# *Multidimensional Data Mining for Discover Association Rules in Various Granularities*

Johannes K. Chiang
Department of Management Information System
National Chengchi University
Taipei, Taiwan
jkchiang@nccu.edu.tw

Rui-Han Yang
Department of Management Information System
National Chengchi University
Taipei, Taiwan

*Abstract*— **Data Mining is one of the most significant tools for discovering association patterns that are useful for many knowledge domains. Yet, there are some drawbacks in existing mining techniques. The three main weaknesses of current data-mining techniques are: 1) re-scanning of the entire database must be done whenever new attributes are added because current methods are based on flat-mining using pre-defined schemata. 2) An association rule may be true on a certain granularity but fail on a smaller ones and vise verse. This may result in loss of important association rules. 3) Current methods can only be used to find either frequent rules or infrequent rules, but not both at the same time.**

**This research proposes a novel data schema and an algorithm that solves the above weaknesses while improving on the efficiency and effectiveness of data mining strategies. Crucial mechanisms in each step will be clarified in this paper. This paper also presents a benchmark which is used to compare the level of efficiency and effectiveness of the proposed algorithm against other known methods. Finally, this paper presents experimental results regarding efficiency, scalability, information loss, etc. of the proposed approach to prove its advantages.**

***Keywords - Multidimensional Data Mining; Granular Computing; Apriori Algorithm; Concept Taxonomy; Association Rule***

## I. INTRODUCTION

The scientific and business communities are increasingly interested in knowledge discovery. Significant examples are finding new drugs for cancers and new portfolios of products/services. The notion of association rule is capable of providing simple but useful form of knowledge [4,6]. Thereafter, methods for discovery of association rules such as machine learning and data mining have been extensively studied.

Most of the conventional mining approaches only perform flat scan over the databank based on a pre-defined schema. Most associations occur in a context of certain breadth, the knowledge usually encompasses multidimensional content. However, adding attributes to the mining task means changing the schema and rescanning is required. This is highly inefficient.

The second problem is most conventional mining approaches assume that the induced rules should be effective throughout a database as a whole. This obviously does not fit with real-life cases [6]. Different association rules can be found in different parts (segments) of database. If a mining tool deals only with the database as a whole, the meaningful at smaller granularities will be lost.

The goal of this research is to provide an approach with novel data structure and efficient. The crucial issue is to explore more efficient and accurate multidimensional mining of association patterns on different granularities in a flexible and robust manner.

## II. RELATED WORKS AND TERMINOLOGIES

### *A. Frequent and Infrequent Rules*

Records in a transactional database contain simple items identified by Transaction IDs using conventional methods. The notion of association is applied to capture the co-occurrence of items in transactions. There are two important factors for association rules: support and confidence. Support means how often the rule applies while confidence refers to how often the rule is true. We are likely to find association rules with high confidence and support. Some data mining approaches allow users to set minimum support/confidence as the threshold for mining [6, 10]. Efficient algorithms for finding infrequent rules are also in development.

### *B. Multidimensional Data Mining*

Finding association rules involving various attributes efficiently is an important subject for data mining. Association Rule Clustering System (ARCS) was proposed in [9], where association clustering is proposed for a 2-dimensional space. The restriction of ARCS is that only one rule is generated in each scan. Hence, it takes massive redundant scans to find all rules.

The method proposed in [9] mines all large itemsets at first and then use a relationship graph to assign attribute according to user given priorities of each attribute. Since the method is meant to discover large itemsets over a database as the whole, infrequent rules that hold in smaller granularities will be lost. Different priorities of the condition attributes

978-1-4673-5285-7/13/$31.00 ©2013 IEEE

will induce different rules so that user may need to try with all possible priorities to discover all rules.

### C. Apriori Algorithm

The Apriori algorithm is a level wise iterative search algorithm for mining frequent itemsets with regards to association rules [1,3-5,13]. The weakness of Apriori algorithm s that it requires k passes of database scans when the cardinality of the longest frequent itemsets is k. the algorithm is also computation intensive in generating the candidate itemsets and computing of support values. If the number of first itemsets element is k, the database will be scanned k times at least. Hence it is not efficient enough.

A variant of Apriori algorithm is the AprioriTID [2]. The AprioriTID reduces the time required for frequency counting by replacing every transaction in the database by a set of candidate sets that occurs in that transaction [2]. Although the AprioriTID algorithm is much faster in later iterations, it is much slower in early iterations as compared to the original Apriori algorithm. Another drawback of AprioriTID is inefficient use of space; the database modified by Apriori-gen can be much larger than the initial database.

