SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TMLAI TRANSACTIONS ON
MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

VOLUME 2, ISSUE 3
ISSN: 2054 - 7390

# Multidimensional Multi-granularities Data Mining for Discover Association Rule

**Johannes K. Chiang[1], Chia-Chi Chu[2]**
*Department of Management Information Systems, Cloud Computing and Operation Innovation Center, National Chengchi University, Taipei, Taiwan*
[1]jkchiang@nccu.edu.tw; [2]102356020@nccu.edu.tw

## ABSTRACT

Data Mining is one of the most significant tools for discovering association patterns for many knowledge domains. Yet, there are deficits of current data-mining techniques, i.e.: 1) current methods are based on plane-mining using pre-defined schemata so that a re-scanning of the entire database is required whenever new attributes are added. 2) An association rule may be true on a certain granularity but false on a smaller ones and vise verse. 3) Existing methods can only find either frequent rules or infrequent rules, but not both at the same time.

This paper proposes a novel algorithm alone with a data structure that together solves the above weaknesses at the same time. Thus, the proposed approach can improve the efficiency and effectiveness of related data mining approach. By means of the data structure, we construct a forest of concept taxonomies which can be applied for representing the knowledge space. On top of the concept taxonomies, the data mining is developed as a compound process to find the large-itemsets, to generate, to update and to output the association patterns that can represent the composition of various taxonomies. This paper also derived a set of benchmarks to demonstrate the level of efficiency and effectiveness of the data mining algorithm. Last but not least, this paper presents the experimental results with respect to efficiency, scalability, information loss, etc. of the proposed approach to prove its advantages.

**Keywords:** Multidimensional Data Mining, Granular Computing, Concept Taxonomy, Association Rules, Infrequent Rule, information Lose Rate

# 1 INTRODUCTION

While Service Innovation is getting more interests in scientific and business communities, Data Mining turns out to be increasingly important for knowledge discovery of innovative services. Association rules can be used to figure out simple yet useful insights on services [5, 13, 17]. Significant examples are finding new purchasing behaviors for shops and new portfolios of rationale services. For example, “52% of the customer those buy product X also buy product Y”.

**DOI:** 10.14738/tmlai.23.259
**Publication Date:** 9th June 2014
**URL:** http://dx.doi.org/10.14738/tmlai.23.259

Given such association rules, we can decrease the costs of the product X, and raise the quality level of product Y to make more benefits.

However, most conventional data mining approaches only perform a plane scan over the databank based on a predefined schema for searching. Questions often arise such as: Should there be any other influencing factor like W on purchasing of product Y taken into account? Since most association rules apply in a context of certain breadth, the knowledge usually exists in multidimensional insides [5]. In the meantime, adding attributes to the data warehouse is meant to change the schema and initiates a full re-scan that would consume extra time.

The second problem of the conventional mining approaches lies in the assumption that the rules derived should be effective throughout a data warehouse as a whole. Nevertheless, this obviously is not true in real-life cases [5]. Different association rules can be found in different segments of the database. If a mining tool deals only with the database as a whole, the meaningful rules that are only partially true will be overlooked.

The goal of this research is to develop an approach with novel data structure and efficient algorithm for the multi-dimensional data mining for association patterns in various granularities. The crucial issue is to explore association patterns with more efficient and accurate multidimensional mining for the association patterns on different granularities. And, the data mining approach has to be very flexible and robust.

# 2 BASELINE OF THE RESEARCH

## 2.1 Multidimensional Data Mining

Finding association rules efficiently involving multi-attributes is an important subject for data mining. Association Rule Clustering System (ARCS) was proposed in [10], where association rule clustering is proposed for a 2-dimensional space. The restriction of ARCS is that it generates only one rule at a time of clustering. Subsequently, it takes massive redundant scans to find all rules.

The method proposed in [16] mines all large itemsets first and then applies a directed graph to assign attributes according to the priorities given by user for each attribute. Since the method is meant to discover large itemsets over a database as the whole, certain infrequent rules may be lost due to several granularities. Different priorities of the condition attributes will infer different rules so that user has to try with all possible priorities to discover all possible rules.

