

Generation and Analysis of Tree Structures Based on Association Rules and Hierarchical Clustering

Mihaela Vranić, Damir Pintar, Zoran Skočir
Faculty of Electrical Engineering and Computing
Zagreb, Croatia

e-mail: mihaela.vranic@fer.hr, damir.pintar@fer.hr, zoran.skocir@fer.hr

Abstract— Detailed inspection of transactional data can reveal various useful information, in which of special importance are relationships between transaction elements. Hierarchical clustering coupled with specific distance measures reveal those relationships from one angle. Additionally, association rules - a natural method of inspecting transactional data - is able to reveal relationships between each pair of transaction elements. With transactional data modulation and multiple usages of this method a tree-like structure can be created. This paper concerns the interpretation of resulting structures for each method as well as providing comparison between them. For each algorithm mathematical base is introduced along with an explanation of result interpretation. Performances of two methods are compared and examined on a real life data set.

Keywords - *Unsupervised learning; association rules; hierarchical clustering; distance measures; tree structures.*

I. INTRODUCTION

Transactional data most commonly describes events through following important segments: transaction initiator, transaction objects (elements or items) and timestamp denoting when the transaction occurred. Many industries (e.g., retail, insurance, telecommunication, medicine) produce and store large quantities of transactional data. There are already developed and well described methods used for transactional data examination [1][2]. Work presented in this paper extends those methods usage by forming a treelike structure connecting transaction objects in a specific way.

Detailed inspection of transactional data can reveal lots of useful information regarding relations between objects occurring in examined transactions. This relation can be referred to as a notion of 'distance' between objects, given that this distance complies with certain predefined characteristics: objects that co-occur in transactions more frequently are 'nearer' - in other words, distance between them is smaller. Existence of many transactions with occurrence of one object and absence of other will result in distance measure between those objects acquiring higher values. Based on these distances transaction elements and element groups can be connected to form a treelike structure. Elements found to be near other elements should connect earlier, while those that do not commonly co-occur in transactions will get connected later in the treelike structure (nearer to its root). Work described in this paper focuses predominantly on the process of procuring described

formation and providing reasoning for its proper interpretation.

Distance between objects has practical connotations. In the retail industry, information about objects' closeness can help ensure more effective placement of products on the shelves. Mentioned findings could also be used to discover connections that can lead to revelations of hidden and interesting causalities. Generally, depending on data characteristics and final goals, different actions could be undertaken after analyzing the results - treelike structures - of proposed approaches.

There are already many tools offering graphical tree-like presentation of association rules. Some tools enable the analyst to pick an antecedent or consequent, which results in rules satisfying this condition to be displayed to him/her in certain manner (e.g., through Oracle Data Miner - a tool bundled with Oracle Database 11g Enterprise Edition [3]). Other tools display tree-like structures where rule's antecedents appear as tree roots while consequents form branches or more complex trees in cases of multiple antecedents and consequents (e.g., Orange - a component-based data mining and predictive modeling cross platform software suite available at [4]). One drawback of such presentation is that the same rule - describing connection between two attributes - often appears in more than one place in a tree, which severely diminishes readability of the structure. The main idea behind this paper as well as its most important goal is to enable the analyst to easily construct and interpret a tree structure where each item will always appear only once, henceforth greatly improving overall readability and helping him/her to discover useful causalities more efficiently.

Here described approach can also help identify potential self-sufficient itemsets, which are defined in [5] as itemsets whose frequency cannot be explained solely by the frequency of either their subsets or their supersets. These itemsets might on their own be of bigger interest to the analyst than the representation of correlations between items in the form of association rules.

Paper is organized as follows. In Section II methods chosen to handle given problem are introduced along with basic reasoning why they were chosen. Transactional data on which results are presented is also introduced along with tools found appropriate to work with for implementation of specific method. In Section III, after short theoretic introduction of association rules and related parameters, approach of multiple usage of basic method for each level of

final treelike structure construction is presented. For each step mathematical base of reasoning is given. Work of the method is at least presented on real life data set. Section IV is reserved for hierarchical clustering where results gained by different distance measure choice are presented. Section V is dedicated to comparison and discussion of two approaches and their results followed by the outline of future work. Finally, the conclusion is given in Section VI.

