On-line data mining, concept hierarchy, frequent patterns, cover, obviosity.

Petr CHMELAR[1], Lukas STRYKA[1]

SIMPLIFIED PROGRESSIVE DATA MINING

There are huge amounts of data stored in databases, but it is very difficult to make decisions based on this data. We propose the OLAM SE system (Self Explaining On-Line Analytical Mining) that is similar to the Han's OLAM [5] in the idea of interactive data mining. The contribution is to simplify on-line analytical data mining to professionals, who understand their data but want more significant, interesting and useful information. It is done by shielding internal concepts (associations, classifications, characterizations) and thresholds (supports, confidences) from the user and by a simple graphical interface that suggests most relevant items.

OLAM SE determines minimum support value from required cover of data with usage of entropy coding principle. This is automatically applied on the structure based on given conceptual hierarchy where present. We also determine the maximum threshold to avoid explaining knowledge that is obvious. Major part of data is thus described by frequent patterns.

The presentation of results is realized using diagram notation similar to UML. In fact, it is a visual graph which nodes are frequent data sets presented as packages including sub packages - data concepts or items. Edges represent links or patterns between them. These patterns can be progressively explored by the user, who gets a detailed view of patterns which are attractive to him. Other possibly interesting sets are offered to the user without any other action. This is well suitable for characterization and descriptive classification equivalent to normal Bayes.

1. INTRODUCTION

Decision support problems have been motivating for a development of sophisticated tools which provide a new view on data for better data understanding for business analysis, medical and scientific research. Many of them are based on data mining techniques and OLAP warehouses to process structured data organized in multi-level conceptual hierarchies [3, 4, 5].

Standard data mining techniques are too complicated for beginners – wrong threshold causes long computational time and irrelevant results. In case of multilevel conceptual hierarchy even experts can guess the proper values only and wait repeatedly a long time. So the main challenge of simple data mining is to find the significant factors progressively and as easily as possible.

Our approach uses some principles from theory of information and simple user interactivity to determine data sets that are interesting for the data mining analysis. The main asset is to provide simple and interactive system for data mining on structured or semi-structured data. The method is proposed to process also data with given conceptual hierarchy.

* DIFS FIT, Brno University of Technology, Czech Republic, chmelarp@fit.vutbr.cz, stryka@fit.vutbr.cz.

1.1. ORGANIZATION OF WORK

The following chapter is dedicated to the Han's OLAM principles based on OLAP and data mining techniques. Chapter 3 and 4 describes our contribution. We propose OLAM SE system. There are two main improved areas compared to data mining, OLAP and Han's OLAM: (1) Simplified usefulness metrics – cover and obviosity. (2) Intuitive graphical interface based on UML diagrams that progressively suggests relevant interesting patterns. Chapter 5 is dedicated to simplified mining of various types of knowledge and the system architecture. The approach is concluded in chapter 6.

2. OLAP + DATA MINING = OLAM

On-line analytical mining (OLAM) [4, 5], also called OLAP mining, integrates OLAP [2] with data mining. OLAM takes advantage of data warehouses such as high quality of data stored in data warehouses. Thus OLAM works on integrated, consistent, and cleaned data so there are no necessary pre-processing steps. OLAM provides facilities for data mining on different subsets of data and at different levels of abstraction, by drilling, pivoting, filtering, dicing, and slicing on a data cube and on some intermediate data mining results

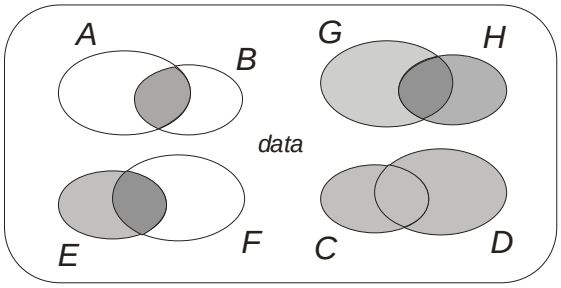
2.1. DATA CHARACTERIZATION & CLASSIFICATION

Data can be associated with classes or concepts. It can be useful to describe individual classes and concepts in summarized, concise, and yet precise terms. This description can be derived via data characterisation. Data characterisation is an aggregation of the general characteristics and features of a class under study.

