Tilburg University
School of Humanities
Data Science: Business and Governance

A cross-category method comparison for doing market basket analysis

Jasmina Kandelaar
318497

Master thesis
July 2016

Supervisor: Max Louwerse
Second reader: Pieter Spronck

Table of contents

Preface

After the first year communication and information sciences, I choose for the premaster Communication design. The year after that I learned about automatization of tasks with a computer, I had my first programming course and I found out that I liked to develop myself further in datamining. I took relevant courses and at the end of that year the announcement of a new study 'Data Science: Business and Governance' got to my attention. This study made me more and more enthusiastic about the different methods that can be used for analysis and what kind of results you get from different methods. This thesis is written to conclude the master Data Science. Instead of focussing on the relationships of products based on buying behaviour, what was the first idea, I decided to compare the results of different methods in their similarity with the actual relationships in the data.

Acknowledgments

There are several persons that have contributed during this research and therefore I want to thank them. My supervisor Max Louwerse who followed me during the whole process and with who I discussed the steps that were taken during the writing of my thesis. SAP as a company and Jan Theodoor Wiltschek and Dave Veerman who are working at SAP and made it possible to do this project and to use the dataset and SAP products. Also from SAP Marcel de Bruin, who helped me with technical tasks and questions. Mariangela Rossi who also did her thesis at SAP and was always there for me during the project. I also want to thank my family and friends who always supported me and in particular Muriël Nooder.

Abstract

In order to have the right product on the right place at the right time without having an overflowing warehouse, insight in purchase patterns and the relationships between products is necessary. The use of datamining techniques in order to get a deeper insight to base decisions on, results in better assortment and replenishment decisions, faster inventory cycle times, less inventory carrying costs and more revenue. Techniques to do a market basket analysis with, are commonly forms of association rule mining and sometimes clustering. Those techniques are often not compatible with today's diversity of the assortment of retailers. If insight in the relationships between products of a small selection of the assortment or in a small timespan is to be found, these methods are applicable. However, in order to get an overview of the relationships between all products, methods that can deal with more information are needed. Eleven methods that can be used to perform a market basket analysis with are discussed in this paper. The theoretical discussion of those methods, indicated that methods of the category network analysis are better suited to get insight in product relationships existing in the assortment, compared with the traditional used methods. Point-of-sale data from a discount pricing retailer is used in order to test not only the applicability but also the similarity of the results with real co-occurrences in the dataset. An existing evaluation measure that can be used to compare the accuracy of the results of the different datamining methods could not be found. Therefore, two comparison methods were created to make a cross-category method comparison possible. During the experiments the results of k-means clustering, the Gaussian Mixture Model, the Louvain method and the Clique Perlocation Method were compared. The product relationships generated with the Louvain method, a network analysis technique, were like expected most similar with the real co-occurrences of the products in the data. The results were also the easiest to interpret and most precise since the relationship strengths between products and groups of products were visualized. The group assignments of the products were almost the same as a division of products based on semantic relatedness and similar usage situations in the real world, which makes the results ecological valid. More research about the use of network analysis techniques when doing a market basket analysis is preferable for this reason. In particular a comparison between the performance of the Louvain method and the combined use of the Community-Affiliation Graph Model and the Cluster Affiliation Model for Big Networks. Based on theoretical grounds, it was expected that the last two mentioned methods would perform best. This could not be tested because the resources needed to implement them were not available. Given the motivation of this study, research in order to improve traditional market basket analysis methods in their applicability for wider product assortments is also needed. For association rule mining methods this means minimizing the amount of association rules that are generated and maximizing the amount of information they hold. For clustering methods this means optimizing distance measures in such a way that they can handle highly dimensional extremely sparse binary data, that represents the whole assortment and contains information about the products that are bought during the same transaction.

The importance of customer demand in the retail business

Retailers buy products in large quantities from manufacturers and sell them in small quantities to the general public. Because of the central role of the customer, it is important that the width and depth of the product assortment, the availability of products and the prices, meet the customer demands. Adapting on insights of customer demand increases customer profitability and enhances customer satisfaction and loyalty (Zentes, Morschett & Schramm-Klein, 2011). From a retailers perspective, it is therefore important to have the right stock, at the right place, at the right moment. This goal implicitly also prevents out of stocks in one store while having excess in another and assortment mismatches. At the same time, inventory cycle times must be as fast as possible and inventory carrying and logistic costs must be minimized. Involving activities to obtain these objectives are among other things in-store inventory management, shelf space management and replenishment (Zentes, Morschett & Schramm-Klein, 2011).

Descriptive and predictive datamining techniques can provide knowledge about purchase patterns and sales predictions. This knowledge can be used to make decisions in a data-driven manner. Research (Brynjolfsson, Hitt & Kim, 2011) shows that data-driven decision making leads to higher output, productivity, asset utilization, return on equity and market value, compared with what would be expected given other investments and information technology usage. For this reason, the number of companies that base their decision making (partly) on data and business analytics increases (Shmueli, Patel & Bruce, 2011).

This paper presents a study in which the best method is sought that can be used to identify relationships between products in terms of frequently bought together. The best method identifies product relationships that are most similar with co-occurrences present in the dataset. There are a variety of methods that can be used to identify relationships among products. The choice of the method enormously influences the result. Because of this reason, different methods will be discussed, implemented and evaluated in their performance in product relationship identification.

Influences on purchase patterns

Purchase patterns contain information about which products are (going to be) bought at which point of time, in combination with which other products and under what circumstances. This information is often generated in the form of business rules. However, these rules are not static, purchase patterns are temporal of nature. Customer demand is influenced by the availability of other products, seasonality, events, price, marketing actions, or gradual or rapid or local or global changes in customer preferences for instance (Žliobaitė, Bakker & Pechenizkiy, 2009).

Only the customers' decision to buy a product from a certain product category can already influence purchases from other product categories. Those cross-category purchase decisions can be categorized as seasonal, co-incidence or complementary. Seasonal decisions are not only influenced by the season. They are repeating periodical patterns, that can also be influenced by (national, religious, school) holidays. A religious holiday like fasting can influence sales for instance (Žliobaitė, Bakker & Pechenizkiy, 2009). Purchase decisions are also influenced by the weather. This is even the case for big purchases like the choice of car, or the added hedonic value of central air and a swimming pool with a house choice (Busse, Pope, Pope & Silva-Risso, 2012). Co-incidence decisions compromise all the 'residual' reasons, like similar purchase cycles (like beer and diapers) or other unobserved factors (Manchanda, Ansari & Gupta, 1999). The decision is complementary if marketing actions concerning a product influence purchases from another category (Manchanda, Ansari & Gupta, 1999). Marketing actions can be event related, a promotion or discount in price.

Complementary decisions can be negative/substitutes or positive/complementary (Manchanda, Ansari & Gupta, 1999). With a substitute decision a product from another category, or from the same category but another brand, is not bought because of the purchase of a particular product. An example is if an customer goes to the retailer to buy a memory stick because some computer files have to be saved, but purchases an external hard drive. The customers' needs are satisfied with another product than he/she intended to purchase and because of that a memory stick is not needed anymore. A positive complementary decision causes cross-category purchases. The purchase of an new frying pan, can cause the purchase of a spatula for instance.

Data driven product promotion strategies

Retailers promote products to convey a favourable store price image, influence the relative sales volumes of merchandise to improve store profitability, or to increase customer traffic. Promotions are most effective if knowledge about relationships between products in the assortment is exploited during their creation. Since complementary purchase decisions are caused by marketing actions, these are relatively most controllable by store and marketing managers (Manchanda, Ansari & Gupta, 1999). Knowledge about complementary purchase decisions gives retailers tools to increase profit by making changes in the marketing mix (Manchanda, Ansari & Gupta, 1999).

The knowledge that products are substitutes can be the reason that more expensive products are promoted, this is called upselling. Research shows that the sales of a higher-price-tier brand (high quality/price) increases when it is promoted, while the sales of lower-price-tier brands (low quality/price) of the same product decreases. Customers' switch to a higher-price-tier brand in response to a price promotion directly influences net revenue when the brands have different profit margins. This effect is asymmetrical, so lower-price-tier brands do not 'steal' sales from higher-price-tier brands when they are promoted. For this reason it is not advised to give discounts on basic brands (Blattberg & Wisniewski, 1989). A strategy that can be used to increase sales of basic brands is to sell additional brands next to basic brands, without giving discounts in that product line (Svetina & Zupančič, 2005).

A retailer can choose to sell certain products during a certain time period, only together in bundles. In this way sales increases since both/all products in the bundle have to be bought, while the price of the bundle does not have to be lower than the products would have cost if they were sold independently. Explicit price/product bundles are effective if customers perceive the bundled products as highly complementary. If the bundle consists of a certain amount of the same product, is can be based on economics of time and effort in purchasing the bundled items together. For bundles of complements this can be based on an improved image of the bundle because one of the products lends credibility to the other products for instance, or am improved satisfaction of the bundle because of co-occurrence in timing of consumption or usage occasions (Simonin & Ruth, 1995).

Increasing profits with the knowledge that products are complements can also be done with implicit price bundling strategies. The prices of products are with an implicit price bundling strategy, based on the multitude of price and purchase effects that are present across all affected products. In other words, if existing substitutional and complementary purchase decision patterns (with their volume) are known, decisions about promotions, shelf-space management and prices, can be made in the form of product price bundles that are not known by the customer. An example is to promote hair dye by drawing attention to it using a special display, or giving it explicitly a small discount while only the new price is shown. The second step is to increase the prices of hair repair products, hair serum, shampoo for coloured hair and a soft brush a little bit. To finally place those complementary products next to the promoted hair dye on the shelf. This implicit price bundling strategy is used to generate extra profit from products that are regarded as a bundle, while they are not promoted as a bundle (Mulhern & Leone, 1991; Walters, 1991).

The greater the discount of a promoted product (of a higher-price-tier brand), the higher the decrease in sales of substitutes and the higher the increase in sales of complementary products. The customers' perceived costs of buying products in the store they are currently visiting, are lower than the costs of buying the same products in competing stores, because the consumer would incur travel and search costs to buy the same items at a different location. For this reason, advertisement or promotion of certain products should be designed to draw customers into the store and to stimulate sales of other, non-promoted products (Mulhern & Leone, 1991).

However, it seems that pricing promotions do not have to give actual discounts on products to be effective. Shelf space allocations, special displays and simply mentioning that something is an action is sometimes enough. There are enough examples of retailers that sell products for higher prices while they are promoted as discounts (Radar, 2016). Another strategy that is used by some grocery retailers is to give product bundling price offers for certain amounts of the same products, that actually do not contain a discount (Boven, 2016). Strategies like these have to be employed with caution because customer satisfaction and loyalty can decrease when they are noticed.

Another advantage of understanding complementary purchase decisions is that the evaluation of sales, pricing strategies, promotions, or store layout for instance, are more reliable (Walters, 1991). If only the sales effects of promoted products is evaluated, the conclusions can be misleading if the sales of non-promoted substitutes and complementary products is also affected but not taken into account. It is also interesting to find out if the overall sales increases if complementary products are placed next to each other on the shelf because people will more probably buy them if they see them (Walters, 1991), or if they are placed as far away from each other as possible, since that can increase the probability that other products that are seen on the way also are bought. If the retailer has a shop online, an understanding of the relationships between products can be used to make product recommendations to visitors, with the goal of increasing sales of products from the long tail for instance. The effectiveness of those data driven actions can directly be translated into revenue.

Product relationships are often described in the form of association or business rules. The three most common types of business rules, or insights, are the useful, the trivial and the inexplicable (Linoff & Berry, 2011). Useful rules contain information of quality that can suggest a course of action, finding these rules is a primary focus in business applications. Trivial rules are composed of facts that can be derived from common knowledge. They can also be the result of a special event or marketing campaign. These rules can be used for the evaluation of a promotion for instance. Inexplicable rules are difficult to understand and explain and do not suggest a course of action. These rules are often filtered out after obtaining them, because they do not have value for the company.

Methods that can be used to identify existing product relationships in the assortment

The first method to identify purchase patterns was a form of association rule mining (ARM) using POS data (Agrawal, Imielinski & Swami, 1993). ARM methods generate association rules via counting identified frequent item (product) sets that customers tend to purchase together. Association rules indicate that if product set *X* is bought, product set *Y* will probably be purchased also ($X \rightarrow Y$). The Apriori algorithm is the most commonly used ARM method (Hipp, Güntzer & Nakhaeizadeh, 2000). This method only uses POS data. There are many extensions and modifications of this algorithm. An apriori-like algorithm called store-chain association rules, also takes the location of the store and the time in which the rules hold into consideration for instance. This makes the method more suitable for a multistore retail company (Chen, Tang, Shen & Hu, 2005). The biggest weakness of ARM methods is that they generate many inexplicable rules, because number of rules grows exponentially with the number of items (Hipp, Güntzer & Nakhaeizadeh, 2000). Those explicable rules often consist of the exact same product combinations in another order. This means that if the rule $X, Y \rightarrow Z$ exist, the rule $X, Z \rightarrow Y$ and the rule $Y, Z \rightarrow X$ also exist. Another reason that many rules are inexplicable, is that they contain arbitrary product combinations that do not have a relationship with each other. The rule that nail varnish is often bought with a jerry can be found for instance, while these products are not related in real life. The large number of explicable rules makes it impossible to get an overview of the existing relationships in the assortment if the assortment contains many products. This has the consequence that ARM methods are not useful for many retailers, since the assortments are too wide to generate useful information with those methods.

In the machine learning field, clustering methods like K-means are also frequently used to identify relationships between products (Videla-Cavieres & Ríos, 2014). Those methods discover which product sets can be considered as groups, based on the similarity of their dimensions, which are in this case co-occurrences during the same transactions. When clustering methods are used to perform a market basket analysis with, the products that are grouped together have a stronger relationship with each other based on co-occurrences during transactions, compared with other products (Jain, 2010). An advantage of clustering methods is that more variables than only POS data can be used as input. K-means clustering is the most popular and robust clustering method (Jain, 2010). It partitions the data in a predefined *k* number of groups of similar objects. The groups have the same convex structure (Jain, 2010; Shmueli, Patel & Bruce, 2010). The Gausian Mixture Model (GMM) algorithm does the same, but the groups are modelled as Gaussians that do not have to have the same structure (Reynolds). This has the consequence that the clusters represent real-world data often better than the clusters generated with k-means. However, in many real world cases, the number of groups in the data is not known beforehand. It is difficult to define the best *k* value without prior knowledge or another method. For this reason it is also difficult to get the best clustering results, since the results heavily depend on this parameter. If the *k* is too small, products with a strong relationship can be divided into different groups, if the *k* is too big, products can be grouped together while there is no apparent relationship between them. The Dirichlet Process Mixture Model (DPMM) is comparable with the GMM, but it does not use static parameters (Görür & Rasmussen, 2010). It has a Bayesian approach in which the number of *k* is dynamically approximated and changed. This is convenient if there is no prior knowledge about the structure of the dataset. A drawback of every clustering technique when used for a market basket analysis, is that the similarity of objects is measured as distance between numerical features in a multi-dimensional feature space (EMC Education Services, 2015). This means that the transactions have to be converted into binary numbers representing the presence of each product at the exact same index. Thus the whole assortment has to be taken into account for each transaction.

In the field of network analysis, the structure of the connections between different nodes in the dataset is analysed. The goal is to find groups of nodes with denser connections between them than with other nodes. Those groups are called communities. The first community detection methods partitioned the data in separate communities, based on high modularity scores. Modularity is the difference between the internal connection density and the fraction of connections what would be expected based on chance. Those methods have two main limitations. The first is that overlapping communities cannot be identified, so products cannot be member of multiple groups, what is often the case in real world data.

The second weakness is that the modularity measure does not perform well on large datasets, this is called the resolution problem of modularity (Fortunato & Barthelemy, 2007). With the next generation of community detection methods it is possible to identify hierarchical and/or partly overlapping community structures. One of these next generation methods is the Louvain method. The Louvain method identifies communities using the modularity measure, without having the resolution problem of modularity. Communities are found at different levels (the meso, macro, micro level) with a hierarchical community structure as result (Blondel, Guillaume, Lambiotte & Lefebvre, 2008). Other new methods identified hierarchical and/or partly overlapping community structures without the use of the modularity measure. The Clique Perlocation Method (Palla, Derényi, Farkas & Vicsek, 2005), Mixed Membership Stochastic Blockmodel (Airoldi, Blei, Fienberg & Xing, 2009) and Link Clustering (Ahn, Bagrow & Lehmann, 2010) are examples of this kind of methods. Although these are frequently used, taught and cited methods, there is reason to invest a possible limitation of those techniques in the context of this case. Previous mentioned community detection models that can identify overlapping communities, have the implicit assumption that there are less connections between the nodes in the overlapping parts of communities, than in the non-overlapping parts of the communities. While the probability of a connection between two nodes actually gets higher if there are more communities they both belong to (Yang & Leskovec, 2014; Yang & Leskovec, 2015). If a model with the assumption that the overlapping communities are more sparsely connected is applied while this is not the case, the overlapping communities will probably be merged into one community or be regarded as a separate community. The Community-Affiliation Graph Model (AMG) and the Cluster Affiliation Model for Big Networks (BigCLAM) are two methods that have the assumption that the overlapping parts of communities are more densely connected than the non-overlapping parts. If those two methods are combined, non-overlapping, overlapping and hierarchically nested community structures can be identified (Yang & Leskovec, 2014; Yang & Leskovec, 2015).

Yang and Leskovec (2015) compared the AMG and BigCLAM method with other community detection methods on 230 large real-world networks in which the real/ground-truth communities were known. The ground-truth was based on explicitly stated memberships, like online membership of a group or a citation. One dataset that was also used to detect communities was the Amazon co-purchasing dataset (Leskovec, 2013). However, the operationalization of a ground-truth community was the hierarchical product category the product belongs to, instead of products that are often bought together what is the primary focus of this research (Leskovec, Adamic & Huberman, 2007; Yang & Leskovec, 2015).

Community detection methods are normally not used to perform a market basket analysis with. Conventionally, social, technological and biological networks are analysed with those methods (Girvan & Newman, 2002). More recent applications of community detection methods which are related to market basket analysis, is their use with recommender system optimization in which customer-product networks are analysed for instance. Huang, Zeng and Chen (2007) induced a product graph and a consumer graph from a customer product graph, which were used together during the similarity measurement to optimize an recommender system. Another study used community detection in a similar way as association rule mining, in which the products were the nodes and the links their support in terms of bought by the same customer instead of at the same time (Kim, Kim & Chen, 2012). When a market basket analysis is done with network analysis methods, the structure of the products in the assortment is captured in a graph with groups, based on the number of transactions in which they appear together. Only two studies could be found which used community detection methods during a market basket analysis. These are described in the section 'Community detection methods' of this paper. Although community detection methods are not commonly used with finding purchase pattern product relationships, based on their characteristics, the expectation is that they will give more accurate results containing more useful information compared with ARM and clustering methods. With more accurate results a higher similarity with real co-occurrences of products in the dataset is meant. More useful information can be interpreted as less inexplicable information compared with ARM methods for instance and information about more different product relationships.

The objective of this study and the expectations about the results

The objective of this research is to find the best method to identify product relationships based on purchase patterns apparent in point-of-sale (POS) data. The best method is defined as the method that generates results that are most similar with the real co-occurrences of products in the dataset. Having insight in relationships between products makes it for retailers possible to extract business rules regarding purchase behaviour that can be used for customer-demand forecasting and replenishment optimization. The comparison of different methods can provide guidelines for retailers, of how POS data can be used to identify relationships among products. This paper provides information about characteristics of the methods (and their assumptions about the data) that have to be thought about before a certain technique is chosen to a market basket analysis with. Table 1 gives an overview of the methods that are going to be discussed in this paper, together with the type of results they generate.

Method	**Output**
Apriori	Association rules *X* → *Y*. If a customer purchases item set *X* it is likely that he will also buy item set *Y*
Store-chain association rules	Same as a-priori
K-means clustering	A partitioning of the product (categories) into a specified number of clusters, in a way that the objects in the cluster are most similar. Similarity means the least distance from the mean of the clusters.
Gaussian Mixture Model (GMM)	Similar as k-means, but the clusters are modelled as gaussians instead of by the mean

Dirichlet Process Gaussian Mixture Model (DPGMM)	Similar with GMM, but the best *k* value is determent by the method
The Louvain method	A hierarchy of communities at different levels, thus communities can be viewed with a different resolution
Clique perlocation method	A network structure with (non)overlapping communities is identified, in which the nodes of a community share a latent functional property
Mixed-Membership Stochastic Block Model	A network structure with (non)overlapping communities and it is possible to decide the level/resolution of community memberships that are identified with this method.
Link clustering	Hierarchically organized community structures in networks in the form of link communities (instead of node communities)
Community-Affiliation Graph Model (AGM)	An unlabelled undirected network graph that visualizes overlapping, nested and non-overlapping communities
Cluster Affiliation Model for Big Networks (BigCLAM)	Non-negative latent factor estimations that model the membership strengths of the nodes to each community given the graph obtained with AGM

Table 1. The methods that are going to be discussed in this paper with a small description of their output

Based on the properties of the methods and their strengths and weaknesses, there are a few expectations about their performance. Because there are no probabilistic inferences made with the ARM methods, results of those methods are probably most close to the co-occurrences in the dataset. However, because there are often that many inexplicable rules generated what makes it impossible to analyse them with the bare eye, it is not possible to get an overview of the relationships that exist in the assortment./ This method is therefore only used with the selection of a subset of the data instead of during the comparison of the results of different methods. The expectation is that community detection methods will perform better than the clustering methods, because of their ability to find complex structures in networks. On theoretical grounds there can be expected, that the AMG and BigCLAM will give more useful and accurate results than the other community detection methods. This expectation is based on the facts that (non)overlapping and nested communities can be detected and the assumption that the nodes in the overlapping parts are more densely connected than the nodes in the non-overlapping parts. This assumption is statistically more probable and also the case when this was examined with real-world networks (Yang & Leskovec, 2014; Yang & Leskovec, 2015). However, it is also possible that although the statistical probability that overlapping communities are more densely connected is higher, this is not the case in this dataset. If that is the case, another community detection method may give better results. Since the data consists of many products and the relationships are based on a very small subset the same transaction, it is more likely that the Louvain method generates better results than the methods that assume sparsely connected community overlaps. This is expected because of the consequence of the iterative search for the most probable community memberships per node and per community, based on the internal edge densities. This is explained more extensively in the concerning paragraph of the

‘Datamining methods’ section. The DPGMM is expected to be the best performing method of the clustering methods, because of the cluster structure of Gaussians and because the *k* does not have to be defined beforehand. The GMM should generate similar results as the DPGMM if the same *k* value is used. Because this parameter is not known beforehand, this is taken into consideration with the formulation of the expectation that it performs second best. K-means probably performs worst because of its robust nature and the forceful convex structure fitting of the data. Not every expectation could be tested during this research because the resources needed to implement some methods were not available. The link clustering method can be implemented with the python package iGraph and the AMG and BigCLAM method with SNAP. Since those packages are better applicable on a Linux computer, those are not included in the comparison. The results of the k-means, Gaussian Mixture Model, Louvain and Clique perlocation method are compared with the real co-occurrences in the dataset during the experiments. An existing evaluation measure that could be used to compare the similarity of the product relationships generated with the methods with the real co-occurrences could not be found. For this reason two evaluation methods that can be used to compare the results of the different techniques were created.

The next section is devoted to a theoretical review of the datamining methods that can be used to identify product relationships. It starts with an schematic overview of the strengths and the weaknesses of the methods that are going to be discussed. The rest of the section is divided in ARM methods, clustering methods and community detection methods. Each subsection starts with a general description of what those methods have in common, followed with a specific description of each method and it ends with the way the obtained results can be evaluated. The methods are discussed in the same order as they appear in Table 1. Because the largest part of this section contains theoretical information about the structure of the methods and assumptions about the data, the section ends with a paragraph about the applicability of purchase pattern mining methods in real-world cases. The section that follows is dedicated to the experimental setup. First the dataset is described, followed by the experimental procedure. The co-occurrences in the dataset are described, just as the input of the different methods and the two evaluation methods that were created in order to make a comparison of the results with the co-occurrences in the data possible. In the section that follows the results of the methods that are compared are described. The results of the comparison of the product relationships with the real co-occurrences are given in the next section. After that, the conclusions of the research and the final section contains a discussion of the results.

Datamining methods

In this section the datamining methods that can be used identify relationships between products based on co-occurrences in transactions are discussed. Since the methods fall into three different datamining categories, association rule mining, clustering and community detection, like it is mentioned in the introduction, they are discussed in three different subsections. In the first subsection two association rule mining methods are described followed by evaluation measures that can be used for association rule mining. The second subsection is dedicated to the explanation of three clustering methods and a paragraph about the way clustering results can be evaluated. In the third section, six community detection methods are described. Those methods fall into different categories, based on their assumptions about structural properties of communities. The first method identifies communities based on modularity. In the second category, three methods that assume sparsely connected community overlaps, are discussed. The last two methods fall into the category of methods that assume dense connections in community overlaps. The community detection subsection ends with different methods that can be used to evaluate the obtained results. This theoretical section about the different datamining methods, ends with a more practical section in which the applicability of different methods in real-world cases is discussed. Because of the amount of information that is given in this section, it starts with a schematic overview of the strengths and weaknesses of each method. The number next to the name of each method corresponds to the subsection it is described in. The number also reveals into which category method falls.

An schematic overview of the strengths and weaknesses of the methods

Method	Strengths	Weaknesses
1.1.1 Apriori	• Applicable if no prior information about the data is known, useful in the explorative data analysis phase • Simple algorithm, output easy to interpret and explain • The output is a pure representation of the real relationships between products because it is not a probabilistic model • A large number of products/categories is not a problem • Output can be used to obtain a kind of a ground-truth that can be used with the evaluation of other methods • Useful if the relationships of a small subset of the products has to be found, or the most apparent relationships in small time spans around an event for instance	• Does not take differences between stores or time intervals into account. This makes it not useful for a chain of stores. Especially not if the products on the shelves varies between locations. • Suitable support threshold has to be found • The level of aggregation of product categories has to be decided • The amount of possible rules of *k* items, is 2^k-2 with the consequence that it generates many inexplicable rules • Only POS data can be taken into consideration
1.1.2 Store-chain association rules	Same as apriori plus: • Applicable to an entire chain/subset of/individual stores • There are no time restrictions (specific intervals possible) • During the calculations, it takes into account that the products on the shelves can differ across stores, or that locations first opened at different times	• Suitable support threshold has to be found • The level of aggregation of product categories has to be decided • The amount of possible rules of *k* items, is 2^k-2 with the consequence that it generates many inexplicable rules • Only POS data can be taken into consideration
2.1.1 K-means clustering	• Simple, fast and robust algorithm • Easy to add more variables • A large number of observations can be used	• The number of clusters *k* (often not known) has to be defined beforehand. Thus the method has to be repeated with different *k* • The clusters are biased to have a similar span/stretch • Overlapping clusters cannot be identified • Black-box method • Possibility of partitioning of one cluster because of the random placement of the centroids during initialization • It is advised to repeat the method with the same *k* multiple times to select the one with the best results (solution for bullet point 4) • Robustness is not an advantage in this case
2.1.2 Gaussian Mixture Model	• Easy to add more variables • A large number of observations can be used • The clusters can have a different size and shape and can fit real-world data better for this reason • The data is modelled jointly with an additional latent variable z that helps to explain the patterns in the data • Overlapping clusters can be identified	• The number of clusters *k* (often not known) has to be given beforehand. Thus the method has to be repeated with different *k* • Black-box method • The method is much slower than k-means

2.1.3 Dirichlet Process Gaussian Mixture Model	The same as GMM plus: • *K* does not have to be specified beforehand, only a loose upperbound • The method does not have to be repeated to find the best output	• Black-box method • There is no guarantee on the best output because it is not a formal model selection procedure
3.1.1 The Louvain method	• Modularity maximization is a desirable property of communities and the primary focus of this method • The resolution limit problem of modularity is bypassed • It is fast • Communities can be viewed at different levels/with different resolutions	• The role of individual nodes in the communities is not specified, it is not possible to see which nodes are the cause of the division of a community into *n* sub-communities at a different level. (But the weights of the links between each pair of nodes is given with this method, which gives more specific information)
3.2.1 Clique perlocation	• Nodes can be part of multiple communities • Overlapping communities can be identified	• The method has the implicit assumption that the overlapping parts of communities are more sparsely connected than the non-overlapping parts and this is often not the case with real-world data • The density of the connections is disregarded, with the consequence that networks with many connections can be considered as 1 community
3.2.2 Mixed-Membership Stochastic Block Model	Same as clique perlocation plus: • With a concentration parameter you can model different levels of overlap. The communities can be reviewed with different resolutions	• The assumption of sparse community overlaps
3.2.3 Link clustering	Same as clique perlocation plus: • The hierarchical organisation in networks can be captured	• The assumption of sparse community overlaps • It only calculates the similarity of neighbouring links and not of non-neighbour links • Long thin communities are generated with the consequence that a node can have more distance between itself and a node at the opposite side of the same community than to nodes of another community.
3.3.1 Community-Affiliation Graph Model (AGM)	• The assumption that overlapping parts of communities are more densely connected than non-overlapping parts reflects real-world data more accurate • Nested and (non)overlapping communities can be identified. This makes the result more precise • More scalable than other community detection methods (over 10 times more data can be used) • Faster than the other methods	• The output is a graph and it is difficult to extract the model from the graph. The BigCLAM method has to be used to do this • With the used sampling method (Metropolis-Hastings algorithm) the samples are not independent. But under regularity conditions there are a lot of numbers and the central limit theorem allows Monte Carlo approximation.
3.3.2 Cluster Affiliation Model for Big Networks (BigCLAM)	Same as AMG	• The search to the best parameter settings is done with stochastic gradient decent, this speeds up the process but it has the disadvantage that it does not converge at the global maximum but close to it. However, it leads to a hypothesis that is good enough for practical purposes.

Table 2: A schematic overview of the strengths and the weaknesses of the methods that are going to be discussed

1. Association Rule Mining

The goal of ARM is to discover meaningful purchase patterns in POS data, which are given as association rules. Purchase patterns are frequent item sets. In this case, sets of products that are frequently bought together by customers. Association rules have the form of: *X* → *Y*. In natural language this rule would be written as: 'If a customer purchases product set *X*, then he/she is likely to buy product set *Y*'. ARM methods are used to find hidden relationships between product categories, not the cause-roots of these relationships (Leskovec, Rajaraman & Ullman, 2014; Shmueli, Patel & Bruce, 2010). In the next two sections two different ARM methods and their strengths and weaknesses are explained. In the third section, measures that are used to evaluate the output of ARM methods are discussed.

1.1.1 Apriori

The procedure of the Apriori method is sequentially and iterative of nature. The products the retailer sells are called items and therefore, each transaction is a small subset of these items. Only the unique items of each transaction are taken into consideration with this method. This means that if a product appears twice in a basket, it is counted once. A schematic overview of the iterative process can be seen in Figure 1, which is followed with a description of the process.

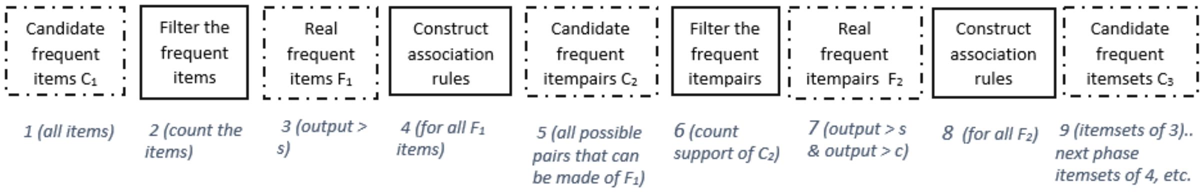

Candidate frequent items C_1
Filter the frequent items
Real frequent items F_1
Construct association rules
Candidate frequent itempairs C_2
Filter the frequent itempairs
Real frequent itempairs F_2
Construct association rules
Candidate frequent itemsets C_3
1 (all items)
2 (count the items)
3 (output > s)
4 (for all F_1 items)
5 (all possible pairs that can be made of F_1)
6 (count support of C_2)
7 (output > s & output > c)
8 (for all F_2)
9 (itemsets of 3).. next phase itemsets of 4, etc.

Figure 1: Schematic representation of the sequential and iterative process of the apriori method. The C represent candidates, the F represents frequent and the small number behind it the phase.

In the first phase the frequent items are filtered. Each item is a candidate frequent item, and if the support of the item is above a predefined threshold *s*, it is frequent F_1. The support is the fraction of transactions containing the item. In other words, the number of transactions containing the item divided by the total number of transactions. After finding the frequent items, association rules are constructed of each possible combination of the frequent items. This are the candidate frequent item pairs (of 2) C_2, that are evaluated in the second phase.

During the second phase the frequent item pairs are filtered in the same way as the frequent items were. Then the confidence of the rule *X* → *Y* is calculated for the frequent item pairs (those that exceeded the support threshold). The confidence represents the conditional probability of the consequent given the antecedent. It is calculated by dividing the support of the item pair by the support of the antecedent (nr. transactions with *X* and *Y* / nr. transactions with *X*). The support threshold excludes item pairs that are not frequent, by doing this it prevents that item pairs with only one transaction have a confidence of 100% for instance. Thus it prevents the generation of rules that only look good. Only association rules that exceed the confidence threshold *c* are accepted. After this candidate frequent itemsets (of 3) C_3 are generated of the frequent item pairs. This process is repeated until no frequent itemsets can be generated anymore. Each frequent item pair and set is given in the output.

There are as many candidate frequent itemsets set taken into consideration as there are items in the set. This is done because of the possibility of an association rule with different confidence values for each item ordering. In other words, the confidence of the rule $X, Y \rightarrow Z$ can be different than that of $X, Z \rightarrow Y$ and $Y,Z \rightarrow X$. However, in some cases it is not needed to check the other direction of the rule, because of the requirement that all items in the rule have to be frequent (Leskovec, Rajaraman & Ullman, 2014; Tan & Kumar, 2005).

Strengths

An advantage of apriori method is that it is a simple algorithm. Another strength is that the output can easily be interpreted and explained. Because it is an undirected datamining technique, it is applicable without prior knowledge of possible purchase patterns or relationships. Making it a useful technique to start an analysis with. Another advantage is that the method can be used to obtain a kind of a ground-truth that can be used for the evaluation of other methods or as input of another method. It is also beneficial that a large number of categories is computational not a problem for the algorithm.