II. TRANSACTIONAL DATA, AVAILABLE METHODS AND TOOLS

To achieve the desired goal of forming a treelike structure as a representation of transactional objects, data set for testing and evaluation was chosen along with data mining methods and appropriate tools.

A. Transactional data set used for testing and evaluation

Guided by the thought that real life datasets are best for different solutions comparison, the decision to work with commonly available real life data set was made. Chosen dataset presents purchasing transactions of computational equipment and is currently bundled with Oracle database software as one of the example datasets in the *sh* schema. Dataset consists of 940 transactions and 14 available transaction elements – computational equipment parts. Average number of items per transaction is 2,98. Most frequently appearing item occurs in 32% of all transactions. Good insight in transaction elements is facilitated through discussion of results (Sections III and IV).

Dataset with limited number of transactional elements was chosen deliberately, to enable better comparison between approaches. Generally speaking, size of datasets will affect overall performance, but main conclusions related to the resulting treelike structure remain the same.

B. Choice of unsupervised learning methods

Association rules is a method developed for mining transactional data and is a natural choice to work with when finding solutions that will in turn build a tree structure based on transactional data. One of the commonly known drawbacks of this technique is that it often discovers a very large number of rules, many of which are trivial, uninteresting or simply variations of each other. Domain experts have to go through many of them to draw a meaningful conclusion that could be acted upon. Realizing the problem of great quantity of association rules, investigations on redundancy elimination were done and results are presented, e.g., [6][7]. This paper proposes a different approach. In the following sections, a modified and widened usage of this method will be presented, with special care given to discuss why input data must be modified with every subsequent execution of the method for each newly constructed level of the tree structure. Detailed examination of some issues describing this matter can be found in [8].

Other commonly used unsupervised learning method is **hierarchical clustering** (nicely presented in [1]), which results in dendograms – treelike structures. They are constructed on the bases of some predefined measures: **distance measures** calculated for every pair of

examples/attributes (depending on the data at hand and wanted interpretation) and **linkage criteria** that is used to further connect already formed groups. By adapting these measures to specific variables appearing in data set, analyst can accomplish better understanding of the set and come up with connections between objects or attributes that make the best sense to him/her. A detailed discussion on influence of different distance measures on the results of hierarchical clustering is given in [9], which presents solutions for distance calculation of examples described by mixed variable type attributes. Transactional data demands special treatment and distance calculation between attributes. Further development of these methods and measures will be presented in following sections.

C. Development Tools and Technologies

For two basic approaches different tools were chosen. Application of association rules, which is known as demanding when it comes to processing time and power, was conducted within Oracle Database 11g Enterprise Edition with inbuilt data mining functionalities known as ODM (Oracle Data Mining). Choice of this technology brought us certain important advantages: data mining models are treated as objects inside the database and data mining processes use inbuilt database functions to maximize scalability and time efficiency. ODM offers a few application programming interfaces that can be used to develop the final application, such as PL/SQL and Java API. Decision to use PL/SQL and direct access to database objects that keep various information on final association rules model was made, due to flexibility it offers an opportunity to tweak the finally developed algorithm to best suit our needs. Algorithm was stored as a script containing both PL/SQL blocks and DDL instructions. Java was chosen to further develop the aforementioned developed algorithm and tweak parameters for every step of tree level generation.

For the other approach, based on hierarchical clustering, Orange data mining tool was used. This is an open source tool for machine learning and data mining based on Python script libraries. Its flexibility and modular structure enabled us to modify an existing widget (program module *Attribute Distance Widget*) while other functionalities were available for use (providing data import functionality, hierarchical clustering resulting in dendogram etc.).

Although approaches to the treelike structure generation were developed through different technologies, the results are still comparable. With reasoning described in further sections, it is possible to realize either of them through other technologies.