Classification predicts categorical class labels and classifies data based on the model which is constructed from the training set. The classification doesn't create such labels but examines properties (often called observations) of each class in the learning phase. In the classification step the classification algorithm compares observations of an unknown object and tries to decide which class is the most suitable for those data.

2.2. FREQUENT PATTERNS & MULTILEVEL ASSOCIATION ANALYSIS

Frequent patterns are patterns that appear in a data frequently. These patterns could be itemsets, subsequences, or substructures. Frequent itemset could be milk and bread in shopping basket analysis, where milk and bread appears frequently together in a transaction data set. To determine which patterns are frequent, the minimum support metrics has to be satisfied. Illustration of support and other metrics we can use is in the figure 1. There are subsets A – H of the investigated data there, they mean data satisfying some conditions e.g. belong to some concept class.



$$support(A, B) = P(A \cap B)$$

$$cover(C, D) = P(C \cup D)$$

$$co\text{-}occurrence(G, H) = support(G, H) / cover(G, H)$$

$$confidence(E \Rightarrow F) = P(F \mid E) = P(E \cap F) / P(E)$$

$$correlation(C, D) = P(C \cap D) / P(C)P(D)$$

Fig. 1. Demonstration of usefulness metrics.

For many mining tasks, it is difficult to find strong patterns in data at low or primitive levels of abstraction. Strong associations discovered at high levels of abstraction may represent common sense knowledge. Sometimes common sense knowledge for one user may be novel for another. Therefore, methods providing capabilities for mining association rules at multiple levels of abstraction with sufficient flexibility for easy traversal among different abstraction spaces has been developed.

These methods use a conceptual hierarchy defining a sequence of mappings from a set of low level concepts (also called classes) to higher-lever, more general concepts. Conceptual hierarchy is represented by rooted tree, where nodes are general item sets, leafs are data items and the root represents most generalized abstraction of all data items. So these data can be generalized by replacing low-level concepts by their higher-level concepts in a concept hierarchy.

2.3. ON-LINE ANALYTICAL MINING

OLAM provides facilities for data mining on different subsets of data and at different levels of abstraction by OLAP operations on a data cube and on some intermediate data mining results. OLAP [2] is a data summarization/aggregation tool that helps simplify data analysis, while data mining allows the automated discovery of implicit patterns and interesting knowledge hidden in large amounts of data. The main difference between OLAP and data mining is that OLAP is based on interactive user-defined hypothesis testing while data mining is relatively slow generation of such hypotheses.

OLAM represents methods which integrate OLAP principles and data mining methods for multi-dimensional data mining in large databases and data warehouses on different granularities (different subsets or different levels of abstraction by drilling, pivoting, filtering, dicing and slicing on a data cube). An OLAM engine performs analytical mining in data cubes in a similar manner as an OLAP engine performs on-line analytical processing. This engine accepts user's on-line queries and work with the data cube in the analysis. So the engine may perform multiple data mining tasks, such as concept description, association, classification, prediction, clustering, etc. OLAM uses more powerful data cube construction than OLAP because OLAM analysis often involves the analysis of large number of dimensions with finer granularities. Construction of data cube is following: if data cube contains a small number of dimensions, or if it generalized to high level, the cube is constructed as compressed sparse array but is still stored in a relational database to reduce costs of construction and indexing of different data structures.

3. SIMPLIFIED TRESHOLDS

Standard data mining techniques are too complicated for beginners – they have to learn the concept, somehow guess threshold values and wait comparatively long time for irrelevant results – due to the wrong threshold specification and expectation. In case of multilevel conceptual hierarchy the users' confusion might go further – see example 3.1 in [3] where the author supposes support "4" for level 1 and "3" for level 2 and 3 which is mystifying not only for a common user, even experts have to guess the proper values.

