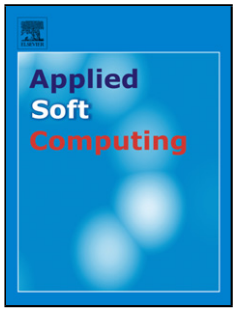


Title: Clustering Retail Products Based on Customer Behaviour

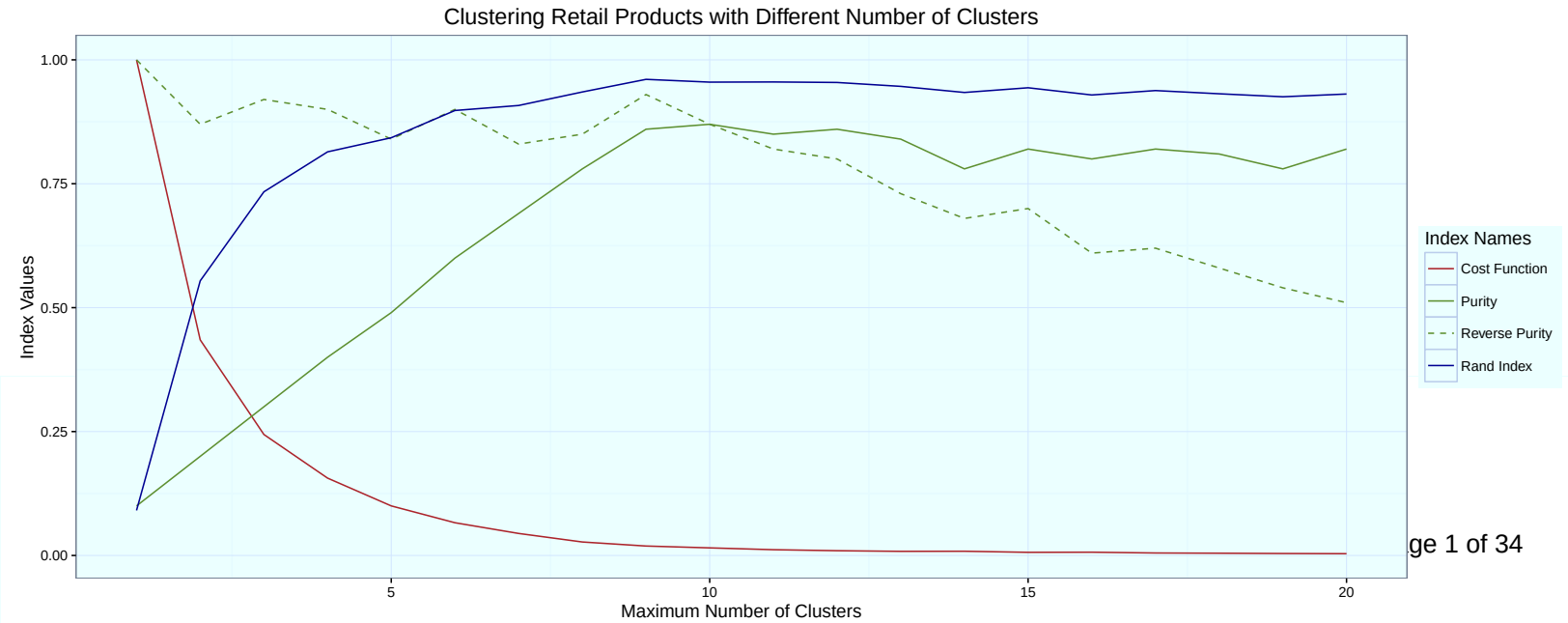
Author: Vladimír Holý Ondřej Sokol Michal Černý

PII: S1568-4946(17)30072-8
DOI: http://dx.doi.org/doi:10.1016/j.asoc.2017.02.004
Reference: ASOC 4051

To appear in: *Applied Soft Computing*

Received date: 15-3-2016
Revised date: 7-10-2016
Accepted date: 2-2-2017

Please cite this article as: Vladimír Holý, Ondřej Sokol, Michal Černý, Clustering Retail Products Based on Customer Behaviour, *<![CDATA[Applied Soft Computing Journal]]>* (2017), http://dx.doi.org/10.1016/j.asoc.2017.02.004



Clustering Retail Products Based on Customer Behaviour

Highlights

- Drugstore market basket data are analyzed.
- A purely data driven method for clustering products is proposed.
- A genetic algorithm is used.
- Interesting subcategories of drugstore products are found.

Clustering Retail Products Based on Customer Behaviour

Vladimír Holý[1]

University of Economics, Prague
Winston Churchill Square 4, 130 67 Prague 3, Czech Republic
vladimir.holy@vse.cz

Ondřej Sokol

University of Economics, Prague
Winston Churchill Square 4, 130 67 Prague 3, Czech Republic
ondrej.sokol@vse.cz

Michal Černý

University of Economics, Prague
Winston Churchill Square 4, 130 67 Prague 3, Czech Republic
cernym@vse.cz

Abstract

The categorization of retail products is essential for the business decision-making process. It is a common practice to classify products based on their quantitative and qualitative characteristics. In this paper, we use a purely data-driven approach. Our clustering of products is based exclusively on the customer behaviour. We propose a method for clustering retail products using market basket data. Our model is formulated as an optimization problem which is solved by a genetic algorithm. It is demonstrated on simulated data how our method behaves in different settings. The application using real data from a Czech drugstore company shows that our method leads to similar results in comparison with the classification by experts. The number of clusters is a parameter of our algorithm. We demonstrate that if more clusters are allowed than the original number of categories is, the method yields additional information about the structure of the product categorization.

[1]Corresponding author

Keywords: product categorization, cluster analysis, genetic algorithm, retail business, drugstore market

1. Introduction

Categorization is important in the retail business decision-making process. Product classification and customer segmentation belong to the most frequently used methods. The customer segmentation is focused on getting knowledge about the structure of customers and is used for targeted marketing. For example [9] dealt with customer segmentation and its usability in marketing. Another approach to determining customer segmentation was used by [12]. Customer segmentation based on self-organizing maps with a priori prioritization in direct marketing was proposed in [19].

The product categorization finds even more applications in marketing, e.g. new product development, optimizing placement of retail products on shelves, analysis of cannibalization between products and more general analysis of the affinity between products. A genetic algorithm to identify optimal new product position was proposed in [5]. A placement of retail products on shelves was studied by [3]. Finding the right categories is also crucial for sales promotions planning. Cross-category sales promotion effect was studied in detail by [11] and [6].

Retail chains try to minimize costs everywhere. Among others, their aim is to minimize the costs of product storage in stores. The storage management reaches the stage when stores often have no reserves in the drugstore storeroom because they are supplied dynamically two or more times per week. Therefore, it may happen that a store runs out of some products. The task is:

1. How to fill a free place on shelves until the storage is restored.
2. How to find a product that best substitutes for the original one.

Sold-out products are usually replaced by other ones from the same category, but it is not clear how to best define the categories from this viewpoint. This is the main business motivation behind this paper.