III. ASSOCIATION RULES

This section covers usage of association rules to form a treelike structure. First the basic method is introduced, then its multiple usages in forming a tree are explained and finally results on a real life data set are presented.

A. Basic usage of association rules

Association rules acquirement (as presented in [2][10]) is an unsupervised data mining method that results in

probabilistic statements of a simple form: if A then B (or simply $A \Rightarrow B$). It actually depicts co-occurrence of certain items in transactions. Following terminology will be used:

$I = \{i_1, i_2, \dots, i_n\}$ - a set of n binary attributes called items

$T = \{t_1, t_2, \dots, t_m\}$ - a set of transactions; each transaction t_j has a unique ID and contains a subset of attributes in I

For every rule following must be valid: $A, B \subseteq I$; $A \cap B = \emptyset$.

A and B are sets of items (shortly itemsets). A is called the antecedent, and B the consequent of the rule. Most common usage of association rules method is so-called 'market basket analysis' where occurrence of one product in market basket often leads to conclusion that with certain probability some other product will also end up in it. In our case 'I' represents a collection of products (14 hardware items) that can be bought in the computer store, and each t_j collects some of the available products depending on the realized transactions by customers. However, depending of the data at hand, association rules can bring to surface various types of regularities.

Amongst all rules that can be formed, only those with minimum interest or significance should count as the result of the analysis. Most commonly known measures that act as minimum thresholds for rules formation are 'support', 'confidence' and 'lift'. Support of itemset A is the probability of appearance of itemset A in a random transaction while confidence is the probability of itemset B appearing in a specific transaction if we know that itemset A is present in it. Most confident rules are generally the most valuable ones. Mathematical expressions used to calculate these parameters:

$$\text{supp}(A) = p(A) \quad (1)$$

$$\text{conf}(A \Rightarrow B) = \frac{\text{supp}(A \cap B)}{\text{supp}(A)} = \frac{p(A, B)}{p(A)} \quad (2)$$

There are some cases where confidence works poorly compared to the random choice of transactions where certain item appears. That is the reason for introducing a new measure: 'rule lift'.

$$\text{lift}(A \Rightarrow B) = \frac{p(A, B)}{p(A)} * \frac{1}{p(B)} \quad (3)$$

Algorithms first find all itemsets that satisfy given support, and then rules that could be formed from this itemsets are examined to check whether they satisfy minimum confidence. Real issue is the first step since number of candidate itemsets grows exponentially with number of items. Among many algorithms for generating association rules presented over time - Apriory algorithm, implemented in ODM, will be

used in the first approach of forming a tree structure.

B. Association rules on multiple levels –tree generation

Following the decision to use inbuilt data mining functionalities within the database, the data resulting from association rules needs to possess certain characteristics:

- it should be straightforward and easily transferable to the tree-like structure
- it should be easily readable
- strong preference that every algorithm step (which generates one tree level) corresponds to one data structure (for example column in a relational table).

The final model is saved in a relational table with each row corresponding to one item and added columns each depicting one tree level (which can be seen in result discussion paragraph (III.D. Results presentation)). Each tree level requires generation of a new association rules model. First tree level (closest to the leaves) will capture items that are most closely related (closest in the terms of the goal presented in the introduction). For every following level parameters of association rules generation will be loosened to the extent permitted by the analyst. Depending on the starting dataset and enforced parameters certain number of levels will be created. It is not necessary to reach the level presenting the tree root.

Data has to be adjusted for each level creation except for the first level. Overall application functionality is presented in Fig.1. At algorithm start-up, analyst is required to specify the following boundaries: top and bottom support and confidence (measures used in association rules methods) together with acceptable number of tree levels (L). Top boundaries are used to form the first level of item tree. All items included in generated association rules with support and confidence greater than parameters used as input for each corresponding step are connected. Through following steps those starting measures are gradually lowered until they reach bottom support and confidence.

The actual meaning behind bottom support and confidence is the fact that analyst is not interested in connections between items that are not supported by certain proportions of transactions.

