Application of association rules learning for studying the store history of a large retail chain

D A Ogurtsov\textsuperscript{1} and M G Dorrer\textsuperscript{2,3}

\textsuperscript{1,2} Reshetnev Siberian State University of Science and Technology, Krasnoyarsk, prospect “Krasnoyarsky rabochy” 31, Russia

\textsuperscript{3} E-mail: mdorrer@mail.ru

Abstract. In this article the problem of goods clustering according to the probability of their joint presence in the transactions of the sale point was considered. The problem was solved by applying the algorithms for the formation of association rules based on real retail store transactions. The applicability of the algorithms of this class was assessed, the Apriori algorithm proposed by R. Agrawal was selected and implemented. Using this algorithm, the possibility of cutting off “random” goods in a cluster to create a consumer basket of the corresponding outlet was studied. It is shown that the most resistant to “random” associations are clusters with a small number of products in the cluster, and the higher the cluster size, the higher the likelihood of noisy search results with “random” goods with a high individual probability of their acquisition (‘support’ indicator). Further development of the proposed method may consist in its distribution to all transactions of the entire trading network and the transition from the tasks of organizing goods within the store (merchandising) to the tasks of forming assortment matrices for stores.

1. Introduction
The task of analyzing the consumer basket is one of the most important for retail chains, it allows to identify which products are significant for the consumer (basic for a given outlet or distribution network) and which are by-products (less purchased). The conclusions made during this analysis allow to conclude the competitiveness of the outlet; make plans for the procurement of goods; change the arrangement in the trading floor, in order to increase demand.

Piatetsky-Shapiro [1] in 1991 described the analysis and presentation of strong rules derived from databases using various measures of interest. In 1993, based on this idea, an algorithm for searching patterns in databases was proposed [2]. In tasks of data mining and dependency search, this method is very popular, because it is a simple but well-studied method for finding meaningful patterns between data in transactional databases.

In this work, we considered a data set obtained from one of the trading networks in the Krasnoyarsk city. For the first time for this retail chain, the task of arranging goods in trading floors was analytically solved based on the study of cash receipts information of a particular store. This task implied obtaining clusters of purchased goods and excluding the “random” (which is independent of other goods) clusters from the list of transactions of the store being studied. The turnover volume of the store studied in this work was about 140 thousand transactions per year. This task was investigated as the task of data mining,
obtaining knowledge from the database and of searching for patterns. Due to the specificity of the task and the data, an algorithm of association rules learning was proposed to solve this problem.

2. Materials and methods.

2.1. Basic formulas.
Association rules learning is based on the basic concepts of probability theory, including Bayes’ theorem.

The first of the association rules learning concepts is support (1).

\[
support(y_1 \cup y_2) = \frac{\sigma(y_1 \cup y_2)}{|T|},
\]

where \( y_1 \) and \( y_2 \) – the products of interest to us; \( \sigma(y_1 \cup y_2) \) – the number of transactions containing \( y_1 \) and \( y_2 \); \( T \) – the total number of transactions. Support shows what percentage of acquisitions of these goods was recorded, relative to the total number of transactions; the frequency of this rule across the entire data set.

The second of the association rules learning concepts is confidence (2).

\[
\text{confidence}(y_1 \cup y_2) = \frac{support(y_1 \cup y_2)}{support(y_1)},
\]

where \( support(y_1) \) – probability of purchase of this product. Meaningfully, this rule reflects the percentage of those who bought product \( y_2 \) after purchasing \( y_1 \).

The third of the association rules learning concepts is lift (3).

\[
lift(y_1 \cup y_2) = \frac{support(y_1 \cup y_2)}{support(y_1) \times support(y_2)},
\]

this fraction shows the relation of the “dependence” of the rule elements to their “independence”. Thus, if lift = 1, we conclude that the goods are independent and there are no rules for joint acquisition of goods. If lift > 1, then the amount by which lift is greater than 1 will show the strength of the relationship, and of the rule as a whole. The bigger than 1 lift is, the stronger is the relationship. If lift <1, it shows that the condition of the rule \( y_1 \) negatively affects the consequence of the rule \( y_2 \) (the opposite is also true).

The fourth and last of the concepts is conviction (4).

\[
\text{conviction}(y_1 \cup y_2) = \frac{1-support(y_2)}{1-confidence(y_1 \cup y_2)},
\]

this fraction reflects the frequency of errors of the rule. For example, when product \( y_1 \) was bought without product \( y_2 \), and contrariwise.

It is worth noting that \( y_1 \) and \( y_2 \) can be represented not by one but by a number of objects. Every rule is composed by two different sets of items, also known as “itemsets”, \( y_1 \) and \( y_2 \), where \( y_1 \) is called antecedent or left-hand-side (LHS) and \( y_2 \) consequent or right-hand-side (RHS).