Products are almost always categorized according to their purpose, package properties, e.g. package size, brand and price level. However, there are different approaches to product categorization. For example [20] used hierarchical clustering while [24] promoted fuzzy clustering. Another interesting possibilistic approach to clustering both customers and products was published by [2].

Retail chains have available huge amount of market basket data, containing sets of items that a buyer acquires in one purchase, which can be used to efficiently model customer behaviour, e.g. [21]. However, these data are rarely taken into account in the product categorization. Data from market baskets are usually used for analysis of cross-category dependence for a priori given categories, e.g. [18], [15], [4] and [13].

This paper proposes a new method for choosing categories utilizing market basket data. Our method classifies products into clusters according to their common occurrences in the shopping baskets. Sets of products in individual shopping baskets as they were registered by the receipts are the only data used by the method which assigns each product to just one category. The method determines product categories under given assumptions of product dependency in the same category. It stems from the assumption that a customer buys only one product per category. Experience shows that customers who buy one product from a given category are generally less likely to buy also another product from the same category. The method applies a genetic algorithm to market basket data to find the best clusters of products based on their joint occurrence in shopping baskets.

Retail companies usually inspect affinity relationship between single products, e.g. sales in the same basket normalised by total sales. However, clustering of products based solely on market basket data in this area is not so common. It can help mainly in organising shelf and/or maximising effect of promotional activities such as newsletter promotions with a significant discount. This kind of promotion should attract customers who do not regularly visit the store. For example, the promotion of two products from the same category is not effective as customer usually buys only one of them. The interesting article focused on

marketing strategies of associated products was published in [22] in which author deals with the problem of association rules when the product is marketed later. Our method for clustering may be helpful mainly in markets with the high proportion of sales in a promotion, such as Czech drugstore market where over half of sales is in the promotion.

The overview of methods for automatic clustering using nature-based metaheuristic methods such as genetic algorithms or swarm intelligence can be found in [10]. Interesting approach using fuzzy chromosome in a genetic algorithm was published by [23]. Another possibilistic approach was presented by [1]. The combination of k-means and ant colony optimization was published in [16]. Clustering method k-means is usually taken as a base method and although k-means was proposed over 50 years ago, it is still one of the most used methods. The overview of k-means and its modification can be found in [7].

The objective of this paper is to present a new method for retail product clustering based on shopping behaviour of customers. The goal of analysis using market basket data is usually finding *complements*, e.g. finding which products are often bought together. Our approach is completely different. Based on market basket data, we are clustering products into clusters of *substitutes*. Therefore, in one cluster we put together products which rarely occur in the same shopping basket.

The resulting categorization can be used not only for choosing the products suitable to replace sold-out ones but also for optimizing placement of retail products on shelves or for maximizing profit of sales promotions. To maximize the profit, it is more effective to spread promotion across different categories instead of stacking multiple promotions in the same category. The resulting clustering can also help in persuading customer into buying more expensive alternative, e.g. promotion which includes a discount on the more expensive product when a product from the same category is bought.

The article is organized as follows: In Section 2, we present the general idea of the proposed method and its assumptions. In Section 3, we test the method using synthetic data to illustrate its performance. We also show how

the violation of the assumptions affects the method's results. In Section 4 the application to drugstore's market basket is presented and its potential to detect clusters in real data sets which have not been found before is demonstrated. We also present the comparison with other methods. The paper concludes with a summary in Section 5.

2. Methods

In this section, we propose a new method for clustering retail products based on customer behaviour. We also present our approach to evaluate resulting clustering.

To clarify terminology in this article we use *categories* meaning the original product category that was defined expertly based on the character and the purpose of the products. On the other hand, *clusters* are results of our method. Clusters are determined using only market basket data.

2.1. Clustering Using Genetic Algorithm

We formulate clustering of retail products as an optimization problem. The goal is to find a clustering that minimizes the number of products within the same cluster in one shopping basket. It is based on the idea that in general customers will not buy more than one product from each cluster (products in clusters are similar so they need only one). We say that customers that buy at most one product from each cluster follow the *ideal behaviour* (IB). We define a cost function which penalizes a violation of this ideal behaviour. For a given clustering the cost function calculates a weighted number of violations of the assumption that in each basket there is at most one product from a cluster.

We approach clustering as a series of decisions. For each pair of products, there is a decision whether these two products should be in the same cluster or not. It is inspired by the Rand index (formulated later in Subsection 2.2). Specifically, we minimize the average ratio of incorrect clustering decisions. Here, incorrect is meant in the sense that they lead to multiple products within the same cluster in one shopping basket.

Let n_B be the number of baskets, n_P the number of products and n_C the maximum number of clusters. We define matrix $\boldsymbol{A}$ with n_B rows, n_P columns and elements $a_{i,j}$ as

$$a_{i,j} = \begin{cases} 1 & \text{if product } j \text{ is present in basket } i, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

A possible clustering is defined as $\boldsymbol{x} = (x_1, \ldots, x_{n_P})'$, where x_j is an integer and $1 \leq x_j \leq n_C$. Elements of vector $\boldsymbol{x}$ correspond to products and their values represent assignment of a product to a cluster.

For each basket $b = 1, \ldots, n_B$ we calculate the total number of decisions D_b as

$$D_b = \binom{d_b}{2}, \qquad d_b = \sum_{j=1}^{n_P} a_{b,j} \tag{2}$$

and the number of decisions that lead to multiple products within the same cluster V_b as

$$V_b(\boldsymbol{x}) = \sum_{c:v_{b,c}(\boldsymbol{x})>1} \binom{v_{b,c}(\boldsymbol{x})}{2}, \qquad v_{b,c}(\boldsymbol{x}) = \sum_{j:x_j=c} a_{b,j}. \tag{3}$$

The number of violating decisions V_b is dependent on clustering $\boldsymbol{x}$. Finally, we define the cost function as

$$f_{cost}(\boldsymbol{x}) = \frac{1}{n_B} \sum_{b=1}^{n_B} \frac{V_b(\boldsymbol{x})}{D_b}. \tag{4}$$

Hence, the cost function equals to the average ratio of decisions in which two products from the same cluster are in the same shopping basket. The range of the cost function is from 0 to 1. If there is no basket containing products from the same cluster, then the cost function is 0. On the other hand, if every basket contain only products from the same cluster, then the cost function is 1.

We have considered several other objective function formulations, most notably:

- the ratio of total number of multiple products within the same cluster,
- the average ratio of multiple products within the same cluster over all baskets,

- the average ratio of multiple products within the same cluster over all products,
- the ratio of total number of baskets with multiple products within the same cluster,

but the best clusterings were given by the objective function (4) inspired by Rand index, which was presented in [17].

The whole optimization problem is formulated as

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f_{cost}(\boldsymbol{x}) \\ \text{s. t.} \quad & x_i \leq n_C \quad \text{for } i = 1, \ldots, n_P, \\ & x_i \in \mathbb{N} \quad \text{for } i = 1, \ldots, n_P. \end{aligned} \tag{5}$$

The cost function (4) is minimized over all possible clusterings $\boldsymbol{x}$. The maximum number of clusters n_C is a fixed number. If we did not limit the number of clusters, each product would be assigned to its own cluster. Optimization problem (5) is an integer non-linear programming, which can easily be shown to be NP-hard. To efficiently solve this or at least to get approximate solution we use heuristic *genetic algorithm*. Details of our genetic algorithm parameters are discussed in Subsection 3.1. Genetic algorithm for solving an integer non-linear program was already used by [8] and more recently by [23].