C. Possible issues

There are few possible issues regarding some decisions on overall algorithm functioning.

1) Transactional data modulation and its consequences

First issue is the one concerning the changes of transactional data. Data is changed to enable performing each consecutive step and to avoid losing information about item 'connections'. The real question is whether the final

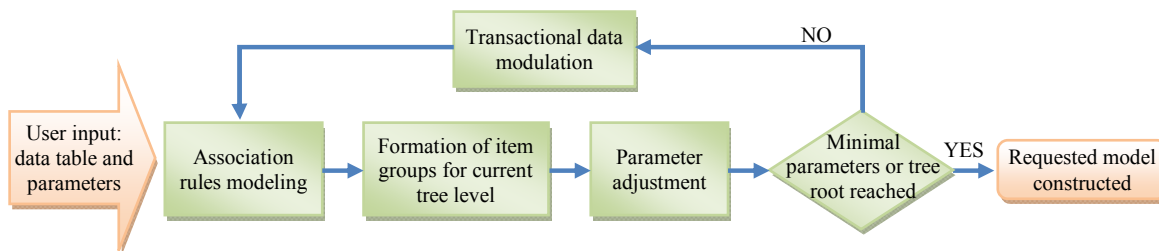


Figure 1. Developed algorithm shown through main functionality blocks

measures (produced by models after transactional data change) are acceptable and how the tree structure should be interpreted.

Implemented algorithm in each transaction replaces original item ID with the newly assigned group mark. Since association rules measures are important for the next level of tree structure generation, a question may arise regarding the consequence of replacing a group of items with one sole item that will represent the entire group.

If parameters for first level creation are set in such a way that only items A and B are connected in a group called X ($\text{conf}(A \Rightarrow B)=1$), than one of the possible representations of final model is depicted on Fig.2.

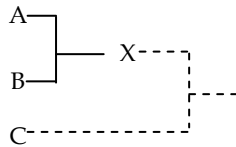


Figure 2. Possible formation of a tree structure

Two important questions may arise concerning the replacement of A and B with a group/item named X:

- what happens to the relationships of items unaffected by this new group?
- what happens to the measures of possible rules that include 'new item' X?

Regarding relationships between items unaffected by formed group, the following facts stand:

- support of those items remains the same
- support of itemsets that don't contain X remains the same (e.g., $\text{supp}(C,D)$)
- previous two entries result in no effect on the possible rules not containing X.

Depicted features are desirable and welcome. Regarding possible rules that include 'new item' X, measure calculations act as follows:

- $\text{supp}(X) \geq \max(\text{supp}(A), \text{supp}(B))$
- $\text{supp}(X) < \text{supp}(A) + \text{supp}(B)$
- support of itemsets that include X increases:
($\text{supp}(X,C) = \text{supp}((A \vee B), C) \geq \max(\text{supp}(A,C), \text{supp}(B,C))$)
- confidence of rules that incorporate X as an antecedent will fall somewhere between $\text{conf}(A \Rightarrow C)$ and $\text{conf}(B \Rightarrow C)$.
- confidence of rules that incorporate X as a consequent will be greater than $\text{conf}(C \Rightarrow A)$ and $\text{conf}(C \Rightarrow B)$ (support of (X,C) is equal or greater than $\text{supp}(A,C)$ or $\text{supp}(B,C)$ while support of antecedent remains the same).

From these observations, following reasoning could be made: item X will display closeness to some other item (e.g., item C) in the case that both of its components (A and B) display closeness to C. If A is 'close' to C but B isn't, than the appearance frequency of A and B in transactional data has to be taken into account. If appearance of A is frequent, while B is rare, than measure of closeness ($\text{conf}(X \Rightarrow C)$) will be more influenced by item A and X would be relatively

close to C. That means that X (as antecedent) exposes more average (or tenderer) features than its components with the number of supporting transactions also affecting the outcome. Explained and depicted features of grouping item X are also desirable and acceptable.

