The impact of threshold parameters in transactional data analysis

M. Vranić, D. Pintar and M. Banek
University of Zagreb
Faculty of Electrical Engineering and Computing
Zagreb, Croatia
E-mail address: {mihaela.vranic, damir.pintar, marko.banek}@fer.hr

Abstract - Today's information systems store large quantities of transactional data which may contain valuable but hidden information. Association rules generation is a method developed for analysis of this type of data. This method is however very resource-intensive. Both the execution time and the final model highly depend on threshold parameters set by the analysts. In this paper we analyze the impact of the minimal support parameter on the number of closed frequent itemsets discovered for various datasets. The analysis is conducted on referential datasets as well as real-life datasets which classify transactional elements in categories organized in hierarchical manner. Datasets that are not originally transactional can be transformed into a transactional form. They exhibit somewhat different relations between the minimal support parameter and the number of discovered closed frequent itemsets. Findings presented in this paper can serve as guidance in setting up support as the most important parameter affecting the final execution time. In order to analyze data characteristics from another perspective, we also present how varying confidence, lift and support can affect the number of formed association rules containing two elements.

I. INTRODUCTION

Current information systems store large quantities of transactional data. This data is often summarized and further analyzed in an aggregated form. However, analysis of raw transactional data at the lowest level can offer different insights into the data. These insights mainly refer to relationships between specific transactional elements. The data mining method developed to explore this specific type of data is association rules generation. This method is very resource-intensive and thus often accompanied with long execution time – especially if the threshold parameters are set too low. In this case very large number of rules is generated, with many of them redundant or uninteresting. Alternatively, if the threshold parameters are set too high, the generation of rules will be finished in short time, but only a small number of rules will be produced, if any, and will most probably express some facts previously known to the business analysts.

The most resource-intensive step in association rules generation is finding closed frequent itemsets (CFIs, definition given in Section III). The number of CFIs determines the total execution time of the final model generation. Knowledge about the relationship between the number of frequent itemsets and the threshold parameters can be a useful guidance for the analysts to ensure acceptable execution times and satisfactory results, especially if coupled with domain knowledge and specificities of similar datasets. The concerned relationship depends on various factors such as: data domain and inherent relationships between the elements, number of transactions, number of available items, average number of items per transaction etc. Some other measureable statistics of observed dataset could also be of importance (such as mode, median, etc.).

In order to investigate the problem from different perspectives, we used various natively transactional datasets: a small referential dataset and a collection of datasets extracted from a large real-life database concerning sales, courtesy of a major Croatian retailer. The extracted datasets present elements across different hierarchical levels of categorization, so that the analyst can observe the behavior of threshold parameter settings in different scenarios. The conclusions drawn from our experiments can serve as guidance for choosing most promising hierarchical level for exploration as well as setting up threshold parameters.

We also considered possible appliance of the developed concepts on other types of data. Non-transactional data can be adapted into a transactional form before conducting the analysis, which may enable new insights into the dependencies between the observed objects or the variables connected to them.

There are also additional parameters that could be set up by analysts which affect the finally presented association rules. The most commonly used parameters are *confidence* and *lift*. When considering relationships between pairs of elements, it is interesting to see how many of them are filtered out by combinations of support and lift or support and confidence. The resulting graphs presenting the number of unfiltered association rules are useful for drawing certain generic conclusions that can be applied across various datasets.

Our analysis was conducted using the Apriori algorithm [10] implemented through a Python program module. It enabled us to control all steps in rule generation process, measure the execution time and the total number of CFIs resulting from different support threshold settings.

It also enabled us to further investigate the filtering of two-element rules using different filter combinations.

The paper is structured as follows: Section II presents the related work on the subject. Section III gives important definitions and explains the used terminology. In Section IV, the solution framework for analysis is introduced. Sections V and VI present the details of our experiments and discuss their results. Finally, in Section VII a conclusion is given.