To ensure that resulting clustering is meaningful we need to make following assumptions:

(A1) The probability that a customer buys at least two products from one category (i.e. a customer does not follow IB) is strictly less than 50 %.

(A2) The true number of clusters is known.

(A3) Each customer has the same nonzero probability of buying a specific product and the probability is constant in time.

Assumption (A1) tells us that although our model allows customers to buy more than one product within the same cluster in one shopping basket, this behaviour is not considered standard but as a model error. Assumption (A2)

reflects formulation of our optimization problem in which we specify the maximum number of clusters. In almost all cases the resulting number of clusters is the maximum number of clusters (more clusters are preferred by the objective function). Finally, Assumption (A3) is meant to prevent situations in which some customers buy product A and never product B while other customers buy product B and never product A. This behaviour would result in assigning products A and B into the same cluster even if they are completely different. Assumption (A3) ensures that with a large enough dataset, all combinations of products will appear in some baskets with probability approaching 1. Later, in Section 3 we discuss in more detail how violation of these assumptions and the ideal behaviour would affect the resulting clustering.

2.2. Evaluation of Clustering

We use three different statistics to evaluate our resulting clustering when true categories are known. The first one is *purity* used for example by [14]. It is computed in a very straightforward way. Each estimated cluster is assigned a category which is the most frequent in the cluster. Purity is then the ratio of products with correctly assigned categories. It is calculated as

$$I_{PUR} = \frac{1}{n_P} \sum_{i=1}^{n_C} \max_j |C_i \cap R_j|, \tag{6}$$

where n_P is the number of products, n_C is the number of estimated clusters, C_i is the set of products in estimated cluster i and R_j is the set of products in real category j. Bad clusterings have the purity close to 0, the perfect clustering has purity equal to 1. However, if the number of estimated clusters is much larger than the number of real categories the purity always has a high value and in this case it is not very meaningful.

For this reason we also use a modification of the purity in which we reverse the role of true categories and estimated clusters. *Reverse purity* is defined as

$$I_{REV} = \frac{1}{n_P} \sum_{j=1}^{n_R} \max_i |C_i \cap R_j|, \tag{7}$$

where n_R is the number of real categories.

The last statistic we use is the *Rand index* proposed by [17]. It is based on the idea that clustering is a series of decisions that either put two products into the same cluster or put them to different clusters. Therefore the number of decisions is the number of product pairs. Rand index is defined as a ratio of correct decisions and is calculated as

$$I_{RAND} = \frac{P_{TP} + P_{TN}}{P}, \tag{8}$$

where P_{TP} is the number of pairs correctly assigned to the same cluster, P_{TN} is the number of pairs correctly assigned to different clusters and P is the number of all pairs. Accurate clusterings have Rand index close to 1.

3. Simulation Study

To reveal properties of our method we perform several simulations. In Subsection 3.1 we compare different parameters of our genetic algorithm. In three remaining subsections, we simulate a violation of the ideal behaviour (IB) and assumptions (A2) and (A3). In all simulations we consider a dataset consisting of 10 000 shopping baskets, each with 4 categories that have one or two products. We have a total of 10 categories with 10 products each. We simulate the behaviour of a customer who selects a set of categories (s)he needs (sets of 4 categories are selected with the same probability, apart from Subsection 3.4). Then he selects which products from that category he wants to buy (products are selected with the same probability and there is also a 10% chance to buy two products instead of one). We have chosen these characteristics of simulated data because they are similar to the size and the structure of the real dataset we use later in Section 4.

3.1. Choosing Genetic Algorithm Parameters

To properly use genetic algorithm we need to choose several parameters. The first one is the *size of population*. With a bigger population of individuals more possible clusterings are explored, which can result in finding a better solution. We set the population size to 500 individuals due to computational complexity.

The initial population is generated randomly and then the algorithm iteratively selects new populations. The number of iterations is another parameter called the *number of generations*. We always use the fixed number of 1 000 generations. However, our simulations show that most clusterings converge much faster. Each candidate solution is represented by an individual. Properties of candidate solutions are encoded in chromosomes of individuals. At each generation a percentage of individuals with lowest values of the cost function (called the elite population) passes to the next generation without any alteration. We consider the *ratio of elite population* from 0 to 0.2. In our case elite population does not have a big impact on the best individual in the last generation, all values result in perfect or almost perfect clustering. It is meaningful to carry over at least the best individual from the previous generation so the quality of solution will not decrease. We set the elite ratio to 0.1.

Chromosomes of the rest of the new individuals are generated by crossover and mutation operations. For each new individual (called the child) crossover selects two individuals from the last generation (called the parents). The parents are selected randomly with weights according to the sorting by their cost function. The child is then created by combining chromosomes of both parents. We use one-point crossover which means that a random number of chromosomes c is selected. Then the first c chromosomes of the child are taken from one parent while the remaining chromosomes are taken from the other parent. Finally, the mutation operation is performed. Each chromosome of the child has a probability of changing its value to a random one. This probability is the last parameter called the *mutation chance*. We consider this parameter to be from 0 to 0.2. Figure 1 shows that positive mutation chances up to 0.04 result in perfect or almost perfect clustering. Simulation with no mutation chance gives a far worse result showing the importance of mutation. We set the mutation chance to 0.01 to allow the algorithm to concentrate on improving one point while retaining some exploratory ability of mutation.

Next, we analyze combinations of the mentioned parameters. In Figure 2 cost functions of the best individuals in each generation are shown for 6

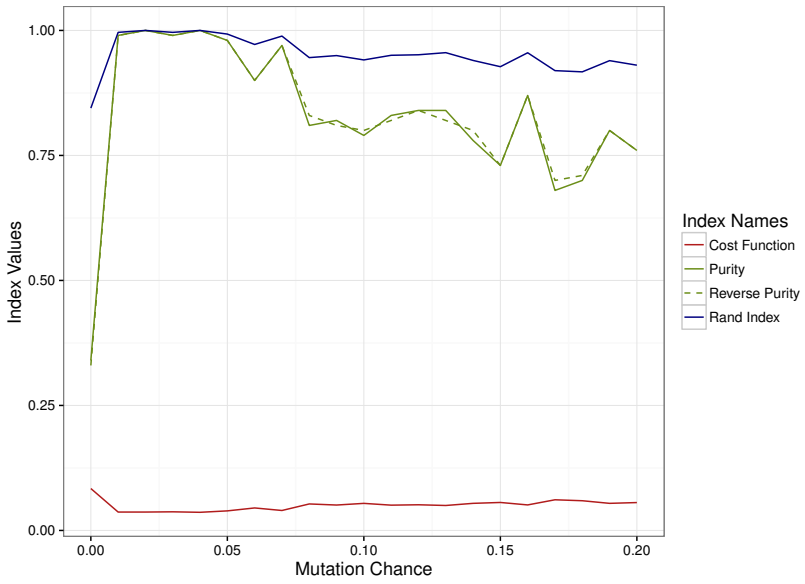


Figure 1: Evaluation statistics for genetic algorithm with different mutation chances.

different parameter settings. We combine population sizes $P = (50, 200, 500)$ with mutation chances $M = (0.01, 0.1)$ while the ratio of elite population is set to 0.1. In Table 1 several statistics of the resulting clustering are presented. We can see that lower mutation chance leads to faster convergence. There is a risk of lower mutation chance to end up in a local minimum but the results show this is not the case. The population size of 500 individuals with mutation chance of 1% resulted in a perfect clustering. In the rest of the paper, these are the genetic algorithm parameters we use.

