppen generiert. Dabei gibt es sowohl verschiedene Typen von Einkäufen (A1, A2, A3 und B1, B2, B3) als auch verschiedene Typen von Einkäufern (α, β, γ). Je nachdem nach welchen Gruppen gesucht wird, wie die Gruppen kombiniert werden und ob der Rest als

eigene Gruppe betrachtet wird, ergeben sich unterschiedliche Gesamtgruppenzahlen. So gibt es zum Beispiel die Möglichkeit nach den 3 Typen von Einkäufern zu suchen oder inklusive der Gruppe „Rest“ 4 Gruppen zu betrachten. Weiters wäre es denkbar zwischen Einkäufen der Kategorie A und und solchen der Kategorie B zu unterschieden und dementsprechend 6 bzw. 7 Typen von Einkäufern zu untersuchen. Auch die Einkäufe selbst kann man so in 6 bzw. 7 Gruppen unterteilen. Eine Kombinierung dieser Einteilung ergibt dann 18 bzw. 19 Gruppen.

Die Gruppe „Rest“, der ca. 25% der Warenkörbe angehören, stellt bewusst einen Störfaktor dar. Wünschenswert wäre es, diese Daten in Form eines eigenen Clusters zu identifizieren, andernfalls sollten sie vor dem Vergleich entfernt werden.

Die Ergebnisse der Clusterverfahren, insbesondere die verbleibende Distanz jedes Punktes zum nächstgelegenen Zentrum, liefern leider unabhängig vom gewählten Verfahren und vom augenblicklich betrachteten Teil des Datensatzes kaum Hinweise auf eine vernünftige Anzahl der Cluster. Daher sollen die Ergebnisse mit unterschiedlichen zugrundeliegenden Gruppenzuordnungen verglichen werden, es werden jedoch verstärkt Lösungen mit drei oder vier Gruppen betrachtet, die den Typen von Einkäufern mit und ohne Rauschen entsprechen.

	#	kmeans		neuralgas		ejaccard
	Cluster	ohne Init	mit Init	ohne Init	mit Init	ohne Init
Einkäufer	3	~ 0	~ 0	~ 0	~ 0	~ 0
	4	0.208	0.208	0.200	0.193	0.100
	6	0.097	0.102	0.092	0.087	0.109
	7	0.339	0.335	0.340	0.342	0.097
Einkäufe	6	0.104		0.103	0.112	0.122
	7	0.340		0.343	0.343	0.110
Komb.	18	0.067	0.075	0.067	0.063	0.064
	19	0.490	0.510	0.513	0.512	0.044

Tabelle 5.1: Werte des korrigierten Rand Index für den Vergleich der zugrundeliegenden Typen mit den Clusterlösungen

Um die Auffindung der Gruppen zu erleichtern, werden die Algorithmen zum Teil auch mit den „richtigen“ Zentren initialisert. Wie Tabelle 5.1 zeigt, liefert das jedoch keine besseren Ergebnisse als die Partitionierung ohne Initialisierung. Als Maßzahl für die Übereinstimmung soll dabei der korrigierte Rand

Index dienen.

Bei 3, 6 und 18 Gruppen werden die Ergebnisse mit den Daten ohne Störterm verglichen, bei 4, 7 und 19 wird der Gruppe "Rest" ein eigener Cluster zugewiesen und somit werden alle Daten für den Vergleich herangezogen. Diese Zahlen zeigen, dass die Ergebnisse der Segmentierung kaum mit den zugrundeliegenden Gruppen übereinstimmen und in Folge dessen, dass die Struktur der Daten auf diesem Weg nicht gefunden werden kann.

HFC	Cluster	kmeans	
		ohne Init.	mit Init.
Einkäufer	3	0.052	0.089
	4	0.110	0.213
	6	0.075	0.103
	7	0.201	0.256
Einkäufe	6	0.066	
	7	0.196	
Komb.	18	0.049	-
	19	0.307	-

Tabelle 5.2: Vergleichswerte bei Segmentierung der HFC Produktkategorien

MFC	Cluster	kmeans		neuralgas	
		ohne Init.	mit Init.	ohne Init.	mit Init.
Einkäufer	3	0.051	0.052	0.050	0.052
	4	0.067	0.067	0.067	0.067
	6	0.172	0.160	0.174	0.160
	7	0.178	0.188	0.173	0.188
Einkäufe	6	0.173		0.174	
	7	0.178		0.177	
Komb.	18	-	-	0.091	-
	19	-	-	0.213	-

Tabelle 5.3: Vergleichswerte bei Segmentierung der MFC Produktkategorien

Möglicherweise wäre es für die Verfahren einfacher die Struktur zu erkennen, wenn nur der Teil der Daten, der die Struktur enthält, untersucht wird. Die Tabellen 5.2 und 5.3 zeigen die entsprechenden Werte für die Untersuchung der 10 HFC Produktkategorien (5.2) sowie der 40 MFC Produktkategorien (5.3). Die erhoffte Verbesserung bleibt jedoch aus.

5.2 Clustern der gruppierten Warenkörbe

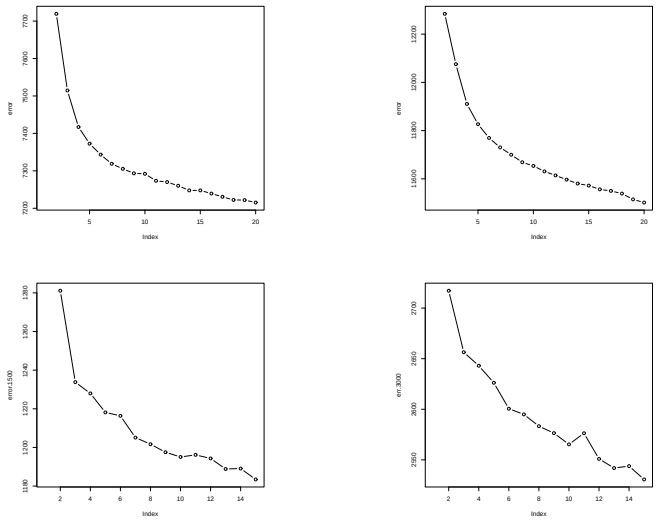

Abbildung 5.3: Distanzen der geclusterten Punkte bei gruppierten Daten

Bei der Gruppierung der Daten werden die drei Typen und das Rauschen berücksichtigt. Je nachdem, ob jenes als eigene Gruppe klassifiziert werden kann, macht es Sinn drei oder vier Cluster zu betrachten. Zunächst sollen dazu wieder Plots der minimal erreichten Summen der Distanzen innerhalb der Cluster betrachtet werden. Abbildung 5.3 zeigt die Distanzen für 2 bis 20 Cluster bei Segmentierung mit K-Means (Berechnung mit `kccaFamiliy("kmeans")`) in der ersten Zeile und für 2 bis 15 Cluster bei K-Centroids Segmentierung mit Jaccard-Distanz (Berechnung mit `kccaFamiliy("jaccard")`) in der zweiten Zeile. Die linken Plots stammen dabei von der Analyse der nach 1500 Kunden gruppierten und anschließend binarisierten Daten, während für die rechten Plots 3000 Kunden angenommen wurden. Auch bei den gruppierten Daten liefern diese Plots keinen Hinweis auf eine zu bevorzugende Clusterzahl.

Um festzustellen inwiefern die Clusteralgorithmen die zugrundeliegenden Kundentypen wiederfinden, zeigt Tabelle 5.4 die Werte des korrigierten Rand Index für den Vergleich einer Einteilung in 4 Cluster mit den inklusive Rauschen 4 Typen. Für eine Gruppierung zu 1500 Kunden zeigt Abbildung A.3 die Barplots der vier Clusterzentren. Analog zeigt Abbildung A.4 die Barplots zur Segmentierung der zu 3000 Kunden gruppierten Daten. Es zeigt sich, dass der K-Means Algorithmus die Typen ausgesprochen gut erkennt. Die euklidische Distanz ist hier offensichtlich besser geeignet als das Jaccard Distanzmaß. Bemerkenswert ist vor allem, dass der K-Means Algorithmus das Rauschen gut erkennt und ihm einen eigenen Cluster zuweist, sowie dass die erzielten Cluster ähnliche Größen aufweisen. Wie die Barplots zeigen, gelingt das der Partitionierung mit absoluter und Jaccard-Distanz kaum.

# versch Kunden	Gruppen-größe	kmeans	kmedians	ejacc.	jaccard
1500	7	0.94	0.69	0.61	0.68
2001	5	0.82	0.36	0.09	0.59
2502	4	0.78	0.37	0.07	0.50
3000	3	0.77	0.50	0.06	0.38
3738	2-3	0.48	0.43	0.07	0.16
5460	2	0.40	0.17	0.08	0.23

Tabelle 5.4: Vergleichswerte bei Partitionierung der gruppierten Daten in 4 Cluster (Type α, β, γ und Rest)

Als alternativen Ansatz zum Vergleich der Clusterlösung mit den Typen kann man sich auch für drei Cluster entscheiden und die Gruppe „Rest“ vor dem Berechnen des korrigierten Rand Index entfernen. Die Ergebnisse, die in Tabelle 5.5 zusammengefasst sind, zeigen, dass K-Means in diesem Fall deutlich schlechtere Ergebnisse liefert, da dieser Algorithmus nach wie vor das Rauschen einem eigenen Cluster zuordnet, der für den Vergleich unbrauchbar ist. Im Gegensatz dazu liefert die Partitionierung mit dem Jaccard Distanzmaß hier bessere Werte. Die Barplots der Segmentierungen der zu 1500 und 3000 Kunden gruppierten Daten zeigen die Abbildungen A.5 und A.6.

Bei den Datensätzen mit 1500 und 3000 verschiedenen Kunden werden auch die MFC- und HFC- Daten getrennt analysiert und in 3 bzw. 4 Cluster eingeteilt. Die zugehörigen Werte des korrigierten Rand Index sind in Tabelle 5.6 zusammengefasst. Obwohl diese Teile des Datensatzes die Struktur der Typen enthalten, zeigt die Analyse der Clusterzuteilungen weniger Übereinstimmungen mit den zugrundeliegenden Typen.

# versch Kunden	Gruppen-größe	kmeans	kmedians	jaccard
1500	7	0.54	0.43	0.91
2001	5	0.47	0.01	0.78
2502	4	0.45	0.20	0.70
3000	3	0.42	0.17	0.53
3738	2-3	0.34	0.17	0.21
5460	2	0.24	0.43	0.31

Tabelle 5.5: Vergleichswerte bei Partitionierung der gruppierten Daten in 3 Cluster (Type α, β und γ)

# versch	#	MFC - Daten			HFC- Daten		
Kunden	Cluster	kmeans	kmedians	jaccard	kmeans	kmedians	jaccard
1500	3	0.46	0.22	0.82	0.37	0.27	0.40
1500	4	0.74	0.27	0.50	0.71	0.56	0.35
3000	3	0.23	0.02	0.49	0.23	0.17	0.24
3000	4	0.33	0.01	0.31	0.46	0.36	0.25

Tabelle 5.6: Vergleichswerte bei Partitionierung des MFC- und HFC-Teils der gruppierten Daten

5.3 Simultanes Clustern der Personen

Die Vergleichsindizes zwischen den zugrundeliegenden Gruppen und den ermittelten Clustern bei Partitionierung der gruppierten Daten zeigen, dass die Typen unter günstigen Voraussetzungen (zB. möglichst viele Einkäufe pro Kunde) gut erkannt werden. Nun soll wieder der gesamte Datensatz betrachtet und geclustert werden, wobei die Gruppen von Einzeleinkäufen wie in Abschnitt 3.3 beschrieben bei der Segmentierung berücksichtigt werden. Die R-Funktion `kcca` bietet dafür die Möglichkeit einen Vektor mit der Indexmenge, die die Gruppen beinhaltet, zu verarbeiten. In jeder Iteration wird sichergestellt, dass verknüpfte Warenkörbe nicht durch unterschiedliche Clusterzuordnungen getrennt werden. Die Werte des korrigierten Rand Index für den Vergleich zwischen vier auf diese Weise ermittelten Clustern und den vorhandenen Typen stehen in Tablle 5.7, die Algorithmen wurden mit den zugrundeliegenden Gruppen initialisert.

Die Vermutung, dass dieses Vorgehen wesentlich erfolgreicher ist als die Seg-

# versch Kunden	Gruppen-größe	kmeans	jaccard
1500	7	0.41	0.39
2001	5	0.36	0.32
2502	4	0.31	0.30
3000	3	0.30	0.11
3738	2-3	0.21	0.11
5460	2	0.23	0.11

Tabelle 5.7: Vergleichswerte bei Partitionierung mit dem modifizierten K-Means Algorithmus

mentierung ohne Berücksichtigung der Gruppen, wird durch die Ergebnisse bestätigt. So leicht wie bei den gruppierten Daten fällt es den Algorithmen in diesem Fall jedoch nicht die Typen wiederzuerkennen.

Von einer Aufteilung in drei Cluster wurde hier Abstand genommen, da dieser zur simultanen Segmentierung der Gruppen modifizierte K-Means Algorithmus vor allem das Rauschen gut als eigenen Cluster identifizieren kann. Clustert man die Daten mit den vorgegebenen Gruppen in drei Cluster, so findet sich zumeist in einem Cluster nur Rauschen oder das Rauschen liegt zur Gänze im selben Cluster und der Vergleich der Cluster mit den zugrundeliegenden Typen ergibt wenig Sinn. Auffällig dabei ist jedoch der Unterschied zwischen euklidischer Distanz und Jaccard Distanz. Während bei einer Segmentierung mit euklidischer Distanz das gesamte Rauschen im selben Cluster liegt, in dem sich auch weitere Objekte befinden, entsteht bei einer Segemtierung mit der Jaccard Distanz ein Cluster, in dem nur (jedoch nicht alle) Datenpunkte des Rauschens liegen.

5.4 Clustern der Personen im Austauschverfahren

Wie in Abschnitt 3.4 beschrieben soll auch eine Clusterlösung ermittelt werden, die - ausgehend von der „korrekten“ Zuteilung - einzeln die Personen den Zentren neu zuweist und nach jeder Neuzuordnung die Zentren neu berechnet. Die Durchführung erfolgte mir der R-Funktion

```
einzelupdate(data, k=NULL, cent=NULL, group, iter.max = 100)
```

Falls Zentren zur Initialsierung des Algorithmus vorhanden sind, werden diese mit `cent` übergeben. Andernfalls muss die Anzahl der Cluster `k` spezifiziert werden. `group` übernimmt die Indexmenge mit den Gruppen, die im selben Cluster liegen müssen.

Wie Tabelle 5.8 für vier Cluster zeigt, weichen die Lösungen etwas weniger von den vorhandenen Typen ab als bei Partitionierungen durch simultane Segmentierung der Gruppen. Betrachtet man nur drei Cluster, verhält sich die Zuordnung des Rauschens insgesamt wie im voranstehenden Abschnitt beschrieben. Die konkrete Zuordnung der einzelnen Personen durch die beiden Algorithmen ist jedoch zum Teil deutlich voneinander verschieden.

# versch Kunden	Gruppen-größe	euklid.	jaccard
1500	7	0.50	0.50
2001	5	0.45	0.42
2502	4	0.40	0.33
3000	3	0.39	0.13
3738	2-3	0.26	0.36
5460	2	0.33	0.12

Tabelle 5.8: Vergleichswerte bei Segmentierung der Personen im Austauschverfahren

5.5 Binäre Mischmodelle

# Cluster		ges. Daten	MFC Daten	HFC Daten
Einkäufer	3	~ 0	0.07	0.05
	4	0.33	0.09	0.12
	6	0.24	0.23	0.10
	7	0.50	0.23	0.21
Einkäufe	6	0.34	0.24	0.08
	7	0.54	0.26	0.20
Kombination	18	0.19	0.15	0.06
	19	0.62	0.20	0.36
1500 Kunden	3	0.55	0.49	0.40
	4	0.96	0.81	0.74
2001 Kunden	3	0.50		
	4	0.89		
2502 Kunden	3	~ 0		
	4	0.54		
3000 Kunden	3	~ 0	0.30	0.27
	4	0.48	0.43	0.43
3738 Kunden	3	~ 0		
	4	0.49		
5460 Kunden	3	~ 0		
	4	0.44		

Tabelle 5.9: Werte des korrigierten Rand Index zwischen der Lösung mit binären Mischmodellen und den zugrundeliegenden Gruppen

Zum Vergleich sollen auch mit binären Mischmodellen ermittelte Klassifizierungen betrachtet werden, diese wurden mit der R-Funktion `binmix` berechnet. Tabelle 5.9 zeigt die ermittelten Werte des korrigierten Rand Index für den Vergleich zwischen der Lösung mit binären Mischmodellen und den dem synthetischen Datensatz zugrundeliegenden Gruppen sowohl für die Klassifizierung des gesamten Datensatzes (oberer Teil) wie auch für die Betrachtung gruppierter Daten (unterer Teil). Die Unterschiede zu den entsprechenden Vergleichswerten der mittels Clustern gefundenen Klassifizierungen sind dabei eher gering. Zum Teil gibt es bei den mit Mischmodellen ermittelten Lösungen mehr Übereinstimmungen mit den tatsächlichen Gruppen und zum

Teil ist es genau umgekehrt. Meistens liegt der Grad der Übereinstimmung jedoch in einer ähnlichen Größenordnung.

