



Cluster Analysis in Marketing Research

Thomas Reutterer and Daniel Dan

Contents

Introduction	2
An Overview of Clustering Methods	3
Data Quality and Proximity Measures	4
Distance-Based Cluster Algorithms	7
Cluster Analysis of Market Basket Data	9
Data Characteristics and Choice of Distance Measure	9
Hierarchical Clustering	12
K-Medoid Clustering	16
K-Centroids Cluster Analysis	20
Conclusions	25
Cross-References	26
References	26

Abstract

Cluster analysis is an exploratory tool for compressing data into a smaller number of groups or representing points. The latter aims at sufficiently summarizing the underlying data structure and as such can serve the analyst for further consideration instead of dealing with the complete data set. Because of this data compression property, cluster analysis remains to be an essential part of the marketing analyst's toolbox in today's data rich business environment. This chapter gives an overview of the various approaches and methods for cluster analysis and links them with the most relevant marketing research contexts. We also provide pointers to the specific packages and functions for performing cluster analysis

T. Reutterer (✉)

Department of Marketing, WU Vienna University of Economics and Business, Vienna, Austria
e-mail: thomas.reutterer@wu.ac.at

D. Dan

Department of New Media, Modul University Vienna, Vienna, Austria
e-mail: daniel.dan@modul.ac.at

using the R ecosystem for statistical computing. A substantial part of this chapter is devoted to the illustration of applying different clustering procedures to a reference data set of shopping basket data. We briefly outline the general approach of the considered techniques, provide a walk-through for the corresponding R code required to perform the analyses, and offer some interpretation of the results.

Keywords

Cluster analysis · Hierarchical clustering · k -centroid clustering · k -medoid clustering · Marketing analysis · Marketing research

Introduction

Cluster analysis is a generic term for exploratory statistical techniques and methods aiming at detecting groupings in data sets that are internally more homogeneous than the entities across the categorized groups. One of the primary goals of clustering is data compression, i.e., to summarize the original entities by a smaller number of groups or representing points instead of considering the complete data set. Cluster analysis has a long history and emerged as a major topic in the 1960s and 1970s under the label “numerical taxonomy” (cf., Sokal and Sneath 1963; Bock 1974). The origins of cluster analysis appeared in disciplines such as biology for deriving taxonomies of species or psychology to study personality traits (Cattell 1943). Over the years, a large variety of clustering techniques has been proposed for numerous types of applications in diverse fields of research. From a historical perspective, excellent books on cluster analysis have been written by Anderberg (1973), Hartigan (1975), Späth (1977), Aldenderfer and Blashfield (1984), Jain and Dubes (1988) or Kaufman and Rousseeuw (1990). Additionally, Arabie and Lawrence (1996) provide an extensive compilation of contributions on various aspects of cluster analysis; for more development updates in the field, see Everitt et al. (2011) or Hennig et al. (2015). From a marketing researcher’s perspective, Punj and Stewart (1983) or Arabie and Lawrence (1994) provide comprehensive reviews of cluster analysis.

The “classical” marketing problems involving the application of clustering methods are market segmentation (Wedel and Kamakura 2000; Dolnicar et al. 2018) and competitive market structure (CMS) analysis (DeSarbo et al. 1993). The former entails deriving segments of customers who either react homogeneously to various marketing mix variables (response-based segmentation) or are more homogeneous with respect to some psychometric constructs such as product attitudes or product images, perceived value, or preferences (construct-based segmentation); see Mazanec and Strasser (2000) or Reutterer (2003) for this distinction and an overview of corresponding clustering methods.

The task of CMS analysis is to derive a configuration of brands in a specific product class which adequately reflects inter-brand competitive relationships as perceived by consumers (DeSarbo et al. 1993). This is typically accomplished via

an arrangement of the rivaling brands in ultrametric trees, overlapping or fuzzy cluster structures (Rao and Sabavala 1981; Srivastava et al. 1981, 1984). Because they utilize identical data structures but just differ in the mode of data compression, segmentation (compression of the consumer mode) and CMS (compression of the brand mode) turn out to be “reverse sides of the same analysis” (Grover and Srinivasan 1987; Reutterer 1998). Yet another very similar data structure arises when companies keep record of their customer transactions (e.g., by tracking them over time in customer relationship management systems). Such data sets tend to be huge and accrue as clickstreams of visitation and corresponding purchasing patterns on a website or as sequences of shopping baskets comprising jointly purchased items or product categories. The data compression tasks involved in the so-called exploratory market basket analysis (Mild and Reutterer 2003; Boztuğ and Reutterer 2008; Reutterer et al. 2017) are analogous to those in market segmentation vs. CMS analysis and also entail some suitable clustering method. The marketing literature refers to the task of discovering subgroups of distinguished cross-category interrelationship patterns among jointly purchased items or product categories also as “affinity analysis” (Russell et al. 1999; Manchanda et al. 1999; Russell and Petersen 2000).

The remainder of this chapter is organized as follows: In the next section, we provide a brief overview of the various clustering methods and focus on how to proceed when conducting distance-based clustering in more detail. We also present the most popular distance/proximity measures and algorithms as well as the most commonly used software implementations in the computational environment R available to analysts. To demonstrate the application and the results obtained from using various clustering methods, we then provide a couple of hands-on examples on how to put cluster analysis into action. Using one and the same data set, we demonstrate the specific quality of data compression achieved when utilizing a specific type of cluster analysis. While most textbooks use market segmentation as the standard case for illustrating cluster analysis in marketing (see, e.g., Chapman and McDonnell Feit (2019) and Dolnicar et al. (2018) for excellent examples), we focus in our demonstration on the exploratory analysis of shopping basket data representing customers’ joint purchase decisions across a wide range of product categories.

An Overview of Clustering Methods

We can distinguish between two major groups of clustering methods: model-based clustering and distance-based clustering. While model-based methods explicitly assume some statistical probability model as an underlying data generating process, the latter are more exploratory by nature. The idea behind model-based methods is that the observations arise from a probability distribution which is a mixture of two or more components (i.e., clusters). Each of these components is a density function with an associated weight in the mixture (e.g., a mixture of multivariate normal distributions) and the task is to determine the mixture distribution which fits the data

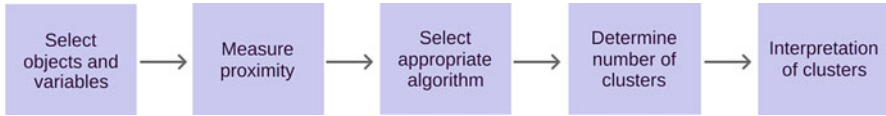


Fig. 1 Steps to conduct cluster analysis

best. This is usually done by varying the number of components and optimizing some fit or information criterion. A specific variant of model-based clustering is discussed in the chapter ▶ “[Finite Mixture Models](#)” of this handbook. For more technical details on model-based clustering, see Titterton et al. (1985), McLachlan and Basford (1988), Fraley and Raftery (2002), and Frühwirth-Schnatter (2006).

In the following, we will focus on distance-based clustering. The aim of distance-based clustering is to find groupings in the data such that the distance between entities within a group is minimized, while it is maximized for entities between groups for some predefined distance measure. The steps required to employ a distance-based clustering procedure are shown in Fig. 1. After the objects and variables of interest are selected from an available data base, the second step involves the choice of an appropriate proximity measure to quantify the (dis-)similarity between the objects to be clustered. In the next step, a specific cluster algorithm is selected, and once the results are obtained, the number of clusters is determined and the cluster solution interpreted accordingly. Because of their crucial impact on the resulting cluster solutions, we next discuss some of the most popular proximity measures and clustering algorithms used in research practice.

Data Quality and Proximity Measures

The choice of a proximity measure depends on the nature of the data to be clustered, more specifically, the scaling properties of the variables at hand. Generally speaking, we can distinguish between numerical (quantitative, metric) and categorical (qualitative, nonmetric) data. Metric data is characterized by a scale with numerically equal distances representing values of the underlying characteristic being measured, such as age, income, or the number of units sold over a month. With this kind of data, any mathematical operation can be performed, it can be displayed from the greatest to the least and vice versa. In contrast, categorical data include binary and nominal data with no natural order, for example: product choice, gender, ethnicity, etc. If categorical data imply an order relationship (such as preference rankings) of the measured objects, the scale is denoted as being ordinal. The latter include rating scales (e.g., brand attitudes measured using itemized scales), which occasionally are called pseudometric as they are treated as numerical data despite their ordinal properties. For a detailed discussion of measurement issues and data quality see, e.g., Mooi et al. (2018). Given the scaling properties of the data and the type of (dis-)similarity desired, the choice of a proximity measure determines how close/

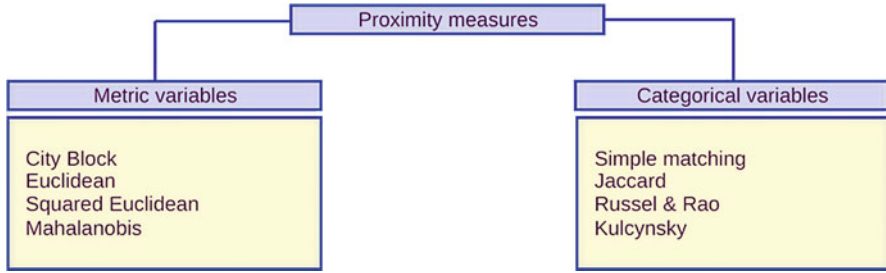


Fig. 2 Common examples of proximity measures

similar or how far/dissimilar objects in a data set are situated. A major distinction of alternative measures arises when we distinguish between metric and nonmetric data, see Fig. 2. In the case of metric data, most of the proximity measures are based on the summed distances of the objects with respect to all variables or dimensions of the data.