II. RELATED WORK

The natural method intended for transactional data examination is association rules generation firstly presented in [1]. One of the problems regarding it is the fact that it is very resource-demanding and dependent on threshold parameters. The best-known algorithm for finding CFIs is the Apriori algorithm. There are many papers concentrated on finding more effective algorithms ([2], [3], [4], [5]) but no significant improvements have been achieved. Recently, the focus has been put on finding rules that would be significant and new to the analysts ([6], [7], [8]). Still, setting up parameters is not an easy task. This paper is focused on examination of various datasets characteristics that strongly affect the process and the final model of association rules generation. The strategy of setting up the threshold parameters should be guided by these examined characteristics.

Association ruled generation can also be used for examination of data which is not originally transactional. Transformation to required form can reveal new cognitions about relationships between variable values [9]. It is useful to explore properties of this specific type of transformed data which expose an extreme case of investigated relationships.

The basic steps of the Apriori algorithm are presented in [10]. The most resource-intensive step is the generation of multi-element frequent itemsets (e.g. containing *M* elements) which are formed by combining of already identified frequent itemsets (FI) containing *M-1* elements. Here, the notion that FI cannot be frequent if not all of its subsets are frequent is used to eliminate certain FIs from further examination. This is the basic Apriori property which speeds up the entire process.

III. IMPORTANT TERMS AND DEFINITIONS

This section provides definitions and explanations of the terms used later in this paper.

Let I be a set of possible elements found in transaction: $I = \{I_1, I_2, \ldots I_n\}$. **Transaction** $t \subseteq I$ is a record of a business event modeled as a set of elements and stored in the input dataset D. **Itemset** $A \subseteq I$ is a set of elements from I used by the data mining process and possibly provided as part of its results. **Support** of the itemset A (*supp(A)*, *s*(*A*)) is a proportion of transactions in D which contain A. Absolute support (*AbsSupp*(*A*)) is the number of transactions in D which contain A. Itemset A is a **superset** of itemset B if $B \subseteq A$. B can be referred to as a **subset** of A.

An **itemset is frequent** if its support is larger than the proposed minimal support on given dataset. An **itemset is closed** if no superset of it exists with the same support on the given dataset. The existence of itemsets that are not closed is implied by appearance of their supersets. That is why the main issue in the analysis of transactional data is finding the **closed frequent itemsets (CFIs)** – itemsets that are both closed and frequent.

Based on certain multi-element CFIs, different association rules can be produced. Association rules appear in the form A⇒B, accompanied with certain parameters. A and B may consist of one or more transaction elements, and A and B together form one CFI ($(A \cup B) \in \mathcal{CFI}$, where $\mathcal{CFI}$ is set of all CFIs with a certain support threshold).

Confidence (c) indicates the probability of an element appearing on the right side of the rule if the left side of the rule is present in the transaction:

$$c(A \Rightarrow B) = \frac{s(A,B)}{s(A)} \quad (1)$$

Confidence evaluates to a number contained in the interval [0, 1]. As opposed to support, however, this is an asymmetrical measure, meaning that the order of attributes is important and that different orderings can produce different values.

Lift (L) is a measure used in association rules generation which implies the strength of the relationship between two elements as opposed to those elements being together in a transaction by random chance:

$$L(A \Rightarrow B) = \frac{c(A \Rightarrow B)}{s(B)} = \frac{s(A,B)}{s(A)s(B)} \quad (2)$$

Lift is a symmetrical measure with preferred values being larger than 1.

The final set of rules depends directly on the number of CFIs. All possible rules that could be formed on the basis of found CFIs are tested to conform to confidence or/and lift threshold parameters. The final model of association rules is consisted of all rules that comply with those parameters.

IV. SOLUTION FRAMEWORK

In order to be able to analyze the characteristics of various datasets our research included development of software application. With regards to the overall functional requirements, our application needed to perform the following:

- interpret and/or adequately transform transactional input data
- prepare foundation models for storing and manipulating frequent datasets (collections)
- implement the Apriori algorithm to mine frequent datasets
- organize the mined datasets in a structure which would facilitate further calculations and manipulations

- allow the input of additional threshold parameters and perform analysis over mined datasets with regards to the effect of said thresholds
- present the overall results in a readable, rapidly interpretable way using tables and graphs

We initially developed data preparation scripts able to pull data out of various sources and convert them to the standardized tab-delimited format. Next we modeled software representations of transactions and collections, with corresponding structures for storing and manipulation. For this, as well as the subsequent data mining and analysis routines, we chose the Python programming language, since it suits the rapid development requirements and also allows easy integration with the open source Orange data mining tool, also developed in Python. In order to enable easy result interpretation, the implementation tool exports the results as text files in tabular form, accompanied by appropriate statements used by the Mathematica software which allows instant representation of results in 2D and 3D graph form.

