ISSN 1512-1232

COMPACT KNOWLEDGE DISCOVERY IN DATABASES: A CORE BASED APPROACH

Anup Bhatkar

Faculty, Dept of CSE, MITCOE-Pune, India
Anup801@gmail.com

Abstract

The purpose of Data Mining is to facilitate understanding large amounts of data by discovering interesting regularities or exceptions. Association rules are simple but powerful regularities in binary data. However, there can be easily hundreds or even more association rules holding in a data set which presents a problem in their utility and understanding. Paradoxically, data mining itself can produce such great amounts of knowledge, consequently presenting new knowledge management problem.

Many researchers realized the need for compact knowledge discovery in databases and have attempted solutions. We present 'A Core Based Approach on Large Item Sets' that succinctly summarizes the information present in the data set. A Core is a set that is found in more than one set. We use the concept of conditional entropy to measure information content in a rule.

The compactness of knowledge discovered can be identified by compactness factors like suitable number of rules, a measure of strength of a rule, degree of useful information conveyed, effect of an addition of item to rule, measure of interdependence between items and so on. Till date no single methodology answers the question of compact knowledge discovery in databases to a satisfactory level. The problem is very much open.

We end up showing that the proposed approach indeed meets some compactness factors and shows promise to satisfy the rest.

1 Introduction

1.1 Data Mining

In today world one is overwhelmed with data. One could easily testify to the growing gap between the generation of data and one's understanding of it. As the volume of the data increases, inexorably, the proportion of it that people "understand" decreases, alarmingly. Lying hidden in all this data is information, potentially useful information that is rarely made explicit or taken advantage of. Thus, looking for patterns becomes important task.

Thus, Data Mining (knowledge discovery in databases) is a field of increasing interest combining databases, artificial intelligence, machine learning and statistics. The purpose of data mining is to facilitate understanding large amounts of data by discovering interesting regularities or exceptions.

In the association rule discovery in super market database, some of the association rules that one gets are like

1. *Milk → Bread.*
2. *Diaper → Beer.*
3. *Diesel → Lubricant.*

The first rule is no new knowledge discovery. The second one is a new knowledge, and one logical explanation for the second pattern is that those who have young children, tend to drink to forget the disturbances that comes from small children being around. The third one was commercially exploited by keeping diesel and lubricant together, reducing the price of diesel but increasing the price of lubricant.

1.2 Need for II order Data Mining

Paradoxically, data mining itself can produce such great amounts of data that there is new knowledge management problem. There is need for succinct or compact summarization of the knowledge present in databases.

For example:

In the problem of distinguishing between AML/ALL, one gets 4000 good plausible classifiers. Although we mined the data the mined information doesn't satisfy the purpose for which mining was done (see [1])

Similarly, one gets number of Association Rules of the order of 10^6.

How to make sense of this mined data?

So there is need for II order data mining or compact knowledge discovery in databases.

1.3 Previous work/Literature survey

Many researchers realized the need for compact knowledge discovery in databases and have attempted solutions. A short list is given below: -

1. The issue of interestingness of discovered knowledge has been discussed in general by Piatetsky-Shapiro[2].
2. Hoschka and Klosgen[3] have used templates for defining interesting knowledge.
3. Psaila[4] defines meta pattern concept to extract quality association rules.
4. Klemettinen et al.[5] gives a regular expression template formalism to describe the structure of interesting rules.
5. Han et al.[6] present an attribute-oriented approach for pruning uninteresting relations.
6. Piatetsky et al.[7] discuss interestingness of deviation.
7. Wang et al[12] uses criteria of weighted interestingness and cover of association rules for compact knowledge discovery.
8. Srinivas et al[13] mining association rules.
9. Hamrouni et al[14] Frequent patterns using conjunctive and disjunctive search space

1.4 Organization of paper

The following is topic wise summary:

In topic 2, titled 'Motivation for our work' we discuss a problem from the field of Bio informatics, the plausible solution for it and how theory of Cores can play really a major part in solving the problem.

