FP-Growth in Discovery of Customer Patterns

Jerzy Korczak, Piotr Skrzypczak

FP-Growth in Discovery of Customer Patterns

Jerzy Korczak[1] , Piotr Skrzypczak[2]

[1]Wrocław University of Economics, Poland,[2]Delikatesy Alma, Wrocław, Poland,
53-345 ul. Komandorska 118/120, Wrocław, Poland
jerzy.korczak at ue.wroc.pl, piotrek.skrzypczak at gmail.com

Abstract. The paper describes a knowledge discovery platform and a novel process for finding association rules based on the algorithm FP-Growth and its variants. Built software solution has been optimized in terms of memory usage and computation time as well as the impact of all modifications made to the whole process of rules discovery The process of rule discovery is illustrated on a real database containing transactions of customers of the e-shop Delicatessen Alma24.

Keywords: Mining of association rules, analysis of customers' transactions, improvement in FP-Growth performance.

1 Introduction

One of the most popular and widely used methods of finding customers' shopping patterns are the methods based on algorithms of association rules. Over several years, a number of search algorithms have been developed for association rules in data sets [Hand, 2005; Kotsiantis, 2006; Morzy, 2010; Pasztyła, 2010]. Many of them have evolved, some proved to be less useful, because of their performance on large data sets and too great demands on available memory. One interesting proposal is the FP-Growth (frequent pattern growth) algorithm developed by J. Han, H. Pei and Y. Yin [Han, 2000; Han 2004]. FP-Growth uses an extended prefix-tree structure, called FP-tree, to store the customer transactions in a compressed form. This algorithm is fast and scalable. The publication of J. Han showed that FP-Growth performance surpasses other popular methods of searching for association rules, such as Apriori or Tree Projection algorithms. The papers of [Zaki, 1997], and [Borgelt, 2005], showed that this algorithm has better performance than Eclat and Relim.

The popularity and effectiveness of the FP-Growth algorithm was appreciated in many studies, in which to improve its efficiency many changes have been proposed to the original algorithm [Gyorödi, 2003; Racz, 2004; Zaki 1997]. These changes are mainly related to accelerating the construction of the FP-tree and its reduction of computing time as well as memory complexity.

The first modification was proposed by C. Gyorödi [Gyorödi, 2003]. It addressed two problems in the FP-Growth algorithm, namely, that the resulting FP-tree is not unique to the same "logical" database, and that in order to create the FP-tree two complete scans of the database are required. The developed algorithm DynFP-Growth solved the first problem by introducing the lexicographical order of support, thus

ensuring the uniqueness of the FP-trees for different but "logically equivalent" databases. In order to solve the second problem, the algorithm changes dynamically the order of elements of the FP-tree by performing the "promotion" (offset) to the higher-order one of the smallest items detected. An important feature of this solution is that it is not necessary to rebuild the FP-tree when the database is updated.

The way to reduce the size of the tree is implemented in the algorithm of FP-Bonsai [Gyorödi, 2003]. The FP-tree is pruned using the data reduction technique ExAnte [Bonchi, 2003]. The originality of this solution is based on the rejection at the first scan of items whose support is less than the required minimum value. In addition, after the first scan and creation of a table header, the data set is sorted by the value of support and re-entries when the support less than the assumed minimum value are rejected. Thanks to these modifications the size of the FP-tree is reduced several times. The pruned FP-tree called FP-Bonsai improves the efficiency of the algorithm.

The latter essential modification refers to the time and memory complexity of the FP-Growth algorithm. The NONORDFP algorithm [Hand, 2005] modifies the structure of the FP-tree, which is thus more compact and does not need to be rebuilt for each conditional step. The new FP-tree representation in the memory assures faster search, faster allocation, and possibly better projection.

The paper will present results of pilot studies of consumer behavior of one of Wroclaw's Alma Delicatessen stores, based on sales from August and September 2009 to September 2009 [Skrzypczak, 2010], and the studies on a new project carried out on the data from August 2009 to January 2010. In the project, the MySQL database system, the authors' software DM Cafe, and RapidMiner package (http://www.rapidminer.com/) have been used to implement the FP-Growth algorithm. The main aim of the platform's development was to assist decision makers in search of interesting and nontrivial association rules that can allow better understanding of customers' profiles and their preferences, and improve sales performance.