When examining X as a consequent, it can be stated that it can more easily be connected to other items than can solely items A and B, which is also logical consequence of grouping.

Final tree representation should be interpreted in the following way: based on transactional data, items A and B are close i.e., connected. If X is further connected to C then we can say that item group of A and B exposes closeness to C. However, direct connection between A and C cannot be stated (likewise for items B and C).

2) Handling transactions containing only one item

As tree structure grows and groups of items in transactions are being replaced by one – grouping item, more and more transactions containing only one item occur. Such transactions seemingly do not contribute to further analysis and could potentially be eliminated. Parameter analysis shows that elimination of such transactions does not affect confidence of rules containing grouping item as consequent. That is also the case with confidence of rules not containing grouping item. However, elimination of considered transactions results in confidence increase for rules containing grouping item as antecedent. Therefore elimination of transactions containing only one item on various levels should not be performed since it could skew the real relationships between items.

3) Different tree structures depicting the same transactional data as a consequence of different parameter input

For the same transactional data, various final tree presentations could be made depending on parameter input. Two extremes would be:

- very loose parameters
- strict parameters with slight changes from one step to another.

For the first scenario many objects would be grouped at one level, while for the second multiple level formation will be needed to bundle objects from the first scenario.

What is loose and what is strict usually depends on characteristics of the input dataset. This is an issue best left to analyst's discretion; analyst should tweak the parameters to get the structure he/she feels is the most informative and could most easily be acted upon.

D. Results presentation

Developed algorithm is used on real life dataset described in Section II. with following input parameters: minimal support. 0,1 – 0,08; minimal confidence=0,1; expected levels=3. Results are presented in Table I.

As it can be observed, there are some items that relatively rarely show up together with any other item in transactions and they stay as solitary leaves (e.g., *Y Box* etc.). Others are connected to groups of which some connect further on (groups 2_2 and 2_3 are connected to group 3_1 at level 3).

TABLE I. RESULTING MODEL FOR COMPUTER STORE TRANSACTIONS

Item_id	Name	Level_1	Level_2	Level_3
12	18" Flat Panel Graphics Monitor	1_3	2_2	3_1
9	SIMM- 16MB PCMCIAII card			
8	Keyboard Wrist Rest			
7	External 8X CD-ROM	1_2	2_3	
10	CD-RW, High Speed Pack of 5			
11	Multimedia speakers- 3" cones			
3	Standard Mouse	1_1	2_1	
4	Extension Cable			
14	Model SM26273 Black Ink Cartridge			
2	Mouse Pad	1_4		
1	Y Box	1_5		
5	Envoy Ambassador	1_6		
6	Envoy 256MB - 40GB	1_7		
13	O/S Documentation Set - English	1_10		

If we take a look at the real items that are connected, some combinations of items are quite logical, for instance: *External 8X CD-ROM* and *CD-RW, High Speed Pack of 5*.

Application testing showed that not only data characteristics influenced final model structure (number of groupings per level and level numbers) but also the nature of data in question. For the optimal use of application analyst should be closely acquainted with business problem, available data and usability of final model.

IV. HIERARCHICAL CLUSTERING

Hierarchical clustering enables treelike presentation of distances between examples (observations) or attributes. In our case transaction elements (items) are presented as binary attributes describing each transaction, which in turn is represented by one row in the table. Presence of specific element is depicted by value 1, and absence by value 0.

A. Applied attribute distance measures and linkage criteria

In *Orange* distance matrix may be created either by using the *Attribute Distance Widget* or *Example Distance Widget*. Due to the structure of our dataset, *Attribute Distance Widget* was suitable for conducting experiments. A few available measures have been examined: measures on discrete attributes (Pearson’s chi-square; 2-way interactions and 3-way interactions – last two are in detail presented in [11]) and measures on continuous attributes (Pearson’s correlation and Spearman’s correlation).