The very first idea of the simplification was based on the Paretto analysis. It shows that usually 80% of consequences stem from 20% of the cause. For instance the 20% of components causes 80% of malfunctions [8]. So we identified the first problem of intuitive data mining – to find the significant factors as simple as possible. This has led to the cover parameter. The second problem of useful data mining is mining unnecessary, obvious information (eg. Sports Shop sells Sporting Goods). That's the reason we employed the obviosity metric. Technically, those interesting and significant data are derived from frequent itemsets with its support values. That is quite usual in (multi-level) association rule mining – it is supposed that the most significant data can be described by frequent itemsets [2, 6]. Description of the rest data can be done using OLAP or Bayesian classification.

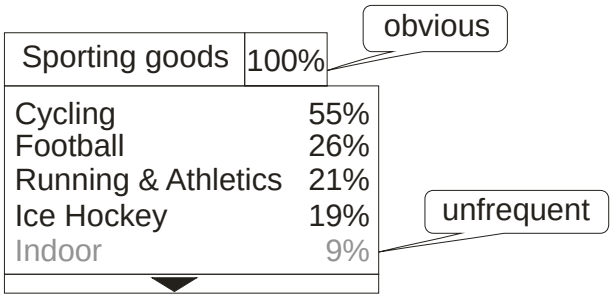


Fig. 2. Illustration of frequent (Cycling, Football, . . .) and obvious concepts.

3.1. COVER

We have introduced the new parameter called cover. It means a percentage cover of examined data. The cover parameter determines which classes are significant and essential to cover required data. We have used ideas of information theory. Especially it is the coding where the entropy is used for lossless data compression – our algorithm works similarly (but reversely) to the Huffman coding technique. The Huffman algorithm (building the coding tree) merges the least significant values – data sets with lowest probability of occurrence, which we understand as support. Our Inverse Huffman algorithm is merging the most significant values and the least significant doesn't take into consideration at all. So we introduced and employed an easy lossy compression of analyzed data. The technique works with data concepts in hierarchies if available. It can be illustrated as merging most frequent classes on each concept level. The covered part is a set of most frequent classes that has higher or equal support than the specified or somehow determined (see the Interactivity Improvement chapter) cover value (all together). We will depict the algorithm in the following example rather than formalizing the problem.

Example 1. Support from cover.

Suppose that we want to analyze shopping data of the Sports Shop oriented on the cycling. The initial setup of the cover threshold is 80% of investigated data (by default). The illustration of the algorithm is in the figure 3.

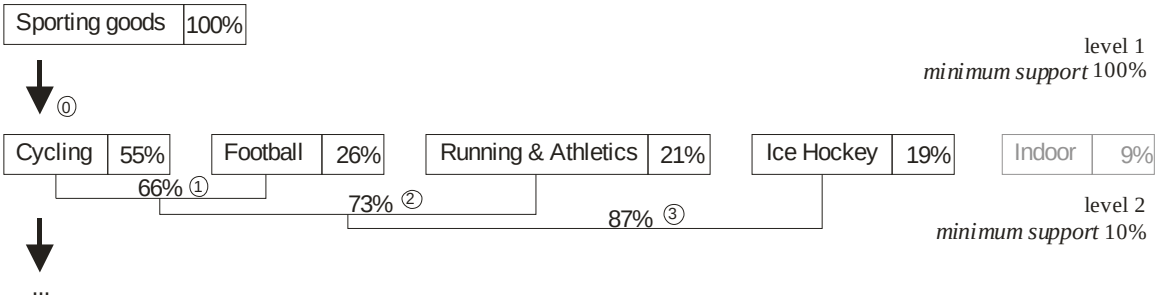


Fig. 3. Example of Inverse Huffman algorithm.

There is only one class at the top-most level of concept hierarchy. The determined minimum support is 100%, but the presence of concept Sporting Goods is obvious in the context of Sports Shop so we won't consider the top-most level at all.

The second concept level contains seven classes but it is necessary to merge only four of them to satisfy the default cover value – 80%. It is done by iterative merging of most frequent concepts. For instance the support of Cycling is 55% and together with Football equipment the cover is 66% and so on. Finally including Ice Hockey goods, it is 87%. So there are four scans over the database as shown in figure 3 in small circles.