Weaknesses

A disadvantage of the apriori method is that a large number of categories may computational not be a problem, but it is in terms of user-friendliness and usefulness of the results. Another weakness is the difficulty to define the right support threshold. This threshold has a big influence on the results. With a high minimum support a small number of short frequent item sets are generated and a lot of information is lost. If many different products are sold, the few association rules that are generated do not really give an insight in the situation. With a low minimum support there is a big chance that the algorithm gives too many rules to process for a human. Another drawback is that the more different products are sold, the harder it is to find useful patterns. With an item set that contains k items, 2^k-2 candidate association rules can be generated if item sets with an empty antecedent or consequent are ignored. This means that if there are 3 product categories 6 rules can be made and with 10 categories 1022 rules. No matter how many rules are generated, many of them are inexplicable because of a different ordering of the same products and because of the insertion of arbitrary sets of unrelated items in the antecedent (Webb, 2006). Popular products (that are bought often) can also be irrelevant. If $X \rightarrow Y$ has a confidence of ½, and Y has a very high support, it can give a high confidence value independently of X. Another issue to take into account, concerns the degree of product category aggregation. It may be more convenient to have the same level of aggregation for each category, but it is also possible that there will be more useful rules generated if some categories are more aggregated than others. The final weakness of the method is that it does not distinguish between sales at different locations and times. This makes it not very useful for a retailer with multiple shops of different sizes at diverse places, with the possibility of having not always the same products on the shelves at the same time. The rules that are given with this method are the result of the transactions of all stores together. Without repeating the method for each individual store, it is not possible to find common association rules in subsets of stores with the apriori. It also does not take the temporal nature of purchasing patterns into account, like seasonal products, the influence of events on purchase patterns, different selling periods or differences in assortment. This kind of information can only be found if the method is repeated multiple times with transactions made in different time spans. The next method that is discussed does take the location and time into consideration during the generation of association rules.

1.1.2 Store-chain association rules

The store-chain association rules method came as consequence of the weaknesses of the apriori. This method also takes the location (the shop) and the time in which the rule holds account. As consequence of this addition, the generated association rules can be applicable to the entire chain, a subset of stores or individual stores, without time restrictions. This makes it possible to look for association rules of a subset of stores that hold in specific time intervals for instance (Chen, Tang, Shen & Hu, 2005). The procedure is comparable with the apriori, with an additional step each iteration. In the *k*th phase the F_k are generated from C_k item sets, and RF_k are generated from the F_k. This is the same as with the apriori, only the step of generating RF_k (from the F_k) is new, these are relative frequent item sets. Relative frequent item sets take the context of the item set into account. The difference between F_k and RF_k is explained in Cadre 1.

A retailer has 20 stores in the Netherlands. In all stores there are 5000 transactions within a fixed time period. Product X is typically Amsterdams. For this reason it is only sold in stores in and around Amsterdam. At the defined time period, product X is only on the shelves at three stores. The total number of transactions containing X is 3000.

*If the apriori method is used, the support of product X would be: 3000 transactions/(20 stores*5000 transactions)=0.03.*

However, with this way of calculating the frequency of products, products that are not sold in each store will always be missed because they are not frequent. To get valid frequent items, the context of the item (set) has to be taken into account. This is done with calculating relative frequent items RF_k. This means that only the number of shops that actually sell the products during that time period are taken into consideration.

*The relative support of product X is: 3000 transactions/(3 stores*5000 transactions)=0.2.*

The support threshold is set to 0.05 by the retailer. If the apriori method is used, the support value of product X does not exceed the threshold. Consequently, product X is not taken into consideration with the generation of association rules. If the store-chain association rules method is used, the context of product X is taken into account. Leading to a relative support of 0.2, which is higher than the support threshold. Information about purchase patterns of product X that is sold significantly at stores around Amsterdam is not lost with the use of this method.

Cadre 1: An example of the difference between the apriori and store-chain association rules method in calculating frequent and relative frequent items.

In order to be able to make RF_k, information about the context of the item has to be saved. The context of an item (set) is referred to as V_x. This includes which products were on the shelves in which stores at which time and the number of transactions they appeared in. This information is stored in two tables. In a PT (store time) table, with stores as rows and items as columns, the week numbers in which the items changed from on-shelf to off-shelf are saved. In a TS (time transaction) table, with stores as rows and time periods as columns, the number of transactions are saved. Thus as reminder, these tables are used to determine the number of transactions associated with the context for a given item set (*Vx*). The total amount of transactions that is used during the search for relative frequent item sets with this method, is not the total number of transactions of all stores together *D*. It is the total number of transactions of all stores that sold item set *X* during a certain time period DV_X.

In each phase, after finding frequent item sets in the same way as with the apriori method, for each F_k, the RF_k is calculated. Just like in the example given in Cadre 1, the number of transactions associated with the context (*Vx)*, is divided by the subset of the total transactions that are made in the context (DV_X*)*. If this value is higher than the relative support threshold, the RF_k are accepted. The calculation of the relative support of association rules is the same, only with the relative support values (thus the relative support of the combination/ DV_X). The next steps are the same as with the apriori. The only difference is that the confidence of the store-chain association rules are calculated with their context. The context of the rule $X \rightarrow Y$ is the subset of the transactions in the database that contain the stores and times that all items in item sets X and Y were sold concurrently ($V_{X \cup Y}$).

Strengths

The biggest strength of this method, is that it takes the location and the time in which the rules hold into account. The output is more reliable compared with the output of the apriori because it takes into consideration that not every store has to sell the same products at the same time. This is in particular useful for retailers with stores in various countries, or with stores that are changing the product mix dynamically to evaluate the effectiveness of their products in different regions for instance, or for those that have product mixes that change rapidly over time for other reasons. Not every store of a retail chain is opened at the same time. This method prevents that purchase patterns of locations that are not even build yet at a certain time period, are taken into consideration. It is also practical for retailers with multiple stores that rules can be generated for every store of the chain, but also for subsets or individual stores.

Weaknesses

Weaknesses of the store-chain association rules method, are just like the apriori, the decisions that have to be made concerning the support threshold and the level of product category aggregation. Just like with the apriori method, most association rules are inexplicable. Another limitation of both techniques is that it is not possible to take other variables, like marketing actions or the weather, into account. Both algorithms involve much computation because of the iterative process in which the database gets scanned multiple times. But on the other hand, the calculations are very simple, what makes the chance of memory errors minuscule. Another method that produces frequent item sets is FP-Growth. With this technique the step of finding candidate frequent item sets is left out. The frequent item sets that exceed the support threshold are converted into a compressed frequent pattern tree (FP-Tree) structure. The association rules are generated from the F_k that are recursively extracted from the FP-Tree. Theoretically, this method should perform faster. However, in a comparative evaluation of the apriori and FP-Growth, it appeared that this was in particular the case with an artificial dataset and less with real-world datasets. In four experiments the apriori was faster with high support thresholds and FP-growth was faster with low support thresholds. These differences were very small, the authors claim that 'the choice of algorithm only matters with support levels that generate more rules than would be useful in practice' (Zheng, Kohavi & Mason, 2001). For this reason, this method is not explored further in this paper.

1.2 Evaluation association rules

The association rules that are obtained with both ARM methods are often evaluated with two measures. Complementary and substitutional effects of products X and Y (described in the section 'Purchase patterns'), can be measured with the lift. The lift of a rule is the ratio of the support of the rule, to what would be expected if the antecedent and consequent where independent. This is calculated by dividing the support of the frequent item set, by the product of the supports of the components (support X and Y / (support X * support Y)). It measures if there really is a co-occurrence effect, thus the performance of a rule. Purchase decisions are complementary with values above 1, independent with a value of 1 and substitute with values below 1 (Silverstein, Brin & Motwani, 1998; Tan & Kumar, 2005). A weakness of this measure is that two rules that consist of the same components but in another order, can have the same lift value while having different confidence scores. Thus, this metric does filter out irrelevant rules, but not all of them. In the 'Weaknesses' section of the apriori method association rules with popular but irrelevant products in the consequent are described. The measure that is to filter those rules, that only have high confidence values because of the support of the consequent, is the interestingness. The interestingness is the difference between the confidence of the rule and the support of *Y* (thus confidence rule – support *Y*). Rules that are interesting have usually values above 0.5 (Leskovec, Rajaraman & Ullman, 2014).

2. Clustering

Relationships between products are in the machine learning field often identified using clustering methods, in particular with the k-means method (Videla-Cavieres & Ríos, 2014). Some clustering methods partition the products in different groups, and others assign them to each group with a soft probability. Some things are the same with each method. For instance, each data point/object is regarded as a row (vector) with the same amount of columns (features/attributes) in the same order. A data point or object is a transaction in this case. The features are all products that are sold by the retailer. The feature values represent if the product is sold during the transaction or not. The grouping of the data points is based on the distance (that can be calculated in many ways) between the features of the vectors. In other words, the distance between the column (product) combinations of the different transactions is calculated. This is the reason that each vector has to be composed of the same features in the same order. If this is not the case, the method would compare apples to oranges, which leads to unreliable results. The features with the least distance in between them are clustered together in a group (EMC Education Services, 2015; Jain, 2010). In the next three sections, different clustering methods that can be used to perform a MBA with, are described. In their descriptions their strengths and weaknesses are also considered. In the fourth section methods that can be used to evaluate clustering results are discussed.

2.1.1 K-means clustering

The k-means method is one of the simplest, most robust, and most commonly used clustering methods. It partitions observations into a predefined number of *k* clusters. This is done in such a way that the distances between the points of the same cluster to the mean of the cluster are minimized. The process to obtain these clusters is iterative.

Each data point is placed in a multidimensional space (based on their attributes) and *k* centroids are placed at random locations in this space during initialization. The centroids are regarded as the mean of a cluster. Then the points are divided in *k* groups in such a manner that the distance between the centroids and the points in the group is as small as possible. The distance is computed with a similarity measure, the most popular one is Euclidian distance. Thus, the points in a cluster have a high similarity if the squared error between the points and the centroid is as small as possible. In the next step the mean of the points is calculated for each cluster, and that becomes the location of the new centroid. This process repeats itself until convergence (Jain, 2010). A visualisation of the result can be seen in Figure 2. The same data is clustered with different *k*-values. The black circles in the middle can be regarded as the centroids, and the wider coloured circles around them can be seen as the scope of each cluster. The data points are coloured according to the cluster they belong to for illustrative purposes. As it can be seen, each cluster has a convex structure.

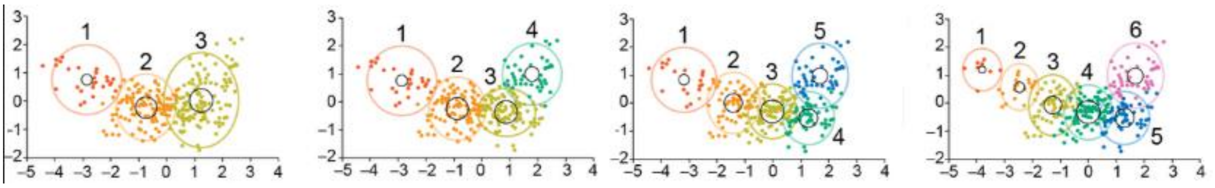


Figure 2: The clustering results of k-means with different *k*-values. The black circles in the middle represent the mean of the cluster, the colours different clusters and all clusters have the same convex structure as it can be seen. Reprinted from ‘Synaptic diversity enables temporal coding of coincident multisensory inputs in single neurons’ by Chabrol, F. P., Arenz, A., Wiechert, M. T., Margrie, T. W. and DiGregorio, D. A., 2015.

Strengths

A strength of this method is that it is simple, what makes it useful for analysing high dimensional data. High dimensional data is complex, which requires a simple model to prevent overfitting. With low dimensional data a more complex model can be used. It is very useful for exploring the overall structure of the data. Another strength is that it is robust, what means that it is applicable for a variety of data. It is also one of the fastest clustering methods (EMC Education Services, 2015; Jain, 2010). The idea behind this method is comparable with ARM, but instead of counting occurrences, the distance between feature vectors is measured. An advantage of this method over ARM methods is that it is possible to include additive features in the analysis.

Weaknesses

A limitation of this method compared with ARM is that it is harder to explain the results because it is kind of a black-box method. Because of the black-box nature of the method it is hard to check if the clustering is done right. Another disadvantage is the data gets forcefully partitioned in different non-overlapping groups with the same convex structure, while real world objects are often member of multiple groups that have different structures. When a market basket analysis is performed using k-means (which is often used for this purpose), only a few product relationships are captured, and a lot of information about relationships with products of other clusters is lost. The random placement of the centroids during the initialization is sometimes not beneficial. If two centroids are placed within or at the edges of one cluster, this cluster will be forcefully partitioned into two clusters with the consequence that the right clusters can never be found. Another consequence of the random initialization is that the results can change if the method is repeated, this is also the case after adding more observations to the dataset. As it can be implied from Figure 2, the results heavily depend on the chosen *k*-value. Even though this value is often not known in real-world data, this parameter has to be specified beforehand. A solution for this weakness is running the algorithm multiple times with different *k*-values, and choose the one that gives the most meaningful clusters and the lowest squared error. Even though all those weaknesses, if the right number of *k* is selected, it finds the right clusters most of the time. Another weakness of the method is that it does not handle categorical data well. Categorical features have to be converted into numerical values. This has to be done with caution. If different products are labelled with different numerical values for instance, the result would not make sense, since shampoo with a value of 1 is not less than the toothpaste that got a value of 2, in the real world. Another solution is to convert the transactions into binary dummy variables. Each transaction has to have the exact same structure. This means that each product in the assortment gets an own column, with a numerical value representing if it was present in the transaction or not. However, when many features are labelled binary, distance calculations can become difficult (EMC Education Services, 2015; Jain, 2010).

2.1.2 Gaussian mixture model

This model is similar to k-means but more sophisticated. The clusters are not modelled by the mean but as Gaussians. Each cluster is described by a Gaussian distribution, thus it gets a mean, a variance and pi value (size). The assumption that the data has a Gaussian distribution is widely accepted in statistics. The Gaussian distribution of the two clusters at the bottom of Figure 3, is visualized at the top of Figure 3. The locations of the two lines that are drawn on top of the Gaussians, with 'GAP' between it, are the means of the clusters. The place where the lines of the Gaussians touch each other in the 'GAP' area, would be the border of the two clusters generated with the k-means method. Since all points are assigned to each cluster with a certain probability, it is possible to extract overlapping clusters with this method. This means that relationships between products from multiple clusters can be identified in the data. The information that would be lost when k-means is used (highlighted in red) can be generated with GMM. Those characteristics make the result less polarized in comparison with k-means. The method performs iteratively in two steps.

The parameters of the Gaussian distribution are kept fixed during the first step. Each data point gets assigned to each cluster with some soft/relative probability *r,* calculated with maximum likelihood estimation. This is the probability that it belongs to a particular cluster, divided by the sum of the probabilities that it belongs to the other clusters. If the value is small the data point does not belong to a cluster and if it is high (almost 1) it belongs to a cluster. During the second step the probabilities *r* are kept fixed and the parameters of the clusters are updated. The parameters (mean, variance and pi) are weighted by *r* of each data point to the cluster. This has the consequence that a data point does not influences the average or the other parameters of the clusters much if it does not belong to the cluster, because of the small *r* value. These two steps are repeated until convergence. The output is a collection of clusters and soft membership probabilities of each feature to the clusters. If the features have to be assigned to a single cluster, the cluster with the highest probability has to be chosen (Reynolds).

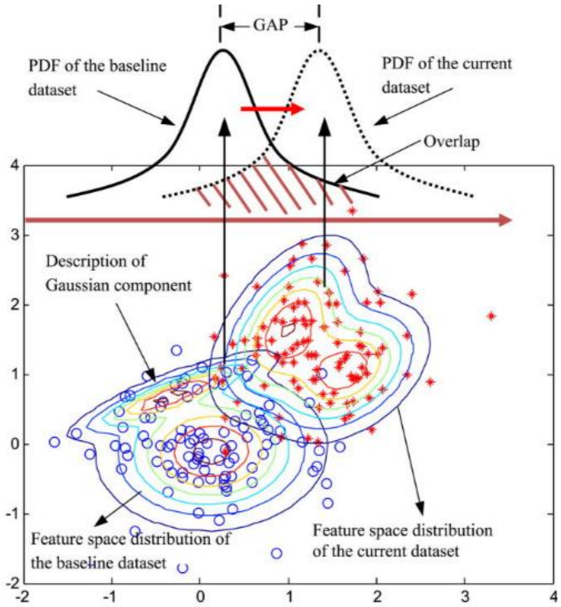


Figure 3: The Gaussian distribution of data from two documents at the top of the figure, and the clustering result of the data obtained with the GMM method at the bottom. Reprinted from Yu (2012).

Strengths and weaknesses

Strengths of this method are just like k-means, the possibility to add other kind of variables and it can be used with high-dimensional data. Extra advantages over k-means are the parameters pi and variance, that circumvent similar size and shape requirements of the clusters. This leads to more precise and realistic clusters. Another strength of the method is that soft probabilities of cluster membership are given instead of hard cluster assignments. The consequence is that less information is lost compared with hard cluster assignments. An important weakness is that *k* has to be specified beforehand. Another weakness is that it is much slower than k-means (Reynolds).

2.1.3 Dirichlet process gaussian mixture model

The DPGMM is similar with GMM, but a difference is that the number of clusters *k* does not has to be specified beforehand. The most probable *k* is approximated using the Dirichlet Process. The parameters that have to be given is a maximum *k* and a concentration parameter (alpha). This is very convenient when a market basket analysis is done since *k* is not known beforehand. In Figure 4 the clustering result obtained with the GMM method with a *k* of 4 is shown at the left and the result obtained with DPGMM is shown at the right. The result of the GMM does not make sense in Figure 4. As can be imagined, it is difficult to find the right *k*-value when hundreds of products have to be clustered. The result of the DPGMM does make sense. As two *k*-values fit the data, the method always chooses the lowest one, preventing overfitting, as is the case with the left clustering result.

The maximum *k* really is an upper limit that will not be crossed. The results will not change much depending on the value that is given because the method will always search for the best fitting lowest *k* value. The concentration parameter (alpha), has more influence on the result. It specifies resolution with which the values are reviewed. The values of the data points in a multi-dimensional space are continuous and the assumption is that the data has a particular distribution. The Dirichlet process draws distributions around data points of an expected base distribution, with the assumption that data points that were observed before are more likely to happen. The Dirichlet process distribution is discrete. If the base distribution has continuous values, like the case with clustering in a multi-dimensional space, the values are discretized in the expected base distribution and the alpha specifies to which degree. The lower the alpha the more discretisation. If the alpha is higher the values are more continuous like the base distribution. The more the data points are discretized, the more likely it becomes to observe the same values as seen before. This means that the concentration of the distribution is lower and therefore less specific, what results in less clusters. If the alpha is high the values in the expected base distribution are more specific, less likely to be observed multiple times and the result will consist of more different clusters. A low alpha can be compared with a high *k* value used with *k*-nearest neighbours clustering and a high alpha with a low *k* value. Thus the resolution with which the distribution of the points and therefore the expected distribution of the points in terms of clusters (what ultimately means the expected number of clusters), depends on the alpha parameter. If the expected distribution is defined (based on local convergence), the clustering is done in the same way as with the GMM clustering method that is described in the previous paragraph (Görür & Rasmussen, 2010; Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel & Vanderplas, 2011; Teh, 2011).

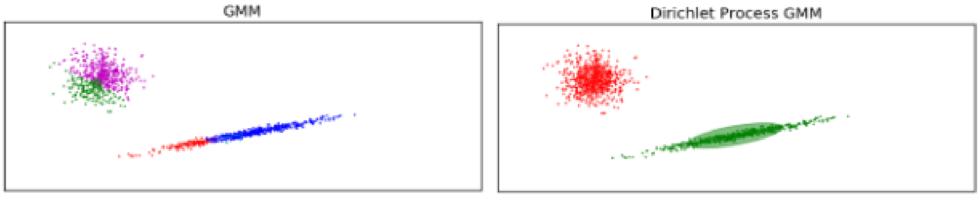


Figure 4: The same data that is clustered with the GMM method with a predefined *k* of 4 at the left and with the DPGMM, with the best *k* automatically and dynamically being approximated at the right of the figure. The *k* value that is used with the GMM is too high and causes overfitting. Reprinted from Pedregosa et al. (2011).

Strengths and weaknesses

The most important strength is that the *k*-value does not has to be defined beforehand. This also means that the clustering does not need to be repeated multiple times to find the best results. The other strengths of the method are the same as the GMM method. A weakness of DPGMM is that it is a Bayesian nonparametric model. These models are at this moment not formally or empirical proven because of the lack of frequent empirical convergence and consistency guarantees

2.2 Evaluation clustering

It is impossible to say that one clustering method is better than the other. The quality of the results depend on the goal of the analysis, preferences of the characteristics of the methods and the structure of your data. If the goal is to have a general overview of product relationships in terms of different groups k-means can be the best to use. If more specific insight in the relationships are desired, more advanced techniques like GMM can be more suitable. When it is needed to have information about all possible product relationships a method that assigns each product to each cluster with a soft probability is better than a method with hard cluster assignments as result. It is also possible that the speed of the method is important, then k-means will perform better. Despite those individual preferences, there are different ways to evaluate clustering methods.

If the data contains labels, the clustering can be done on a test set without label and the (dis)similarity of the cluster assignment with the true labels can be measured. The percentage of correct assignments can be calculated with the rand measure for instance (Rand, 1971). If false positive or false negative cluster assignments are less desirable, errors of those kinds can get another weight when the F-measure is used. Both measures can be used on the whole clustering result or on each cluster independently. The F-measure is not applicable on the clustering of products based on co-occurrences in transactions, because this is not labelled data. The rand measure can also be used to compare different clustering results with each other or to compare the results of the same method on different data samples from the same source, to access the cluster stability/self-consistency. Thus if the products are clustered multiple times, for example with subsets of the POS data with different time periods, it is possible to use this measure. Another evaluation possibility is to calculate the root mean squared error (RMSE) or mean absolute error (MAE), evaluation measures often used with forecasting. The sum of the squared (RMSE) or absolute (MAE) differences with the real co-occurrences of products, divided by the number of observations gives the average error percentage of the clustering (Chai & Draxler, 2014).

Clusters are defined as points that are grouped together with their nearest neighbours (connectedness) with a high intra-cluster similarity (homogeneity) and a high inter-cluster distance (separation). These structure characteristics play an important role during the generation of clusters. They can also be used to evaluate the quality of the cluster. There are many measures that evaluate those cluster characteristics.

The sum of the average distances of all points in each cluster to their centroids divided by the distances between the centroids, times 1/k is the Davies-Bouldin index (Davies & Bouldin, 1979). A low score with this measure means that is a high intra- and low inter-cluster similarity. The Dunn index is the ratio between the minimal inter-cluster distance to the maximal intra-cluster distance. It measures the same cluster properties as the D-B index. A high value means high intra- and low inter-cluster similarity. The silhouette coefficient defines similarity of a data point to each cluster, as the difference between the lowest average similarity of the point to its own cluster and the most similar neighbouring cluster, divided by the maximum average distance of the point to its own and to the most dissimilar cluster. The Krzanowski-lai cluster index is a cluster quality measure, with which the difference between the squared differences of the points to their cluster centroid with *k* clusters and *k*-1 clusters is divided by the same difference between *k* and *k*+1 clusters. Based on the structural properties, the highest value is the best one (Krzanowski & Lai, 1988). A minimum within cluster variance is a structural property of a cluster that is used to generate clusters from the data during the computations. This means that it can also can be used to access how well the clusters are made (Chen, Jaradat, SBanerjee, Tanaka, Ko & Zhang, 2002). Those structural measures are often used when the data is not labelled. POS data is also not labelled, but because of the structure of the data, those measures are not suited to evaluate the clusters. Normally the feature values are continuous, which makes the calculation of distances to the mean natural. The feature values of POS data are made continuous to make it possible to use those methods, but they still have a binary nature. The product is present in the transaction or it is not. This leads to polarized values that does not say much if those measures are used. Because of the black-box nature of clustering methods, it is important to evaluate the clustering result by interpretation. This means reviewing the results to see if the result looks right, plausible, similar to real world cases, or if it makes sense.

3. Community detection

The goal of community detection is to understand the structure of the network that can be represented in a graph. Communities are cohesive groups of nodes with many connections between them. In other words, it are dense subgraphs in the network. In this case the nodes are the products. The nodes that are member of the same community are connected by a shared latent/unobserved functional property. The main difference between clustering and community detection methods, is that the structure of the dataset is based on closeness of points with clustering, and on edge density with community detection (Gratzer, 2013). Although these methods are generally not used to find relationships between products in a transaction network, given their characteristics, they should be well suited to use them. Today's width in product assortments of retailers has led that the traditional methods are not useful to identify relationships between products. Community detection methods have the ability to identify the structure of complex networks. This led to the expectation that those methods are better in detecting the relationships between the products in the assortment.

Only two studies with the same goal of this paper could be found. With one study support thresholds are evaluated using modularity scores, which indicate the quality of communities that various support values. The edges between products are the confidence scores of product pairs. An utility measure of the communities was created by dividing the average edge strength of nodes in the community, by the number of edges divided by the number of edges +1. However, this measure could only be used for each independent community and not for the comparison between different communities (Raeder & Chawla, 2011). The other study also identified communities based on modularity, and proposed different link thresholds that can be used to filter the connections between nodes. Node pairs with less connections between them than the threshold do not get an edge with this method, resulting in less relationships of more significance (Videla-Cavieres & Ríos, 2014).

Community detection methods can be based on the structural property of modularity, sparsely connected overlapping parts of communities and densely connected overlapping parts of communities. In the next section the concept of modularity is explained, followed by the description of the Louvain method. A community detection method based on modularity maximization. The three methods that are explained after the Louvain method, have the assumption that overlapping parts of communities are sparser connected than the non-overlapping parts. A general explanation of overlapping communities is given. Followed by those three methods, the clique perlocation method, mixed-membership stochastic block model and link clustering. The strengths of these methods are discussed during the description, eventually together with a weakness. The biggest weakness of those methods is the assumption of sparse connections in the overlapping parts. This weakness is explained in the section 'The weakness of sparse community overlaps'. This section is followed by the concept of dense community overlaps. Then two methods, the Community-Affiliation Graph Model (AGM) and the Cluster Affiliation Model for Big

Networks (BigCLAM), that have this assumption are discussed. Those methods can be used combined, to find communities. The last section concerns the evaluation of community detection methods.

3.1 Modularity

One of the first community detection methods was based on the measure modularity. A community was defined as a subgroup in the network with denser internal connections and sparser external connections. Modularity is a measure that represents the degree of density of the connections of the nodes within and between communities. The highest value is 1 and lowest value is -1. If the network has a high modularity value, there are many connections between the nodes in the same community and a small number of connections between nodes of different communities. If the modularity has a low value it is the other way around. Modularity is calculated by subtracting the fraction of edges that would be expected if they were distributed in the network randomly, of the fraction of edges that fall within the communities. Thus if the value is positive, there are more connections within communities, than the fraction that would be based on chance. The amount of nodes and the amount of edges per node is taken into consideration with this chance calculation (Fortunato, 2010).

Communities are identified by calculating the modularity for different candidate communities. The divisions with the highest values are accepted as communities. This process is called modularity optimization. Just like the 'p-value problem' in statistics, with which small effects become significant if the sample size is big enough, the modularity measure has the same problem. If there are many nodes in the network, the chance on an edge between different groups decreases. This is due to the fact that the amount of edges per node is taken into account and the network is regarded as random. This means that the connections between nodes will be above the random chance level (which is very low). What has the consequence that they are regarded as a community, even though they are not well connected. It also means that different communities that have a dense interconnection, will be merged with other communities because of the unrealistic edge fraction value based on random chance calculation. This is called the resolution limit problem of modularity because it is not possible to detect well interconnected communities that are smaller than a certain scale, in other words: there is a limit on the resolution of the communities (Fortunato & Barthelemy, 2007).

Another weakness is that the result of methods that are based on modularity (and other traditional/older community detection methods) are non-overlapping communities (Leskovec, Lang, & Mahoney, 2010). An example of two non-overlapping communities can be seen at the top of Figure 5. An adjacency matrix is made with a row and column for each node and the connections between the nodes are visualized as dots. As it can be seen at the bottom of Figure 5, it is clear that the distribution of the edges is strictly separated between the different communities and that there are many connections between the nodes of the same community. Like it is mentioned in the previous paragraphs, the right method for analysis depends both on the goal and on the data. The traditional community detection methods can be useful if the data has a structure of separated groups. They are also useful if this is not the case, but the goal is to have a more general structure of the data with separate groups. If the data consists of groups that are also overlapping, the methods work fine but the generated groups will not reflect the real structure as it is apparent in the dataset. Because overlapping communities are significantly presented in real-world data (Palla, Derényi, Farkas & Vicsek, 2005), new community detection methods are created to identify those complex structures.

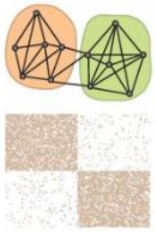


Figure 5: A representation of two non-overlapping communities at the top of the figure. The adjacency matrix underneath it visualizes the edges (dots) of the nodes (rows/columns). Reprinted from 'Overlapping communities explain core–periphery organization of networks' by Yang, J., and Leskovec, J., 2014.

3.1.1 The Louvain method

With the Louvain method, communities are identified based on the measure modularity, but the resolution limit problem of modularity is passed by (Blondel, Guillaume, Lambiotte & Lefebvre, 2008). Most algorithms do not calculate the modularity for each possible candidate community because of the large computations that would entail. With the Louvain method communities are identified in two steps that are repeated iteratively until the modularity does not increase anymore. The algorithm identifies communities at different levels (macro, meso, micro level). The macro level represents the network of communities with the lowest resolution, and the micro level with the highest resolution. The result is a hierarchical community structure. An visualisation of a hierarchical community network structure can be seen in Figure 6. The dots are communities that are found at the lowest (macro) level and the lines in between two communities represent a connection/edge. The meaning of the colours of the dots is not relevant for the purpose of this example. The community in the middle is visualised with a higher resolution. Those dots are the identified communities at the meso level. If one of those communities is visualized at the micro level, the individual nodes (and the connections between them) can be seen.

During the first step, the network of *k* nodes is partitioned in *k* communities. For each node, the neighbouring nodes are considered as candidate communities. The increase in the modularity value of the network is calculated for the hypothesized situation, in which the node is removed from the community to each neighbouring community. The node gets assigned to the community that caused the highest modularity increase for the network. If this was with the same community the node stays in the same community. This is done iteratively with all nodes (and often multiple times) until the modularity does not increase anymore. If moving individual nodes does not increase the value anymore, the local maxima of modularity is found.

Throughout the second step, the communities obtained with the first step, are regarded as nodes in a new network. In this meta-network, which is actually an aggregation of the first network, the internal edges of the communities are regarded as self-loops on the community node. The external edges are regarded as a weighted edge to the other community nodes. After this new network is created, the first step gets repeated. With each iteration of those two steps the number of meta-communities decreases. The result is a hierarchical community structure. Thus it is possible to look at the communities at different levels, or with different resolutions (Blondel, Guillaume, Lambiotte & Lefebvre, 2008).

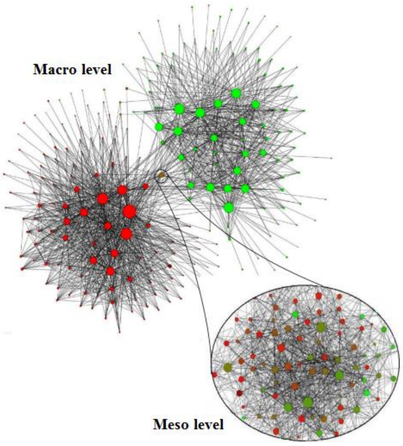


Figure 6: A graphical representation of a network of communities at two different levels. The dots represent communities. A zoom at a higher resolution of one of the communities at the macro level, shows that this community is composed of multiple sub-communities at the meso level. If one of those communities is visualized with a higher resolution, the structure of the network can be seen at the node level. Reprinted from Blondel et al (2008).

Strengths

A strength of this method is that modularity maximization, a desirable characteristic of communities is the primary focus with this method. The hierarchical structure of the method, the same number of communities as nodes during the initialization and the movement of individual nodes during the first step bypass the resolution limit problem. Merging communities that are not internally densely connected does not happen if nodes are moved individually. Another strength of this method is that it is much faster than other methods that are based on modularity (Blondel, Guillaume, Lambiotte & Lefebvre, 2008).

Weaknesses

Although the result of the Louvain method is a hierarchy of communities, it does not specify which individual nodes are in an overlapping part of a community and which are not. It is possible to look at the communities at different levels and to see that a community in one level consist of three smaller communities for instance, but a further specification like the nodes that are responsible for the connections between those three communities that cause the merge on a higher level is not given. On the other hand, the weights of the links between each pair of nodes is given with this method. Thus it is possible to review more specified information of the connections between the nodes.

3.2 Sparsely connected overlapping communities

Palla et al. (2005) where the first who showed that real-world data often consists of overlapping communities. They showed the concept of overlapping communities with the visualisation at the left of Figure 7. The black dot in the middle represents 'the authors' of their paper with several communities they are member of around it. The 'Scientists' community contains four nested communities, that are also partly overlapping. Those communities cannot be identified with divisive and agglomerative methods, the traditional community detection methods that divide the nodes of the dataset in separate groups. An illustration of the output those traditional methods would generate from the same dataset, can be seen at the right of Figure 7. 'The authors' (black dot) are grouped together with parts of the communities they are member of. Information about the structure or relationships of the members of the community with those of other communities are not given in this result.

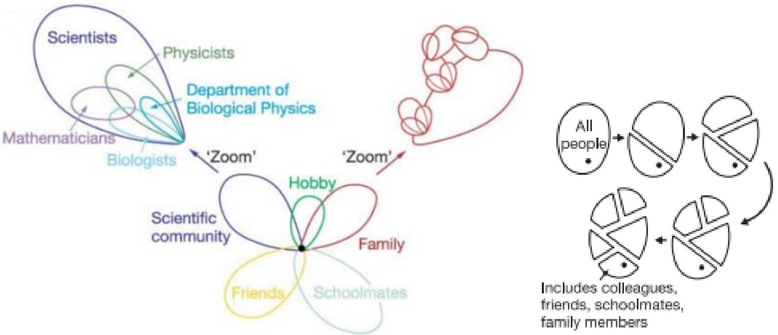


Figure 7: A visualisation of the concept of overlapping communities that can be found with the Clique Perlocation Method at the left. The community structure of the same network that would be generated with the traditional community detection methods at the right. Reprinted from 'Uncovering the overlapping community structure of complex networks in nature and society' by Palla et al., 2005

3.2.1 Clique Perlocation Method (CPM)

With this method (non)overlapping and nested communities are identified by analysing the connections between nodes (Palla et al., 2005). If *k* nodes in the network are fully connected (all nodes are linked with each other) they form a *k*-clique. An adjacent clique is formed if *k*-1 nodes are connected. In Figure 8, three 3-cliques are shown: *a, b* and *c*. Clique *b* and *c* are adjacent, because they share 3-1 nodes.

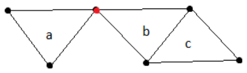


Figure 8: An example of a network consisting of 6 nodes and 8 edges, forming three 3-cliques (*a, b, c*) and two communities (*a* and *bc*) and one (red) node that is member of two communities.

The union of the *k*-cliques that are connected through adjacent k-cliques, form a community together. Other *k*-cliques that share *k*-2 nodes for instance, are not adjacent and therefore, not part of the community. The network in Figure 8 consists of community *a* and community *bc*. The nodes that are part of multiple communities, reside in the overlapping parts of different communities. This is the case with the red node, it is part of community *a* and part of community *bc*. The method identifies maximal cliques and converts them to *k*-clique communities.