Object representations of the input data were named Transaction and Collection. Transaction is basically a binary vector (or list) corresponding to one of the rows in the input data. Collection is a subclass of Transaction, and it represents one of the ultimately mined frequent datasets. It is expanded with some additional numeric parameters (such as support). Transactions are bundled inside a TransactionSet, which, aside from being a container for transactions, also stores attribute names, original filename etc. Collections, on the other hand, are placed inside a LayeredCollectionSet – a structure that houses but also organizes mined frequent collections based on their support, number of attributes etc.

CollectionSetFactory is the main module for converting input data into TransactionSets and, subsequently, LayeredCollectionSets. It implements the Apriori algorithm but also a number of extra functionalities which calculate additional numerical parameters for the set. It allows caching of already processed datasets to avoid unnecessary repeated analyses and even enables internal linkage of collections with custom-typed links, which can be used to superimpose additional organizational structures over the mined collections.

Finally, Threshold Analysis Module uses the LayeredCollectionSet to perform additional analysis described in Section VI. As a result, it produces tables in textual form as well as prepared statements for the Mathematica tool.

Figure 1 presents the developed software tool from the perspective of process execution flow, together with all the implemented components.

Our implementation also enabled us to analyze time consumption, which depends mainly on the support threshold parameter affecting the final number of CFIs (dependencies are presented through graphs in Section V). Of course, somewhat faster algorithms will demonstrate quicker results generation, but results we gathered is a good guidance for an analyst to estimate the execution time even when new algorithms are used.

The application was operating on a PC with the following characteristics: Intel® Core™ i3 CPU running at 3.07 GHz, with 4.00 GB of RAM and with Windows®7 Professional installation (32-bit operating system). Although this machine is rather suboptimal concerning the expected performance of data mining analysis hardware, the required analysis has been done successfully and within reasonable time.

V. Dependency of CFI number on support threshold

The interdependency between number of CFIs and support threshold parameter is a characteristic of every dataset. It depends on relationships between transactional elements, data domain and semantics. Taking market basket analysis for example, it can be stated that combinations of transaction elements can be a consequence of customer habits, season, weather, various customers' characteristics, but also specific store offerings, product characteristics and so on. However, examination of various datasets revealed that regardless of dataset specifics and origin, lowering the support threshold commonly results in an explosion of the total number of CFIs. An immediate consequence of this is a drastic increase in execution time and resource requirements.

In order to show an interdependency between support, execution time and the number of CFIs, a real-life dataset describing the appearance of certain category members in

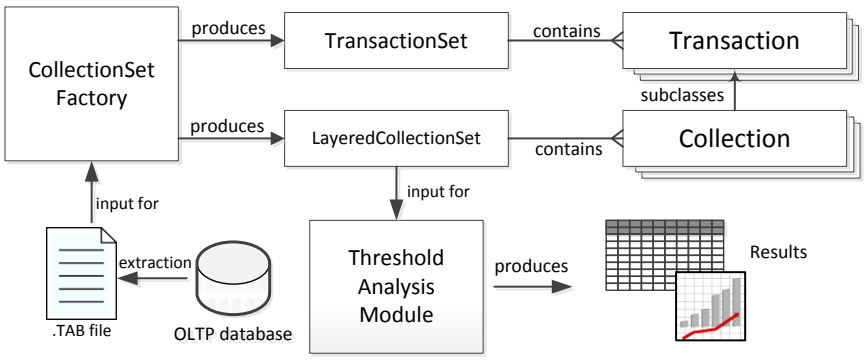

Figure 1. Important entities of developed tool together with denoted process flow