The article is divided into four sections. The next section describes the algorithm of FP-Growth and characteristics of its main parameters. In the third section, a database containing transaction data and information on commodities is presented. The fourth section describes the process of extracting association rules using RapidMiner [Bereta, 2010]. A modified process of rule discovery, implemented in [Skrzypczak, 2010], is detailed, as well as its impact on system performance.

2 Database of customer transactions and mining algorithm

The aim of the project was to design a platform to integrate the existing transactional system with new functionalities of rule extraction. In the experimental platform three functional components have been identified, namely, a transactional database with the store commodities tables, software for exchanging information between the stock information system and cash management, and an application to mine association rules.

The database, managed by the MySQL server, consists of the following tables: Items, Groups, Departments, Stands, Cash transactions, and Supplementary Data (Fig. 1).

To exchange information between systems and the database, DM Cafe is applied. It is a program to write data to the stock information system and the cash management whose data are used in the study. Data for the test are read directly from the database using the appropriate SQL query in RapidMiner.

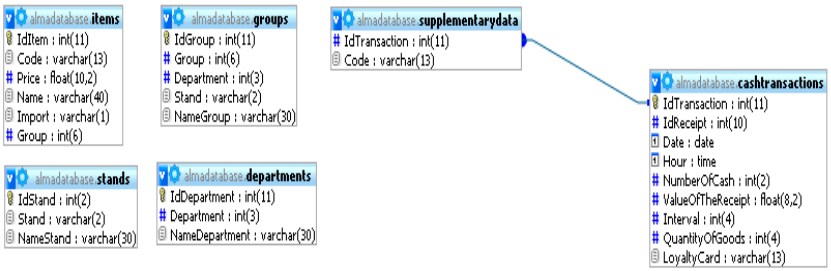

Fig. 1. Database schema

Generally, a database contains a set of customers transactions $T_1,..., T_n$, where each transaction T_i describes a set of items bought by a customer. In the case study the database contains over 470 thousand records in six tables. Most of the data are collected in the table Supplementary Data; there are codes assigned to the transaction IDs. There are over 370 thousand records for the entire store, and over 25 thousand Alma24 records (Alma24 is an online store). The table Items stores information about more than 63 thousand of Alma Delicatessen commodity codes. During the evaluation period, the point of sales registered over 39 thousand transactions, of which more than 1,000 belonged to Alma24.

To discover customer basket patterns, the FP-Growth algorithm has been used [Han, 2000]. The algorithm is looking for the complete set of frequent patterns understood as patterns with the occurrence frequency no less than a predefined by managers minimum support ratio. Among these frequent patterns, the confident ones are selected to create classification rules. These rules are used to predict items to be bought and class labels for future transactions.

The algorithm contains two basic steps: compression of the data set in a form of FP-tree and mining of association rules from FP-tree. The FP-tree is built using two passes over the database. In the first pass, the algorithm searches the database for all frequent 1-item and then removes infrequent items from the transaction T_i. As a result, a modified set of transactions $T^* = T_{1*},..., T_{m*}$ is created consisting of only frequent 1-item sets. Then a set of transactions is sorted in descending order according to the support ratio of each transaction and transformed into a compact tree structure called a FP-tree. The FP-tree is a rooted acyclic graph with non-labeled vertices. The root graph has a label 'null', the remaining graph vertices, both internal nodes and leaves, represent 1-item sets. Each graph node, except the root, is linked to the label that represents a 1-item set and the counter of transactions, representing the number of transactions supporting a given set (Fig. 2). In the second scan of the transaction database the FP-tree is constructed and then can be used for mining frequent customer basket patterns. It is important to note that the FP-tree efficiently

compresses the database and avoids costly and repeated data scans as Apriori type algorithms. More information about the theoretical foundations of the FP-tree construction can be found in [Han, 2000; Han, 2004].