In an n -dimensional space, the most well-known and widely used distance measure between two data points $X = (x_1, x_2, \dots, x_n)$ and $Y = (y_1, y_2, \dots, y_n)$ arises as a family of metrics denoted as Minkowski distance or L_p norm (Adams and Fournier 2003), i.e., a metric where the distance between two vectors is given by the norm of their difference. The outcome of this metric is given by Eq. 1:

$$d_p(X,Y) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p} \tag{1}$$

The distance is a metric if $p \geq 1$. The most commonly used norms are L_1 known as the Manhattan norm and L_2 known as the Euclidean norm. The distances derived from these norms are called Manhattan distance

$$d_1(X,Y) = \sum_{i=1}^n |x_i - y_i|, \tag{2}$$

and Euclidean distance

$$d_2(X,Y) = \sqrt{\sum_{i=1}^n |x_i - y_i|^2}, \tag{3}$$

respectively. In cases where more weight should be put on the measurement of very distant data objects, the squared Euclidean distance can be used:

$$d_2^2(X,Y) = \sum_{i=1}^n |x_i - y_i|^2 \tag{4}$$

If the data is nonmetric (i.e., nominal, binary, or ordinal scales), the most common way of quantifying the (dis)similarity between data points is based on a two-way cross-classification of objects which counts for having a binary attribute: present or absent (note that ordinal data can be transformed into a series of binary variables accordingly). The corresponding similarity coefficients mentioned in Fig. 2 mainly differ in their assumptions on whether the common absence of a characteristic reflects similarity (such as the simple matching coefficient) or not and how much weight they put on the matched presence of an attribute.

Since we will use the Jaccard (dis)similarity coefficient in the following applications of cluster analysis using shopping basket data, we briefly illustrate the construction of the Jaccard index. The latter is used to assess the similarity s between two sets A , B or categories c_A , c_B . Formally, it measures the size ratio of their intersection $c_A \cap c_B$ divided by their union $c_A \cup c_B$ and can be written as follows:

$$s_{c_{AB}} = \frac{c_A \cap c_B}{c_A \cup c_B} = \frac{c_A \cap c_B}{c_A + c_B - c_A \cap c_B} \quad (5)$$

As discussed below in more detail, in the context of market-basket analysis the analyst's interest is in quantifying the (dis)similarity of products or product categories depending on whether they are jointly purchased in a set of transactions or not. In doing so, the product purchases are represented as binary elements with (1) denoting presence and (0) denoting absence of the specific product in a shopping basket. By cross-classifying a pair of products, we can calculate the Jaccard coefficient with the help of the following contingency table (cf. Sneath 1957; Kaufman and Rousseeuw 1990; Leisch 2006).

		Product 1		sum
		1	0	
Product 2	1	a	b	a+b
	0	c	d	c+d
sum		a+c	b+d	p

For a set of p shopping baskets, the Jaccard similarity coefficient (also often referred to as the Tanimoto similarity coefficient (Anderberg 1973)) for products 1 and 2 can be calculated as:

$$s_{\text{prod1,prod2}} = \frac{a}{a + b + c}, \quad (6)$$

and the corresponding dissimilarity coefficient is:

$$d_{\text{prod1,prod2}} = \frac{b + c}{a + b + c}, \quad (7)$$

with the elements in the contingency table representing:

- a , the number of transactions with purchases of both product 1 and 2
- b , the number of incidences of product 2 but no product 1 purchases

- c , the number of incidences of product 1 but not product 2 purchases
- d , the number of transactions with neither product 1 nor product 2 purchases

Note that $s_{prod1, prod2} = 1 - d_{prod1, prod2}$ and in practice d in the above contingency table is usually the cell with the (by far) highest counts. This particularly applies to the context of shopping basket analysis but is not limited to this case. In such situations, any proximity measure that treats co-incidences of common zeros (in our case: nonpurchases of two specific products or categories) the same way as common ones would be biased towards the absence of two characteristics. For example, this is the case for the simple matching coefficient (which is $s_{prod1, prod2} = \frac{a+d}{p}$) or the Hamming distance (i.e., the number of different bits $d_{prod1, prod2} = b + c$). Thus, in many scenarios it makes sense to use asymmetric proximity measures like the abovementioned Jaccard coefficient which gives more weight to common ones than to common zeros.

Distance-Based Cluster Algorithms

Regarding the choice of a cluster algorithm (step three in the above by Fig. 1), one popular way to distinguish variations of distance-based clustering methods is to divide them into hierarchical and nonhierarchical procedures. As illustrated by Fig. 3, the former can be split into agglomerative and divisive, while the later

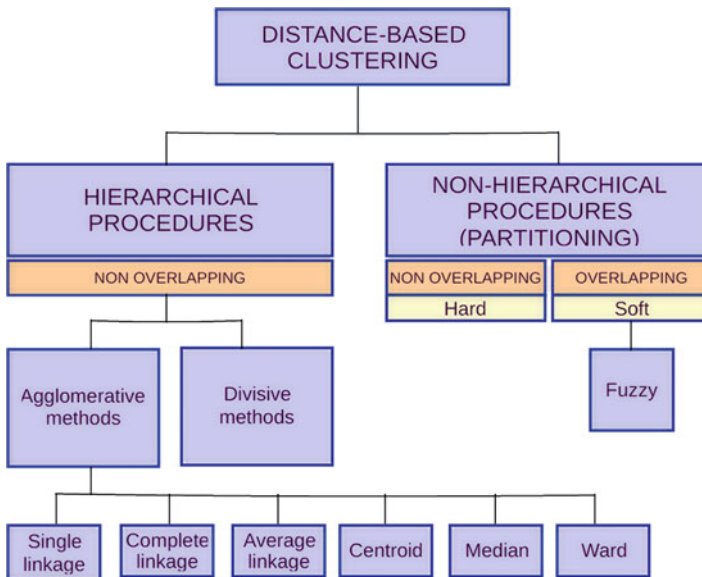


Fig. 3 Overview of distance-based clustering algorithms

can be branched into nonoverlapping, overlapping, or fuzzy clustering methods (Hruschka 1986; Wedel and Kamakura 2000).

While in the case of nonoverlapping clustering each entity is assigned to one single group, overlapping clustering techniques allow for the simultaneous membership of objects to multiple groups. For example, depending on the consumption, brands and/or consumers might belong to more than one cluster of products or to several segments, respectively (see, e.g., Arabie et al. 1981). Fuzzy clustering abandons the idea of a “hard” partitioning of the data and replaces fixed cluster assignments by a degree of membership assigned to each entity and cluster (Hruschka 1986). Note that despite similar in idea but different in conception and interpretation, the notion of a “soft” group membership assignment becomes also apparent in model-based clustering methods when probabilities of cluster memberships are estimated and thus each data point can be assigned to more than one cluster. In this case, the inclusion of a data point in multiple clusters is due to a probabilistic approach, not of a distance.

Nonoverlapping clustering approaches can be further classified into hierarchical and nonhierarchical methods (Punj and Stewart 1983). Techniques for hierarchical clustering either start out with all entities in a single cluster (divisive algorithms or top down) or with each entity in its own cluster (agglomerative algorithms or bottom up). The latter approach is more popular among marketing researchers and successively links pairs of clusters (or still isolated entities) from a previous stage based on their shortest mutual distance. The agglomeration schedule stops when all entities are combined into one single cluster. We illustrate the application of some common hierarchical clustering procedures below in section “[Hierarchical Clustering](#).”

Nonhierarchical clustering starts with a (typically randomly initialized) grouping of the data for a prespecified number k of clusters and aims to gradually improve the partition by optimizing a “minimum variance criterion,” i.e., by minimizing the inner (within-group) dispersion of the k -partition (cf. Bock 1974; Strasser 2000). The k -means algorithm was first proposed by MacQueen (1967) and its many variations (see, e.g., Jain and Dubes 1988; Kaufman and Rousseeuw 1990) are popular examples for such nonhierarchical distance-based clustering procedures. There is a huge variety of clustering procedures available in packages and functions provided by the R (R Core Team 2019) ecosystem for statistical computing. The most commonly used packages are the following:

- `stats`: The base R package provides a number of implementations for both partitioning and hierarchical clustering techniques. Function `kmeans()` comprises several algorithms for computing Euclidean distance-based partitions, while `hclust()` provides agglomerative hierarchical clustering algorithms. The `stats` package also provides various auxiliary functions like `dendrogram()` for visualizing cluster hierarchical solutions.
- `cluster`: This package provides R implementations of methods introduced in Kaufman and Rousseeuw (1990) and comprises a number or both partitioning (`pam()`, `clara()`, and `fanny()`) and hierarchical cluster algorithms (`agnes()`,

`diana()`, and `mona()`). The package also contains many extensions of these base methods and visualization functions (Struyf et al. 1996).

- `mclust`: A set of model-based clustering methods for fitting Gaussian finite mixture models using an expectation maximization (EM) algorithm is provided by the `mclust` package. It also provides numerous functions to assist cluster validation and evaluating the number of mixture components using the Bayesian Information Criterion (BIC) (Fraley and Raftery 2003).
- `flexclust`: This package provides an environment for partitioning cluster analysis with non-Euclidean distance measures using k -centroids cluster algorithms (KCCA) (Leisch 2006). There are also functions for deriving neighborhood graphs and image plots for visualization of partitions.

A comprehensive list of R packages for performing model-based or distance-based clustering is maintained by Friedrich Leisch and Bettina Grün and made available via the following CRAN task view: <https://CRAN.R-project.org/view=Cluster>.