## 2.2 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.

## 2.3 Apriori Algorithm

### *2.3.1 Apriori Algorithm*

The Apriori algorithm is a level-wise iterative search algorithm for mining frequent itemsets w.r.t association rules [1, 3, 5, 7, 13, 14, 17]. The key drawback of the Apriori algorithm is that it requires k passes of database scans when the cardinality of the longest frequent itemsets is k. In addition, the algorithm is computation intensive in generating the candidate itemsets and computing the support values, especially for applications with very low support threshold and/or vast amount of items. In this algorithm, if the number of first itemsets element is k, the database will be scanned k times at least. So, it is not efficient enough. The key point for improving the algorithm is to reduce the number of itemsets.

### *2.3.2 AprioriTID Algorithm [9]*

The AprioriTID is a variant of the aforementioned Apriori algorithm which reduces the time needed for the frequency counting procedure by replacing every transaction in the database by the set of candidate sets that occur in that transaction [9]. This is done by iterating each candidate sets repeatedly.While the AprioriTID algorithm is much faster in later iterations, it is much slower than original Apriori in early iterations. This is mainly due to the additional overhead that is created when the adapted transaction database Ck does not fit into main memory and has to be written into disk [4]. If a transaction does not contain any candidate k-sets, then Ck will not have an entry for the transaction. Hence, the number of entries in Ck may be smaller than the number of transactions in the database, especially at later iterations of the algorithm. Other drawbacks of ApririTID are that the database modified by Apriori-Gen can be much larger than the initial database and only faster in the later stages of the scans.

## 2.4 Concept Description and Knowledge Taxonomy

The issues of data structures and concept description models for data mining when comparing works dealing with algorithms are less discussed till. The concept description task is problematic, since the term “concept description” is used in quite different ways in related discussions. In this situation, researchers argue for a de facto standard definition for the concept description [8, 18]. At this beginning stage, it is easier to deal with normal criterion on higher abstraction level for the concept description, such as comprehension [8] and compatibility [4].

Researchers view concept description as a form of data generalization and define the concept description as a task that generates descriptions for the characterization and comparison of the data [8]. Similar concept appears in the development of ontology for Semantic Web/GRID. Semantic Web can be described as an extension of the existing Web where information is considered with priori well-defined meaning, enabling computer and people to work in cooperation centric to Internet [11]. The objective of such techniques is to enhance ill-structured content so that it can be interpreted universally by machines or humans.

In practical applications, ontology provides a vocabulary for specific domains and defines the meaning of the terms and relationships between them. In this paper, ontology refers to the shared understanding (comprehension) of domains of interests which is often conceived as a set of concepts, relations, axioms etc. Hence, the term “Taxonomy” is hereby similar to “Ontology” and both terms can be used to denote the classification or categorization of concepts that describe entities and relations among them. This paper applies the term Taxonomy rather than Ontology because the former is more flexible and even can cover the case with no semantic meaning.

# 3 METHODOLOGY

## 3.1 Representation schema and data structure

As mentioned in section 2.4, the issues of data structures regarding descriptive models are less discussed when comparing R&D works dealing with data mining algorithms. Therefore, we will present the building blocks of our representation schema and data structure, which are namely (1) Taxonomy, (2) Forest of Concept Taxonomies and (3) Association Rules.

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 the 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:

### *3.1.1 Taxonomy*

A category consists of domain concepts in a latticed hierarchical structure, while each member per se can be in turn taxonomy. An Example (see Figure 1) for customer’s characteristics can be [Age, Sex, Marry, Urbanization], while for instance the taxonomy of Sex can [Male, Female] and Marry can [Married, Unmarried] so on.