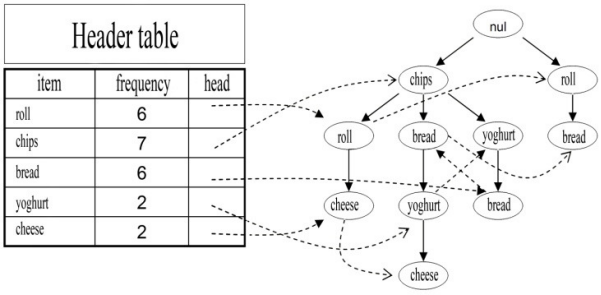

Fig. 2. Example of the header table and the corresponding FP-Tree.

To explore information stored in an FP-tree and extract the complete set of frequent patterns, the algorithm FP-Growth, has been applied [Han, 2004]. The FP-Growth starts to mine the frequent patterns 1-itemset and progressively grows each such itemset by mining its conditional pattern-base. A conditional pattern-base is a set of patterns that co-occur with a particular node in a given path. All the computed frequent patterns related with node a_i creates a small FP-tree, called a_i-conditional FP-tree and denoted as "*FP-tree*|a_i". The processes of construction of conditional pattern-bases and conditional FP-trees are carried out by pattern growth recursively.

The pseudo-code of the algorithm FP-Growth is given below:

Procedure *FP-Growth*(*Tree, α*)
// *α is an itemset in transactional database*
// *β is an itemset in α's conditional pattern-base*
{
if Tree contains a single prefix path // Mining single prefix-path FP-tree
then {
let P be the single prefix-path part of *Tree*;
let Q be the multipath part with the top branching node replaced by a *null* root;
for each combination (denoted as *β*) of the nodes in the path *P do*
generate pattern *β* ∪ *α* with *support* = *minimum support of nodes in β*;
let freq_pattern_set(*P*) be the set of patterns so generated; *}*
else let Q be *Tree*;
for each item a_i in *Q do {* // Mining multipath FP-tree
generate pattern $\beta = a_i \cup \alpha$ with *support* = a_i *.support*;
construct β's conditional pattern-base and then *β*'s conditional FP-tree *Treeβ* ;
if Treeβ = ∅
then call FP-Growth(*Treeβ, β*);
let freq pattern set(*Q*) be the set of patterns so generated; *}*
return(*freq_pattern_set*(*P*) ∪ *freq_pattern_set*(*Q*) ∪ (*freq_pattern_set*(*P*) × *freq_pattern_set*(*Q*)))

The algorithm has two initial parameters: *Tree = FP-tree*, and *α = null*. If the *FP-tree* has only a single path *P*, then for each combination of β vertex path *P* is created a set of $\beta \cup \alpha$ with the support equal to the minimum support of items belonging to the set

β. If the *FP-tree* contains more than one path, then for each element belonging to the array, α_i *Tree* header is created with a set of $\beta = \alpha_i \cup \alpha$ supporting corresponding elements α_i. Next is generated a conditional pattern base of β and conditional *FP-tree* pattern of β, denoted *Tree*β. After this step, it is verified whether *Tree*β is empty or not. If it is empty, the algorithm is ended, otherwise the procedure FP-Growth is restarted with parameters of *Tree* = *Tree*β, and $\alpha = \beta$. The last line of the procedure returns the three sets the generated frequent patterns from *P, Q*, and $P \times Q$.

3 Process of association rules discovery

The rule extraction process depends not only on the size of the database, but also on two very important measures of rule interestingness: support and confidence ratios. A support ratio defines the frequency of a given combination of items in the database, and a confidence ratio that reflects the likelihood that a particular rule appears. The threshold values of the support and the confidence are set by the managers to indicate which pattern or group of items can be considered as a frequent pattern or frequent itemset. Depending on these values, a different number of frequent sets (by increasing or decreasing the value of the support ratio) and association rules (by changing the value of the confidence ratio) may be generated.