5.6 Zusammenfassung

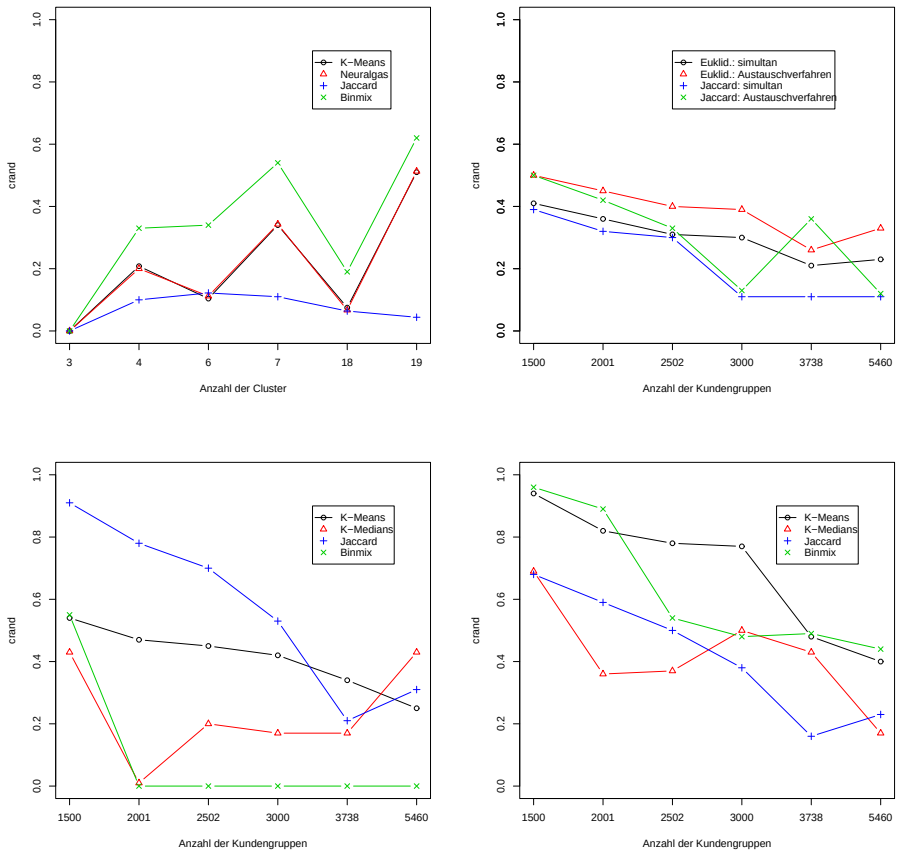

Abbildung 5.4: Vergleich des *Corrected Rand Indexes* für die unterschiedlichen Vorgehensweisen zur Segmentierung des synthetischen Warenkorbes

Abschließend für die Analyse des synthetischen Warenkorbes stellt Abbildung 5.4 die Werte des korrigierten Rand Index graphisch dar. Die Plots zeigen die Verläufe des Indexes für unterschiedliche Clusteralgorithmen und Distanzmaße um aufzuzeigen, welche Vorgehensweisen in welchen Situationen überlegen sind.

Der erste Plot (links oben) zeigt den Verlauf für die Partitionierung der Gesamtdaten ohne Berücksichtigung der Gruppen. Die Werte stammen aus Tabelle 5.1 und 5.9. Man erkennt, dass die Ergebnisse von binären Mischmodellen hier näher an der tatsächlichen Lösung liegen als jene von reinem Clustern. Außerdem zeigt die Graphik, dass die Übereinstimmung abgesehen von der Partitionierung mit Jaccard Distanzmaß deutlich höher ist, wenn man für das Rauschen einen eigenen Cluster zulässt, was bei 4, 7 und 19 Clustern der Fall ist.

Der zweite Plot vergleicht die Grade der Übereinstimmung bei Partitionierung mit simultanem Clustern der Gruppen und im Austauschverfahren (Tabelle 5.7 und 5.8) mit euklidischem und Jaccard Distanzmaß. Die Ergebnisse des Clusterns im Austauschverfahren wiesen dabei eine etwas höhere Übereinstimmung mit den zugrundeliegenden Gruppen auf. Da hier Lösungen mit 4 Clustern betrachtet werden und das Rauschen mit euklidischer Distanz besser identifiziert werden kann, liegen die entsprechenden Werte des korrigierten Rand Indexes tendenziell höher.

Der dritte und vierte Plot zeigen die Werte für die Segmentierung der a priori gruppierten Daten (Tabelle 5.4, 5.5 und 5.9). Die linke Graphik stellt den Verlauf des Vergleichs ohne Rauschen für Lösungen mit 3 Clustern dar. Dabei liefern Ergebnisse, die mit Hilfe des Jaccard Distanzmaßes ermittelt wurden, die größten Übereinstimmungen mit den zugrundeliegenden Gruppen. Die rechte Graphik zeigt die Werte des korrigierten Rand Indexes für Lösungen mit 4 Clustern. In diesem Fall ist durchschnittlich wieder das verallgemeinerte K-Means mit euklidischer Distanz überlegen.

Kapitel 6

Binäre Umfragedaten zum Einkaufsverhalten

6.1 Der Datensatz

Die in diesem Kapitel benutzten Daten entstammen dem Datensatz, den die GfK Nürnberg als Unterstichprobe des ConsumerScan Haushaltspanels von 1995 ZUMA zur Verfügung gestellt hat. Dieser ZUMA-Datensatz enthält alle Haushalte, für die im Jahr 1995 durchgehend Kaufdaten gesammelt worden sind. Für eine nähere Beschreibung der Verbraucherpaneldaten siehe Papastefanou et al. (1995).

Der Datensatz enthält Information über die Einkäufe von ingesamt 40.000 Haushalten über ein Jahr. Durchschnittlich existieren für jeden Haushalt 100 Einkäufe, wobei die konkrete Anzahl zwischen 11 und 360 Einkäufen liegt. Geht man davon aus, dass die Personen eines Haushaltes tatsächlich alle Einkäufe notiert haben, bedeutet das, dass die Spannbreite von Personen, die nur einmal pro Monat einkaufen, bis zu solchen, die täglich einkaufen, reicht.

Die einzelnen Einkäufe sind kodiert als binäre Variablen und geben über 65 Produktgruppen Auskunft, ob diese gekauft oder nicht gekauft wurden. Zur Vereinfachung der Interpretation werden diese Gruppen nach aufsteigender Häufigkeit ihres Auftretens sortiert.

Zur genaueren Analyse werden zunächst nur die ersten 1800 Haushalte in drei Gruppen zu je 600 Haushalten herangezogen. Abbildung 6.1 zeigt in der ersten Zeile die Barplots der jeweils von 600 Haushalten gewählten Produktgruppen. Wie man unschwer erkennen kann, weisen die Haushalte der

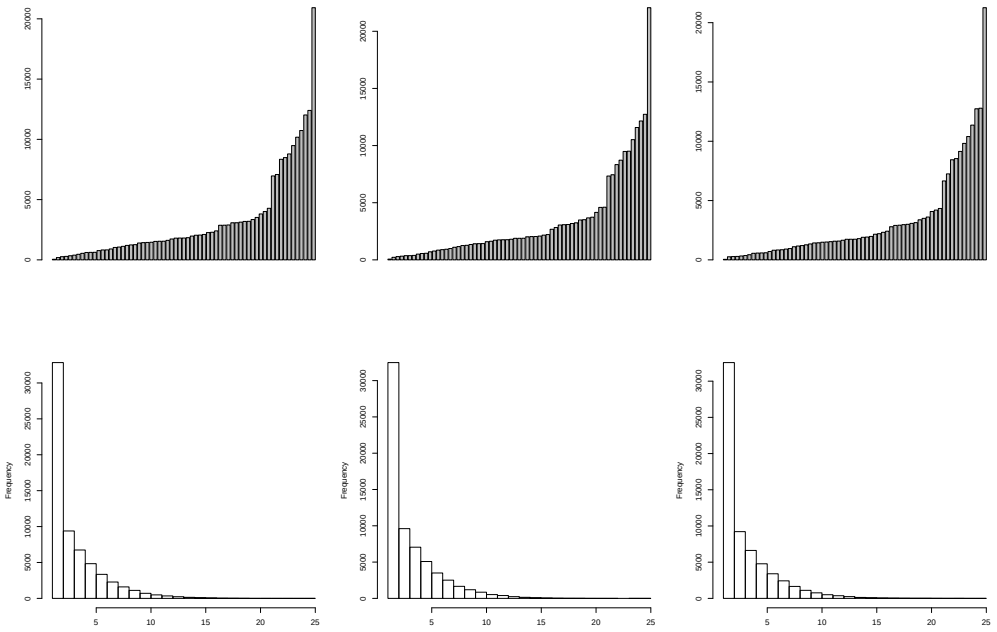

Abbildung 6.1: 1.Zeile: Barplots der Einkäufe; 2.Zeile: Größenverteilung der Warenkörbe

drei Teile ein sehr ähnliches Verhalten auf. Mit Abstand am häufigsten konsumiert wird die Produktgruppe 15, die Frisch- und Haltbarmilch enthält und in knapp 1/3 aller Warenkörbe enthalten ist. Bei genauerer Betrachtung zeigt sich, dass bestimmte Haushalte jedesmal Milch kaufen, andere Haushalte nur sehr wenig beziehungsweise selten Milch benötigen und wieder andere Haushalte in dem betrachteten Jahr nur einmal Milch gekauft haben.

Danach folgt die Produktgruppe 50 mit Weich-, Frisch- und Schmelzkäse und die Gruppe 58, die Joghurt enthält. Diese beiden sind in zirka 19% der Warenkörbe enthalten. Das Schlusslicht bildet die Gruppe 2 mit Teppich- und Polsterreinigern, die nur bei 0.07% der Einkäufe benötigt wurde. Während die Mehrzahl der Haushalte diese Produkte nicht kauft, haben sie andere im Lauf des Jahres durchschnittlich bis zu 3 Mal benötigt. Ein Haushalt gab sogar 15 Mal an Teppich- oder Polsterreiniger gekauft zu haben. Die Daten wurden von derartigen Ausreißern bisher nicht bereinigt.

In der zweiten Zeile zeigt Abbildung 6.1 die Verteilung der Größen der einzelnen Einkäufe in Form von Histogrammen über die Anzahl der voneinander verschiedenen gewählten Produktgruppen pro Einkauf für jeweils 600 Haus-

halte. In knapp 1/3 aller Warenkörbe ist genau eine Produktgruppe enthalten, ca. 16% enthalten mehr als 5 verschiedene Produkte und nur ca. 2,3% mehr als 10. Bei detaillierter Betrachtung fällt zum Beispiel noch auf, dass ca 7% der Warenkörbe nur Milch enthalten.

6.2 Segmentierung der Daten

Die drei Teile, die jeweils 600 Haushalte und damit durchschnittlich 64600 Einkäufe enthalten, werden getrennt geclustert. Dabei wird die initiale Partitionierung zur Segmentierung des ersten Teils zufällig gewählt; an die Segmentierung des zweiten und dritten Teils werden jeweils die am Ende ermittelten Zentren der Partitionierung des vorigen Teils zur Initialisierung weitergegeben. Durch mehrmalige Wiederholung dieses Vorgehens wird versucht ein möglichst „gutes“ lokales Optimum als Lösung zu finden, wobei als Kriterium wieder die Summe der Distanzen innerhalb der Cluster herangezogen wird. Abbildung 6.2 zeigt den Verlauf dieser minimal erreichten summierten Distanzen für 2 bis 15 Klassen bei Segmentierung der „reinen“ Daten (dh. ohne Berücksichtigung der Haushalte) aufgeteilt nach den drei Teilen. Zur Berechnung wurde die R-Funktion `kcca` mit `kccaFamily("kmeans")` und `kccaFamily("ejaccard")` genutzt. Für die Segmentierung mit euklidischer Distanz (links) liegen die Distanzen zwischen minimal zirka 99000 für 2 Cluster und minimal zirka 81000 für 15 Cluster. Unter Benützung des Jaccard Distanzmaßes (rechts) liegen die Werte zwischen zirka 59000 und 50000.

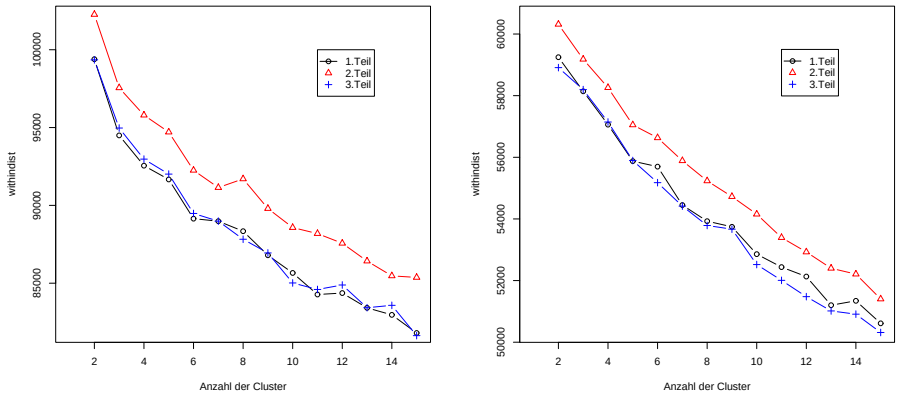

Abbildung 6.2: Distanzen innerhalb jedes Clusters mit euklidischer und Jaccard Distanz

Wie schon bei den synthetischen Warenkorbdaten liefert dieser Plot kaum Hinweise auf eine zu bevorzugende Anzahl an Clustern. Vielmehr zeigt sich, dass der Datensatz keine „echte“ Gruppenstruktur enthält, die durch die Partitionierung identifiziert werden kann. Dennoch kann ein Clusteralgorithmus möglichst homogene Gruppen finden und damit den Daten künstlich eine Gruppenstruktur geben, die das Verständnis der Daten verbessert. Im Marketing-Bereich sind diese künstlichen Gruppen sehr hilfreich. Ähnliche Einkäufe und im Folgenden Haushalte mit ähnlichem Einkaufsverhalten werden zu Clustern zusammengefasst und können gemeinsam zielgerichtet adressiert werden. Abbildung A.7 und A.8 im Anhang zeigen die Struktur der Cluster bei einer Einteilung der Einkäufe in 10 Gruppen nach euklidischer und Jaccard Distanz.

6.3 Segmentierung der a priori zu Haushalten gruppierten Daten

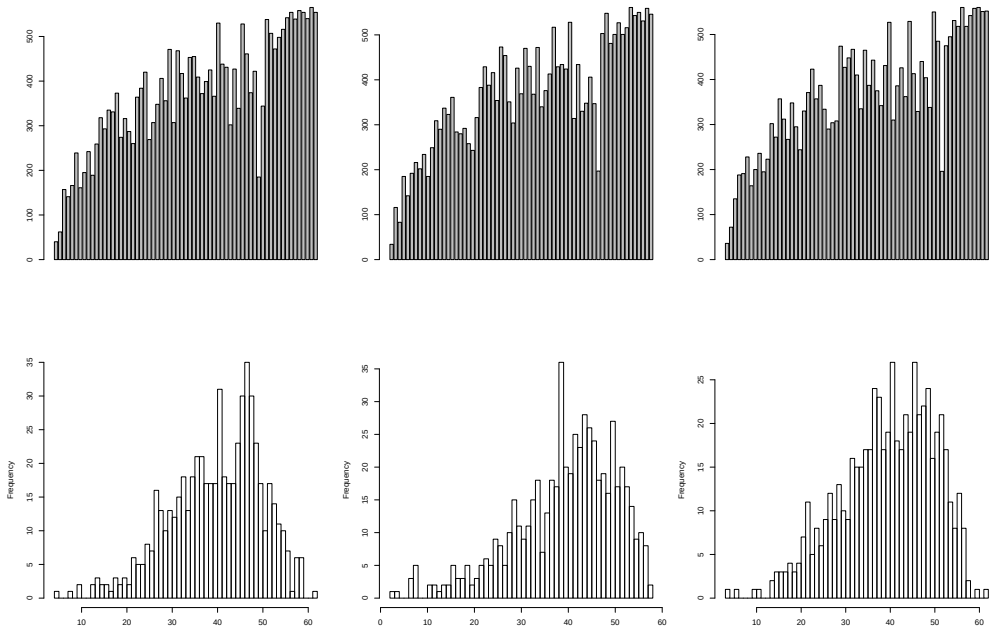

Abbildung 6.3: 1.Zeile: Barplots der gewählten Produktgruppen der gruppierten Daten; 2.Zeile: Histogramme über die Gesamtzahlen gewählter Produkte

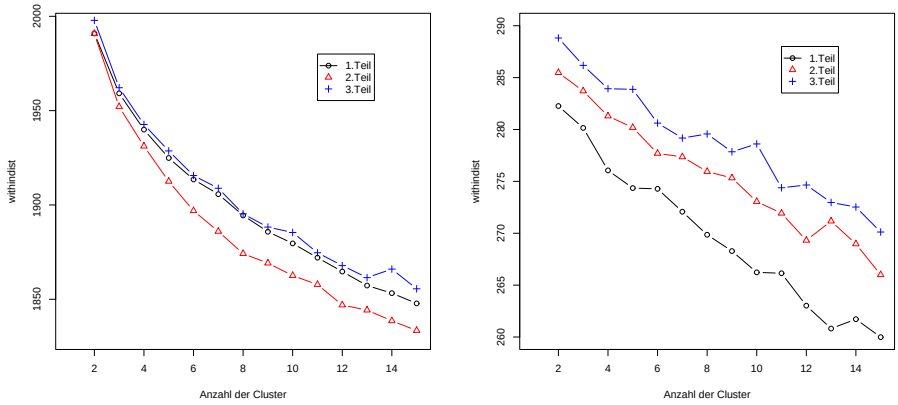

Abbildung 6.4: „Withindist“ bei Segmentierung der gruppierten Daten mit euklidischer und Jaccard- Distanz

Wie in Abschnitt 3.2 beschrieben werden die Daten zu Haushalten gruppiert, indem Vektoren gebildet werden, in denen eine 1 bei jeder Produktgruppe steht, die der betrachtete Haushalt zumindest einmal gekauft hat. Analog zu Abbildung 6.1 zeigt Abbildung 6.3 in der ersten Zeile Barplots der gewählten Produkte der gruppierten Daten. Diese Daten zeigen für jeden Haushalt, ob eine bestimmte Produktgruppe im Lauf des betrachteten Zeitraums gewählt wurde oder nicht; wie oft sie gekauft wurde, ist dabei jedoch irrelevant. In der zweiten Zeile zeigt Abbildung 6.3 die Verteilung der von den Haushalten insgesamt gewählten Produktgruppen, das heißt, sie gibt Auskunft über die Anzahl verschiedener Produkte, die ein Haushalt im Lauf des betrachteten Jahres gewählt hat.