An example of a sale point rule is \{ soda, chocolate, chips\} → \{ chewing gum \}. This rule reflects the following pattern: if soda, water, chocolate and chips are bought, then chewing gum will be bought.

2.2. Algorithms.
There are a large number of algorithms for associative rules generating. The main ones are Apriori, Eclat and FP-Growth algorithms. A comparative analysis of these methods allows us to conclude that, on the available data set of the problem to be solved, the choice of algorithm is not critical, since the density of the transaction base (the number of transactions containing frequently encountered goods) does not exceed 20%, and the length of transactions (the number of goods in a transaction) does not exceed 20 [3]. As a result, the Apriori algorithm was chosen to solve the problem. [2,4,5].

2.3 Apriori algorithm description.
INPUT: D - database containing a list of transactions; \( \sigma \) - user-defined threshold support;
OUTPUT: Transaction list \( F(D, \sigma) \)

APPROACH:
1. \( C_1 \leftarrow \{i\} | I \in J \)
2. \( K \leftarrow 1 \)
3. While \( C_k \neq 1 \) do:
   4. We consider all support for all candidates for all transactions \((tid, I) \in D\) do:
      for all candidates \( X \in C_k \) do:
         if \( X \in I \):
            \( X. \text{support}++ \)
   5. Then, we pull out all the frequent transactions:
      \( F_k = \{X | X.\text{support} > \sigma \} \)
   6. Generating of new candidates
      \( \forall X, Y \in F_i, X[i] = Y[i] \) for \( 1 \leq i \leq k-1 \) and \( X[k] \leq Y[k] \) do:
         \( I = X \cup \{Y[k]\} \)
      if \( \forall J \subseteq I, |J| = k: J \in F_k \) then
         \( C_{k+1} \in C_k + I \)
      \( k++ \)

2.4. Cut-off metric of "random" product.
During the study, the possibility of buying a "random" product was considered. In this case, "random" we call a product purchased independently of the rest of the cluster goods. Since clusters are cascaded (first, clusters with the highest support, then lower), the exclusion of these clusters leads to the fact that we can consider the same cluster without “random” goods on a higher level of support. For example, we have a cluster \{product1, product2, product3\} with a support level of 10%, but we know that the product “product3” is “random”, therefore, we cut off this cluster, since we have more high-level clusters \{product1, product2\}, \{product3, product2\}, \{product1, product3\}, which may be of interest to our study.

The following heuristic method was used as a cut-off metric: we took the cluster product with the maximum support (support for this single product) and the cluster product with the minimum support (support for this single product), then, if the difference between these goods was considered significant, this cluster was cut off (with a difference value = 5%).

3. Results.
For this application, Python was chosen as one of the languages that allows working with data efficiently, due to such libraries as pandas, numpy and to visualization tool matplotlib. The application was based on a package created by Yu Mochizuki with the implementation of the Apriory algorithm [6].

The initial type of data was presented in the form of the table 1, and had fields that were not interesting for the analysis (such as ticket office number, date, etc.), therefore, pre-processing was required.

Table 1. An example of the presentation of the source data for the classification of goods

| Transaction number | Item name | Amount |
|--------------------|-----------|--------|
| 1001               | A         | 2      |
| 1001               | D         | 3      |
| 1001               | E         | 1      |
| 1002               | A         | 2      |
| 1002               | F         | 1      |
| 1003               | B         | 2      |
| 1003               | A         | 2      |
It should be noted that initially the product names were presented as elements of the string type written in Cyrillic, but due to the processing problems in Python, we had to change the names to product codes.

During pre-processing, extra columns were removed; product groups have been collapsed (for example: before the transformation, “fresh-frozen beef ham, cellophane”; after the transformation, “fresh-frozen beef”); the data was presented in the form of a list, where each element was a separate transaction (where the goods were presented in the form of a list).

As a result, the data took the view: [[1001, 1002, 1003], [1001, 1003, 1004, 1005], [1005, 3003], …].

The following is the analysis of association rules at the level of the support minimum value = 0.0005 (0.05%). Subsequently, item names were restored and table 2 was obtained, where items_base are the basic goods (LHS), items_add are the addable goods (RHS).