In our application the customer transactions are read by RapidMiner using SQL queries. The transactions are then transformed into a matrix and passed to the FP-Growth algorithm. The matrix contains the items, and 0 and 1 (0 means no item occurrence of the transaction, 1 - the item was in the shopping cart) is needed to establish the structure of the FP-Tree. Figure 3 shows the process created to discover rules in the transactional data of the online store Alma24.

The approach to rule discovery presented in this article can be applied to any transactional database system. Just a suitable SQL query has to be used that returns two columns: transaction ID and the name of the item, group, department, etc. By default, the process finds all association rules for given support and confidence ratios. In addition, the association rules can be found for specific items, group or department; for example, if a customer shopping cart contains tomato, what else is there? The system is able to answer the specific question asked by the stand manager or market data analyst.

After preprocessing and searching frequent itemsets, association rules are created. When the process finishes, the results are visualized.

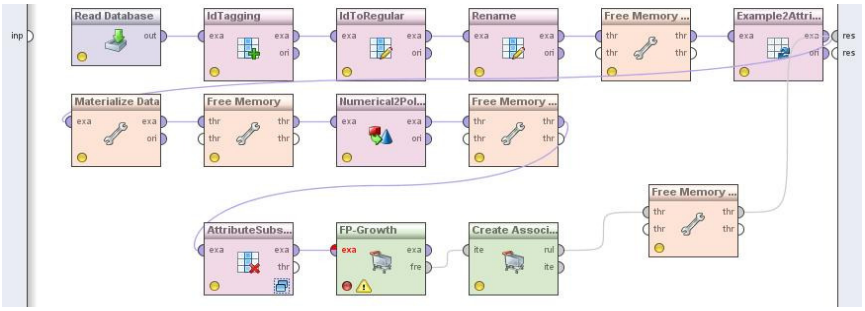

Fig.3. Schema of the rule discovery process

The obtained association rules can be presented in a table; if required, they may be sorted. In addition, the display mode can be changed using the sliders and drop-down list located on the left side of the results (Fig. 4). It is also possible to visualize the rules as a graph (Graph View option), as well as text, similar to those written to the file by the operator Write as Text. In the paper, the rules are presented in the tables ordered by Confidence level. If needed, part or all of the rules can be exported to various external formats, such as a PDF, XML.

Fig.4. Changing the display mode

The data to be analyzed represented the Alma24 customer purchases from August to September 2009. Each customer transaction contained information on purchased items, ie., commodity code, quantity, price, value of purchases, information about the possession of Connaisseur Club card (sort of loyalty card), and also the mode of payment (cash, card, gift certificate, bank transfer). The study was conducted at the level of the item code for the customers whose shopping cart value was greater than 200 zł (ca 50 euro). In general, a shopping cart with the value greater than 200 zł is supposed to contain more than 2-3 items, so in consequence, the program can be able to discover association rules that are more interesting and useful for managers.

The minimum support value was set at 2%, while the minimum confidence was set at 80%. The minimum values of the parameters were defined in cooperation with the Sales Department of Alma24. The results are shown in part in Table 1. Thirty one rules were found respecting 2% minimum support. Among the interesting association rules can be pointed out those with a probability of 100% (confidence = 1), notably :

"At least 2% of the customers of tomatoes and a quarter chicken always buy the new potatoes",

"At least 2% of the customers of tomatoes per kg and chopped Cirio tomatoes in pieces always buy new potatoes"

"At least 2% of the customers of tomatoes per kg and Lubella brand wheat flour wheat always buy new potatoes"

"At least 2% of the customers of red pepper per kg and Hajnowka butter always buy tomatoes per kg"