In topic 3, titled 'Core based Approach: Theory and Implementation', we formulate our theory of Core Based Approach on the set of Large Item Sets.

In topic 4, tilted 'Implementation Results', we discuss the results of our proposed theory.

In topic 5, tilted 'Conclusions and future work', we discuss the conclusions arisen from our work and give many pointers for taking the theory forward.

2 Motivation for the Core based approach

2.1 A problem in bio informatics

This problem is part of research paper 'Mining Microarray Expression Data for Classifier Gene-Cores' by Dr. Raj Bhatnager, Dept of ECECS, University of Cincinnati [1].

Recently there has been a lot of interest in classifying various tumors using gene expression data. It has been shown that gene expression data acquired from leukemia patients can be used to build predictors that can discriminate between two acute leukemia subtypes, acute lymphoblastic leukemia (ALL) and acute myleloid leukemia (AML). To maximize the efficacy of cancer treatment while a the same time reducing it's toxicity, it is imperative to target specific therapies to path genetically distinct tumor types. Thus, improved cancer classification is valuable to advances in cancer treatment. Subtype of leukemia, ALL and ALL, have similar histopathological appearance.

Correct prediction of the subtypes of cancer at an early stage from gene expression data can vastly improve accuracy of diagnosis and effectiveness of treatment. Currently, no single test is sufficient to make a diagnosis – leukemia classification still remains imperfect. Based on gene expression data collected from 72 patients suffering from either AML or ALL, it has been shown that a large number of genes (approximately 1100) have a higher correlation with the AML/ALL distinction than can be expected by chance. Classifiers built with this small subset of genes, selected based on their individual correlation with the cancer subtypes, have been used to predict leukemia subtypes with some accuracy. Since there is limited knowledge about the functional relevance of most genes, and a strong possibility exist of noise and biasing by inter gene dependence and gene interactions, the selection of set of genes that constitute a good and biologically meaningful classifier is very important. What genes should be chosen to form the classifier and how many genes should be included are the two important issues in designing a classifier.

2.2 Exploratory analysis of data

Most statistical pattern recognition methods require an abundance of data in order to generate inferences with sufficient confidence. Typically, a dataset on which statistical methods would be relevant consists of a large number(upwards of 1000s) of data points and very few variables(orders of 10s) beings measured for each data point. In the case of gene expression datasets, each data point consists of few thousand measured variables (genes) and few order of tens data points. An exploratory analysis of such a dataset requires that we generate a large number of plausible hypotheses (good classifiers, in our case) and examine them. Any inference of high confidence can be based only on its consistency with a large fraction of plausible hypotheses. This approach was adopted.

The adopted strategy works in the following two phases

During the first phase a heuristic search to identify a large number of small-sized subsets of genes that discriminate between classes better than similar sized subsets. During the second phase these classifiers have been trained by finding suitable weight vectors to optimize the discrimination potential of each subset. From relatively large population of such trained classifiers better performing individual classifiers were selected.

Using the above methodology, it was demonstrated for the ALL-AML case, existence of gene sets that have 100% accuracy in predicting the subtype of leukemia.

One of the main thrust behind the strategy is the belief that it is not enough to look at the correlation between individual genes and the classes, but the effect of set of genes as a whole, given the possibility of inter-gene dependencies.

It was found out that there are about 4000 good classifiers. Now, among these set of classifiers, which classifier or a small set of classifier should be used to distinguish between AML and ALL?

2.3 Core: Frequent Patterns in Classifiers

To glean insight from a large set of well-performing classifiers, an algorithm to extract or mine cores from the set of classifiers was introduced. A core is defined as a subset of genes that is an integral part of several distinct good classifiers. The number of candidate classifiers in which the core appears defines the strength of the core.