3.2. Multiple Products Within the Same Category in One Shopping Basket

In this subsection, we study the sensitivity of the proposed method to the situation, in which customers buy more than one product from the same category. We simulate data for different probabilities of buying the second product. Results are shown in Figure 3. As we can see the method gives almost perfect clustering for the probability of the second product up to 0.18. At probability 0.20 there is a significant decrease in accuracy. This is caused by the loss

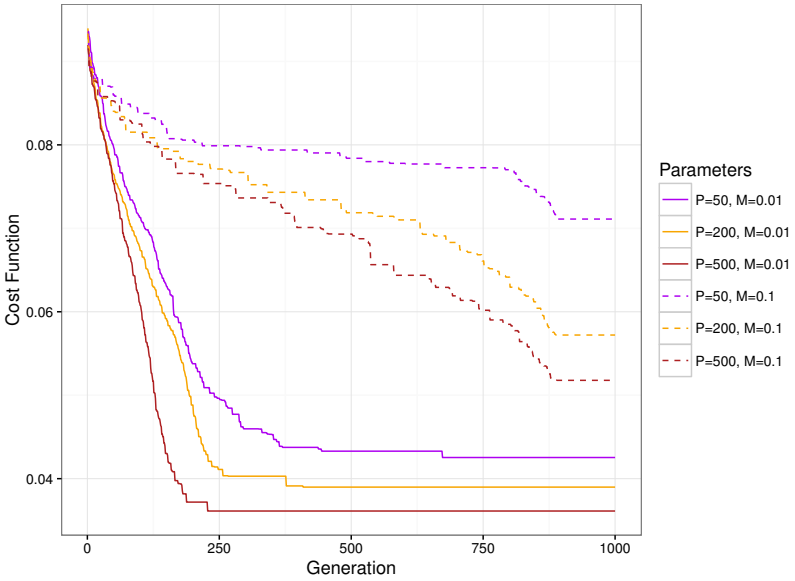


Figure 2: Cost function of best individuals in each generation for different population sizes P and mutation chances M.

Population Size	50	200	500	50	200	500
Mutation Chance	0.01	0.01	0.01	0.1	0.1	0.1
Cost Function	0.043	0.039	0.036	0.071	0.057	0.052
Purity	0.940	0.980	1.000	0.550	0.740	0.081
Reverse purity	0.940	0.980	1.000	0.550	0.750	0.081
Rand index	0.978	0.993	1.000	0.886	0.930	0.946

Table 1: Statistics of the best individual in the final generation for different population sizes P and mutation chances M.

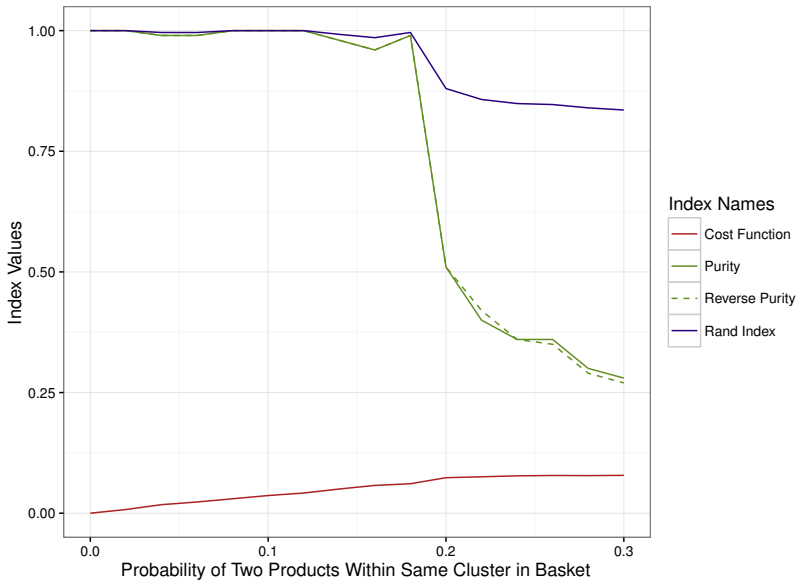


Figure 3: Evaluation statistics for clustering data with different probabilities of second product in the same category in one shopping basket.

of relevant information contained in shopping basket data supplied to the cost function. If we could increase the number of observed shopping baskets or the average number of products in a shopping basket we would get more precise results even for the second product probability of 0.20 or higher.

3.3. Unknown Number of Clusters

We have assumed in our simulations so far that the true number of categories is known. Now we inspect the behaviour of our method when used with different numbers of clusters. Results are shown in Figure 4. The question is if we can identify the correct number of categories. In Figure 4 we can see that the purity, the reverse purity and the Rand index have a value of 1 for 10 clusters, indicating the perfect clustering. However, in a real application we do not know the true categorization and therefore we cannot calculate the purity statistics or the Rand index. A way to determine what number of clusters should be used is to analyse the shape of the cost function. For a number of clusters

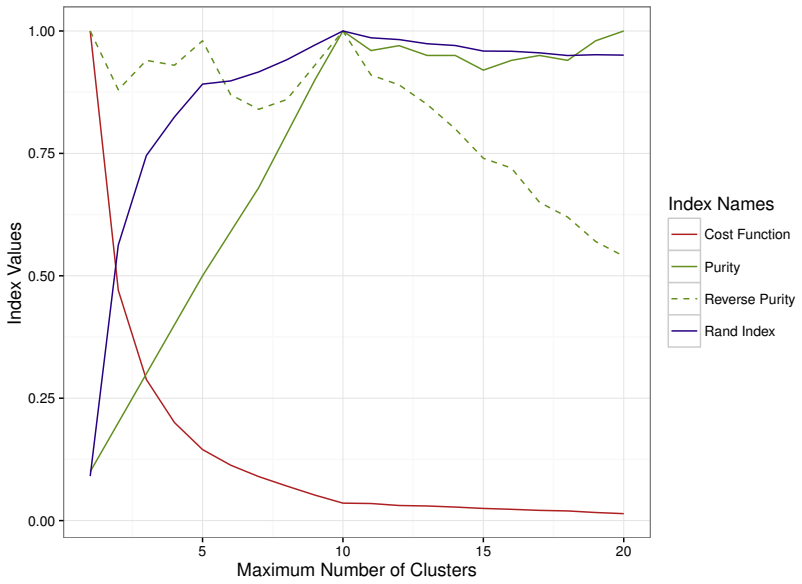


Figure 4: Evaluation statistics for clustering when number of categories is unknown.

less than the true number the cost function is significantly decreasing with more clusters. When the true number of categories has been reached the cost function continues to decrease only by a small amount. As we can see in Figure 4 the cost function stops rapidly decreasing around 10 clusters which is the true number of categories.