Tab.1. Subset of discovered rules

Premisses	**Conclusion**	**Confidence**
tomato kg, quarter chicken	new potato kg	1
tomato kg, Cirio chopped tomatoes 400g	new potato kg	1
tomato kg, Lubella wheat flour 1kg	new potato kg	1
red pepper kg, Hajnowka butter 200g extra	tomato kg	1
white grape kg, watermelon kg	tomato kg	1
banana chiquita kg, white grape kg	new potato kg	1
new potato kg, UHT milk 3.2% 0.5l	chicken drumsticks	1
chicken drumsticks, UHT milk 3.2% 0.5l	new potato kg	1
apple kg, red onion kg	red pepper kg	1
tomato kg, banana chiquita kg, white grape kg	new potato kg	1
tomato kg, Piatnica cottage cheese	new potato kg	0.889
apple kg, cheese gouda	tomato kg	0.889
tomato kg, celery kg	parsley kg	0.889
tomato kg, parsley kg	celery kg	0.889
chicken breast, Piatnica cottage cheese	new potato kg	0.889
bunch chives, red onion kg	red pepper kg	0.889
new potato kg, Cirio chopped tomatoes 400g	tomato kg	0.875
tomato kg, peaches kg	new potato kg	0.875
new potato kg, peaches kg	tomato kg	0.875
dark rye bread, Piatnica cottage cheese	tomato kg	0.875
dark rye bread, Piatnica cottage cheese	new potato kg	0.875
UHT milk 3.2% 1l , Piatnica cottage cheese	new potato kg	0.875
banana chiquita kg, chicken drumstick	apple kg	0.875
tomato kg, red pepper kg, white grape kg	new potato kg	0.875
new potato kg, red pepper kg, white grape kg	tomato kg	0.875
red pepper kg, parsley bunch	tomato kg	0.846
red pepper kg, Danone yogurt 135g natural	tomato kg	0.818
red pepper kg, white grape kg	tomato kg	0.800
red pepper kg, white grape kg	young potato kg	0.800
wholemeal bread, ground cucumber	red pepper kg	0.800
banana chiquita kg, red onion kg	red pepper kg	0.800

The majority of the resulting association rules refer to best-selling products and are generally known to the sales department of the Alma Delicatessen. Some interesting rules were discovered concerning the fruit and vegetable stand; however it should be

pointed out that they were related to the period of August-September. During this period, products such as tomatoes, potatoes, watermelon, and parsley were very cheap and were often found in shopping baskets. Puzzling is the rule "*At least 2% of the customers of tomatoes per kg and chopped Cirio tomatoes always buy new potatoes*", because the chopped tomatoes should rather be related to pasta, chicken and herbs, the ingredients used to cook spaghetti.

The second study was carried out on transactions from August 2009 to January 2010. Due to the fact that the transactions involved a longer period than previously, the value of the minimum support is set to 2%, so as to be able to find association rules useful for the sales department. Table 2 shows the results of the experiment for customers with shopping cart value more than 200 zł.

Tab. 2. The association rules relating to shopping cart value more than 200 zł

The number of item sets: 503	**The number of association rules: 7**	
Min support: 2%	**Min confidence: 80%**	
Rules		**Confidence**
carrots kg, celery kg	parsley kg	0,882
mandarin kg, parsley kg	carrots kg	0,871
onion kg, celery kg	parsley kg	0,861
tomato kg, celery kg	parsley kg	0,846
lemon kg, parsley kg	carrots kg	0,811
banana chiquita kg, celery kg	parsley kg	0,806
onion kg, parsley kg	carrots kg	0,8

More specific analysis might be carried out. For instance, from the standpoint of sales, interesting rules might be also retrieved in more narrow itemsets containing items from specific groups or departments. Such analysis might be more useful for managers because of binding association rules with some specific goods. Managers are also interested to filter out the transactions with best-selling products. Some of these cases will be illustrated further in this section. However, in our experiments most of this pre-processing resulted in the rejection of items and generating known and useless association rules.

To carry out the process of discovering association rules for specific items, a predefined SQL query has been created. It consists of two parts:

- query asking the names of items that will create a set of rules to look for;
- subqueries returning the transaction IDs that include the items of interest.

This query composition has been used in all three experiments presented below.

Experiment 1. Market managers are also interested in the discovery of association rules related to particular items. To illustrate the problem, two queries were asked:

1) What items except pastas are in the shopping cart above 200 zł?
2) What items besides milk are in the shopping cart above 200 zł?

The minimum support ratio was set up after consultation with the managers equal to 4%, and the minimum confidence ratio of 70%.

As a result of the first question, 33 rules were discovered; the most interesting are presented in Table 3.