### D. Concept Description and Taxonomy

The issue of data structure and descriptive model for mining is less discussed when comparing works on algorithms. The concept description task is problematic since the term concept description is used in different ways. In this situation, researchers argue for a de facto standard definition for the term [10, 11]. At this stage it is easier to deal with common criterion on higher abstraction level for concept description such as comprehension [11] and compatibility [6].

Han & Kamber view concept description as a form of data generalization and define concept description as a task that generates descriptions for the characterization and comparison of the data [11].

Ontology provides a vocabulary for specific domains and defines the meaning of the terms and relationships between them in practical situations. The term Taxonomy is used in this paper as it is more flexible and can cover cases with no semantic meaning.

## III. THE MULTIDIMENSIONAL MINING APPROACH

### A. Representation Schema and Data Structure

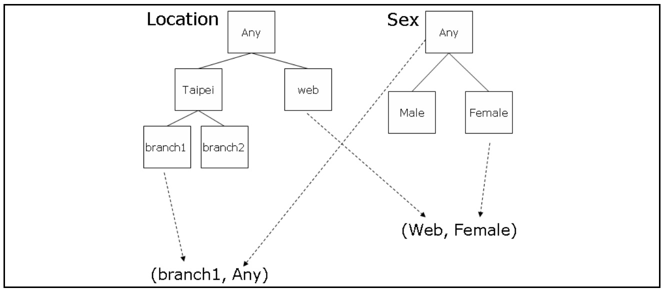

Figure 1. Forest of Concept Taxonomy

For the sakes of comprehension and compatibility, we use the forest structure consisting of Concept-Taxonomies to represent the overall searching space, i.e. the set of all propositions of the concepts. On top of this structure, the sets of association patterns can be formed by selecting concepts from individual taxonomies. The notions can be clarified with examples as follows:

1). *Taxonomy: A category consists of domain concepts in a latticed hierarchical structure, while each member per se can be in turn a taxonomy. An example for customer's characteristics can be [Age, Sex, Occupation…], while the taxonomy of occupation can be [manager, labor, teacher, engineer…].*

2). *Forest of concept taxonomies: A hyper-graph for representing universe of discourse or the closed-world of interests built with domain taxonomies with regards to location and Sex of customers is shown in Fig 1.*

3). *Association Rule: a pattern consisting of elements taken from various concept taxonomies such as [(location=web),(Sex=female),(Goods=milk)].*

By multidimensional data mining of association rules, the notion of relation often refers to the belonging relationship between elementary patterns and generalized patterns rather than semantics [6]. Other notations will be used in this paper are shown in Table 1.

TABLE I. CONCEPTS AND NOTATIONS

| Notation | Definition |
|---|---|
| TID | Transaction identifier |
| MD | Multidimensional database |
| CT | Concept Taxonomy |
| $E_i$ | The *i*-th element segment |
| $T[_{Ei}]$ | An element segment over $E_i$ in MD |
| $G_i$ | The j-th generalized pattern |
| $T[_{Gj}]$ | The j-th combined segment over $G_j$ |
| $R_{Ei}$ | Rules with regards to the *i*-th element segment |
| $R_{Gj}$ | Rules with regards to the *j*-th generalized pattern |
| $(G_j,r)$ | Association rules over $G_j$ with regards to to match ratio *r* |
| m | Ratio for a relax match, given by the user. |

### B. The Data Mining Algorithm

Outline of the proposed algorithm is shown in Fig 2. The input of the mining process involves 5 entities: 1) a multidimensional transaction database MD (optional when a default MD is assigned), 2) a set of concept taxonomies for each dimension (CT), 3) a minimal support *viz. minSup*, 4) a minimal confidence, viz. *minConf*, and 5) a match ration m for the relaxed match.