When talking about further merging of clusters (at the first level generated solely using values in the distance matrix), linkage criteria must be used. It computes the distance between clusters as a function of distances between

members of different clusters. There are three types of linkage criteria:

- single or minimum: $\min \{d(a,b): a \in A, b \in B\}$
- complete or maximum: $\max \{d(a,b): a \in A, b \in B\}$
- average or mean: $\frac{1}{|A||B|} \sum_{a \in A} \sum_{b \in B} d(a, b)$

where d is the chosen metrics; A, B already formed clusters.

Due to the nature of transactional data, experiments with Hamming distance usage were also conducted. For this purpose data was transposed and *Example Distance Widget* used. Hamming distance may be considered a natural choice to experiment with since - for a fixed length n - it’s a metric on the vector space as it fulfills the conditions of non-negativity, identity of indiscernibles, symmetry and the triangle inequality. It actually counts number of positions at which the corresponding symbols are different and - in our case - where binary values populate dataset the Hamming distance is equal to the number of ones in $a \text{ XOR } b$. In other words, if items show up together in transactions or they mutually don’t appear in them - than they are closer, which is the exact goal for the treelike structure intended to show. However, as we expected – because of the sparsity of the data set – items that showed up very rarely often demonstrated great closeness in the end results.

B. Results presentation

All introduced measures and linkage criteria were experimented with extensively (not only with here described data set, but also with other available data). Best results were reached by using Pearson’s chi-square measure. Results are presented in Fig. 3. When observing the most closely related items it may be seen they are indeed connected very early (near leaves): *Extension Cable*, *Standard Mouse* and *Mouse Pad*; *18inch Flat Panel Graphics Monitor* and *SIMM- 16MB PCMCIAII card*; *CD-RW, High Speed Pack of 5* and *External 8X CD-ROM*. However, items *Keyboard Wrist Rest* and *Y Box* are also demonstrating closeness. Detailed inspection of transactional data reveals that there is only 1 transaction containing those two items, while there are 763 transactions where both of them are absent. Based on the underlying idea of how final structure should look like, this is not a desirable feature of the final treelike structure. As further clusters are formed on the bases of the ones formed near leaves we can expect further propagation of undesirable characteristics of the structure.

When observing structure formed using the Hamming distance (Fig. 4), we can notice that items mostly absent in transactions are connected first. A connection exists between *Extension Cable*, *Standard Mouse* and *Mouse Pad* for example, but in that sparse data, characteristic showing that they quite often appear together is demonstrated much ‘later’ in a tree structure.

V. RESULTS DISCUSSION AND FURTHER WORK

Resulting structures based on two chosen approaches are quite different. Association rules, while more resource-demanding as well as in need of more direct interaction

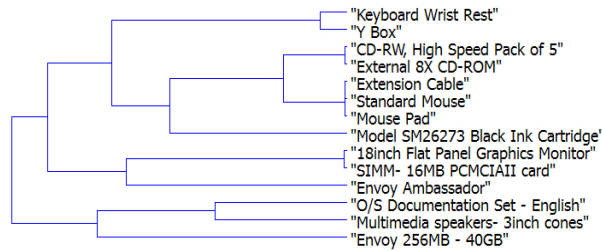


Figure 3. Dendrogram based on transactional data – usage of Pearson's chi-square distance measure and average linkage criteria

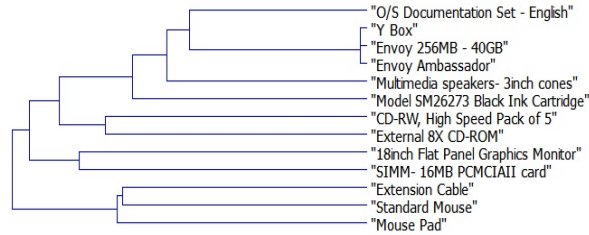


Figure 4. Dendrogram based on transactional data – usage of Hamming distance measure and average linkage criteria

between the analyst and the system, ultimately provide better, more useful results.

Hierarchical clustering offers simpler solutions, which through the usage of specific distance measures function well while constructing the first few levels of the treelike structure (those closer to the leaves), but degrade rapidly as the levels get nearer to the tree root. Data sparsity is the main cause for this, but since this is a common feature of the transactional data it's an issue one cannot easily avoid.