Tab. 3. The association rules relating to shopping cart value more than 200 zł contain any pasta

The number of itemsets: 197	**The number of association rules: 33**	
Min support: 4%	**Min confidence: 70%**	
Rules		**Confidence**
new potato kg, chicken legs	gouda cheese	1
cheese Gouda, chicken legs	new potato kg	1
cheese Gouda, chicken legs	new potato kg	1
chicken wings	new potato kg	0.889
water nes.wat. Naleczowianka 1.50l n	chicken fillet	0.875
new potato kg, white grapes kg	tomato kg	0.875
red pepper kg, banana chiquita kg	tomato kg	0.875
red pepper kg, parslay bunch p.	tomato kg	0.875
flour lubella 1kg Poznańska pszenna	new potato kg	0.857
butter Hajnówka 200g extra	cheese „gazda z dziurami”	0.857
new potato kg, cheese „Gazda z dziurami”	tomato kg	0.857
red pepper kg, white grapes kg	tomato kg	0.857
banan chiquita kg, carrots kg	tomato kg	0.857

Among the association rules obtained there are many items from the fruit-vegetable stand. Some of these rules can be interpreted using one's cooking experience, and say that the customers of this target group buy these items to cook a particular dish, for instance, spaghetti (if pasta, red pepper and parsley, then tomato) or potatoes au gratin (if pasta, gouda cheese, and chicken sticks, then potato).

Experiment 2. In the second experiment the fruits and vegetables have been removed from the itemsets. If not, something very similar to the previous rules would be generated. But we wanted to find a unique relationship between the shopping cart in which there were milk and other products.

The discovery process drew 59 rules. Most of the rules describe a combination of milk, butter, gouda cheese, cream and chicken legs (or chicken wings). Sample rules are shown in Table 4.

Tab. 4. The association rules related to shopping cart value more than 200 zł contain any milk

The number of item-sets: 412	**The number of association rules: 59**	
Min support: 4%	**Min confidence: 70%**	
Rules		**Confidence**
butter kerrygold 200g irish	gouda cheese	1
butter kerrygold 200g irish	sour cream piątnica 18% 200ml	1
butter kerrygold 200g irish	chicken wing	1
butter kerrygold 200g irish	gouda cheese, sour cream piątnica 18% 200ml	1
sour cream Piątnica 18% 200ml, butter	Gouda cheese	1

kerrygold irish 200g		
butter kerrygold irish 200g	Gouda cheese , chicken wing	1
chicken wing, butter kerrygold irish 200g	Gouda cheese	1
butter kerrygold irish 200g	sour cream Piątnica 18% 200ml, chicken wing	1

It can be stated that customers belonging to this group prepare a dish usually made up of these items. Among this group are also those with children; this is due to the following rules:

"4% of customers who had in the basket milk and semolina porridge ml., always buy the nectar Bobo Frut 300ml apple-raspberry-cherry "

"4% of customers who bought milk and semolina porridge 190g, organic vanilla, always buy the yogurt with apples or black berries"

"4% of customers who bought milk 190g and dessert nutricia biscuits bobovita banana, always bought porridge nutricia vita 190g and creamy bobo" „

"4% of customers who bought milk and chocolate 100g kraft nussbeisser alpen 100g, bought chicken fillet in 77.8% of cases".

Further analysis of the rules for chocolate and chicken would give a precise explanation of the relationship between these items. Among the rules was found also this quite obvious association::

"4% of customers who bought milk and coffee Jacobs Cronat gold200g, bought sugar in 83.3% of cases."

Experiment 3. The third experiment focused on multi-level frequent pattern mining, based on item hierarchies in Alma. The FP-Growth was applied several times reusing the existing tree structure to discover multi-level association rules. The algorithm examined the tree structure in a bottom-up manner, it means starting at the leaves and proceeds all the way up until the root of the tree collecting information about item names, groups of items and related frequencies. Rules have been generated respecting the support and confidence ratios.

The table 5 illustrates some of the interesting rules.