Abbildung 6.4 zeigt den Verlauf der Summen der Distanzen innerhalb der Cluster für die Segmentierung der zu dreimal 600 Haushalten gruppierten Daten mit euklidischer Distanz (links) und mit dem Jaccard-Distanzmaß (rechts). Die Summen sind aufgrund der wenigen Datenpunkte deutlich niedriger als bei den in anderen Abschnitten betrachteten Lösungen. Da durch die Gruppierung eine Datenmatrix entsteht, die deutlich dichter ist als die ursprünglichen Daten, eignet sich die euklidische Distanz sehr gut. Das Verfahren konvergiert rasch und es ergeben sich adressierbare homogene Gruppen ähnlicher Größe. Abbildung A.9 zeigt eine Partitionierung mit euklidischer Distanz.

Die Segmentierung mit dem Jaccard Distanzmaß ergibt stets einen großen Cluster, der $80 - 90\%$ der Haushalte umfasst, und abhängig von der Gesamtzahl einige kleine Cluster mit 5-10 Haushalten, die ein sehr ähnliches

Verhalten aufweisen. Abbildung A.10 zeigt die Barplots der Zentren einer solchen Partitionierung, die für ein zielgerichtetes Marketing vermutlich kaum nützlich ist.

6.4 Simultanes Clustern der Haushalte

Zur Segmentierung der Haushalte ausgehend von den Gesamtdaten aller ihrer Einkäufe wird zunächst die in Abschnitt 3.3 und 5.3 beschriebene Vorgehensweise herangezogen. Die Einkäufe eines Haushaltes entsprechen den Objekten, die demselben Cluster angehören müssen. Die Clusterzugehörigkeit wird aufgrund der Distanzen der Einzeleinkäufe zu den Clusterzentren für jedes Objekt bestimmt und die Haushalte werden dann nach Mehrheitsabstimmung zugeordnet. Wie bei der Segmentierung ohne Bedingungen durch die Haushalte wird die erste Zuordnung des ersten Teils der Daten zufällig gewählt und die weiteren beiden Teile werden mit den aus der vorgehenden Segmentierung bekannten Zentren initialisiert. Die folgenden Ergebnisse und Analysen (Barplots, Visualiserung, etc.) werden, sofern nicht extra erwähnt, anhand des dritten Datenteils mit 600 Haushalten erstellt.

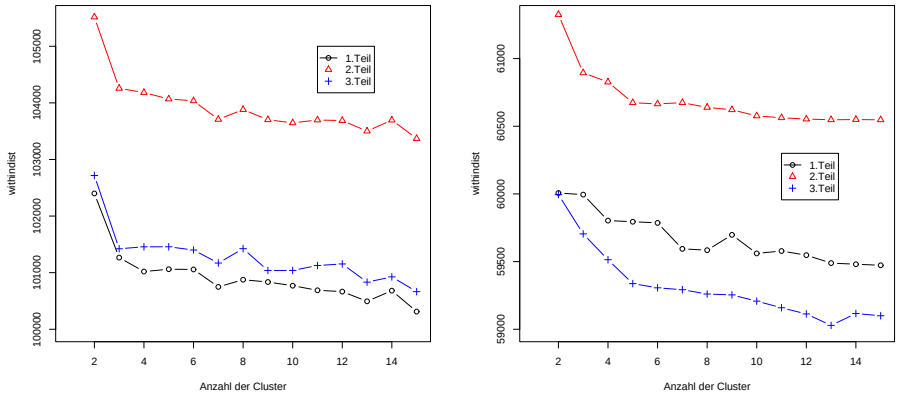

Abbildung 6.5: „Withindist“ bei Segmentierung mit simultanem Clustern der Haushalte mit euklidischer und Jaccard Distanz

Abbildung 6.5 zeigt den Verlauf der Summen der Distanzen der Punkte zu den Clusterzentren nach Konvergenz des modifizierten K-Means Algorithmus zu einem lokalen Optimum für die Initialiserung mit 2 bis 15 Zentren in euklidischer Distanz (links) und mit Jaccard-Distanzmaß (rechts). Die

tatsächlich erreichte Zuteilung besteht häufig aus deutlich weniger Clustern als die anfängliche Initialisierung.

Bei euklidischer Distanz liegen diese Summen zwischen minimal 102400 für 2 Cluster und minimal 100300 für 15 Cluster, bei der Segmentierung mit Jaccard Distanzmaß zwischen 60000 und 59100. Im Vergleich zu den Summen der Distanzen bei der Segmentierung ohne Gruppenindex in Abbildung 6.2 liegen diese Werte ein wenig höher, da die Segmentierung mit Gruppen durch die Bedingungen eingeschränkt ist. Auffällig ist auch, dass die Werte bei Segmentierung ohne Gruppenindex mit zunehmender Clusterzahl mehr abnehmen.

Abbildung A.11 zeigt einen Barplot mit 10 Clustern, die mittels des zur simultanen Segmentierung der Haushalte modifizierten K-Means Algorithmus und der Jaccard Distanz bestimmt wurden. Auch hier bildete sich ein deutlich größerer Cluster (3), der zirka 47% aller Einkäufe, jedoch nur 42% der Haushalte enthält. Er beinhaltet jene Haushalte, die vorwiegend kleine Einkäufe machen und die kein besonders typisches Verhalten aufweisen, das sie vom Rest unterscheiden würde. Die Haushalte, die in einem der kleinen Cluster zusammengefasst sind, zeigen meistens ein ähnliches Einkaufsverhalten. Anhand der herausragenden Spitzen in den Barplots erkennt man zum Beispiel deutlich, dass die Personen der Haushalte in Cluster 7 gern mehr Joghurt konsumieren als andere und jene in Cluster 8 deutlich mehr Mineralwasser kaufen als andere Produkte und auch als der Durchschnitt der anderen Personen. Vor einer genaueren Analyse dieser Lösung soll aber noch ein weiteres Verfahren zur Segmentierung des Datensatzes betrachtet werden.

6.5 Clustern der Haushalte im Austauschverfahren

Als alternative Methode zur Segmentierung wird auch die in Abschnitt 3.4 und 5.4 beschriebene Austauschvariante des K-Means Algorithmus, bei der in jeder Iteration ein Haushalt neu zugeordnet wird, auf die Daten angewandt. Dabei werden zur Gruppierung der ersten 600 Haushalte diese zunächst zufällig in Cluster geteilt. Anschließend werden wie beschrieben die Haushalte einzeln neuen Zentren zugeteilt und die Zentren aktualisiert, bis sich die Zuordnung nicht mehr verändert. Ausgehend von diesen Zentren wird der zweite Datenteil geclustert und von diesem Ergebnis ausgehend der dritte.

Vergleicht man die Homogenität der Cluster unterschiedlicher erzielter Lösungen, ergibt sich der in Abbildung 6.6 gezeigte Verlauf der Distanzen inner-

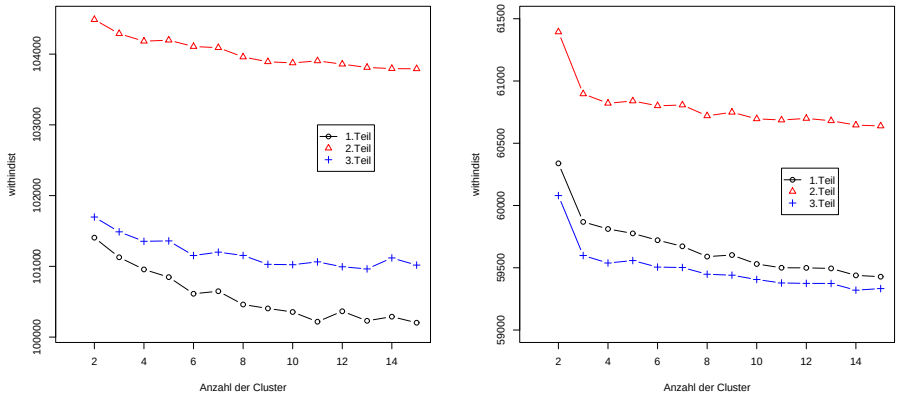

Abbildung 6.6: „Withindist“ bei Segmentierung mit der Variante des K-Means nach Hartigan

halb von 2 bis 15 Clustern bei Segmentierung mit euklidischer Distanz (links) und mit Jaccard-Distanzmaß (rechts). Die minimal erreichten Summen der Distanzen liegen dabei in derselben Größenordnung wie in Abbildung 6.5. Auch der gesamte Verlauf der „Withindist“ der beiden Vorgehensweisen zur Segmentierung vor allem mit Jaccard Distanzmaß verhält sich ähnlich, was vermuten lässt, dass auch die Partitionierungen selbst einander ähnlich sind. Abbildung A.13 zeigt die Barplots zur Segmentierung der Haushalte im Austauschverfahren mit Jaccard Distanzmaß für 10 Cluster. Auch hier enthält ein Cluster (7) fast 50% der Einkäufe. Vergleicht man diese Barplots mit jenen in Abbildung A.11 stellt man zum Beispiel fest, dass nicht nur Cluster 3 in der ersten Lösung und Cluster 7 in der zweiten Lösung einander sehr ähnlich sind. Auch Cluster 2 und 4 sowie Cluster 8 und 10 enthalten weitgehend diesselben Einkäufe. Ein systematischer Vergleich der beiden Lösungen ergibt einen korrigierten Rand Index von 0.60. Zum Vergleich zeigt Abbildung A.12 Barplots zur Segmentierung der Haushalte im Austauschverfahren mit euklidischer Distanz.

6.6 Visualisierung und Interpretation

Die Clusteranalyse projeziert Daten aus einem hochdimensionalen Raum auf eine nominale Variable, die Clusterzugehörigkeit. Ebenso hilft eine Visualisierung durch Projektion auf zwei Dimensionen hochdimensionale Daten besser zu verstehen und geht damit Hand in Hand mit der Clusteranalyse.

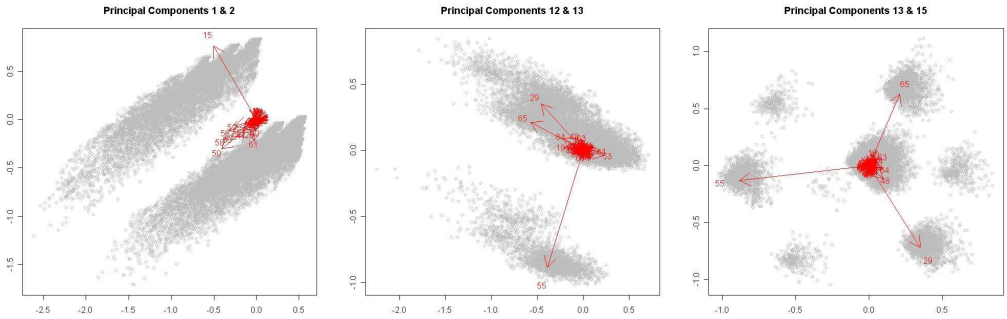

Abbildung 6.7: Hauptkomponentenanalyse der Daten

Abbildung 6.7 zeigt eine Hauptkomponentenanalyse der Daten. Die linke Graphik zeigt einen Scatterplot aller Datenpunkte projeziert auf die erste und zweite Hauptkomponente. Diese teilen die Datenpunkte nach Faktor 15 (Frisch- und Haltbarmilch) klar in 2 Teile. Der mittlere Plot zeigt, dass bei einer Projektion auf die 12. und 13. Hauptkomponente eine Teilung nach Faktor 55 (Tiernahrung, Katzenstreu) entsteht. Die rechte Graphik zeigt eine Projekion auf die 13. und 15. Hauptkomponente. Abbildung B.1 zeigt Barplots der 2. bis 16. Hauptkomponente. Der erste dieser Plots zeigt deutlich, dass die 2. Hauptkomponente mit der häufigsten Produktgruppe Milch zusammenhängt. Auch die weiteren Hauptkomponenten hängen häufig direkt mit einem Faktor zusammen, manche werden auch von zwei oder mehreren Faktoren beeinflusst.

Die Hauptkomponentenanalyse veranschaulicht, dass in den Daten Faktoren existieren, die für manche der zahlreichen Eigenschaften der Gesamtdaten bestimmend sind. Anhand dieser Faktoren sollte es der Clusteranalyse möglich sein Gruppen zu identifizieren, deren Verhalten sich vom Rest der Daten unterscheidet. Färbt man jedoch einfach die Punkte in den Plots von Abbildung 6.7 nach ihrer Clusterzugehörigkeit ein, erkennt man die Struktur der Cluster kaum, da diese zur sehr überlappen und die einzelnen Cluster optisch nicht erkennbar sind. Auch gewöhnliche Nachbarschaftsgraphen geben möglicherweise die Form der Cluster, die nicht elliptisch sein muss, nicht richtig wieder. Zur Visualisierung der Cluster in der Projektion sollen diese daher mit konvexen Hüllen dargestellt werden, wobei sich der Begriff konvex auf das jeweilige Distanzmaß bezieht.

Abbildung 6.8 zeigt anhand der oberhalb betrachteten Hauptkomponenten die konvexen Hüllen der Segmentierung aus Abschnitt 6.4 (Abbildung A.11), wobei jede Farbe einem Cluster entspricht. Die Ordnung der Punkte eines

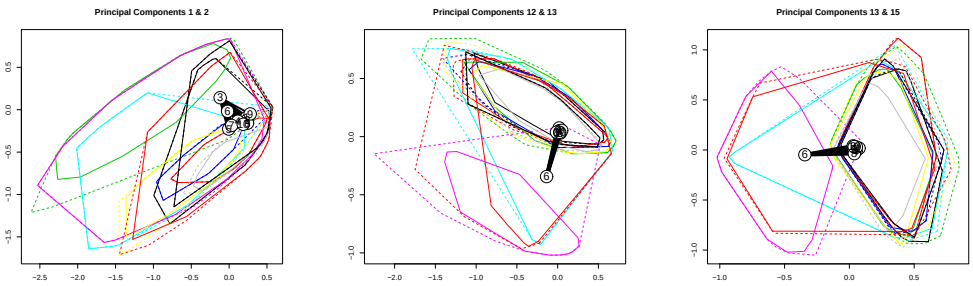

Abbildung 6.8: Darstellung der Cluster mit konvexen Hüllen

Clusters erfolgt anhand der Distanzen der Punkte zu ihren Clusterzentren $d(\mathbf{x}, c(\mathbf{x}))$.

$$m_k = \text{median}\{d(\mathbf{x}_n, \mathbf{c}_k) | \mathbf{x}_n \in C_k\}$$

sei der Median der Distanzen aller Punkte in Cluster k zum Zentrum $\mathbf{c}_k$. Analog zur Box eines Boxplots soll der innere Bereich der Cluster visualisert und durch $d(\mathbf{x}_n, \mathbf{c}_k) \leq m_k$ beschränkt werden. Er ist in der Graphik mit einer durchgehenden Linie markiert. Der äußere Bereich wird definiert durch alle Punkte, die nicht weiter als $2.5m_k$ von $\mathbf{c}_k$ entfernt sind, und wird durch eine gestrichelte Linie gekennzeichnet. Punkte außerhalb dieses Bereichs werden als Ausreißer betrachtet (siehe Leisch, 2005a).

Cluster 3 (grün), der fast 50% aller Einkäufe umfasst, enthält vorwiegend Haushalte mit hohem Milchkonsum. Die Personen in diesem Cluster weisen keine besonders typischen Eigenschaften auf, die sie von der Mehrheit unterscheiden würden. Sie entsprechen in etwa dem Durchschnitt der Einkäufer, bilden damit den mit Abstand größten Cluster, sind aber schwer durch eine bestimmte Eigenschaft von den anderen zu differenzieren.

Die mittlere und die rechte Graphik zeigen, dass bei richtiger Projektion der 6. Cluster klar von den anderen getrennt werden kann. Bereits der Barplot machte sichtbar, dass dieser Cluster Haushalte enthält, die häufig Tierprodukte kaufen, was von der Darstellung der Hauptkomponentenanalyse bestätigt wird. Knapp 10% der Haushalte können damit gut von den anderen differenziert werden und können gegebenenfalls zum Beispiel gezielt beworben werden.

Ebenfalls fast 10% der Haushalte werden Cluster 5 (in Abbildung 6.8 hellblau dargestellt) zugeordnet. Er fällt in der Projektion durch die Darstellung auf, kann jedoch nicht deutlich von den anderen Clustern getrennt werden.

Diese Kunden kaufen verhältnismäßig viel Käse (sowohl die Produktgruppe Hartkäse als auch Weichkäse), jedoch nicht so sehr, dass man dies auch an der Visualisierung erkennen könnte. Charakteristisch für sie ist vor allem, dass sie wenig Milch kaufen, sich aber sonst eher durchschnittlich verhalten, das heißt, sie fallen nicht durch eine Produktgruppe auf, die sie deutlich mehr als alle anderen benötigen.

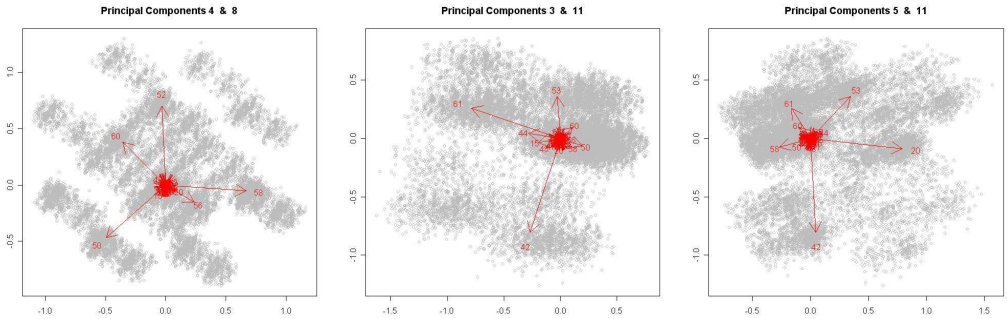

Abbildung 6.9: Weitere Projektionen der Daten

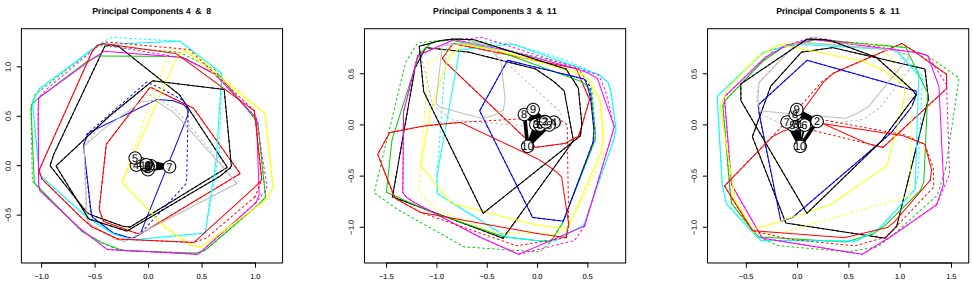

Abbildung 6.10: Darstellung der Cluster mit konvexen Hüllen

Der viergrößte Cluster ist der 7., der bereits in Abschnitt 6.4 angesprochen wurde und in der Visualisierung gelb dargestellt wird. Ihm werden alle Kunden zugeteilt, die häufig Produktgruppe 58 (Joghurt) konsumieren. Die Projektion auf die 4. und 8. Hauptkomponente (rechts in Abbildung 6.9 und 6.10) zeigt, dass die Konsumation dieser Produkte sich leicht gegenläufig verhält zur Konsumation anderer Milchprodukte wie den Gruppen 50, 52 und 60. Dadurch kann der 7. Cluster von den anderen zumindest teilweise getrennt werden, was auch die Darstellung der konvexen Hüllen bestätigt.