transactions was chosen (a detailed description of this dataset is given in subsection C). Graph presented in Figure 2 shows how lowering the absolute support value results in a rapid increase of total CFIs, especially when support is set lower than a certain threshold (this brink actually depends on the dataset at hand, in this case 30 transactions is the approximate value at which the number of CFIs explodes). Plotting the interdependency between execution time and support would result in an almost identical graph. In order to gain a better view of relationship between execution time and number of CFIs, the correlation of these two variables is shown in graph in Figure 3. As expected, these two variables exhibit an almost linear interdependency. In other words, execution time is directly affected by the number of gained number of CFIs at a certain support threshold. Specific coordinates in Figure 2 (represented as dots) correspond to the CFIs represented by coordinates in Figure 3, resulting in a high density of measured values for the absolute support threshold higher than 50 transactions.

The dependency between the number of CFIs and the minimal support threshold is further investigated in the following subsections. Each subsection is dedicated to specific dataset.

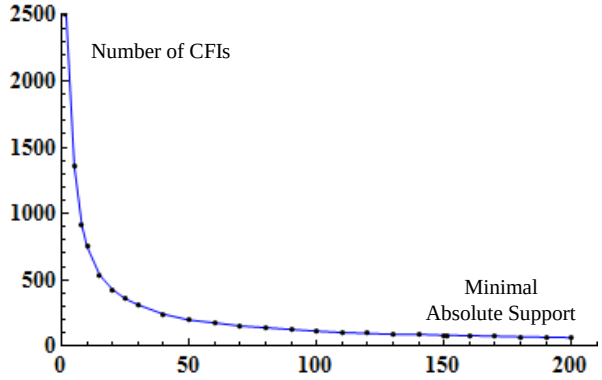

Figure 2. Relationship between number of CFIs and minimal absolute support threshold – real-life retail dataset

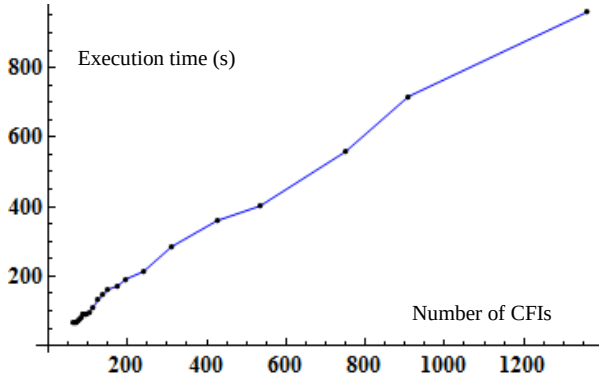

Figure 3. Relationship between execution time and number of CFIs

A. *Referential dataset analysis*

The first analysis was conducted on a small, realistic, natively transactional dataset describing purchases of computer equipment. Dataset is called *Computer shop* dataset and is available as one of the test datasets for the *Oracle 11g Database Management System*. Some earlier research regarding tree structure formations over this dataset is presented in [11] and [12]. *Computer Shop* dataset contains 940 transactions and 14 transaction elements. Each transaction is a record of a purchase in a shop dealing with electronic equipment. Average number of elements per transaction is 2.98, and the most frequent element appears in 32% of transactions.

The density of this transaction dataset is relatively high, meaning that each transaction holds a large proportion of available elements. Eleven elements show up in more than 18% of all transactions. Furthermore, the total number of elements and transactions isn't too large which allows for acceptable execution times even with low support thresholds. Lowering the support threshold from 9% to 1% causes a ten-fold increase in the total number of CFIs (at the minimal support of 9.38% the total number of CFIs is 21, while 1.25% results in 235 CFIs).

The behavior of 2-element CFIs is perhaps the most interesting, since their number is pretty stable from 5% support threshold downwards (Figure 4). If the analyst is interested in relationships between pairs of elements, 5% represents a threshold under which no further benefits are gained. On the other hand, the number of collections with three or more elements is rising steadily at a nearly exponential rate as the support threshold is lowered – at lower values more and more subtle relationships are constantly being revealed, at the cost of execution time and final model presentation clarity.