The output of the algorithm is the multidimensional association with regards to a full/relaxed match in the MD. The mining process can be characterized in two independent steps: 1) finding all itemsets in each element segment and 2) updating all combinations of segments by using the output of step 1. For practical reasons, step 1 of the algorithm can be replaced by similar algorithms such as Apriori. This segregation of the two steps enables flexible mining and ease of use in distributed environments.

```
 Input:
  Multidimensional Transaction Database MD
  Concept taxonomies for each dimension: CTx(X= 1-n)
  User given threshold: minSup, minConf, match ratio m
Procedure:
   Phase0:
      to generate all Ei and Gj by CTx (x = 1 to n);
      build the pattern table;
 Phase1:
         For all $Ei \subset G$
         to discover all association rules r in T[Ei] as $R_{Ei}$
 Phase2:
      for all Ei
         for all Gj that $Ei \subset Gj$
            to update $R_{Gj}$ using $R_{Ei}$;
 Phase3:
      for all Gj
         For all r (which satisfy m) in $R_{Gj}$
            output ($G_j$, r);
 Output:
  all multidimensional association rules(p, r)
```

Figure 2. Outline of algorithm

The task of the algorithm is to discover all association rules $R_{Ei}$ in the element segment T[$E_i$] for each element pattern $E_i$. It then uses $R_{Ei}$ to update $R_{Gj}$, i.e the set of association rules for every generalized pattern $G_j$ which covers $E_i$. The task done by each element pattern is to find large itemsets in itself and acknowledge its super generalized patterns with the association rules. The task don by each generalized pattern is o decide which rules hold within it according to the acknowledgements from the element patterns. The mining procedure needs only to work on each element segment and uses the output from each segment to determine which rules hold in the combined segments. Thus, there is no need to scan all of the possible segments for finding the rules.

### C. Generating all patterns and the pattern table

The procedure generates all elementary and generalized patterns with the given forest, where a pattern table for recording the belonging relationship between the elementary and generalized patterns is built. Given a set of concept taxonomies, a multi-dimensional pattern can be generated by choosing a node from each concept taxonomy. The combination of different choices represents all the represents all the multidimensional patterns. For example, Fig 3 presents a situation of 12 patterns.

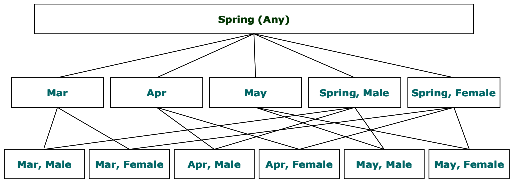

Figure 3. Belonging relationships between patterns

TABLE II. THE PATTERN TABLE FOR RELATIONS SHOWN IN FIG 3

| | Mar | Apr | May | Spring, Male | Spring, Female | Spring |
|---|---|---|---|---|---|---|
| Male | 1 | 0 | 0 | 1 | 0 | 1 |
| Female | 1 | 0 | 0 | 0 | 1 | 1 |
| Male | 0 | 1 | 0 | 1 | 0 | 1 |
| Female | 0 | 1 | 0 | 0 | 1 | 1 |
| Male | 0 | 0 | 1 | 1 | 0 | 1 |
| Female | 0 | 0 | 1 | 0 | 1 | 1 |

Fig 3 also shows the belonging relationship between patterns in a lattice structure. The relationships are recorded in the form of bit map which includes element patterns and generalized patterns. In the table, a “1” indicates that the element pattern belongs to the corresponding generalized pattern and “0” indicates case vice versa. Table II shows a bit map which stores such relationships.

### D. Update Process

```
for all $R_{Ei}$
    for all $G_j \supset E_i$
        if ($R_{Gj}$ never be updated)
            $R_{Gj} = R_{Ei}$;
        else
            $R_{Gj} = R_{Gj} \cap R_{Ei}$;
```

Figure 4. The “update” algorithm

After all patterns have been generated and the pattern table has been built, the procedure begins to read the transactions with regards to each element segment to discover all association patterns. Besides our algorithm, the Apriori algorithm can also be used in this phase. The output of this phase is all of $R_{Ei}$ for each element patter $E_i$. It will be fed as the input to the next phase for updating each $R_{Gj}$ using $R_{Ei}$. Fig 4 shows the outline of the update procedure with a full match.

### E. The output function

For a full match, the mining process outputs all ($G_j$, r) pairs for every r left in each $R_{Gj}$. For a relaxed match, it outputs all ($G_j$,r) pair for every r in each $R_{Gj}$ where the count exceeds |mT[$G_j$]| by a relax match. By means of algorithm described above, loss of finding the rules that only hold in some segments can be prevented and pickup of multidimensional association rules that do not holder over all the range of the domain can be avoided.