Tab. 5. Some interesting association rules for a group of customers with shopping carts above 200 zł containing any butter or chicken

Min support: 4%	**Min confidence: 70%**	
Rules		**Confidence**
butter, pasta	garlic	1
butter, cheese, eggs	toilet paper	1
butter, toilet paper, eggs	cheese	1
butter, cheese, toilet paper	eggs	1
chicken, toilet paper	milk 2%	0.800
chicken, margarine	sparkling water	0.800
chicken, bread	garbage bags	0.800
chicken, canned tomatoes	cheese	0.800

From this experiment the following rules are discovered:

- 4% of the customer group with a basket above 200 zł buy items that are needed to prepare spaghetti (rule: if someone bought butter and pasta, it always bought the garlic, and if someone bought the chicken and tomatoes , 80% of cases, also bought the cheese);
- 4% of customers in this target group have in their shopping carts butter, cheese, eggs and toilet paper
- 4% of customers order items that are not logically related to each other, eg.

  If the shopping cart contained a chicken and toilet paper, in 80% of cases there were 2% milk

  If the shopping cart contained a chicken and bread, and in 80% of cases were also garbage bags.

The rules of Table 5 can be used directly to increase sales in the online shop Alma24. One way would be a campaign to promote butter or chicken at reduced prices, ei. in the form of a discount coupon that can be sent to the customers of Alma24. At the same time, in order to increase the shop's income, every customer who buys the items using the coupon would receive an additional offer immediately after the purchase of items from the association rules at reduced prices, eg. for 90% of the normal price. This is a typical use of the method of up-selling, which is very effective at increasing the income from a single transaction.

4 Performance analysis

Initially, the pilot study of the process was developed and evaluated. The transactional data were collected from CSV files, which resulted in many sub-processes that complicated the task of of rule discovery. t. Several experiments demonstrated that the process excessively uses memory and is very time consuming [Skrzypczak, 2010].

The low performance of the prototype has lead to redesign the whole process of rule discovery. The data were imported using SQL queries directly into Rapid Miner, transformed in the matrix from which frequent itemsets were searched, and, finally, association rules were discovered.

In the new solution, the demand for memory was decreased respectively from 1000 MB to 800 MB. The reduction of memory, however, was not significant for a smaller data set. More advantages generated the operators Materialize Data (writes data from memory) and Free Memory (clears working memory) that considerably decreased usage of memory. With these modifications the process runs almost six times faster than the pilot version. The computing time was diminished (respectively 16 s. for data used in the article, and 87 s. for the pilot study).

Figure 5 shows the effect of the new solution on the memory size and the duration of the current process.

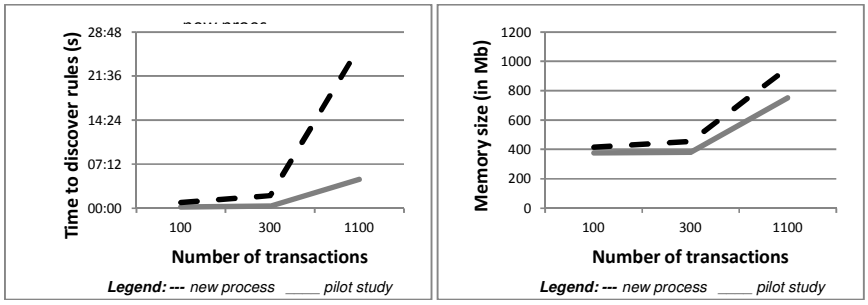

Fig.5. Performance of rule discovery system depending on the number of transactions

Thanks to these improvements the whole process is much efficient, despite the use of the same algorithm for finding frequent items and construction rules. The new process also consumes less memory, making it possible to conduct experiments on a much larger data sets with an identical hardware configuration.

5 Conclusions

The article has presented the process of extracting association rules in customers' transactions of the Internet Delicatessen Alma24. Initially, the process of rule discovery was developed and used in pilot studies. After conducting several experiments, it turned out that the process is inefficient and uses large amounts of memory. Therefore the process was redesigned.