Cluster Analysis of Market Basket Data

We next illustrate three different clustering procedures applied to a reference data set of shopping basket data. In doing so, we briefly outline the data used and the general approach of the selected procedure, then provide a walk-through for the corresponding R code required to perform the analyses and give some interpretation of the results. The three clustering procedures under consideration are: hierarchical clustering using the function `hclust()` and two prototype generating clustering methods: function `pam()` from the `cluster` package and function `kcca()` from the `flexclust` package.

Data Characteristics and Choice of Distance Measure

To illustrate the clustering methods, we use one month (30-days) of real-world point-of-sale transaction data from a local grocery outlet. The data set is included in the widely used R package `arules` and consists of an easy-to-handle set of 9835 retail transactions representing purchases in 169 different categories. The data come as a sparse matrix with each observation (row) representing a retail transaction and each column a binary variable with 1 denoting that a specific grocery category is present in the transaction and 0 otherwise. Thus, the row-wise sums indicate the number of categories purchased together in each transaction. A typical transaction is expressed as a list of categories such as: {tropical fruit, yogurt, coffee}, {citrus fruit, semifinished bread, margarine, ready soups}, and so on.

To obtain the data, in the R console, we add the `arules` package and activate the `Groceries` data set included in the library. To get a first visual impression of the proposed data set, we plot a histogram for the basket sizes, see Fig. 4, which

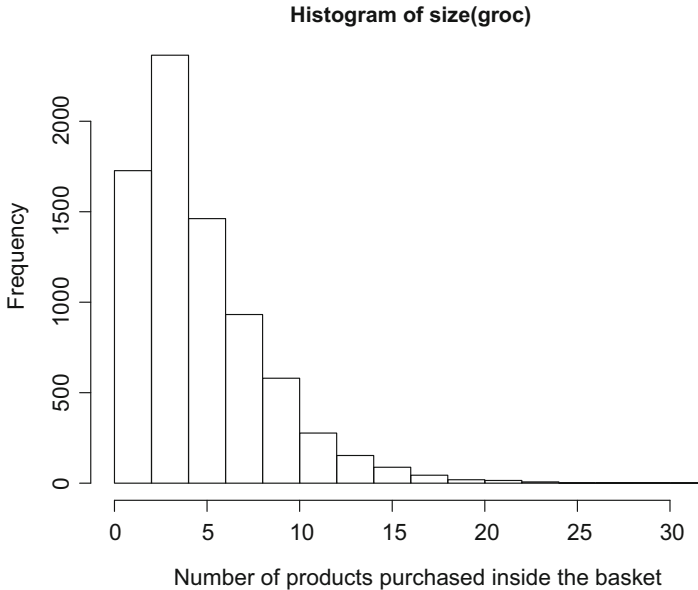


Fig. 4 Histogram of basket sizes

shows a right-skewed pattern we typically observe in supermarket transaction data: the majority of baskets are composed of only a few categories. Thus, the binary basket data are extremely sparse, with a mean basket size `mean(size(Groceries))` of only 4.41 categories per shopping trip. Note that throughout the chapter, we will omit the category “shopping bags” because the latter does not reflect any specific consumption preference but in a grocery shopping context merely serves to carry the bought items around. The `arules` library, the `Groceries` data and the `groc` variable indicated below will be the same throughout all the further examples and will be reported only once in the R code examples.

To get a better understanding about which categories are purchased most frequently, we can plot the frequency distribution of categories exceeding a threshold (support) of 5%. As we can see from Fig. 5, the most frequently purchased categories are typical grocery products such as whole milk, other vegetables, rolls/buns, soda, yogurt, etc.

```
library("arules")
data("Groceries")
groc <- Groceries[size(Groceries)>1,
  which(itemLabels(Groceries) != "shopping bags")]
hist(size(groc), xlab =
  "Number of products purchased inside the basket")
```

With the `itemFrequencyPlot` function, we plot the most frequent items.

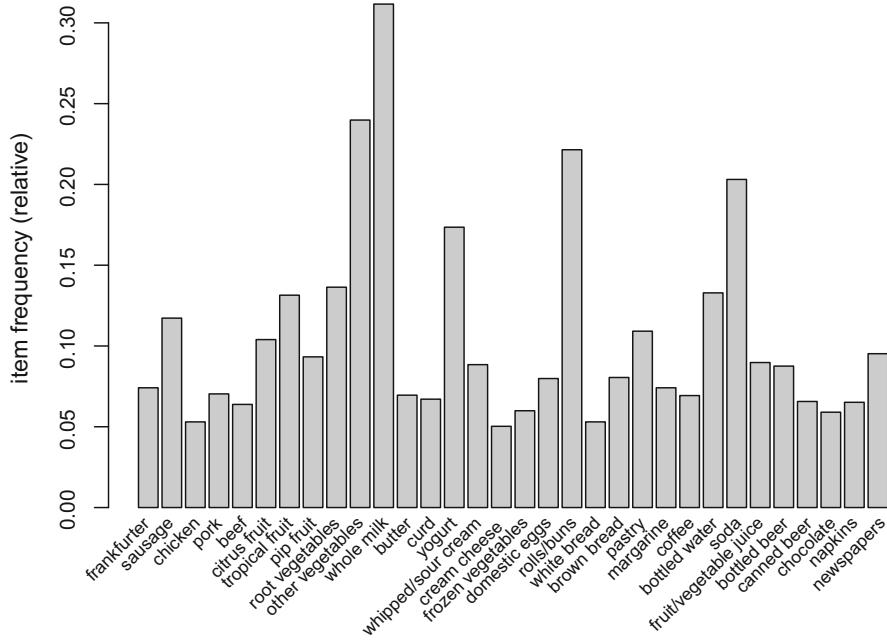


Fig. 5 Histogram of the most frequent items purchased

```
itemFrequencyPlot(groc, support = 0.05)
```

As we discussed in the previous section, in order to perform distance-based clustering, we need to specify a proximity measure which quantifies the distance between the objects to be clustered. Two aspects need to be considered in our present application. Firstly, an appropriate distance measure needs to adequately account for the data sparsity we observe for our market basket data at hand. Secondly, from a more substantive perspective, we are typically interested in finding groupings in the data which reflect jointly purchased categories, i.e., we aim at detecting complementary cross-purchase incidences. Thus, an asymmetric distance measure giving more weight to joint purchases than to common zeros (i.e., nonpurchases) is preferred in such situations. The previously discussed Jaccard coefficient (cf. Kaufman and Rousseeuw 1990) has such properties and is used in the present application.

For a given data set of market baskets $X^T = [x_n]$, $n = 1, \dots, N$ containing binary purchase incidences $x_n \in \{1, 0\}^J$ we can compute a frequency matrix $X^T \times X = C = [c_{ij}]$ of pairwise co-purchases of categories $i, j = 1, \dots, J$ and derive the Jaccard distance as follows (cf. Sneath 1957):

$$d_{ij} = 1 - \frac{c_{ij}}{c_{ii} + c_{jj} - c_{ij}}, \forall i, j = 1, \dots, J \quad (8)$$

Note that in the present context, the corresponding Jaccard similarity $s_{ij} = 1 - dij$ measures the percentage of joint purchases in all baskets which contained at least one of the two categories.

Hierarchical Clustering

In our first example, we employ hierarchical clustering to the described set of shopping basket data. In such a setting, the task is to explore subgroups of jointly purchased product categories based on pairwise co-purchase “affinities” across the categories included in the data set. Thus, we aim at detecting clusters of product categories which tend to be purchased together more often by the customers of the local supermarket.

As already mentioned in section “[Distance-Based Cluster Algorithms](#)”, the most popular hierarchical clustering method is agglomerative clustering, which can be performed by using the function `hclust()`. In this family of clustering methods, the agglomeration procedure initially considers each singleton object (here: product category) as a cluster and then, step by step, merges the objects iteratively into groups of clusters until one final cluster is generated. The merging mechanism is directed by (i) the proximity or (dis)similarity measure and (ii) a linking criterion.

In our example, we use the Jaccard distance as defined above. As a starting point, we thus compute a dissimilarity matrix $D = [d_{ij}]$ according to Eq. 8:

```
diss <- dissimilarity(groc[, itemFrequency(groc) > 0.02], method =
"jaccard", which = "items")
```

Note that for simplicity reasons and to keep the resulting tree structure easy to inspect, we only select categories which are contained in more than 2% of the retail transactions at hand. This is done by specifying `itemFrequency(Groceries) > 0.02`.

The choice of a linking criterion determines how the distance between two groups of observations is calculated during the agglomeration procedure. It uses the previously computed (dis)similarity measure and one of the many available methods for measuring the distance between two clusters or between a cluster and a singleton object. The most popular linkage methods are represented in Fig. 6. While single linkage uses the distance between the two closest elements of two clusters, complete linkage measures the distance between the two farthest or most distant elements in two sets. The average linkage criterion compromises between the two previously mentioned ones and takes the mean distance between the elements of each cluster. Ward’s method aims at minimizing the within-cluster variance and at each step merges the pair of clusters that leads to a minimum increase in total within-cluster variance after merging (cf. Kaufman and Rousseeuw 1990).