## IV. The Experiment and Evaluation

### *A. Experiment scenario of a wholesale case*

To measure and prove the performance of the method, a scenario for a wholesales business using synthetic data are established for the test. The wholesales enterprise has various business branches and a web-site for its business operations.

Data from four branches and the website are gathered for the experiment. We take five of the various attributes (Abode, Sex, Occupation, Age and Marriage) as the dimensions for the test. Adding with the product catalog and price/profit record, there are 7 dimensions and we build the concept taxonomies for each dimension.

To examine the effect of different customer behaviors, we generate three data types as illustrated in Table III. The parameters and the default values of the data sets are shown in Table IV. There are 118 multidimensional patterns from these taxonomies, 44 of them are element patterns and the other 74 of them are generalized ones. The mining tool should find all large itemsets for the 74 generalized patterns.

TABLE III. Three Types of data Sets

| | |
|---|---|
| Type 1 | To generate a single set of maximal potentially large itemsets and then generate transactions for each element pattern Ei following apriori-gen.[4] |
| Type 2 | Besides a set of common maximal potential large itemsets, to genreate maximal potentially large itemsets for each element pattern Ei, and then genreate transactions for each element pattern Ei. The common maximal potentially large itemsets respectively following the apriori-gen[4]. |
| Type 3 | Generating a set of maximal potentially large itemsets for each element pattern Ei, and generating transactions for each element pattern Ei from its own maximal potentially large items following the apriori-gen.[4] |

TABLE IV. Parameters and default values of data sets

| Notation | Meaning | Default |
|---|---|---|
| \|D\| | Number of transactions | 100K |
| \|T\| | Average size of transactions | 6 |
| \|I\| | Average size of maximal potentially large itemsets | 4 |
| \|L\| | NBumber of maximal potentially large itemsets | 1000 |
| N | Number of items | 1000 |
| $S_M$ | The maximum size of segmentation | 50 |

### *B. Experiment Results*

At first, the 74 generalized patterns are successfully found. The key feature of the algorithm as shown in Fig. 5 is that it is linear (and hence highly scalable) to the number of records and that it is flexible in terms of reading various data types. The test result w.r.t scalability in Fig. 5 shows that the algorithm takes execution time linear to the number of transactions of all three data types. The experiment results of both the test (see Figure 5 and 6) shows that our algorithm is superior to conventional methods in several areas:

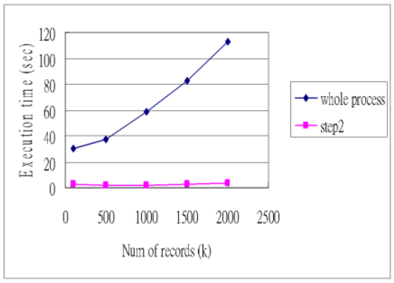

Figure 5. Execution Time w.r.t to the no. of transactions

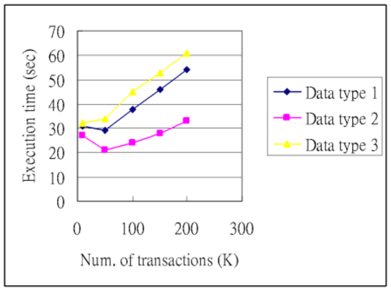

Figure 6. Scalability test w.r.t. the no. of records

Execution time with regards to number of transactions is linear for the data types tested for the whole process. This means that the time and space cost of executing our algorithm do not increase exponentially as compared to conventional methods.

Phase 2 ( the update phase ) of our algorithm is an important space and time saver as shown by the Figure 8; execution time is also linear and time taken to read up to 2000k records took less than 5 seconds. This means that data patterns from new data can be quickly extracted and used to update the existing pattern table for immediate use. result in Fig 5 shows that the algorithm takes linear time to the number of transactions.

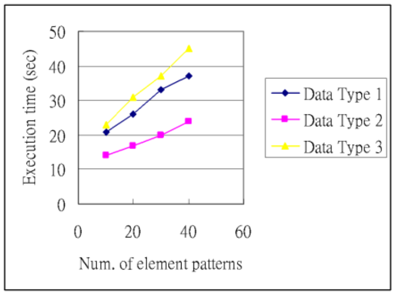

Figure 7. Efficiency in relation to the number of element patterns

In general, an increase of element patterns with result in an increase in execution time; the key to scalability is having the execution time increasing in a linear manner with an increase in element patterns. In Figure 7, all three data types experienced an increase of execution time with an increase of element pattern in a linear fashion, thus making our algorithm efficient.