Strengths and weaknesses

A strength of this method is that it is acknowledges the possibility that nodes can be part of multiple communities. This was the first community detection method that identified overlapping groups. On the grounds that communities in real-world data are often overlapping, this method gives more accurate results than the earlier created methods. A weakness is that it does not identify dense subgraphs (the definition of communities). It identifies cliques with neighbouring adjacent cliques, the density of the connections is disregarded. This has the consequence that networks with many connections can be regarded as one community, because single connections are enough to form edges (Grätzer, 2013).

3.2.2 Mixed-Membership Stochastic Block Model

The presence or absence of connections between nodes is analysed with this method (pairwise measurements). For every node in the network there are some kind of hidden community structures. There are *k* communities and the network is represented in a graph G = *(n, r). N* is the number of nodes in the network and *r* are the fractional edge weights between a pair of nodes, which are conditionally independent given their community memberships. This means that edge formation is explained by the memberships of the nodes to the communities they are member of. The edge *a*→*b* can have another value than *b*→*a*. Thus there is node specific variability in the connections (mixed memberships). The directed network graph that is generated is based on the latent distributions of community memberships of each node (the fractional node to community memberships) and the interaction strength between the different communities. The interaction strength between the different communities are the probabilities of connections between communities defined by a *k* x *k* matrix of Bernoulli rates. For each node in the network a *k* dimensional mixed membership vector is drawn given the Dirichlet process. The randomly drawn vector *πi* specifies the latent per-node distribution of node *i* and the probability that node *i* belongs to a group *g πig*, which is different for different groups because of the fractional memberships a node has. The distribution of the latent variables are inferred according to some probability equitation containing a Dirichlet, multinomial and variational parameters that is solved using coordinate ascent. Because of the fractional memberships of a node, they are normalized to one (1 = 100%). Thus in other words the nodes have independent edge probabilities for each community, that are averaged. The fractional edge weights between two nodes is the weighted average over all community membership edge probabilities of those two nodes. Because of the conditional independence of fractional edge weights (directed network graph), the network graph consists of *n* fractional edge weights (Airoldi, Blei, Fienberg, & Xing, 2009)..

Strengths

An advantage of this method, besides the possibility to find overlapping communities, is that it is possible to model different levels of overlap is by changing the concentration parameter.

3.2.3 Link clustering

Link clustering is used to identify hierarchically organized community structures in networks (mi Bagrow & Lehmann, 2010). These have the form of link communities (b in Figure 9), instead of node communities (a in Figure 9). With the Jaccard distance measure, the similarity between neighboring links (of connected edges) is calculated. The similarity between links e_{ik} and e_{jk} at the right of Figure 9 for instance, is the intersection of the nodes *i* and *j* divided by the union of the nodes *i* and *j*. The intersection of the nodes is the number of nodes with the blue square around them. The union of the nodes are all distinct nodes that are connected. This makes the similarity between e_{ik} and e_{jk} 4/12. The similarity scores are put in a transform matrix and single-linkage hierarchical clustering is applied on it. Thus each link is a community at initialization, and the pair of links with the highest similarity are merged, until all links belong to a single cluster. These steps are visualized in a dendogram (c in Figure 9) that contains the information about the hierarchy of the network. The similarity score at which clusters merge represents the strength of the community. The darker the dot in the matrix the higher the similarity score. Because all edges end up in the same community at the end, a threshold is used to decide the final community partitioning. This threshold is the minimum average linkage density, the weighted average of the density of the communities. With a low threshold a few large communities are generated with many edges that are clustered together. With a high threshold many small communities are identified with only highly similar edges in the same community.

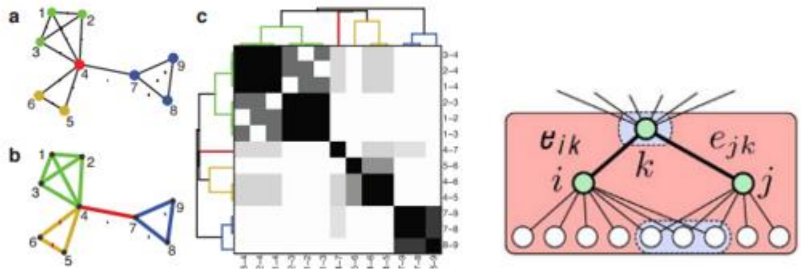


Figure 9: The node network (a) is seen as a link network (b), of which the similarity of neighboring links are calculated and put in a matrix that gets clustered hierarchically (c). At the right of the figure the similarity calculation between links e_{ik} and e_{jk} is visualized, the intersection (blue) is divided by the union (all distinct ones). Reprinted from ‘Link communities reveal multiscale complexity in networks’ by Ahn et al., 2010.

Strengths and weaknesses

The strength of this method is that this it can identify overlapping communities and the hierarchical structure of the nodes. The hierarchy of nodes in networks with communities where the nodes belong to multiple groups cannot be captured with the traditional methods because of the overlapping groups. This method analyses clusters as a set of links instead of nodes, which makes it possible to identify overlap and hierarchical organisation in networks. A weakness of the method is that it only calculates the similarity of neighbouring links and not of non-neighbour links. Due to the hierarchical clustering long thin communities are identified. It is possible that a node can have more distance between itself and another node at the opposite side of the community than to nodes of another community.

3.3 Densely connected overlapping communities

The three previously described methods, all have the implicit assumption that the probability that nodes in an overlapping community are connected is lower, than the probability that nodes in the separate communities are connected (Yang & Leskovec, 2012; Yang & Leskovec, 2013). An example of this kind of community structure can be seen at the top left of Figure 10. The adjacency matrix underneath it visualizes the edges (dots) of the nodes (rows/columns). As it can be seen in this visualization, the density of the edges is higher in the non-overlapping parts of the communities (the darker squares in the corners) than in the overlapping part (the lighter square in the middle).

However, if the dataset is very big, there can be made a lot of different clusters or communities that will be partly overlapping. This results in a nested core-periphery network in which the core gets denser and denser (because of the overlapping clusters) and contains 60% of the nodes and 80% of the edges (Yang & Leskovec, 2014). An example of a nested core-periphery network can be seen at the middle of Figure 10.

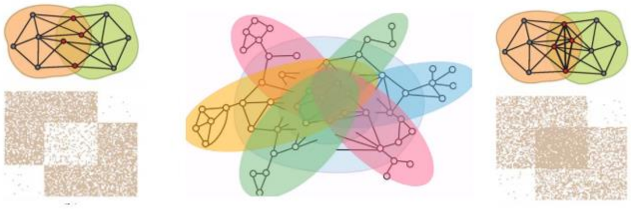


Figure 10: A sparsely connected overlapping community structure at the left. In the middle a nested core-periphery network, with 60% of the nodes and 80% of the edges within the light blue circle at the middle. A community structure with densely connected overlapping communities at the right of the figure. Reprinted from 'Overlapping communities explain core–periphery organization of networks' by Yang, J., and Leskovec, J., 2014; 'Affiliation Network Models for Densely Overlapping Communities in Networks**'** by J. Leskovec, 2012.

The edge probability between two nodes that share *k* communities, is the probability that they are connected, given the number of communities they share. If this edge probability is calculated the value gets higher if *k* increases. This means that if nodes share multiple communities, the probability that they are connected gets higher. Thus in the overlapping parts of the communities the connections are more dense instead of sparse. A visualisation of this concept is shown at the right of Figure 10. At the top it can be seen that there are more links between the nodes in the overlapping parts of the communities. At the adjacency matrix underneath it the density of the edges is higher at the middle square that represents the overlapping part, compared with the non-overlapping parts at the corners.

Previously discussed community detection methods will probably generate results of such a network that are too general. The overlapping communities will be seen as one instead of independent clusters that overlap (or are nested). An example of such a (too general) community is visualized as the light blue circle in the middle of the nested core-periphery network in the middle of Figure 10 (Leskovec, 2012; Yang & Leskovec, 2014). Another possibility is that the dense overlapping community is identified as a separate community, instead of different communities that have an overlap.

A method that can be used to identify communities, that has the assumption of dense community overlaps, can be deployed in two steps. Like it is mentioned in the introduction, the first step is to make an unlabelled undirected social network graph with the Community-Affiliation Graph Model (AGM) in order to extract the communities from the network. The second step is to find the underlying model with the Cluster Affiliation Model for Big Networks (BigCLAM) based on the network graph made during the first step and the number of communities and parameters. In the next two sections, those two methods are described in the same way as the previous method descriptions.

3.3.1 Community-Affiliation Graph Model (AGM)

The goal of this method is to identify overlapping nested and non-overlapping communities. First a generative model for networks (bipartite affiliation network) is defined. This bipartite affiliation network has parameters representing how communities, given a set of nodes, generate edges of the network. The result is an unlabelled undirected social network graph $G(V, E)$, of the set of nodes with their edges. This result is obtained in three steps (Leskovec, Rajaraman & Ullman, 2014; Yang & Leskovec, 2012a; 2012b).

First a bipartite affiliation network $B(V, C, M)$ is made with the set of nodes V of the original network G and a set of nodes representing a predefined number of communities C. In order to define the memberships M of the nodes to the communities C, the edge probability $(u,c) \in M$ from node $u \in V$ to community $c \in C$ needs to be computed. This is done with maximum likelihood estimation. Thus the bipartite affiliation network $B(V, C, M)$ is a generative model, that contains information about which nodes are member of which communities. An theoretical visualization of the result of this step can be seen at the left side of Figure 11. It shows which nodes are probably member of which community.

In order to draw a network graph from the model, the edges between pairs of nodes of the same community are generated with a probability parameter p_c. These edges between pairs of nodes of the same community are generated for each community independently. Pairs of nodes cannot have multiple edges between them in the graph, but the more communities two nodes share the higher the probability of an edge of course.

The parameter $\{p_c\}$ is the set of probabilities of nodes in the same community being connected for all communities. In the third step this parameter is (just like the p_c) calculated with the maximum likelihood of the observed edges in G, given the bipartite community affiliation network $B(V, C, M)$. The $\{p_c\}$ parameter has to have the maximum likelihood for all the independent edges p_c and for the set of all edges $\{p_c\}$. For easier computation the logarithm of the likelihood is calculated instead of the maximum likelihood function itself. The globally optimal values are found with gradient descent. A few random nodes are picked with the Metropolis-Hastings algorithm and a stochastic search with two steps is done. During the first step $\{p_c\}$ is kept fixed to find the optimal (maximum likelihood) values for *B*. During the second step *B* is kept fixed to optimize the values $\{p_c\}$. After a few iterations Monte Carlo approximation is used to accept (average) a maximum likelihood value, at the moment that the values do not change much anymore. From this model a network is drawn, as it is visualized at the right side of Figure 11.

Because the number of communities has to be given beforehand, a method that first fits a candidate community affiliation graph B_0 *(V, C_0, M_0)* with a large number of communities is made. Then a l_1 penalty term is set on parameters $\{p_c\}$ and the optimization problem with a regularization intensity parameter gets solved. The l_1 penalty forces the p_c to 0. This has the consequence that the communities with the p_c value 0 are ignored.

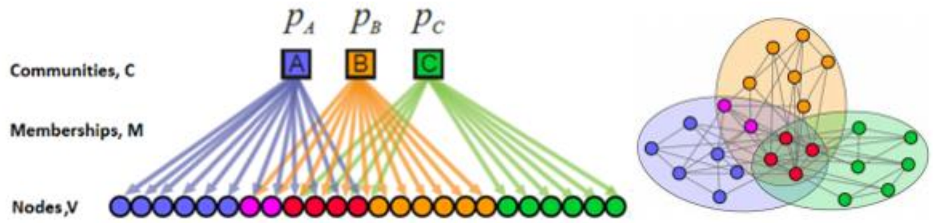


Figure 11: At the left a bipartite community affiliation graph of probable community *C* memberships *M* of the nodes *V* in the network *G*. The output of the AMG method is visualized at the right of the figure, an unlabelled network graph *G*(*V*, *E*) with the links representing the edges between nodes. Reprinted from ‘Overlapping communities explain core–periphery organization of networks’ by Yang, J., and Leskovec, J., 2014.

Strengths

The AMG method has the assumption that nodes are more densely connected in overlapping parts of communities. That assumption about the structure of the data leads to a more accurate understanding of the network structure, compared with previous described methods. It is also possible to find nested communities with this method.

The problem of finding the parameters $\{p_c\}$ with the maximum likelihood of the observed edges in *G*, given the graph *G(V, E)* and the bipartite community affiliation network *B(V, C, M)* is transformed into a convex optimization problem with a change in the variables to maximize the logarithm of the likelihood ($1- p_k = e^{-xk}$) with a non-negativity constraint. This transformation speeds up the computation. Products become sums with this formula what makes it easier to calculate. The chance on numerical rounding errors is also decreased with this log transformation. After all, the concatenation of many numbers is less prone to this error than taking the product of many small numbers (Leskovec, Rajaraman & Ullman, 2014).

The Metropolis-Hastings method is used when the density is not known beforehand, this method evaluates it at least up to a proportionality constant. If the proposed value x_* for the next draw $x_{(n)}$, has a higher density under the target distribution than the current value $x_{(n-1)}$ it is always accepted with an acceptance probability. The acceptance probability is the ratio of the proposed value to the current value. In other words, if the proposed value x_* has a higher density than the current value $x_{(n)}$, it gets accepted (it becomes the new current value) with some probability, which is the ratio of $x_{(n)}$ to $x_{(n)}$. With a rejection of the proposed value, the current value $x_{(n-1)}$ becomes the new current value $x_{(n)}$ with an acceptance probability. That probability is in this case 1-the acceptance probability. This makes the method much faster because not the whole process gets repeated if a proposed value is not accepted. With this method it is also possible to adjust the membership strengths of individual nodes to a community, or the probability associated with a community in the affiliation graph model continuously. Another strength of this method is that it is applicable with high dimensional data. With acceptance-rejection sampling methods the probability that the proposed value gets rejected increases if the dimensionality of the data increases (Chib & Greenberg, 1995; Tsvetkov, Hristov & Angelova-Slavova, 2013).

Summarized, the strengths are that communities are found more accurately because of the assumption of dense overlaps. Calculating the logarithm of the likelihood instead of the likelihood makes it less prone to errors and speeds up the process. Using the Metropolis-Hastings algorithm during the stochastic search with gradient descent also makes the method faster. AMG can be used with ten times more data than the other described methods, what makes it more scalable (Yang, J., & Leskovec, 2012b).

Weaknesses

With the used sampling method, the Metropolis-Hastings algorithm, the samples are not independent. They depend on the previous value of the proposal and the acceptance probability calculations. However, under regularity conditions there are a lot of numbers and the central limit theorem allows Monte Carlo approximation. Monte Carlo approximation is the last step of the search to the global maximum. The Metropolis-Hastings algorithm can only find local maximum values in a region close to the gobal maximum. Monte Carlo looks at the last few parameter values if there is almost convergence. Then it takes the average of it, which gets accepted as the global maximum (Niemi, 2013). Another weakness of the method is that the output is a graph and it is difficult to extract the model (the final values of the bipartite affiliation network *B(V, C, M, {pc}*) from a graph. A solution is to use the BigCLAM algorithm to extract the model from the graph.

3.3.2 Cluster Affiliation Model for Big Networks (BigCLAM)

With this method non-negative latent factor estimations are computed. They model the membership strengths of the nodes to each community, given the unlabelled undirected network graph obtained with the AMG method described in the previous section.

The assumption is that data is generated by some model *f* with model parameters. These parameters are *B(V, C, M, {pc})*. The task is to find the most likely model that could have generated the data. Each edge between a node *u* and a community *A* (C_A) gets a non-negative latent factor estimation that represents the membership strength F_{uA}. Those non-negative latent factor estimations are put in a matrix *F,* with communities as columns and nodes as rows. Thus a whole row of the matrix is the vector of community membership strengths of a node. The probability that nodes in the same network are connected depends on the *F* values. The product of the community membership strengths is proportional to the probability of a link between two nodes. An example of an *F* matrix is given in Table 2. Node *u* and *w* are both member of community *B*. The product of their community membership strength to *B* is 1.2*1.8 = 2.16. Thus the probability of an connection between those nodes is $1\text{-}e^{-2.16} = 0.88$. The probability of an edge between node *v* and *w* is $1\text{-}e^{-(0.15+0.6)}$=0.53, and that of node *u* and *v* is 0.

	C_A	C_B	C_C	C_D
$\boldsymbol{F_u}$	0	1.2	0	0.2
$\boldsymbol{F_v}$	0.5	0	0.6	0
$\boldsymbol{F_w}$	0.3	1.8	1	0

Table 3: An example of an *F* matrix with the membership strengths of three nodes (*u, v, w)* to four communities (*A*, *B*, *C*, *D*).

Thus, if one of the nodes does not belong to a community the probability will be 0 and if two nodes share multiple communities the probability will be higher. To allow a very small probability that nodes that do not share a community have a connection, an additional ε-community is made. This ε-community connects any pair of nodes, with the back ground edge probability between a random pair of nodes.

To find the most likely model that could have generated the data, the most likely affiliation factor matrix *^F* to the underlying unlabelled undirected network *G(V,E)* has to be found. The optimization problem is to find the maximum log-likelihood of the network given *F*. This is done with a variant of nonnegative matrix factorization, in which *F* estimates the adjacency matrix *A* of a given network best. The negative log-likelihood *F* is used as a loss function *D* instead of the l_2 norm (which is normally used with NMF) and 1-exp(*) as a link function in the transformed new problem. Block gradient ascent is used to solve the optimization problem/learn the model. The F_u (membership strength) for each node gets updated when the membership strengths of the other nodes F_v are kept fixed. By fixing F_v, updating F_u becomes a convex optimization problem that is solved with projected gradient ascent. The updated F_u is projected into a space of nonnegative vectors, by setting the max likelihood of F_{uc} or 0. The F_u iteratively gets updated for each single node until the likelihood almost does not increases anymore. After this ^*F* is known, if the value of F_{uc} is above a threshold (the probability ε), it gets assigned to community *c*. To finalize *F*, locally minimal neighbourhoods is used. Node *u* is with its neighbours (connected nodes *v*) locally minimal, if they are less densely connected than all the neighbours of *v* (Leskovec, Rajaraman & Ullman, 2014; Yang & Leskovec, 2013).

Strengths

The latent factors are interpreted using non-negative matrix factorization (NMF). NMF also reduces the dimensionality of the data, what speeds up the computation. With large datasets it is not convenient to consider each component every time an observation is handled. With NMF the components are broken down into fewer representative components. The observations are reduced to a linear combination of sparse bases and sparse weights that represent them. Not all parts have to be used when they are put together to get the representation of the observation. This means that computation time decreases. This method outperforms Vector Quantization (VQ) for instance in accuracy of representation of the objects. VQ forms dense bases of prototypes of the observation (so not parts-based representations) and selects the most similar one as representation. It also outperforms PCA, which forms also holistic representations that are not sparse. With PCA there are positive and negative components and the weights blends them together. But PCA and VQ are also less suitable in this case, since they do not have a non-negativity constraint (Lee & Seung, 1999). The strengths of the BigCLAM method are that it identifies community structures more accurately. It can be used with bigger networks, networks that have more than 35 million edges. While it almost has linear running time, this is with other community detection methods quadratic or exponential (Yang & Leskovec, 2013).

Weaknesses

Stochastic gradient descent is a better method than (batch) gradient descent because it does not take all observations into account to modify the parameters in each iteration. Instead it modifies the parameters a little bit so it fits the (in a random order chosen) training examples better. This makes it much faster and useable for big datasets. However, it does not converge at the global maximum like batch does. It walks continuously in some region close to the global maximum, but does not just get it and stays there. Fortunately most of the time the parameters end up close to the global maximum and that is most of the time a hypothesis that is good enough for practical purposes (Ng, 2015). Because it is possible to scale up to much bigger datasets with stochastic gradient descent, the added value of the scalability and speed makes it a very useful step. The assumption is that the connections in the overlapping parts of the communities are more dense. This assumption is statistically most probable the case. However, it is possible that nodes in a dataset are more sparsely connected in the overlapping parts. If this is the case, the other community detection methods may fit the data better.

3.4 Evaluation community detection methods

Although different community detection methods have different assumptions about specific connectivity structures, the global idea is the same. Communities are dense subgraphs. Groups of nodes with more internal connections than external. Some methods only identify non-overlapping communities, some identify sparsely connected and others densely connected overlapping communities. In the end a community, overlapping or not, is more densely connected compared with its nodes with the rest of the network. There are many evaluation measures that are used to identify and to evaluate the quality of communities (Yang & Leskovec, 2015).

The evaluation measures can be focussed on the internal connectivity of the communities, like the internal density, the number of internal edges, the fraction of nodes in a community with a higher internal degree than the median or the fraction of nodes that belong to a triad. Scorings functions that focus on external connectivity measure the external connection degree or the fraction of external links for instance. There are also multiple measures that combine internal and external scoring functions, and modularity, which is based on a network model. As it is mentioned in the introduction, Yang and Leskovec (2015) created an evaluation method based on ground truth communities. This method can be used for each community detection method if the real communities of the network are known, which is often not the case. They also developed four goodness metrics that describe desirable characteristics of communities. These can be used to evaluate the goodness of the identified communities. These definitions state that good communities have a low external internal connections ratio (separability) and many internal connections (density). The internal nodes are evenly well connected, many nodes should be moved before it can be split into two sub communities (cohesiveness) and the locally distributed connections have the characteristics of small world networks (clustering coefficient).

4. Applicability purchase pattern mining methods with real world cases

Both discussed ARM methods give an insight in the relationships between products that is easy to interpret. The output reflects the real relationships between products, because no probabilistic assumptions are made. If only a few different products are sold, these methods are easy to use. Most stores however, sell many different products. The more products that are taken into consideration, the harder it is to extract meaningful information using these methods. The amount of rules can be decreased with increasing the minimum support threshold. Most rules are inexplicable because they are composed of the same products with an different order. The consequence is that the relationships between the other products in the assortment still remains unknown. Selecting certain product categories for the analysis is not an option if you want to have a representative picture of the situation, because it is not possible to include only selected products in the antecedent and all (thus also not selected) products in the consequent. However, it is possible to extract many association rules and filter the ones one is interested in.

This makes ARM methods useful for stores with a big assortment if they are used for a small timespan around a certain event or holiday for instance. ARM used in the period around Christmas can show that Christmas ball hooks are often bought together with (power) sockets for instance. A rule in which the consequent is obvious if you hear it, because people need extra sockets for the Christmas decoration, but not something you will think of first hand. Those methods are also useful if only a selection of the rules (that contain certain products) are filtered out afterwards. By selecting all rules that contain a product that was promoted for instance, the effectiveness of the marketing action can be evaluated. Another possibility is to get insight in only a few significant product relationships, or to compare those top *n* most frequently bought item sets per store.

If a real market basket analysis has to be done. In which the general relationships between all products of the assortment become clear, ARM methods are not applicable in real-world cases. These methods by their self are also not applicable, if the differences between product relationships per month or per store are sought. Only a fraction of the products will be included in the rules that can be analysed with by a human.

A possible solution for this problem is to use an ARM method in combination with a clustering method. The large amount of association rules gets partitioned in *k* similar groups. A downside of this procedure is that the products are forced into groups, what gives the possibility that products that are not that often bought together are clustered with products that occur often in the same transaction. However the clusters can be analysed independently afterwards. In this way there is a bit more structure in the hundreds or thousands of rules. This structure can be used as framework to dive deeper into the generated association rules (selecting them) and finding out which patterns are more apparent than others, by looking at the interestingness scores or making more targeted visualisations.

This is theoretically a suitable course of action, but the results obtained from this combined method will differ enormously depending on the clustering method that is used. That is why an understanding of the method, the assumptions it makes about the data and the calculations it makes during computation that lead to the groups that are made, is necessary before using it. K-means divides the groups in such a way that the similarity between the data points (rules in this case) in the same group is as high as possible and the similarity between those points with those of different groups is as low as possible. The data gets forcefully separated while some points may belong to multiple groups. Besides that the groups are forced to have the same size and symmetrical structure (in the form of a convex), while if a human would separate (a small part of) the data manually, there can be some groups with a few data points in it, some with many, and some will have the form of an very sharp oval while others may be perfect circles. In other words, this method is not applicable for finding groups of association rules that reflect the reality. Because the components in the association rules are often not represented equally and there is also a lot of overlap, a mixture model may be better suited to group the rules. With the DPGMM the number of groups does not have to be defined beforehand which prevents unrealistic forceful partitioning. Because this method is not empirically proven, the output can be verified using the GMM with the same and a few higher/lower *k* values.

Now maybe the question arises, why one would first generate association rules to cluster them afterwards. The clustering can also be done on the POS data right away. The reason is, that the goal is to find useful rules. If each transaction has to be taken into consideration with the clustering of the product categories, a lot of noise is captured in the groups. Each transaction is a data point that has to be assigned to a group, while not all product combinations in the transactions are of the same importance. With this is meant that combinations that almost never occur together in a transaction, also have to be assigned to a group. This is also the case if association rules are clustered, but then more noise is already filtered out. Of course the soft probabilities of the cluster assignments can be reviewed as solution for this problem, but this takes a lot of time and effort. Taking everything together, clustering methods will give a better overview of the relationships within the assortment than ARM methods. Still, if an accurate overview has to be found, the use of a clustering method by itself is often not enough.

For this reason it may be best to use ARM methods for certain events or products for instance, and community detection methods that capture for an overview of the relationships. Those methods capture the whole structure of the data. The results are given together with visualizations which makes it possible to interpret a lot of information in a fast and easy way. The relationships are reflected with connections between the products instead of partitions of the data. Because of this, less valuable information is lost. The strength of the relationships between products can be given in thickness of the link between them. With community detection methods that identify (non)overlapping and nested communities, the groupings of products can be reviewed at different levels. It is possible to analyse the relationships between products in an individual community, or all relationships in the assortment at once.

Experimental setup

The dataset

The dataset of consisted of point-of-sale (POS) data of the low price retailer the Action. It contained more than 380 million transactions made in >600 stores within the period of 02-01-2013 till 24-11-2015. The POS data captured inter alia the following information: the date and time of the transaction, ticket-number, the location of the store (plus some aggregations of this: location code, the city, municipality, country, longitude/lattitude), the name of the article (plus some aggregations of this: goods group, presentation group, article group, main category) and which articles were promoted during that period. This data was stored in column based tables in a SAP Hana database.

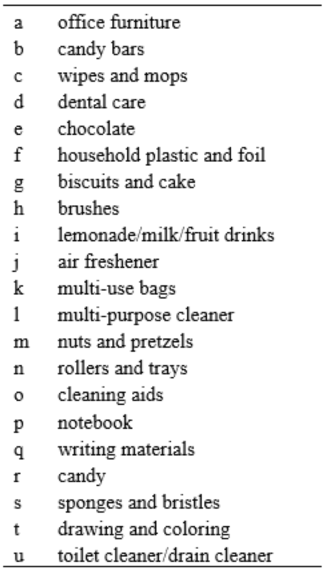
Since objective of this research was to compare the accuracy of different methods in finding relationships between products based on co-occurrences, only a subset of the data is used. The choice is made to use the transactions made in one store located in Amsterdam, during the first week of October 2014. This selection consisted of 42.138 transactions and 303 product categories and is referred to as the whole dataset from now on. The data was stored in 142.348 rows, with the transaction number and the name of the product group as columns. Because multiple are products bought per transaction, the dataset was composed of 142.348 products that were bought in 42.138 different transactions. The product group is referred to as product from now on. The city of Amsterdam was chosen because it has a lot of variety in the nationalities of the inhabitants (and visitors) making it a heterogeneous sample (Centraal Bureau voor de Statistiek, 2014). The month October is chosen because there no national feasts or holidays that could influence results in terms of extraordinary relationships. In sum, for the purposes of this thesis, there is focussed on a representative sample of the transactions of the retailers' stores in the Netherlands during one week.

Procedure

The real co-occurrences in the dataset

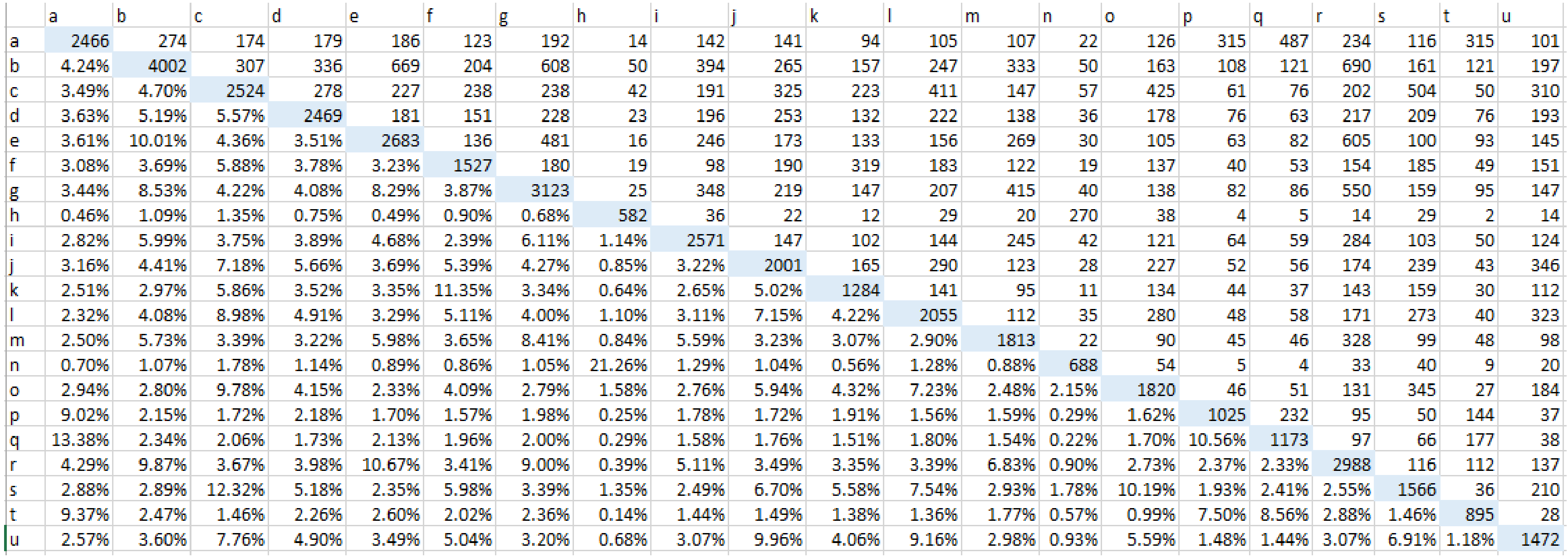
In order to make a comparison of the generated product relationships with the real co-occurrences in the dataset possible, a co-occurrence matrix is made. In order to show and interpret the co-occurrences and the results easily the Apriori method is used to make a selection of the products of which the co-occurrences are compared. The apriori is applied with a low support threshold and the 50 association rules that had the highest support values were selected. The lowest support value one of those 50 association rules had was 0.008 and the highest was 0.018. Other measures like the confidence, lift or interestingness give more valuable information for retailers when relationships between products are sought. Given that this research is about the representativeness of the generated product relationships in terms of similarity with the real (co-occurrences in the) dataset, the real fraction of co-occurences is more suitable than other measures. The support is the fraction of times that item sets are present in the dataset, what makes it the most reliable measure to use.

The 50 rules with the highest support values contained 21 unique products. Each rule consisted of two products and almost every second rule was composed of the same products (in the opposite order) as the previous one. The 21 selected items were sold 40.727 times during that week, in 22.466 distinct transactions. If the whole product names would be used, the column names of the co-occurrence matrix would not be readable anymore. For this reason the product names are labeled with one character. The 21 selected products with their letters can be seen in Table 4. In Table 5 the real co-occurrences of products are given at the right top of the matrix and the co-occurrences in percentages at the left bottom. The numbers in the blue cells are the total amount of times the products were bought. The co-occurrence percentage of products *a* and *b* is 274/(2466+4002) = 4.24%. During the comparison of the methods the co-occurrence matrix is made symmetrical with the values in percentages.



a	office furniture
b	candy bars
c	wipes and mops
d	dental care
e	chocolate
f	household plastic and foil
g	biscuits and cake
h	brushes
i	lemonade/milk/fruit drinks
j	air freshener
k	multi-use bags
l	multi-purpose cleaner
m	nuts and pretzels
n	rollers and trays
o	cleaning aids
p	notebook
q	writing materials
r	candy
s	sponges and bristles
t	drawing and coloring
u	toilet cleaner/drain cleaner

Table 4: The 21 products that were present in the 50 association rules that had the highest support values are listed in the second column of the table. The characters in the first column are the letters that are used to refer to those products in the rest of this paper.



	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u
a	2466	274	174	179	186	123	192	14	142	141	94	105	107	22	126	315	487	234	116	315	101
b	4.24%	4002	307	336	669	204	608	50	394	265	157	247	333	50	163	108	121	690	161	121	197
c	3.49%	4.70%	2524	278	227	238	238	42	191	325	223	411	147	57	425	61	76	202	504	50	310
d	3.63%	5.19%	5.57%	2469	181	151	228	23	196	253	132	222	138	36	178	76	63	217	209	76	193
e	3.61%	10.01%	4.36%	3.51%	2683	136	481	16	246	173	133	156	269	30	105	63	82	605	100	93	145
f	3.08%	3.69%	5.88%	3.78%	3.23%	1527	180	19	98	190	319	183	122	19	137	40	53	154	185	49	151
g	3.44%	8.53%	4.22%	4.08%	8.29%	3.87%	3123	25	348	219	147	207	415	40	138	82	86	550	159	95	147
h	0.46%	1.09%	1.35%	0.75%	0.49%	0.90%	0.68%	582	36	22	12	29	20	270	38	4	5	14	29	2	14
i	2.82%	5.99%	3.75%	3.89%	4.68%	2.39%	6.11%	1.14%	2571	147	102	144	245	42	121	64	59	284	103	50	124
j	3.16%	4.41%	7.18%	5.66%	3.69%	5.39%	4.27%	0.85%	3.22%	2001	165	290	123	28	227	52	56	174	239	43	346
k	2.51%	2.97%	5.86%	3.52%	3.35%	11.35%	3.34%	0.64%	2.65%	5.02%	1284	141	95	11	134	44	37	143	159	30	112
l	2.32%	4.08%	8.98%	4.91%	3.29%	5.11%	4.00%	1.10%	3.11%	7.15%	4.22%	2055	112	35	280	48	58	171	273	40	323
m	2.50%	5.73%	3.39%	3.22%	5.98%	3.65%	8.41%	0.84%	5.59%	3.23%	3.07%	2.90%	1813	22	90	45	46	328	99	48	98
n	0.70%	1.07%	1.78%	1.14%	0.89%	0.86%	1.05%	21.26%	1.29%	1.04%	0.56%	1.28%	0.88%	688	54	5	4	33	40	9	20
o	2.94%	2.80%	9.78%	4.15%	2.33%	4.09%	2.79%	1.58%	2.76%	5.94%	4.32%	7.23%	2.48%	2.15%	1820	46	51	131	345	27	184
p	9.02%	2.15%	1.72%	2.18%	1.70%	1.57%	1.98%	0.25%	1.78%	1.72%	1.91%	1.56%	1.59%	0.29%	1.62%	1025	232	95	50	144	37
q	13.38%	2.34%	2.06%	1.73%	2.13%	1.96%	2.00%	0.29%	1.58%	1.76%	1.51%	1.80%	1.54%	0.22%	1.70%	10.56%	1173	97	66	177	38
r	4.29%	9.87%	3.67%	3.98%	10.67%	3.41%	9.00%	0.39%	5.11%	3.49%	3.35%	3.39%	6.83%	0.90%	2.73%	2.37%	2.33%	2988	116	112	137
s	2.88%	2.89%	12.32%	5.18%	2.35%	5.98%	3.39%	1.35%	2.49%	6.70%	5.58%	7.54%	2.93%	1.78%	10.19%	1.93%	2.41%	2.55%	1566	36	210
t	9.37%	2.47%	1.46%	2.26%	2.60%	2.02%	2.36%	0.14%	1.44%	1.49%	1.38%	1.36%	1.77%	0.57%	0.99%	7.50%	8.56%	2.88%	1.46%	895	28
u	2.57%	3.60%	7.76%	4.90%	3.49%	5.04%	3.20%	0.68%	3.07%	9.96%	4.06%	9.16%	2.98%	0.93%	5.59%	1.48%	1.44%	3.07%	6.91%	1.18%	1472

Table 5: The co-occurrence matrix of the 21 selected products used during the comparison of the methods. At the right top the real numbers are given and at the left bottom the co-occurrences in percentages.