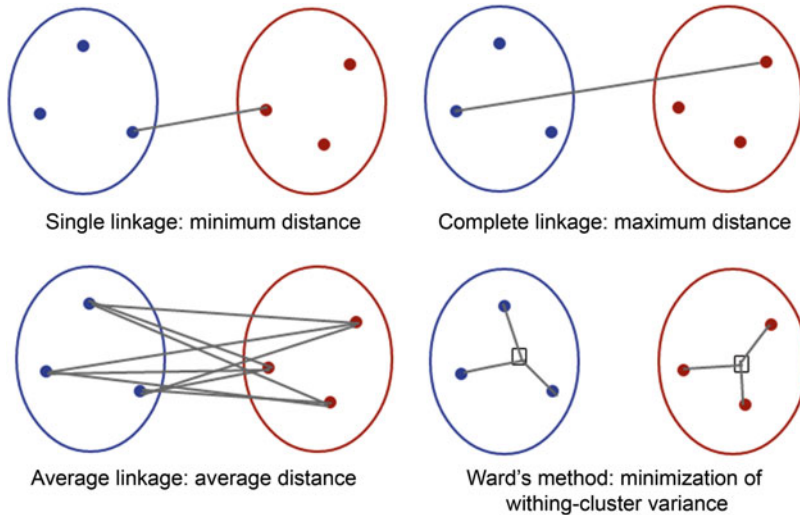


Fig. 6 A selection of popular linkage criteria in hierarchical clustering

Note that the choice of a cluster linkage method has a decisive impact on the resulting cluster solution. To illustrate this, consider the following two cases using single linkage clustering and the minimum variance method proposed by Ward (1963). Using the function `hclust()`, single linkage clustering can be performed as follows:

```
# Single linkage method
hc.single <- hclust(diss, method = "single")
plot(hc.single, cex=0.7)
abline(h = c(0.75, 0.80, 0.85, 0.90, 0.95),
col = "gray", lty = 3)
```

One common way to visualize the outcome of hierarchical clustering is by using a so-called dendrogram. The word dendrogram comes from the combination of two ancient Greek words: *déndron* (“tree”) and *grámma* (“written character, letter, that which is drawn.”) The analogy with a reversed tree is obvious, each leaf represents one object (in our case a product category), each branch represents one cluster at a certain point of the agglomeration process, and the root encompasses all the clusters. Notice that at a certain “height” of the cluster dendrogram, the branches are merged together. The height of the fusion, also known as the *cophenetic distance* (Farris 1969), is inversely proportional to the similarity of the objects. A common way to assess how the generated dendrogram reflects the data is to compare the cophenetic distances with the original distances by correlating them.

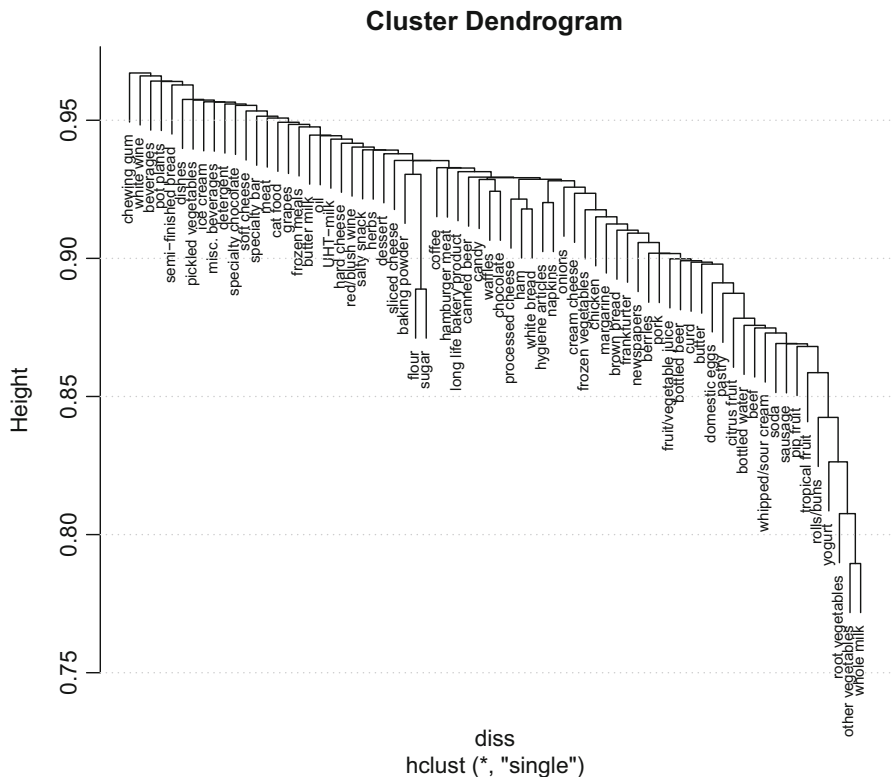


Fig. 7 Hierarchical agglomerative clustering applying single linkage method

A strong correlation indicates a good linking of the objects in the dendrogram (Saraçlı et al. 2013).

The dendrogram derived for our market basket data is obtained by the `plot()` command for object `hc.simple`. It is given in Fig. 7 and depicts a typical property inherent to single linkage clustering, namely that very “similar” categories (i.e., those which are purchased together very often, here whole milk, other vegetables, followed by root vegetables, yogurt, etc.) are merged at a very early stage of the agglomeration process and those categories which rarely appear in the same shopping baskets (chewing gum, beverages, etc.) at a later stage.

The single linkage criterion employs a “nearest neighbor” rule to merge sets and thus is able to reveal rather complex, elongated, or snake-shaped data structures (Kaufman and Rousseeuw 1990; Dolnicar et al. 2018). On the other hand, single linkage typically induces a chaining effect in the hierarchical agglomeration procedure, which can be clearly seen in Fig. 7 in the creation of long straggling “clusters”. This is due to the fact that objects are added sequentially to clusters, and at each stage, the “closest distant” (or most similar) object is merged with the already

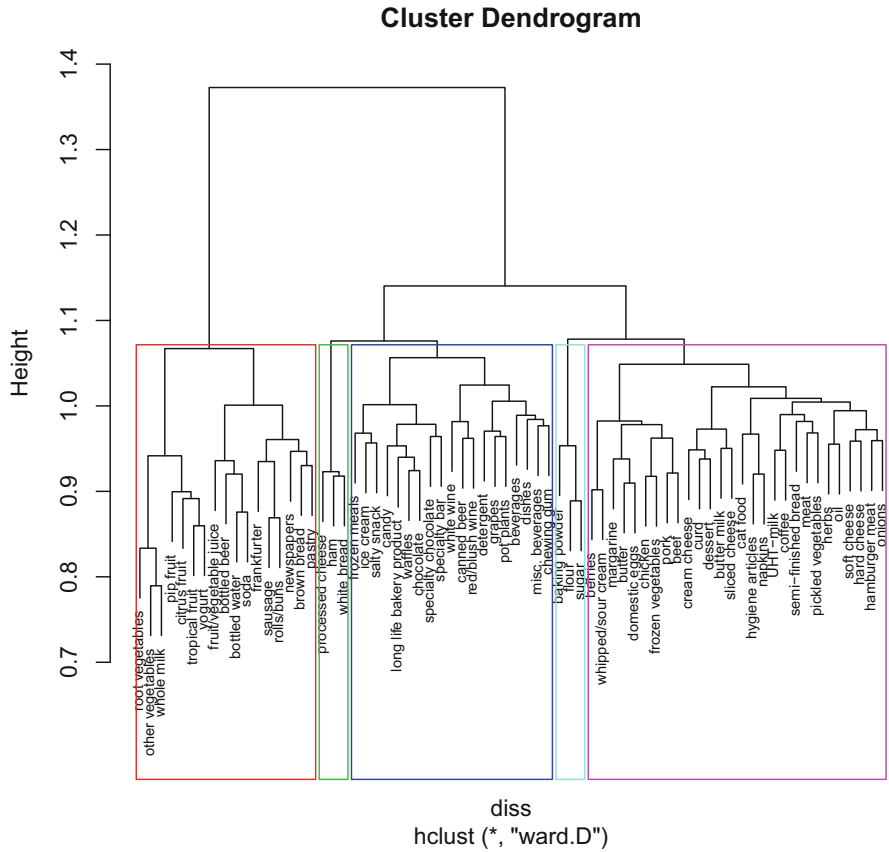


Fig. 8 Hierarchical agglomerative clustering applying Ward’s method

existing configuration. Because of this property, single linkage cluster is also sometimes used for outlier detection (i.e., those entities merged with the configuration towards the end of the agglomeration process).

In contrast, Ward’s linking method aims at forming minimum inner variance partitions. To achieve this, at each step, the pair of clusters which minimizes the incremental increase in within-cluster variance is merged. Ward’s method can be called in the R code by specifying the `ward.D` option. As we see from the dendrogram in Fig. 8, using this method, the categories are merged more evenly from the start, and thus we have a more balanced distribution of clusters.

```
# Ward linkage method
hc.wardd <- hclust(diss, method = "ward.D")
plot(hc.wardd, cex=0.7)
rect.hclust(hc.wardd, k = 5, border = 2:7)
```

As we move up from the leaves to the root, we notice that the branches get linked together at a variable height. The height of the linkage, indicated on the vertical axis, measures the (dis)similarity between two objects or clusters. The more we move towards the root of the dendrogram, the more dissimilar the merged objects are. In our illustrative example, we can see that several branches merge frequently co-purchased categories from the assortment of fresh products (e.g., vegetables, whole milk, fruits, yogurt) at the left-hand side of the tree, which are later linked with drinks (e.g., bottled beer, juices, water, soda, etc.) and a combination of categories associated with snacks (e.g., frankfurter, sausage, rolls, etc.) into one cluster marked by a red box. In the other (right-hand side) branch from the root, other distinct category combinations (such as salty and sweet snacks, candies and chocolate categories, etc.) are represented.

Notice that at any horizontal “cut” of the tree structure a specific cluster solution consisting of groups of product categories with internally more intense cross-category purchase relationships emerge. For example, in Fig. 8, we marked a solution with $k = 5$ clusters: three larger clusters and two smaller, each representing only three categories which are purchased together very frequently (i.e., {processed cheese, ham, white bread} and {baking powder, flour, sugar}). Such formations of “supra-categories” can be helpful for store managers to design shelf placements of categories within the store but also to consider the representation of categories in leaflets or promotional activities.