The presence of many such different cores across many different classifiers can indicate that there may be several different processes leading to AML/ALL distinction. A core itself is an indication that the genes constituting it are strongly related, at least in the context of the processes leading to the AML/ALL distinction. The examination of the cores mined can lead to the better understanding of the working of the genome with respect to tumors.

Thus these cores of genes are potentially very useful for a biologist and may reveal much about inter-gene dependencies and gene functions.

It was found out that many of the mined cores where themselves very good classifiers. Also biologist confirmed that the genes present in some of the cores do have strong inter-relationships.

ISSN 1512-1232

Thus justifying the immense knowledge that can be extracted from the set of cores as pointed out in the previous paragraph.

3 Core based approach: Theory and Implementation

3.1 Focus of our work

The need for compact knowledge discovery in databases has been discussed previously. Also in the last Topic we discussed the promising potential of a Core based approach. Our work focused on using a Core based approach on set of large item sets to do a compact knowledge discovery.

Association rules generated from a set of large item sets don't answer questions like:

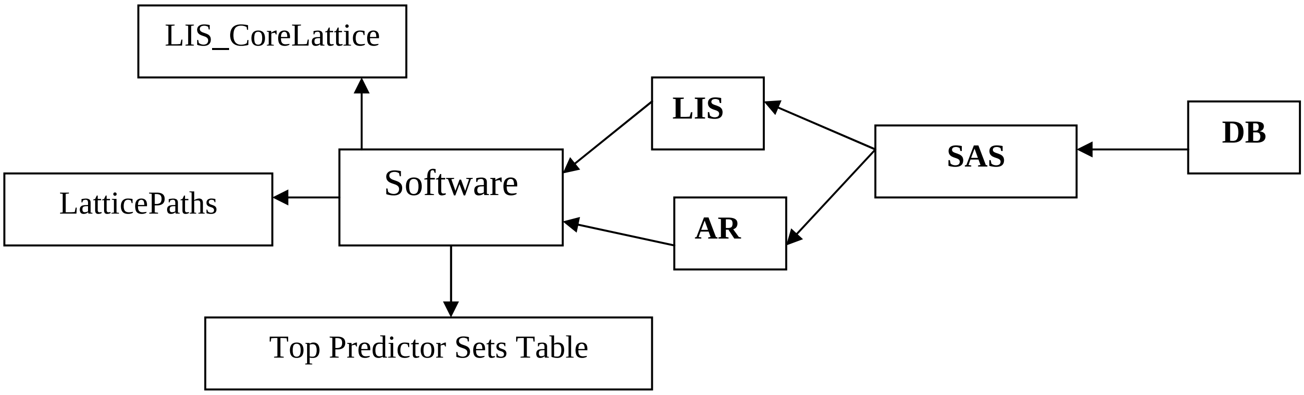
1. Given an item, which is, it's most important predictors?
2. How is the prediction power of a set of items for a given item change with the addition of one or more items to the set.
3. What is the interdependence between items?

and so on.

Thus we are basically continuing forward the Core based solutions mentioned in [1].

3.2 Architecture of system

3.2.1 Figurative design



Now, we will explain the architecture from right to left, that's the sequence in time also.

3.2.2 Database(DB)

We needed a transactional database, which can give sufficiently large number of association rules. We found the required database in SAS system libraries. It's a super market database. The database gave 7000 association rules at 5% support and confidence.

3.2.3 SAS

We used SAS system to generate large item set and association rules. This saved time and also the required job was done efficiently and without mistakes.

3.2.4 Large item set file(LIS)

This is the exported text file of large item sets generated through SAS system when the association rule discoveries were applied on the database.

A sample file will look like this (the same sample example will be used throughout as that facilitates good explanation.):

heineken
cracker
heineken cracker

soda olives heineken cracker baguette

3.2.5 Software

This is the software which takes large items sets text file as input, parses it and outputs LIS_CoreLattice, LatticePaths and Top PredictorSets-Items table.

3.2.6 Large item set core Lattice(LIS_CoreLattice)

The sample file shown above represent the 0th level of the lattice. The ith level of lattice is construct from (i-1)th level in the following way.