Table 2. Data after applying the Apriori algorithm.

| Cluster products              | Base products and addable products                                                                 | Support % |
|-------------------------------|----------------------------------------------------------------------------------------------------|-----------|
| 'Pasteurized milk'           | [{'items_base': [], 'items_add': ['Pasteurized milk'], 'confidence': 0.0832, 'lift': 1.0}]        | 9.63      |
| ['Bananas']                  | [{'items_base': [], 'items_add': ['Bananas'], 'confidence': 0.0712, 'lift': 1.0}]                | 8.98      |
| 'Wheat bread'                | [{'items_base': [], 'items_add': ['Wheat bread'], 'confidence': 0.0632, 'lift': 1.0}]             | 8.25      |
| ['Apples']                   | [{'items_base': [], 'items_add': ['Apples'], 'confidence': 0.0576, 'lift': 1.0}]                 | 5.25      |
| ['Cucumbers', 'Sweet pepper'] | [{'items_base': ['Cucumbers'], 'items_add': ['Sweet pepper'], 'confidence': 0.2245, 'lift': 8.4325}, {'items_base': ['Sweet pepper'], 'items_add': ['Cucumbers'], 'confidence': 0.3245, 'lift': 8.4325}] | 0.38      |

Further, in figures 1-4, the distribution of goods by cluster is shown, in order of increasing number of cluster objects. The ordinate shows the difference between the maximum value of the cluster support indicator (the most frequently purchased cluster product) and the minimum value of the cluster support (the least frequently purchased cluster product) - there are points (clusters) with “random” goods above, closer to zero with their absence; on the abscissa axis - cluster support (rule triggering frequency).
Figure 1. Distribution of clusters consisting of 2 products.

Figure 2. Distribution of clusters consisting of 3 products.
Figure 3. Distribution of clusters consisting of 4 products.

Figure 4. Distribution of clusters consisting of 5 products.
In figures 5-8, the distribution of goods by cluster is shown, in increasing order of the number of cluster objects, together with the names of the goods in this cluster. The first 6 clusters are presented, which are closest to zero (with threshold cutoff = 5%, along the ordinate axis) and with the greatest support (with the most purchases, along the abscissa axis).

**Figure 5.** Distribution of 6 clusters, maximum by value of support, consisting of 2 products.

**Figure 6.** Distribution of 6 clusters, maximum by value of support, consisting of 3 products.
Figure 7. Distribution of 6 clusters, maximum by value of support, consisting of 4 products.

Figure 8. Distribution of 6 clusters, maximum by value of support, consisting of 5 products.
In the tables 3-6, the distribution of goods by clusters, which were presented in figures 5-8, in increasing order of the number of cluster objects, together with the names of the goods in this cluster, is shown. The first 6 clusters are presented, which are near to zero (with threshold cutoff = 5%) and with the highest support (probability of cluster meeting).

**Table 3.** Distribution of 6 clusters, maximized by the support value, consisting of 2 products.

| Support | Max-min | Cluster                      |
|---------|---------|------------------------------|
| 3.72    | 1.38    | Milk pasterized; Wheat bread |
| 1.83    | 0.28    | Cucumbers; Tomatoes          |
| 1.76    | 0.00    | Beer; Tobacco products       |
| 1.33    | 0.00    | Wheat bread; Tobacco products|
| 1.33    | 0.00    | Sour cream; Wheat bread      |
| 1.32    | 0.00    | Milk pasterized; Tobacco products |

**Table 4.** Distribution of 6 clusters, maximized by the support value, consisting of 3 products.

| Support | Max-min | Cluster                                      |
|---------|---------|----------------------------------------------|
| 0.43    | 1.45    | Milk pasterized; Wheat bread; Tobacco products|
| 0.34    | 4.36    | Milk pasterized; Wheat bread; Beer           |
| 0.30    | 0.28    | Green cultures; Cucumbers; Tomatoes          |
| 0.27    | 1.4     | Mayonnaise; Milk pasterized; Wheat bread     |
| 0.25    | 2.62    | Cucumbers; Tomatoes; Bananas                 |
| 0.24    | 2.77    | Cucumbers; Sweet pepper; Tomatoes            |

**Table 5.** Distribution of 6 clusters, maximized by the support value, consisting of 4 products.

| Support | Max-min | Cluster                                           |
|---------|---------|---------------------------------------------------|
| 0.073   | 2.55    | Cucumbers; Tomatoes; Bananas; Apples              |
| 0.072   | 2.72    | Green cultures; Cucumbers; Sweet pepper; Tomatoes |
| 0.071   | 4.47    | Milk pasterized; Wheat bread; Beer; Tobacco products |
| 0.056   | 2.83    | Chicken egg; Green cultures; Cucumbers; Tomatoes |
| 0.056   | 2.55    | Green cultures; Cucumbers; Tomatoes; Bananas      |
| 0.055   | 3.67    | Sour cream; Green cultures; Cucumbers; Tomatoes   |

**Table 6.** Distribution of 6 clusters, maximized by the support value, consisting of 5 products.

| Support | Max-min | Cluster                                                                 |
|---------|---------|-------------------------------------------------------------------------|
| 0.02    | 4.53    | Sour cream; Chicken egg; Buckwheat; Sunflower oil; sugar                |
| 0.019   | 2.63    | Potatoes; Carrot; Cucumbers; Tomatoes; White, red, bulb onion          |
| 0.019   | 2.6     | Green cultures; Cucumbers; Tomatoes; Bananas; Apples                   |
| 0.017   | 3.62    | Cucumbers; Tomatoes; Bananas; Apples; Tangerines                       |
| 0.016   | 5.00    | Cucumbers; Sweet pepper; Tomatoes; Bananas; Apples                     |
| 0.016   | 2.74    | Green cultures; Cucumbers; Sweet pepper; Tomatoes; White, red, bulb onion |
Table 7 considers 6 clusters, maximum in support, consisting of 2 products, with the addition of a lift column, to assess the dependence of products on each other.