Main reason for difference in structures accomplished by two presented approaches is that in hierarchical clustering computation of distances is done only once and distances are calculated only for pairs of items. A linkage criterion is the one that enables computation of some kind of the distance between formed clusters but it cannot take into account facts about more than two items appearing in transactions. Including input data for linkage criteria calculation would lower the difference between resulting structures' appearance but would also significantly slow down the process and change the basic principles behind hierarchical clustering.

Considering hierarchical clustering approach, certain investigation in new measures will be undertaken together with analysis of impact of data density on the final results.

Hamming distance measure will be further investigated and tweaked to better satisfy our needs.

Association rules approach can also be improved. The most important investigation concerns parameter adjusting since it has great effect on this approach functionality and final outcome. Some researches regarding usage of data density in mining association rules have already been undertaken. Its usage is even further investigated in [12] where new measures of data density for quantitative attributes are introduced.

VI. CONCLUSION

Presented work resulted from seeking a way to present relationships between transaction elements in a manner that would enable better understanding and faster reaction of business analyst responsible for forming actionable business decisions. Two approaches were investigated and decisions that had to be made were elaborated along with their consequences.

We showed that it is feasible to broaden application of association rules and finally accomplish a tree structure that in concise manner depicts relationships between transaction objects.

Although graphical tree structure reflects main relationships between objects, one has to keep in mind that any simple form of depicting those relationships hides richer information beneath which can potentially be unearthed with other available tools or approaches.

REFERENCES

- [1] J. Han and M. Kamber, "Data Mining: Concepts and Techniques", 2nd Edn., Morgan Kaufmann, San Francisco, 2006.
- [2] D. Hand, H. Mannila, and P. Smyth, Principles of Data Mining, The MIT Press, ISBN: 026208290, USA, 2001.
- [3] Oracle Data miner:
<http://www.oracle.com/technology/products/bi/odm/odminer.html>
(last accessed on 18.06.2010.)
- [4] Orange 2.0b for Windows: <http://www.aillab.si/orange/>
(last accessed on 18.06.2010.)
- [5] G. I. Webb, "Self-sufficient itemsets: An approach to screening potentially interesting associations between items," ACM Transactions on Knowledge Discovery from Data, vol. 4, number 1, pp. 3:1–3:20, ISSN: 1556-4681, January 2010.
- [6] N. Pasquier, R. Taouil, Y. Bastide, G. Stumme, and L. Lakhal, "Generating a condensed representation for association rules", Journal of Intelligent Information Systems, vol. 24, No. 1., Jan. 2005, pp. 29-60, 0925-9902
- [7] Y. Xu and Y. Li, "Mining non-redundant association rules based on concise bases", International Journal of Pattern Recognition and Artificial Intelligence, vol. 21, No. 4, June 2007, pp. 659-675, 0218-0014
- [8] M. Vranić, D. Pintar, and Z. Skočir, "Building an application - generation of 'items tree' based on transactional data". In: Y. Zhang "Application of Machine Learning", In-Teh, ISBN: 978-953-307-035-3, Vukovar, Croatia, pp. 109-126, 2010.
- [9] S. Pinjušić and M Vranić, "Influence of distance measure choice on the results of hierarchical clustering," Proc. of 33rd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO 2010), in press
- [10] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," Proc. of the 1993 ACM SIGMOD international conference on Management of data, May 1993, pp. 207-216, 0-89791-592-5, Washington, D.C., United States
- [11] A. Jakulin, "Machine learning based on attribute interactions." PhD Dissertation, University of Ljubljana, Faculty of Computer and Information Science, 2005.
- [12] D.W. Cheung, L. Wang, S. M. Yiu, and B. Zhou, "Density-based mining of quantitative association rules," Knowledge Discovery and Data Mining. Current Issues and New Applications, 2007, pp. 257-268, Springer Berlin/Heidelberg, 978-3-540-67382-8, Germany