It is not necessary to investigate the fifths (Indoor) and other classes but there should be embedded a politic to determine the minimum support value. In the second level the minimum support may be any value between 9% and 19%. The value 19% is a simple solution, but for further mining of more interesting rules it should be less. We prefer the algorithm sets minimum support a bit more than the first infrequent class – 10% in this case. So there can be one more database scan. The algorithm continues processing the next level in the same manner.

3.2. OBVIOSITY

Next problem of useful data mining is mining unnecessary, obvious information (e.g. Sports Shop sells *Sporting goods*). A high-level concept is supposed to be obvious when it (self) has higher support than the minimum cover (*80%*) and so the information gain of this information is low, because it can be presumed in mostly cases.

Obvious concepts are not processed by the data mining except leaf classes at the lowest level of hierarchy. If there is a level in a concept hierarchy containing only obvious classes (eg. Customers buy Products and Solutions), whole level is removed and is moved to the trash, as described in the next chapter.

4. PROGRESIVE PRESENTATION OF MINED RESULTS

The important topic of intuitive knowledge discovery is also the presentation. The OLAM SE presentation layer have been inspired by the UML (Unified Modelling Language) Structure diagrams [7] that are commonly used in application development and business modelling so even economists may be familiar with it. We presume description of aggregated data in its concept

hierarchies which is somehow different to the Class diagram. It is a graph. Each node represents a Concept as described in the chapter 2. In addition it contains the support value on right of a name of class (concept) and sub-concepts. Its example is in figure 2.

Edges represent relations. There are three types of relations in OLAM SE:

Relationship is an undirected line. Technically it is a relation among frequent concepts that merges two concepts. Relations haven't names but are marked with numbers (per cent) that means the support or other metrics of interrelated concepts (see figure 1.). Also the temporal evaluation of the relations is possible – it describes trends of support, confidence, cover and correlation metrics. Association is a special kind of relationship, it is a directed line and the confidence value(s) are situated near appropriate side of the association.

Aggregation is a special and important relation, it is an undirected link terminated by a diamond. It represents a concept hierarchy or an attribute relation. It is very important relation in OLAM SE because each item of the concept (sub-concept in conceptual hierarchy) can be easily extracted by a mouse – using drag and drop.

The expansion of an attribute concept to a new concept on the desktop is similar to the drilldown in OLAP. Hiding an attribute to its super-concept is equivalent to the OLAP roll-up operation. The class can be dropped anywhere on the desktop but relations are built only within the on-line workspace. See the figure 4. for illustration.

4.1. INTERACTIVITY

However, not all kinds of knowledge can be mined interactively. We propose, the knowledge that cannot be discovered by a query on-line or using drag and drop, is mined off-line – as a background processes with lower priority and its particular results are displayed progressively. In fact it is the frequent pattern search using iterative modification of Apriori [2]. According to this, on-line patterns are discovered first and then off-line concepts are sorted according to the relevance (support) to the on-line ones. We call it progressive hypothesis suggestion – user should wait some while for iterative results that improve its quality all the time without any user action. In this way, also the progressive characterisation and descriptive classification is realized.

The application consists of at least two workspaces – on-line and off-line. In the figure 4., the top window represents the off-line one. It contains classes or itemsets that are not currently under on-line investigation. Although the off-line window provides more data mining functions, we will think of it as a storage space for data that are temporarily unused at the moment.

The off-line workspace is necessary for two reasons. First it is the ability of slice/dice operation on one or more concept hierarchies (OLAP dimensions) simply by dragging concepts (subconcepts) from on-line to off-line workspace similarly to OLAP. The second reason is the on-line computational efficiency.

In the figure 4., the lower window is an on-line workspace. It works similarly to Han's OLAM [5] but it's simplified. OLAM SE uses Apriori algorithm [2]. There are investigated frequent itemsets that correspond to the frequent concept sets. First step of Apriori (generation of frequented itemsets) is done using Inverse Huffman algorithm.