3.4. Different Types of Customers

Finally, we discuss a violation of Assumption (A3). We consider three types of customers. Customer A can buy products from all categories with equal probability. Customer B can buy products only from a half of categories while customer C can buy products only from the other half of categories. We study the behaviour of the proposed method for customer structures ranging from all customers being of type A (this was the case of all previous simulations) to half customers being type B and half type C. Results are shown in Figure 5. If the customers violating assumption (A3) are in the minority the resulting clustering is not affected. However, from the point where customer composition is 50%

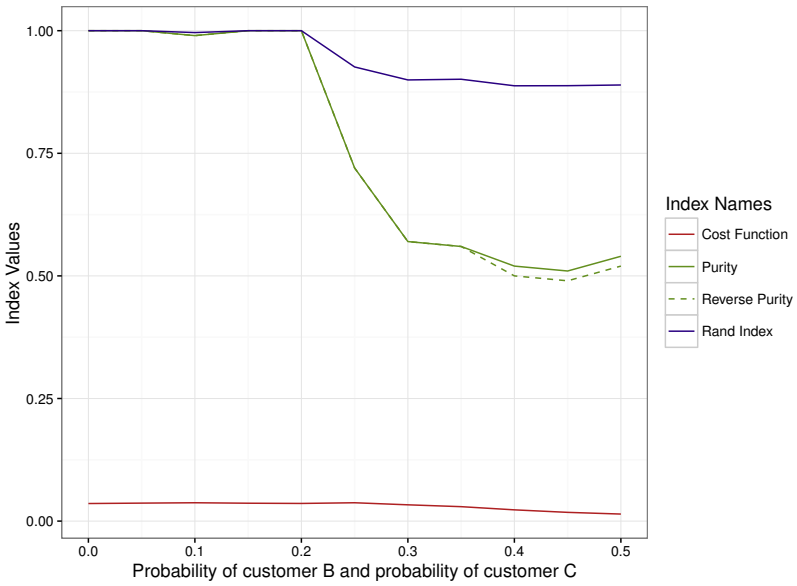


Figure 5: Evaluation statistics for clustering data with different probabilities of occurrence of customers B and C.

type A, 25% type B and 25% type C the resulting clustering becomes quite chaotic.

4. Data Analysis

In this section, we use our method with a sample of real data. Our dataset consists of individual purchase data of one of the retail chains in drugstore market in the Czech Republic. We take original categories applied in the retail chain that are defined expertly, according to the character and the purpose of the products, as a reference classification.

Customer behaviour in the Czech drugstore market is specific. As there is a high density of malls and hypermarkets, customers who visit drugstores usually buy just a few products. The average number of items in a shopping basket in the drugstore is around 3.

4.1. Description of Dataset

In this paper we use a sample of receipts containing at least 4 products from the whole year 2015. The size of our dataset is 10 608 baskets containing 10 best-selling products from 10 most-popular categories defined by the drugstore. Original categories are based mostly on product purpose and price level. Thus, we have 100 different products. The sample is mainly for illustration how our method work and how the clusters are defined – that is the reason why we have chosen exactly 100 products. Parameters of the genetic algorithm are the same as presented in Section 3.1.

To clarify the terminology in this section, original categories are denoted by letters while clusters found by our method are denoted by numbers.

4.2. Model with 10 Clusters

We applied the proposed method to the maximum number of 10 clusters and we compared the found clusters with the original categories. Using the real dataset we have found that the ideal behaviour (IB) was violated in approximately 19% of baskets on average. The ratios of violations significantly differ for each category as shown in Table 2.

The assignment of products to clusters is shown in Table 3. It is apparent that the proposed method had a problem with assigning products from categories *C, D* and *E*. Those are the categories with the highest percentage of violations of the ideal behaviour (IB).

The method produced 10 clusters which was the maximum allowed. Evaluation statistics of the results of this test are shown in Table 4. Purity as well as reverse purity statistics show that some of the products were not assigned as in original categorization. The cost function value of the assignment is 0.0153 which is lower than the value of the expert estimate assignment which is 0.0182. The reason is that the method minimizes the number of products bought together within one cluster. Therefore, if a category suffers from a violation of the ideal behaviour (IB), our method puts together products from different original categories into one cluster to minimize the cost function.

Category	Name	Occurences	Violation ratio
A	Dishwashing liquid	4387	0.022
B	WC liquid cleaners	4212	0.047
C	Handkerchieves and napkins	5769	0.077
D	Soap	2026	0.179
E	Tampons	1837	0.104
F	Toilet paper	6993	0.020
G	Trash bags	4543	0.068
H	Paper towels	4544	0.012
I	Cotton wool and cotton buds	3991	0.026
J	Facial pads	5224	0.012

Table 2: Violations of the ideal behaviour (IB) for each category.

Using our method we have found out that categories *C, D* and *E* are perceived differently by the customers and by the management. This finding can be further used in designing new product categorization or in defining subcategories.

4.3. Model with 8 Clusters

In our next test, we assign the same 100 products of 10 categories into 8 clusters. The results are shown in Table 5. There is a good correspondence between 8 clusters and 8 categories, *A, B, C, F, G, H, I* and *J*. Products from the *problematic* categories *D* and *E* were assigned quite randomly to clusters 2 to 8. Evaluation statistics of this model are in Table 6. Results confirmed that categories *A, B, C, F, G, H, I* and *J* are perceived similarly by the customers and by the managers. As expected, purity statistic is lower than in the test of Section 3 with more clusters and the cost function has a higher value. Purity has to be lower as the size of categories is generally larger than the size of clusters if the cost function is minimized.

Original categories									
A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*
7	6	8	3	5	1	4	3	9	10
7	5	2	2	5	1	4	3	9	10
7	6	8	2	5	1	4	3	9	10
7	6	8	5	6	1	4	3	9	10
7	6	2	5	5	1	4	3	9	10
7	6	8	2	5	1	5	3	9	10
7	6	8	2	7	1	4	3	9	10
7	6	8	2	5	1	4	3	9	10
7	6	8	5	5	1	4	5	9	10
7	6	2	2	5	1	4	3	5	10

Table 3: Assignment of 100 products from original categories *A*-*J* to clusters 1–10.

Number of classes	10
Purity	0.870
Reverse purity	0.870
Rand index	0.955
Cost function value	0.0153

Table 4: Evaluation statistics for model with 10 clusters.