**Table 7.** Consideration of 6 clusters with maximum support, consisting of 2 products, with the addition of a lift.

| Product 1       | Product 2       | Support | Lift |
|-----------------|-----------------|---------|------|
| Wheat bread     | Tobacco products| 1,3     | 0,701|
| Milk pasteurized| Tobacco products| 1,32    | 0,615|
| Milk pasteurized| Wheat bread     | 3,72    | 1,851|
| Cucumbers       | Tomatoes        | 1,83    | 10,976|
| Beer            | Tobacco products| 1,76    | 1,132|
| Sour cream      | Wheat bread     | 1,33    | 1,914|

4. Discussion

The images above show that the distribution of product clusters, consisting of a different number of products, with the cluster volume = 2 has the largest number of points, which allows us to assume that most of the relationships in the sample under consideration exist between the two products. The distribution along the ordinate axis allows us to judge the “randomness” of this cluster. With clusters of volume = 3 and above, we see that there is an increase in the percentage of “random” products in the sample. When considering the graphs of clusters, in which 6 groups of products are presented, one can see that they have some discrepancy in the graph field along the ordinate axis, even though values closer to zero were taken. As a result of the above we can conclude that with an increase in the number of goods in the cluster, the initial value along the ordinate axis increases (the difference between the support of the most bought and the least bought), since each new product increases the likelihood of “random” occurrence. Therefore, the more goods are represented in the cluster, the greater the likelihood of a “random” product, and the less analysis results are resistant to statistical emissions. This situation corresponds to a behavioral situation when a buyer comes for ordinary goods and takes along with these goods some very popular product (for example, bread) that has a high level of support (the probability of buying this product as a whole for the entire sample of analyzed transactions).

The data in tables 3-6 heuristically confirm the hypothesis of clusters without “random” goods. The only exception to this hypothesis is that LHS and RHS can both be “random”. In this case, the task may be solved by the consideration of the following data. We can assume that when considering the cluster and the lift value from it, we can reliably say whether the relationship is reliable, and in the case of lift <1, we can say that the goods affect each other negatively and we can remove this cluster from consideration so that there are no goods “artificially” creating a cluster. Examining table 7 (for clusters of volume = 2, for which “random” goods were cut off), we can assume that this hypothesis can be considered heuristically fair and possible to apply in practice, for further planning the layout of the sales area, which will increase consumer demand.

5. Conclusions

This article explores the application of the algorithm for association rules learning (Apriory Algorithm), the real store data in one of the Krasnoyarsk retail chains were analyzed for their further application in the formation of a procurement plan and the formation of the arrangement of goods on the trading floor (merchandising). The cut-off metric was applied to form statistically stable clusters that would be represented by homogeneous goods, to form a consumer basket for this supermarket. In the future, it is possible to consider such data for all retail outlets in the network to assess the competitiveness of the store in comparison with other supermarkets in the distribution network. Another useful result of the
proposed transaction analysis procedure can be a generalization of the task of arranging goods in stores of the distribution network, and the formation of an assortment matrix. To solve this problem, the algorithms proposed in the work should be applied to the set of transactions of all supermarkets in the distribution network.

To clarify the methods of analysis and confirm the correctness of heuristic techniques, a comparative analysis of the transactions of several stores of the distribution network can be applied.

References
[1] Piatetsky-Shapiro G 1991 Knowledge Discovery in Databases (AAAI/MIT Press) pp 229–48
[2] Agrawal R, Imielinski T and Swami A 1993 Proc. of the 1993 ACM SIGMOD Int. Conf. on Management of Data 22(2) pp 207–16
[3] Heaton J 2016 IEEE SoutheastCon 2016 pp 1-7
[4] Mohammed Z, Parthasarathy S, Ogihara M and Wei L 1997 Proc. of the Third Int. Conf. on Knowledge Discovery and Data Mining (KDD-97) pp 283-6
[5] Rakesh A and Ramakrishnan S 1994 Proc. of the 20th Int. Conf. on Very Large Data Bases (VLDB ’94) pp 487-99
[6] Mochizuki Y 2017 Repository with realization of Apriory Algorithm URL: https://github.com/ymoch/apyori