An element in (i-1)th level goes to ith level if there exist an element in the (i-1)th level which is proper super set of the element in question. Mathematically it can be written as:

$\text{LatticeLevel}_i = \{x \mid x \in \text{LatticeLevel}_{(i-1)}$ and $\exists\ y \in \text{LatticeLevel}_{(i-1)}$ s.t. $x \subset y\}$.

The lattice will have i levels if (i +1) level has zero elements.

The LIS_CoreLattice for the above sample file is as follows:

LatticeLevel_0	**LatticeLevel_1**	**LatticeLevel_2**	
Heineken	*heineken*	*heineken*	
Cracker	*cracker*	*cracker*	
cracker, heineken	*cracker, heineken*		
cracker, heineken, soda, olives, baguette			

The elements present farther in the lattice level are more frequently occurring.

3.2.7 Conditional entropy of a set

Before we move on further in the architecture, it’s very important to understand the concept of conditional entropy. We use conditional entropy as a measure of information. The definition of conditional entropy captures the information changes taking place in a set by addition of items in the set.

The definition is as follows:

LIS = $[I_1, I_2,\dots,I_n]$ be a large item set then

CE = $[CE_1, CE_2, \dots\dots CE_n]$ is the conditional entropy of set LIS where

CE_1 is defined as follows:

$P_1 = \text{count}(I_1, I_2\dots I_n) / \text{count}(I_2, I_3,\dots I_n)$.

$P_2 = \text{count}(\sim I_1, I_2\dots I_n) / \text{count}(I_2, I_3,\dots I_n)$.

$P_3 = \text{count}(I_1, \sim(I_2\dots I_n))/\text{count}(\sim(I_2, I_3,\dots I_n))$.

$P_4 = \text{count}(\sim I_1, \sim(I_2\dots I_n))/ \text{count}(\sim(I_2, I_3,\dots I_n))$.

$\text{count}(I_1, I_2,\dots,I_m)$ = # times $\{I_1, I_2,\dots,I_m\}$ occurs in set $G(I_1, I_2,\dots,I_n)$.

$G(I_1, I_2,\dots,I_n) = \{x \mid x \in \text{LatticeLevel}_0\}$ **if n=1**
$= \{x \mid x \in \text{LatticeLevel}_0$ and $I_i \subset x$ and $1 \le i \le n\}$ **if n>1**

$\text{ModifiedLog}(P_i) = 0$ **if $P_i = 0$**
$= \log_4 P_i$ **if $P_i > 0$**

~ is the symbol for negation.

$CE_1 = -(\Sigma\ (P_i * \text{ModifiedLog}(P_i)))$ where $1 \le i \le 4$.

Similarly we can define CE_2, CE_3, …,CE_n.

Thus we can write: $[I_1, I_2,….,I_n] =>_{ce} [CE_1, CE_2, ….. CE_n]$

Informally saying the value of CE_i tells us the information that other elements of set has about the item I_i.

From the sample example file we can give the following illustrations:

Example 1:

LIS = [heineken]
CE = $[CE_1]$
G([heineken]) = {
{ heineken },
{ cracker},
{cracker, heineken},
{cracker, heineken, soda, olives, baguette }
}

Calculating CE_1: -
P_1 = count(heineken, ())/count()
= 3/4
= 0.75
P_2 = count(~heineken, ()) / count()
= 1/4
= 0.25
P_3 = count(heineken, ~())/count(~()).
= 3/4
= 0.75
P_4 = count(~heineken, ~()) / count(~()).
= 1/4
= 0.25

CE_1 = - (Σ P_i ModifiedLog(P_i)) where $1 \le i \le 4$.
= 0.810

Thus we can write the rule: **[heineken] =>$_{ce}$ [0.810]**

Example 2:

LIS = [heineken, cracker]
CE = $[CE_1, CE_2]$
G([heineken]) = {
{ heineken },
{ cracker},
{cracker, heineken},
{cracker, heineken, soda, olives, baguette }
}