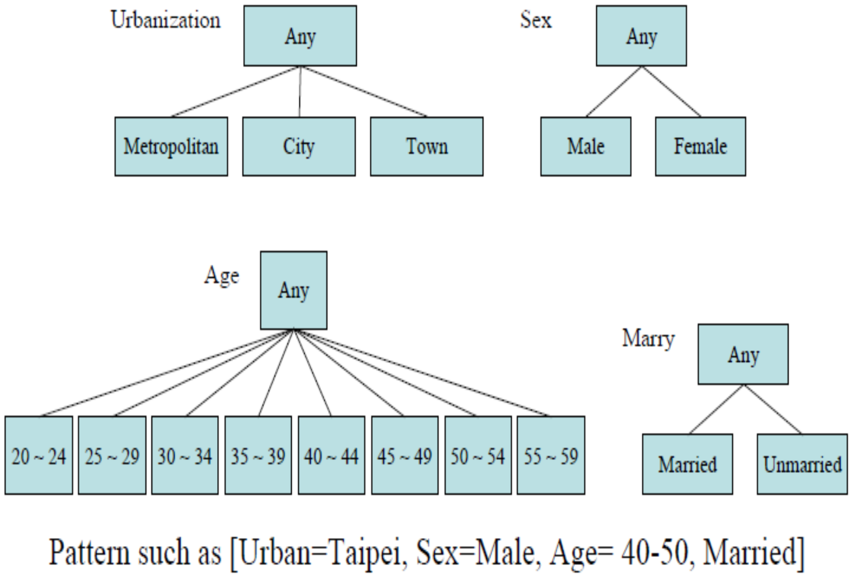


**Figure 1: An Example for Forest of Concept Taxonomies**

### *3.1.2 Forest of concept taxonomies*

A hyper-graph for representing the universe of discourse or the closed-world of interests is built with taxonomies under consideration. An example of forest of taxonomies with respect to the location and Sex of customers is shown in Figure 2 below:

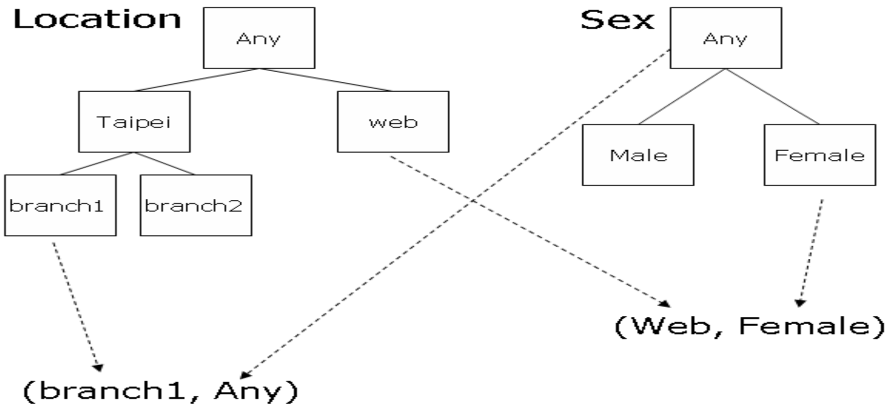


**Figure 2: Examples of Forest Concept Taxonomies**

### *3.1.3 Association Rule*

An association rule typically refers to a portfolio's pattern which consists of elements taken from various concept taxonomies such as [(Location=branch1), (Sex=female)]. It owns support and confidence greater than the user-specified minSup and minConf respectively [4].

### *3.1.4 Element patterns and generalized patterns*

An element pattern is composed of dimension atoms. On the other hand, if at least one of them is a dimension compound which combine several dimension atoms, we call this pattern a generalized pattern. For example, <web, Female> is an element pattern, <branch1, Any> is a generalized pattern, and both them are multi-dimension patterns. We use to denote the i-th element pattern, and use to denote the j-th generalized pattern.

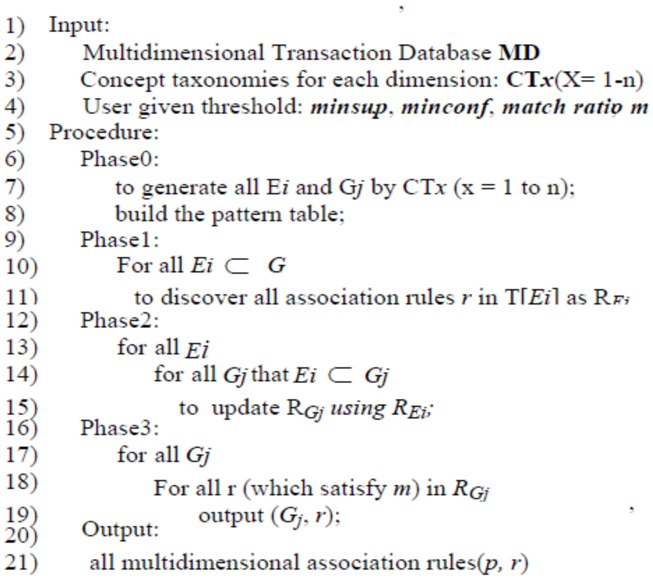
By the proposed multidimensional data mining of association rules, the notion of relation will be implemented by the belonging relationship between elementary patterns and generalized patterns rather than the semantics [4]. Other notations to be used in the following text are shown in Table 3 below:

**Table 1: Concepts and Notations**

| **Notation** | **Meaning** |
|---|---|
| CT | Concept Taxonomy |
| *Ei.* | The *i*-th element segment |
| *T*[*Ei*] | an element segment over Ei in MD |
| Gi | The *j*-th generalized pattern |
| *T*[*Gj*] | The *j*-th combined segment over G |
| R*E* i | Rules w.r.t the *i*-th element segment |
| R*Gj* | Rules w.r.t the *j*-th generalized pattern |
| (*Gj* , *r*) | association rules over *Gj* w.r.t to match ratio *r* |

### 3.2 The Multidimensional Multi-granularity data mining algorithm

The proposed data mining process can be formulated essentially with two cascading steps: (1) finding all itemsets in each elementary segment and (2) updating all combinations of the segments by the output of Phase 0. For the practical reason, the algorithm in Phase 0 can be replaced by any tool available elsewhere such as the Apriori algorithm so that an easy realization the phase 0 and then a segregation of the two steps enable the flexible mining on a distributed environment like Cloud and Grid. Figure 3 illustrates outline of the proposed algorithm extending the mining process into four phases.



```
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 3: Outline of the proposed algorithm.

Outline of the proposed algorithm is shown in Figure 3. The input of the mining process involves 5 entities, namely (1) a multidimensional transaction database MD which is optional when a default MD is assigned, (2) a set of concept taxonomies for each dimension (CTs), (3) a minimal support, viz. minSup, (4) a minimal confidence, viz. minConf, and (5) a match ratio m for the relaxed match. The output of the algorithm encompasses all multi-dimensional associations with respect to the fully-relaxed match within the MD. The last three settings can help with finding frequent or infrequent rules.

The most significant feature of the algorithm is it's capability to discover both frequent and infrequent associations rules $R_{Ei}$ (based on different levels of granularities) in the element segment T[Ei] for each element pattern E*i*. After it, $R_{Ei}$ is used to update $R_{Gj}$, i.e. the set of association patterns for every generalized pattern G*j* which includes E*i*. The heuristic regarding each element pattern is to find the large-itemsets per se and acknowledge its super generalized patterns with the result. The task of each generalized pattern is to 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 to determine which rules hold in the compound segments. Thus, it is not necessary to scan all of the potential segments for finding the rules.

### 3.3 Pattern Generation and the Pattern Table

Being a pre-processing mechanism, the algorithm generates at first 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 of the taxonomy. The compound of different choices represents all the multidimensional patterns.

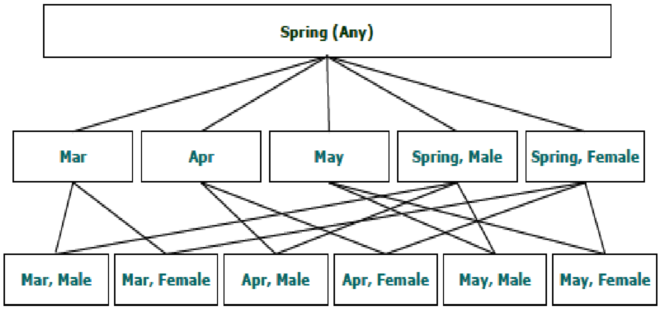
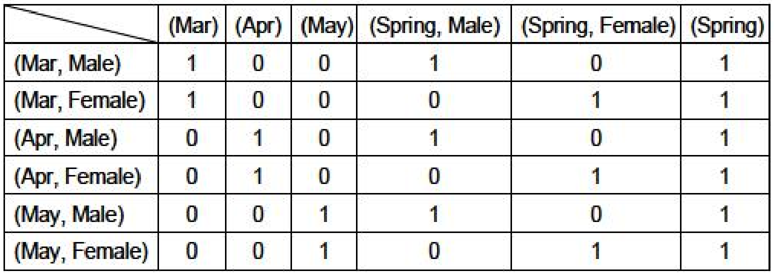


**Figure 5: Belonging relationships between patterns**



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

**Figure 6: The pattern table (for the relations in Figure 4) shows an example of the belonging relationship between 12 patterns in a lattice structure. The relationships are recorded in the form of bit map as shown in Figure 5 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 the case vice versa.**

## 3.4 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 6: The "Update" algorithm for the full match**

In order to be more optimization algorithm, we proposed full match and relaxed match method for update process. After all patterns and the pattern table have been generated, the procedure reads the transactions of each element segment and then discovers all the association rules. The output of this phase is all $R_{Ei}$ for each element pattern E*i* that will be fed as the input to the next phase for updating each $R_{Gj}$ using $R_{Ei}$. For a full match illustrated in Figure 6, the update is done by intersection of the set $R_{Gj}$ and the set $R_{Ei}$, where E*i* belongs to G*j* , let $R_{Gj}$ = $R_{Ei}$ if $R_{Gj}$ is updated for the first time. After all the intersections, the association pattern r left in $R_{Gj}$ holds in all element segments covered by T[G*j*].

