Association Rule Mining for Specific New Course

Nhabhat Chaimongkol* and Phayung Meesad**

Department of Information Technology, Faculty of Information Technology*
Department of Teacher Training in Electrical Engineering**, Faculty of Technical Education
King Mongkut's University of Technology North Bangkok
nhabhat@hotmail.com*, pym@kmutnb.ac.th**

ABSTRACT – Most language schools devote a significant portion of their budget on new courses to distinguish their school from their competitors and to increase the number of students. The schools should specify courses that fulfill the students' needs. This will raise the competitiveness of the schools. Also the schools will earn higher loyalties and profits because of the increase of new students. This article proposes a Mining Course Map (MCM) algorithm for investigating on the relationships among students' demands, type of course and transaction records. MCM is a modified association rule analysis based-on FP-growth algorithm. For comparison study, the proposed method was compared with Association Rule Miner And Deduction Analysis (ARMADA). The results show that the execution time of MCM is less than ARMADA which means that MCM is more efficient than the ARMADA. In addition, the results show that different knowledge and rules can be extracted from students to specify new courses for new and old members. This paper suggests that the school should extract knowledge from student demands. The knowledge can be used to manage new courses properly.

KEY WORDS – Data mining, Association Rule, FP-growth, Mining Course Maps, ARMADA

1. Introduction

Specific new course (SNC) is an important and difficult problem in language schools. Most schools are aware of the importance and the need to specify new courses for their students. But it is not an easy task because student's preferences are different, inefficient utilization can render the data collected useless causing databases to become "data dumps" [1]. So we need to increase the student data effectively, which is one of the benefits of analysis. Data mining is a methodology of discovering significant knowledge, such as anomalies, changes, significant and associations from large amounts of data stored in a database or data warehouse [2]. To filter the literature, we can implement a variety of techniques such as decision tree, estimation, clustering, classification and association rules, also many applications for methodology including association rules and classification. The mapping approach focuses on the use of IT, and can be used as a tool to support specific new course, thus the perception of map configurations is a new concept for analyzing perceptions in business relationship studies [3]. Daniel, Wilson, and McDonald [4] utilized a marketing map to represent the best practice in marketing and also used the process map to understand how IT can be deployed in order to support a marketing information system. This research investigates the following issues in specifying new courses for language schools: What exactly are the students "needs" and "wants" for a language course? What is the required course level? The research attempts to efficiency integrate more decision variables and levels using the MATLAB program. The methodology of this research utilizes the FP-growth algorithm, which is an association rule set for data mining. In section 2 we present the proposed data mining technique MCM (Mining Course Map) methodology. Section 3 illustrates the experiment results, also comparison between ARMADA and MCM. Section 4 presents a brief conclusion and possible future work.

2. Related Works

The product maps for a new product development [5] contain a lot of datasets in the database. Data mining tools are used by SPSS Clementine to make decisions via small datasets, but this is the limitation and problem of the research. If a school has large datasets, the development cannot progress further because the tool's can only handle limited analysis. Thus this research proposes a new data mining tool based on FP-growth algorithm and ARMADA (Association Rule Miner And Deduction Analysis) tool [6].

2.1 Data mining

Data mining technology has emerged as a means for identifying patterns and trends from large quantities of data. Data mining, also known as Knowledge Discovery in Databases, has been defined as "The nontrivial extraction of implicit, previously unknown, and potentially useful information from data" [11]. Data mining is used to extract structured knowledge automatically from large data sets [12]. The information that is "mined" is expressed as a model of the semantic structure of the dataset, where in the prediction or classification of the obtained data is facilitated with the aid of the model [13].

Descriptive mining and Predictive mining are the two categories of data mining tasks. The descriptive mining refers to the method in which the essential characteristics or general properties of the data in the database are depicted. The descriptive mining techniques involve tasks like Clustering, Association and Sequential mining. The method of predictive mining deduces patterns from the data such that predictions can be made. The predictive mining techniques involve tasks like Classification, Regression and Deviation detection. Mining Frequent Itemsets from transaction databases is a fundamental task for several forms of knowledge discovery such as association rules, sequential patterns, and classification [14].