K-Medoid Clustering

Hierarchical clustering is a useful tool for data compression and visualization if the number of objects used for clustering is reasonably small. For example, this is the case if we are interested in analyzing the joint purchase affinities among the product categories J represented in a matrix X of shopping baskets as just illustrated above. However, as J increases, the dimensionality of the to-be-derived distance-matrix reaches computational limits and/or visualization of the dendrogram for the linkage procedure becomes intractable and interpretation cumbersome. The latter also applies to the task of deriving a segmentation of the shopping baskets or the households behind the observed retail transactions. From a substantive perspective, such a focus also implies moving away from studying category purchase interdependencies for a pooled set of transaction data. When we compress the shopping baskets into a smaller number of representing basket classes, we aim at finding a partition of the data set at hand with outstanding or more distinguished complementary cross-category purchase incidences within the detected classes (for more details, see Boztuğ and Reutterer 2008; Reutterer et al. 2006, 2017).

For such tasks, nonhierarchical or partitioning clustering is a feasible alternative. Formally, the task is to find a partition $P = \{P_1, \dots, P_K\}$ of the data set into a fixed number of K basket classes which fulfills the following objective function:

$$\sum_k \sum_{n \in P_k} d(x_n, c(x_n)) \rightarrow \min_{P, C} \quad (9)$$

where $C = (c_1, \dots, c_K)$ is a set of centroids or prototypes and $d(\cdot)$ a distance measure, such as the Jaccard distance in Eq. 8, we are using in the present application. In the clustering and classification literature, the “minimum dispersion criterion” in Eq. 9 is also known as the principal point or k -centroids problem (Jain and Dubes 1988; Leisch 2006). One important property of resolving Eq. 9 is that for any optimum configuration (P^*, C^*) , the condition $c^*(x_n) = \arg \min \{d(x_n, c_k), \forall k\}$ holds, which warrants that each basket x_n is mapped onto its minimum distant or closest centroid. With the notable exception of Ward’s method (which follows a similar objective function), this is in sharp contrast to the way most linkage procedures proceed in forming clusters. Instead of minimizing a global objective function, agglomerative hierarchical clustering aims at minimizing a distance function at each step of the cluster fusion but can result in a potentially suboptimal global solution.

Before we illustrate using a generic method for solving the k -centroids problem in the next section, we first employ an iterative, easy-to-implement, relocation-based heuristic proposed by Kaufman and Rousseeuw (1990) under the name Partitioning Around Medoids (PAM). Combined with clustering objective function (Eq. 9), this algorithm requires from the centroid to have the property $c_k \in \{x_n\}_{n \in P_k} \forall k$, i.e., the “medoid” is defined as the shopping basket which minimizes the mean distance with all other transactions in the same cluster P_k . This medoid property guarantees that the centroids are real shopping baskets, which tend to result in more robust cluster solutions in the presence of outliers and facilitates interpretation. On the other hand, PAM is suitable for relatively small- to medium-sized data sets, but this problem can be overcome by selecting randomly from the available data or following other resampling methodologies.

For clustering larger data sets using the medoid approach, one may use CLARA (Clustering LARGE Applications; see Kaufman and Rousseeuw 1990) or CLARANS (Clustering Large Applications based upon RANdomized Search; see Ng and Han 2002). The former does not use the entire data, but it randomly chooses multiple samples with fixed size and repeatedly applies PAM to each of these samples and selects the representative k -medoids. Afterwards, the objects in the data set are assigned to the closest medoid. CLARA finds the best clustering if the sampled medoid is among the best k -medoids by calculating the mean of the dissimilarities of the data to their closest medoid. CLARANS interprets the search space as a hypergraph, where each node represents a set of k -medoids. The algorithm randomly chooses a set of neighbor nodes as new medoids in an iterative manner. If the neighbor discovered is better than the previous one, a local optimum is discovered. The whole process is repeated until the whole graph is sufficiently explored and an optimal solution is found.

We apply the k -medoid partitioning to the `Groceries` data set by taking into account only transactions that contain at least two different product categories.

After this preselection, we are left with 7,676 transactions and 168 categories. To retain a dissimilarity matrix of moderate size, we randomly select 2,000 transactions and use the Jaccard coefficient as distance measure. The cluster solutions for a sequence of $K = 1, \dots, 8$ clusters are generated using the function `pam()`. Setting a seed value to secure reproducibility of the obtained results completes the following code for performing k -medoid clustering of the available shopping basket data:

```
library("cluster")
set.seed(42)
samp <- sample(groc, 2000)
diss <- dissimilarity(samp, method = "Jaccard")
clust <- lapply(1:8, function (x) pam(diss, k= x))
```

Determining a suitable number of clusters based on distance-based clustering methods is an open issue and the relevant literature offers a huge variety of “validity” metrics to assist the analyst with this task. Popular metrics include the cluster separation measure proposed by Davies and Bouldin (1979) or indices based on the agreement of repeated cluster solutions like the measures proposed by Rand (1971). An overview and detailed performance comparisons of alternative metrics for determining the number of clusters is provided by Milligan and Cooper (1985) or Dimitriadou et al. (2002). Among these heuristics is also the easy-to-use silhouette coefficient proposed by Rousseeuw (1987) which takes into consideration the discrepancies of the average within-cluster dissimilarities and the nearest data points of each neighboring cluster. Based on this heuristic, we opt for a solution of $K=5$ clusters (see also the discussion in Reutterer et al. 2007 or Reutterer et al. 2017).

As discussed before, the derived clusters should reflect classes of shopping baskets with more distinguished complementary cross-category purchase incidences within the detected basket classes. To explore these particular patterns, we select two exemplary clusters, namely cluster number 2 and 5, and characterize their specific properties using the function `itemFrequencyPlot()`. The resulting plots exhibited in Figs. 9 and 10 represent the relative purchase frequencies across categories in the complete data set as continuous lines and contrast them with the respective cluster-specific distributions. Note that for space and illustrative reasons, we include only categories which are present in at least 5% of transactions.

```
itemFrequencyPlot(samp[clust[[5]]$clustering == 2],
population = groc, support = 0.05)
```

```
set.seed(42)
inspect(samp[clust[[5]]$medoids[2]])
##      items
##      [1] {citrus fruit,
##          tropical fruit,
##          root vegetables,
```

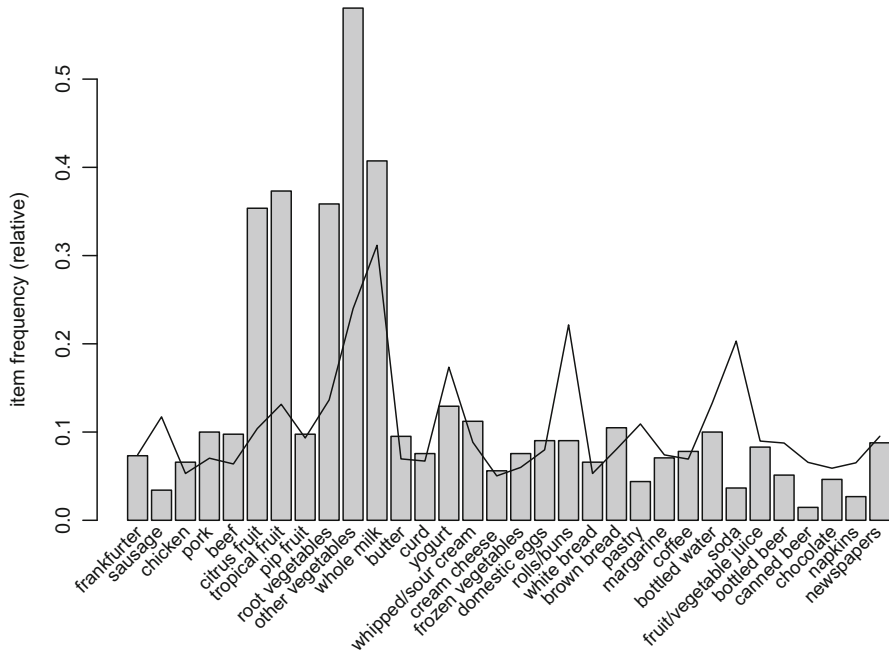


Fig. 9 Profile of relative purchase frequencies across categories in cluster 2

```
##          other vegetables,
##          whole milk}
```

Comparing the two clusters, they clearly point to considerable differences between the shopping baskets summarized by them. The transactions represented by cluster 2 are characterized by a shopping pattern with elevated purchase likelihood in fruits (citrus fruit, tropical fruit) and vegetables (root vegetables, other vegetables) categories as well as whole milk. In contrast, the purchase behavior behind cluster 5 transactions is clearly dominated by remarkably high purchase incidences in certain beverage categories (bottled water, soda, bottled beer) and only moderate class-conditional choice probabilities in the remaining categories.