Original categories									
A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*
8	5	2	4	3	1	7	3	6	4
8	5	2	6	8	1	7	3	6	4
8	5	2	3	5	1	7	3	6	4
8	5	2	2	3	1	7	3	6	4
8	5	2	5	8	1	7	3	6	4
8	5	2	3	5	1	7	3	6	4
8	5	2	6	8	1	7	3	6	4
8	5	2	4	5	1	7	3	6	4
8	5	2	2	7	1	7	4	5	4
8	5	5	6	7	1	7	3	6	4

Table 5: Assignment of 100 products from original categories *A*-*J* to clusters 1-8.

Number of classes	8
Purity	0.770
Reverse purity	0.830
Rand index	0.931
Cost function value	0.027

Table 6: Evaluation statistics for model with 8 clusters.

4.4. Model with 13 Clusters

In this model, we tried to assign 100 products from 10 categories into 13 clusters – a little more clusters than the number of given categories. The resulting assignment is shown in Table 7.

The method created cluster 4 which contains products of 3 different categories. Again we can see that category D (soap), which has the highest violation ratio, tends to be split up. On the other hand, category J (facial pads) which has the lowest violation ratio remains the same. Category G (thrash bags) is split up into two exclusive clusters. That makes sense as this category includes both thick and thin thrash bags. It is apparent that categories C, D and E were split into more clusters. Therefore customers buying items from these categories are more likely to buy more different products within the same category. This finding could help in planning promotions where the customer gets a discount on the more expensive product when a product from the same category is bought – those promotions are more effective for clusters which are not split.

Evaluation statistics are shown in Table 8. Reverse purity statistics is lower than in the previous cases. That is expected result as we estimated more categories than the number of the original ones.

4.5. Model with 20 Clusters

We have shown that the proposed method can determine categories which were originally defined expertly based on the nature of the products if the ideal behaviour (IB) is not significantly violated. In this test, we assign products to 20 clusters. The resulting assignment is shown in Table 9.

From Table 10 it follows that categories *A, B, C, F, G, H, I* and J were split into two or three clusters which can be used to define subcategories. Conversely, the categories D and E contain more clusters. None of these clusters are limited only to a single category. Categories D and E violate the ideal behaviour (IB) more than other categories. On the other hand, the method made some interesting and reasonable clusters. For example in category B all four WC liquid cleaners were clustered with pine aroma. That leads us to a fact that

Original categories									
A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*
11	9	13	1	6	10	2	8	12	3
11	1	5	4	6	10	2	8	12	3
11	9	13	6	4	10	7	8	12	3
11	9	5	13	7	10	2	8	12	3
11	9	5	1	6	10	2	4	12	3
11	9	5	6	4	10	7	8	12	3
11	9	13	4	4	1	7	8	12	3
6	9	5	1	4	10	2	8	4	3
11	9	5	1	4	10	7	4	12	3
6	9	5	4	1	10	7	8	2	3

Table 7: Assignment of 100 products from original categories *A*-*J* to clusters 1–13.

Number of classes	13
Purity	0.840
Reverse purity	0.730
Rand index	0.946
Cost function value	0.0083

Table 8: Evaluation statistics for model with 13 clusters.

Original categories									
A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*
19	4	11	18	18	3	5	16	17	13
10	15	8	18	15	1	14	20	2	13
19	7	6	17	12	3	9	20	2	13
19	7	6	11	15	1	14	16	17	13
10	4	8	12	18	3	5	20	2	13
10	7	8	15	12	3	14	16	17	13
10	4	11	12	5	1	9	16	2	13
10	4	8	12	18	1	14	20	2	4
10	7	8	18	5	3	9	15	11	19
10	4	8	15	12	1	5	20	11	13

Table 9: Assignment of 100 products from original categories *A*-*J* to clusters 1-20.

customers usually do not buy more liquid cleaners with the same aroma. Hence, products with the same function and aroma should be placed next to each other instead of sorting by brand as customers are choosing one within the products with the same aroma.

Some other clusters can not be easily described. Finding not so obvious clusters is the advantage of our method.

Evaluation statistics are shown in Table 10. Reverse purity statistics is again significantly lower as we assign to more clusters. The cost function value is also significantly lower than in the model of 3 as expected.

4.6. Computational Complexity

As we can see in Figure 6, the cost function decreases smoothly with each generation and it seems that it converges to the final solution.

Regarding actual time, it took approximately two hours to finish 1000 generations using common PC (i7 CPU with 4 cores). It seems that it is not needed to include such a large number of generations. As can be seen in Figure 6,

Number of classes	20
Purity	0.820
Reverse purity	0.510
Rand index	0.931
Cost function value	0.0036

Table 10: Evaluation statistics for model with 20 clusters.

the final assignment is found approximately by the 400th generation using our dataset based on real data. The rest of the computation was not needed. As we use the genetic algorithm to minimise the cost function and we do not know optimal solution beforehand; we cannot prove that the solution is indeed optimal. To reduce computational time, it may be useful to stop the algorithm after a given number of generations without improvement. For example, we can stop the computation if the value of the cost function is not improved in the last 50 iterations. In our case this would greatly reduce the computational time without affecting the final solution.

4.7. Comparison with Other Methods

We compared the results of our approach with other basic clustering methods – namely k-means, Ward's hierarchical clustering and self-organized maps.

Unlike our method, basic clustering methods requires some characteristics of products. We cannot directly use our cost function in these cases. We have to transform market basket data to *charasteristics* of given products.

For each product, we count the percentage of shopping baskets in which the product is bought together with each other product. Therefore, we get the assymetric square matrix of dimension 100 – the number of products in our sample. We set the diagonal of this matrix to zeros. Note that during the data transformation there is a significant loss of information.

We used implementations of these methods in R, particularly *kmeans* and *hclust* functions from package *stats* and *som* function from the package of the

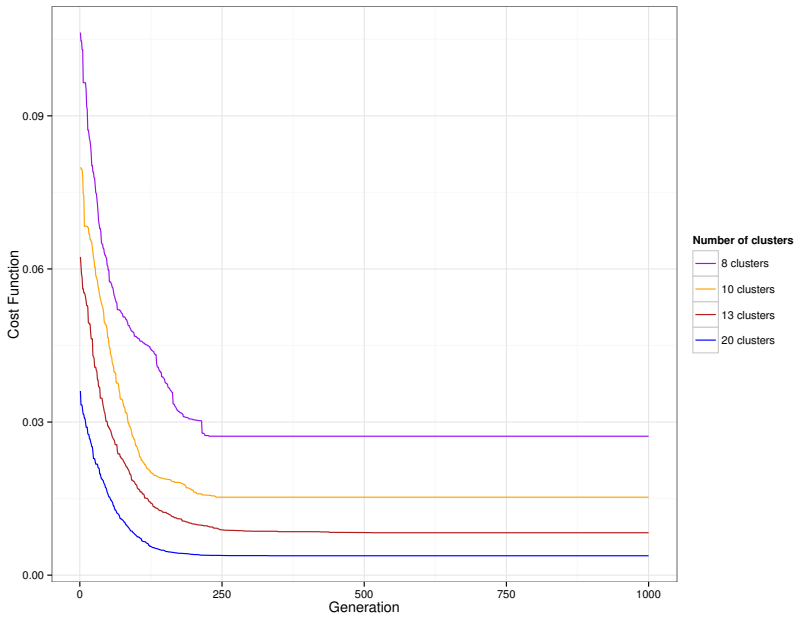


Figure 6: Cost function of best individuals in each generation in models with 8, 10, 13 and 20 clusters.

same name. For k-means method we used Hartigan-Wong's algorithm with 1000 starting locations and a maximal number of iterations set to 1000. For Ward's hierarchical clustering we set the distance to be Euclidean while other parameters were set to default values. These setups gave the best results. The self-organized map did not work well. We suspect that the bottleneck of this approach is the dimension-reduction step which is not appropriate method here. Products may not be easily represented in 2D space when our goal is to cluster products which are not commonly bought together.