Der Größe nach folgen nun mit einem Anteil von jweils gut 6% die Cluster 2

und 10 , die rot dargestellt sind. Kunden in Cluster 2 kaufen bevorzugt Bohnenkaffee (Röstware, Produktgruppe 20) und dazu Kaffeemilch (53), während die Kunden in Cluster 10 diejenigen sind, die oft Bier kaufen. Wie die Projektionen mit Hilfe der 11. Hauptkomponente zeigen, verhalten sich diese Faktoren stark gegenläufig und die beiden Cluster können voneinander getrennt dargestellt werden.

Die verbleibenden Cluster 1, 4 und 9 können aufgrund ihrer geringen Größe kaum visualisiert werden, sie sind in den Plots dunkelblau und schwarz dargestellt. Der kleinste Cluster (9) enthält Haushalte, die häufig Tiefkühlprodukte und vor allem vergleichsweise viel Tiefkühl- Fisch konsumieren. Da dieser aber insgesamt sehr wenig konsumiert wird, erfolgt einen Trennung nach dem entsprechenden Faktor erst durch die 28. Hauptkomponente und selbst dort nicht besonders klar. Kunden in Cluster 1 kaufen vermehrt Limonaden und fruchhaltige Getränke, dafür aber wenig Bier. Kunden in Cluster 4 konsumieren viel Käse, jedoch im Vergleich zu Cluster 5 vorwiegend Weichkäse und wenig Hartkäse.

6.7 Soziodemographische Daten

Zu den Haushalten des Umfrage- Datensatzes stehen auch soziodemographische Daten zur Verfügung, die mit der betrachteten Clusterlösung (Abbildung A.11) verglichen werden sollen. Dazu werden die 7 größten Cluster herangezogen, Cluster, die weniger als 3% der Warenkörbe enthalten, werden nicht mehr eingehender betrachtet.

Abbildung 6.11 gibt einen Überblick über die Verteilung der Einwohnerzahlen der Orte, in denen die befragten Personen wohnen, in Form eines Mosaikplots. Darin gibt die Spaltenbreite Auskunft über die Größe der Cluster, während die Zeilenhöhe Auskunft gibt über die Anzahl der Haushalte in der entsprechenden Kategorie. Je größer die Rechtecke ausfallen umso mehr Haushalte liegen im entsprechenden Cluster und in dieser Kategorie. Eine Färbung der Rechtecke bedeutet, dass diese signifikant größer (blau) oder kleiner (rot) als der Durchschnitt sind. Betrachtet man zunächst Cluster 2 gibt es keine signifikanten Zusammenhänge zwischen der Clusterzuteilung und der Ortsgröße. Es fällt jedoch auf, dass diese Haushalte tendenziell eher in Großstädten liegen und dass ebenso verhältnismäßig viele in Dörfern mit unter 2000 Einwohnern leben. Anhand der nächsten Abbildung 6.12 erfährt man, dass viele dieser Personen über 70 Jahre alt sind. Abbildung 6.13 und 6.14 zeigen, dass die Personen in Cluster 2 deutlich traditionsbewusster leben als andere und dass sie eine traditionell bürgerliche Küche bevorzugen.

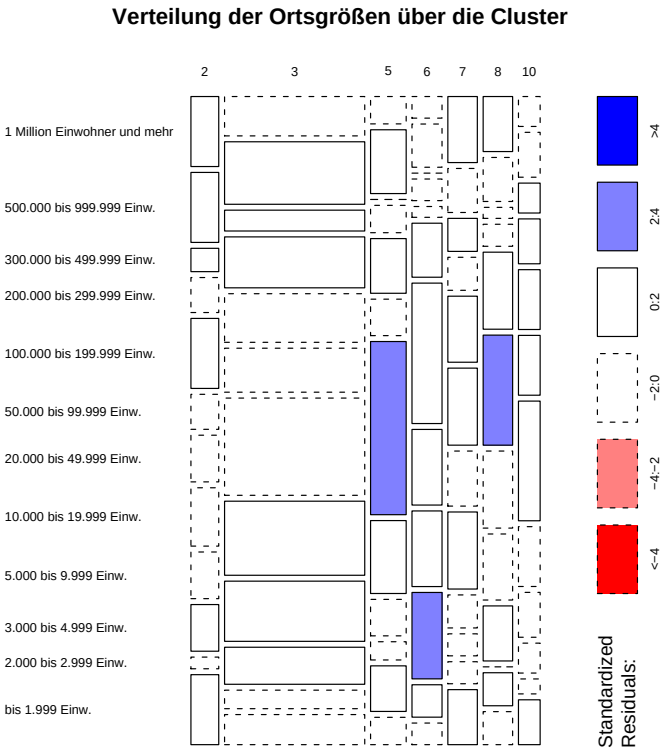

Abbildung 6.11: Verteilung der Ortsgrößen

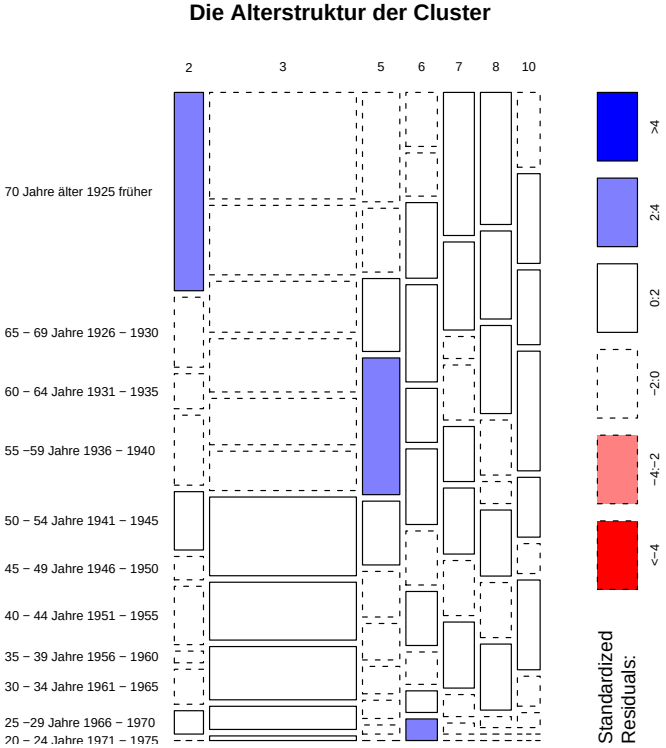

Abbildung 6.12: Die Altersstruktur der Cluster

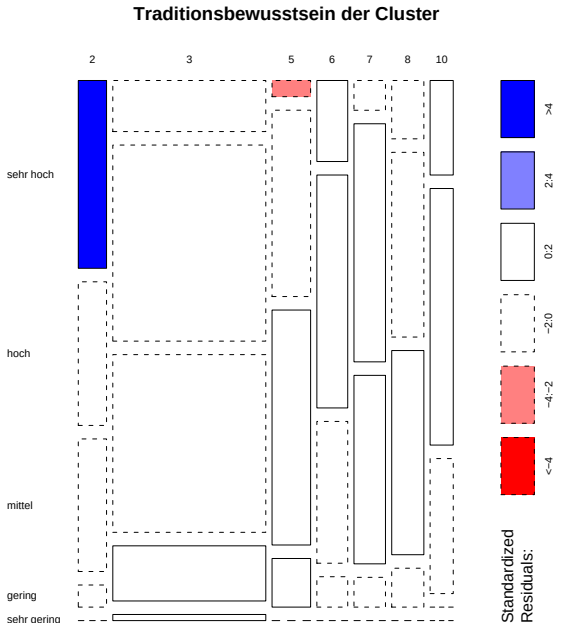

Abbildung 6.13: Verteilung des Traditionsbewusstseins der Haushalte

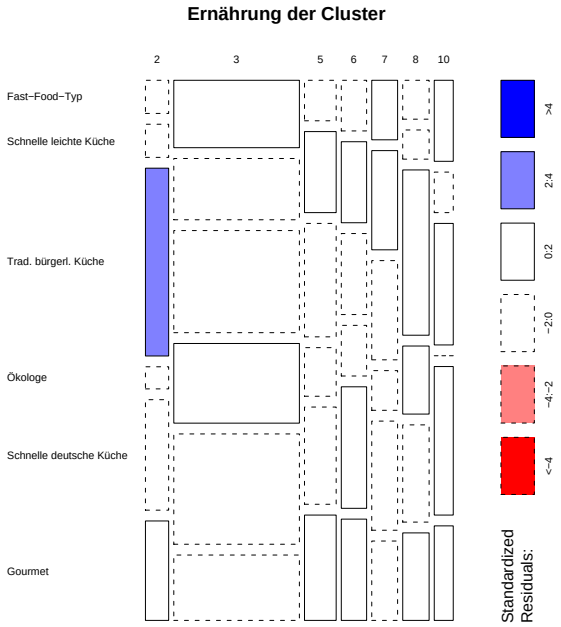

Abbildung 6.14: Bevorzugte Ernährungsweise der Haushalte

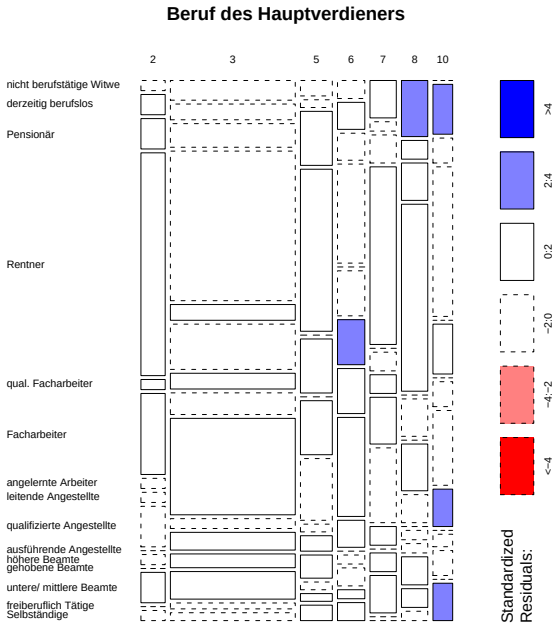

Abbildung 6.15: Verteilung der Berufsgruppen über die Cluster

Cluster 3 mit 252 der 600 betrachteten Haushalte verhält sich erwartungsgemäß wenig vom Durchschnitt abweichend. Die Darstellung der soziodemographischen Daten dieser Haushalte ist dem Durchschnitt der anderen betrachteten Haushalte stets ähnlich.

In Cluster 5 sind die meisten Haushalte Familien, deren Haushaltsvorsteher zwischen 50 und 65 Jahren alt ist. Sie wohnen häufig in Kleinstädten und legen verstärkt Wert auf eine gute Ernährung, was sich auch in den Produkten, die sie wählen, und einem geringeren Preis- und Traditionsbewusstsein widerspiegelt.

Interessant sind die soziodemographischen Daten der Haushalte in Cluster 6, die mit großem Abstand am meisten Tiernahrung kaufen. Sie wohnen eher in kleinen Orten und die haushaltsführenden Personen sind im Vergleich zum Durchschnitt eher jung. Meistens handelt es sich dabei um Familien mit 2-3 Kindern unter 18 Jahren. Man erkennt anhand weiterer soziodemographischer Daten, dass diese Personen eher preisbewusst leben und Markenartikel nur selten bevorzugen. In Cluster 6 befinden sich 25% der Haushalte, die ein Tier haben, wobei diese Haushalte großteils Katzen besitzen. Haushalte mit kleineren Tieren wie Hamster, Vögel oder Fische werden eher nicht diesem

Cluster zugeordnet. Auch dass nur $1,7\%$ derer, die kein Haustier besitzen, in Cluster 6 sind, bestätigt, dass dieser Cluster sinnvoll gewählt wurde.

Die haushaltsführenden Personen in Cluster 7 und 8 sind tendenziell eher älter, jedoch nicht auffällig traditionsbewusst. Obwohl viele Haushalte aus Cluster 8 angeben traditionell bürgerlich zu kochen, erkennt man auch hier keinen signifikanten Unterschied. Cluster 8 enthält viele Witwen, Pensionisten und Rentner. Auffälliger verhalten sich die Daten der Haushalte in Cluster 10. Die haushaltsführenden Personen sind jünger als der Durchschnitt und es gibt kaum Kinder und sehr selten Haustiere in diesen Haushalten. Beruflich handelt es sich hier um viele Selbstständige, Berufslose und qualifizierte Angestellte.

6.8 Trennung der Groß- und Kleineinkäufe

Bei der Simulation des synthetischen Warenkorbes wurden unterschiedlich große Einkäufe nach bestimmten Typen generiert um festzustellen, ob diese bei der Partitionierung wiedergefunden werden können. Die Analyse hat gezeigt, dass die Einkaufstypen anhand der gekauften Produktgruppen und anhand der Größen der Einkäufe ähnlich gut identifiziert werden können.

Anhand der Umfragedaten über das Einkaufsverhalten soll jetzt die Segmentierung gruppierter Daten mit der Beachtung von Groß- und Kleineinkäufen kombiniert werden, indem für jeden Kunden zwei Clusterzuteilungen zugelassen werden. Für die simultane Segmentierung der Gruppen, die hierfür herangezogen werden soll, bedeutet das, dass die Indexmenge mit den Must-link constraints für jeden Haushalt zwei Objekte enthält, die jeweils die Kleineinkäufe und die Großeinkäufe aneinander binden.

Abbildung 6.16 zeigt den zugehörigen Verlauf der Distanzen der Punkte zu ihren Clusterzentren für 2 bis 15 Cluster mit euklidischer (links) und Jaccard (rechts) Distanz. Abbildung A.14 zeigt eine Segmentierung mit der Möglichkeit Haushalte nach Groß- und Kleineinkäufen getrennt zuzuordnen für 10 Cluster mit Jaccard Distanzmaß. Diese soll im Folgenden näher betrachtet werden.

Dazu zeigt Abbildung 6.17 die Darstellung der 10 Cluster mit konvexen Hüllen für die Projektion auf die bereits im voranstehenden Abschnitt betrachteten Hauptkomponenten. Der größte Cluster (8) mit 59% der Warenkörbe enthält den Großteil der Einkäufe mit mehr als 5 Produktgruppen, was man deutlich an der Darstellung der Größen der Einkäufe in Abbildung 6.18 erkennt. Die Boxplots zeigen die Anzahl verschiedener gewählter Pro-

duktgruppen je Einkauf, aber da die Einkäufe der Haushalte in Cluster 8 so eindeutig größer ausfallen als in anderen Clustern, ist bei gleicher Skalierung in den Boxplots der Cluster mit kleinen Einkäufen wenig zu erkennen. Die Zuordnung zu Cluster 8 erfolgt nicht aufgrund der gekauften Produkte, sondern fast ausschließlich aufgrund der Größen der Warenkörbe.