The input of the methods

With the clustering algorithms, k-means clustering, DPGMM and GMM, the whole final dataset is used. As it is mentioned in ‘The dataset’, the whole dataset consist of the transactions made in one store located in Amsterdam, during the first week of October 2014. It is possible to include the amount of products that were sold during the transaction as extra variable with those methods. This is not done, because this variable cannot be used with the network analysis methods. If this information was added with the clustering methods, the dataset would not be the same anymore and for that reason the comparison would not be valid. The format of the dataset is changed in order to fulfil the input structure requirements of clustering methods. As it is explained in the literature review, all features (products) of the objects (transactions) have to be composed of the same information in the same order. In order to fulfil this requirement, a list is made with each transaction represented as a dictionary, with all products in it as keys. The value of the key was 1 if the product was bought during the transaction and 0 if this was not the case. When each transaction was in this list, the DictVectorizer of scikit-learn is used to make arrays of the type float of them, and the input requirements were fulfilled (Pedregosa et al., 2011).

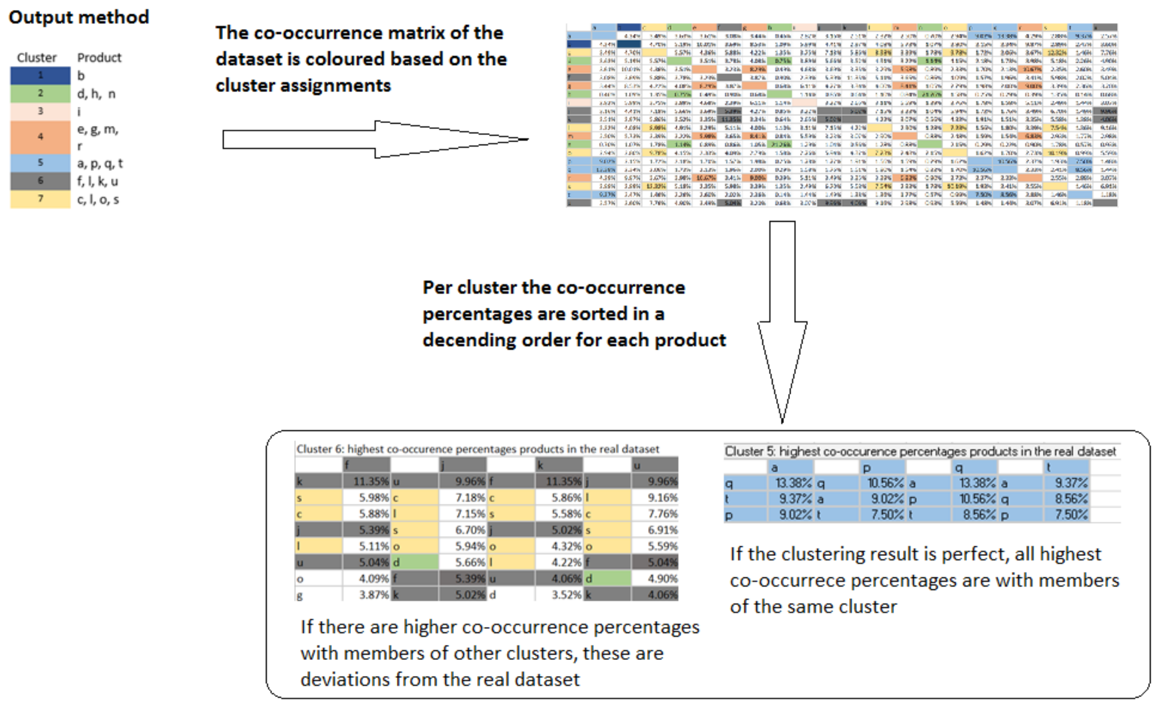
The network analysis methods have different input structure requirements than the clustering methods. They require pairs of nodes (products that were bought during the same transaction) as input. For this reason all possible product combinations of products with the same transaction number were combined as pairs on a row with an unique identifier. There were 335,581 pairs present in the data. Just like with the clustering methods, the whole dataset is used with the Louvain method. With the CPM only the product pairs that consisted of two of the 21 selected products were used because of insufficient computer memory (this is further explained in concerning paragraph in the ‘Experiments’ section).

Evaluation of the methods

After obtaining the results, consisting of groups of products (based on co-occurrences in the transactions), the 21 selected products were filtered. Thus only the products that were used during the comparison with the real co-occurrences were left in the groups. The reason that the other products are removed after the experiment and not beforehand, is that the dataset would be less similar with datasets that are used during applications of the methods in real world situations. Since no existing evaluation measure that can be used to compare the similarity of the product relationships generated with the methods with the real co-occurrences could not be found, two comparison methods were created to make a cross-category method comparison possible. The first comparison is applied on each method. Due to the structure of the output of some methods, it was not possible to evaluate the results of each method with the second comparison method.

The first evaluation method that compared product relationships of group members

As it is mentioned before, the results of each method are evaluated with the first evaluation method that compared product relationships of group members, which is created for this research. During the first evaluation method the co-occurrence matrix of the dataset is used. Each method gave groups (clusters or communities) of products as output. For illustrative purposes, the products that were grouped together were colored, just like their co-occurrence percentage. After this, for each group and for each product in the group the product combinations were sorted by their co-occurrence percentages in an descending order. This process is visualized in Figure 12. If the grouping went perfect all top products were from the same group. If a product occurred more often with a member of another group than with at least one other member of the same group (the white cells in Figure 12), this combination was regarded as an error. The error percentage is the difference between the co-occurrence percentage of the error combination and the lowest co-occurrence percentage with a member of the same group.


Output method
The co-occurrence matrix of the dataset is coloured based on the cluster assignments
Per cluster the co-occurrence percentages are sorted in a decending order for each product
If there are higher co-occurrence percentages with members of other clusters, these are deviations from the real dataset
If the clustering result is perfect, all highest co-occurrece percentages are with members of the same cluster

Cluster	Product
1	b
2	d, h, n
3	i
4	e, g, m, r
5	a, p, q, t
6	f, l, k, u
7	c, l, o, s

Cluster 6: highest co-occurence percentages products in the real dataset

	f		j		k		u
k	11.35%	u	9.96%	f	11.35%	j	9.96%
s	5.98%	c	7.18%	c	5.86%	l	9.16%
c	5.88%	l	7.15%	s	5.58%	c	7.76%
j	5.39%	s	6.70%	j	5.02%	s	6.91%
l	5.11%	o	5.94%	o	4.32%	o	5.59%
u	5.04%	d	5.66%	l	4.22%	f	5.04%
o	4.09%	f	5.39%	u	4.06%	d	4.90%
g	3.87%	k	5.02%	d	3.52%	k	4.06%

Cluster 5: highest co-occurence percentages products in the real dataset

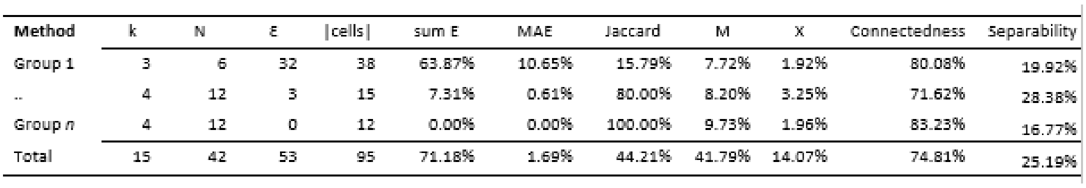
	a		p		q		t
q	13.38%	q	10.56%	a	13.38%	a	9.37%
t	9.37%	a	9.02%	p	10.56%	q	8.56%
p	9.02%	t	7.50%	t	8.56%	p	7.50%

Figure 12: A visualization of the first comparison method. The results of each method are compared with the real co-occurrences of members of the groups in the dataset.

Thus only the deviations between the products that were assigned to a group, and the most co-occurring products in the data were measured. It is not measured if a product should be assigned to a group while it is not. The reason that the products that are not assigned to the cluster and have lower co-occurrence percentages than every co-occurrence combination of members of the same cluster are not taken into account, is that it is not possible to state with certainty that those products are clustered right or wrong. It is only possible to see which products are not assigned to a cluster while they occur more often with a product of the concerning cluster than the products that are assigned to the same group.

This is explained with an example in the ‘K-means’ section of the experiments. If the grouping was done perfectly, the deviation and similarity are not measured because they would be 0% and 100%. Groups that contained only one of the selected products are also not evaluated, because it is not possible to compare the relationship between different products if there is only one product.

There is a difference between one nearly perfect group and one very bad one, and two ‘ok’ groups, while those can have the same evaluation score when they are taken as a whole. If the two generated groups are 80% similar and 10% similar, the evaluation score for the grouping as a whole is 45% similarity with the real data for instance. However, with a grouping in which one group is 40% similar and the other 50%, the evaluation score of the grouping as a whole is the same. For this reason the evaluation is done for the results of each individual group included in the results and for the results of the method as a whole. The evaluation of the result is done with different measures that are listed in a same table as Table 6. The number of group members is given as *k*. The number of co-occurrences with members of the group is given as N, this value is k*(k-1), it is the same number as the number of coloured cells per group. The ε in the table are the number of error combinations (the white cells), the column |cells| represents the number of cells of the cluster that are taken into account during the comparison. This are the co-occurrences of members plus the error combinations (N+ε). In order to measure the deviation with the dataset, the mean absolute error (MAE) is calculated for each group and for the grouping result as a whole. Because of the nature of the output, the calculation is done in a slightly different way than it is done by convention. As it is explained previously, the error percentages (the white cells in Figure 12), are the difference between the co-occurrence percentage of the error combination and the lowest co-occurrence percentage with a member of the group. In this way (error percentage and therefore) the results are not dependent on the frequency the products were bought together.



Method	k	N	ε	|cells|	sum E	MAE	Jaccard	M	X	Connectedness	Separability
Group 1	3	6	32	38	63.87%	10.65%	15.79%	7.72%	1.92%	80.08%	19.92%
..	4	12	3	15	7.31%	0.61%	80.00%	8.20%	3.25%	71.62%	28.38%
Group *n*	4	12	0	12	0.00%	0.00%	100.00%	9.73%	1.96%	83.23%	16.77%
Total	15	42	53	95	71.18%	1.69%	44.21%	41.79%	14.07%	74.81%	25.19%

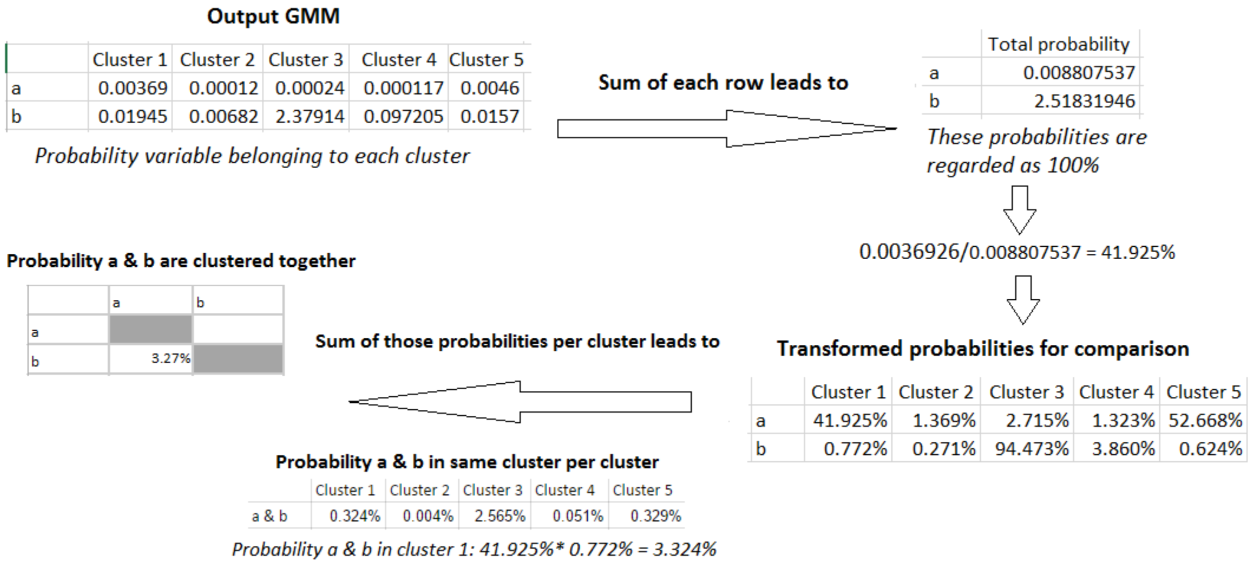
Table 6: A comparison of the results obtained with a method and the real co-occurrence percentages. The *k* in the table represents the number of selected products that is assigned to the group, *N* the amount of co-occurrences of members, ε the number of products that are not a member but occurred more often with members of the cluster, M the average co-occurrence percentage of members and X the average co-occurrence percentage with products of other clusters.

Per cluster the sum of the error percentages ε is divided by the number of co-occurrences with group members *N*, to obtain the MAE. The MAE should be as low as possible. The differences in error percentages do not give a full understanding of the components it is based on. It is possible that the evaluation score of a method with a few very big mistakes is the same as the score of a method that makes many little mistakes. To get more insight in the performance of the methods, also a binary measure is used to evaluate the similarity of the results with the dataset. With the Jaccard's similarity coefficient the similarity between the results and the dataset is given as the ratio of 'good' and 'wrong' group assignments. This means that the cluster assignment is either the same as the most occurring product combinations of the dataset or not. The Jaccard's similarity coefficient is calculated by dividing the intersect of the community assignment and the real occurrences of the dataset by the union of them. The formula is |a∩b|/|a∪b|, where |a∪b| is the same as: |a|+|b| - |a∩b|. In this case the intersect is the number of letters that are the same in the tables (*N*). The union is the number of cells that are taken into account from the clustering selection (|cells|) + the number of cells that are taken into account from the dataset (*N*), minus the intersect. However, in this case the number of cells that are taken into account from the dataset is always the same as the intersect (*N* == k*(k-1)). Thus the union has the same value as the number of cells that are taken into account from the clustering (union = |cells| + *N* – k*(k-1)). Thus with *N*/|cells| the Jaccard's similarity is obtained. This value should be as high as possible. As it is mentioned in the evaluation section of the methods in the literature review, connectedness and separability are important features for clustering and community detection methods. The connectedness of a group represents the ratio of internal-external connections, what is the co-occurrences with members to the co-occurrences with non-members ratio in this case. The seperatbility tis the opposite, thus the co-occurrence with non-members to the co-occurrence with members ratio. For each group the average co-occurrence percentage with group members is calculated (M in Table 5), and the average co-occurrence percentage of the members of a group with products that are not assigned to the same group is calculated (X in the table). These percentages together are not 100%. For this reason they can only be used to evaluate the groups per method, but not for a comparison of the different methods. In order to make a cross-method comparison possible the columns connectedness and separability are added. M+X is regarded as 100%. When M is divided by the total (M+X), the ratio of connectedness with other members to the total is obtained, this value should be high. The external internal connections ratio (separability) is the opposite, X is divided by the total, what should be low with good grouping.

The second evaluation method that compared all possible product relationships with the data

The second evaluation method that was created for this research, could only compare the results of the GMM and the Louvain method with the real co-occurrences of the dataset. The GMM assigned each product to each cluster with a certain weight. The output of the Louvain method also contained links between product pairs with a certain weight. This made it possible to take all product combinations into account during the comparison with the real co-occurrences. For both methods the results are transformed into a co-occurrence matrix. This made a comparison of with the co-occurrence matrix of the dataset easy. The difference of the values of the cells of the two co-occurrence matrices was regarded as the error percentage. A binary similarity measure is not used because of the extreme low chance on the exact same percentage in the matrices. For this reason only the MAE is calculated, which should be as low as possible.

In order to make a co-occurrence matrix of the results of the GMM, a few steps were taken. Those steps are visualized in Figure 13. First for each product, the total probability is calculated. This is the sum of the probabilities that the product belongs to one of the five clusters. Then the probabilities that the products belong to each cluster are recalculated with the total probability of step 1 as 100%. During the third step the probability that two products are clustered together is calculated for each cluster by taking the product of those probabilities. The total probability that the products are clustered together is the sum of those 5 probabilities. The probability that a pair of products is clustered together is calculated for each possible product combination (of the 21 selected products), and stored in a co-occurrence matrix.



Output GMM

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
a	0.00369	0.00012	0.00024	0.000117	0.0046
b	0.01945	0.00682	2.37914	0.097205	0.0157

Probability variable belonging to each cluster

	Total probability
a	0.008807537
b	2.51831946

These probabilities are regarded as 100%

0.0036926/0.008807537 = 41.925%

Transformed probabilities for comparison

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
a	41.925%	1.369%	2.715%	1.323%	52.668%
b	0.772%	0.271%	94.473%	3.860%	0.624%

Probability a & b in same cluster per cluster

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
a & b	0.324%	0.004%	2.565%	0.051%	0.329%

Probability a & b in cluster 1: 41.925% 0.772% = 3.324%*

Probability a & b are clustered together

	a	b
a		
b	3.27%	

Figure 13: The transformation of the result obtained with the GMM into a co-occurrence matrix.

The result of the Louvain method also provided a file with the links between the nodes with a weight. The file had the columns ‘node1’, ‘node2’, ‘weight’. The rows with one of the 21 selected items in both node columns were saved in another file. Then for each individual product, just like it is done with the analysis of the GMM results, the sum of the weights is regarded as 100%. During the second step, a co-occurrence matrix of the probabilities of product combinations is made. The co-occurrence probability is calculated by dividing the weight of the node combination by the total weight of that node.

The results of the methods that are compared

In this part the results that are obtained with the five datamining methods are described. For each method the software or source code that is used to apply the method is given. Followed by the final parameter settings, with the arguments that led to those decisions. Each part concludes with the results of the method and an evaluation by interpretation. The results of the method are the cluster or community assignments of the products, not the evaluation of the results of the comparison. These are discussed in the results section. As was mentioned before, the evaluation by interpretation describes whether the results make sense in the real world.

When do the results make sense in the real world?

Based on semantic similarity or co-occurrence in usage situations, five or six groups can be made, of which one or two are overlapping. Candy bars, chocolate, biscuits and cake, lemonade/milk/fruit drinks, nuts and pretzels, candy can be assigned to the same group because they are all products to eat or drink. Office furniture, notebook, writing materials, drawing and coloring can be grouped together because they can all be regarded as products that are used at school or at the office, and most of them are to write something down. Drawing and coloring can also be used for a more creative purpose, together with brushes and rollers and trays. Those three products can all be used for painting. Wipes and mops, air freshener, multi-use bags, multi-purpose cleaner, cleaning aids, sponges and bristles and toilet cleaner/drain cleaner can be grouped together since they are all products that are used to clean something, to put trash in, or in the kitchen/toilet/bathroom. It is also possible to make a separate group of wipes and mops, air freshener, cleaning aids and toilet cleaner/drain cleaner, since they are used for bigger and more dirty purposes. The other separate group would then consist of household plastic and foil, multi-use bags, multi-purpose cleaner, cleaning aids, sponges and bristles, with the reason that those products are often used in the kitchen or with smaller cleaning events. Dental care does not occur in similar usage occasions and it is semantically not really similar with the other products. It is also a cleaning product, but for the teeth, a part of the human body and not of a house like the previous mentioned group. It can be advised to use dental care products after eating and drinking, to prevent plaque and holes in the teeth. But this makes intuitively no sense to group dental care with those products since it would be the only product that is not for oral intake. The grouping of the products as discussed in this subparagraph is listed in Table 7. The products in group 4 and 5 can be divided in two groups like in Table 7, as well as grouped together like discussed before.

Group	Products
1 (food)	candy bars, chocolate, biscuits and cake, lemonade/milk/fruit drinks, nuts and pretzels, candy
2 (office)	office furniture, notebook, writing materials, drawing and coloring
3 (painting)	drawing and coloring, brushes, rollers and trays
4 (cleaning)	wipes and mops, air freshener, cleaning aids, multi-purpose cleaner, toilet cleaner/drain cleaner

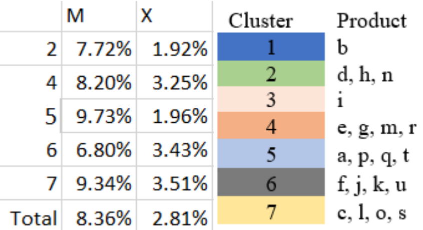
5 (kitchen)	multi-use bags, household plastic and foil, multi-purpose cleaner, cleaning aids, sponges and bristles
6 (p. hygiene)	dental care

Table 7: Group assignments of the 21 selected products based on semantic similarity or co-occurrence in usage situations in the real world. Group 4 and 5 can be regarded as one group as well as two separate groups, like in this table. The products drawing and coloring, cleaning aids, multi-purpose cleaner can belong to two groups.

K-means

K-means clustering is applied using python code provided by scikit-learn (Pedregosa et al., 2011). With k-means, the number of clusters (*k*) has to be defined beforehand. A *k* of seven is chosen because this partitioning led to variation in the cluster assignments of the selected products. Normally the within-cluster variance and the sum of squares are used as internal evaluation measures. Because of the nature of the task that is performed with this experiment, the decision is based on the distribution of the cluster assignments. The task is to find as many product relationships using the clustering method as possible. For this reason it is important that there are not many products that are alone in a cluster. With a *k* of 7, only products *b* and *i* were clustered alone. To be able to extract product relationships, a multiple different clusters with a few products in it are needed. This was also the case with a *k* of 7.

However, the possibility to evaluate the cluster assignments of the products by itself, says nothing about the quality of the clustering result. For this reason the connectedness and separation are reviewed to see if the clusters consisted of products that were more similar with members and less similar with non-members in terms of transaction co-occurrences. For each cluster the average co-occurrence percentage with members is calculated. This is just like in the 'Evaluation of the methods' section, M in Table 8. The average co-occurrence percentage with non-members is given in column X in Table 7. If the clustering went well, the difference between M and X is big, with M having the highest percentage. As it can be seen in the table, the products of the same cluster occur on average more often together in the real dataset (8.36%) than with products that are not in the same cluster (2.81%). This is also the case for each cluster independently. For this reason this partitioning of the dataset is kept for further analysis. The cluster assignments can be seen at the right of Table 8.



	M	X
2	7.72%	1.92%
4	8.20%	3.25%
5	9.73%	1.96%
6	6.80%	3.43%
7	9.34%	3.51%
Total	8.36%	2.81%

Cluster	Product
1	b
2	d, h, n
3	i
4	e, g, m, r
5	a, p, q, t
6	f, j, k, u
7	c, l, o, s

Table 8: At the left the cluster assignments per product, at the right the average co-occurrence percentage of products with other products that are assigned to the same cluster (M) and of products with those of other clusters (X). At the right the members of each cluster is shown.

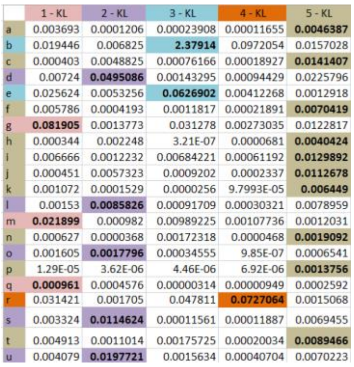
As can be seen in Table 9, some of the clusters make sense when evaluated by interpretation. Cluster 4 and 5 do not contain products that do not make sense in the real world. However, the products of cluster 1 and 3 can be added to that cluster. All products that are assigned to cluster 6 and 7, can be grouped together based on real-world knowledge. In the previous paragraph they were listed in the group cleaning and kitchen, which can also be grouped together. Nevertheless, if a division of those clusters is made, the products household plastic and foil and wipes and mops can best be switched. Dental care in cluster 2 does not make sense in combination with brushes and rollers and trays, that are used in similar situations.

Cluster	Product description
1	candy bars
2	dental care, brushes, rollers and trays
3	lemonade/milk/fruit drinks
4	chocolate, biscuits and cake, nuts and pretzels, candy
5	office furniture, notebook, writing materials, drawing and coloring
6	household plastic and foil, air freshener, multi-use bags, toilet cleaner / drain cleaner
7	wipes and mops, multi-purpose cleaner, cleaning aids, sponges and bristles

Table 9: The names of the products that are clustered together with the k-means algorithm.

GMM clustering

Clustering with the GMM is done with the use of the predictive analytics library of SAP. The data was split randomly into an estimation set that compromises 75% of the data and an validation set containing 25% of the data. The tool had an option to give a minimum and maximum value of *k* and it clustered the data twice (the estimation and the validation set) for each *k* value between (and including) the minimum and maximum. The results of each analysis with every *k* were given when the clustering was done. In this way it was easy to find the right *k* value without having to repeat the analysis and to save every result separately from each other manually. As it is described in the literature review, normally the number of *k* is fixed and has to be given beforehand. The algorithm used here was not different than the one that is described in the literature review, but the tooling that was used to do the analysis with had an additive option that made it possible to try different *k* values during the same analysis. Thus it is different than the DPGMM with which only the maximum *k* has to be given and the *k* that fits the data best is found during the computations using the Dirichlet Process. During the implementation the *k* values between 5 and 20 are used and program indicated that 5 clusters fitted the data best. The 21 selected products with the probabilities of them belonging to each cluster are given in Table 10. The highest cluster probabilities are colored based on the color each cluster got.



	1 - KL	2 - KL	3 - KL	4 - KL	5 - KL
a	0.003693	0.0001206	0.00023908	0.00011655	**0.0046387**
b	0.019446	0.006825	**2.37914**	0.0972054	0.0157028
c	0.000403	0.0048825	0.00076166	0.00018927	**0.0141407**
d	0.00724	**0.0495086**	0.00143295	0.00094429	0.0225796
e	0.025624	0.0053256	**0.0626902**	0.00412268	0.0012918
f	0.005786	0.0004193	0.0011817	0.00021891	**0.0070419**
g	**0.081905**	0.0013773	0.031278	0.00273035	0.0122817
h	0.000344	0.002248	3.21E-07	0.0000681	**0.0040424**
i	0.006666	0.0012232	0.00684221	0.00061192	**0.0129892**
j	0.000451	0.0057323	0.0009202	0.0002337	**0.0112678**
k	0.001072	0.0001529	0.0000256	9.7993E-05	**0.006449**
l	0.00153	**0.0085826**	0.00091709	0.00030321	0.0078959
m	**0.021899**	0.000982	0.00989225	0.00107736	0.0012031
n	0.000627	0.0000368	0.00172318	0.0000468	**0.0019092**
o	0.001605	**0.0017796**	0.00034555	9.85E-07	0.0006541
p	1.29E-05	3.62E-06	4.46E-06	6.92E-06	**0.0013756**
q	**0.000961**	0.0004576	0.00000314	0.00000949	0.0002592
r	0.031421	0.001705	0.047811	**0.0727064**	0.0015068
s	0.003324	**0.0114624**	0.00011561	0.00011887	0.0069455
t	0.004913	0.0011014	0.00175725	0.00020034	**0.0089466**
u	0.004079	**0.0197721**	0.0015634	0.00040704	0.0070223

Table 10: A selection of the clustering output obtained with the GMM with a *k* of 5. The selection contains the 21 selected products that are used during the comparison. The highest probabilities are coloured.

The hard cluster assignments obtained with the GMM algorithm make less sense than those obtained with the k-means if they are evaluated by interpretation. An overview of the products per cluster is given in Table 11. Writing materials does logically not belong in cluster 1 and cluster 3 and 4 can be merged with cluster 1. Dental care should be clustered alone, the other four products in cluster 2 are related to each other in real life. Cluster 5 is composed of the most products, those products fall into the categories with the subjects office, painting and cleaning/kitchen.

Cluster	Product description
1	biscuits and cake, nuts and pretzels, writing materials
2	dental care, multi-purpose cleaner, cleaning aids, sponges and bristles, toilet cleaner/drain cleaner
3	candy bars, chocolate
4	candy
5	office furniture, wipes and mops, household plastic and foil, brushes, lemonade/milk/fruit drinks, air freshener, multi-use bags, rollers and trays, notebook, drawing and coloring

Table 11: The names of the products that are clustered together with the GMM.

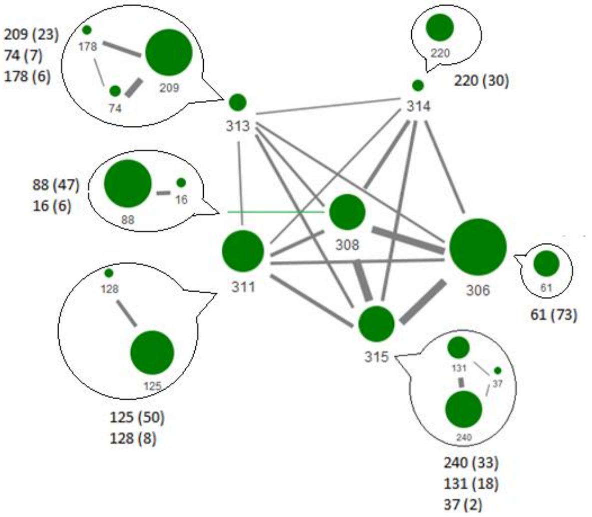
DPGMM

The idea was to do use the DPGMM algorithm before the implementation of the GMM, to have some approximation about a good *k* value (and to check if the results would be as good as the GMM). Scikit-learn is used for the implementation of the DPGMM. The first time the maximum *k* value was set to 30 and the output was one cluster. This is repeated for different upper bounds, but the result was always one cluster. A possibility is that the method (just like most other clustering methods) are actually created for continuous data. In this case the only variable (product) is categorical, and highly dimensional. In order to use the algorithm the dataset is transformed into a list of dictionaries. Each dictionary contained all products that were sold in that period (303) as keys with a '1' as value as the transaction contained the product and otherwise a '0'. These dictionaries were then transformed with the DictVectorizer of scikit-learn, in order to make arrays of floats of the data. Although this is often a good solution when categorical data has to be clustered, it can be the case that the transformed data was just to sparse with too many (solely binary) dimensions. The clusters are made based on the distances between the vectors, if each vector has 303 dimensions that are either a 1. or a 0. (and most of the time 0.), it can be hard to find points that are more close to each other than to other groups of points.

The Louvain method

The automated predictive library of SAP is used to apply the Louvain method. The algorithm detected on the lowest level six communities with 18 links and a modularity of 0.128. This means that there were a little bit more edges within the communities than that would be expected on the basis of chance. The communities with their structure are visualized at the left of Figure 14. The size of the dot represents the amount of nodes in the community. The thickness of the lines between the communities represent the amount of links between the different communities. On the highest level, those six communities consisted of 12 sub communities. Those are added next to the community in the balloon in the figure, with the identifier and the number of nodes in the communities within the brackets next to it. Those 12 communities consisted of 71 links and had a modularity of 0.121.

The 21 nodes that are selected for the comparison were member of five different communities. The table at the right of Figure 14 shows which nodes belonged to which community. The column 'Role' describes the distribution of the connections of the nodes. There are four types of roles a node can have. Local nodes are densely connected with the other nodes of the same community and have not many connections to other communities. Passive nodes have both a few intra and inter community connections. Social nodes have many intra and inter community connections. Bridge nodes have many connections with other communities and less connections with their own community, they act like a bridge between the different communities. All nodes that are selected for the evaluation are bridge (N=10) or social (N=11) nodes.



Product	Community	Role	Intra	Extra
a	220	Bridge	29	257
b	88	Social	45	246
c	240	Bridge	31	254
d	61	Social	71	215
e	88	Social	45	246
f	240	Social	32	243
g	88	Social	44	245
h	209	Bridge	18	215
i	88	Social	43	240
j	240	Social	32	247
k	240	Social	32	242
l	240	Bridge	31	252
m	88	Social	42	235
n	209	Bridge	20	225
o	240	Bridge	31	249
p	220	Bridge	28	237
q	220	Bridge	28	239
r	88	Social	44	243
s	240	Social	32	241
t	220	Bridge	29	232
u	240	Bridge	31	245

Figure 14: At the left the community structure generated with the Louvain method. The graph in the middle are the communities at the lowest level. The graph in the balloons are the communities at the highest level. In the table at the right of the figure for each product, the community it belongs to, the role it has and the number of internal and external connections are given.

The names of the products that are assigned to each community are listed in Table 12. There is no product membership assignment that does not make sense. The cleaning and kitchen groups are merged together and the products that can be regarded as a member of two groups are assigned to one of them.

Community	Product description
220	office furniture, notebook, writing materials, drawing and coloring
88	candy bars, chocolate, biscuits and cake, lemonade/milk/fruit drinks, nuts and pretzels, candy
240	wipes and mops, air freshener, multi-use bags, multi-purpose cleaner, cleaning aids, sponges and bristles, toilet cleaner/drain cleaner
61	dental care
209	brushes, rollers and trays

Table 12: The names of the products that are clustered together with the Louvain method.

Clique perlocation method

The CPM is implemented using CFinder of Palla et. al (2005). As it is mentioned in the 'The input of the methods' section, first the same input data that was used with the Louvain method was used, but the

CPM method did not work because of an ‘out of memory error’. For this reason all rows that contained one the 21 selected product combinations in both columns were saved in a new dataset that was used as input. The result was one community that consisted of each product. The community is visualized at the left of Figure 15. It had 21 vertices and 207 edges. For a *k* value of 3 till a *k* value of 18 there were 4 cliques with 18 to 19 products in it. As reminder, all nodes are connected with each other in a clique. The community with a *k* of 19 consisted of 20 vertices and 189 edges, product *h* was excluded from this community. This community had two cliques in it, with 18 and 19 products in it.

Cliques consisting of 19 of the 21 products indicate that the method does not fit the data. There were 210 unique product combinations in this dataset while there were only 21 different products. This means all the products were connected at least one time. The frequency that each combination occurred can be seen at the right of Figure 15.