```
for all $R_{Ei}$
    for all $G_j \supset E_i$
        for all $r$ in $R_{Ei}$
            if ($r \notin R_{Gj}$)
                add $r$ to $R_{Gj}$;
                $R_{Gj}.r$.count = 1;
            else
                $R_{Gj}.r$.count++;
```

Figure 7: The "Update" procedure for the relaxed match

For the relaxed match as shown in Fig. 7, a counter for each rule in $R_{Gj}$ is set. While using $R_{Ei}$ for updating $R_{Gj}$, the counters of both $R_{Gj}$ and $R_{Ei}$ are incremented by one and the rules, those appear in $R_{Ei}$ but not in $R_{Gj}$, will be added to $R_{Gj}$ while setting the counter to one. After all the update process, the association rule r in $R_{Gj}$ whose counts exceed m|T[G*j*]| holds in at least m *100% of the element segments T[$E_i$] that are covered by T[G*j*], and thus (G*j*, r) is a multidimensional association rule for the relaxed match in MD.

Full match can ensure that all association rule be found in various granularities. But, it may be too restrictive to ignore some rules. On the other hand, relaxed match can solve "restrictive" problem and hold more association rules which may be our interesting rules. User can adjust the m ratio which ranges between 0 and 1.

For example, suppose we have a generalized segment <Spring> which covers three element segment <March>, <April>, and <May>. Finding patterns of each element segment <March>{A},{B},{C}、<April>{B},{C} and <May>{B},{E}. As we above-mentioned algorithm that update each $R_{Gj}$ using the $R_{Ei}$ come from previous phase. For the full match case, we just can hold rule B in <Spring> generalized segment $R_{Gj}$ because only rule B exists every element segment $R_{Ei}$. For the relaxed match case, we suppose m = 0.6 (result of count numbers should greater than 1.5 times) and count numbers of all rules in each element segment $R_{Ei}$: {A=1}、{B=3}、{C=2}、{E=1}. Hence, we hold rule B and C in <Spring> generalized segment $R_{Gj}$.

### 3.5 The Output Function

For a full match, the algorithm outputs all the (G*j*, r) pairs for every r left in each $R_{Gj}$. For a relaxed match, it outputs all the (G*j*, r) pair for every r in each $R_{Gj}$ where the count exceed |mT[G*j*]|. By means of this approach, loss of finding the rules that only hold in some segments can be prevented. And, pickup of multidimensional association rules that do not hold over all the range of the domain can also be avoided. For example, the full match can guarantee that the corresponding rules, those hold only in two months of spring but fail in the rest one, will never be counted as an association rules with respect to whole spring.

### 3.6 The Breakthroughs for Incremental Data Mining

A breakthrough hereby is that the incremental data mining can be realized with the proposed approach. By keeping out the rules deduced in each element segment, we only need to search the new data. That is, using the proposed approach, we can produce the new association rules by combining the rules discovered from the new data with existing rules to reduce redundant scan on the old data. The following section will present our experimentation results.

### 3.7 Design of metrics for measuring data mining

In order to assure the performance, we need to design metrics for measuring the mining performance, at least to measure whether it is better than the prior algorithms. By cascade evaluating the results of a hypothetical measurement, we can evaluate the consequence from any sequence of measurements to determine the optimal next measure. For this reason, a one-step look-ahead strategy based on Shannon's Entropy Function is adopted and the capacity of ICT systems can be described in the following form [4, 15]:

$$\mathrm{C} = \mathrm{B} * [\log 2\ (1 + \mathrm{S/N})] \tag{3}$$

where B is the bandwidth, (S/N) is Signal-to-Noise(S/N) ratio.

Drawing on this equation, the function for the performance of data mining can be formulated as follow:

C=|D| [log2 (1+information lost ratio)], where |D| is the number of transactions in whole transaction database [4].

While WSE*i* denotes each element segment in the measure, the WSE*i* of an element segment T[E*i*] can be generated by a uniform distribution between 0 and SM. Suppose there are N element segments, the number of transactions in the element segment T[E*i*] is:

$$|D_{Ei}| = \frac{|D|}{\sum_{a\to 0}^{n} WS_{Ea}} WS_{Ei} \tag{4}$$

Thereafter, the definitions of information loss are:

$$\text{discrete ratio} = \frac{|\{ r \mid r \text{ holds in } T[Gj] < Gj,r> \text{ doesn't hold in } \mathbf{MD} \}|}{|\{ r \mid r \text{ holds in } T[Gj] \}|} \tag{5}$$

Definition 1: 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.

$$\text{lost ratio} = \frac{|\{\text{<Gj,r>} \mid \text{<Gj,r> holds in MD r doesn't hold in T[Gj]}\}|}{|\{\text{<Gj,r>} \mid \text{<Gj,r> holds in MD}\}|} \tag{6}$$

Definition 2: lost ratio is the ratio of the number of rules discovered by the improved algorithm but lost in the previous mining approaches to the number of rules discovered by the improved algorithm.

# 4 EXPERIMENT AND EVALUATION

## 4.1 Experiment scenario

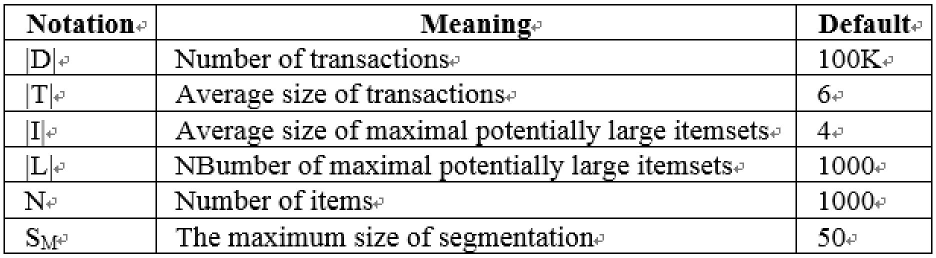
To measure and prove the performance of the method, a scenario for a wholesale business using synthetic data are established for the test. The wholesales enterprise runs various business branches and a web-site for its operations. Data from four branches and the website are gathered for the experiment. 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.

**Table 2: Three Types of Experimental Data Set**

| | |
|---|---|
| Type1 | To generate a single set of maximal potentially large itemsets and then generate transactions for each element pattern Ei following apriori-gen.[3] |
| Type2 | Diagnosis-2, Therapy1. Beside a set of common maximal potential large itemsets, to generate maximal potentially large itemsets for each element pattern Ei. and then generate transactions for each element pattern Ei and the common maximal potentially large itemsets respectively following the apriori-gen[3] |
| Type3 | generating a set of maximal potentially large item-sets for each element pattern Ei, and then generating transactions for each element pattern Ei from its own maximal potentially large itemsets following the apriori-gen.[3] |

The test bench is implemented with Java on a PC Server with an AMD processor and the data mining software is implemented with Java. Data from different branches and the website are collected for our experiment. To examine the effect of different customer behaviors, we generate three data types as illustrated in Table 2. The parameters and the default values of the data sets are illustrated in Table 3. There are 118 multidimensional patterns from these taxonomies, 44 of them are element patterns and the other 74 of them are generalized patterns. The mining tool should find all large item sets for the 74 generalized patterns.

**Table 3: 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 |

## 4.2 The Results of Experiment

At first, the 74 generalized patterns are successfully found. The key feature of the algorithm as illustrated in Figure 9 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