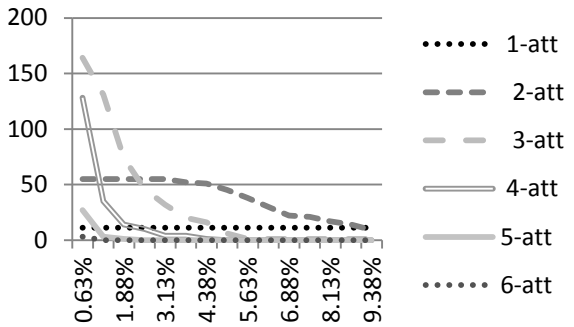

Figure 4. Relationship between minimal support threshold and number of CFIs – Computer shop dataset

B. *Analysis of a non-natively transactional dataset*

In this subsection an analysis of the well-known car evaluation dataset [13] is given. It is intensely explored in evaluation of different predictive data mining methods and thus is suited to serve as an example of dataset not natively transactional. Dataset keeps data about 1728 cars described by 7 attributes, each having a domain containing several values (Table I). Attribute describing class will be treated the same as other attributes. Other attributes describe different car characteristics – such as price (*buying* attribute), maintenance cost (*maint* attribute) etc. Transformation of the dataset to an appropriate transactional form is done by representing each attribute value as a separate attribute in the new transactional dataset. Consequently, for each car instance a new attribute was set to value *1* if in the original dataset the corresponding attribute was described by this value. All other attributes corresponding to values not connected to

TABLE I. ATTRIBUTES AND VALUES FOR CAR EVALUATION DATASET

Attribute	*Possible values*
Class variable	unacc, acc, good, vgood
buying	vhigh, high, med, low
maint	vhigh, high, med, low
doors	2, 3, 4, 5more
persons	2, 4, more
lug_boot	small, med, big
safety	low, med, high

specific instance were set to *0*. The final transactional dataset contains 1728 instances described by 25 attributes.

An inherent characteristic of this kind of dataset – i.e. a non-natively transactional dataset transformed in the described fashion – is its fixed density. Each row will contain the exact number of present elements as was the number of attributes in the original dataset. In this case this density is 7/25=28%. This high density together with coverage of different car descriptions results in a vast number of CFIs as the support threshold is lowered – lowering the support from 9% to 1.25% results in 23 times more CFIs (from 75 to 1378). If the analyst chooses to use the minimal support value lower than 10%, he must use additional stronger filtering mechanisms such as confidence or lift to keep the results manageable.

A careful examination of 2-element itemsets reveals that relatively high support values already result in a large number of CFIs and that their number stays relatively stable with lowering the support threshold (Figure 5). For example, the minimal support of 6.25% results in 175 2-element CFIs, while lowering the minimal support to 1,25% generates 193 2-element CFIs. The total number of possible combinations is $\binom{25}{2} - 4 \cdot \binom{4}{2} - 3 \cdot \binom{3}{2} = 267$, which means that a significant portion of all combinations has been covered. This is especially true when taking into consideration that the mentioned number of 2-element CFIs does not include those (which are contained in larger CFIs which mask them by having the same support, as is the property of closed collections).

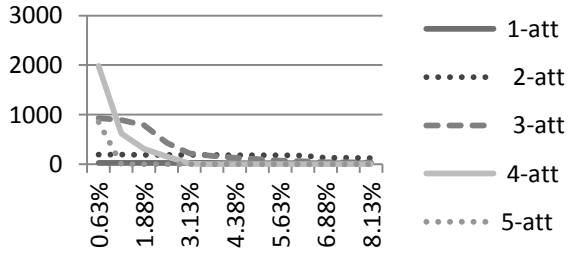

Figure 5. Relationship between minimal support threshold and number of CFIs – Car evaluation dataset

What is also specific for this dataset is a sudden increase in the number of 5-element CFIs with low support values – over 150 CFIs has exactly the same absolute support value of 12. This indicates that the dataset is most likely artificially constructed to cover most scenarios and present a challenge for predictive analysis. Additional proof for this conclusion stems from the fact that counting the exact support of certain maximal CFIs result in a repeated appearance of same numbers.