The second cycle of Apriori - discovery of frequented 2-itemsets is important. These results are displayed using relationships. The support is counted and edges are evaluated. It means maximally $N\frac{N-1}{2}$ database scans for N itemsets in on-line workspace (each other). Larger itemsets the OLAM SE computes progressively after that it displays first results.

OLAM SE investigates multilevel data – different minimum support values derived from cover using Inverse Huffman should be considered between different concept's levels. We propose using the lower minimum support value, belonging to the deeper level where more levels are investigated together.

If the threshold is not reached, no association within classes is displayed – this can be forced and a dashed line appears and is evaluated. Note that the support is not counted among sets and its' supersets or subclasses (aggregation). It is similar to the obviosity metric.

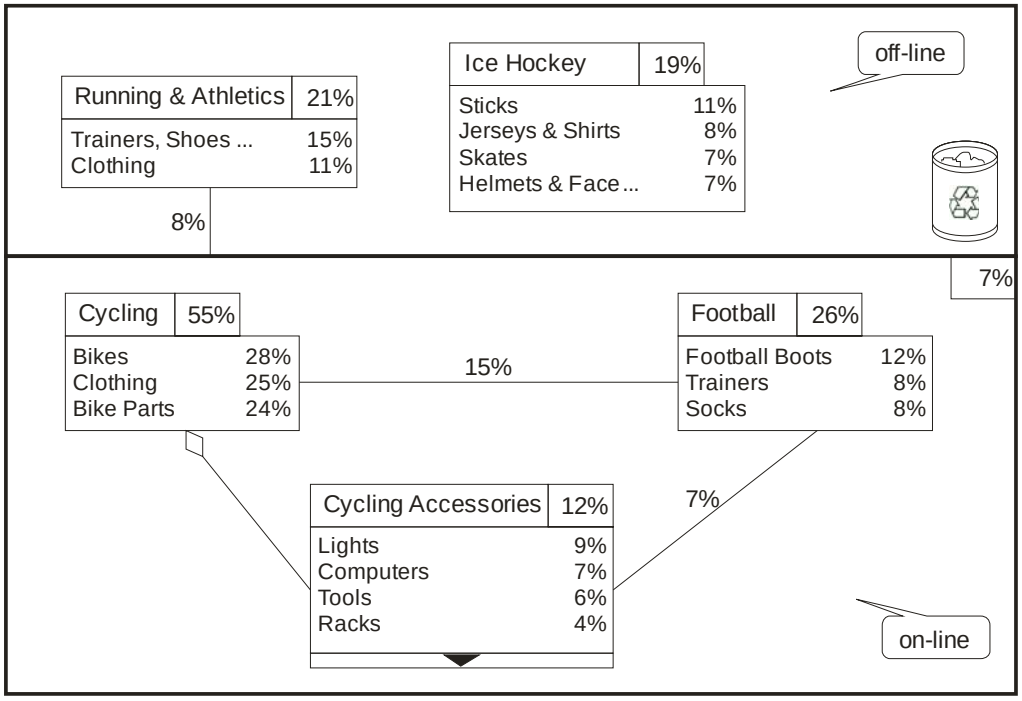


Fig. 4. Work space.

In such way we get (connected) Concept diagram as shown in the figure 4. After the OLAM SE display frequented 2-concepts Apriori goes further if all relations on the on-line workspace are frequent (together) and it shows the overall support value in right corner. Mined rules are displayed as oriented associations with proper values or classically in a separate window. The minimum confidence is set to 50% by default to display rules better than a chance. But it can be limited using user defined count of desired rules.

If the overall support value is below the minimum support value, only some patterns are showed. The workspace is investigated by the OLAM SE system to find the weakest relation – a concept that is connected by weakest relation. The OLAM SE tries to suggest removing such concepts – association is marked dashed, to obtain frequent itemset out of the on-line workspace to the off-line as described in the following chapter.