Figure 9 illustrates 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 8 and 9) illustrates that the new algorithm is superior to conventional methods in several areas:

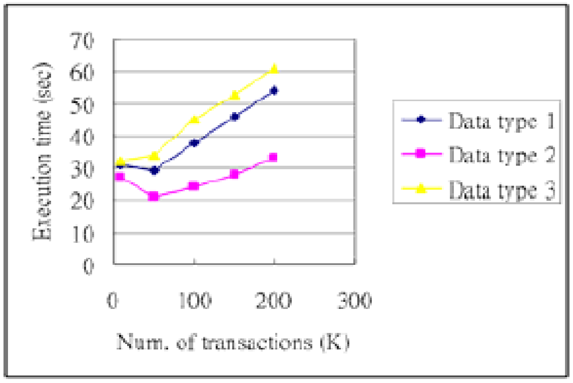


Figure 8: Scalability test w.r.t. the no. of transactions

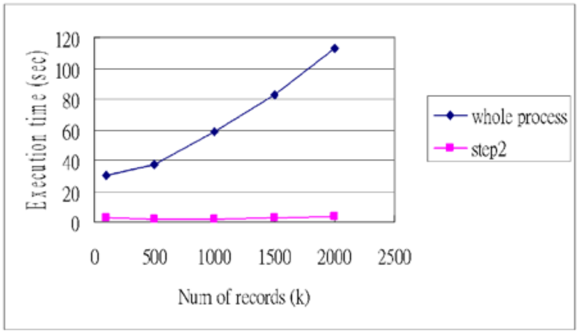


Figure 9: Scalability experiment w.r.t. the no. of records

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 illustrated 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.

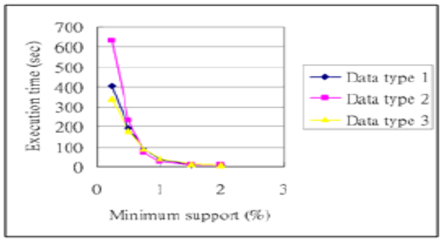


Figure 10: Efficiency in Relation to Minimum Support

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 11, 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 10, a 4 time increase of 30 element patterns from 10 to 40 will result in:

- 75 times increase in execution time for data type 1 from 20 seconds to 35 seconds.
- 1.67 times increase in execution time for data type 2 from 15 seconds to 25 seconds.
- 2.05 times increase in execution time for data type 3 from approximately 22 seconds to 45 seconds.

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 Figure 8).

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 11, 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:

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

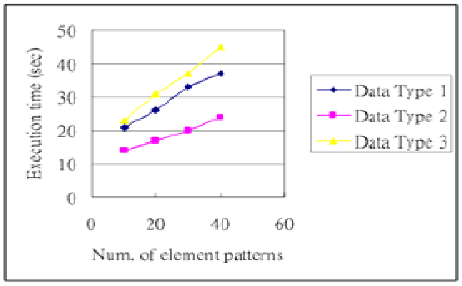


**Figure 11: Efficiency in Relation to Minimum Support**

The discrete ratio is the ratio of the number of rules pruned by the proposed algorithm to the number of rules discovered by prior mining approaches. Figure 12 illustrates the ratio of rules pruned by the proposed 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%.

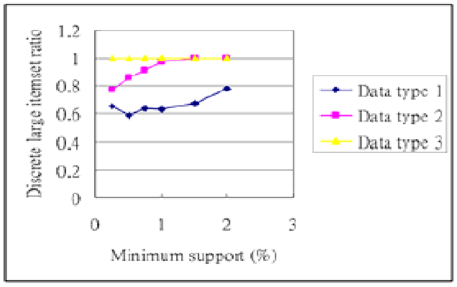


Figure 12: Effects of MinSup on discrete large itemsets ratio

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 30-50, buy SK-II > will be found instead of <Age 30-50, buy SK-II>.

Figure 13 illustrates 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.

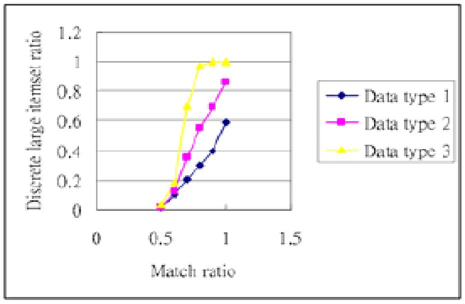


Figure 13: Effects of match ratio on discrete large itemsets ratio

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.

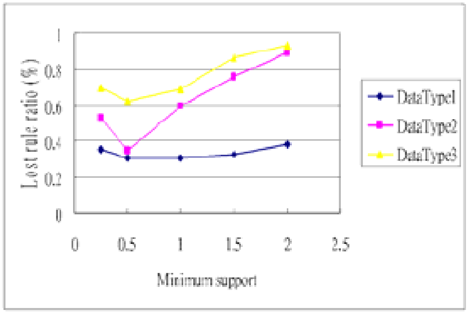


Figure 14: Effects of match ratio on lost itemsets.

Increasing the match ratio would decrease unwanted data patterns in general. Figure 13 illustrates 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.

## 5 SUMMARY

The paper proposes an approach including a novel data structure and an efficient algorithm for mining association rules on various granularities. The advantages of this approach over existing approaches include (1) more comprehensive and easy-to-use (2) more efficient with limited scans (3) more effective with finding rules hold in different granularity levels (4) capable of finding frequent patterns and infrequent patterns while users can choose the full match and the relaxed match (5) low information loss rate (6) capable of incremental mining of association rules to avoid unnecessary re-scan.

The whole development process and experimental measurement of the multidimensional data mining approach were discussed in this paper. The test result reveals 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 big data strategy are worthy of further investigation. And, deployment of big data mining over the MapReduce on top of cloud computing architecture is our target of our future research.