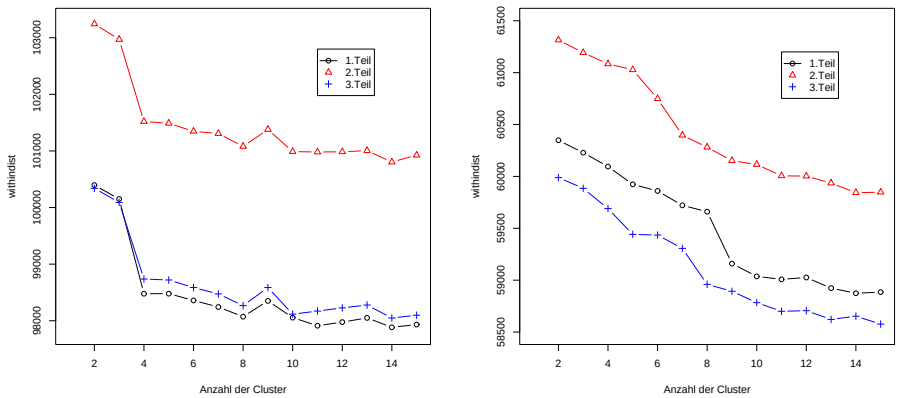

Abbildung 6.16: „Withindist“ bei simultaner Segmentierung der Gruppen mit einer Trennung der Groß- und Kleineinkäufe

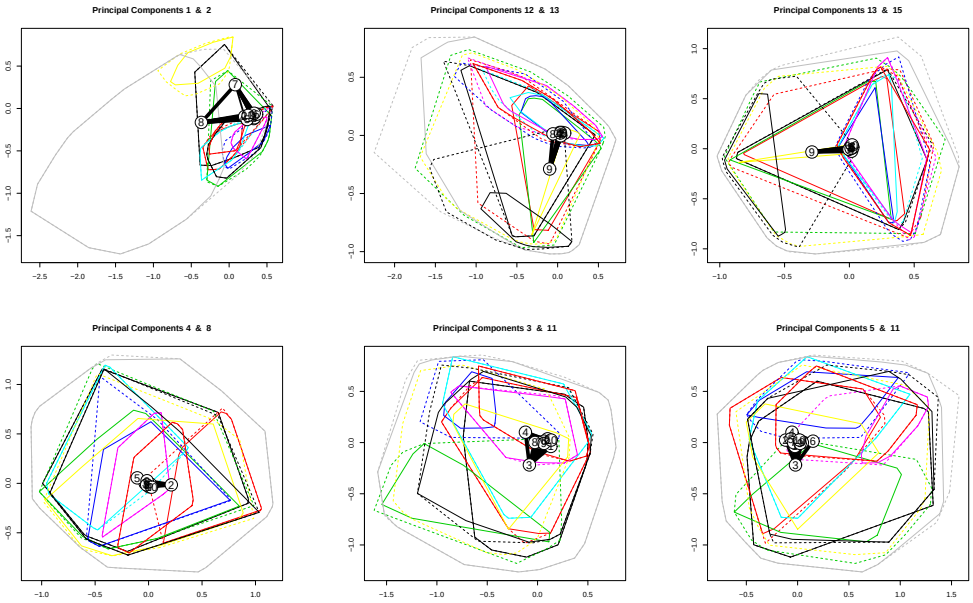

Abbildung 6.17: Darstellung der Cluster mit konvexen Hüllen

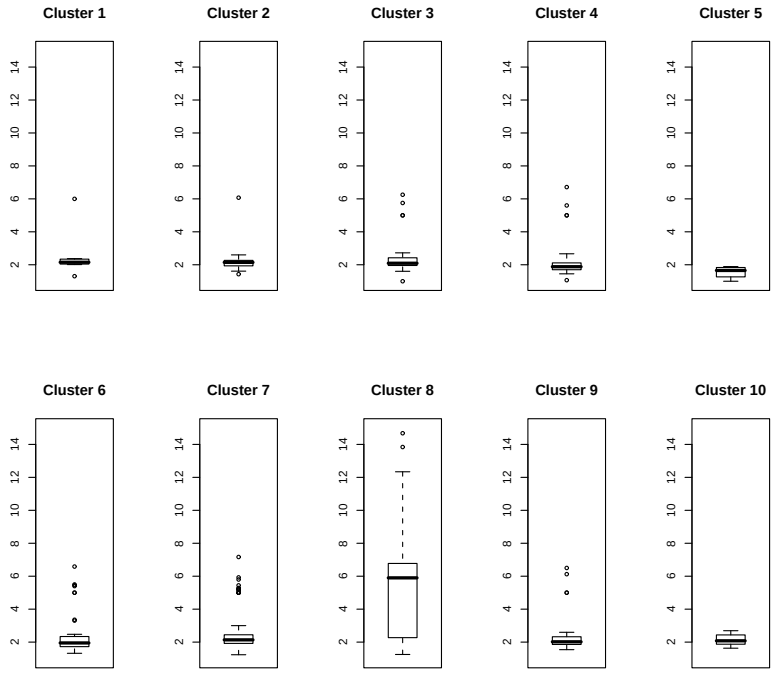

Abbildung 6.18: Boxplots der 10 Cluster über die Anzahl verschiedener Produktgruppen je Warenkorb

Der zweite größere Cluster (7) enthält 20% der Warenkörbe und ist in Abbildung 6.17 gelb dargestellt. Er enthält alle Haushalte, die viel Milch konsumieren und wird damit durch die Projektion auf die 1. und 2. Hauptkomponente von den anderen Clustern getrennt dargestellt.

Die Cluster 3 (grün) und 4 (dunkelblau) enthalten $4 - 5\%$ der Warenkörbe und fallen in den Projektionen auf die 3., 5. und 11. Hauptkomponente (Abbildung 6.17, 2. Zeile mitte und rechts) etwas auf. Der 3. Cluster enthält Haushalte, die viel Bier und verstärkt Mineralwasser konsumieren, während der 4. Cluster Haushalte enthält, die nur Mineralwasser häufig kaufen. Der 6. Cluster (zyklam) wird ebenfalls durch die genannten Projektionen von den anderen Clustern teilweise getrennt. Er enthält Haushalte, die viel Kaffee konsumieren.

Durch eine Projektion auf die 13. Hauptkomponente (Abbildung 6.17, 1. Zeile mitte und rechts) wird der 9.Cluster (schwarz) von den anderen getrennt. In diesem Cluster finden sich wieder jene Haushalte, die Tierprodukte kaufen.

Auch den Joghurt-Konsumenten wird wieder ein eigener Cluster zugewiesen, er hat die Nummer 2 und ist rot dargestellt, wobei er vor allem in der Projektion auf die 4. und 8. Hauptkomponente auffällt.

Der 1. und 10. Cluster enthalten zirka 1% der Warenkörbe, wodurch es schwierig wird diese in der Projektion klar darzustellen. Der 1. Cluster enthält Haushalte, die häufig Wein und Spiritousen kaufen, während der 10. Cluster die Konsumenten von Topfen und verschiedenen Topfenspeisen enthält.

Insgesamt fällt die Segmentierung der kleinen Einkäufe der Haushalte also ähnlich aus wie die Segmentierung der Gesamthaushalte in Abschnitt 6.5. Für eine differenzierte Betrachtung der Großeinkäufe könnte man Cluster 8 weiter aufteilen.

Kapitel 7

Zusammenfassung und Ausblick

Zum Testen verschiedener Clusterverfahren wurde zunächst nach wirtschaftlichen Überlegungen ein Warenkorb erzeugt, wie er auch in der Praxis hätte entstehen können. Dieser wurde mit unterschiedlichen Algorithmen und Vorgehensweisen mit und ohne Gruppieren der Daten segmentiert. Die Frage nach der Anzahl der Segmente konnte anhand der Daten nicht beantwortet werden, daher wurde auf verschiedene bei der Simulation berücksichtigte Anzahlen eingegangen. Ebenso konnte die Frage nach dem Distanzmaß nicht restlos geklärt werden, da sich unterschiedliche Distanzen in unterschiedlichen Situationen als nützlich erwiesen. So liefert zum Beispiel die Analyse mit der Jaccard Distanz bessere Ergebnisse, wenn dem Rauschen kein eigener Cluster zugewiesen werden kann, während im gegenteiligen Fall die Ergebnisse der Segmentierung mit euklidischer Distanz näher an die tatsächlichen Typen herankommen.

Das simultane Clustern der Gruppen und das Clustern der Gruppen im Austauschverfahren zeigten sich gleichermaßen durchaus erfolgversprechend. Die Segmentierung mit a priori Gruppierung der Daten lieferte die höchsten Übereinstimmungen mit den tatsächlichen Typen des synthetischen Warenkorbes, der Grad der Übereinstimmung hängt dabei stark von der Gruppengröße ab. Im Limit stimmen die Segmentierung und die zugrundeliegenden Typen exakt überein (Zentraler Grenzverteilungssatz). Gewöhnliche Clusteranalyse ohne ein Gruppieren der Daten brachte kaum zielführende Lösungen.

Der zweite betrachtete Datensatz stammt aus einer Umfrage über das Einkaufsverhalten der Konsumenten. Auch hier war der Versuch die Clusterzahl zu bestimmen wenig erfolgreich, vielmehr stellte sich heraus, dass der Datensatz keine natürliche Gruppenstruktur aufweist, dass aber eine künstliche Struktur dennoch zweckmäßig ist. Auch bei diesen Daten lieferte die Seg-

mentierung ohne Gruppierung der Daten keine interpretierbaren Ergebnisse.

Nach der Gruppierung waren die Daten bedeutend leichter zu clustern, vor allem eine Partitionierung mit euklidischer Distanz ergab sinnvolle adressierbare Kundengruppen. Auch das simultane Clustern der Haushalte sowie die Segmentierung der Haushalte im Austauschverfahren lieferten leicht interpretierbare Lösungen. Die erzielten Segmentierungen der beiden Verfahren waren einander zudem großteils ähnlich. Für sinnvolle Partitionierungen wurden weniger Segmente benötigt als beim Clustern ohne Gruppierung; durch das Verschieben der Gruppen in einen gemeinsam Cluster sind sogar häufig Cluster leer geworden und weggefallen.

Durch die Visualiserung mit der Hauptkomponentenanalyse wurde eine Interpretation der Partitionierungen möglich und die konvexen Clusterhüllen ließen die Struktur der Segmente erkennen. So konnte das spezifische Einkaufsverhalten der Kunden ein und desselben Clusters teilweise genau identifiziert werden. Auch die soziodemographischen Daten der Haushalte, die an der Umfrage teilnahmen, zeigten einige Zusammenhänge zur Clusterstruktur. Personen mit ähnlichem Einkaufsverhalten weisen also zum Teil auch ähnliche soziodemographische Daten auf.

In dieser Arbeit wurden die gezeigten Vorgehensweisen zur Segmentierung gruppierter Daten erstmals betrachtet und angewandt. Die entwickelten Verfahren bieten zahlreiche Möglichkeiten zur Partitionierung gruppierter Warenkörbe sowie zur Anwendung in anderen Bereichen. Sie können auch mit weiteren Verfahren zur Marktsegmentierung kombiniert angewandt werden und damit der Marktforschung von großem Nutzen sein.

Anhang A

Barplots der Segmentierungen

A.1 Analyse des synthetischen Warenkorbes

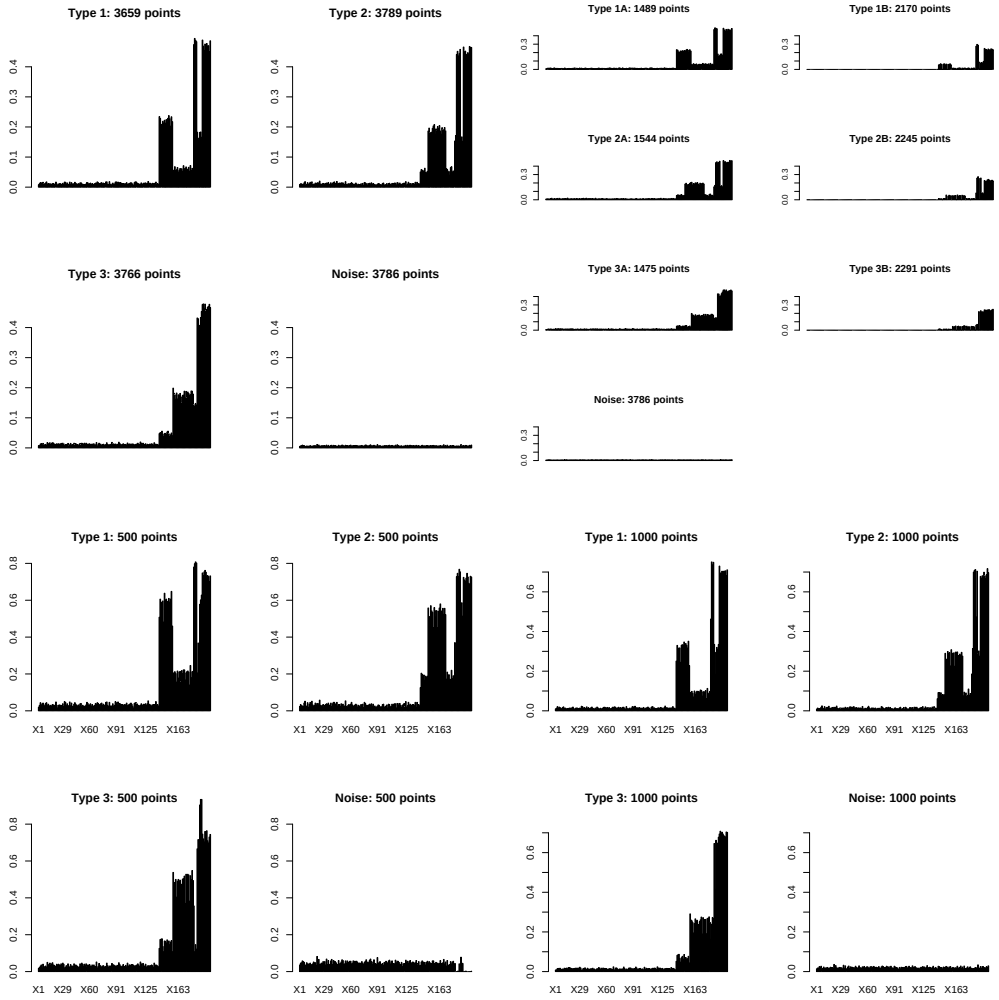

Abbildung A.1: Die zugrundeliegenden Cluster des gesamten (1.Zeile) und des zu 1500 und 3000 Kunden gruppierten (2.Zeile) Datensatzes

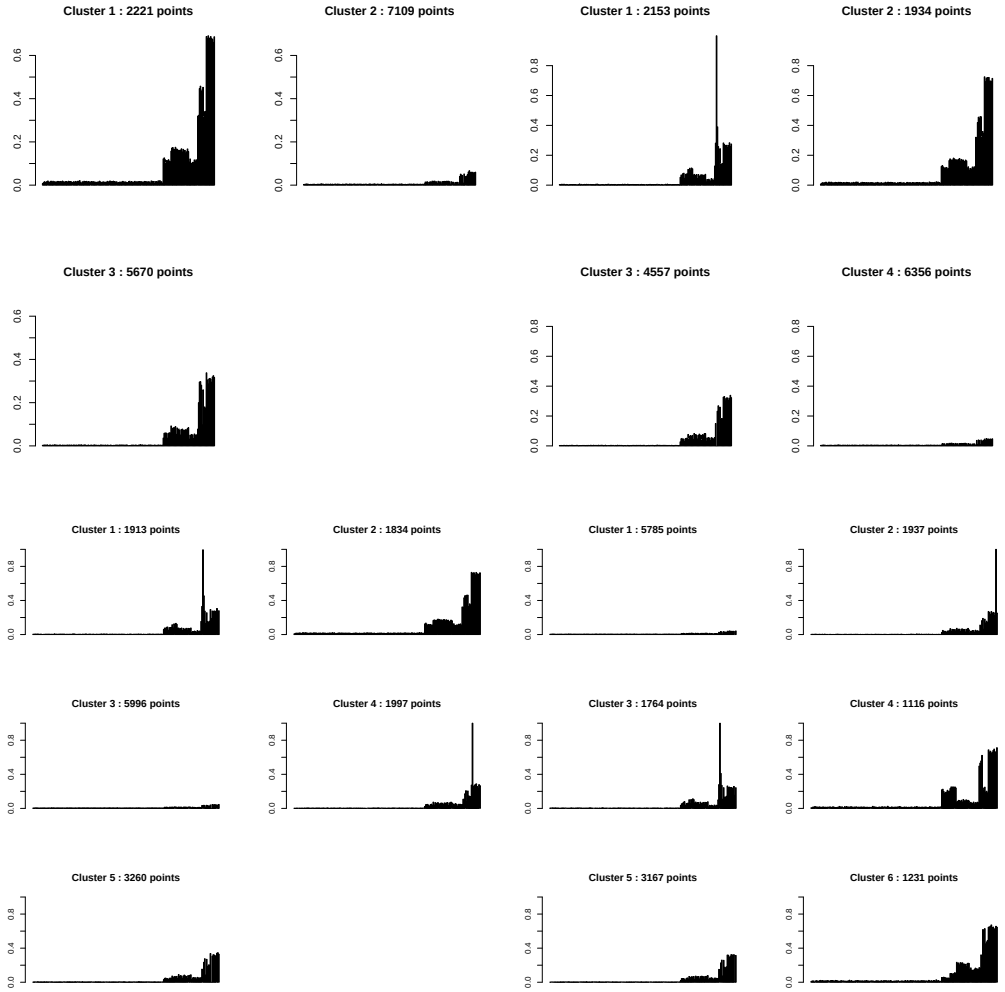

Abbildung A.2: Clusterzentren bei Segmentierung des gesamten Datensatzes mit K-Means und euklidischem Distanzmaß in 3, 4, 5 und 6 Cluster

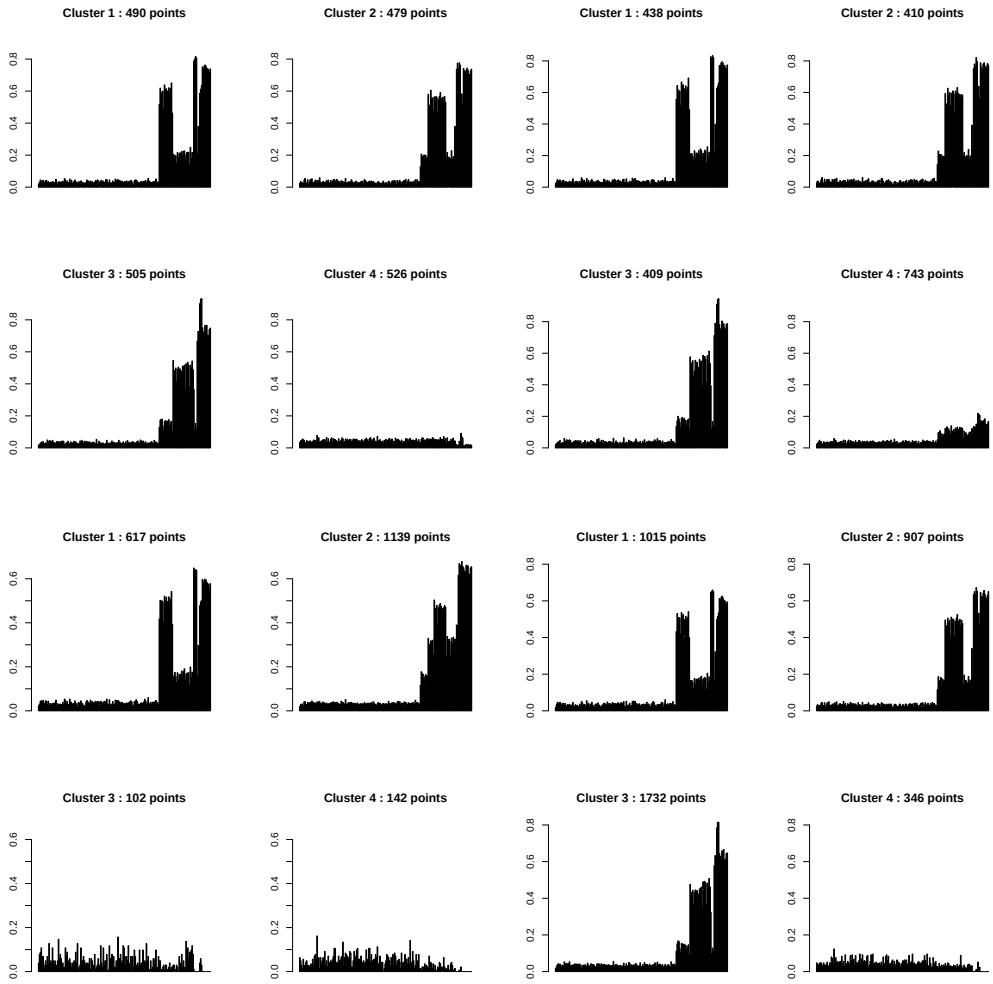

Abbildung A.3: 4 Clusterzentren des zu 1500 Kunden gruppierten Datensatzes; Verfahren und Distanzmaß: links oben kmeans, rechts oben kmedians, links unten ejaccard und rechts unten jaccard