The `inspect()` function returns us the respective medoid shopping baskets for these two clusters which confirm the above interpretation (i.e., {citrus fruit, tropical fruit, root vegetables, other vegetables, whole milk} for cluster 2 and {bottled water, soda, bottled beer} for cluster 5).

```
itemFrequencyPlot(samp[clust[[5]]$clustering == 5],
population = groc, support = 0.05)
```

```
set.seed(42)
```

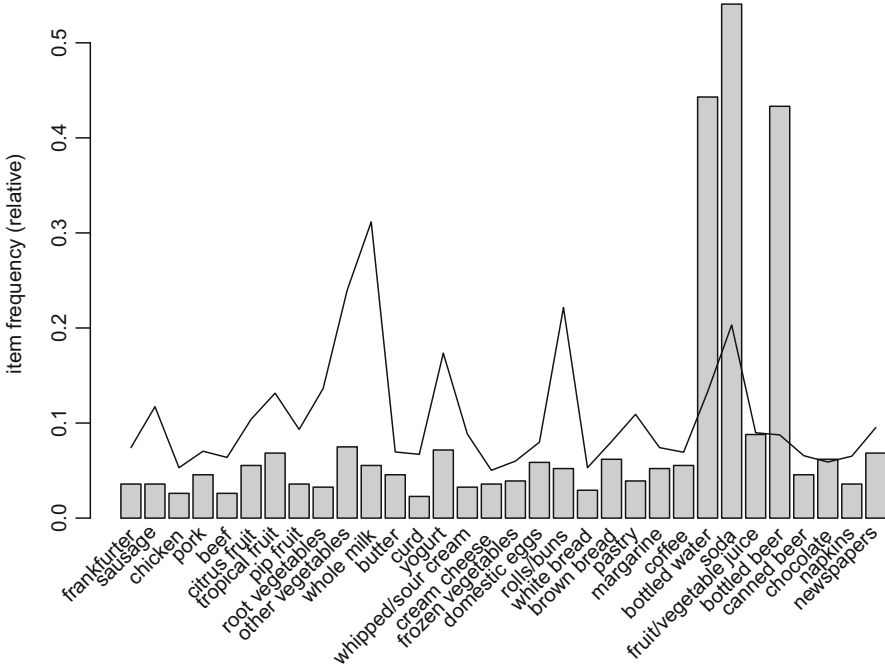


Fig. 10 Profile of relative purchase frequencies across categories in cluster 5

```
inspect(samp[clust[[5]]$medoids[5]])
##      items
##      [1] {bottled water,soda,bottled beer}
```

K-Centroids Cluster Analysis

A more flexible approach to solve the k -centroids problem introduced in the previous section is offered by the function `kcca()` in package `flexclust` (Leisch 2006). The R package `flexclust` includes a multitude of functions for various cluster algorithms. The main function in this package is `kcca()` which implements generalizations of k -means clustering for arbitrary distance measures. Thus, it can be used as a unifying partitioning framework for finding canonical centroids in both metric and nonmetric spaces including binary data (Hartigan and Wong 1979). Like the previous PAM application, once applied to the data structure at hand and using the same distance measure (i.e., Jaccard), this method also derives clusters of retail transactions with internally more homogeneous and pronounced cross-category dependencies.

To illustrate this approach, we use again the `Groceries` data set from the `arules` library and apply the same preselection for categories (threshold of being present in more than 2% of all transactions) as we did in the above application of hierarchical clustering. In order to be able to perform the necessary computations, we

transform the remaining data set into a matrix `grc` with logical values structure and then just add a zero to transform the result into a usable numerical values container.

The `flexclust` family provides, among others, the Jaccard coefficient option as a distance measure for binary data and the `kcca()` function returns a set of real-valued centroids representing class conditional expectations which are directly accessible for the interpretation of the derived cluster solution. The issue of choosing a suitable number of clusters follows the same line of arguments as discussed in the previous subsection and is usually addressed by generating cluster solutions for a sequence of increasing K numbers of clusters and applying some internal “validity” metrics or by systematically studying the stability of alternative solutions using ensemble clustering techniques (Hornik 2005). Here, to exemplify and simplify our illustration, we use five clusters like in the above illustration of the PAM method (`num_clusters = 5`).

One way to inspect the separation of the derived clusters is by visualizing a lower-dimensional representation of the cluster solution. This can be accomplished by applying a data projection method, such as principal component analysis to the data set at hand. We note that principal components or factor analysis is problematic for non-Gaussian (here binary) data, but following Leisch (2006) we consider using it as appropriate for the mere purpose as a simple and easy-to-use data projection device used to visualize a cluster solution (we are not interested in the underlying interpretation of the derived dimensions). Other appropriate methods would be, for example, correspondence or homogeneity analysis.

Combining the results of `kcca()` and `prcomp()`, the projection of the data points together with indicators of their cluster membership on the first two dimensions can be done by using the `plot()` function. In Fig. 11, the centroids of the five clusters solution are plotted as numbers and connected by a neighborhood graph, which thickness represents the degree of connectedness. Even though the projected clusters apparently overlap, the scatter plot suggests an underlying structure of five diagonally separated groups of data points.

```
library("ggplot2")
library("flexclust")
Gr <- groc[, itemFrequency(groc) > 0.02]
grc <- as(Gr, "matrix")
grc <- grc + 0

# flexclustControl object holds the "hyperparameters"

fxc <- new("flexclustControl")
lc <- list(iter.max=500, tol=0.001, verbose=0)
fxc <- as(lc, "flexclustControl")
fc_seed <- 100
num_clusters <- 5

set.seed(fc_seed) }
```

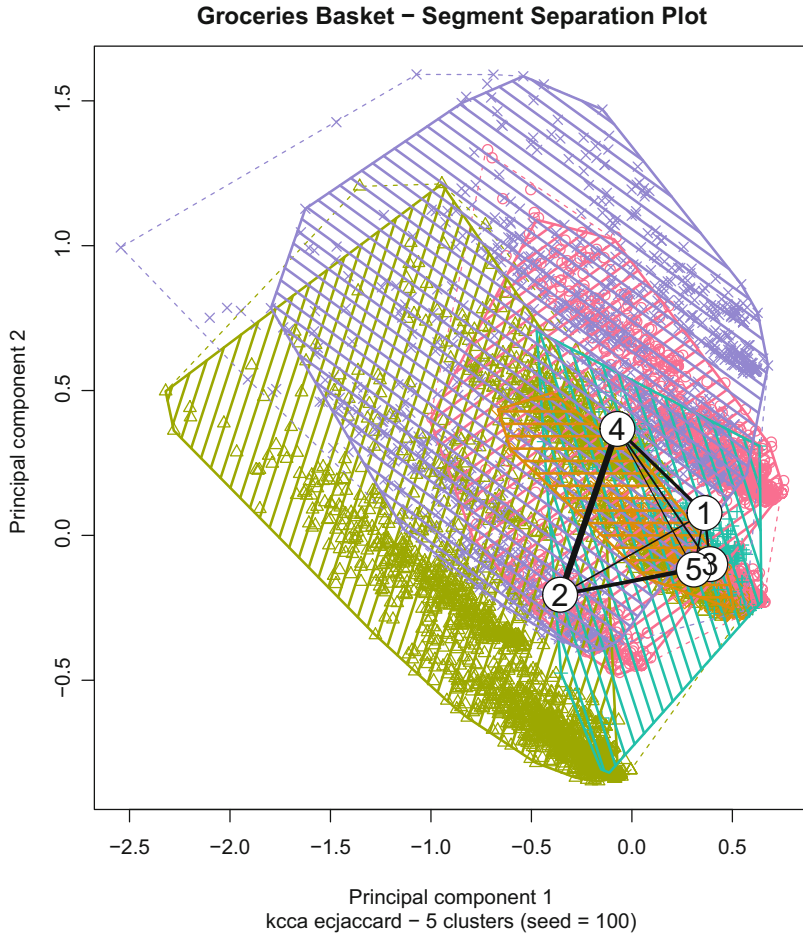


Fig. 11 Neighborhood plot of the Groceries data using the Jaccard distance on the two principal components

```
# verbose > 0 will show iterations
vol.cl <- kcca(grc, k = num_clusters, save.data = TRUE,
control = fxc, family = kccaFamily("ejaccard"))
main_text1 <- "Groceries Basket"
sub_text <- "kcca ejaccard - 5 clusters (seed = 100)"

# plot on first two principal components
vol.pca <- prcomp(grc)
plot(vol.cl, data = grc, project = vol.pca, which = 1:2,
main = paste0(main_text1, " - Segment Separation Plot"),
xlab = "Principal component 1",
```

```
ylab = "Principal component 2",
points = TRUE, hull.args = list(density=10),
sub = sub_text)

barchart(vol.cl, strip.prefix = "# ",
shade = TRUE, layout = c(vol.cl@k, 1),
main = paste0(main_text1, " - Cluster Profile Plot"),
which = hc.wardd$order)
```

Similar to the cluster-specific barplots for the relative category purchase frequencies of the k -medoid cluster solution, the `barchart()` function allows for a graphical representation of the cluster solution contained in a `kcca()` object and thus helps interpreting the findings (see Dolnicar et al. 2014). The argument `shade = TRUE` detects and displays the marker variables in colors. The argument `which` defines the order of the variables. In our case, we chose the order of the product categories such that it corresponds to the order in the dendrogram represented in Fig. 8.

Figure 12 shows in the header the absolute number (i.e., the number of shopping baskets assigned to the respective cluster) and the percentage size of each cluster. The individual barplots help to compare the overall against the cluster-specific centres or mean values per category, which in the present context can be interpreted as the respective percentage of transactions containing a specific category. The line with the full dot represents the relative purchase frequencies over the complete sample. Thus, as we already know from above, the categories with highest shares in the shopping baskets are whole milk, other vegetables, rolls/buns, etc. The bars represent the respective within-cluster purchase shares for each category. They are colored if the difference to the overall mean exceeds a certain threshold value, the bar has a gray contour if this difference is not relevant for interpretation but might be a relevant characteristic of the cluster.