Calculating CE_1:
P_1 = count(heineken, (cracker))/count(cracker)
= 2/3
= 0.66
P_2 = count(~heineken, (cracker))/count(cracker)
= 1/3
= 0.33
P_3 = count(heineken, ~(cracker)) /count(~(cracker)).
= 1/1

= 1
P_4 = count(~heineken, ~(cracker))/ count(~(cracker)).
= 0/1
= 0
CE_1 = - (Σ P_i ModifiedLog(P_i)) where $1 \leq i \leq 4$.
= 0.464
Similarly,
CE_2 = 0.464
Thus we can write the rule: **[heineken, cracker] =>** $_{\mathbf{ce}}$ **[0.464, 0.464]**

Example 3:

LIS = [heineken, cracker, soda, olives, baguette]
CE = [CE_1, CE_2, CE_3, CE_4, CE_5]
G([heineken]) = {
{ heineken },
{ cracker},
{cracker, heineken},
{cracker, heineken, soda, olives, baguette }
}

<u>Calculating CE_1</u>:
P_1 = count(heineken, (cracker, soda, olives, baguette)) / count(cracker, soda, olives, baguette)
= 1/1
= 1
P_2 = count(~heineken, (cracker, soda, olives, baguette)) / count(cracker, soda, olives, baguette)
= 0/1
= 0
P_3 = count(heineken, ~(cracker, soda, olives, baguette)) / count(~(cracker, soda, olives, baguette))
= 2/3
= 0.66
P_4 = count(~heineken, ~(cracker, soda, olives, baguette)) / count(~(cracker, soda, olives, baguette))
= 1/3
= 0.33
CE_1 = - (Σ P_i ModifiedLog(P_i)) where $1 \leq i \leq 4$.
= 0.464
Similarly,
CE_2 = 0.464
CE_3 = 0
CE_4 = 0
CE_5 = 0
Thus we can write the rule:
[soda, olives, baguette, heineken, cracker] => $_{\mathbf{ce}}$ **[0, 0, 0, 0.464, 0.464]**

3.2.8 Large item set core lattice paths(LatticePaths)

Now, we want to traverse all the paths from right to left in the lattice and see how the information in the large item set changes. The measure of information is the concept of ***conditional entropy***. The LatticePaths file for the sample example would look like this: -

(Note: - Square bracket represent the large item set. Each item is given an integer identifier for the sake of readability. In the Appendix A integer identifier for each item is given. The values in the curly braces represent the values of the conditional entropy).

[1] =>$_{ce}$ [0.810]	[1, 2]=>$_{ce}$ [0.464, 0.464]	[3, 4, 5, 1, 2]=>$_{ce}$ [0, 0, 0, 0.464, 0.464]
[2]=>$_{ce}$ [0. 810]	[1, 2]=>$_{ce}$ [0.464, 0.464]	[3, 4, 5, 1, 2]=>$_{ce}$ [0, 0, 0, 0.464, 0.464]
[1]=>$_{ce}$ [0. 810]	[3, 4, 5, 1, 2]=>$_{ce}$ [0, 0, 0, 0.464, 0.464]	
[1]=>$_{ce}$ [0. 810]	[1, 2]=>$_{ce}$ [0.464, 0.464]	
[2]=>$_{ce}$ [0. 810]	[3, 4, 5, 1, 2]=>$_{ce}$ [0, 0, 0, 0.464, 0.464]	
[2]=>$_{ce}$ [0. 810]	[1, 2]=>$_{ce}$ [0.464, 0.464]	
[1]=>$_{ce}$ [0. 810]		
[2]=>$_{ce}$ [0. 810]		

Table 3.1. LatticePaths table

3.2.9 Top Predictor Sets table

Now, from the LatticePaths file we want to extract top predictor for a given item, for all the items. For the current running sample example the TopPredictorSet-Item table will look like this:

(Note: - The column would represent the items)

Heineken	**Cracker**	**Soda**	**olives**	**Baguette**
[2] =>$_{ce}$ 0.464	[1] =>$_{ce}$ 0.464	[4,5,1,2] =>$_{ce}$ 0	[4, 5, 1, 2] =>$_{ce}$ 0	[4, 5, 1, 2] =>$_{ce}$ 0
[3,4,5,2] =>$_{ce}$ 0.464	[3,4,5,1]=>$_{ce}$0.464			

Table 3.2. Top Predictor set table.