On the other hand, the results of k-means and Ward's hierarchical clustering were interesting. Both methods gave almost identical results in every evaluation statistics. According to evaluation statistics such as purity, reverse purity and Rand index, for a lower number of clusters our proposed method gave significantly better results. However, with more clusters than original categories Ward's hierarchical clustering and k-means had better evaluations statistics. On the other hand, the value of the cost function applied on the resulting clustering of Ward's hierarchical clustering and k-means is significantly larger for every number of clusters. The value of the cost function is often more than ten times larger compared to our method. In Figure 7 we show resulting statistics for k-means method based on a number of clusters.

It is worth noting that evaluation statistics purity, reverse purity and Rand index are based on the belief that original categories were correctly set (e.g. they fit our assumptions). Only cost function is purely data driven statistic and as we show in Section 2.1, the objective function we propose should be more appropriate for our goal as we describe in Section 2.1.

Our method and k-means (or Ward's hierarchical clustering) found slightly different subcategories. Therefore, for practical use we recommend to explore the results of both methods.

4.8. Summary

The evaluation statistics depend on the number of clusters. Dependency on the number of clusters on real data is similar to the one presented on simulated

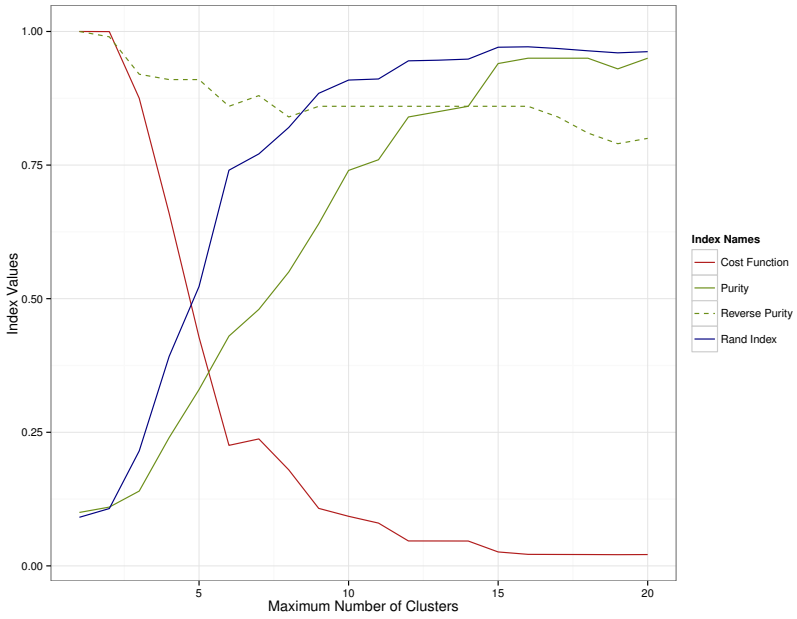


Figure 7: Evaluation statistics for different number of clusters using k-means.

data in Section 3.3 as can be seen in Fig. 8.

We have found that the product categorization of the retail chain is not perfect. The proposed method was able to find clusters which lead to interesting subcategories. That may be used for example in choosing products which are sold in small stores where space on shelves is limited.

Splitting up categories into more exclusive clusters can help with organising the shelves, e.g. not ordering products by the brand but by the other characteristic (which may be found by our method) while the products are in the same category. We remind that in this is dataset we tried to find what were the reasons that made clusters, that may not be necessary needed in the real business with sufficient amount of data.

To maximise utility, the results that are obtained by using our method should be combined with other methods, such as categorising of products by function, brand or price appeal.

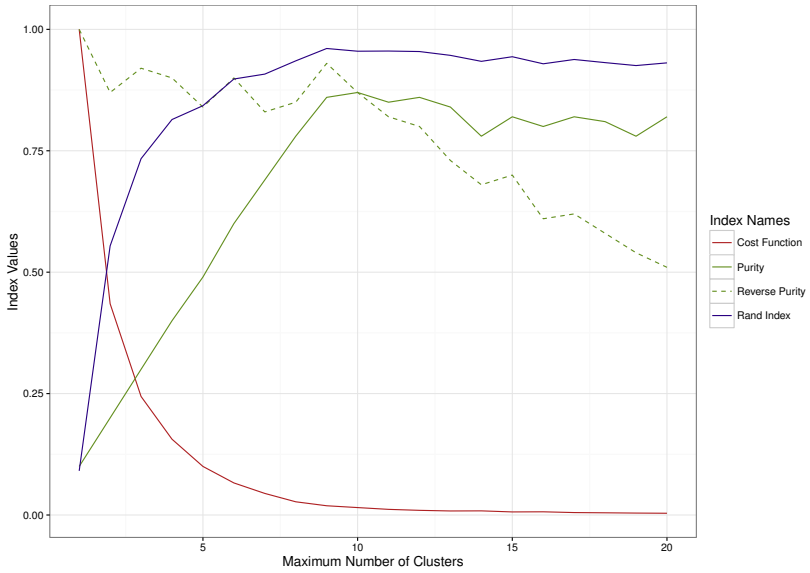


Figure 8: Evaluation statistics for different number of clusters using proposed method.

The problem we have encountered is that common drugstore's basket contains only a few items while drugstore's assortment is usually much larger than supermarket's. If we have more data the method will give better results. Therefore, we expect our method will work better on supermarket's market basket data with the larger amount of different items in the basket and also with thinner range of assortment.

5. Conclusion

We introduced a new method for the product categorization based solely on the market basket data. The method uses a genetic algorithm for dividing products into a given number of clusters.

We tested the method using synthetic and real data. The method performs well at synthetic data even if the assumptions are violated to some point. We verified our method using real market basket data from a drugstore's retail market. We found that the method accurately identified categories which do

not significantly violated the assumptions. When the assumption that customers buy at most one product from each category is violated then the products from that category were spread into several clusters instead of assigning to one cluster. It is worth noting that the original categories were subjectively chosen. Our method identified several *hidden* subcategories using only market basket data that may be widely used in marketing and in general in decision-making processes.

We found out that a common feature of customer's behaviour in the Czech drugstore market is that there are not enough receipts with a larger amount of different products, which lead to a violation of the ideal behaviour (IB) and the method's assumptions. If we had more data, we suppose that the method would give even more accurate results. Simulations using synthetic data strongly support this hypothesis.