From the visual inspection of Fig. 12, it becomes obvious that the five derived basket classes differ in their basket composition from the overall “average” shopping basket by only a few categories, which makes them distinctive from each other. For example, the shopping baskets represented by cluster 4 are characterized by an outstanding share of rolls/buns and clearly above average purchase incidences in the sausage and frankfurter categories (the three categories together representing typical items demanded for making snacks). A slightly different variation of this cluster is represented by segment 3, in which drinks are represented by the bottled beer and food by sausages and bread. In contrast, the 36% of shopping baskets represented by cluster 2 contain typical grocery shopping categories, such as whole milk, other vegetables, root vegetables, butter, and domestic eggs above average.

From a managerial perspective, knowledge of such behavioral segments is an important prerequisite for designing customized target marketing actions (see Reutterer et al. 2006). For example, categories with distinguished purchase propensities within a specific segment (such as beer in segment 3 or water and soda

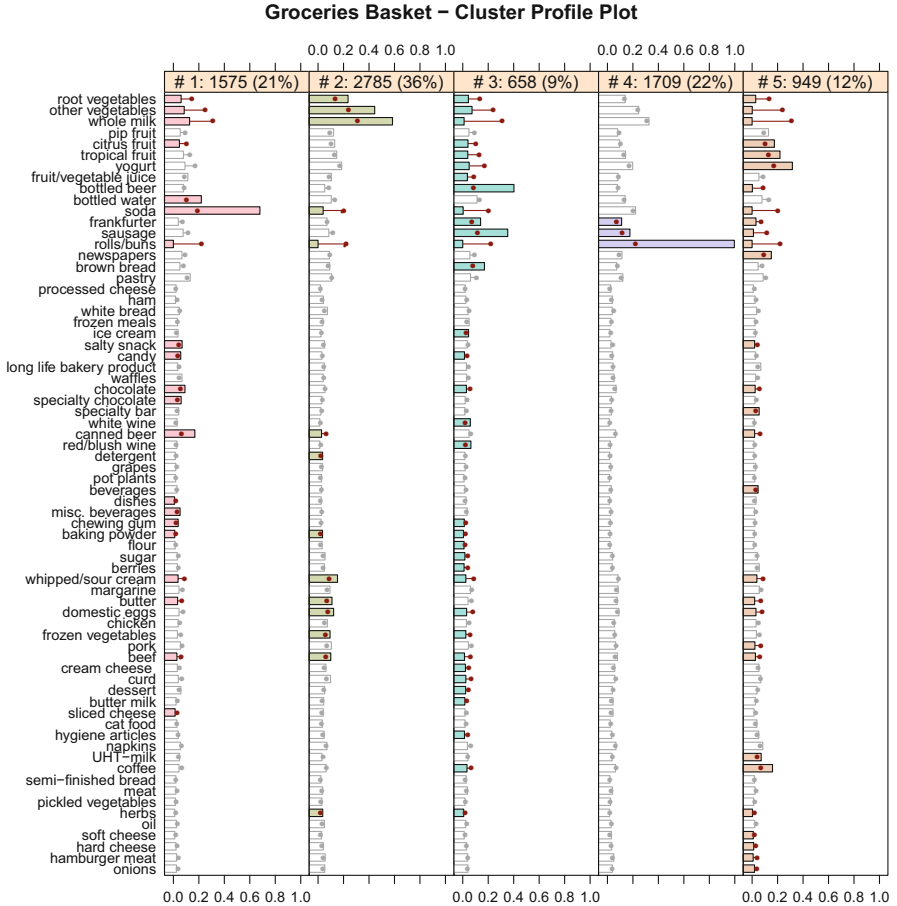


Fig. 12 Segment profile plot for the five-segment solution based on the *K*-centroids algorithm

in segment 1) are promising candidates for deriving targeted promotions to increase store traffic. On the other hand, direct marketers could also aim at promoting categories which are systematically underrepresented in certain segments by cross-promoting them in combination with certain “draw” categories; for example, in segment 3, customers could be stimulated to purchase more vegetables or milk by cross-promoting bottled beer in a way similar to “get a bottle of beer free if you purchase one liter of milk” (Breugelmans et al. 2010; Dréze and Hoch 1998).

Conclusions

As demonstrated in the previous section, the cluster solutions and corresponding interpretations vary considerably depending on the applied clustering procedure. This clearly reflects the exploratory nature of cluster analysis which implies that there is no single “correct” or natural solution for a specific data set at hand. The achieved data compression effect and thus the specific data structure uncovered by a particular approach to cluster analysis rather depends on a number of factors under the analyst’s control. The main factors are the choice of the cluster analysis procedure, the distance measure, the number of clusters, and the data mode to be clustered.

While there is some more or less sophisticated methodological guidance available to decide on some of these factors, it merely depends on the specific research objective for others. For example, there is extensive research on the determination of the number of clusters (Milligan and Cooper 1985; Dimitriadou et al. 2002) or on the stability properties of cluster solutions (Hornik 2004; Hornik 2005). However, the choice of an appropriate clustering procedure or the specific focus on a particular data mode to be compressed is a decision that is determined by the substantive research question or the analysts subjective judgement. Generally speaking, hierarchical linkage methods have their merits when the task is to explore differences and commonalities among objects on a more fine-granular level (e.g., by “zooming in” the representing dendrogram) and the number of clusters is not fixed a priori. On the other hand, partitioning methods like k -centroid clustering tend to be the preferred method when the analyst aims at compressing larger data sets into a smaller number of representing points (centroids or prototypes), each characterizing a subset of the data as accurately as possible and simultaneously are distinctive from the other cluster centroids.

Using a widely used benchmark set of market basket data, we demonstrated in this chapter that the analytical task of exploring the specific structures of cross-category purchase relationships can be achieved by reducing the dimensionality of the data set using hierarchical clustering (i.e., by analyzing the (dis)similarity structure among the variables of the data matrix). We have also shown that the derived structural patterns strongly depend on the specific method applied and they would also vary if we chose a different distance metric. On the other hand, compressing the number of baskets using nonhierarchical partitioning methods results in a set of specific classes of shopping baskets with distinguished complementary cross-category purchase incidences within the classes. The latter effect is obtained by choosing an appropriate distance measure (in our case Jaccard distances) and the partitioning as well as the interpretation of the classes which would be different for other distance metrics. All these examples demonstrate the generic idea behind cluster analysis as an exploratory data compression tool. This “idea” is to uncover structure in the data, which in the case of distance-based clustering is based on a specific conceptual understanding of quantifying proximity between data points.

The field of cluster analysis is a very dynamic one and the analysts’ toolbox is constantly growing. New methods which aim to cope with the specific challenges of

today's data-rich environments are emerging. Such challenges are not limited to but include real-time (online) updating of cluster solutions for data streams (Ghesmoune et al. 2016), clustering of very high dimensional data sets (Strehl and Ghosh 2003), bootstrap aggregated clustering (Dolnicar and Leisch 2003), and other ensemble methods to improve the quality and robustness of cluster solutions (Hornik 2004; Hornik 2005); see also the extension package `clue` for R (R Core Team 2019) which provides a computational environment for cluster ensembles.

Modern clustering methods also comprise a variety of unsupervised machine learning methods (for an overview, see Hastie et al. 2009). Marketing applications of such machine learning methodologies include the employment of vector quantization techniques (e.g., Decker 2005; Reutterer et al. 2006), neural networks (e.g., Hruschka and Natter 1986; Mazanec 1999; Reutterer and Natter 2000), topic models for "soft-clustering" unstructured texts (e.g., Tirunillai and Tellis 2014; Büschken and Allenby 2016), or graph partitioning methods (Netzer et al. 2012).

Cross-References

- ▶ [Finite Mixture Models](#)
- ▶ [Market Segmentation](#)

References

- Adams, R. A., & Fournier, J. J. (2003). *Sobolev spaces* (Pure and applied mathematics) (Vol. 140). Amsterdam: Elsevier.
- Aldenderfer, M. S., & Blashfield, R. K. (1984). *Cluster analysis*. Beverly Hills: Sage.
- Anderberg, M. R. (1973). *Cluster analysis for applications*. New York: Academic.
- Arabie, P., & Lawrence, J. H. (1994). Cluster analysis in marketing research. In R. P. Bagozzi (Ed.), *Advanced methods of marketing research* (pp. 160–189). Cambridge, MA: Blackwell.
- Arabie, P., & Lawrence, J. H. (1996). *An overview of combinatorial data analysis. Clustering and classification* (pp. 5–63). Singapore: World Scientific.
- Arabie, P., Carroll, J. D., DeSarbo, W., & Wind, J. (1981). Overlapping clustering: A new method for product positioning. *Journal of Marketing Research*, 28(3), 310–317.
- Bock, H. H. (1974). *Automatische Klassifikation*. Göttingen: Vandenhoeck & Ruprecht.
- Boztuğ, Y., & Reutterer, T. (2008). A combined approach for segment-specific market basket analysis. *European Journal of Operational Research*, 187(1), 294–312.
- Breugelmans, E., Boztuğ, Y., & Reutterer, T. (2010). A multistep approach to derive targeted category promotions. Working paper series of the Marketing Science Institute, MSI report no. 10-118, Cambridge, MA.
- Büschken, J., & Allenby, G. M. (2016). Sentence-based text analysis for customer reviews. *Marketing Science*, 35(6), 953–975.
- Cattell, R. B. (1943). The description of personality: Basic traits resolved into clusters. *Journal of Abnormal and Social Psychology*, 38(4), 476–506.
- Chapman, C., & McDonnell Feit, E. (2019). *Segmentation: Clustering and classification. R for marketing research and analytics* (pp. 299–338). New York: Springer.
- Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2), 224–227.