4 Implementation Results

The whole architecture of the system as described in the previous Topic was implemented successfully in JAVA 1.5.

Large item set core lattice file was successfully generated according to the theory mentioned in the previous Topic. The lattice had 5 levels. So it was a fast convergence.

Large item set core lattice paths file was also successfully generated enumerating all the possible paths that exist in core lattice. It was sorted in the increasing order w.r.t. conditional entropy. Increasing order was chosen for the simple reason that minimum entropy means maximum information.

By observing the file one could see that in most cases addition of item to the predictor set increases the information/prediction capacity of the predictor set (as one intuitively would expect).

This results paves way to answering important questions that were raised in section 3.1.

Top Predictor Sets table was also successfully generated by scanning the Lattice paths. We outputted top 100 predictors for an item. Now instead of 7000 association rules we have 100 rows and that too with much richer information content as confirmed by expert in supermarket domain.

Also in the next Topic many ideas with some details are given for measuring the interdependence between items, core strength, how prediction measure of an item changes with addition of attribute and areas where this approach can be very effective.

Time complexity of the program is $O(n^4)$ where n = total number of large item set. This is a very high upper estimate.

We get this time complexity from the fact that maximally in the program there is full lattice scan for each element in the lattice and maximally lattice can contain $O(n^2)$ elements.

5 Conclusions and future work

We started by seeing the important role that data is playing in our lives and the importance of mining this data. We also saw that not just any arbitrary mining will yield meaningful results; we have to mine data in a compact way. With the problem of II order data mining, we went on to see the potential of Core based approach for solving the II order data mining problem and finally, formulated a core based theory for solving it. We successfully applied the theory in case of Association rule discovery.

Our theory can still be refined (we give many pointer in the next paragraph in this context.) It’s not that we have solved all the problems in data mining but we surely have initiated a novel and valid way of mining data.

We have just touched the tip of iceberg. We believe that a wealth of information can be extracted from the Lattice paths. The following are some of the suggestions in this regard:

1. How addition of item to the predictor set affect prediction for a given item
   This may lead to power set of options so statically outputting this information in a tabular form may not be possible unless the number of distinct items are very small (<10). But this functionality can be achieved dynamically. At run time input the predicted item and predictor set and output the information change that takes place for the predicted item by the addition or deletion of items in the predictor set.

2. Interdependence between attributes
   Consider the following example:
   B $\rightarrow_{ce}$ A : 0.70
   BC $\rightarrow_{ce}$ A : 0.65
   BCD $\rightarrow_{ce}$ A : 0.20
   From the above example one can conclude that either BD or CD are strongly related. Such observation can point out to which items are strongly related and which are not.
3. Core strength
   One can also think of some mathematical way of measuring strength of a core.

4. Domain knowledge
   One can use domain knowledge in form of templates (as used in klemettin’s work) or any other form on the top of core based approach to refine one’s results.

5. Bio-informatics
   Discovering the functionality of genes and the interdependence between genes is time consuming, expensive as well as difficult process. The theory developed in this project can prove to be very handy tool in solving many of the problems encountered in the field of Bio-informatics.

6. Time complexity
   The focus of our project was mainly to formulate a theory which can mine data in compact way. There is room for efficient implementation of the proposed theory thus improving the time complexity.