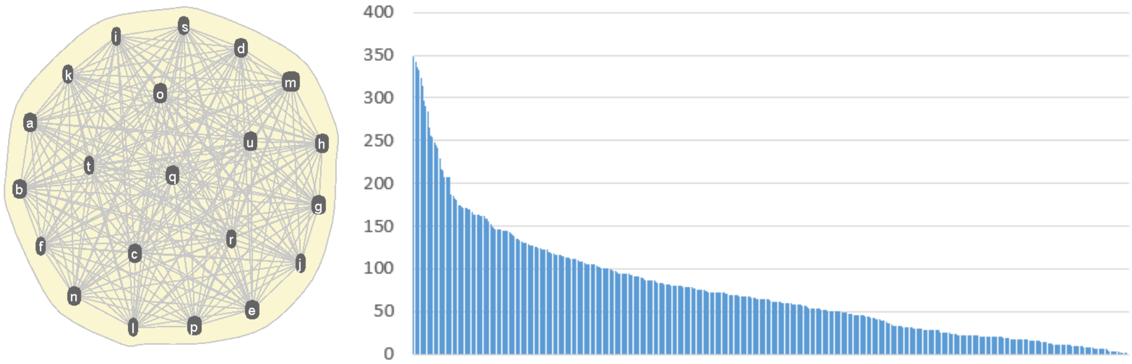


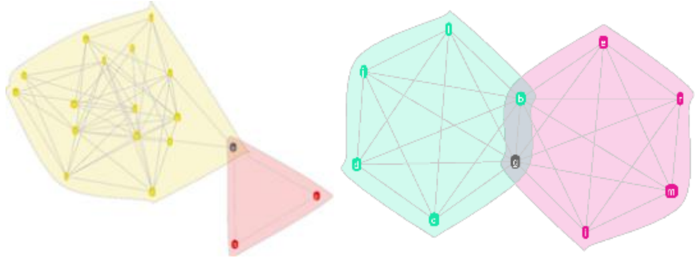
Figure 15: At the left the first result of the CPM, all products were part of the same community. At the right the frequency (y-axis) of each product combination (x-axis) occurred in the dataset is visualized in a bar plot.

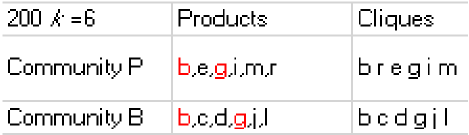
In order to try to generate more than one community with this method, different selections of the product combinations are used, based on link thresholds. The selection of the product combinations with a link threshold of 150, contained 40% of the product pairs in the dataset. With this subset 2 communities were generated with a *k* of 3 and 1 community for the *k* values above 3. This was also the case for the selection with a link threshold of 160. With a threshold of 170, 35% of the product pairs were included. As it can be seen in Table 13, two communities were found with a *k* of 7. One with 10 products, 39 edges and 4 cliques in it, the other with 7 items, 21 edges and one clique. The products *c, d, j, l* were member of both communities. Since half of the data belonged to one community, a higher link threshold is used.

170 *k* =7	Products	Cliques
Community 1	a,b,c,d,e,g,i,j,l,r	a b c d e g r b c d e g r i b c d e g r j b c d g r j l
Community 2	c,d,j,l,o,s,u	c d j l u o s

Table 13: The result of the CPM with a selection of the data, in which each product pair occurred at least 170 times. The product combinations that occurred less often were excluded.

With a link threshold of 200, 55 unique product combinations were selected. This selection contained 26% of the product pairs in the data. Products *h*, *n* and *t* were not member of a community. With two different *k*-values multiple overlapping communities were found. A visualization of those communities can be seen in Figure 16. If *k* was 3 and the selected product was *a*, one community had 16 products as member and 50 edges. The other community had 3 members with 3 edges, product *a* was the only node in the overlapping parts of the communities. There were also two overlapping communities detected with a *k* of 6. In the table at the right of Figure 16, the members and cliques of those communities are specified. Both communities consisted of 6 items and 15 edges. Products *b* and *g* were member of both communities.





200 *k* =6	Products	Cliques
Community P	b,e,g,i,m,r	b r e g i m
Community B	b,c,d,g,j,l	b c d g j l

Figure 16: The CPM result with the subset of the data that contained 55 unique product combinations that occurred at least 200 times in the dataset. At the left two overlapping communities with *k* =3 and *a* as selected product are visualized and in the middle two overlapping communities that were identified with *k* =6. The table at the right specifies the products and cliques in those communities.

The evaluation of the product membership assignments based on semantic similarity and similar usage situations can be done by inspection of Table 14. Community P is almost the same as the first group (food) that is described in the beginning of this section. All those products are to eat or drink. The product assignments to community B are not logically in the real world. The products that can be grouped together because they are used in painting situations are not present in the communities at all. The products in community B are fall (besides the painting group), under all other in the real world logically divided categories.

Community	Product description
P	candy bars, chocolate, biscuits and cake, lemonade/milk/fruit drinks, nuts and pretzels, candy
B	candy bars, wipes and mops, dental care, biscuits and cake, air freshener, multi-purpose cleaner

Table 14: The names of the products that are clustered together with the CPM.

Results of the comparison of the product relationships with the co-occurrences

The objective of this study was to find the method that can be used to find relationships between products in the assortment, with results that were most similar with the real co-occurrences in the data. The results obtained with k-means, GMM, the Louvain method and CPM are compared with the real co-occurrences in the dataset in two different ways. With the first evaluation method that compared product relationships of group members it was possible to compare the results of each method. With this method hard group assignments are used, with the result that not each possible product relationship could be evaluated due to the structure of the output. With the second evaluation method that compared all possible product relationships with the data, soft group assignments are used with the GMM and link weights with the Louvain method. The structure of the results of those two methods made it possible to compare all possible product relationships that were generated with the method with the real co-occurrences in the dataset. Each table that is used to obtain the results that are given in this section, can be found in the appendix.

Results of the first evaluation method that compared product relationships of group members

The results of all methods are compared with the first evaluation method. During the first evaluation method the co-occurrence percentages of group members were coloured in the co-occurrence matrix of the dataset for illustrative purposes during the first step. Per cluster, the co-occurrence percentages were ordered descendently and the co-occurrence percentages with non-members that were higher than those of the lowest co-occurrence percentage with a member were regarded as error combinations. The error percentage was defined as the difference between the co-occurrence percentage of the error combination and the lowest co-occurrence percentage with another member of the same group. This way of analysing similarity with the real co-occurrences had the consequence that not all possible product relationships are taken into consideration.

K-means

The results of the comparison between the k-means clustering result and the real co-occurrences in the dataset can be seen in Table 15. Only one fifth of the existing product relationships could be analysed with the results obtained with the k-means method with a *k* of 7 (42 of the 210). This means that a lot of information is lost and not analysed. Two of the seven clusters had one of the selected products in it and could not be compared for this reason.

The sum of the error percentages in cluster 2 was the highest with a value higher than 60%. On average the product relationships deviated 10.65% in that cluster. The number of product relationships per cluster (*N*) is compared with the amount of errors per cluster, using Jaccard's similarity coefficient. With six product relationships and with 32 errors in it, the clustering result of cluster 2 was only 15.79% similar with the dataset. Cluster 6 had in total an error percentage higher than 25%. Since there were 12 product relationships evaluated in this cluster, the average deviation in percentages per product relationship was 2,26% in this cluster. The clustering result was measured with Jaccard's similarity coefficient 41.38% similar with the dataset. The other clusters had a mean absolute error of less than 1%. Using the first comparison method, cluster 5 was clustered perfectly with a mean absolute error of 0% and a similarity of 100%. Like it is explained before, it is not possible to evaluate clustering result with the first comparison method, in terms of products that occur less often with members of a cluster but should be a member. However, is possible to evaluate in a more general way if clusters are missing members or not. If the connectedness and separability values are close to each other, the indication is that some products should be member of the community while they are not, or the other way around.

Cluster 6 had the lowest connectedness percentage with a value of 66.5%. This indicates that the product relationships apparent in the clustering result, are not much higher than the relationships with members of other clusters. Cluster 5 is clustered the best with a connectedness of 83% and a separability of 17%. Cluster 2 was clustered second best when the connectedness and separability measures are used. That did not make a lot of sense because MAE was the the highest with 10,65% and Jaccard's similarity coefficient was the lowest with 15.79%. Despite those bad evaluation scores, the connectedness, which is an important feature of good clustering, was the second highest with 80,08%. When the cluster membership assignments and the co-occurrences in were reviewed, it became clear right away. The cluster had three products in it, of which two had a co-occurrence percentage of 21%. This was extremely high compared with the other co-occurrence percentages, which had values between 0.14% and 5.66%. When the clustering result is taken as a whole, the co-occurrences with members was almost 75% and the co-occurrences of products with non-cluster members was 25,19%. Summarized, measured with the mean absolute error and Jaccard's similarity the best to worst clustering results were from cluster: 5, 7, 4, 6, 2. When the best to worst clustering result is measured with connectedness/separability the ordering was cluster: 5, 2, 7, 4, 6.

K-means	k	N	Ɛ	|cells|	sum E	MAE	Jaccard	M	X	Connectedness	Separability
Cluster 2	3	6	32	38	63.87%	10.65%	15.79%	7.72%	1.92%	80.08%	19.92%
Cluster 4	4	12	3	15	7.31%	0.61%	80.00%	8.20%	3.25%	71.62%	28.38%
Cluster 5	4	12	0	12	0.00%	0.00%	100.00%	9.73%	1.96%	83.23%	16.77%
Cluster 6	4	12	17	29	27.14%	2.26%	41.38%	6.80%	3.43%	66.47%	33.53%
Cluster 7	4	12	1	13	1.93%	0.16%	92.31%	9.34%	3.51%	72.68%	27.32%
Total	15	42	53	95	100.25%	2.39%	44.21%	41.79%	14.07%	74.81%	25.19%

Table 15: The results of the comparison between *k*-means with a *k* of 7 and the real co-occurrences in the dataset.

Gaussian Mixture Model

Half of the existing product relationships could be compared with those obtained that were with the GMM method with a *k* of 5 (118 of the 210). The results are listed in Table 16. Cluster 1 had with 17.81% the highest average error percentage. Cluster 1 had three products in it, resulting in 6 product relationships. The products of the cluster occurred 45 times more often with products of other clusters.

This means that the cluster had a similarity of 11.76% with the dataset if it is measured with the Jaccard's similarity coefficient. This is very low. The result of cluster 2 shows poor performance if the errors are compared with the real co-occurrences in a binary way. Around 55% of the co-occurrences with members is similar with the highest co-occurrences in the dataset, the other half were with members of another cluster. If the deviation in percentages is reviewed, cluster 2 has an average deviation of 1.80% per product relationship. This means that the errors that were measured with the Jaccard's similarity coefficient, where not very big in terms of co-occurrence percentages. Cluster 3, which captured two product relationships, had the highest similarity and the lowest mean absolute error compared with the other clusters. Cluster 5 had an average co-occurrence percentage error of 3.07%. The Jaccard's similarity of this cluster with the dataset was the second lowest with 45%, compared with the other clusters. The total mean absolute error of the product relationships obtained with the GMM method was 3.56% and the Jaccard similarity of the whole result with the dataset was 40.83%.

Cluster 5, which captured most product relationships compared with the other clusters, had the lowest connectedness value of 48.41% and highest separability of 51.59%. Cluster 1 had a connectedness of 53.71% and a separability of 46.29%. Those results almost look like random clustering results and indicate bad clustering. Cluster 4 is not analysed because only one of the selected products was in it. Cluster 3 had the highest connectedness with a value of 71.91%, followed by cluster 2 with a score of 65.73%. As the clustering result is taken as a whole, the connectedness was 62.39%. Those values indicate bad clustering. Summarized, from best to worst clustering result measured with Jaccard's similarity and connectedness/separability were cluster: 3, 2, 1, 5. Measured with the mean absolute error the best to worst clustering results were from cluster: 3, 2, 5, 1.

GMM	k	N	Ɛ	|cells|	sum E	MAE	Jaccard	M	X	Connectedness	Separability
Cluster 1	3	6	45	51	106.83%	17.81%	11.76%	3.98%	3.43%	53.71%	46.29%
Cluster 2	5	20	16	36	35.99%	1.80%	55.56%	6.58%	3.43%	65.73%	34.27%
Cluster 3	2	2	1	3	0.66%	0.33%	66.67%	10.01%	3.91%	71.91%	28.09%
Cluster 5	10	90	109	199	276.68%	3.07%	45.23%	3.20%	3.41%	48.41%	51.59%
Total	20	118	171	289	420.16%	3.56%	40.83%	5.94%	3.55%	62.64%	37.36%

Table 16: The results of the comparison between the GMM with a *k* of 5 and the real co-occurrences in the dataset.

The Louvain method

As it can be seen in Table 17, around 40% of the existing product relationships in the data could be compared with the results of the Louvain method (86 of 210 product relationships). The selected products fell into 5 communities, but community 61 could not be evaluated during the first comparison because only one of the selected products was member of this community. Community 240 was the only community in which the members not always occurred most frequently with other members. As it can be seen in the concerning tables in the appendix, product *d*, which was the only selected product in community 61, was the only product that occurred four times more often with members of community 240. This led to a Jaccard similarity of 91.30%. The average sum of the error percentages was 0.05%

per community assignment was 0.05% in this community. The other three communities that could be evaluated, consisted only of members that occurred most frequently together in the dataset. As it is mentioned in the previous paragraphs, it is not possible to evaluate the community membership assignments of the communities that are grouped ‘perfectly’. It is possible to evaluate in a more general way if those communities are missing members or not. If the connectedness and separability values are close to each other, the indication is that some products should be member of the community while they are not.

The connectedness and separability scores lay far from each other for each community. The highest separability score is 27.77% for community 240, which was also the only community with some deviation in it. The average connectedness of the result obtained with the Louvain method, was 84,49%. This indicates that the grouping went well. There are no apparent indications that some products should be member of another community. Summarized, community 240 was the only community in which the products were not grouped perfectly according to the first comparison method. The connectedness scores were from high to low from community: 209, 220, 88, 240.

Louvain	k	N	Ɛ	|cells|	sum E	MAE	Jaccard	M	X	Connectedness	Separability
Com. 88	6	30	0	30	0	0	100%	7.39%	2.78%	72.66%	27.34%
Com. 209	2	2	0	2	0	0	100%	21.26%	0.90%	95.94%	4.06%
Com. 220	4	12	0	12	0	0	100%	9.73%	1.96%	83.23%	16.77%
Com. 240	7	42	4	46	2.23%	0.05%	91.30%	6.92%	2.66%	72.23%	27.77%
Total 1st	5 (19)	86	4	90	2.23%	0.03%	95.56%	11.33%	2.08%	84.49%	15.51%

Table 17: The results of the comparison between the results obtained with the Louvain method and the real co-occurrences in the dataset. The selected products were assigned to 5 different communities.

Clique perlocation method

It was possible to compare 60 of the 210 existing product relationships using the CPM. The results can be seen in Table 18. The method identified two overlapping communities, which both captured 30 product relationships. The co-occurrence percentages of the six members of community P were all with other members. For this reason this community did not deviate from the real data using the comparison measures of the first comparison method. Community B, had almost the same number of errors (28) than product relationships that were compared (30). The Jaccard similarity was 51.72%, which is very low. The mean absolute error of this cluster was 2.58%. Because there were only two different communities in this result, the Jaccard similarity of the result of this method taken as a whole, was 68.18%, the average error per product relationship of the whole result was 1.29%.

Since less than 30% of the product relationships could be compared, the connectedness and separability can be used to get extra information about the community membership assignments. After all, those measures took each possible product relationship into consideration. The separability scores were high for both communities. Community P which was grouped perfectly with the use of the first comparison method, had separability score of 27.34% and a connectedness of 65.84%. Community B, which had a very low Jaccard similarity score, had a separability score of 33.45%. Taking the communities as a whole, the average connectedness was 63.52% and the separability 36.48%. Those values indicate bad grouping. Summarized, community P scores best with all measures used during the first comparison.

CPM	k	N	Ɛ	|cells|	sum E	MAE	Jaccard	M	X	Connectedness	Separability
Com. P	6	30	0	30	0.00%	0.00%	100%	7.39%	2.78%	65.84%	27.34%
Com. B	6	30	28	58	77.36%	2.58%	51.72%	5.53%	2.78%	60.85%	33.45%
CPM	2 (12)	60	28	88	77.36%	1.29%	68.18%	6.72%	3.86%	63.52%	36.48%

Table 18: The comparison results of the product relationships obtained with the Clique Perlocation Method and the real co-occurrences in the data.

Results of the first comparison

Using the first evaluation method, the most similar and least deviating grouping results, measured with the MAE and Jaccard's similarity, were obtained with the Louvain method. The second best with the CPM, then k-means, followed by the GMM. However, these results were obtained from an arbitrary number of product relationships. In terms of completeness, the GMM scored the highest with 188 product relationships that were taken into account. The Louvain method scored the second highest with the comparison of 86 co-occurrences, then CPM with 60 product relationships that were compared and k-means scored the lowest with the comparison of 42 product relationships.

When the results of the first evaluation and the number of product relationships that are considered during the comparison with the co-occurrences in the dataset are combined, it is possible to obtain relative deviation and similarity scores. This makes an cross-method comparison of the results possible. The results of the comparison of the (results of the) methods can be seen in Table 19. The completeness is the number of product relationships that are taken into account with the method, divided by the number of product relationships in the data. This is *N* divided by 210. The relative deviation is the total mean absolute error times 1 minus the completeness score (MAE* (1-completeness)). The relative similarity is the total Jaccard's similarity times the completeness. The values in Table 12 give a more reliable view of the performance of the methods in terms of similarity with the real data, because they take the number of product relationships that are taken into account with each method, into consideration.

The Louvain method had relatively the least deviation with the dataset, in terms of product relationships between members in percentages and co-occurrence percentages with non-members. The relative similarity was also the highest compared with the other methods. This method performed best.

The method with the second lowest relative deviation score was the CPM, followed by GMM and k-means. The method with the second highest relative similarity score was the GMM, followed by the CPM and k-means. This means that the CPM method had the third most errors in terms of non-members that occurred more frequently with members of a community. But those errors were measured in percentages, smaller than the error percentages of the GMM method, which had less errors. Thus the

GMM method generated more product relationships and made relatively less mistakes than the CPM method, but the mistakes it made in percentages were bigger than the CPM method. The k-means method performed the worst when the relative deviation and relative similarity scores are compared.

When the grouping results were evaluated in terms of connectedness and separability, all product relationships were taken into account. The Louvain method scored best, followed by k-means, CPM and GMM. That k-means scored second best while it scored the worst with the other comparison measures, was the only unexpected result of this research. However, when the co-occurrences and membership assignments of cluster two were reviewed, the mystery was solved right away. Three products were clustered together, of which two had a co-occurrence percentage of 21.26%. In the other clusters there were also a few members with very high co-occurrence percentages compared with the other co-occurrence percentages. With such a division/distance between the values, the amount of errors does not really influence the connectedness.

Method	Completeness	Relative deviation	Relative similarity
K-means	20.00%	1.91%	8.84%
GMM	56.19%	1.56%	22.94%
Louvain	40.95%	0.02%	39.13%
CPM	28.57%	0.92%	19.48%

Table 19: The similarity of the results with the real co-occurrences in the data per method, measured with the first evaluation method that compared product relationships of group members

Results of the second evaluation method that compared all possible product relationships

During the second comparison the results of the GMM and the Louvain method were transformed into co-occurrence probabilities, that were saved in a co-occurrence matrix. The difference between each cell of the co-occurrence matrix of the dataset and the corresponding cell of the co-occurrence matrix obtained with the experiment was saved in a new table.

The results of the evaluation method are listed in Table 20. The number of product combinations (*N*) is 420 instead of 210, because the co-occurrence matrices were made symmetrical for easier computation. This does not influence the result because each cell of the table is taken into account when the average of the sum of the error percentages (MAE) was obtained. The Louvain method had a lower mean absolute error (5.89%) than the GMM (19.97%). This means that the Louvain method identified product relationships that were most similar with the co-occurrences in the dataset.

	N	sum ε	MAE
GMM	420	8389.00%	19.97%
Louvain	420	2474.90%	5.89%

Table 20: The results of the GMM and the Louvain method, measured with the second evaluation method that compared all possible product relationships with the data.

Conclusion

Retailers have to have the right stock at the right place at the right time. Inventory carrying costs have to be minimized, just as out of stocks or excess of products at a location have to be prevented. The logistic costs have to be minimized and marketing actions have to be as profitable as possible. Not only assortment planning, but also shelf space management must be done with an increase in revenue as goal. Fulfilling those kind of requirements, can only be done with a certain insight in the purchase relationships between products in the assortment. Those insights can be obtained with different datamining methods (Zentes, Morschett & Schramm-Klein, 2011). The term that is used if a retailer makes decisions concerning those requirements that have to be fulfilled, (partly) based on valuable information obtained with datamining techniques, is data driven decision making. An increasing number of companies makes their decisions partly in a data driven manner and research shows that data driven decision making leads to more economic value (Brynjolfsson, Hitt & Kim, 2011; Shmueli, Patel & Bruce, 2011). If it is done right of course. Not only if the right questions are asked, but also if the right method is used to gather the information that is needed. The goal of this study was to find the best method that can be used to identify product relationships based on purchase patterns apparent in point-of-sale (POS) data. The best method was defined as the method that generates product relationships that are most similar with the real co-occurrences of products in the dataset. Eleven methods that can be used for this purpose were described, together with a discussion of their strengths and weaknesses.

The most popular techniques to identify purchase patterns with, are association rule mining (ARM) methods (Hipp, Güntzer & Nakhaeizadeh, 2000). The output of those methods is generated without probabilistic inferences and therefore close to the real co-occurrences in the dataset. If the relationships of a few specific items with others, or the most apparent relationships in a small timeframe around an event are wished to be found, ARM methods can be very useful (Leskovec, Rajaraman & Ullman, 2014; Shmueli, Patel & Bruce, 2010). The ARM methods that were discussed in this paper are the apriori and store-chain association rules. If ARM methods are going to be used by a retailer with multiple stores, the last mentioned method is advised to use, since it also takes the location and time in which the rules hold into account. Different opening dates of stores, or different products on the shelves do not blur the results with this method (Chen, Tang, Shen & Hu, 2005). A downside of ARM methods is that they generate many inexplicable rules. This is also experienced during this research. The apriori method is used to generate 50 association rules with the highest support values. Only 21 unique products were present in those 50 rules. Almost each second rule consisted of the same products as the rule before, only in the opposite order. Different measures like confidence, lift and interestingness are attempts to reduce the amount of inexplicable rules (Leskovec, Rajaraman & Ullman, 2014; Tan & Kumar, 2005). Despite those attempts, if the goal is to get an overall insight in the relationships between the products, or of every product, those methods are not effective.

The first discussed clustering method was k-means, which is also a state-of-the-art frequent item set algorithm (Videla-Cavieres & Ríos, 2014). Followed by the Gaussian Mixture Model and the Dirichlet Process Gaussian Mixture model. The other methods were network analysis methods. One of those community detection methods was the Louvain method, which is based on modularity. The discussed community detection methods that assume sparsely connected overlapping communities were the clique perlocation method (CPM), mixed-membership stochastic block model and link clustering. Finally, the combined use of the Community-Affiliation Graph Model (AGM) and the Cluster Affiliation Model for Big Networks (BigCLAM) was discussed, which both assume densely connected overlapping communities. On theoretical grounds, a few expectations about the performance of the methods were made. However, because the results of some methods could not be compared since the resources that were needed to implement them were not available, not every expectation could be tested.

The k-means, GMM, Louvain method and CPM were implemented and evaluated in terms of similarity with the real co-occurrences in the data. The input data consisted of POS data of 303 different products. A co-occurrence matrix that was used during the evaluation, contained each possible product relationship of two of the 21 selected products in percentages. As it can be implied, only the product relationships of this product selection were filtered from the results of the methods. Since no cross-category method comparison measure could be found, two different comparison methods were created, both based on the probability theory. During the first evaluation method that compared product relationships of group members, the results of all implemented methods were compared with the real co-occurrences in the dataset. During the second evaluation method that compared all possible product relationships with the data, the results the GMM and the Louvain method were compared with the co-occurrences.

The first expectation was that the community detection methods would generate product relationships that were most similar with the real co-occurrences in the data. As it will be described in this section, the best performing method was indeed a community detection method, but the second best method was not. More specifically, based on the assumption of dense connections in the overlapping parts of communities and other structural properties of the methods (Yang & Leskovec, 2014; Yang & Leskovec, 2015), it was expected that the AMG and BigCLAM would give more accurate results than the other community detection methods. Unfortunately those methods could not be implemented because of the practical reason that was just mentioned.

The second expectation was that the Louvain method would generate more similar results with the real co-occurrences than the community detection methods that assume sparsely connected community overlaps. This expectation was based on the knowledge that individual nodes move only when the global modularity score increases with the Louvain method. The probability of the generation of communities of nodes that are sparsely connected is very low with this procedure (Blondel, et al., 2008). The result of both evaluation methods was that the Louvain method generated product relationships that were most similar with the real co-occurrences in the dataset, compared with the other methods. The results that were generated with the hard community assignments of the first evaluation method that compared product relationships of group members, covered 40.95% of the existing product relationships in the data. This completeness score was the second highest of all methods. One of the five communities had four deviating cells in it. This was each time product *d*, which was the only member of a community that was not compared, with the reason that *d* was the only member of that community. The other communities were grouped perfectly. The relative deviation percentage was with a value of 0.02% the lowest with the Louvain method and relative similarity was the highest with 39.14%, compared with the other methods. The method also had with 84.49% the highest connectedness score and the lowest similarity score. The mean absolute error (MAE) of the product relationships generated with the link weights during the second evaluation method that compared all possible product relationships with the data, was with the value of 5.89% lower than the GMM. This means that the second expectation did hold true. With an evaluation of the group assignments of the products by interpretation, the Louvain method also performed best. The grouping results were compared with a division of the products based on semantic relatedness and co-occurrence in usage situations in real world situations. Product *d* which was the only selected product that was assigned to a community that did not contain one of the other selected products, was dental care. Dental care ended with the grouping of the products based on similarity in real-world situations also alone in a group. Summarized, the Louvain method identified groups of products that were most similar with the real co-occurrences in the dataset and also most similar with the grouping based on semantic relatedness and similar usage situations in real world situations. Together with the GMM it gave the most complete overview of all existing product relationships in the data. However, the overview of the product relationships generated with the Louvain method were easier to interpret because of the different visualisations that contained information about the size of and connections between the different groups, between the products and the strength of them. For those reasons there can be concluded that this method can best be used to identify product relationships in the assortment.

The results obtained with the DPGMM method were expected to be the most accurate of the clustering methods, because the data is clustered as Gaussians that can have different structures and because the best *k* value does not have to be given beforehand (Görür & Rasmussen, 2010; Pedregosa et al., 2011). The results of this method could not be compared because only one cluster was generated during the implementation of the DPGMM. The GMM was expected to give similar results as the DPGMM method when the same *k* value is used, and worse results in general because of the difficulty to find the best *k* value (Reynolds). With the APL software of SAP different *k* values could be compared during the same analysis, what made it easier to find the best *k*. The results obtained with the first evaluation method that compared product relationships of group members with the co-occurrences in the data, were second best with the GMM method. The completeness score of the GMM was with a value of 56.19% the highest compared with all other methods. One of the five clusters had a high MAE and a low similarity and the other clusters did not deviate much and were similar with the dataset. With the relative similarity, the completeness score is taken into account, this was the second highest with the GMM with a value of 22.94%. However, the relative deviation was also the second highest with 1.56%. The relative deviation score was the second lowest for the CPM method (0.92%) and the relative similarity of the product relationships generated with the CPM and the real co-occurrences was also the second lowest with a value of 19.48%. This means that the GMM made less but bigger errors than the CPM, which made more but smaller errors during product relationship identification. Thus the CPM and GMM scored almost similar using the evaluation measures of the first evaluation method.

In order to make a decision about which method performed better, other information is also considered. The CPM generated only two groups, one community with perfect member assignments and the other one looked like a random community. Next to that, only a different and smaller subset of the data is used with the CPM. As it is mentioned before, the completeness of the results of the GMM was with the first evaluation method higher in comparison with the CPM, which had a completeness score of 28.57%. The connectedness score was a little higher with the results of the CPM (63.52%) than with the results of the GMM (62.64%). If there is a quiz with ten questions for instance, the chance on the least mistakes is higher when only two questions are answered than when all ten questions are answered. This is the same with the number of product relationships that are given per method. Next to that reason, the objective was to find the method with the highest similarity with the real relationships in the dataset. The GMM considered more product relationships and those were also more precise since more different groups were made. When the results were evaluated by interpretation, both methods made groups that did not make sense. The CPM identified one group that consisted of products that would also be grouped together when a human would do that by hand. The other group consisted of products from all remaining groups, which made it semantically also look totally random. The division of the products made with the GMM did make more sense than the results of the CPM. Because of the soft cluster assignments with the GMM, it was also possible to compare the results with the second evaluation method, in which all possible product relationships are compared with the real co-occurrences. For this reason, it is better to use the GMM to identify product relationships in the assortment of a retailer than the CPM.

Despite the popularity of the k-means method, given its properties and weaknesses, the expectation was that the obtained results would be least similar with the co-occurrences in the data (Jain, 2010; Videla-Cavieres & Ríos, 2014). As it can be implied after reading the previous stated conclusions, the k-means performed as it was expected, the worst using the evaluation method that compared product relationships of group members with the co-occurrences in the data, compared with the other methods. Only one fifth of the existing product relationships could be analysed with the results obtained with the k-means method with a *k* of 7. This means that a lot of information is lost. Two of the seven clusters had one of the selected products in it and could not be compared for this reason. The relative deviation was the highest (1.91%) and the relative similarity with the co-occurrences in the data was the lowest with a value of 8.84%. Despite the bad results obtained with the first evaluation method, they should be considered in their context. The connectivity and separability scores of the groups generated with the method were the second best compared with those of the other methods. That the results of the k-means method were the most deviating and the least similar with the data compared with the other methods, while it had the second highest the connectedness score, can partly be explained by a few very high co-occurrence percentages of members. When the distance between the co-occurrence values in the same cluster is big, the amount of errors does not really influence the connectedness. However, when the results were evaluated by interpretation, the results of the k-means method were more similar with a logical division of the products made with real-world knowledge, compared with those of the GMM and CPM. The completeness of the method influences the results of the first evaluation method and the k-means identified the least product relationships of all methods. When the completeness of the method is not taken into consideration, k-means scores better than the GMM.

It can be concluded that, compared with k-means, GMM and CPM, the Louvain method generates product relationships that are most similar with the co-occurrences of the products in the dataset. The main output of the method is easy to interpret and contains a lot of information. The relationship strengths between products and groups of products are included. Those product relationships are also most similar with the division of products based on semantic relatedness and similar usage situations in the real world, which makes it an ecological valid method. Together with the GMM the Louvain method also scored the highest in terms of completeness. Additive output contains relationships strengths of all product combinations in the data, which makes it possible to get a complete overview of all relationships in the assortment. The structure of the data and the assumptions the methods have about the data, that determine the calculations that are made, influence the results most. For this reason it can be assumed that the Louvain method will give better insights in the existing product relationships in the assortment than the other methods, if many different products are sold in a retail store and only a fraction of the products is bought at the same time by customers. It can also be assumed that this is the case for discount and competitive pricing retailers.

Discussion

Today's diversity in product assortments of retailers has the consequence that the traditional methods are not useful anymore for the identification of existing relationships between products in the assortment. Association rule mining techniques, which are traditionally used for this purpose, cannot give a general overview of valuable product relationships, because of the large amount of (inexplicable) rules they generate (Webb, 2006). They can be used to look at the relationships of a subset of products, or to compare the most obvious relationships among different time periods, but not to get insight in the real dependency structures of products that are sold at a retail company. For this reason other methods that can be used to get a deeper insight in existing relationships that are based on purchase behaviour are compared in this research.

K-means clustering is the most widely known clustering method, also for the identification of products that are frequently bought together (Videla-Cavieres & Ríos, 2014). The method is simple, robust and scalable, but also has multiple drawbacks. As it is described in the conclusion, the k-means method performed the worst during the comparison of the product relationships that were given as result and the real co-occurrences in the data. This can partly be justified with the notion that the results were obtained from a larger dataset with 303 different products in it, while the results of 21 products were compared with the real co-occurrences in the data. The transactions that contained the 21 selected items were extracted from the clustering result and this selection of the data has been regarded as 100% during the calculations. The clusters consisted originally of more products, that also influenced the obtained results. On the other hand, the results of the other methods were evaluated in the exact same way, and the result of the Louvain method gave nearly a perfect representation of the real product relationships. This means that the structure of the results, that contains less information about the relationships does not have to be the reason of bad performance per se. However, if the completeness of the identified product relationships was not taken into account, the method scored second worst. The amount of product relationships that were not taken into consideration influenced the results and the k-means generated the least product relationships. Another explanation can be that the number of groups has to be known beforehand with this method, what was not the case. This could have let to the wrong *k* value. Another weakness of the method that possibly can explain the bad performance is that the data gets forcefully partitioned in groups with the same convex structure, with the least distance between the groups and most distance between the groups (EMC Education Services, 2015; Jain, 2010). Products are often member of multiple groups. If a certain degree of specificity of the relationships is sought with a k-means, a lot of information is lost because many groups have to be generated with the consequence that the relationships between products of different groups are not considered. However, if only a few groups of products are generated, more product relationships are obtained, with the loss of information because only general product relationships can be extracted from clusters that contain many products. In sum, if this method is chosen a lot of information about the relationships between products is lost.

This is in line with the results of the experiment, only 20% of the existing product relationships could be extracted with this method. The cluster assignments that were obtained with this method were next to those of the Louvain method, second most similar with a division made by hand, based on semantic and usage situation similarity. Thus the results were overall the least similar with the co-occurrence matrix of the dataset, the second least similar when only the product relationships generated with the method were compared, and the second most similar when evaluated by interpretation.

The Gaussian mixture model (GMM) is another, more sophisticated clustering method. It has a *k*, variance and pi parameter, with the consequence that the clusters are Gaussians instead of the similar convex structures around the mean like with k-means. The method also has an additional step in which each data point gets assigned to each cluster with a soft probability, based on maximum likelihood estimation (Reynolds). This has the consequence that the information about the (less obvious) relationships with products from other clusters is not totally lost like it is with k-means. When hard cluster assignments are desired, the cluster with the highest probability value should be chosen. A weakness of the method is that *k* has to be specified beforehand. However, the SAP APL software has the possibility to specify a minimum and a maximum *k* and clusters the data for each *k* value twice, so it bypasses this disadvantage of the method. With the possibility of clusters of different shapes and the soft probability cluster assignments, the expectation was that this method would give a result that was a better reflection of the product relationships in the data than the result obtained with k-means. Like expected, based on the properties of the method, the GMM performed second best during the comparison based on product co-occurrence similarity.