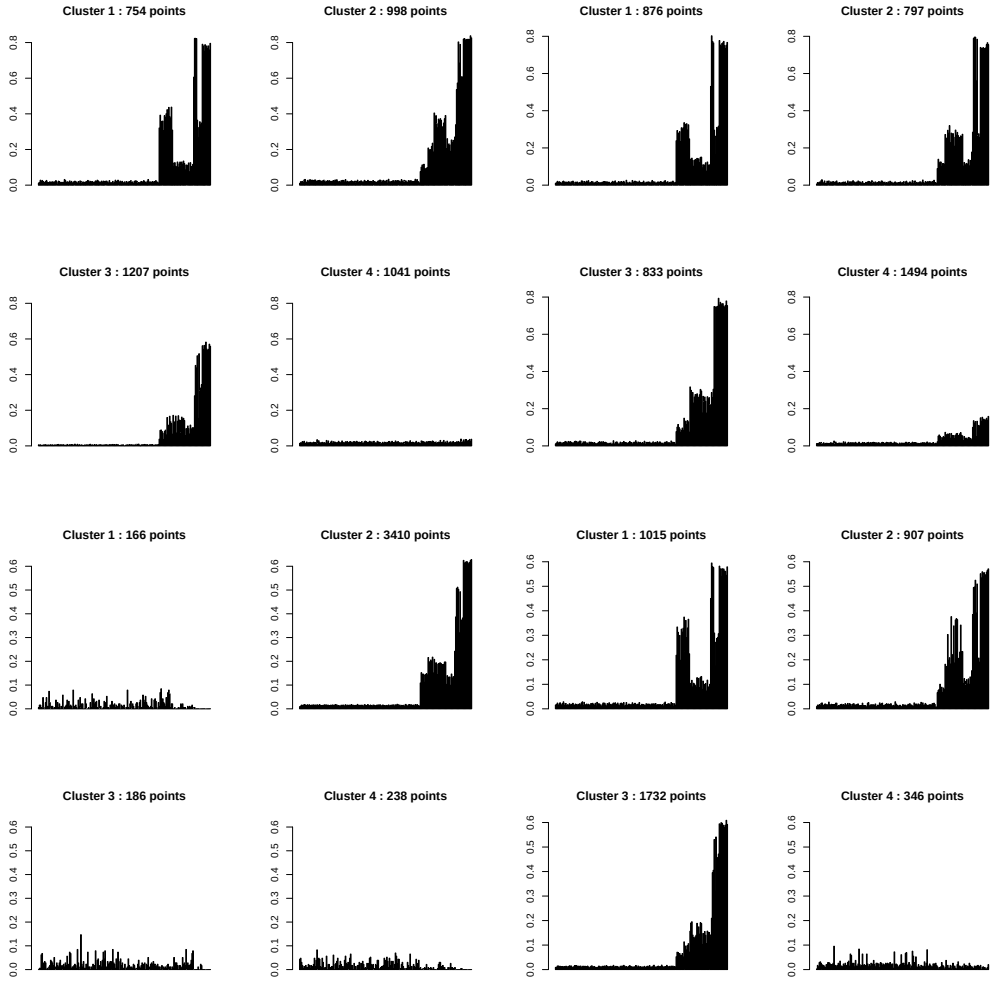

Abbildung A.4: 4 Clusterzentren des zu 3000 Kunden gruppierten Datensatzes

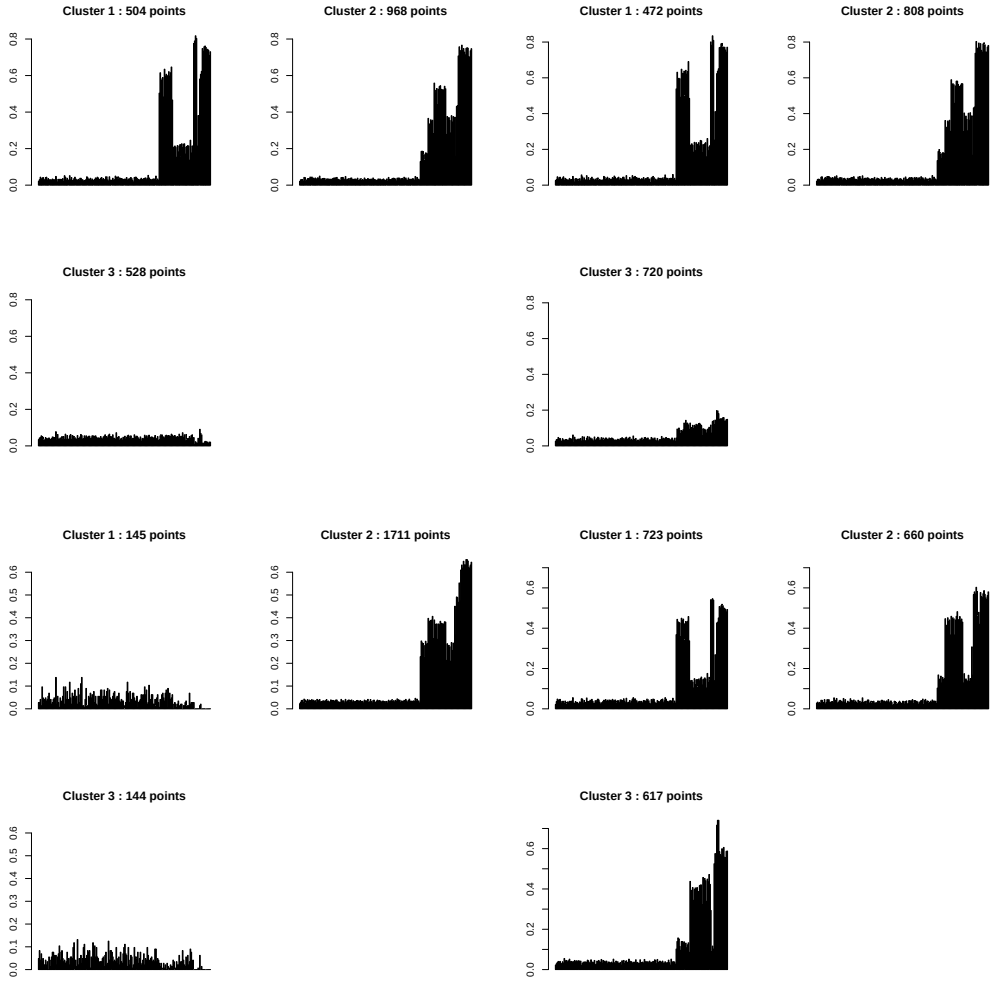

Abbildung A.5: 3 Clusterzentren des zu 1500 Kunden gruppierten Datensatzes

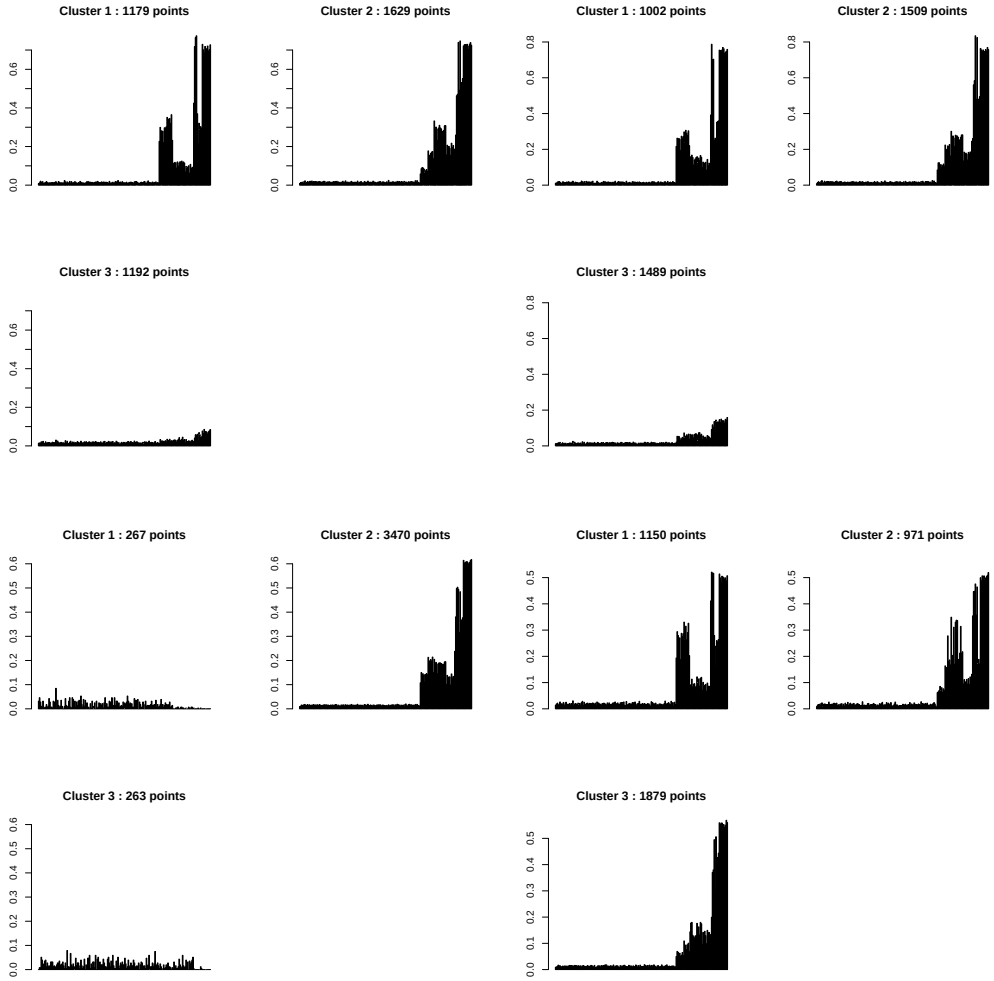

Abbildung A.6: 3 Clusterzentren des zu 3000 Kunden gruppierten Datensatzes

A.2 Segmentierung der Umfragedaten

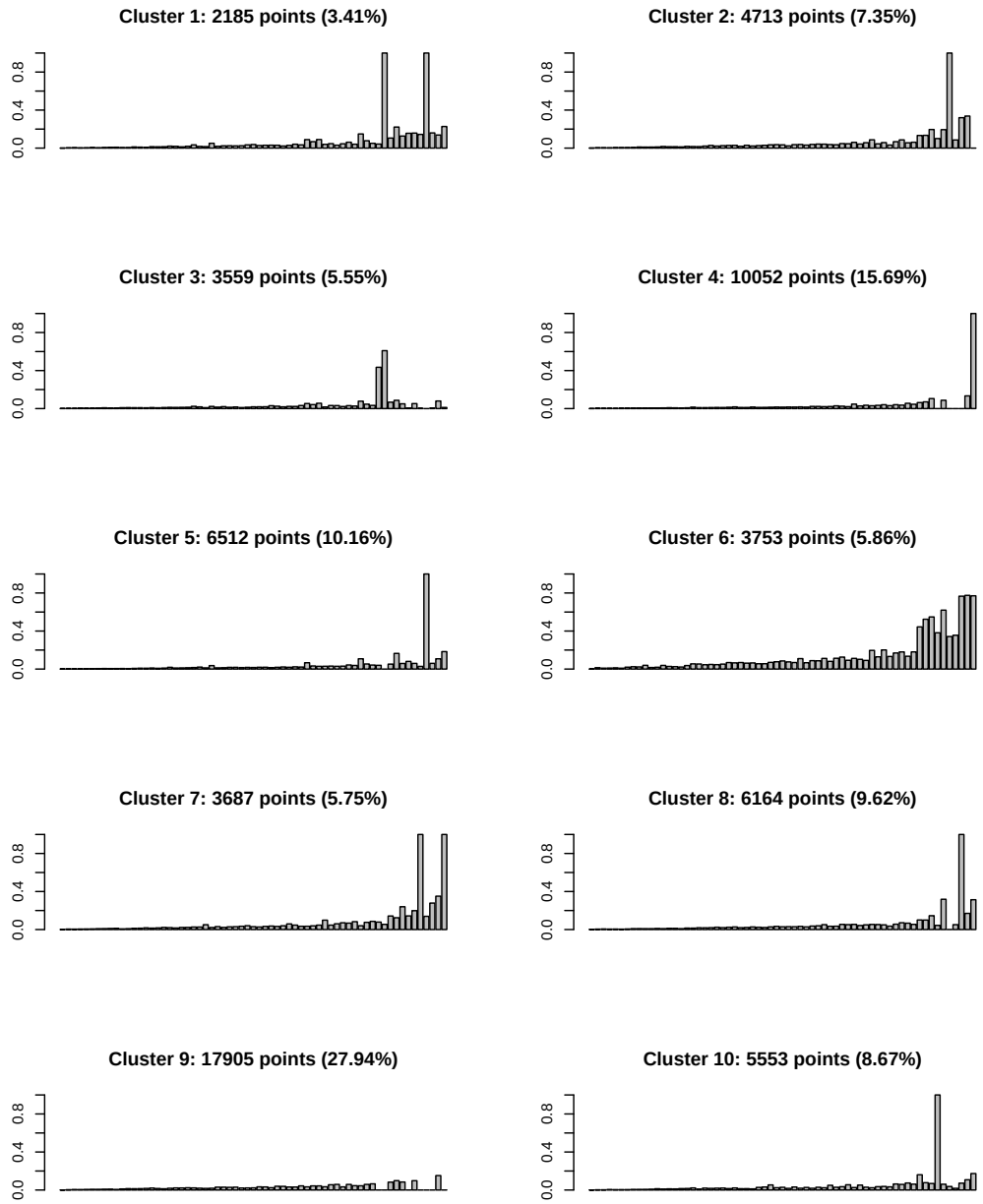

Abbildung A.7: Segmentierung des dritten Datenteils mit euklidischer Distanz