C. *Real-life dataset analysis with presentation of elements on different levels of categorization*

Analysis of different datasets extracted from the real life database enabled interesting and applicable conclusions. Data about 73009 transactions were available, which referred to 19893 different products. There were 1368 product categories organized in 4 hierarchical layers. Multiple analyses were conducted, out of which one will be briefly discussed. This one concerns the subset of 29 lowest layer categories dealing with cosmetic products. The dataset depicted 23674 transactions. Each observed category held approximately 60 products.

A typical graph for the lowest level of categorization can be seen in Figure 6. Even though the number of transactions is significantly higher than in the previous datasets, the number of CFIs is noticeably lower even with using very low support values. Also, the lines are smoother, meaning that the number of CFIs changes more gradually and constantly with support lowering.

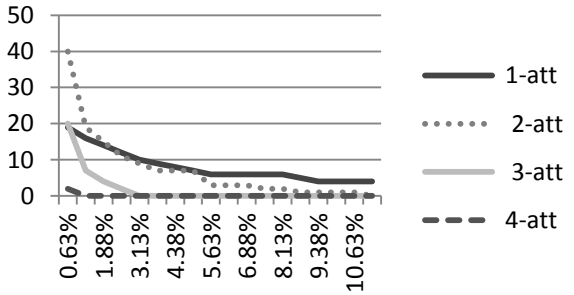

Figure 6. Relationship between execution time and number of CFIs – real-life retail dataset, 1st level of categorization

Apart from working with categories, we also prepared a dataset describing the products at the basic level. With that dataset, density was extremely low so supports had to be drastically lowered to get any interpretable results. We conclude that staying on the category level should be recommended since the research at that level tends to be more productive. The results are therefore easier to interpret and can be used as a basis for the decision making process.

VI. DEPENDENCY OF NUMBER OF TWO-ITEM ASSOCIATION RULES ON DIFFERENT THRESHOLDS

Observing just relationships between pairs of elements can also showcase certain characteristics of the dataset at hand. For this reason an analysis of the total number of two-element association rules in relation to pairs of threshold parameters was conducted. The most common combination of parameters is (support and confidence) or (support and lift). The results were plotted in 3D graphs, a sample of which can be seen in Figure 7 (*Car evaluation* dataset, support/lift) and Figure 8 (*Retail dataset*, support/lift).

The graph in Figure 7 tied to the transformed dataset results in visible plateaus and emphasizes specific threshold values. The brinks on the graph are thresholds

where large number of rules gets filtered out. Lift measure of 1.0 is particularly interesting, since at that value almost all rules are instantly being discarded. Using confidence, a very similar graph with sharp slopes is produced. This dataset exposes an extreme case of interconnections between transactional elements. Therefore, if an analyst is interested in two-element rules, he will gain very few of them by setting lift above 1. However, an analyst will probably also be interested in more complex rules, with low support, high confidence and/or high lift, but care should be taken in order to avoid overfitting.

For the retail datasets, graphs of entirely different appearance are produced (Figure 8), with no plateaus and a constant but gradual change of the rule number. Still, the sharp edge of the graph means the support parameter needs to be lowered significantly to achieve a certain number of results. However, attention is needed in order to avoid an explosion due to the exponential nature of the support projection of the graph. The lift value should preferably be kept larger than 1.0 though, in concordance with the semantic interpretation of this particular measure. All researched real-life datasets presenting products at a specific categorization level expose similar characteristics.

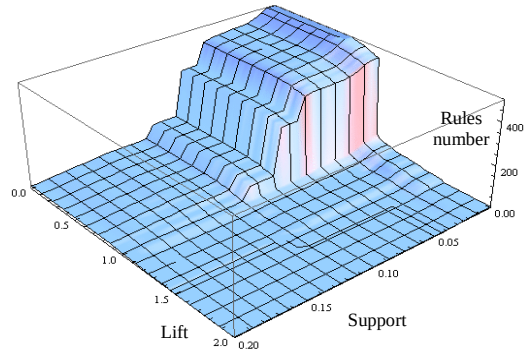

Figure 7. Number of survived rules with varying support and lift thresholds – Car evaluation dataset