The idea was to try the DPGMM algorithm before the GMM. The DPGMM is almost the same as the GMM. The output is also comparable, but the main advantage is that the *k* value does not have to be specified beforehand. The Dirichlet Process is used to identify the best *k*. The only parameters that have to be given beforehand are a maximum *k* value that does not really influences the result and an alpha parameter, which influences the concentration of the clustering result. Changes in the alpha parameter can lead to more and detailed, or less and more general clusters. A disadvantage of the method is that it is not an empirical proven model selection procedure, thus there are no guarantees on the result (Görür & Rasmussen, 2010; Pedregosa et al., 2011). This is experienced during the experiments, because with different maximum *k* and alpha settings, only one cluster was generated. This is probably due to the highly dimensional (303 features) sparse and solely binary feature vectors. Finding relationships between products is finding relationships in categorical data. In order to use clustering algorithms the data has to be transformed. The product names have to be made numerical because clustering methods require numerical input. With the transformation of the input data, each transaction has to have the exact same structure. This means that each product gets an own column, with a numerical value representing if it was present in the transaction or not. This data transformation makes it possible to cluster categorical data. However, despite this transformation, many clustering methods do not perform (well) when big amounts of categorical data are being clustered (EMC Education Services, 2015). This makes some of the clustering methods less applicable when the wish is to cluster the whole assortment into groups based on co-occurrences.

Community detection methods have the ability to identify the structure of complex networks. This makes them theoretically suitable methods that can be used to identify purchase patterns in big transactional datasets. An advantage of community detection methods, is that the structure of the output can easily be interpreted, because it is visualized in a graph. The graph contains sometimes not only information about the existence of relationships between the products, but also about the strength of these relationships. Another advantage is that the products can be part of multiple groups, with the latest community detection methods. Although community detection methods seem to be convenient methods to use for this purpose, only two studies could be found that used them with to identify relationships between products based on purchase behaviour (Raeder & Chawla, 2011; Videla-Cavieres & Ríos, 2014). To extend the existing literature in this field, six community detection methods were discussed and two methods were implemented and evaluated during this research.

The Louvain method is a community detection algorithm that is based on modularity maximization and can be performed in two steps. At initialization each node is regarded as a community, which is merged with another community or stays in the same community based on the modularity increase for the whole network. When the modularity does not increase anymore, the second step begins in which each community is regarded as a node. A similar process takes place in which internal edges are regarded as self-loops and external as weighed edges to other communities. When the modularity does not increase anymore the first step gets repeated. The obtained result is a hierarchical community structure. The biggest strength of this method is that it is based on modularity, but does not suffer from the resolution limit problem of modularity, since individual nodes are only moved if the global modularity value increases. Another strength is that the method is faster than other modularity methods (Blondel, Guillaume, Lambiotte & Lefebvre, 2008). An weakness of the method is that it does identify communities at an different level, but not overlapping communities. This weakness can be overcome when the weights of each product relationship in the data is analysed. Based on the properties of the method, the expectation was that it would perform second best if the AMG and BigCLAM methods were taken into account. Since the AMG and BigCLAM were not compared during the experiments of this study, the expectation was that the method would generate results that were most similar with the dataset, compared with the other methods. This was like expected the case, with both comparison methods and also with the evaluation by interpretation. The products were grouped the same way as a human would group them based on semantic relatedness and similar usage situations.

The CMP was the first community detection method that identified overlapping communities. The method regards each link between nodes as an edge. Fully connected nodes with *k* edges is a *k*-clique, and communities are the union of the *k*-cliques that are connected by adjacent cliques, which contain *k*-1 connections (Palla et al., 2005). A weakness of the method is that it regards each link as edge and does not consider the density of the connections. The consequence can be, that only one community can be found in highly connected networks, what makes the method less applicable when a market basket analysis is performed. This was also the case during the experiments, when the whole dataset was used only one community was generated with each product in it. In order to try to generate more than one community with this method, different selections of the product combinations are used, based on link thresholds. The distribution of product combination frequencies of the different product combinations had a long tail, thus the set of rows that was filtered from the dataset each time was relatively big. It could be expected that the number of communities would increase with the removal of this data, but this was still not the case wen only one third of the data was used. The two communities that were identified, both had a *k* of 6 and two products were member of both communities. One of those communities was grouped perfectly and the other very badly. A possible explanation can be that the method assumes less connections in overlapping parts of communities, what is not the case with POS data of a retailer because of the large number of different products that can be bought and the low average number of products purchased during the same transaction.

The mixed membership stochastic block model (Airoldi, et al., 2009) and link clustering (Ahn, et al., 2010) are other community detection methods, that have the assumption that the overlapping parts of communities are more sparsely connected compared with the non-overlapping parts. The results of these methods were not compared with the co-occurrences in the dataset, because the resources needed to implement them were not available. However, since background information about the products and transactions of the data is known, the implementation of those methods would not add extra value since the data is well connected, which causes a bad fit.

The AMG and BigCLAM method are the first community detection methods that assume densely connected community overlaps (Yang & Leskovec, 2012; Yang & Leskovec, 2013). Combined they can identify (non)overlapping and nested communities that reflect product relationships in POS data more accurate and precise, because of this assumption. First AMG can be used to generate an undirected unlabelled social network graph and then the BigCLAM method can be used to model de membership strengths, given the undirected unlabelled social network graph that was generated with the AMG. According to comparative research of community detection methods, the AMG and BigCLAM are faster and can deal with ten times more data than the previously mentioned community detection methods (Yang & Leskovec, 2015).

The first evaluation method that compared product relationships of group members

During the first comparison the results obtained with the k-means, GMM, Louvain method and CPM, which had the form of groups of products were evaluated. For each group and for each product in the group the co-occurrence percentages were sorted in a descending order. If a product occurred more often with a member of another group than with at least one member of the same group, this combination was regarded as an error. As it can be implied from this description, only the co-occurrences with a higher value than the lowest co-occurrence percentage with another member are compared. Clusters with only one of the selected products in it were also not compared using the first comparison method. The deviation of the group assignments with the real co-occurrences in the dataset is measured with the mean absolute error (MAE). The error percentages are obtained by taking the difference between the co-occurrence percentage with the non-member and the lowest co-occurrence percentage with a member. The sum of all error percentages is divided by the number of product relationships *N* according to the group assignment (which is always k*(k-1)). The MAE is calculated per cluster and per method as a whole. Since there is a difference between one nearly perfect group and a very bad one, and two 'oke' groups while those can have the same MAE, the similarity of the results is also measured in a binary way. With Jaccard's similarity coefficient, the number of product relationships captured in the grouping result *N* is divided by the number of cells that are taken into account during the comparison. The number of cells that are taken into consideration is the sum of *N* and the number of errors. The most important drawback of this way of comparing the results, is that not all product relationships are taken into account, which means that a lot of information is lost. In order to get more information about the goodness of the grouping, the connectedness and separability are measured using product relationships. The average co-occurrence percentages with members and the average co-occurrence percentages with non-members are divided by the total co-occurrence percentage of the group, to obtain those values. The completeness of each method is the ratio of product relationships that are considered in the grouping result to the total number of product relationships (which is 210). In order to make a valid comparison between the different methods possible, the deviation and the similarity are transformed to relative deviation and similarity scores which take the completeness of the result into account. The relative deviation is obtained by taking the product of 1 minus the completeness score and the MAE. The relative similarity is obtained with taking the product of the completeness score and Jaccard's similarity coefficient.

The second evaluation method that compared all possible product relationships with the data

During the second comparison method all possible product relationships were taken into account. Only the results of the GMM and the Louvain method generated enough information to calculate the probability of each possible product relationship. The results obtained with the GMM consisted of soft group assignments of each product to each cluster with a certain probability. During the first evaluation method that compared product relationships of group members, the product was assigned to the cluster with the highest probability percentage. During the second comparison, each cluster assignment probability is re-calculated with taking the sum of all cluster assignments as 100%. The product relationship percentage of two products, is measured as the sum of the products, of the cluster assignment probabilities of each group. Each possible product relationship is calculated and stored in a new co-occurrence matrix. The Louvain method provided an additional file with all product pairs in the dataset and their link weight. Just like with the GMM, per product, the sum of the weights is regarded as the total weight (100%). Each possible product relationship percentage, calculated by dividing the weight by the total weight, is saved in a new co-occurrence matrix. The difference between each cell of the co-occurrence matrix of the dataset and each corresponding cell of the co-occurrence matrix of the grouping result is saved in an error percentages table. The MAE is obtained by taking the average of this table.

Drawbacks of this study

A drawback of the first evaluation method that compared product relationships of group members, is that only the co-occurrences with members and with non-members which have a higher value than the lowest co-occurrence percentage with members are compared. The reason that non-members that have a lower co-occurrence percentage than each member, are not considered as a possible deviating non-member (that should be a member), is simply because this is not possible without the use of other algorithms. The only certain information that is available are the hard group membership assignments and possibly higher co-occurrence percentages of non-members group members. Since this is the only certain information, it is the only information that was used. No spurious causations are made in this way.

Another drawback of this method is that groups that contain only one of the selected products that are compared, cannot be evaluated. It is not possible to compare product relationships of only one product. However, when this method is used without filtering a small selection of products that is compared with the real co-occurrences, clusters that have only one product in it are not likely. Thus this is only a drawback in situations in which such a small selection of the data is used that the traditional method made for market basket analysis (the apriori) can be implemented. Not when a retailer wants to have an overview in the relationships between products in a wide product assortment. A limitation that came as consequence of the first drawback, is that the relative deviation and similarity scores are inferred with the assumption that the rest of the grouping result (which consist of product relationships that cannot be measured) are the same as the measured result. It is true that the degree of completeness of the result (in terms of the number of product relationships that are taken into account) and the deviation/similarity of the results with the real data influence each other. The more product relationships that are taken into account the more precise the results are. The chance on an error is also higher when more relationships are described. However, there is no guarantee that this is a linear relationship, like as it has been treated during this comparison. This was also visible in the results of this study. The k-means method had the highest relative deviation and the lowest relative similarity score, while those scores were the second lowest when the completeness of the (number of) product relationships was not taken into account. Besides that, the results of k-means were the second highest when they were evaluated by interpretation. Despite the weaknesses of the first comparison method, it does provide a new way of evaluating methods with different output structures, what can be seen as the first step into an interesting research area. No cross-category datamining evaluation measure for unlabelled data could be found in existing literature and this method makes it possible to compare results of different methods that have varying structures, with factual data. Taking this all together, the comparison of probabilities generated by the different methods (or transforming group assignments into probabilities that are compared) with real ratios, leads to an internally valid comparison measure. However, all aspects of the results have to be taken into account concurrently when different methods are compared. The last limitation of this study that will be mentioned, is that the AMG and BigCLAM methods could not be implemented, since those methods were expected to give the best results.

Insights and suggestions for future research

This research can provide guidelines for retailers of how they can use POS data in order to extract as many accurate product relationships that exist in the assortment as possible. Different methods are compared and the one with the best performance is sought, because the traditional methods are not useful to get an full understanding of the existing relationships in an assortment that encompasses many products. ARM methods generate too many association rules to analyse and the informative value of most generated rules is very low. This leads to the suggestion for future research to improve ARM methods in such a way that less rules are generated, that contain information about more different product relationships. Another research area that should be focussed on is to improve or create similarity measures for clustering methods that can handle highly dimensional sparse binary data, what represents all products in the assortment together with information which product is bought during the same transaction.

When the structural and methodological properties of different clustering and community detection methods were considered, community detection methods that do not have the assumption of sparsely connected community overlaps, seemed best suited to deal with large amounts of product relationships in terms of speed, accuracy the ease of the interpretation of the results and their ability to identify complex relational structures in the data. The main finding of the experiments, was that the Louvain method gave insights in the product relationships using visualisations that were easy to interpret, and with product relationships presented as edge weights, that were almost similar with the real co-occurrence percentages of products in the data. The relationship strength between products that were grouped together and between different groups of products was also visualised, what made the results more precise in comparison with the other methods. Those results were as it can be expected not only most similar with the real co-occurrences of the products in the dataset, but also most similar with a handmade grouping of the products based on semantic relatedness and co-occurrences in usage situations in real life. This means that the results of this method were ecological valid. In the context of existing research, this result indicates that a shift has to take place from the attention that traditional

MBA methods get, towards methods that are able to capture the structure of product relationships, even with today's diversity in the assortments of the retail industry. Retailers that want to get insight in the relationships between products are advised to use this method, or to familiarize themselves with other community detection methods that are able to capture complex structures in networks and that do not have the assumption of sparsely connected community overlaps.

Based on theoretical grounds, the expectation was that the AGM combined with the BigCLAM algorithm would generate most accurate and precise results in terms of similarity with the co-occurrences in the dataset. Because of practical reasons those two could not be deployed. The results of the Louvain method reflected the co-occurrences in the real dataset nearly perfect. For this reason it would be interesting for further research to apply the AMG and BigCLAM on POS data of a retailer with a wide product assortment, and evaluate the results with real co-occurrence percentages in the dataset. It would also be interesting to compare the results of those two methods with the result of the Louvain method. The expectation was that the AMG and BigCLAM would perform better and give more detailed information because community overlaps are specified at the node level instead of at the community level like with the Louvain method. However, it is almost impossible to generate results that are better than those that were generated with the Louvain method during this experiment, because they were nearly perfect. When different methods are compared, it is advised to use a bigger comparison dataset in order to get insights in the generalizability of the obtained results. In this study the relationships of a subset of 7% of the products in the data is examined on similarity with the real co-occurrences. It can be interesting to find out if the method performs just as well when relationships between products that occur less frequently together in transactions are also compared with their real co-occurrences in percentages. In this this thesis we have seen that retailers that have wide product assortments and/or multiple stores, can easily get an accurate overview of the relationships between products based on purchase patterns, if the Louvain method is used. The methods that are traditionally used to get insight in the product relationships are still useful for other purposes, such as examination of the most popular product combinations bought in the week of a certain event, or to evaluate marketing actions for example. Since the difficulty to extract useful information with those techniques increases when many products are sold concurrently, the Louvain method can be a valuable complement in today's retailing industry.

Literature

Ahn, Y. Y., Bagrow, J. P., & Lehmann, S. (2010). Link communities reveal multiscale complexity in networks. *Nature*, 466(7307), 761-764.

Airoldi, E. M., Blei, D. M., Fienberg, S. E., & Xing, E. P. (2009). Mixed membership stochastic blockmodels. In *Advances in Neural Information Processing Systems* (pp. 33-40).

Agrawal, R., Imieliński, T., & Swami, A. (1993, June). Mining association rules between sets of items in large databases. In *Acm sigmod record* (Vol. 22, No. 2, pp. 207-216). ACM.

Blattberg, R. C., Eppen, G. D., & Lieberman, J. (1981). A theoretical and empirical evaluation of price deals for consumer nondurables. *The Journal of Marketing*, 116-129.

Blondel, V. D., Guillaume, J. L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, *2008*(10), P10008.

Boven, Z. (2016). Doei Jumbo: aanbieding duurder dan normale prijs. Metronieuws. [web page] Retrieved from http://www.metronieuws.nl/nieuws/binnenland/2016/01/doei-jumbo-aanbieding-duurder-dan-normale-prijs

Brijs, T., Swinnen, G., Vanhoof, K., & Wets, G. (2004). Building an association rules framework to improve product assortment decisions. *Datamining and Knowledge Discovery*, *8*(1), 7-23.

Brynjolfsson, E., Hitt, L. M., & Kim, H. H. (2011). Strength in numbers: How does data-driven decisionmaking affect firm performance?. *Available at SSRN 1819486*.

Busse, M. R., Pope, D. G., Pope, J. C., & Silva-Risso, J. (2012). *Projection bias in the car and housing markets* (No. w18212). National Bureau of Economic Research.

Chabrol, F. P., Arenz, A., Wiechert, M. T., Margrie, T. W., & DiGregorio, D. A. (2015). Synaptic diversity enables temporal coding of coincident multisensory inputs in single neurons. *Nature neuroscience*, *18*(5), 718-727.

Centraal Bureau voor de Statistiek (2014). Demografische kerncijfers per gemeente 2014. Retrieved from https://www.cbs.nl/NR/rdonlyres/68092452-2D41-416C-B5D5-C77737DBDE80/0/demografischekerncijfers2014.pdf

Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, *7*(3), 1247-1250.

Chen, Y. L., Tang, K., Shen, R. J., & Hu, Y. H. (2005). Market basket analysis in a multiple store environment. *Decision support systems*, *40*(2), 339-354.

Chen, G., Jaradat, S. A., Banerjee, N., Tanaka, T. S., Ko, M. S., & Zhang, M. Q. (2002). Evaluation and comparison of clustering algorithms in analyzing ES cell gene expression data. *Statistica Sinica*, 241-262.

Chib, S., & Greenberg, E. (1995). Understanding the metropolis-hastings algorithm. *The American Statistician*, *49*(4), 327-335.

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

EMC Education Services (2015). *Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data*. John Wiley & Sons. Fortunato, S. (2010). Community detection in graphs. *Physics reports*,*486*(3), 75-174.

Fortunato, S., & Barthelemy, M. (2007). Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, *104*(1), 36-41.

Görür, D., & Rasmussen, C. E. (2010). Dirichlet process gaussian mixture models: Choice of the base distribution. *Journal of Computer Science and Technology, 25*(4), 653-664.

Grätzer, S. (2013). Community Detection: an introduction. *Community Detection - Data Management and Data Exploration,* Rheinisch-Westfalische Technische Hochschule Aachen. [Proseminar]. Retrieved from http://dme.rwth-aachen.de/en/system/files/file_upload/course/12/elementary-data-mining-techniques/community-detection-simon.pdf

Girvan, M., & Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the national academy of sciences*,*99*(12), 7821-7826.

Hipp, J., Güntzer, U., & Nakhaeizadeh, G. (2000). Algorithms for association rule mining—a general survey and comparison. *ACM sigkdd explorations newsletter*, *2*(1), 58-64.

Huang, Z., Zeng, D. D., & Chen, H. (2007). Analyzing consumer-product graphs: Empirical findings and applications in recommender systems. *Management science*, *53*(7), 1146-1164.

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition letters, 31*(8), 651 -666.

Kim, H. K., Kim, J. K., & Chen, Q. Y. (2012). A product network analysis for extending the market basket analysis. *Expert Systems with Applications*,*39*(8), 7403-7410.

Krzanowski, W. J., & Lai, Y. T. (1988). A criterion for determining the number of groups in a data set using sum-of-squares clustering. *Biometrics,* 23-34.

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*(6755), 788-791.

J. Leskovec. Amazon product co-purchasing network, march 02 2003. http://snap.stanford.edu/data/amazon0302.html.

Leskovec, J., Adamic, L. A., & Huberman, B. A. (2007). The dynamics of viral marketing. *ACM Transactions on the Web (TWEB), 1*(1), 5.

Leskovec, J. (2012). Affiliation network models for densely overlapping communities in networks: workshop on algorithms for modern massive data sets [video file]. Retrieved from https://www.youtube.com/watch?v=htWQWN1xAZQ

Leskovec, J. (2013). Stanford network analysis platform. *Online: http://snap. stanford. edu/snap/index. html, f evereiro*.

Leskovec, J., Lang, K. J., & Mahoney, M. W. (2010). Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th international conference on World wide web* (pp. 631-640). ACM New York, NY, USA, doi: 10.1145/1772690.1772755

Leskovec, J., Rajaraman, A., & Ullman, J. D. (2014). *Mining of massive datasets*. Cambridge University Press.

Linoff, G. S., & Berry, M. J. (2011). *Datamining techniques: for marketing, sales, and customer relationship management*. John Wiley & Sons.

Manchanda, P., Ansari, A., & Gupta, S. (1999). The “shopping basket”: A model for multicategory purchase incidence decisions. *Marketing Science,18*(2), 95-114.

Mulhern, F. J., & Leone, R. P. (1991). Implicit price bundling of retail products: A multiproduct approach to maximizing store profitability. *The Journal of Marketing*, 63-76.

Ng, A. (2015). IV. Linear Regression with Multiple Variables (Week 2). Coursera Machine Learning course Retrieved from https://class.coursera.org/ml-005/lecture

Niemi, J (2013). Metropolis-Hastings algorithm [video file]. Retrieved from https://www.youtube.com/watch?v=VGRVRjr0vyw

Palla, G., Derényi, I., Farkas, I., & Vicsek, T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, *435*(7043), 814-818.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, *12*, 2825-2830.

Radar (2016). Btw-actie Media Markt: sigaar uit eigen doos? [web page] Retrieved from http://radar.avrotros.nl/nieuws/detail/btw-actie-media-markt-sigaar-uit-eigen-doos/

Raeder, T., & Chawla, N. V. (2011). Market basket analysis with networks. *Social network analysis and mining*, *1*(2), 97-113.

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

Reynolds, D. Gaussian Mixture Models* Retrieved from http://www.ee.iisc.ernet.in/new/people/faculty/prasantg/downloads/GMM_Tutorial_Reynolds.pdf

Shmueli, G., Patel, N. R., & Bruce, P. C. (2011). *Datamining for business intelligence: Concepts, techniques, and applications in Microsoft Office Excel with XLMiner*. John Wiley and Sons.

Silverstein, C., Brin, S., & Motwani, R. (1998). Beyond market baskets: generalizing association rules to dependence rules. *Datamining and knowledge discovery*, *2*(1), 39-68.

Simonin, B. L., & Ruth, J. A. (1995). Bundling as a strategy for new product introduction: Effects on consumers' reservation prices for the bundle, the new product, and its tie-in. *Journal of Business Research*, *33*(3), 219-230.

Svetina, M., & Zupančič, J. (2005). How to increase sales in retail with market basket analysis. *Systems Integration*, 418-428.

Tan, P. N., Steinbach, M., & Kumar, V. (2005). Chapter 6. Association Analysis: Basic Concepts and Algorithms. *Introduction to Datamining.* Addison-Wesley, Boston ISBN, 321321367.

Teh, Y. W. (2011). Dirichlet process. In *Encyclopedia of machine learning*(pp. 280-287). Springer US.

Tsvetkov, D., Hristov, L., & Angelova-Slavova, R. (2013). On the convergence of the Metropolis-Hastings Markov chains. *arXiv preprint arXiv:1302.0654*.

Videla-Cavieres, I. F., & Ríos, S. A. (2014). Extending market basket analysis with graph mining techniques: A real case. *Expert Systems with Applications*, *41*(4), 1928-1936.

Walters, R. G. (1991). Assessing the impact of retail price promotions on product substitution, complementary purchase, and interstore sales displacement. *The Journal of Marketing*, 17-28.

Webb, G. I. (2006, August). Discovering significant rules. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and datamining* (pp. 434-443). ACM.

Yang, J., & Leskovec, J. (2012a). Structure and overlaps of communities in networks. *arXiv preprint arXiv:1205.6228*.

Yang, J., & Leskovec, J. (2012b, December). Community-affiliation graph model for overlapping network community detection. In *Datamining (ICDM), 2012 IEEE 12th International Conference on* (pp. 1170-1175). IEEE.

Yang, J., & Leskovec, J. (2013, February). Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Proceedings of the sixth ACM international conference on Web search and datamining* (pp. 587-596). ACM New York, NY, USA, doi: 10.1145/2433396.2433471

Yang, J., & Leskovec, J. (2014). Overlapping communities explain core–periphery organization of networks. *Proceedings of the IEEE*, *102*(12), 1892-1902.

Yang, J., & Leskovec, J. (2015). Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems,* 42(1), 181-213.

Yu, J. (2012). Machine tool condition monitoring based on an adaptive Gaussian mixture model. *Journal of Manufacturing Science and Engineering,134*(3), 031004.

Zentes, J., Morschett, D., & Schramm-Klein, H. (2007). *Strategic retail management*. Betriebswirtschaftlicher Verlag Dr. Th. Gabler GWV Fachverlage GmbH, Wiesbaden (GWV).

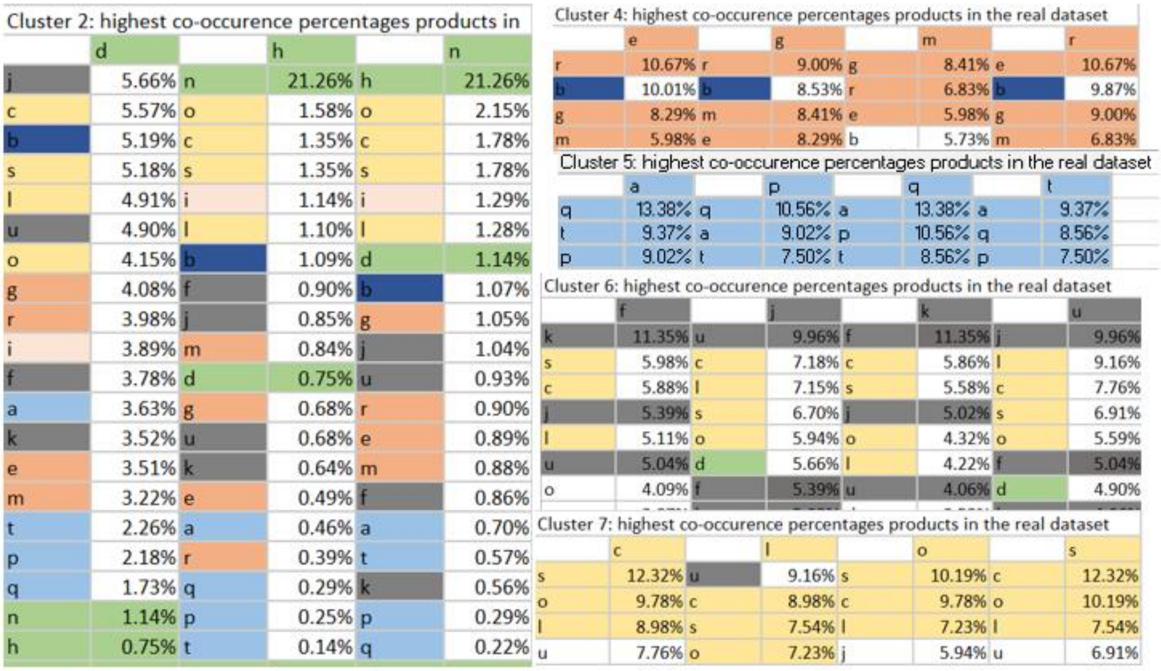
Zheng, Z., Kohavi, R., & Mason, L. (2001, August). Real world performance of association rule algorithms. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and datamining* (pp. 401-406). ACM.

Žliobaitė, I., Bakker, J., Pechenizkiy, M. (2009). Towards Context Aware Food Sales Prediction. *ICDMW*, 2009, 2013 IEEE 13th International Conference on Data Mining Workshops, 2013 IEEE 13th International Conference on Data Mining Workshops 2009, pp. 94-99, doi:10.1109/ICDMW.2009.60

Appendix and supplementary material

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u
a		4.24%	3.49%	3.63%	3.61%	3.08%	3.44%	0.46%	2.82%	3.16%	2.51%	2.32%	2.50%	0.70%	2.94%	9.02%	13.38%	4.29%	2.88%	9.37%	2.57%
b	4.24%		4.70%	5.19%	10.01%	3.69%	8.53%	1.09%	5.99%	4.41%	2.97%	4.08%	5.73%	1.07%	2.80%	2.15%	2.34%	9.87%	2.89%	2.47%	3.60%
c	3.49%	4.70%		5.57%	4.36%	5.88%	4.22%	1.35%	3.75%	7.18%	5.86%	8.98%	3.39%	1.78%	9.78%	1.72%	2.06%	3.67%	12.32%	1.46%	7.76%
d	3.63%	5.19%	5.57%		3.51%	3.78%	4.08%	0.75%	3.89%	5.66%	3.52%	4.91%	3.22%	1.14%	4.15%	2.18%	1.73%	3.98%	5.18%	2.26%	4.90%
e	3.61%	10.01%	4.36%	3.51%		3.23%	8.29%	0.49%	4.68%	3.69%	3.35%	3.29%	5.98%	0.89%	2.33%	1.70%	2.13%	10.67%	2.35%	2.60%	3.49%
f	3.08%	3.69%	5.88%	3.78%	3.23%		3.87%	0.90%	2.39%	5.39%	11.35%	5.11%	3.65%	0.86%	4.09%	1.57%	1.96%	3.41%	5.98%	2.02%	5.04%
g	3.44%	8.53%	4.22%	4.08%	8.29%	3.87%		0.68%	6.11%	4.27%	3.34%	4.00%	8.41%	1.05%	2.79%	1.98%	2.00%	9.00%	3.39%	2.36%	3.20%
h	0.46%	1.09%	1.35%	0.75%	0.49%	0.90%	0.68%		1.14%	0.85%	0.64%	1.10%	0.84%	21.26%	1.58%	0.25%	0.29%	0.39%	1.35%	0.14%	0.68%
i	2.82%	5.99%	3.75%	3.89%	4.68%	2.39%	6.11%	1.14%		3.22%	2.65%	3.11%	5.59%	1.29%	2.76%	1.78%	1.58%	5.11%	2.49%	1.44%	3.07%
j	3.16%	4.41%	7.18%	5.66%	3.69%	5.39%	4.27%	0.85%	3.22%		5.02%	7.15%	3.23%	1.04%	5.94%	1.72%	1.76%	3.49%	6.70%	1.49%	9.96%
k	2.51%	2.97%	5.86%	3.52%	3.35%	11.35%	3.34%	0.64%	2.65%	5.02%		4.22%	3.07%	0.56%	4.32%	1.91%	1.51%	3.35%	5.58%	1.38%	4.06%
l	2.32%	4.08%	8.98%	4.91%	3.29%	5.11%	4.00%	1.10%	3.11%	7.15%	4.22%		2.90%	1.28%	7.23%	1.56%	1.80%	3.39%	7.54%	1.36%	9.16%
m	2.50%	5.73%	3.39%	3.22%	5.98%	3.65%	8.41%	0.84%	5.59%	3.23%	3.07%	2.90%		0.88%	2.48%	1.59%	1.54%	6.83%	2.93%	1.77%	2.98%
n	0.70%	1.07%	1.78%	1.14%	0.89%	0.86%	1.05%	21.26%	1.29%	1.04%	0.56%	1.28%	0.88%		2.15%	0.29%	0.22%	0.90%	1.78%	0.57%	0.93%
o	2.94%	2.80%	9.78%	4.15%	2.33%	4.09%	2.79%	1.58%	2.76%	5.94%	4.32%	7.23%	2.48%	2.15%		1.62%	1.70%	2.73%	10.19%	0.99%	5.59%
p	9.02%	2.15%	1.72%	2.18%	1.70%	1.57%	1.98%	0.25%	1.78%	1.72%	1.91%	1.56%	1.59%	0.29%	1.62%		10.56%	2.37%	1.93%	7.50%	1.48%
q	13.38%	2.34%	2.06%	1.73%	2.13%	1.96%	2.00%	0.29%	1.58%	1.76%	1.51%	1.80%	1.54%	0.22%	1.70%	10.56%		2.33%	2.41%	8.56%	1.44%
r	4.29%	9.87%	3.67%	3.98%	10.67%	3.41%	9.00%	0.39%	5.11%	3.49%	3.35%	3.39%	6.83%	0.90%	2.73%	2.37%	2.33%		2.55%	2.88%	3.07%
s	2.88%	2.89%	12.32%	5.18%	2.35%	5.98%	3.39%	1.35%	2.49%	6.70%	5.58%	7.54%	2.93%	1.78%	10.19%	1.93%	2.41%	2.55%		1.46%	6.91%
t	9.37%	2.47%	1.46%	2.26%	2.60%	2.02%	2.36%	0.14%	1.44%	1.49%	1.38%	1.36%	1.77%	0.57%	0.99%	7.50%	8.56%	2.88%	1.46%		1.18%
u	2.57%	3.60%	7.76%	4.90%	3.49%	5.04%	3.20%	0.68%	3.07%	9.96%	4.06%	9.16%	2.98%	0.93%	5.59%	1.48%	1.44%	3.07%	6.91%	1.18%	

Table 1: The first evaluation method that compared product relationships of group members K-means



Cluster 2: highest co-occurence percentages products in

	d		h		n
j	5.66%	n	21.26%	h	21.26%
c	5.57%	o	1.58%	o	2.15%
b	5.19%	c	1.35%	c	1.78%
s	5.18%	s	1.35%	s	1.78%
l	4.91%	i	1.14%	i	1.29%
u	4.90%	l	1.10%	l	1.28%
o	4.15%	b	1.09%	d	1.14%
g	4.08%	f	0.90%	b	1.07%
r	3.98%	j	0.85%	g	1.05%
i	3.89%	m	0.84%	j	1.04%
f	3.78%	d	0.75%	u	0.93%
a	3.63%	g	0.68%	r	0.90%
k	3.52%	u	0.68%	e	0.89%
e	3.51%	k	0.64%	m	0.88%
m	3.22%	e	0.49%	f	0.86%
t	2.26%	a	0.46%	a	0.70%
p	2.18%	r	0.39%	t	0.57%
q	1.73%	q	0.29%	k	0.56%
n	1.14%	p	0.25%	p	0.29%
h	0.75%	t	0.14%	q	0.22%

Cluster 4: highest co-occurence percentages products in the real dataset

	e		g		m		r
r	10.67%	r	9.00%	g	8.41%	e	10.67%
b	10.01%	b	8.53%	r	6.83%	b	9.87%
g	8.29%	m	8.41%	e	5.98%	g	9.00%
m	5.98%	e	8.29%	b	5.73%	m	6.83%

Cluster 5: highest co-occurence percentages products in the real dataset

	a		p		q		t
q	13.38%	q	10.56%	a	13.38%	a	9.37%
t	9.37%	a	9.02%	p	10.56%	q	8.56%
p	9.02%	t	7.50%	t	8.56%	p	7.50%

Cluster 6: highest co-occurence percentages products in the real dataset

	f		j		k		u
k	11.35%	u	9.96%	f	11.35%	j	9.96%
s	5.98%	c	7.18%	c	5.86%	l	9.16%
c	5.88%	l	7.15%	s	5.58%	c	7.76%
j	5.39%	s	6.70%	j	5.02%	s	6.91%
l	5.11%	o	5.94%	o	4.32%	o	5.59%
u	5.04%	d	5.66%	l	4.22%	f	5.04%
o	4.09%	f	5.39%	u	4.06%	d	4.90%