ACKNOWLEDGEMENTS

I would like to acknowledge Dr. S.K.Gupta, Prof IIT-Delhi, CSE dept and Dr. Raj Bhatanagar, Prof. University of Cincinnati, USA for guiding me this work. This work was done as part of M.Tech thesis at Indian Institute of Technology, Delhi.

REFERENCES

[1] Dr. Raj Bhatnagar et al. ‘Mining Microarray Expression Data for Classifier Gene-Cores’, BIOKDD01: Workshop on Data Mining in Bioinformatics (with SIGKDD01 conference).

[2] Gregory Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. In Gregory Piatetsky-Shapiro and William J. Frawley, editors, Knowledge Discovery in Databases, pages 229-248. AAAI Press / The MIT Press, Menlo Park, CA, 1991.

[3] Peter Hoschka and Willi Klosgen. A support system for interpreting statistical data. In Gregory Piatetsky-Shapiro and William J. Frawley, editors, Knowledge Discovery in Databases, pages 325-345. AAAI Press / The MIT Press, Menlo Park, CA, 1991.

[4] Guiseppe Psaila. Discovery of Association Rule Meta-Patterns, DaWaK 1999: 173 – 182

[5] Mika Klemettinen et al. Finding Interesting Rules from Large Sets of Discovered Association Rules. Appeared in the Third International Conference on Information and Knowledge management, Ed. Nabil R. Adam, Bharat K. Bhargava and Yelena Yesha, 401 – 407. Nov 29 – Dec 2, 1994, Gaithursburg, Maryland. ACM Press.

[6] Jiawei Hans, Yandong Cai, and Nick Cercone. Knowledge discovery in databases: an attribute oriented approach. In Proceedings of the 18th International Conference on Very Large Databases (VLDB), pages 547 – 559, August 1992.

[7] Gregory Piatetsky-Shapiro and Christoper J. Matheus. The interestingness of deviation. In Usama M. Fayyad and Ramasamy Uthurusamy, editors, AAAI Workshop on Knowledge Discovery in Databases, pages 25 – 36, Seattle, Washington, July 1994.

[8] R.Agrawal, T.Imielinski and A.Swami. Mining association rules between sets of items in large databases. In *Proc. 1993 ACM SIGMOD Int. Conf. Management of Data (SIGMOD’93),* pages 207-216, Washington, DC, May 1993.

[9] R.Agrawal, H.Mannila, R.Srikant, H.Toivonen and A.I.Verkamo. Fast discovery of association rules. In U.M.Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy, editors, *Advances in knowledge discovery and Data Mining*, pages 307-328. AAAI/MIT Press, 1996.

[10] D.W.Cheung, J.Han,V.Ng and C.W.Wong. Maintenance of discovered association rules in large databases: An incremental updating technique. In *Proc. 1996 Int. Conf. Data Engineering (ICDE’96),* pages 106-114, New Orleans, La, Feb. 1996.

[11] R.Meo, *A New Approach for the Discovery of Frequent Itemsets,* Proc. of the International Conference on Data Warehouse and Knowledge Discovery, August/September 1999, Firenze, Italia, Springer-Verlag LNCS 1676, pp. 193-202.

[12] De-Xing Wang Xue-Gang Hu Xiao-Ping Liu Hao Wang Jun Guo , *Research on model of association rules mining with added-newly measure criteria*, Machine Learning and Cybernetics, 2004. , IEEE On page(s): 1187- 1190 vol.2.

[13] Srinivasan, V. Aruna, M. *Mining association rules to discover calendar based temporal classification* Computing, Communication and Networking, 2008. ICCCn 2008. International Conference on, IEEE On page(s): 1-12

ISSN 1512-1232

[14] Tarek Hamrouni, *Mining Concise Representations of Frequent Patterns through Conjunctive and Disjunctive Search Spaces*, Crossroad Summer 2009 ACM page 24

Appendix A. Integer identifier for items

Item	**Integer identifier**
Heineken	1
Cracker	2
Soda	3
Olives	4
Baguette	5

Article received: 2009-11-17