Acknowledgements

The work of Vladimír Holý and Ondřej Sokol on this paper was supported by IGS F4/63/2016, University of Economics, Prague. The work of Michal Černý was supported by the Czech Science Foundation under Grant P402/12/G097. We would like to thank Miroslav Rada for his kind comments and Alena Holá for proofreading the paper.

References

[1] Ammar, A., Elouedi, Z., Lingras, P., 2015. Semantically Segmented Clustering Based on Possibilistic and Rough Set Theories: Semantically Segmented Clustering. International Journal of Intelligent Systems 30, 676–706. URL: http://doi.wiley.com/10.1002/int.21723, doi:10.1002/int.21723.

[2] Ammar, A., Elouedi, Z., Lingras, P., 2016. Meta-clustering of possibilistically segmented retail datasets. Fuzzy Sets and Systems

286, 173–196. URL: http://linkinghub.elsevier.com/retrieve/pii/S0165011415003619, doi:10.1016/j.fss.2015.07.019.

[3] Borin, N., Farris, P.W., Freeland, J.R., 1994. A Model for Determining Retail Product Category Assortment and Shelf Space Allocation. Decision Sciences 25, 359–384. doi:10.1111/j.1540-5915.1994.tb00809.x.

[4] Decker, R., Monien, K., 2003. Market basket analysis with neural gas networks and self-organising maps. Journal of Targeting, Measurement and Analysis for Marketing 11, 373–386. doi:10.1057/palgrave.jt.5740092.

[5] Gruca, T.S., Klemz, B.R., 2003. Optimal new product positioning: A genetic algorithm approach. European Journal of Operational Research 146, 621–633. doi:10.1016/S0377-2217(02)00349-1.

[6] Hruschka, H., Lukanowicz, M., Buchta, C., 1999. Cross-category sales promotion effects. Journal of Retailing and Consumer Services 6, 99–105. doi:10.1016/S0969-6989(98)00026-5.

[7] Jain, A.K., 2010. Data clustering: 50 years beyond K-means. Pattern Recognition Letters 31, 651–666. URL: http://linkinghub.elsevier.com/retrieve/pii/S0167865509002323, doi:10.1016/j.patrec.2009.09.011.

[8] Jiang, J.H., Wang, J.H., Chu, X., Yu, R.Q., 1997. Clustering data using a modified integer genetic algorithm (IGA). Analytica Chimica Acta 354, 263–274. doi:10.1016/S0003-2670(97)00462-5.

[9] Jonker, J.J., Piersma, N., Van den Poel, D., 2004. Joint optimization of customer segmentation and marketing policy to maximize long-term profitability. Expert Systems with Applications 27, 159–168. doi:10.1016/j.eswa.2004.01.010.

[10] José-García, A., Gómez-Flores, W., 2016. Automatic clustering using nature-inspired metaheuristics: A survey. Applied Soft Computing 41, 192–213. doi:10.1016/j.asoc.2015.12.001.

[11] Leeflang, P.S., Parreo Selva, J., Van Dijk, A., Wittink, D.R., 2008. Decomposing the sales promotion bump accounting for cross-category effects. International Journal of Research in Marketing 25, 201–214. doi:10.1016/j.ijresmar.2008.03.003.

[12] Lockshin, L.S., Spawton, A.L., Macintosh, G., 1997. Using product, brand and purchasing involvement for retail segmentation. Journal of Retailing and Consumer Services 4, 171–183. doi:10.1016/S0969-6989(96)00048-3.

[13] Manchanda, P., Ansari, A., Gupta, S., 1999. The Shopping Basket: A Model for Multicategory Purchase Incidence Decisions. Marketing Science 18, 95–114. doi:10.1287/mksc.18.2.95.

[14] Manning, C.D., Raghavan, P., Schtze, H., 2008. Introduction to Information Retrieval. 1st edition ed., Cambridge University Press, New York.

[15] Mild, A., Reutterer, T., 2003. An improved collaborative filtering approach for predicting cross-category purchases based on binary market basket data. Journal of Retailing and Consumer Services 10, 123–133. doi:10.1016/S0969-6989(03)00003-1.

[16] Niknam, T., Amiri, B., 2010. An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis. Applied Soft Computing 10, 183–197. URL: http://linkinghub.elsevier.com/retrieve/pii/S1568494609000854, doi:10.1016/j.asoc.2009.07.001.

[17] Rand, W.M., 1971. Objective Criteria for the Evaluation of Clustering Methods. Journal of the American Statistical Association 66, 846–850. doi:10.2307/2284239.

[18] Russell, G.J., Petersen, A., 2000. Analysis of cross category dependence in market basket selection. Journal of Retailing 76, 367–392. doi:10.1016/S0022-4359(00)00030-0.

[19] Seret, A., Verbraken, T., Baesens, B., 2014. A new knowledge-based constrained clustering approach: Theory and application in direct marketing. Applied Soft Computing 24, 316–327. doi:10.1016/j.asoc.2014.06.002.

[20] Srivastava, R.K., Leone, R.P., Shocker, A.D., 1981. Market Structure Analysis: Hierarchical Clustering of Products Based on Substitution-in-Use. Journal of Marketing 45, 38. doi:10.2307/1251540.

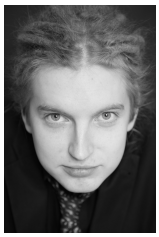
[21] Tsai, C.Y., Chiu, C.C., 2004. A purchase-based market segmentation methodology. Expert Systems with Applications 27, 265–276. doi:10.1016/j.eswa.2004.02.005.

[22] Weng, C.H., 2016. Identifying association rules of specific later-marketed products. Applied Soft Computing 38, 518–529. URL: http://linkinghub.elsevier.com/retrieve/pii/S1568494615006286, doi:10.1016/j.asoc.2015.09.047.

[23] Yang, C.L., Kuo, R., Chien, C.H., Quyen, N.T.P., 2015. Non-dominated sorting genetic algorithm using fuzzy membership chromosome for categorical data clustering. Applied Soft Computing 30, 113–122. doi:10.1016/j.asoc.2015.01.031.

[24] Zhang, Y., Jiao, J., Ma, Y., 2007. Market segmentation for product family positioning based on fuzzy clustering. Journal of Engineering Design 18, 227–241. doi:10.1080/09544820600752781.

Vitae



Vladimír Holý. Vladimír graduated in Econometrics from the Faculty of Mathematics and Physics, Charles University in Prague. He has been a PhD student at the Department of Econometrics, University of Economics in Prague since September 2014.

Ondřej Sokol. Ondřej graduated in Econometrics and Operational Research from the University of Economics in Prague. He has been a PhD student at the Department of Econometrics, University of Economics in Prague since September 2015.

Michal Černý. Michal is an Associate Professor of Econometrics and Operations Research at the Department of Econometrics, University of Economics in Prague, Czech Republic. His general research interests focus on computational problems in data analysis, ranging from complexity-theoretic analysis of statistical algorithms to practical problems in large-scale data processing. His recent work has been aimed at computational methods for data suffering from various types of imprecision, instability or indeterminacy.