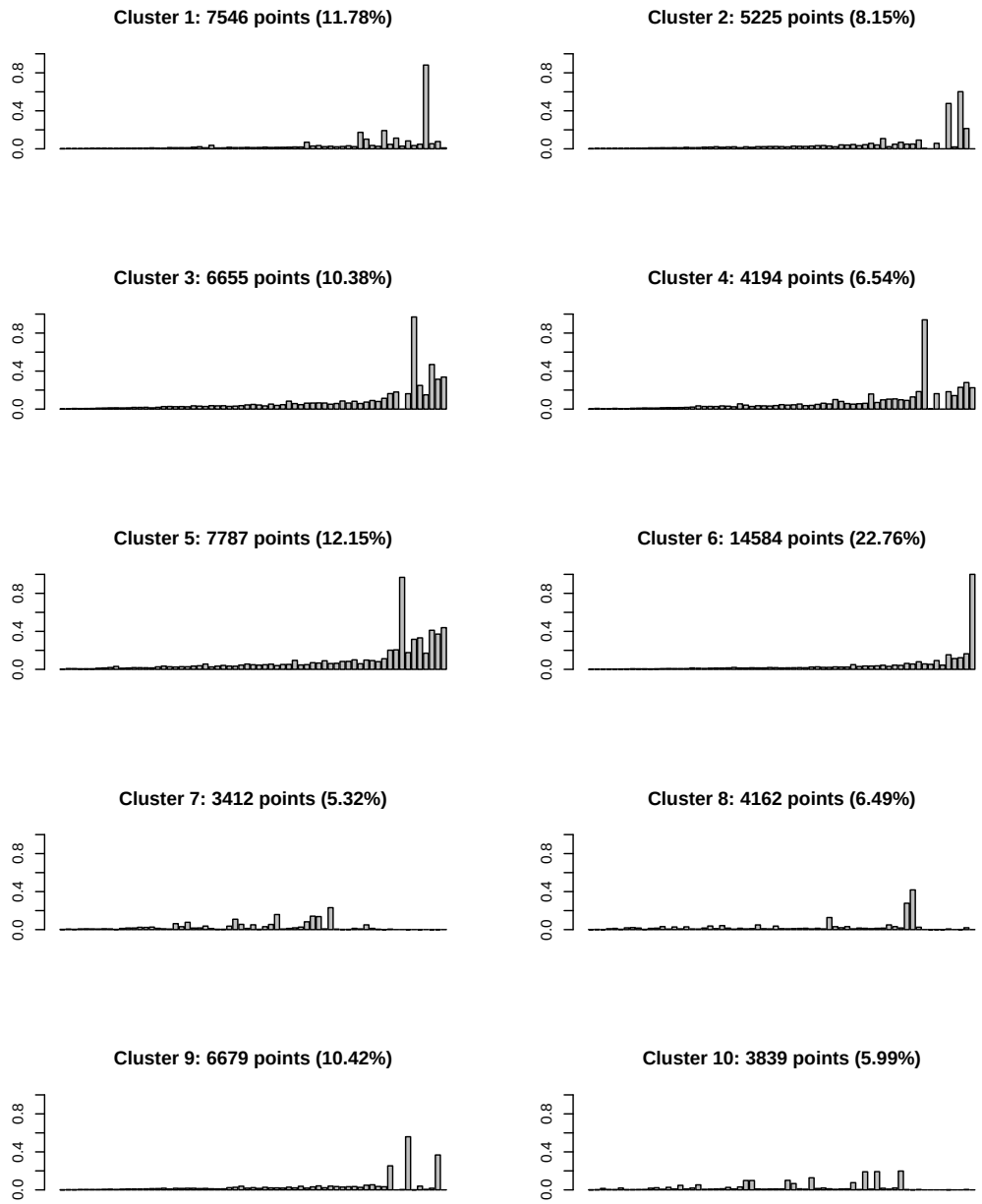

Abbildung A.8: Segmentierung des dritten Datenteils mit Jaccard Distanzmaß

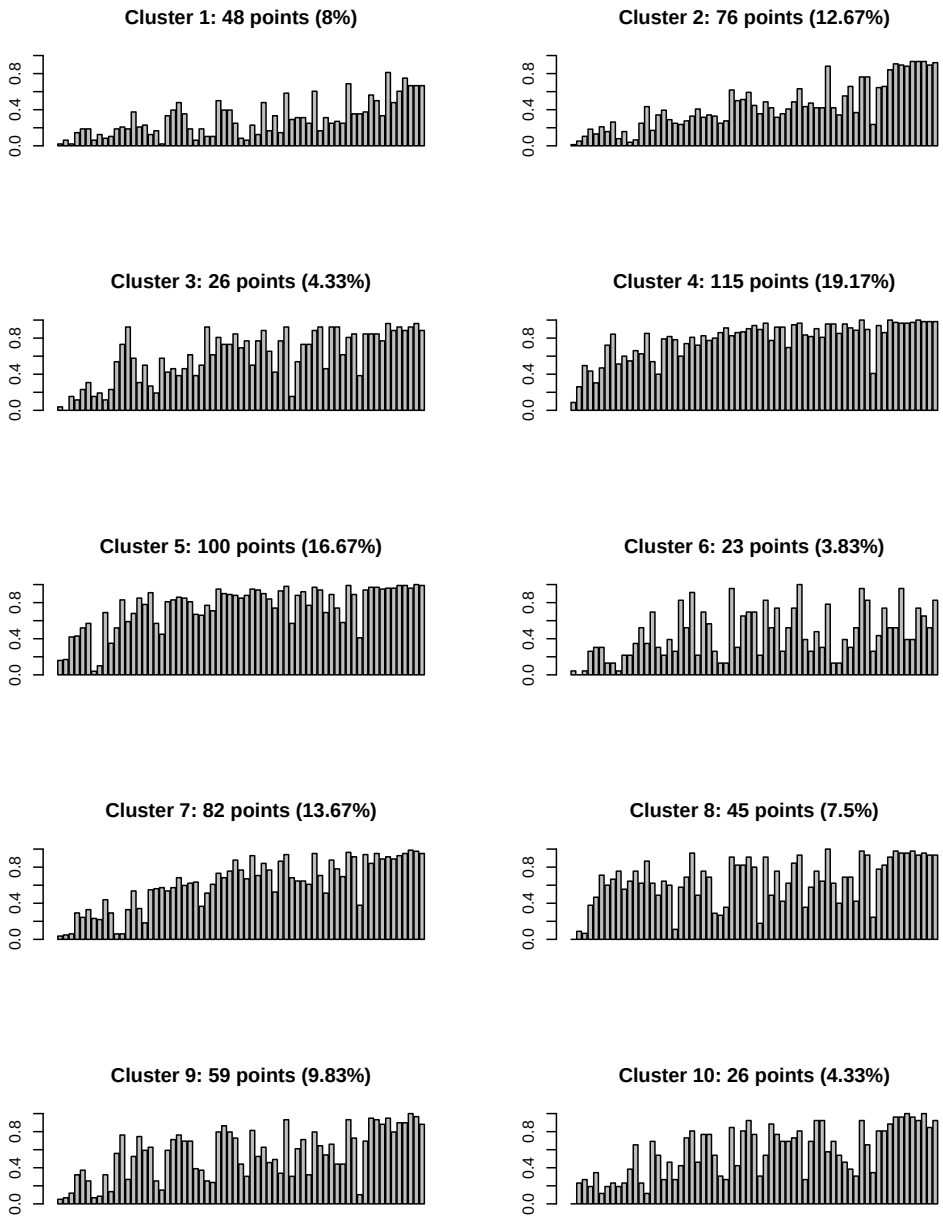

Abbildung A.9: Segmentierung der a priori gruppierten Daten mit euklidischer Distanz

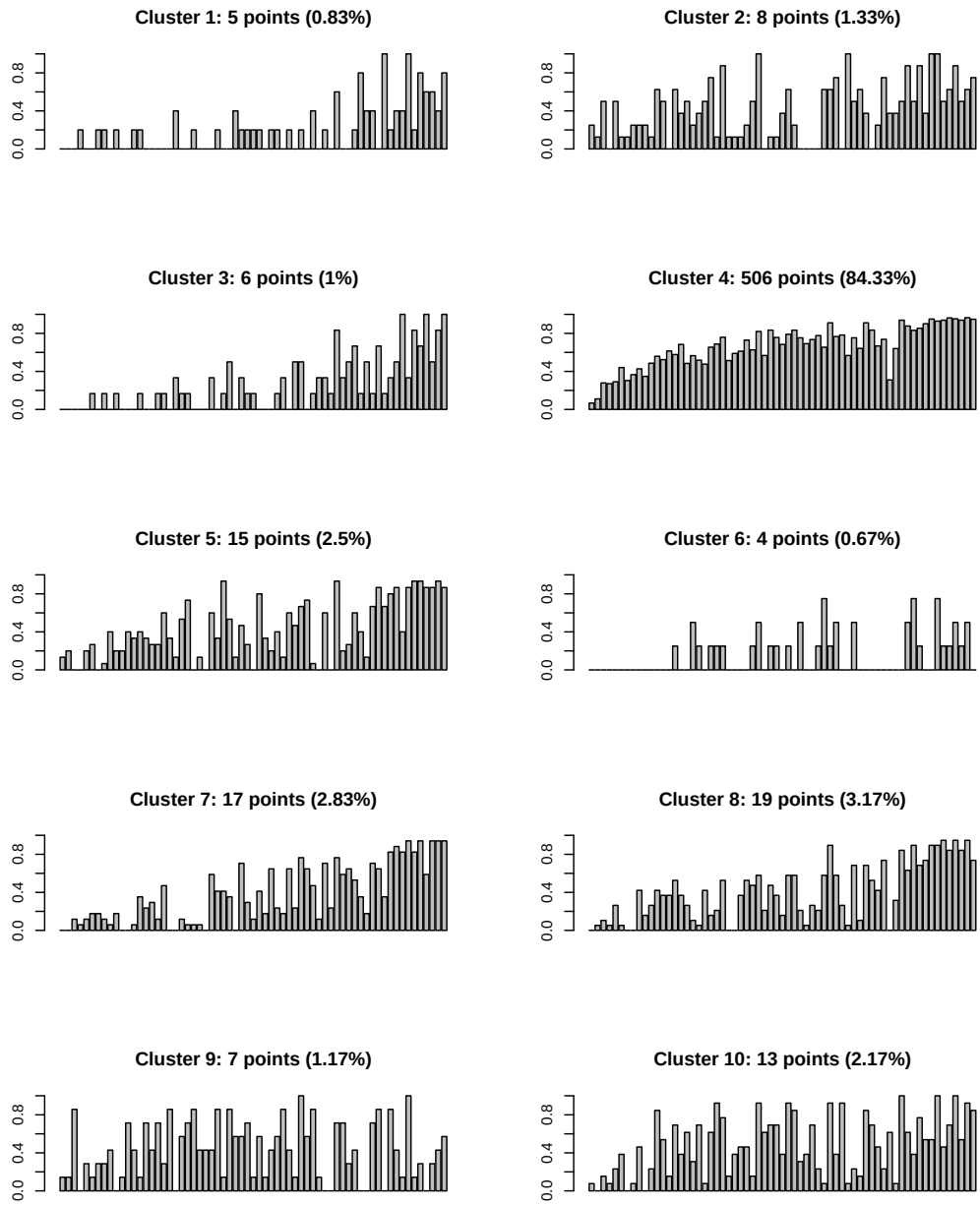

Abbildung A.10: Segmentierung der a priori gruppierten Daten mit Jaccard Distanzmaß

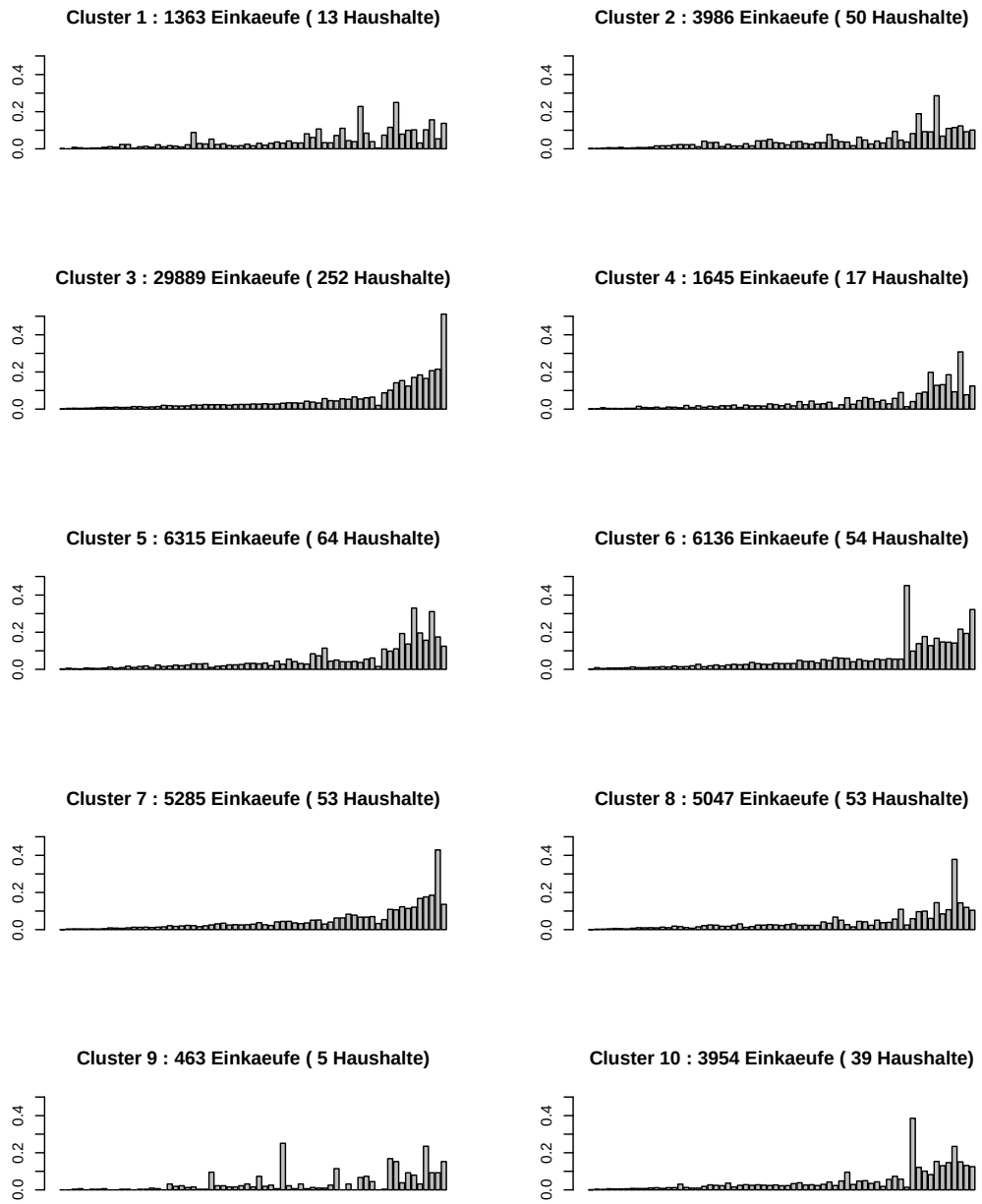

Abbildung A.11: Segmentierung durch simultanes Cluster der Haushalte mit Jaccard Distanzmaß