Most importantly, an increase in element patterns leads to a less than proportion increase in execution time, making out the algorithm highly scalable. Reading off Figure 7, a 4 time increase of 30 element patterns from 10 to 40 will result in:

*1) 75 times increase in execution time for data type 1 from 20 seconds to 35 seconds*

*2) 67 times increase in execution time for data type 2 from 15 seconds to 25 seconds.*

*3) 05 times increase in execution time for data type 3 from approximately 22 seconds to 45 seconds.*

The test results shows that for increasing the number of element patterns will not decrease the efficiency of the algorithm. It fulfills the requirement of scalability in terms of number of element patterns as well.

After we have understood the execution effectiveness and flexibility of the algorithm, the next step is evaluate the impact of various user inputs on the algorithm. As described earlier in this paper, the main user inputs are minimum support *minSup*, minimum confidence *minConf*, and match ratio *m*. The impact of user input on the algorithm is shown in Figure 8, 9, 10 and 11 respectively.

### *C. Impact of User Input on the Algorithm*

The impact of *minSup* on the algorithm can be categorized in terms of efficiency, discrete ratio and lost ratio. All of such algorithms are sensitive to the minimum support; the smaller the minimum support, the longer the execution time. However, we have shown that the real execution time of the step 2 (the update) in the proposed algorithm is relatively much shorter than the whole process (see Fig. 5).

The test results proved that an increase in *minSup* will lead to greater returns of investment in terms of time efficiency; this is in line with one of the core objectives of building an efficient algorithm. Our algorithm is more efficient than conventional methods in terms of execution time over data. For instance in Figure 8, a 10 time increase (from 0.1 to 1) in *minSup* leads to a more than proportionate decrease in execution time across all data types:

*1) Execution time for data type 1 decreased by approximately 10 times, from approximately 400 seconds to approximately 40 seconds in terms of execution time.*

*2) Execution time for data type 2 decreased by more than 30 times, from more than 600 seconds to approximately 20 seconds in terms of execution time.*

*3) Execution time for data type 3 decreased by more than 11 times, from approximately 350 seconds to approximately 30 seconds in terms of execution time.*

The test results proved that an increase in *minSup* will lead to greater returns of investment in terms of time efficiency; this is in line with one of the core objectives of building an efficient algorithm.

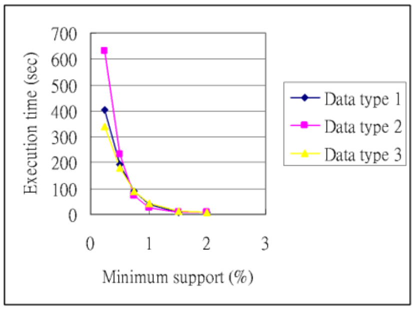

Figure 8. Efficiency in Relation to Minimum Support

The discrete ratio is the ratio of the number of rules pruned by the improved algorithm to the number of rules discovered by prior mining approaches. Figure 10 shows the ratio of rules pruned by the improved algorithm against *minSup*.

In general, all three data types (except for data type 1) exhibited an increase of ratio with an increase of *minSup* from approximately 0.2% to 2%.

The test results point the fact that the proposed algorithm can effectively decrease unwanted generalized patterns in which elemental data patterns is not true. This greatly helps users to focus on data patterns that are useful for their organizations while uncovering niche data patterns. For instance with a higher setting value, only <Female, Age over 60, take AP transfusions> will be found instead of <Age over 60, take AP transfusions>.

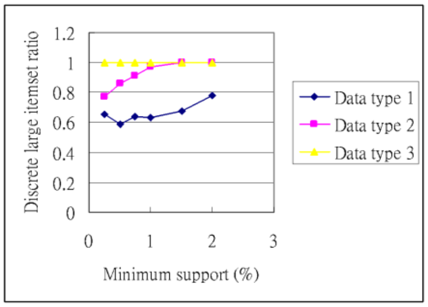

Figure 9. Effects of *minSup* on Discrete Large Itemsets Ratio

Figure 10 shows the test result on lost ratio, i.e. the influence of *minSup* values on the lost rules by other mining tools in comparison to this approach. All three data types experienced an increase in lost ratio over an increase in *minSup* from 0.25% to 2%, with the greatest increase in data type 2, followed by data type 3 and finally data type 1.