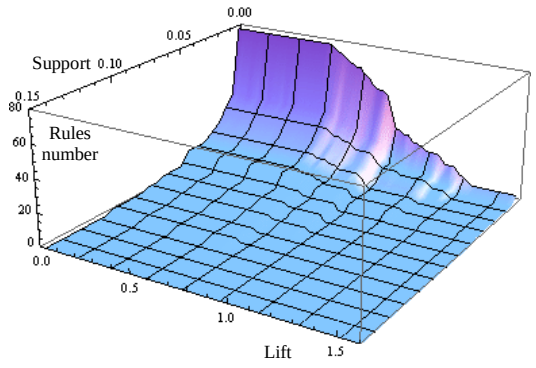

Figure 8. Number of survived rules with varying support and lift thresholds – retail, 1st level of categorization

VII. CONCLUSION

Transactional data plays an increasingly larger role in business decision making process. Today's analyses require methods which can be executed in real time and provide meaningful and rapidly interpretable results. This paper demonstrated the impact of the threshold parameter choice in association rules generation on the execution time and the resulting model, concerning its volume and the ease of its interpretation. Different datasets with varying characteristics were analyzed – a referential transactional dataset, a dataset non-natively transactional but transformed into the transactional form, and – most importantly – a real-life retail transactional dataset. The choice of datasets covered extreme scenarios that can be expected in association rules generation process.

Generally speaking, in order to properly leverage the impact of threshold parameters on the transactional data analysis, the analyst needs to have certain *a priori* knowledge about the dataset or at least its characteristics shared with similar datasets. Execution time needs to be kept acceptably low, while the number of CFIs should stay manageable to allow efficient interpretation. With real-life datasets, which are the true focus of this work, the analyst needs to adapt to the chosen level of categorization and to tweak the support threshold accordingly.

REFERENCES

[1] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases" In Proceedings of the 1993 ACM SIGMOD international conference on Management of data, SIGMOD '93, pp. 207-216, New York, NY, USA, 1993

[2] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo, "Advances in knowledge discovery and data mining", pp. 307-328. American Association for Artificial Intelligence, Menlo Park, CA, USA, 1996

[3] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation", In SIGMOD Conference '00, pp. 1-12, 2000

[4] J. Pei, J. Han, and R. Mao. Closet, "An efficient algorithm for mining frequent closed itemsets", In ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery'00, pages 21-30, 2000

[5] K. Gouda and M. J. Zaki. Genmax, "An efficient algorithm for mining maximal frequent itemsets", Data Min. Knowl. Discov., pages 223-242, 2005

[6] P.-N. Tan, V. Kumar, and J. Srivastava, "Selecting the right objective measure for association analysis", Inf. Syst., 29:293-313, June 2004

[7] M. J. Zaki, "Generating non-redundant association rules", In 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Aug 2000

[8] G. I. Webb. Self-sufficient itemsets, "An approach to screening potentially interesting associations between items", ACM Transactions on Knowledge Discovery From Data, 4:1-20, 2010

[9] M. Vranić, "Designing concise representation of correlations among elements in transactional data", PhD thesis, FER, Zagreb, Croatia, 2011

[10] J. Han and M. Kamber, "Data mining: concepts and techniques", The Morgan Kaufmann series in data management systems, Elsevier, 2006

[11] M. Vranić, D. Pintar, and Z. Skočir, "Generation and analysis of tree structures based on association rules and hierarchical clustering", In Proceedings of the 2010. Fifth International Multi-conference on Computing in the Global Information Technology, ICCGI '10, pp. 48-53, IEEE Computer Society, Washington, DC, USA, 2010

[12] M. Vranić, D. Pintar, and D. Gamberger, "Adapting hierarchical clustering distance measures for improved presentation of relationships between transaction elements", Journal of Information and Organizational Sciences, vol. 36, No. 1, Varaždin, Croatia, 2012., in press.

[13] C.L.Blake and C.J.Merz, "UCI repository of machine learning databases", 1998. http://www.ics.edu/~mlearn/MLRepository.html accessed February 1st, 2012.