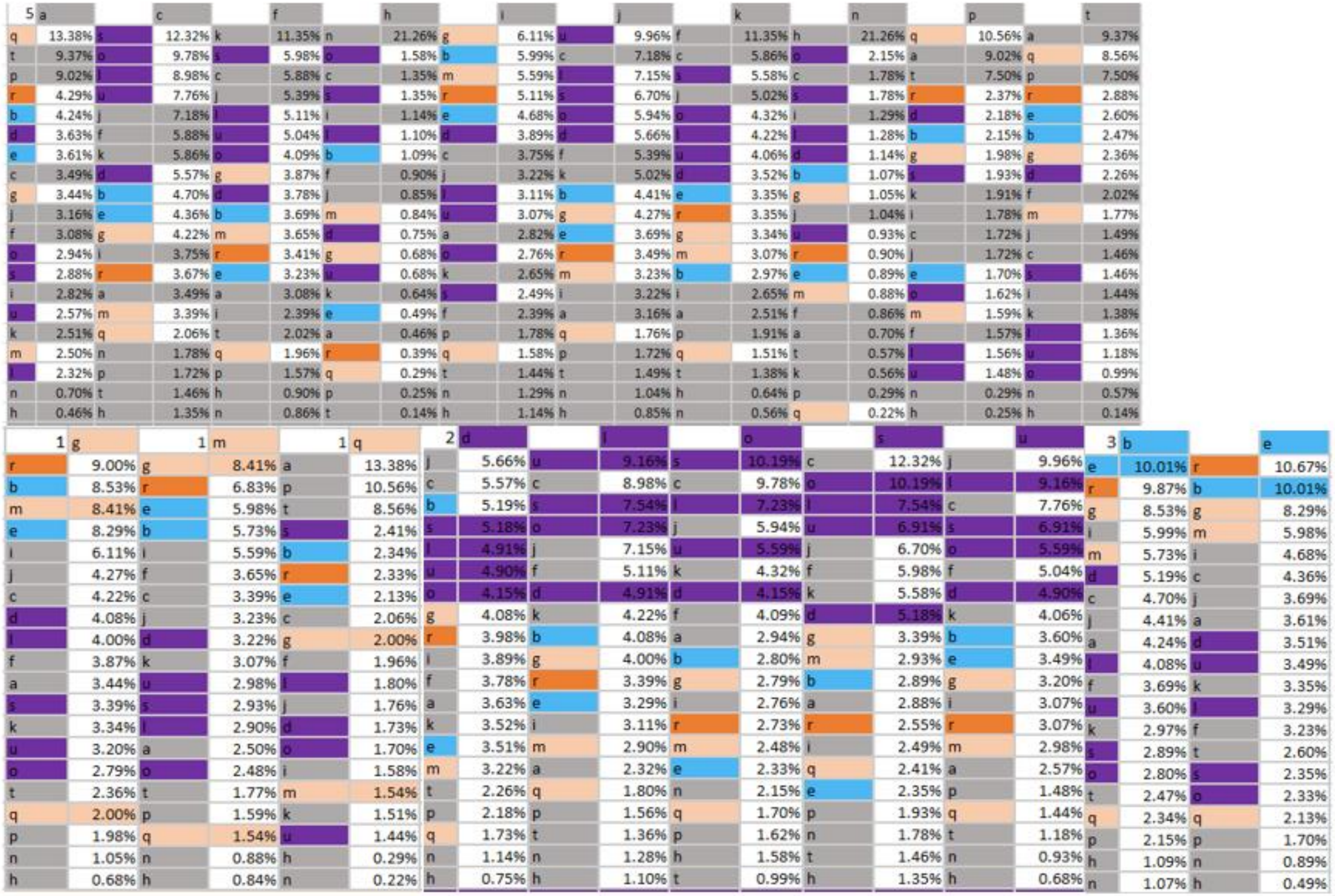
Cluster 7: highest co-occurence percentages products in the real dataset

	c		l		o		s
s	12.32%	u	9.16%	s	10.19%	c	12.32%
o	9.78%	c	8.98%	c	9.78%	o	10.19%
l	8.98%	s	7.54%	l	7.23%	l	7.54%
u	7.76%	o	7.23%	j	5.94%	u	6.91%

Table 2: The first evaluation method that compared product relationships of group members K-means

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u
a		4.24%	3.49%	3.63%	3.61%	3.08%	3.44%	0.46%	2.82%	3.16%	2.51%	2.32%	2.50%	0.70%	2.94%	9.02%	13.38%	4.29%	2.88%	9.37%	2.57%
b	4.24%		4.70%	5.19%	10.01%	3.69%	8.53%	1.09%	5.99%	4.41%	2.97%	4.08%	5.73%	1.07%	2.80%	2.15%	2.34%	9.87%	2.89%	2.47%	3.60%
c	3.49%	4.70%		5.57%	4.36%	5.88%	4.22%	1.35%	3.75%	7.18%	5.86%	8.98%	3.39%	1.78%	9.78%	1.72%	2.06%	3.67%	#####	1.46%	7.76%
d	3.63%	5.19%	5.57%		3.51%	3.78%	4.08%	0.75%	3.89%	5.66%	3.52%	4.91%	3.22%	1.14%	4.15%	2.18%	1.73%	3.98%	5.18%	2.26%	4.90%
e	3.61%	10.01%	4.36%	3.51%		3.23%	8.29%	0.49%	4.68%	3.69%	3.35%	3.29%	5.98%	0.89%	2.33%	1.70%	2.13%	10.67%	2.35%	2.60%	3.49%
f	3.08%	3.69%	5.88%	3.78%	3.23%		3.87%	0.90%	2.39%	5.39%	11.35%	5.11%	3.65%	0.86%	4.09%	1.57%	1.96%	3.41%	5.98%	2.02%	5.04%
g	3.44%	8.53%	4.22%	4.08%	8.29%	3.87%		0.68%	6.11%	4.27%	3.34%	4.00%	8.41%	1.05%	2.79%	1.98%	2.00%	9.00%	3.39%	2.36%	3.20%
h	0.46%	1.09%	1.35%	0.75%	0.49%	0.90%	0.68%		1.14%	0.85%	0.64%	1.10%	0.84%	21.26%	1.58%	0.25%	0.29%	0.39%	1.35%	0.14%	0.68%
i	2.82%	5.99%	3.75%	3.89%	4.68%	2.39%	6.11%	1.14%		3.22%	2.65%	3.11%	5.59%	1.29%	2.76%	1.78%	1.58%	5.11%	2.49%	1.44%	3.07%
j	3.16%	4.41%	7.18%	5.66%	3.69%	5.39%	4.27%	0.85%	3.22%		5.02%	7.15%	3.23%	1.04%	5.94%	1.72%	1.76%	3.49%	6.70%	1.49%	9.96%
k	2.51%	2.97%	5.86%	3.52%	3.35%	11.35%	3.34%	0.64%	2.65%	5.02%		4.22%	3.07%	0.56%	4.32%	1.91%	1.51%	3.35%	5.58%	1.38%	4.06%
l	2.32%	4.08%	8.98%	4.91%	3.29%	5.11%	4.00%	1.10%	3.11%	7.15%	4.22%		2.90%	1.28%	7.23%	1.56%	1.80%	3.39%	7.54%	1.36%	9.16%
m	2.50%	5.73%	3.39%	3.22%	5.98%	3.65%	8.41%	0.84%	5.59%	3.23%	3.07%	2.90%		0.88%	2.48%	1.59%	1.54%	6.83%	2.93%	1.77%	2.98%
n	0.70%	1.07%	1.78%	1.14%	0.89%	0.86%	1.05%	21.26%	1.29%	1.04%	0.56%	1.28%	0.88%		2.15%	0.29%	0.22%	0.90%	1.78%	0.57%	0.93%
o	2.94%	2.80%	9.78%	4.15%	2.33%	4.09%	2.79%	1.58%	2.76%	5.94%	4.32%	7.23%	2.48%	2.15%		1.62%	1.70%	2.73%	#####	0.99%	5.59%
p	9.02%	2.15%	1.72%	2.18%	1.70%	1.57%	1.98%	0.25%	1.78%	1.72%	1.91%	1.56%	1.59%	0.29%	1.62%		10.56%	2.37%	1.93%	7.50%	1.48%
q	13.38%	2.34%	2.06%	1.73%	2.13%	1.96%	2.00%	0.29%	1.58%	1.76%	1.51%	1.80%	1.54%	0.22%	1.70%	10.56%		2.33%	2.41%	8.56%	1.44%
r	4.29%	9.87%	3.67%	3.98%	10.67%	3.41%	9.00%	0.39%	5.11%	3.49%	3.35%	3.39%	6.83%	0.90%	2.73%	2.37%	2.33%		2.55%	2.88%	3.07%
s	2.88%	2.89%	12.32%	5.18%	2.35%	5.98%	3.39%	1.35%	2.49%	6.70%	5.58%	7.54%	2.93%	1.78%	10.19%	1.93%	2.41%	2.55%		1.46%	6.91%
t	9.37%	2.47%	1.46%	2.26%	2.60%	2.02%	2.36%	0.14%	1.44%	1.49%	1.38%	1.36%	1.77%	0.57%	0.99%	7.50%	8.56%	2.88%	1.46%		1.18%
u	2.57%	3.60%	7.76%	4.90%	3.49%	5.04%	3.20%	0.68%	3.07%	9.96%	4.06%	9.16%	2.98%	0.93%	5.59%	1.48%	1.44%	3.07%	6.91%	1.18%	

Table 3: The first evaluation method that compared product relationships of group members GMM



5	a		c		f		h		i		j		k		n		p		t
q	13.38%	s	12.32%	k	11.35%	n	21.26%	g	6.11%	u	9.96%	f	11.35%	h	21.26%	q	10.56%	a	9.37%
t	9.37%	o	9.78%	s	5.98%	o	1.58%	b	5.99%	c	7.18%	c	5.86%	o	2.15%	a	9.02%	q	8.56%
p	9.02%	l	8.98%	c	5.88%	c	1.35%	m	5.59%	l	7.15%	s	5.58%	c	1.78%	t	7.50%	p	7.50%
r	4.29%	u	7.76%	j	5.39%	s	1.35%	r	5.11%	s	6.70%	j	5.02%	s	1.78%	r	2.37%	r	2.88%
b	4.24%	j	7.18%	l	5.11%	i	1.14%	e	4.68%	o	5.94%	o	4.32%	i	1.29%	d	2.18%	e	2.60%
d	3.63%	f	5.88%	u	5.04%	l	1.10%	d	3.89%	d	5.66%	l	4.22%	l	1.28%	b	2.15%	b	2.47%
e	3.61%	k	5.86%	o	4.09%	b	1.09%	c	3.75%	f	5.39%	u	4.06%	d	1.14%	g	1.98%	g	2.36%
c	3.49%	d	5.57%	g	3.87%	f	0.90%	j	3.22%	k	5.02%	d	3.52%	b	1.07%	s	1.93%	d	2.26%
g	3.44%	b	4.70%	d	3.78%	j	0.85%	l	3.11%	b	4.41%	e	3.35%	g	1.05%	k	1.91%	f	2.02%
j	3.16%	e	4.36%	b	3.69%	m	0.84%	u	3.07%	g	4.27%	r	3.35%	j	1.04%	i	1.78%	m	1.77%
f	3.08%	g	4.22%	m	3.65%	d	0.75%	a	2.82%	e	3.69%	g	3.34%	u	0.93%	c	1.72%	j	1.49%
o	2.94%	i	3.75%	r	3.41%	g	0.68%	o	2.76%	r	3.49%	m	3.07%	r	0.90%	j	1.72%	c	1.46%
s	2.88%	r	3.67%	e	3.23%	u	0.68%	k	2.65%	m	3.23%	b	2.97%	e	0.89%	e	1.70%	s	1.46%
i	2.82%	a	3.49%	a	3.08%	k	0.64%	s	2.49%	i	3.22%	i	2.65%	m	0.88%	o	1.62%	i	1.44%
u	2.57%	m	3.39%	i	2.39%	e	0.49%	f	2.39%	a	3.16%	a	2.51%	f	0.86%	m	1.59%	k	1.38%
k	2.51%	q	2.06%	t	2.02%	a	0.46%	p	1.78%	q	1.76%	p	1.91%	a	0.70%	f	1.57%	l	1.36%
m	2.50%	n	1.78%	q	1.96%	r	0.39%	q	1.58%	p	1.72%	q	1.51%	t	0.57%	l	1.56%	u	1.18%
l	2.32%	p	1.72%	p	1.57%	q	0.29%	t	1.44%	t	1.49%	t	1.38%	k	0.56%	u	1.48%	o	0.99%
n	0.70%	t	1.46%	h	0.90%	p	0.25%	n	1.29%	n	1.04%	h	0.64%	p	0.29%	n	0.29%	n	0.57%
h	0.46%	h	1.35%	n	0.86%	t	0.14%	h	1.14%	h	0.85%	n	0.56%	q	0.22%	h	0.25%	h	0.14%

1	g	1	m	1	q	2	d		l		o		s		u	3	b		e
r	9.00%	g	8.41%	a	13.38%	j	5.66%	u	9.16%	s	10.19%	c	12.32%	j	9.96%	e	10.01%	r	10.67%
b	8.53%	r	6.83%	p	10.56%	c	5.57%	c	8.98%	c	9.78%	o	10.19%	l	9.16%	r	9.87%	b	10.01%
m	8.41%	e	5.98%	t	8.56%	b	5.19%	s	7.54%	l	7.23%	l	7.54%	c	7.76%	g	8.53%	g	8.29%
e	8.29%	b	5.73%	s	2.41%	s	5.18%	o	7.23%	j	5.94%	u	6.91%	s	6.91%	i	5.99%	m	5.98%
i	6.11%	i	5.59%	b	2.34%	l	4.91%	j	7.15%	u	5.59%	j	6.70%	o	5.59%	m	5.73%	i	4.68%
j	4.27%	f	3.65%	r	2.33%	u	4.90%	f	5.11%	k	4.32%	f	5.98%	f	5.04%	d	5.19%	c	4.36%
c	4.22%	c	3.39%	e	2.13%	o	4.15%	d	4.91%	d	4.15%	k	5.58%	d	4.90%	c	4.70%	j	3.69%
d	4.08%	j	3.23%	c	2.06%	g	4.08%	k	4.22%	f	4.09%	d	5.18%	k	4.06%	j	4.41%	a	3.61%
l	4.00%	d	3.22%	g	2.00%	r	3.98%	b	4.08%	a	2.94%	g	3.39%	b	3.60%	a	4.24%	d	3.51%
f	3.87%	k	3.07%	f	1.96%	i	3.89%	g	4.00%	b	2.80%	m	2.93%	e	3.49%	l	4.08%	u	3.49%
a	3.44%	u	2.98%	l	1.80%	f	3.78%	r	3.39%	g	2.79%	b	2.89%	g	3.20%	f	3.69%	k	3.35%
s	3.39%	s	2.93%	j	1.76%	a	3.63%	e	3.29%	i	2.76%	a	2.88%	i	3.07%	u	3.60%	l	3.29%
k	3.34%	l	2.90%	d	1.73%	k	3.52%	i	3.11%	r	2.73%	r	2.55%	r	3.07%	k	2.97%	f	3.23%
u	3.20%	a	2.50%	o	1.70%	e	3.51%	m	2.90%	m	2.48%	i	2.49%	m	2.98%	s	2.89%	t	2.60%
o	2.79%	o	2.48%	i	1.58%	m	3.22%	a	2.32%	e	2.33%	q	2.41%	a	2.57%	o	2.80%	s	2.35%
t	2.36%	t	1.77%	m	1.54%	t	2.26%	q	1.80%	n	2.15%	e	2.35%	p	1.48%	t	2.47%	o	2.33%
q	2.00%	p	1.59%	k	1.51%	p	2.18%	p	1.56%	q	1.70%	p	1.93%	q	1.44%	q	2.34%	q	2.13%
p	1.98%	q	1.54%	u	1.44%	q	1.73%	t	1.36%	p	1.62%	n	1.78%	t	1.18%	p	2.15%	p	1.70%
n	1.05%	n	0.88%	h	0.29%	n	1.14%	n	1.28%	h	1.58%	t	1.46%	n	0.93%	h	1.09%	n	0.89%
h	0.68%	h	0.84%	n	0.22%	h	0.75%	h	1.10%	t	0.99%	h	1.35%	h	0.68%	n	1.07%	h	0.49%

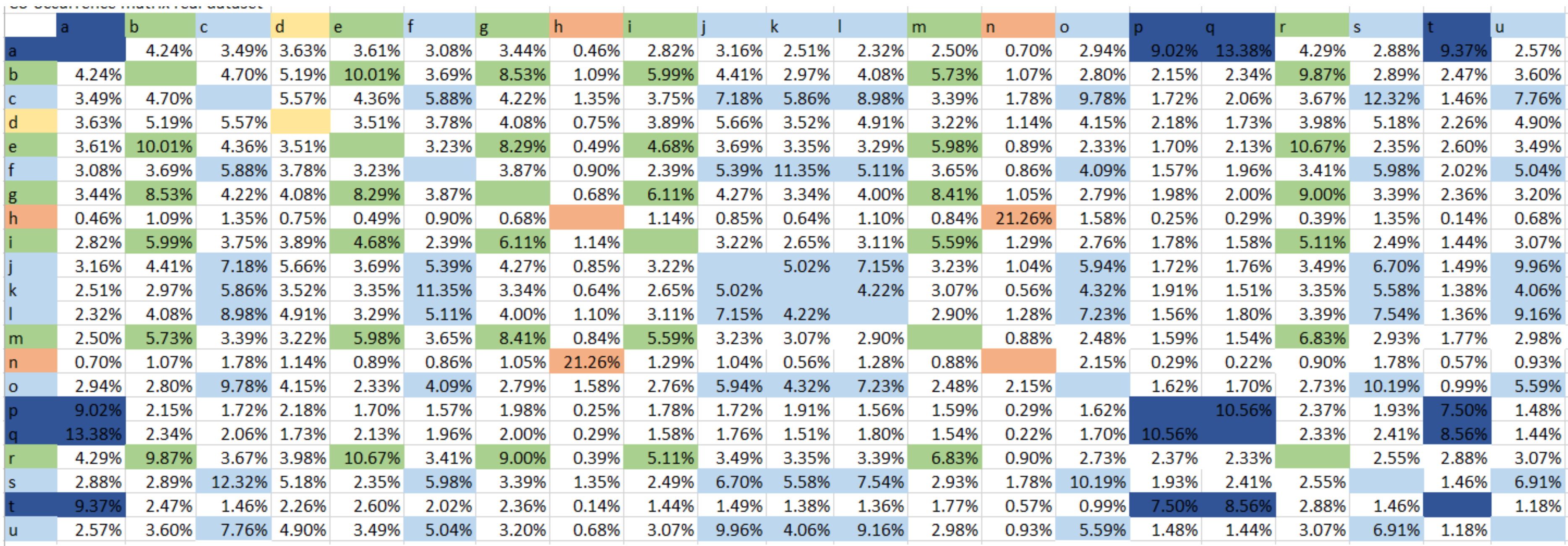
Table 4: The first evaluation method that compared product relationships of group members GMM

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u
a		3.27%	37.82%	19.16%	13.38%	42.16%	32.19%	34.39%	34.75%	33.49%	49.38%	25.72%	28.84%	30.31%	23.97%	32.30%	10.47%	23.73%	40.41%	17.44%	17.44%
b	3.27%		4.08%	2.11%	60.17%	8.29%	23.44%	0.55%	23.38%	5.20%	0.99%	5.00%	27.29%	37.92%	7.93%	0.80%	31.09%	0.97%	10.43%	4.94%	4.94%
c	37.82%	4.08%		33.95%	5.11%	18.68%	31.90%	32.00%	25.14%	46.99%	41.34%	39.07%	11.62%	21.49%	26.74%	15.25%	10.44%	27.58%	10.22%	40.09%	19.34%
d	19.16%	2.11%	33.95%		7.07%	18.68%	9.31%	37.45%	17.82%	35.72%	25.28%	39.20%	8.71%	14.65%	32.10%	25.68%	3.81%	41.71%	21.33%	43.59%	43.59%
e	13.38%	60.17%	5.11%	7.07%		16.17%	31.90%	3.96%	22.29%	6.26%	5.00%	8.08%	34.34%	29.51%	16.83%	16.50%	26.76%	7.49%	15.17%	9.79%	9.79%
f	42.16%	8.29%	18.68%	18.68%	16.17%		31.53%	32.00%	33.44%	31.37%	45.29%	24.57%	28.73%	30.08%	23.42%	30.63%	11.68%	22.72%	37.94%	17.31%	17.31%
g	32.19%	23.44%	31.90%	9.31%	31.90%	31.53%		9.34%	25.14%	8.82%	16.66%	10.58%	46.72%	22.90%	26.88%	37.74%	21.33%	13.25%	25.97%	11.69%	11.69%
h	34.39%	0.55%	32.00%	37.45%	3.96%	32.00%	9.34%		30.33%	46.99%	51.25%	40.16%	6.25%	27.55%	24.49%	21.25%	2.47%	37.35%	35.58%	33.73%	33.73%
i	34.75%	23.38%	25.14%	17.82%	22.29%	33.44%	25.14%	30.33%		30.89%	41.34%	23.81%	23.27%	33.19%	19.10%	21.63%	13.71%	20.45%	33.89%	16.50%	16.50%
j	33.49%	5.20%	46.99%	35.72%	6.26%	31.37%	8.82%	46.99%	30.89%		51.06%	39.07%	5.89%	29.21%	22.82%	19.02%	3.53%	35.63%	35.26%	32.05%	32.05%
k	49.38%	0.99%	41.34%	25.28%	5.00%	45.29%	16.66%	51.25%	41.34%	51.06%		35.96%	11.62%	38.50%	18.19%	21.04%	4.30%	29.26%	47.90%	20.60%	20.60%
l	25.72%	5.00%	39.07%	39.20%	8.08%	24.57%	10.58%	40.16%	23.81%	39.07%	35.96%		9.03%	21.49%	27.53%	22.92%	4.71%	37.51%	27.44%	36.88%	36.88%
m	28.84%	27.29%	11.62%	8.71%	34.34%	28.73%	46.72%	6.25%	23.27%	5.89%	11.62%	9.03%		21.78%	26.74%	36.88%	22.85%	12.16%	23.11%	11.56%	11.56%
n	30.31%	37.92%	21.49%	14.65%	29.51%	30.08%	22.90%	27.55%	33.19%	29.21%	38.50%	21.49%	21.78%		15.31%	15.25%	16.09%	16.74%	31.63%	13.60%	13.60%
o	23.97%	7.93%	26.74%	32.10%	16.83%	23.42%	26.88%	24.49%	19.10%	22.82%	18.19%	27.53%	26.74%	15.31%		34.09%	10.44%	31.47%	21.98%	32.54%	32.54%
p	32.30%	0.80%	15.25%	25.68%	16.50%	30.63%	37.74%	21.25%	21.63%	19.02%	21.04%	22.92%	36.88%	15.25%	34.09%		12.28%	27.58%	26.41%	26.65%	26.65%
q	10.47%	31.09%	10.44%	3.81%	26.76%	11.68%	21.33%	2.47%	13.71%	3.53%	4.30%	4.71%	22.85%	16.09%	10.44%	12.28%		4.36%	10.22%	5.43%	5.43%
r	23.73%	0.97%	27.58%	41.71%	7.49%	22.72%	13.25%	37.35%	20.45%	35.63%	29.26%	37.51%	12.16%	16.74%	31.47%	27.58%	4.36%		24.57%	40.09%	40.09%
s	40.41%	10.43%	10.22%	21.33%	15.17%	37.94%	25.97%	35.58%	33.89%	35.26%	47.90%	27.44%	23.11%	31.63%	21.98%	26.41%	10.22%	24.57%		19.34%	19.34%
t	17.44%	4.94%	40.09%	43.59%	9.79%	17.31%	11.69%	33.73%	16.50%	32.05%	20.60%	36.88%	11.56%	13.60%	32.54%	26.65%	5.43%	40.09%	19.34%		42.60%
u	17.44%	4.94%	19.34%	43.59%	9.79%	17.31%	11.69%	33.73%	16.50%	32.05%	20.60%	36.88%	11.56%	13.60%	32.54%	26.65%	5.43%	40.09%	19.34%	42.60%	

Table 5: The second evaluation method that compared all possible product relationships with the data GMM

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u
a		0.97%	34.33%	15.53%	9.77%	39.08%	28.75%	33.93%	31.93%	30.33%	46.87%	23.40%	26.34%	29.61%	21.03%	23.28%	2.91%	19.44%	37.53%	8.07%	14.87%
b	0.97%		0.62%	3.08%	50.16%	4.60%	14.91%	0.54%	17.39%	0.79%	1.98%	0.92%	21.56%	36.85%	5.13%	1.35%	28.75%	8.90%	7.54%	2.47%	1.34%
c	34.33%	0.62%		28.38%	0.75%	12.80%	27.68%	30.65%	21.39%	39.81%	35.48%	30.09%	8.23%	19.71%	16.96%	13.53%	8.38%	23.91%	2.10%	38.63%	11.58%
d	15.53%	3.08%	28.38%		3.56%	14.90%	5.23%	36.70%	13.93%	30.06%	21.76%	34.29%	5.49%	13.51%	27.95%	23.50%	2.08%	37.73%	16.15%	41.33%	38.69%
e	9.77%	50.16%	0.75%	3.56%		12.94%	23.61%	3.47%	17.61%	2.57%	1.65%	4.79%	28.36%	28.62%	14.50%	14.80%	24.63%	3.18%	12.82%	7.19%	6.30%
f	39.08%	4.60%	12.80%	14.90%	12.94%		27.66%	31.10%	31.05%	25.98%	33.94%	19.46%	25.08%	29.22%	19.33%	29.06%	9.72%	19.31%	31.96%	15.29%	12.27%
g	28.75%	14.91%	27.68%	5.23%	23.61%	27.66%		8.66%	19.03%	4.55%	13.32%	6.58%	38.31%	21.85%	24.09%	35.76%	19.33%	4.25%	22.58%	9.33%	8.49%
h	33.93%	0.54%	30.65%	36.70%	3.47%	31.10%	8.66%		29.19%	46.14%	50.61%	39.06%	5.41%	6.29%	22.91%	21.00%	2.18%	36.96%	34.23%	33.59%	33.05%
i	31.93%	17.39%	21.39%	13.93%	17.61%	31.05%	19.03%	29.19%		27.67%	38.69%	20.70%	17.68%	31.90%	16.34%	19.85%	12.13%	15.34%	31.40%	15.06%	13.43%
j	30.33%	0.79%	39.81%	30.06%	2.57%	25.98%	4.55%	46.14%	27.67%		46.04%	31.92%	2.66%	28.17%	16.88%	17.30%	1.77%	32.14%	28.56%	30.56%	22.09%
k	46.87%	1.98%	35.48%	21.76%	1.65%	33.94%	13.32%	50.61%	38.69%	46.04%		31.74%	8.55%	37.94%	13.87%	19.13%	2.79%	25.91%	42.32%	19.22%	16.54%
l	23.40%	0.92%	30.09%	34.29%	4.79%	19.46%	6.58%	39.06%	20.70%	31.92%	31.74%		6.13%	20.21%	20.30%	21.36%	2.91%	34.12%	19.90%	35.52%	27.72%
m	26.34%	21.56%	8.23%	5.49%	28.36%	25.08%	38.31%	5.41%	17.68%	2.66%	8.55%	6.13%		20.90%	24.26%	35.29%	21.31%	5.33%	20.18%	9.79%	8.58%
n	29.61%	36.85%	19.71%	13.51%	28.62%	29.22%	21.85%	6.29%	31.90%	28.17%	37.94%	20.21%	20.90%		13.16%	14.96%	15.87%	15.84%	29.85%	13.03%	12.67%
o	21.03%	5.13%	16.96%	27.95%	14.50%	19.33%	24.09%	22.91%	16.34%	16.88%	13.87%	20.30%	24.26%	13.16%		32.47%	8.74%	28.74%	11.79%	31.55%	26.95%
p	23.28%	1.35%	13.53%	23.50%	14.80%	29.06%	35.76%	21.00%	19.85%	17.30%	19.13%	21.36%	35.29%	14.96%	32.47%		1.72%	25.21%	24.48%	19.15%	25.17%
q	2.91%	28.75%	8.38%	2.08%	24.63%	9.72%	19.33%	2.18%	12.13%	1.77%	2.79%	2.91%	21.31%	15.87%	8.74%	1.72%		2.03%	7.81%	3.13%	3.99%
r	19.44%	8.90%	23.91%	37.73%	3.18%	19.31%	4.25%	36.96%	15.34%	32.14%	25.91%	34.12%	5.33%	15.84%	28.74%	25.21%	2.03%		22.02%	37.21%	37.02%
s	37.53%	7.54%	2.10%	16.15%	12.82%	31.96%	22.58%	34.23%	31.40%	28.56%	42.32%	19.90%	20.18%	29.85%	11.79%	24.48%	7.81%	22.02%		17.88%	12.43%
t	8.07%	2.47%	38.63%	41.33%	7.19%	15.29%	9.33%	33.59%	15.06%	30.56%	19.22%	35.52%	9.79%	13.03%	31.55%	19.15%	3.13%	37.21%	17.88%		41.42%
u	14.87%	1.34%	11.58%	38.69%	6.30%	12.27%	8.49%	33.05%	13.43%	22.09%	16.54%	27.72%	8.58%	12.67%	26.95%	25.17%	3.99%	37.02%	12.43%	41.42%	

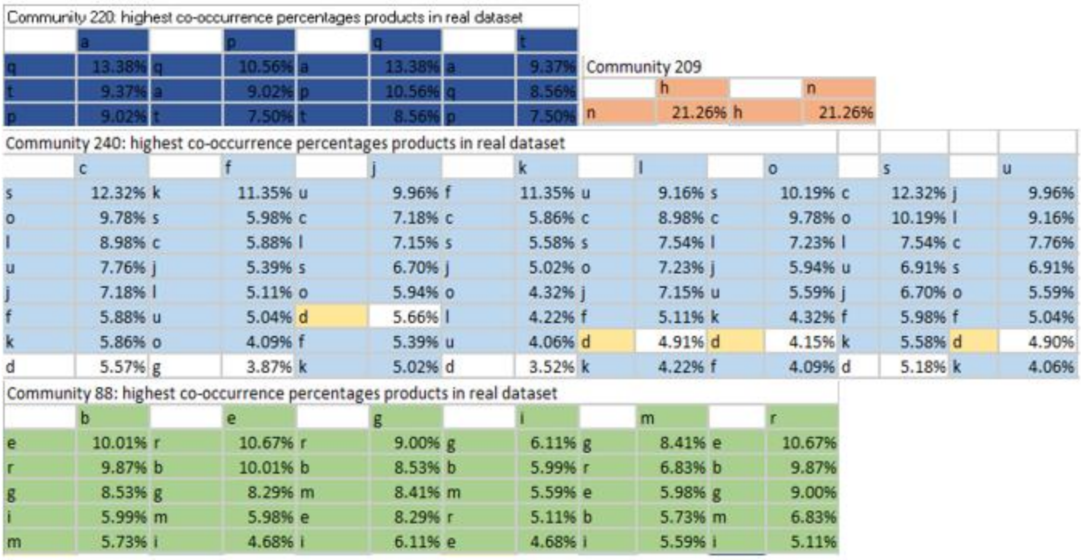
Table 6: The second evaluation method that compared all possible product relationships with the data GMM differences with the co-occurrences



Co-occurrence matrix real dataset

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u
a		4.24%	3.49%	3.63%	3.61%	3.08%	3.44%	0.46%	2.82%	3.16%	2.51%	2.32%	2.50%	0.70%	2.94%	9.02%	13.38%	4.29%	2.88%	9.37%	2.57%
b	4.24%		4.70%	5.19%	10.01%	3.69%	8.53%	1.09%	5.99%	4.41%	2.97%	4.08%	5.73%	1.07%	2.80%	2.15%	2.34%	9.87%	2.89%	2.47%	3.60%
c	3.49%	4.70%		5.57%	4.36%	5.88%	4.22%	1.35%	3.75%	7.18%	5.86%	8.98%	3.39%	1.78%	9.78%	1.72%	2.06%	3.67%	12.32%	1.46%	7.76%
d	3.63%	5.19%	5.57%		3.51%	3.78%	4.08%	0.75%	3.89%	5.66%	3.52%	4.91%	3.22%	1.14%	4.15%	2.18%	1.73%	3.98%	5.18%	2.26%	4.90%
e	3.61%	10.01%	4.36%	3.51%		3.23%	8.29%	0.49%	4.68%	3.69%	3.35%	3.29%	5.98%	0.89%	2.33%	1.70%	2.13%	10.67%	2.35%	2.60%	3.49%
f	3.08%	3.69%	5.88%	3.78%	3.23%		3.87%	0.90%	2.39%	5.39%	11.35%	5.11%	3.65%	0.86%	4.09%	1.57%	1.96%	3.41%	5.98%	2.02%	5.04%
g	3.44%	8.53%	4.22%	4.08%	8.29%	3.87%		0.68%	6.11%	4.27%	3.34%	4.00%	8.41%	1.05%	2.79%	1.98%	2.00%	9.00%	3.39%	2.36%	3.20%
h	0.46%	1.09%	1.35%	0.75%	0.49%	0.90%	0.68%		1.14%	0.85%	0.64%	1.10%	0.84%	21.26%	1.58%	0.25%	0.29%	0.39%	1.35%	0.14%	0.68%
i	2.82%	5.99%	3.75%	3.89%	4.68%	2.39%	6.11%	1.14%		3.22%	2.65%	3.11%	5.59%	1.29%	2.76%	1.78%	1.58%	5.11%	2.49%	1.44%	3.07%
j	3.16%	4.41%	7.18%	5.66%	3.69%	5.39%	4.27%	0.85%	3.22%		5.02%	7.15%	3.23%	1.04%	5.94%	1.72%	1.76%	3.49%	6.70%	1.49%	9.96%
k	2.51%	2.97%	5.86%	3.52%	3.35%	11.35%	3.34%	0.64%	2.65%	5.02%		4.22%	3.07%	0.56%	4.32%	1.91%	1.51%	3.35%	5.58%	1.38%	4.06%
l	2.32%	4.08%	8.98%	4.91%	3.29%	5.11%	4.00%	1.10%	3.11%	7.15%	4.22%		2.90%	1.28%	7.23%	1.56%	1.80%	3.39%	7.54%	1.36%	9.16%
m	2.50%	5.73%	3.39%	3.22%	5.98%	3.65%	8.41%	0.84%	5.59%	3.23%	3.07%	2.90%		0.88%	2.48%	1.59%	1.54%	6.83%	2.93%	1.77%	2.98%
n	0.70%	1.07%	1.78%	1.14%	0.89%	0.86%	1.05%	21.26%	1.29%	1.04%	0.56%	1.28%	0.88%		2.15%	0.29%	0.22%	0.90%	1.78%	0.57%	0.93%
o	2.94%	2.80%	9.78%	4.15%	2.33%	4.09%	2.79%	1.58%	2.76%	5.94%	4.32%	7.23%	2.48%	2.15%		1.62%	1.70%	2.73%	10.19%	0.99%	5.59%
p	9.02%	2.15%	1.72%	2.18%	1.70%	1.57%	1.98%	0.25%	1.78%	1.72%	1.91%	1.56%	1.59%	0.29%	1.62%		10.56%	2.37%	1.93%	7.50%	1.48%
q	13.38%	2.34%	2.06%	1.73%	2.13%	1.96%	2.00%	0.29%	1.58%	1.76%	1.51%	1.80%	1.54%	0.22%	1.70%	10.56%		2.33%	2.41%	8.56%	1.44%
r	4.29%	9.87%	3.67%	3.98%	10.67%	3.41%	9.00%	0.39%	5.11%	3.49%	3.35%	3.39%	6.83%	0.90%	2.73%	2.37%	2.33%		2.55%	2.88%	3.07%
s	2.88%	2.89%	12.32%	5.18%	2.35%	5.98%	3.39%	1.35%	2.49%	6.70%	5.58%	7.54%	2.93%	1.78%	10.19%	1.93%	2.41%	2.55%		1.46%	6.91%
t	9.37%	2.47%	1.46%	2.26%	2.60%	2.02%	2.36%	0.14%	1.44%	1.49%	1.38%	1.36%	1.77%	0.57%	0.99%	7.50%	8.56%	2.88%	1.46%		1.18%
u	2.57%	3.60%	7.76%	4.90%	3.49%	5.04%	3.20%	0.68%	3.07%	9.96%	4.06%	9.16%	2.98%	0.93%	5.59%	1.48%	1.44%	3.07%	6.91%	1.18%	