- Decker, R. (2005). Market basket analysis by means of a growing neural network. *The International Review of Retail, Distribution and Consumer Research*, 15(2), 151–169.
- DeSarbo, W. S., Ajay, K. M., & Lalita, A. M. (1993). Non-spatial tree models for the assessment of competitive market structure: An integrated review of the marketing and psychometric literature. In J. Eliashberg & G. L. Lilien (Eds.), *Handbooks in operations research and management science* (Vol. 5, pp. 193–257). Amsterdam: Elsevier.
- Dimitriadou, E., Dolničar, S., & Weingessel, A. (2002). An examination of indexes for determining the number of clusters in binary data sets. *Psychometrika*, 67(1), 137–159.
- Dolnicar, S., & Leisch, F. (2003). Winter tourist segments in Austria: Identifying stable vacation styles using bagged clustering techniques. *Journal of Travel Research*, 41(3), 281–292.
- Dolnicar, S., Grün, B., Leisch, F., & Schmidt, K. (2014). Required sample sizes for data-driven market segmentation analyses in tourism. *Journal of Travel Research*, 53(3), 296–306.
- Dolnicar, S., Grün, B., & Leisch, F. (2018). *Market segmentation analysis. Understanding it, doing it, and making it useful*. Singapore: Springer.
- Drèze, X., & Hoch, S. J. (1998). Exploiting the installed base using cross-merchandising and category destination programs. *International Journal of Research in Marketing*, 15(5), 459–471.
- Everitt, B. S., Landau, S., Leese, M., & Stahl, D. (2011). *Cluster analysis: Wiley series in probability and statistics*. New York: Wiley.
- Farris, J. S. (1969). On the cophenetic correlation coefficient. *Systematic Zoology*, 18(3), 279–285.
- Fraley, C., & Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458), 611–631.
- Fraley, C., & Raftery, A. E. (2003). Enhanced model-based clustering, density estimation, and discriminant analysis software: MCLUST. *Journal of Classification*, 20(2), 263–286.
- Frühwirth-Schnatter, S. (2006). *Finite mixture and Markov switching models*. New York: Springer Science & Business Media.
- Ghesmoune, M., Lebbah, M., & Azzag, H. (2016). State-of-the-art on clustering data streams. *Big Data Analytics*, 1(13), 1–27.
- Grover, R., & Srinivasan, V. (1987). A simultaneous approach to market segmentation and market structuring. *Journal of Marketing Research*, 24, 139–153.
- Hartigan, J. A. (1975). *Clustering algorithms*. New York: Wiley.
- Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society: Series C: Applied Statistics*, 28(1), 100–108.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). Unsupervised learning. In *The elements of statistical learning* (pp. 485–585). New York: Springer.
- Hennig, C., Meila, M., Murtagh, F., & Rocci, R. (2015). *Handbook of cluster analysis*. Boca Raton/London/New York: CRC Press.
- Hornik, K. (2004). Cluster ensembles. In C. Weihs, W. Gaul (Eds.), *Classification – The ubiquitous challenge. Proceedings of the 28th annual conference of the Gesellschaft für Klassifikation E.V* (pp. 65–72). Heidelberg: University of Dortmund/Springer.
- Hornik, K. (2005). A clue for cluster ensembles. *Journal of Statistical Software*, 14(12), 1–25.
- Hruschka, H. (1986). Market definition and segmentation using fuzzy clustering methods. *International Journal of Research in Marketing*, 3(2), 117–134.
- Hruschka, H., & Natter, M. (1986). Comparing performance of feedforward neural nets and K-means for cluster-based market segmentation. *European Journal of Operational Research*, 114(2), 346–353.
- Jain, A. K., & Dubes, R. C. (1988). *Algorithms for clustering data*. Upper Saddle River: Prentice-Hall.
- Kaufman, L., & Rousseeuw, P. J. (1990). *Finding groups in data: An introduction to cluster analysis*. Hoboken: Wiley.
- Leisch, F. (2006). A toolbox for k-centroids cluster analysis. *Computational Statistics & Data Analysis*, 51(2), 526–544.

- MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1(14), 281–297.
- Manchanda, P., Ansari, A., & Gupta, S. (1999). The “shopping basket”: A model for multicategory purchase incidence decisions. *Marketing Science*, 18(2), 95–114.
- Mazanec, J. A. (1999). Simultaneous positioning and segmentation analysis with topologically ordered feature maps: A tour operator example. *Journal of Retailing and Customer Services*, 6(4), 219–235.
- Mazanec, J. A., & Strasser, H. (2000). *A nonparametric approach to perceptions-based market segmentation: Foundations* (Vol. 1). Wien: Springer.
- McLachlan, G. J., & Basford, K. E. (1988). *Mixture models: Inference and applications to clustering*. New York: Marcel Dekker.
- Mild, A., & Reutterer, T. (2003). An improved collaborative filtering approach for predicting cross-category purchases based on binary market basket data. *Journal of Retailing and Consumer Services*, 10(3), 123–133.
- Milligan, G. W., & Cooper, M. C. (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2), 159–179.
- Mooi, E., Sarstedt, M., & Mooi-Reci, I. (2018). Data. In *Market research* (pp. 27–50). Singapore: Springer.
- Netzer, O., Feldman, R., Goldenberg, J., & Fresko, M. (2012). Mine your own business: Market-structure surveillance through text mining. *Marketing Science*, 31(3), 521–543.
- Ng, R. T., & Han, J. (2002). CLARANS: A method for clustering objects for spatial data mining. *IEEE Transactions on Knowledge and Data Engineering*, 14(5), 1003–1016.
- Punj, G., & Stewart, D. W. (1983). Cluster analysis in marketing research: Review and suggestions for application. *Journal of Marketing Research*, 20(2), 134–148.
- R Core Team. (2019). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. Vienna: R Development Core Team.
- Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66, 846850.
- Rao, V. R., & Sabavala, D. J. (1981). Inference of hierarchical choice processes from panel data. *Journal of Consumer Research*, 8(1), 85–96.
- Reutterer, T. (1998). Competitive market structure and segmentation analysis with self-organizing feature maps. In P. Anderson (Ed.), *Proceedings of the 27th EMAC conference. Track 5: Marketing research* (pp. 85–105). Stockholm: EMAC.
- Reutterer, T. (2003). Bestandsaufnahme und aktuelle Entwicklungen bei der Segmentierungsanalyse von Produktmärkten. *Journal für Betriebswirtschaft*, 53(2), 52–74.
- Reutterer, T., & Natter, M. (2000). Segmentation-based competitive analysis with MULTICLUS and topology representing networks. *Computers & Operations Research*, 27(11–12), 1227–1247.
- Reutterer, T., Mild, A., Natter, M., & Taudes, A. (2006). A dynamic segmentation approach for targeting and customizing direct marketing campaigns. *Journal of Interactive Marketing*, 20(3–4), 43–57.
- Reutterer, T., Hahsler, M., & Hornik, K. (2007). Data mining und marketing am beispiel der explorativen warenkorbanalyse. *Marketing ZFP*, 29(3), 163–180.
- Reutterer, T., Hornik, K., March, N., & Gruber, K. (2017). A data mining framework for targeted category promotions. *Journal of Business Economics*, 87(3), 337–358.
- Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20, 53–65.
- Russell, G. J., & Petersen, A. (2000). Analysis of cross category dependence in market basket selection. *Journal of Retailing*, 76(3), 367–392.
- Russell, G. J., Ratneshwar, S., Schocker, A. D., Bell, D., Bodapat, A., Degeratu, A., Hildebrandt, L., Kim, N., Ramaswami, S., & Shankar, V. H. (1999). Multiple-category decision-making: Review and synthesis. *Marketing Letters*, 10(3), 319–332.

- Saraçlı, S., Doğan, N., & Doğan, I. (2013). Comparison of hierarchical cluster analysis methods by cophenetic correlation. *Journal of Inequalities and Applications*, 2013(1), 203.
- Sneath, P. H. (1957). Some thoughts on bacterial classification. *Journal of General Microbiology*, 17, 184–200.
- Sokal, R. R., & Sneath, P. H. A. (1963). *Principles of numerical taxonomy* (A series of books in biology). San Francisco: W.H. Freeman.
- Späth, H. (1977). *Cluster-analyse – Algorithmen zur Objektklassifizierung und Datenreduktion* (2nd ed.). München/Wien: Oldenbourg Wissenschaftsverlag.
- Srivastava, R. K., Leone, R. P., & Shocker, A. D. (1981). Market structure analysis: Hierarchical clustering of products based on substitution-in-use. *Journal of Marketing*, 45(3), 38–48.
- Srivastava, R. K., Alpert, M. I., & Shocker, A. D. (1984). A customer-oriented approach for determining market structures. *Journal of Marketing*, 48(2), 32–45.
- Strasser, H. (2000). Reduction of complexity. In J. Mazanec & H. Strasser (Eds.), *A nonparametric approach to perceptions-based market segmentation: Foundations* (pp. 99–140). Wien/New York: Springer.
- Strehl, A., & Ghosh, J. (2003). Relationship-based clustering and visualization for high-dimensional data mining. *INFORMS Journal on Computing*, 15(2), 208–230.
- Struyf, A., Hubert, M., & Rousseeuw, P. (1996). Clustering in an object-oriented environment. *Journal of Statistical Software*, 1(4), 1.
- Tirunillai, S., & Tellis, G. J. (2014). Mining marketing meaning from online chatter: Strategic brand analysis of big data using Latent Dirichlet allocation. *Journal of Marketing Research*, 51(4), 463–479.
- Titterton, D. M., Smith, A. F. M., & Makov, U. E. (1985). *Statistical analysis of finite mixture distributions*. Chichester: Wiley.
- Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58, 236–244.
- Wedel, M., & Kamakura, W. A. (2000). *Market segmentation – Conceptual and methodological foundations*. New York: Springer.