4.2. LITTER BIN

The straightest way how to shift (decrease) the cover and (increase) the obviosity threshold is to drag & drop the hidden in the litter bin concept to the working sheet, illustrated in the figure 4. After confirmation – it may take a long time, the appropriate parameter automatically shifts to the corresponding threshold. It is performed using Inverse Huffman algorithm again. If a user doesn't confirm the dialog, the itemset is added but the threshold is not shifted. Decreasing those thresholds

is also intuitive. You can remove the unnecessary concept from the workspace in two ways. The first way is moving the concept to the off-line part. The second is deleting it from both sheets – dropping into the litter bin. The dialog asks whether to hide it or to decrease the affected threshold, which may last a long time.

4.3. PROGRESSIVE MINING

In contrast to the relatively fast on-line hypothesis testing techniques (OLAP) and ad-hoc query based data mining (OLAM), the standard data mining techniques provide knowledge that cannot be discovered interactively due to its computational complexity.

We propose running these methods "off-line" as background processes with lower priority than on-line operations. We call it progressive suggestion in contrast to more or less hypothesis testing in OLAM. It works in idle (system) time when the user optically analyzes the on-line patterns. The example of an off-line operation is the previous chapter – removing itemsets that prohibit creation of frequent itemsets among whole workspace. The second is progressive suggestion of most relevant concept in the off-line sheet to the on-line. This is quite simple.

OLAM SE sorts the offline concepts by their support (zero DB scans). After that it analyzes one by one offline concept to the on-line patterns. This process creates some frequent itemsets (on-line and 1 off-line) similarly to the online analysis but on the background without any user activity. The offline workspace is then ordered by the newly counted support. After finishing these computations, OLAM SE continues computation for each sub-concept of every offline class. In that manner also aggregated concepts are ordered by relevance to the on-line in the available time. Any on-line computation aborts off-line operations. So whenever the user adds the first frequented off-line (sub) concept to the on-line worksheet, she might be sure that it is the best selection available at the moment. The other possibility is wait a while.

5. SIMPLIFIED DATA MINING

The previous part of the paper was concerning on frequent patterns mining mostly from transactional data (e.g. N:N relation in ER diagram) that we can imagine as a sparse table. But there is something more about the data mining than association analysis of multilevel data in OLAM SE. It is classification and characterization. For that purpose we use more than one off-line windows.

Suppose customer data (e.g. 1:N relation in ER). OLAM SE user selects one class containing all customers living in towns between 30 000 and 100 000 (internally using data warehouse, OLAP operations or an SQL query) and drags it to the on-line workspace. After some while she can see what products these customers most likely buy and other characteristics like how old they are if she has these features in off-line windows. Characterization works without any other action, just watching. The user may then specify other interesting characteristics using drag & drop.

5.1. CLASSIFICATION

A simple classification (Bayesian) can be performed in the similar way. The user drags & drops some concepts (features) from off-line to on-line sheet and puts other concepts (to the litter bin, another off-line window) so that only examined concepts (classes) are present in off-line workspaces. The OLAM SE without any other user action determines classes according to the features. Investigation of places of living with highest consume is thus trivial.

Technically it is an improved Bayesian MAP (maximum a posteriori) classification [3].

Suppose that x is an (on-line) observation and y an unknown (off-line) class:

$$* estimation(y) = \arg\max_{x} P(y \mid x)P(x) \quad (1)$$

$$p(y \mid x)P(x) = \frac{P(x \cap y)}{P(x)} P(x) = P(x \cap y) = support(x, y) \quad (2)$$

$$* estimation(y) = \arg\max_{x} support(x, y) \quad (3)$$

In such a way it appears that maximizing the support is the optimal descriptive Bayesian classification. And as the improvement to the naïve Bayesian classification is that OLAM SE doesn't have to presume that observations are independent [6]. They are interrelated using similarly to the association rules. So OLAM SE incidentally includes the not-naïve Bayesian classification. In addition, it works without any user action.

5.2. OLAM SE SYSTEM

The system works as follows. Firstly it derives important thresholds (different for each concept layer) that can be after that modified using drag & drop and the litter bin. Secondly the user selects concepts interesting for her. Then the system interactively creates frequent patterns and after some time it iteratively shows association analysis, concept classification or characteristics which are intuitively offered to the user. The generated hypotheses might be then interactively tested and modified by her.