Table 7: The first evaluation method that compared product relationships of group members the Louvain method



Community 220: highest co-occurrence percentages products in real dataset

	a		p		q		t
q	13.38%	q	10.56%	a	13.38%	a	9.37%
t	9.37%	a	9.02%	p	10.56%	q	8.56%
p	9.02%	t	7.50%	t	8.56%	p	7.50%

Community 209

	h		n
n	21.26%	h	21.26%

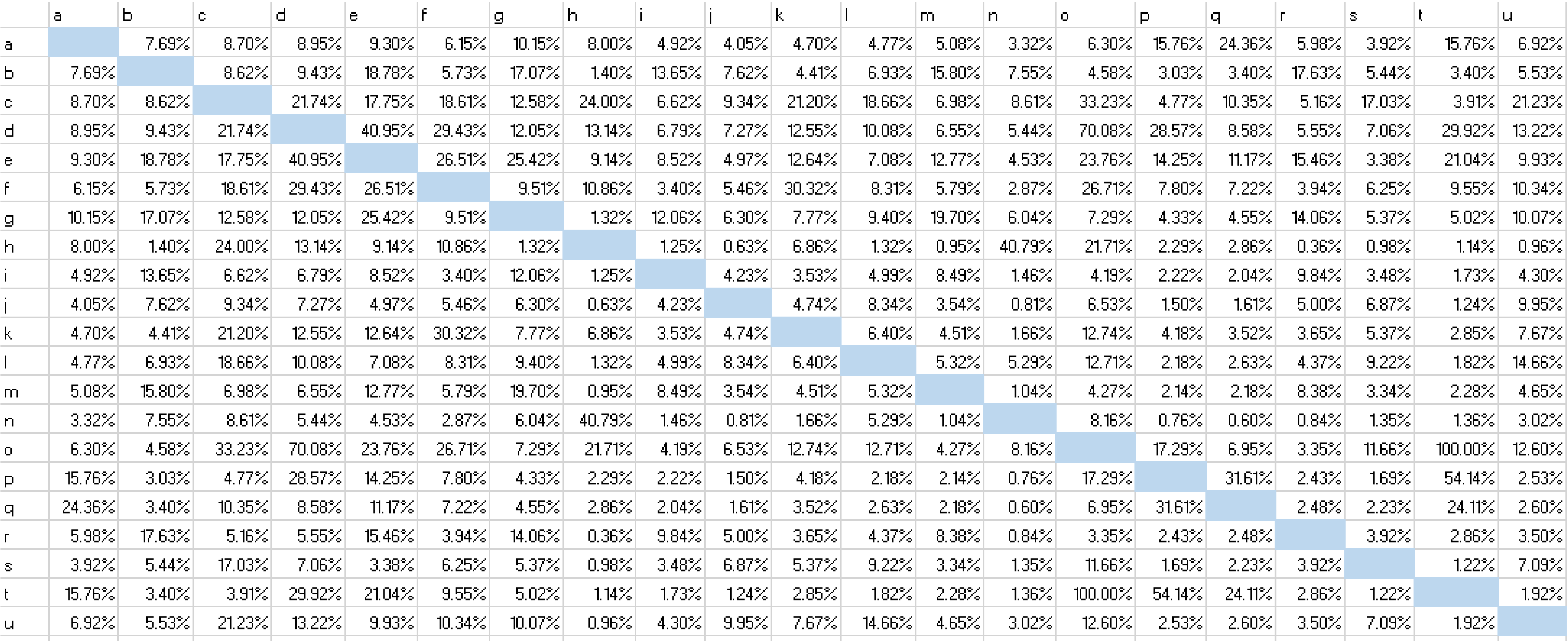
Community 240: highest co-occurrence percentages products in real dataset

	c		f		j		k		l		o		s		u
s	12.32%	k	11.35%	u	9.96%	f	11.35%	u	9.16%	s	10.19%	c	12.32%	j	9.96%
o	9.78%	s	5.98%	c	7.18%	c	5.86%	c	8.98%	c	9.78%	o	10.19%	l	9.16%
l	8.98%	c	5.88%	l	7.15%	s	5.58%	s	7.54%	l	7.23%	l	7.54%	c	7.76%
u	7.76%	j	5.39%	s	6.70%	j	5.02%	o	7.23%	j	5.94%	u	6.91%	s	6.91%
j	7.18%	l	5.11%	o	5.94%	o	4.32%	j	7.15%	u	5.59%	j	6.70%	o	5.59%
f	5.88%	u	5.04%	d	5.66%	l	4.22%	f	5.11%	k	4.32%	f	5.98%	f	5.04%
k	5.86%	o	4.09%	f	5.39%	u	4.06%	d	4.91%	d	4.15%	k	5.58%	d	4.90%
d	5.57%	g	3.87%	k	5.02%	d	3.52%	k	4.22%	f	4.09%	d	5.18%	k	4.06%

Community 88: highest co-occurrence percentages products in real dataset

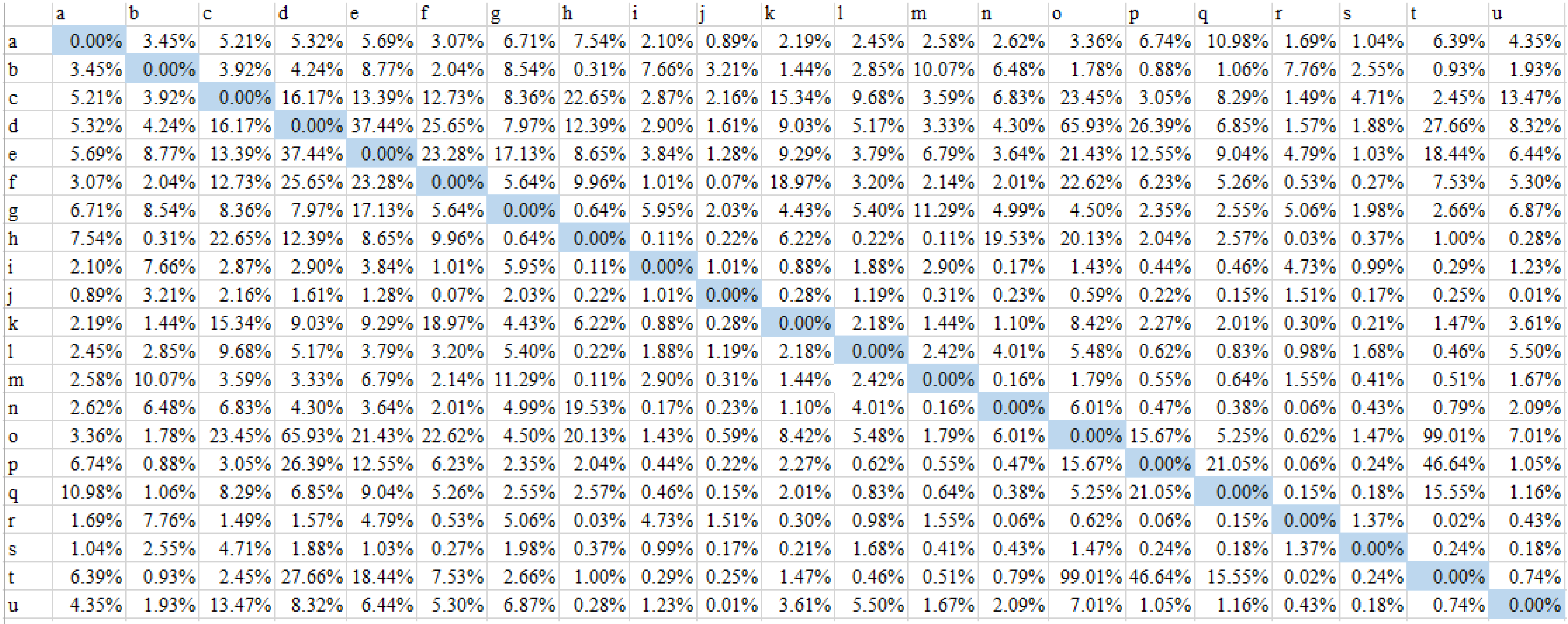
	b		e		g		i		m		r
e	10.01%	r	10.67%	r	9.00%	g	6.11%	g	8.41%	e	10.67%
r	9.87%	b	10.01%	b	8.53%	b	5.99%	r	6.83%	b	9.87%
g	8.53%	g	8.29%	m	8.41%	m	5.59%	e	5.98%	g	9.00%
i	5.99%	m	5.98%	e	8.29%	r	5.11%	b	5.73%	m	6.83%
m	5.73%	i	4.68%	i	6.11%	e	4.68%	i	5.59%	i	5.11%

Table 7: The first evaluation method that compared product relationships of group members the Louvain method



	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u
a		7.69%	8.70%	8.95%	9.30%	6.15%	10.15%	8.00%	4.92%	4.05%	4.70%	4.77%	5.08%	3.32%	6.30%	15.76%	24.36%	5.98%	3.92%	15.76%	6.92%
b	7.69%		8.62%	9.43%	18.78%	5.73%	17.07%	1.40%	13.65%	7.62%	4.41%	6.93%	15.80%	7.55%	4.58%	3.03%	3.40%	17.63%	5.44%	3.40%	5.53%
c	8.70%	8.62%		21.74%	17.75%	18.61%	12.58%	24.00%	6.62%	9.34%	21.20%	18.66%	6.98%	8.61%	33.23%	4.77%	10.35%	5.16%	17.03%	3.91%	21.23%
d	8.95%	9.43%	21.74%		40.95%	29.43%	12.05%	13.14%	6.79%	7.27%	12.55%	10.08%	6.55%	5.44%	70.08%	28.57%	8.58%	5.55%	7.06%	29.92%	13.22%
e	9.30%	18.78%	17.75%	40.95%		26.51%	25.42%	9.14%	8.52%	4.97%	12.64%	7.08%	12.77%	4.53%	23.76%	14.25%	11.17%	15.46%	3.38%	21.04%	9.93%
f	6.15%	5.73%	18.61%	29.43%	26.51%		9.51%	10.86%	3.40%	5.46%	30.32%	8.31%	5.79%	2.87%	26.71%	7.80%	7.22%	3.94%	6.25%	9.55%	10.34%
g	10.15%	17.07%	12.58%	12.05%	25.42%	9.51%		1.32%	12.06%	6.30%	7.77%	9.40%	19.70%	6.04%	7.29%	4.33%	4.55%	14.06%	5.37%	5.02%	10.07%
h	8.00%	1.40%	24.00%	13.14%	9.14%	10.86%	1.32%		1.25%	0.63%	6.86%	1.32%	0.95%	40.79%	21.71%	2.29%	2.86%	0.36%	0.98%	1.14%	0.96%
i	4.92%	13.65%	6.62%	6.79%	8.52%	3.40%	12.06%	1.25%		4.23%	3.53%	4.99%	8.49%	1.46%	4.19%	2.22%	2.04%	9.84%	3.48%	1.73%	4.30%
j	4.05%	7.62%	9.34%	7.27%	4.97%	5.46%	6.30%	0.63%	4.23%		4.74%	8.34%	3.54%	0.81%	6.53%	1.50%	1.61%	5.00%	6.87%	1.24%	9.95%
k	4.70%	4.41%	21.20%	12.55%	12.64%	30.32%	7.77%	6.86%	3.53%	4.74%		6.40%	4.51%	1.66%	12.74%	4.18%	3.52%	3.65%	5.37%	2.85%	7.67%
l	4.77%	6.93%	18.66%	10.08%	7.08%	8.31%	9.40%	1.32%	4.99%	8.34%	6.40%		5.32%	5.29%	12.71%	2.18%	2.63%	4.37%	9.22%	1.82%	14.66%
m	5.08%	15.80%	6.98%	6.55%	12.77%	5.79%	19.70%	0.95%	8.49%	3.54%	4.51%	5.32%		1.04%	4.27%	2.14%	2.18%	8.38%	3.34%	2.28%	4.65%
n	3.32%	7.55%	8.61%	5.44%	4.53%	2.87%	6.04%	40.79%	1.46%	0.81%	1.66%	5.29%	1.04%		8.16%	0.76%	0.60%	0.84%	1.35%	1.36%	3.02%
o	6.30%	4.58%	33.23%	70.08%	23.76%	26.71%	7.29%	21.71%	4.19%	6.53%	12.74%	12.71%	4.27%	8.16%		17.29%	6.95%	3.35%	11.66%	100.00%	12.60%
p	15.76%	3.03%	4.77%	28.57%	14.25%	7.80%	4.33%	2.29%	2.22%	1.50%	4.18%	2.18%	2.14%	0.76%	17.29%		31.61%	2.43%	1.69%	54.14%	2.53%
q	24.36%	3.40%	10.35%	8.58%	11.17%	7.22%	4.55%	2.86%	2.04%	1.61%	3.52%	2.63%	2.18%	0.60%	6.95%	31.61%		2.48%	2.23%	24.11%	2.60%
r	5.98%	17.63%	5.16%	5.55%	15.46%	3.94%	14.06%	0.36%	9.84%	5.00%	3.65%	4.37%	8.38%	0.84%	3.35%	2.43%	2.48%		3.92%	2.86%	3.50%
s	3.92%	5.44%	17.03%	7.06%	3.38%	6.25%	5.37%	0.98%	3.48%	6.87%	5.37%	9.22%	3.34%	1.35%	11.66%	1.69%	2.23%	3.92%		1.22%	7.09%
t	15.76%	3.40%	3.91%	29.92%	21.04%	9.55%	5.02%	1.14%	1.73%	1.24%	2.85%	1.82%	2.28%	1.36%	100.00%	54.14%	24.11%	2.86%	1.22%		1.92%
u	6.92%	5.53%	21.23%	13.22%	9.93%	10.34%	10.07%	0.96%	4.30%	9.95%	7.67%	14.66%	4.65%	3.02%	12.60%	2.53%	2.60%	3.50%	7.09%	1.92%	

Table 8: The second evaluation method that compared all possible product relationships with the data the Louvain method

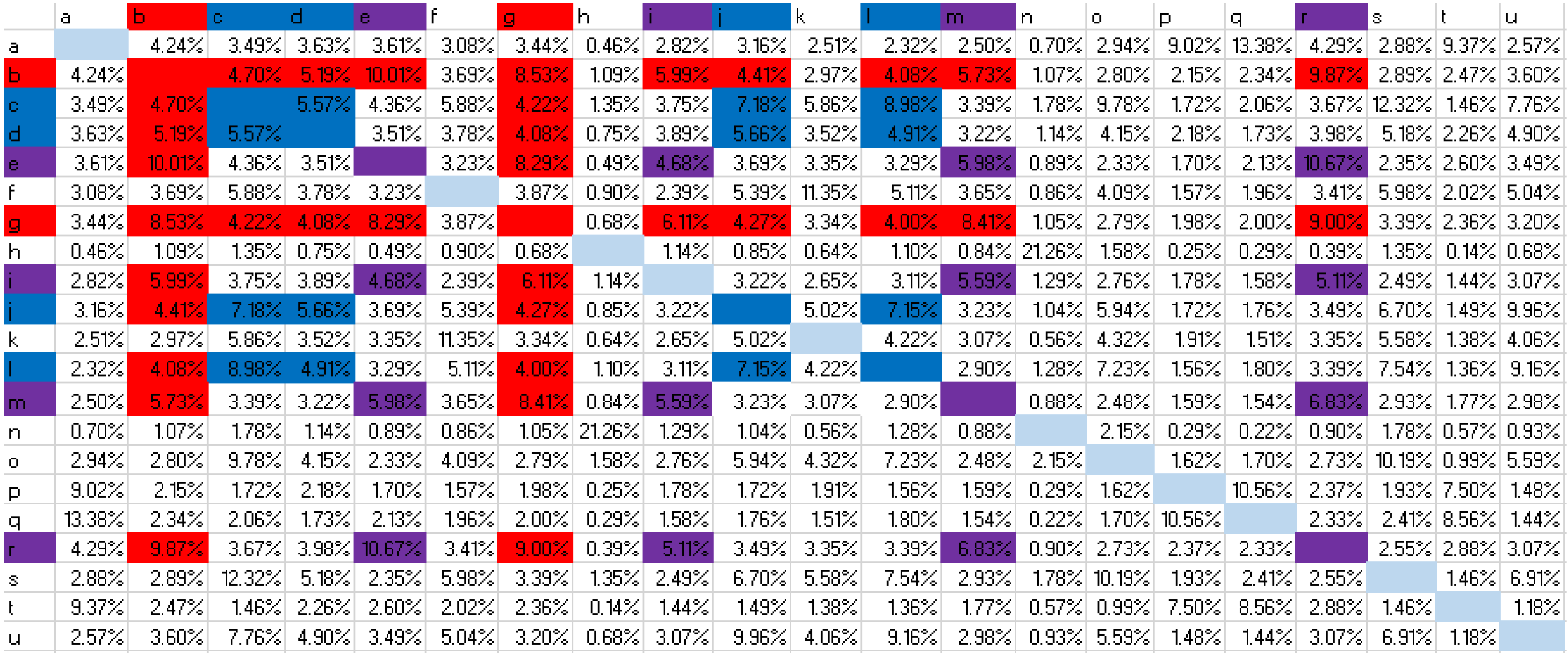


	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u
a	0.00%	3.45%	5.21%	5.32%	5.69%	3.07%	6.71%	7.54%	2.10%	0.89%	2.19%	2.45%	2.58%	2.62%	3.36%	6.74%	10.98%	1.69%	1.04%	6.39%	4.35%
b	3.45%	0.00%	3.92%	4.24%	8.77%	2.04%	8.54%	0.31%	7.66%	3.21%	1.44%	2.85%	10.07%	6.48%	1.78%	0.88%	1.06%	7.76%	2.55%	0.93%	1.93%
c	5.21%	3.92%	0.00%	16.17%	13.39%	12.73%	8.36%	22.65%	2.87%	2.16%	15.34%	9.68%	3.59%	6.83%	23.45%	3.05%	8.29%	1.49%	4.71%	2.45%	13.47%
d	5.32%	4.24%	16.17%	0.00%	37.44%	25.65%	7.97%	12.39%	2.90%	1.61%	9.03%	5.17%	3.33%	4.30%	65.93%	26.39%	6.85%	1.57%	1.88%	27.66%	8.32%
e	5.69%	8.77%	13.39%	37.44%	0.00%	23.28%	17.13%	8.65%	3.84%	1.28%	9.29%	3.79%	6.79%	3.64%	21.43%	12.55%	9.04%	4.79%	1.03%	18.44%	6.44%
f	3.07%	2.04%	12.73%	25.65%	23.28%	0.00%	5.64%	9.96%	1.01%	0.07%	18.97%	3.20%	2.14%	2.01%	22.62%	6.23%	5.26%	0.53%	0.27%	7.53%	5.30%
g	6.71%	8.54%	8.36%	7.97%	17.13%	5.64%	0.00%	0.64%	5.95%	2.03%	4.43%	5.40%	11.29%	4.99%	4.50%	2.35%	2.55%	5.06%	1.98%	2.66%	6.87%
h	7.54%	0.31%	22.65%	12.39%	8.65%	9.96%	0.64%	0.00%	0.11%	0.22%	6.22%	0.22%	0.11%	19.53%	20.13%	2.04%	2.57%	0.03%	0.37%	1.00%	0.28%
i	2.10%	7.66%	2.87%	2.90%	3.84%	1.01%	5.95%	0.11%	0.00%	1.01%	0.88%	1.88%	2.90%	0.17%	1.43%	0.44%	0.46%	4.73%	0.99%	0.29%	1.23%
j	0.89%	3.21%	2.16%	1.61%	1.28%	0.07%	2.03%	0.22%	1.01%	0.00%	0.28%	1.19%	0.31%	0.23%	0.59%	0.22%	0.15%	1.51%	0.17%	0.25%	0.01%
k	2.19%	1.44%	15.34%	9.03%	9.29%	18.97%	4.43%	6.22%	0.88%	0.28%	0.00%	2.18%	1.44%	1.10%	8.42%	2.27%	2.01%	0.30%	0.21%	1.47%	3.61%
l	2.45%	2.85%	9.68%	5.17%	3.79%	3.20%	5.40%	0.22%	1.88%	1.19%	2.18%	0.00%	2.42%	4.01%	5.48%	0.62%	0.83%	0.98%	1.68%	0.46%	5.50%
m	2.58%	10.07%	3.59%	3.33%	6.79%	2.14%	11.29%	0.11%	2.90%	0.31%	1.44%	2.42%	0.00%	0.16%	1.79%	0.55%	0.64%	1.55%	0.41%	0.51%	1.67%
n	2.62%	6.48%	6.83%	4.30%	3.64%	2.01%	4.99%	19.53%	0.17%	0.23%	1.10%	4.01%	0.16%	0.00%	6.01%	0.47%	0.38%	0.06%	0.43%	0.79%	2.09%
o	3.36%	1.78%	23.45%	65.93%	21.43%	22.62%	4.50%	20.13%	1.43%	0.59%	8.42%	5.48%	1.79%	6.01%	0.00%	15.67%	5.25%	0.62%	1.47%	99.01%	7.01%
p	6.74%	0.88%	3.05%	26.39%	12.55%	6.23%	2.35%	2.04%	0.44%	0.22%	2.27%	0.62%	0.55%	0.47%	15.67%	0.00%	21.05%	0.06%	0.24%	46.64%	1.05%
q	10.98%	1.06%	8.29%	6.85%	9.04%	5.26%	2.55%	2.57%	0.46%	0.15%	2.01%	0.83%	0.64%	0.38%	5.25%	21.05%	0.00%	0.15%	0.18%	15.55%	1.16%
r	1.69%	7.76%	1.49%	1.57%	4.79%	0.53%	5.06%	0.03%	4.73%	1.51%	0.30%	0.98%	1.55%	0.06%	0.62%	0.06%	0.15%	0.00%	1.37%	0.02%	0.43%
s	1.04%	2.55%	4.71%	1.88%	1.03%	0.27%	1.98%	0.37%	0.99%	0.17%	0.21%	1.68%	0.41%	0.43%	1.47%	0.24%	0.18%	1.37%	0.00%	0.24%	0.18%
t	6.39%	0.93%	2.45%	27.66%	18.44%	7.53%	2.66%	1.00%	0.29%	0.25%	1.47%	0.46%	0.51%	0.79%	99.01%	46.64%	15.55%	0.02%	0.24%	0.00%	0.74%
u	4.35%	1.93%	13.47%	8.32%	6.44%	5.30%	6.87%	0.28%	1.23%	0.01%	3.61%	5.50%	1.67%	2.09%	7.01%	1.05%	1.16%	0.43%	0.18%	0.74%	0.00%

Table 8: The second evaluation method that compared all possible product relationships with the data the Louvain method differences with the co-occurrences

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u
a	0.00%	3.45%	5.21%	5.32%	5.69%	3.07%	6.71%	7.54%	2.10%	0.89%	2.19%	2.45%	2.58%	2.62%	3.36%	6.74%	10.98%	1.69%	1.04%	6.39%	4.35%
b	3.45%	0.00%	3.92%	4.24%	8.77%	2.04%	8.54%	0.31%	7.66%	3.21%	1.44%	2.85%	10.07%	6.48%	1.78%	0.88%	1.06%	7.76%	2.55%	0.93%	1.93%
c	5.21%	3.92%	0.00%	16.17%	13.39%	12.73%	8.36%	22.65%	2.87%	2.16%	15.34%	9.68%	3.59%	6.83%	23.45%	3.05%	8.29%	1.49%	4.71%	2.45%	13.47%
d	5.32%	4.24%	16.17%	0.00%	37.44%	25.65%	7.97%	12.39%	2.90%	1.61%	9.03%	5.17%	3.33%	4.30%	65.93%	26.39%	6.85%	1.57%	1.88%	27.66%	8.32%
e	5.69%	8.77%	13.39%	37.44%	0.00%	23.28%	17.13%	8.65%	3.84%	1.28%	9.29%	3.79%	6.79%	3.64%	21.43%	12.55%	9.04%	4.79%	1.03%	18.44%	6.44%
f	3.07%	2.04%	12.73%	25.65%	23.28%	0.00%	5.64%	9.96%	1.01%	0.07%	18.97%	3.20%	2.14%	2.01%	22.62%	6.23%	5.26%	0.53%	0.27%	7.53%	5.30%
g	6.71%	8.54%	8.36%	7.97%	17.13%	5.64%	0.00%	0.64%	5.95%	2.03%	4.43%	5.40%	11.29%	4.99%	4.50%	2.35%	2.55%	5.06%	1.98%	2.66%	6.87%
h	7.54%	0.31%	22.65%	12.39%	8.65%	9.96%	0.64%	0.00%	0.11%	0.22%	6.22%	0.22%	0.11%	19.53%	20.13%	2.04%	2.57%	0.03%	0.37%	1.00%	0.28%
i	2.10%	7.66%	2.87%	2.90%	3.84%	1.01%	5.95%	0.11%	0.00%	1.01%	0.88%	1.88%	2.90%	0.17%	1.43%	0.44%	0.46%	4.73%	0.99%	0.29%	1.23%
j	0.89%	3.21%	2.16%	1.61%	1.28%	0.07%	2.03%	0.22%	1.01%	0.00%	0.28%	1.19%	0.31%	0.23%	0.59%	0.22%	0.15%	1.51%	0.17%	0.25%	0.01%
k	2.19%	1.44%	15.34%	9.03%	9.29%	18.97%	4.43%	6.22%	0.88%	0.28%	0.00%	2.18%	1.44%	1.10%	8.42%	2.27%	2.01%	0.30%	0.21%	1.47%	3.61%
l	2.45%	2.85%	9.68%	5.17%	3.79%	3.20%	5.40%	0.22%	1.88%	1.19%	2.18%	0.00%	2.42%	4.01%	5.48%	0.62%	0.83%	0.98%	1.68%	0.46%	5.50%
m	2.58%	10.07%	3.59%	3.33%	6.79%	2.14%	11.29%	0.11%	2.90%	0.31%	1.44%	2.42%	0.00%	0.16%	1.79%	0.55%	0.64%	1.55%	0.41%	0.51%	1.67%
n	2.62%	6.48%	6.83%	4.30%	3.64%	2.01%	4.99%	19.53%	0.17%	0.23%	1.10%	4.01%	0.16%	0.00%	6.01%	0.47%	0.38%	0.06%	0.43%	0.79%	2.09%
o	3.36%	1.78%	23.45%	65.93%	21.43%	22.62%	4.50%	20.13%	1.43%	0.59%	8.42%	5.48%	1.79%	6.01%	0.00%	15.67%	5.25%	0.62%	1.47%	99.01%	7.01%
p	6.74%	0.88%	3.05%	26.39%	12.55%	6.23%	2.35%	2.04%	0.44%	0.22%	2.27%	0.62%	0.55%	0.47%	15.67%	0.00%	21.05%	0.06%	0.24%	46.64%	1.05%
q	10.98%	1.06%	8.29%	6.85%	9.04%	5.26%	2.55%	2.57%	0.46%	0.15%	2.01%	0.83%	0.64%	0.38%	5.25%	21.05%	0.00%	0.15%	0.18%	15.55%	1.16%
r	1.69%	7.76%	1.49%	1.57%	4.79%	0.53%	5.06%	0.03%	4.73%	1.51%	0.30%	0.98%	1.55%	0.06%	0.62%	0.06%	0.15%	0.00%	1.37%	0.02%	0.43%
s	1.04%	2.55%	4.71%	1.88%	1.03%	0.27%	1.98%	0.37%	0.99%	0.17%	0.21%	1.68%	0.41%	0.43%	1.47%	0.24%	0.18%	1.37%	0.00%	0.24%	0.18%
t	6.39%	0.93%	2.45%	27.66%	18.44%	7.53%	2.66%	1.00%	0.29%	0.25%	1.47%	0.46%	0.51%	0.79%	99.01%	46.64%	15.55%	0.02%	0.24%	0.00%	0.74%
u	4.35%	1.93%	13.47%	8.32%	6.44%	5.30%	6.87%	0.28%	1.23%	0.01%	3.61%	5.50%	1.67%	2.09%	7.01%	1.05%	1.16%	0.43%	0.18%	0.74%	0.00%

Table 9: The second evaluation method that compared all possible product relationships with the data the Louvain method differences with the co-occurrences



	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u
a		4.24%	3.49%	3.63%	3.61%	3.08%	3.44%	0.46%	2.82%	3.16%	2.51%	2.32%	2.50%	0.70%	2.94%	9.02%	13.38%	4.29%	2.88%	9.37%	2.57%
b	4.24%		4.70%	5.19%	10.01%	3.69%	8.53%	1.09%	5.99%	4.41%	2.97%	4.08%	5.73%	1.07%	2.80%	2.15%	2.34%	9.87%	2.89%	2.47%	3.60%
c	3.49%	4.70%		5.57%	4.36%	5.88%	4.22%	1.35%	3.75%	7.18%	5.86%	8.98%	3.39%	1.78%	9.78%	1.72%	2.06%	3.67%	12.32%	1.46%	7.76%
d	3.63%	5.19%	5.57%		3.51%	3.78%	4.08%	0.75%	3.89%	5.66%	3.52%	4.91%	3.22%	1.14%	4.15%	2.18%	1.73%	3.98%	5.18%	2.26%	4.90%
e	3.61%	10.01%	4.36%	3.51%		3.23%	8.29%	0.49%	4.68%	3.69%	3.35%	3.29%	5.98%	0.89%	2.33%	1.70%	2.13%	10.67%	2.35%	2.60%	3.49%
f	3.08%	3.69%	5.88%	3.78%	3.23%		3.87%	0.90%	2.39%	5.39%	11.35%	5.11%	3.65%	0.86%	4.09%	1.57%	1.96%	3.41%	5.98%	2.02%	5.04%
g	3.44%	8.53%	4.22%	4.08%	8.29%	3.87%		0.68%	6.11%	4.27%	3.34%	4.00%	8.41%	1.05%	2.79%	1.98%	2.00%	9.00%	3.39%	2.36%	3.20%
h	0.46%	1.09%	1.35%	0.75%	0.49%	0.90%	0.68%		1.14%	0.85%	0.64%	1.10%	0.84%	21.26%	1.58%	0.25%	0.29%	0.39%	1.35%	0.14%	0.68%
i	2.82%	5.99%	3.75%	3.89%	4.68%	2.39%	6.11%	1.14%		3.22%	2.65%	3.11%	5.59%	1.29%	2.76%	1.78%	1.58%	5.11%	2.49%	1.44%	3.07%
j	3.16%	4.41%	7.18%	5.66%	3.69%	5.39%	4.27%	0.85%	3.22%		5.02%	7.15%	3.23%	1.04%	5.94%	1.72%	1.76%	3.49%	6.70%	1.49%	9.96%
k	2.51%	2.97%	5.86%	3.52%	3.35%	11.35%	3.34%	0.64%	2.65%	5.02%		4.22%	3.07%	0.56%	4.32%	1.91%	1.51%	3.35%	5.58%	1.38%	4.06%
l	2.32%	4.08%	8.98%	4.91%	3.29%	5.11%	4.00%	1.10%	3.11%	7.15%	4.22%		2.90%	1.28%	7.23%	1.56%	1.80%	3.39%	7.54%	1.36%	9.16%
m	2.50%	5.73%	3.39%	3.22%	5.98%	3.65%	8.41%	0.84%	5.59%	3.23%	3.07%	2.90%		0.88%	2.48%	1.59%	1.54%	6.83%	2.93%	1.77%	2.98%
n	0.70%	1.07%	1.78%	1.14%	0.89%	0.86%	1.05%	21.26%	1.29%	1.04%	0.56%	1.28%	0.88%		2.15%	0.29%	0.22%	0.90%	1.78%	0.57%	0.93%
o	2.94%	2.80%	9.78%	4.15%	2.33%	4.09%	2.79%	1.58%	2.76%	5.94%	4.32%	7.23%	2.48%	2.15%		1.62%	1.70%	2.73%	10.19%	0.99%	5.59%
p	9.02%	2.15%	1.72%	2.18%	1.70%	1.57%	1.98%	0.25%	1.78%	1.72%	1.91%	1.56%	1.59%	0.29%	1.62%		10.56%	2.37%	1.93%	7.50%	1.48%
q	13.38%	2.34%	2.06%	1.73%	2.13%	1.96%	2.00%	0.29%	1.58%	1.76%	1.51%	1.80%	1.54%	0.22%	1.70%	10.56%		2.33%	2.41%	8.56%	1.44%
r	4.29%	9.87%	3.67%	3.98%	10.67%	3.41%	9.00%	0.39%	5.11%	3.49%	3.35%	3.39%	6.83%	0.90%	2.73%	2.37%	2.33%		2.55%	2.88%	3.07%
s	2.88%	2.89%	12.32%	5.18%	2.35%	5.98%	3.39%	1.35%	2.49%	6.70%	5.58%	7.54%	2.93%	1.78%	10.19%	1.93%	2.41%	2.55%		1.46%	6.91%
t	9.37%	2.47%	1.46%	2.26%	2.60%	2.02%	2.36%	0.14%	1.44%	1.49%	1.38%	1.36%	1.77%	0.57%	0.99%	7.50%	8.56%	2.88%	1.46%		1.18%
u	2.57%	3.60%	7.76%	4.90%	3.49%	5.04%	3.20%	0.68%	3.07%	9.96%	4.06%	9.16%	2.98%	0.93%	5.59%	1.48%	1.44%	3.07%	6.91%	1.18%	

Table 10: The first evaluation method that compared product relationships of group members CPM

Community B: highest co-occurrence percentages products in the real dataset

	b		c		d		g		j		l
e	10.01%	s	12.32%	j	5.66%	r	9.00%	u	9.96%	u	9.16%
r	9.87%	o	9.78%	c	5.57%	b	8.53%	c	7.18%	c	8.98%
g	8.53%	l	8.98%	b	5.19%	m	8.41%	l	7.15%	s	7.54%
i	5.99%	u	7.76%	s	5.18%	e	8.29%	s	6.70%	o	7.23%
m	5.73%	j	7.18%	l	4.91%	i	6.11%	o	5.94%	j	7.15%
d	5.19%	f	5.88%	u	4.90%	j	4.27%	d	5.66%	f	5.11%
c	4.70%	k	5.86%	o	4.15%	c	4.22%	f	5.39%	d	4.91%
j	4.41%	d	5.57%	g	4.08%	d	4.08%	k	5.02%	k	4.22%
a	4.24%	b	4.70%			l	4.00%	b	4.41%	b	4.08%
l	4.08%	e	4.36%					g	4.27%	g	4.00%
		g	4.22%								

Community P: highest co-occurrence percentages products in the real dataset

	b		e		g		i		m		r
e	10.01%	r	10.67%	r	9.00%	g	6.11%	g	8.41%	e	10.67%
r	9.87%	b	10.01%	b	8.53%	b	5.99%	r	6.83%	b	9.87%
g	8.53%	g	8.29%	m	8.41%	m	5.59%	e	5.98%	g	9.00%
i	5.99%	m	5.98%	e	8.29%	r	5.11%	b	5.73%	m	6.83%
m	5.73%	i	4.68%	i	6.11%	e	4.68%	i	5.59%	i	5.11%

Table 11: The first evaluation method that compared product relationships of group members CPM