The test results prove that the proposed algorithm will help users uncover useful data patterns which otherwise would be uncovered by traditional approaches. Thus, our objective of uncovering niche data patterns that would otherwise be left out is met and proved by this test result.

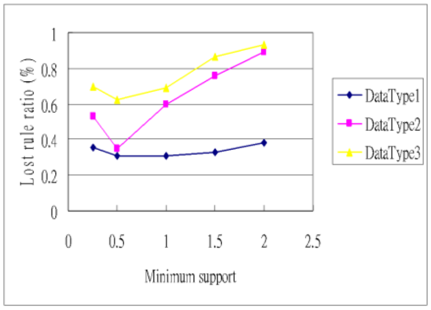

Figure 10. Effects of *minSup* on Lost Itemsets

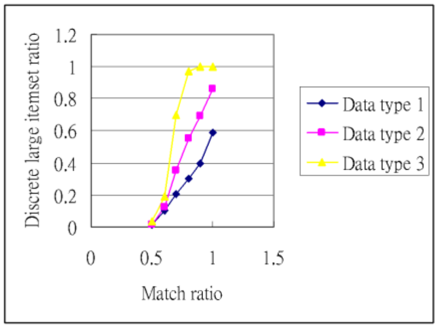

Figure 11. Effects of match ratio on discrete large itemsets ratio

Increasing the match ratio would reduce unwanted data patterns in general. Figure 11 shows the effect of match ratio (r) on discrete ratio.

Similar to the above test results, an increase of m from 0.5 to 1 results in a more than proportional increase in discrete ratio across all three forms of data types. The significance of this test result is congruent with the test results above; the algorithm is efficient and scalable without losing flexibility and helps uncover niche data patterns.

## V. Summary

The article proposes an approach which includes a novel data structure and an efficient algorithm for mining association rules on various granularities. The advantages of this approach over existing ones include 1) greater comprehensiveness and easy of use. 2) More efficient with limited scans and storage I/O operations. 3) More effective in terms of finding rules that hold in different granularity levels. 4) Association patterns can be stored so that incremental mining is possible. 5) Information loss rate is very low.

The whole process of the development and experimental measurement of the multidimensional data mining approach was discussed in this paper. The test result shows that its performance, efficiency, scalability and information loss rate are better than the current approaches. The effects of perceived issues and potential development of data mining and concept description are worthy of further investigation.

## References

Article in a journal:

[1] Agrawal,R and Shafer, J.C, "Parallel mining of association rules", IEEE Transactions on Knowledge and Data Engineering, 8(6) (1996) 962-969.

[2] He, L-J., Chen, L. C. and Liu, S.-Y., "Improvement of aprioriTID algorithm for mining association rules", Journal of Yangtai University, Vol. 16, No. 4, 2003

Article in conference proceedings:

[3] Srikant, R. and Agrawal, R, "Mining generalized association rules, Proceedings of the 21th VLDB Conference, Zurich, Swizerland, 1995

[4] Agrawal, R. and Srikant, R, "Fast algorithms for mining association rules in large databases", in Proceedings of the '94 International Conference on Very Large Data Bases, 1994, pp. 487–499.

[5] Agrawal, R., Imielinski, T. and Swami, A.N, "Mining association rules between sets of items in large databases", in Proceedings of the ACM-SIGMOD 1993 International Conference on Management of Data, 1993, pp. 207–216.

[6] Chiang, J. K. and Wu, J.C, "Mining multi-dimension rules in multiple database segmentation-on examples of cross-selling", 16th ICIM Conference, 2005, Taipei, Taiwan.

[7] Lent, B, Swami A. and Widon, J, "Clustering association rules", in:13th International Conference on Data Engineering

[8] Liu, B. Hsu, W. and Ma, Y. "Mining association rules with multiple minumum supports", in ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-99)

[9] Tasi, S. M. Pauray and Chen, C-M, "Mining interesting association rules from customer databases and transaction databases".

[10] The CRISP-DM Consortium, CRISP-DM 1.0 www.crisp-dm.org

Textbook reference:

[11] Han, J. and Kamber, M. "Data mining - concepts and techniques 2nd ed, Morgan Kaufman, 2006.

[12] Li, M. and Baker, M. : "The GRID – core technologies", Willy 2005.

[13] Feldman, R, and Sanger, J. "The text mining handbook – advanced approaches in analyzing unstructured data", Cambridge University Press, 2007