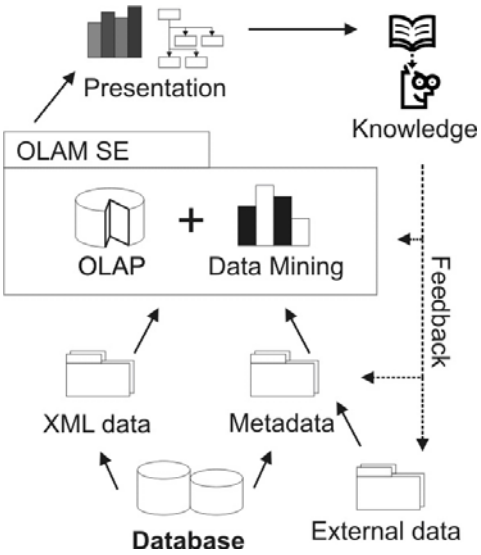


Fig. 5. OLAM SE Architecture.

OLAM SE consists of four layers as in the figure 5. The lowest layer is a database management layer. It provides access to data sources such as databases or XML documents. Next layer is MDDB (Multidimensional Database) layer. This layer provides us multidimensional view on data. It can use metadata describing conceptual hierarchy of data. Third layer is OLAM SE layer. This is a crucial layer for data mining and analysis. It works over data with described methods and algorithms. The last layer is GUI (Graphical User Interface) layer which provides the interactivity between user and OLAM SE and it enables construction of constraint based knowledge.

6. CONCLUSIONS

The data mining tasks are very useful in selective marketing, decision analysis, business management and many other areas. We have focused on multi-level frequent pattern analysis, which provides us an information about interesting relationships between data sets on the same or different levels of given conceptual hierarchy.

Our OLAM SE system provides user simplification of processing and understanding huge amounts of data. We use interactivity principles of OLAP to user-driven processing of input data. We have established two parameters – cover and obviosity. The cover parameter is based on Paretto analysis and entropy coding to determine interesting patterns. It's leading to the lossy compression of data sets on each level of conceptual hierarchy. It can be used to automatically determine minimum support value for each level of conceptual hierarchy in multilevel association rules mining tasks as well. The second parameter is the obviosity. On the basis of this parameter the frequent patterns with low information gain are moved to the litter bin.

Our approach operates in two concurrent modes. In online mode it is processed interactive and fast mining of knowledge. In the offline mode the data is processed by data mining algorithms with high computation complexity but progressively – depending on how much time the user has. The main goal is the user that knows the data doesn't have to know OLAP or data mining techniques – it is either characterization, not-naïve Bayesian classification, frequent pattern analysis, association and correlation rules mining nor the appropriate thresholds and parameters.

We are hard working on the implementation of OLAM SE system at the moment, to see the experimental results of proposed algorithms, its quality and performance.

REFERENCES

[1] HAN, J., FU, Y.: *Discovery of multiple-level association rules from large databases*. In Proc. of 1995 Int'l Conf. on Very Large Data Bases, (1995) 420-431.

[2] HAN, J., KAMBER, M.: *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publish-ers, Boston, 2nd edition (2006).

[3] HAN J.: *Towards on-line analytical mining in large databases*. SIGMOD Record (ACM Special Interest Group on Management of Data), (1998).

[4] ZHU, H.: *On-line analytical mining of association rules*. Master's thesis, Burnaby University, Burnaby, British Columbia V5A 1S6, Canada (1998).

[5] HAN, J. et al.: DBMiner: *A system for mining knowledge in large relational databases*. In Proc. 1996 Int'l Conf. on Data Mining and Knowledge Discovery, (1996) 250-255.

[6] CHMELAR, P.: Bayesian Concepts for Human Tracking and Behavior Discovery. Student EEICT 2006, Vol. 4,360-364, Brno, CZ, VUT v Brne (2006).

[7] UML: Unified Modeling Language.online: http://www.uml.org, (2007).

[8] JURAN, J.J.: Juran's quality handbook. New york, NY [u.a.] McGraw-Hill, 5th ed. (1999).