## REFERENCES

[1]. R. Agrawal and J. C. Shafer (1996). "*Parallel Mining of Association Rules,*" IEEE Transactions on Knowledge and Data Engineering, vol. 8, no. 6, pp. 962-969.

[2]. R. Agrawal and R. Srikant (1994). "*Fast Algorithms for Mining Association Rules in Large Databases*," in Proceedings of the 20th International Conference on Very Large Data Bases.

[3]. R. Agrawal, T. Imielinski and A. N. Swami (1993). "*Mining Association Rules between Sets of Items in Large Databases,*" in Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data.

[4]. J. K. Chiang (2007). "*Developing an Approach for Multidimensional Data Mining on various Granularities ~ on Example of Financial Portfolio Discovery,*" in ISIS 2007 Proceedings of the 8th Symposium on Advanced Intelligent Systems, Sokcho City, Korea.

[5]. J. K. Chiang and J. C. Wu (2005). "*Mining Multi-Dimension Rules in Multiple Database Segmentation-on Examples of Cross Selling,"* in Proceedings of the 16th International Conference on Information Management, Taipei, Taiwan.

[6]. T. M. Cover and J. A. Thomas (2006). *Elements of Information Theory,* 2nd ed., Wiley.

[7]. R. Feldman and J. Sanger (2007). *The Text Mining Handbook – Advanced Approaches in Analyzing Unstructured Data,* Cambridge University Press.

[8]. J. Han and M. Kamber (2006). *Data Mining - Concepts and Techniques,* 2nd ed., Morgan Kaufman.

[9]. L. J. He, L. C. Chen and S. Y. Liu (2003) "*Improvement of AprioriTid Algorithm for Mining Association Rules*," Journal of Yantai University(Natural Science and Engineering Edition), vol. 16, no. 4.

[10]. B. Lent, A. Swami and J. Widom (1997). "*Clustering Association Rules,*" in Proceedings of the 13th International Conference on Data Engineering.

[11]. M. Li and M. Baker (2005). The GRID – Core Technologies, Wiley.

[12]. B. Liu, W. Hsu and Y. Ma (1999), "*Mining Association Rules with Multiple Minimum Supports,"* in Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

[13]. G. Shmueli, N. R. Patel and P. C. Bruce (2007)."*Association Rules*," in Data Mining for Business Intelligence, Concepts, Techniques, and Applications, Wiley, pp. 203-215.

[14]. R. Srikant and R. Agrawal (1995). "*Mining Generalized Association Rules,*" in Proceedings of the 21th International Conference on Very Large Data Bases, Zurich, Switzerland.

[15]. W. Stallings (2004). "*Channel Capacity*," in Business Data Communications, 6th ed., Pretice Hall, pp. 470-471.

[16]. P. S. Tsai and C. M. Chen (2004). "*Mining interesting association rules from customer databases and transaction databases,*" Information Systems, vol. 29, no. 8, p. 685–696.

[17]. C. Vercellis (2009). "*Association Rules*," in Business Intelligence, Data Mining and optimization for Decision Making, Wiley, pp. 277-290.

[18]. The CRISP-DM Consortium, CRISP-DM 1.0 (2000), www.crisp-dm.org.

## The Author

Prof. Dr.-Ing. Johannes K. Chiang is now a faculty member of the Department of MIS and the Deputy Director of the Center for Cloud Computing and Operation Innovation at National Chengchi University Taipei. He received his academic degree of Doctor in Engineering Science (*Dr.-Ing., Summa Cum laude*) from the RWTH University of Aachen Germany. His current research interests include Cloud Computing, Semantic Web, Business Intelligence, Data Mining, e-Business and ebXML. He also serves as a consultant for several government agencies in Taiwan and as an active member of various international affiliations, such as IEEE, ACM, CSIM and ITMA etc. before 1995, he has been a research fellow at RWTH of Aachen and a Manager of EU/CEC ESPRIT Programs