The new solution demonstrated that the design of rule discovery process has an impact not only on the required amount of memory, but also on the computing time to obtain the final result. Thanks to this solution the whole process is much shorter, despite the use of the same algorithm for finding frequent items and construction rules. The FP-Growth is fast and scalable avoiding the costly process of candidate generation and testing used by Apriori algorithm

Of course, it is important to choose efficient data mining algorithms. However, one has to take into account the process of data cleaning, consolidation, and transformation of data into appropriate form.

The implemented solution helped the Alma managers to make better profitable sale decisions by discovering the buying habits of their customers. The rules allowed to focus on items and group items that are most likely to buy by customers. In general, the rules have been used to improve the shopping environment, customize marketing efforts and provided location-aware recommendations to customers. From the viewpoint of internet shop, the obtained knowledge was used to improve Website services, advertisements and increase volume of sales.

Acknowledgements. The authors thank the Board of Alma Delicatessen in Wroclaw for access and the use of corporate data for the article.

References

1. Bereta M.: *Data Mining z wykorzystaniem programu RapidMiner*. (http://michalbereta.pl/dydaktyka/ZSI/Lab%20Data%20Mining%201.pdf, czerwiec 2010), (http://michalbereta.pl/dydaktyka/ZSI/Lab%20Data%20Mining%202.pdf, czerwiec 2010).
2. Bonchi F., Giannotti F., Mazzanti A. and Pedreschi D.: *Exante: Anticipated Data Reduction in Constrained Pattern Mining*. In: Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'03), pp. 3-7, Cavtat-Dubrovnik (2003)
3. Borgelt C.: *Keeping Things Simple: Finding Frequent Item Sets by Recursive Elimination*. Workshop Open Source Data Mining Software (OSDM'05), ACM Press, pp. 66-70, Chicago (2005)
4. Gyorödi C., Gyorödi R., Cofeey T. and Holban S.: *Mining Association Rules using Dynamic FP-trees*. In: Proceedings of The Irish Signal and Systems Conference, University of Limerick, pp.76-82, (2003)
5. Han J., Pei J, Yin Y.: *Mining Frequent Patterns without Candidature Generation*. In: Proc. of the 200 ACM SIGMOD Int. Conf. on Management of Data, pp. 1-12, Dallas (2000)
6. Han J., Yin Y., Mao R.: Mining Frequent Patterns without Candidat Generation: A Frequent-Pattern Tree Approach. In: Data Mining and Knowledge Discovery, pp.53-82, Kluwer Academic Publ., 8, (2004)
7. Hand D., Mannila H., Smyth P.: *Eksploracja danych*. Wydawnictwo Naukowo-Techniczne WNT, Warszawa (2005)
8. Kotsiantis S.,Kanellopoulos D.: Association Rules Mining: A Recent Overview. In: Proc. GESTS Internat. Trans. on Computer Science and Eng., pp.71-82, vol.32 (1), (2006)
9. Morzy T.: *Eksploracja danych*.(http://www.portalwiedzy.pan.pl/images/stories/pliki/publikacje/nauka/2007/03/N_307_06_Morzy.pdf, (2010)
10. Pasztyła A.: *Analiza koszykowa danych transakcyjnych – cele i metody*. (http://www.statsoft.pl/pdf/artykuly/basket.pdf, (2010).
11. Rácz B.: *NONORDFP: An FP-Growth Variation without Rebuilding the FP-Tree*. In: 2nd Int'l Workshop on Frequent Itemset Mining Implementations FIMI, (2004)
12. *RapidMiner 4.3. User Guide. Operator Reference. Developer Tutorial*. (http://docs.huihoo.com/rapidminer/rapidminer-4.3-tutorial.pdf, (2010)
13. Skrzypczak P. : *Modelowanie wzorców zachowań klientów Delikatesów Alma przy wykorzystaniu reguł asocjacyjnych,* Master Thesis, Uniwersytet Ekonomiczny, Wrocław (2010)
14. Zaki M., Parthasarathy S., Ogihara M., Li W.: *New Algorithms for Fast Discovery of Association Rules.* In: Proc. 3rd Int. Conf. on Knowledge Discovery and Data Mining (KDD'97), pp.283–296, AAAI Press, (1997)