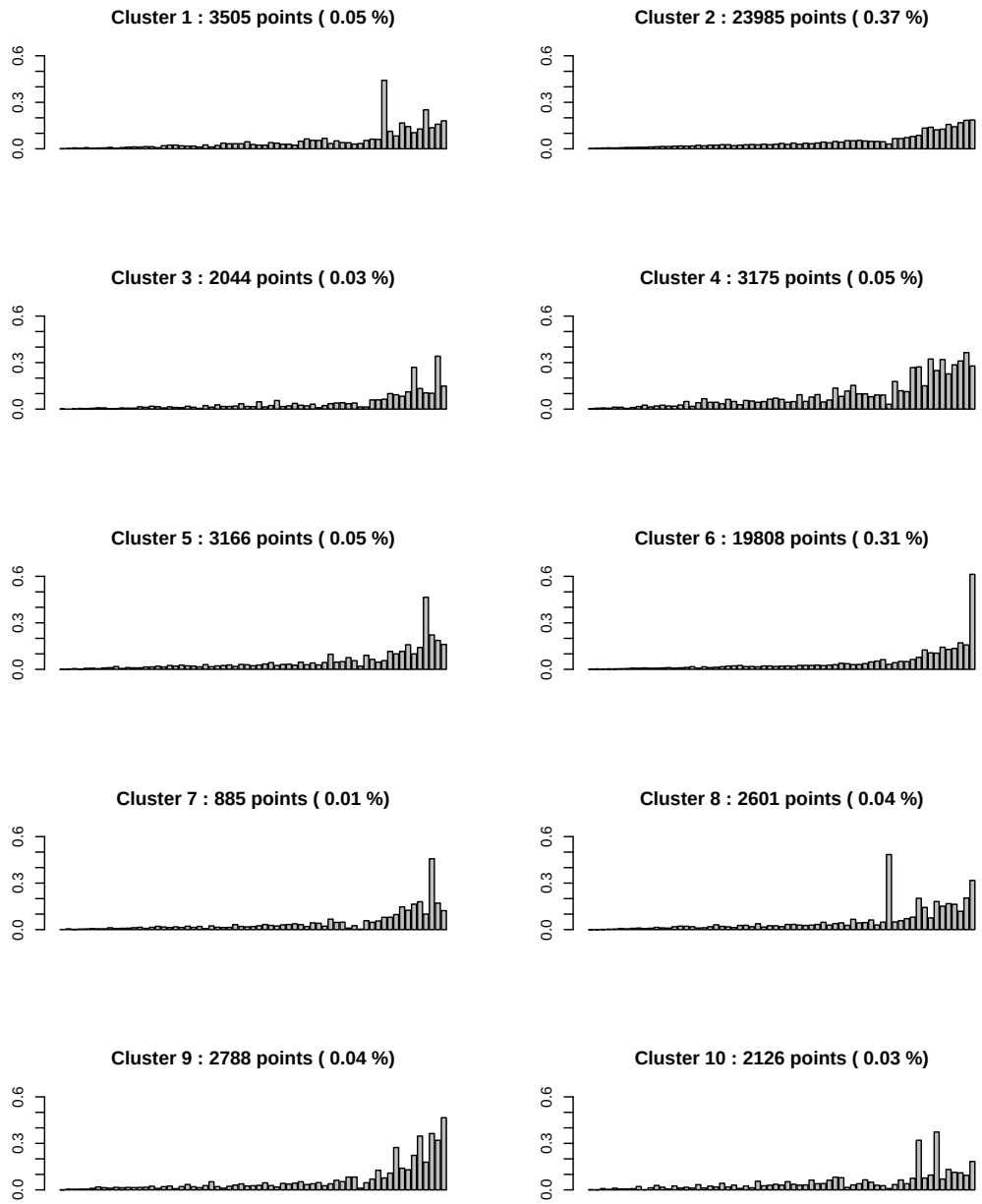

Abbildung A.12: Segmentierung im Austauschverfahren mit euklidischem Distanzmaß

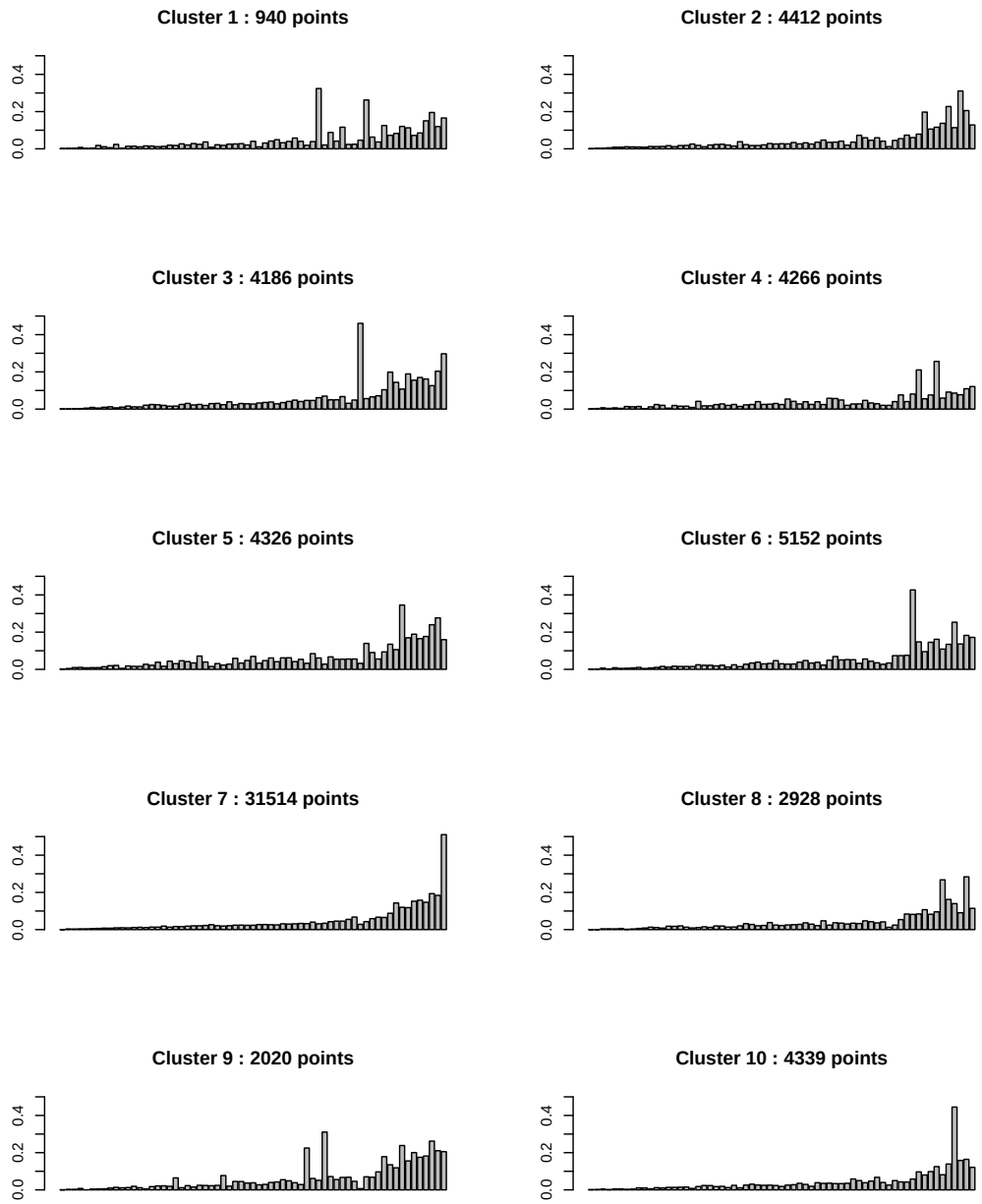

Abbildung A.13: Segmentierung im Austauschverfahren mit Jaccard Distanzmaß

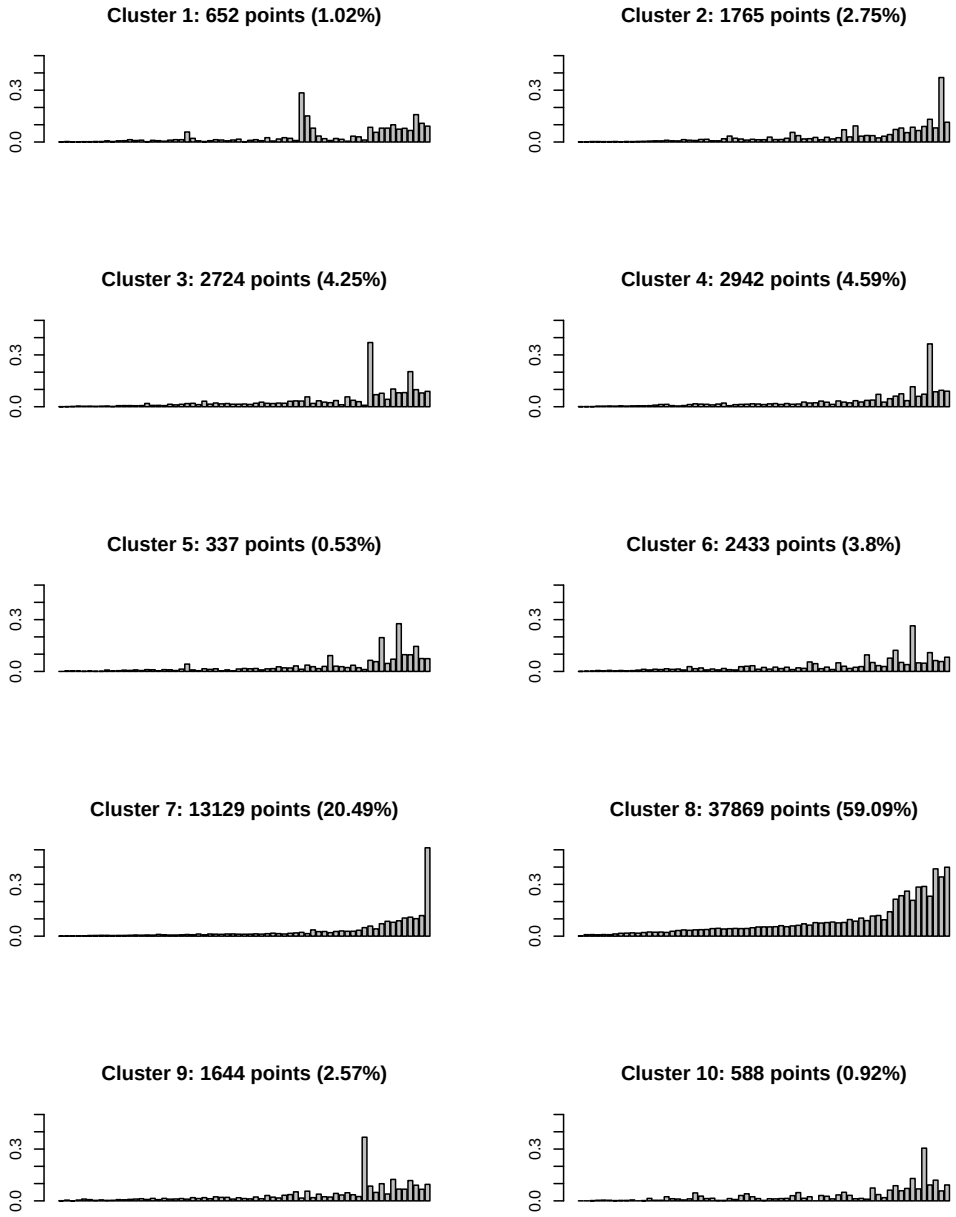

Abbildung A.14: Segmentierung durch simultanes Cluster der Haushalte mit Jaccard Distanzmaß und der Möglichkeit Haushalte nach Groß- und Kleineinkäufen getrennt zuzuordnen

Anhang B

Der Datensatz der Umfragedaten

B.1 Die Produktgruppen

Tabelle B.1 gibt Auskunft über die Produktgruppen des Datensatzes aus Kapitel 6. Die erste Spalte zeigt die der Produktgruppe zugewiesene Nummer, die zweite Spalte die Gruppenbezeichnung im Originaldatensatz.

1	wg01	Fensterreiniger
2	wg02	Teppich-, Polsterreiniger
3	wg03	WC-Reiniger
4	wg11	Tomaten-Ketchup und -mark
5	wg12	Würzsoßen und diverses Ketchup (außer Tomaten)
6	wg2	Mayonnaise und Salatsoßen
7	wg31	Buntwaschnittel
8	wg32	Feinwaschmittel
9	wg41	Klarspüler, Spezialsalz (für Geschirrspüler)
10	wg42	Maschinengeschirrspülmittel
11	wg43	Handgeschirrspülmittel
12	wg5	Haushaltsreiniger, Scheuermittel
13	wg61	Zahnbürsten
14	wg62	Mundwasser
15	wg81	Frisch- und Haltbarmilch
16	wg82	Milchprodukte (ohne Frisch- und H-Milch)
17	wg101	Wäscheweichspüler
18	wg102	Waschmittelzusätze
19	wg11	Zahncreme
20	wg12	Bohnenkaffee - Röstware

21	wg13	Bohnenkaffee - Extrakt u. Kaffeemittel
22	wg171	TK Kuchen/Gebäck, Baguette, Snackprodukte
23	wg172	TK Fisch
24	wg173	TK Gemüse
25	wg174	TK Pommes Frites u. Kartoffelprodukte
26	wg175	TK Pizza
27	wg176	TK Fertiggerichte u. Suppen
28	wg177	TK Obst
29	wg18	Tee
30	wg19	Kakao
31	wg20	Spirituosen, Wermut, Aperitif, Sherry, Portwein
32	wg21	Universalwaschmittel
33	wg23	Senf, Kren
34	wg25	Feinseifen, Waschlotion
35	wg26	Sekt
36	wg28	Bodenpflege
37	wg29	Badezusätze
38	wg30	Kartoffelfertigprodukte
39	wg31	Fertigpudding, Dessert
40	wg321	Küchenrollen
41	wg322	Kosmetiktücher
42	wg33	Bier
43	wg35	Wein, Glühwein, Apfelcidre
44	wg36	Alkoholfreie Getränke ohne Kohlensäure (Fruchthaltige)
45	wg39	Schuh- und Lederpflege
46	wg45	Pudding-, Dessertpulver
47	wg461	Cola- und Cola-Mixgetränke
48	wg462	Limonaden
49	wg463	Andere alkoholfreie Getränke mit Kohlensäure (kein Mineralwasser, Cola, Limo)
50	wg47	Weichkäse, Frischkäse, Schmelzkäse
51	wg48	Stärken, Steifen, Gardinenpflege; Backofen-, Spezial- und Badreiniger
52	wg50	Schlagobers
53	wg51	Dosenmilch, Kaffeesahne, Kaffeeweisser
54	wg55	Servietten
55	wg66	Tiernahrung, Katzenstreu
56	wg73	Topfen, Topfenspeisen
57	wg75	Filterpapier
58	wg78	Joghurt
59	wg80	Topfreiniger, Putzschwämme, Haushaltstücher
60	wg81	Hartkäse
61	wg84	Mineralwasser
62	wg85	Hauhaltsfolien, Butterbrotpapier
63	wg86	Eiscreme (Haushaltspackungen)
64	wg90	Cerealien (Müsli, Cornflakes etc.)
65	wg99	Toilettenpapier

Tabelle B.1: Übersicht über die Produktgruppen der Umfragedaten

B.2 Darstellung der Hauptkomponenten

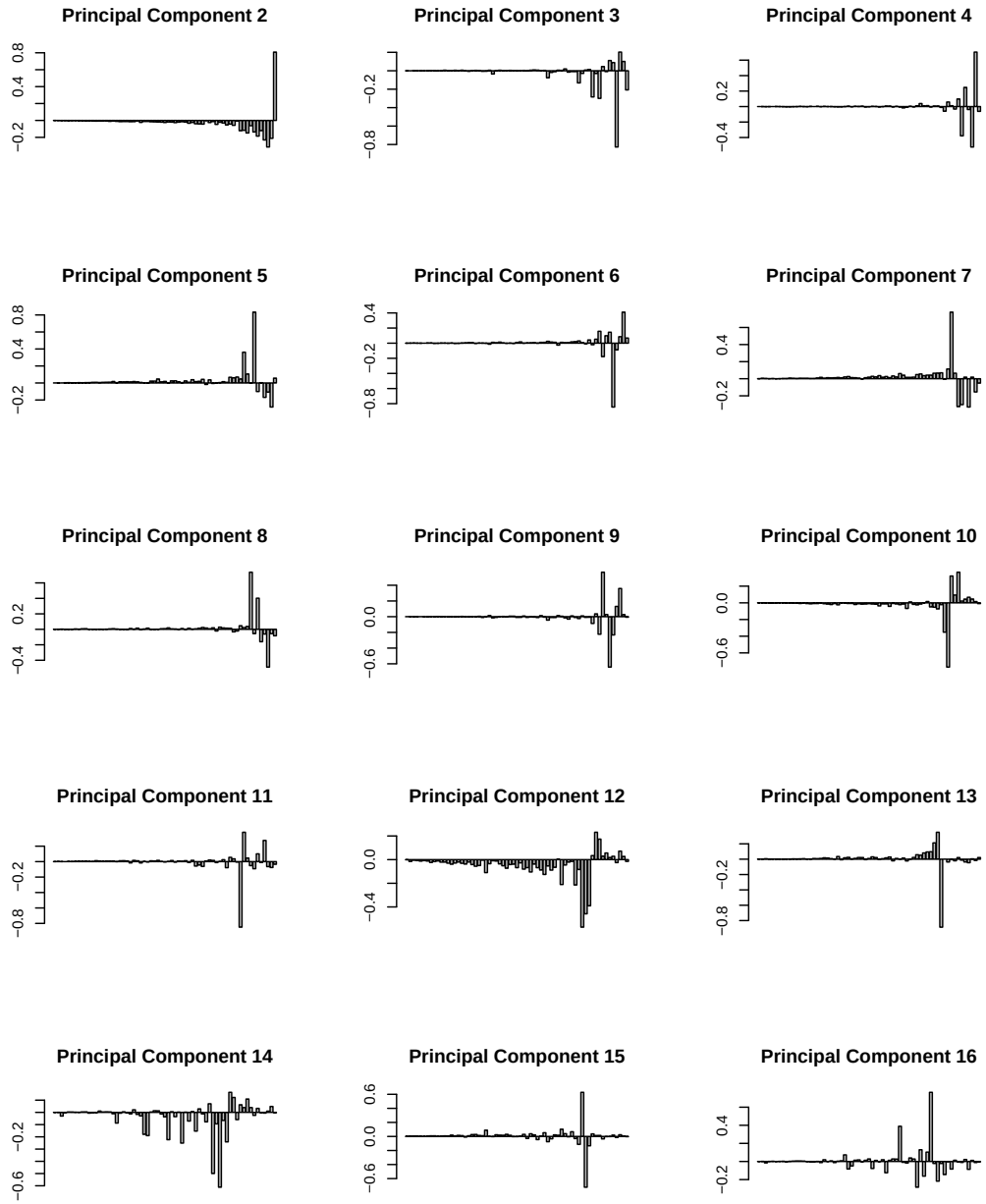

Abbildung B.1: Barplots der Hauptkomponenten

Literaturverzeichnis

Agrawal, R., T. Imielinski, und A. Swami (1993). *Mining Association Rules between Sets of Items in Large Databases.* URL http://www.almaden.ibm.com/software/quest/Publications/papers/ sigmod93.pdf.

Agrawal, R. und R. Srikant (1994). *Fast Algorithms for Mining Association Rules.* URL http://www.almaden.ibm.com/ software/quest/Publications/papers/vldb94.pdf.

Andrews, R. und I. Currim (2002). Identifying segments with identical choice behaviors across product categories: An Intercategory Logit Mixture Model. *International Journal of Research in Marketing 19*, 65–79.

Backhaus, K., B. Erichson, W. Plinke, und R. Weiber (1980). *Multivariate Analysemethoden.* Springer.

Baumgartner, H. und C. Homburg (1996). Applications of structural euqation modeling in marketing and consumer research: A review. *International Journal of Research in Marketing 13*, 139 – 161.

Brin, S., R. Motwani, und C. Silverstein (1997). *Beyond Market Baskets: Generalizing Association Rules to Correlations.* URL http://theory.stanford.edu/people/rajeev/postscripts/sigmod97a.ps.gz.

Chaturvedi, A., P. Green, und J. Carroll (2001). K-modes Clustering. *Journal of Classification 18*, 35–55.

Davidson, I. und S. Ravi (2005). *Clustering with Constraints: Feasibility Issues and the k-Means Algorithm.* URL http://www.cs.albany.edu/∼davidson/Publications/SDM2005Final.pdf.

Dolnicar, S., F. Leisch, A. Weingessel, C. Buchta, und E. Dimitriadou (1998). *A Comparison of Several Cluster Algorithms on Artificial Binary Scearios from Travel Market Segmentation.*

Hahsler, M., B. Grün, und K. Hornik (2005). *A Computational Environment for Mining Association Rules and Frequent Item Sets*.

Hartigan, J. und W. Wong (1979). A K-Means Clustering Algorithm. *Applied Statistics 28*, 100–108.

Heuser, U. und W. Rosenstiel (1998). *Internetsuche und neurale Netze: Stand der Technik*. Wilhelm-Schickard-Institut, Universität Tübingen.

Jedidi, K., H. Jagpal, und W. DeSarbo (1997). Finite-Mixture Structural Equation Models for Response-Based Segmentation and Unobserved Heterogeneity. *Marketing Science 16*, 39–59.

Kaufman, L. und P. Rousseeuw (1990). *Finding Groups in Data*. Wiley.

Law, M., A. Topchy, und A. Jain (2004). *Clustering with Soft and Group Constraints*. Joint IAPR International Workshop on Syntactical and Structural Pattern Recognition and Statistical Pattern Recognition.

Leisch, F. (2005a). *Neighborhood Graphs and Shadow Plots for Cluster Visualization*.

Leisch, F. (2005b). *A Toolbox for K-Centroids Cluster Analysis*.

MacQueen, J. (1967). *Some methods for classification and analysis of multivariate observations*. Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability 1.

Magidson, J. und J. Vermunt (2002). Latent class models for clustering: A comparison with K-means. *Canadian Journal of Marketing Research 20*, 37–44.

Martinetz, T., S. Berkovich, und K. Schulten (1993). "Neural-Gas" Network for Vector Quantization and its Application to Time-Series Prediction. *IEEE Transactions on Neural Networks 4*, 558–569.

Müller, J. (2001). *Clusteranalyse mit autologistischen Mischmodellen*. Diplomarbeit, Institut fuer Statistik und Wahrscheinlichkeitstheorie, Technische Universitaet Wien.

Papastefanou, G., P. Schmidt, A. Börsch-Supan, H. Lüdtke, und U. Oltersdorf (1995). *The ZUMA scientific use file of the GfK ConsumerScan Household Panel*. 2001, Social and Economic Analyses of Consumer Panel Data, Zentrum für Umfragen, Meinungen und Analysen (ZUMA), Mannheim.

Park, J., M. Chen, und P. Yu (1995). *An Effective Hash-Based Algorithm for Mining Association Rules*. URL http://www.ee.ntu.edu.tw/~mschen/paperps/smod95.ps.

Punj, G. und D. Stewart (1983). Cluster Analysis in Marketing Research: Review and Suggestions for Applications. *Journal of Marketing Research 20*, 134–148.

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

Ripley, B. (1996). *Pattern Recognition and Neural Networks*. Cambridge University Press.

Seppaenen, J., E. Bingham, und J. Mannila (2003). *A Simple Algorithm for Topic Identification in 0-1 Data*.

Smith, W. (1956). Product Differentiation and Market Segmentation as Alternative Marketing Strategies. *Journal of Marketing 21*, 3–8.

Venables, W. und B. Ripley (1996). *Modern Applied Statistics with S-Plus*. Springer.

Wagstaff, K., C. Cardie, S. Rogers, und S. Schroedl (2001). Constrained K-means Clustering with Background Knowledge. *Proceedings of the Eighteenth International Conference on Machine Learning* , 577–584.

Wedel, M. und W. Kamakura (1998). *Market Segmentation*. Kluwer Academic Publishers.

Wind, Y. (1978). Issues and Advances in Segmentation Research. *Journal of Marketing Research 15*, 317–337.