Data mining involves the use of sophisticated data analysis tools to discover previously unknown, valid patterns and relationships in large data sets. These tools can include statistical models, mathematical algorithms, and machine learning methods (algorithms that improve their performance automatically through experience, such as neural networks or decision trees). Consequently, data mining consists of more than collecting and managing data, it also includes analysis and prediction.

Data mining can be performed on data represented in quantitative, textual, or multimedia forms. Data mining applications can use a variety of parameters to examine the data. They include association (patterns where one event is connected to another event, such as purchasing a pen and purchasing paper), sequence or path analysis (patterns where one event leads to another event, such as the birth of a child and purchasing diapers), classification (identification of new patterns, such as coincidences between duct tape purchases and plastic sheeting purchases), clustering (finding and visually documenting groups of previously unknown facts, such as geographic location and brand preferences), and forecasting (discovering patterns from which one can make reasonable predictions regarding future activities, such as the prediction that people who join an athletic club may take exercise classes) [8].

The data mining process can be broken down into four distinct phases.

2.1.1. Decision, whether or not to carry out data mining on a given data set.

2.2.2 Data preparation, preparing the data for analysis.

2.2.3 Model building, building a prediction model is undertaken.

2.2.4 Interpretation, which is largely carried out by individuals, but which can be greatly assisted using automated means, such as graphical representations of the results.

This process is illustrated using a flow chart in Figure 1 [9].

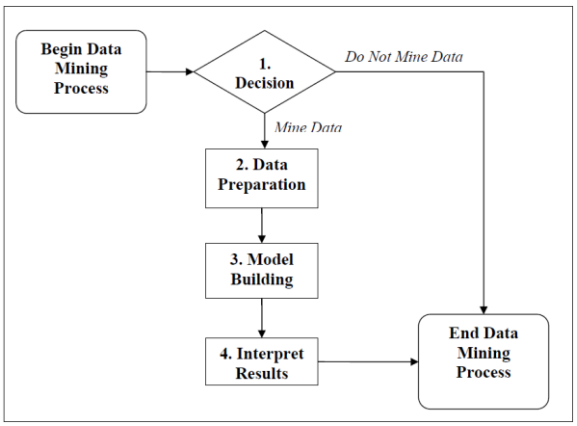

Figure1: The Four Phases of the Data Mining Process

2.2 Association rules

An association rule is an implication expression of the form X→Y, where X and Y are disjoint itemsets, i.e., $X \cap Y = \emptyset$. The strength of an association rule can be measured in terms of its support and confidence. Support determines how often a rule is applicable to a given data set, while confidence determines how frequently items in Y appear in transactions that contain X. The formal definitions of these metrics are:

$$\text{Support, } s(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N} \quad (1)$$

$$\text{Confidence, } c(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)} \quad (2)$$

where X is the antecedent part of the rule and Y is the consequent part of the rule; $\sigma(X \cup Y)$ is the support count of both X and Y simultaneously

appearing in the rule; N is the number of transactions; and $\sigma(X)$ is the count of antecedent.

The problem of mining association rules can be divided into two steps. The first step is to detect a large item set, whose support is greater than Minsup and the second step is to generate association rules, using the large item set. Such rules must satisfy two conditions:

$$\text{Sup}(X \cup Y, D) \geq \text{Minsup} \quad (3)$$

$$\text{Conf}(X \rightarrow Y) \geq \text{Minconf} \quad (4)$$

Complete details of association rule analysis can be found in [5].

2.3 Data mining tool (ARMADA)

ARMADA (Association Rule Miner and Deduction Analysis) is a data mining tool that extracts association rules from numerical data files using a variety of selectable techniques and criteria. ARMADA is available by The MathWorks on their central file exchange. The program integrates several mining methods which allow the efficient extraction of rules, while allowing the thoroughness of the mine to be specified at the user's discretion. The program was designed as a tool to assist in the analysis both of the knowledge extracted and the deduction processes by which such a task is undertaken. However, the program can also be used as a straightforward data mining tool for the efficient extraction of association rules [6].

ARMADA is a rule mining tool which is a powerful technique used to discover interesting associations between attributes contained in a database. Association rules can have one or several output attributes and an output attribute from one rule can be used as the input of another rule. Association rules are thus useful, both for obtaining an idea of what concept structures exist in the data (as with unsupervised clustering) and for model creation. In the second instance, the rules generated provide the underlying concepts used in the construction of decision trees and even neural networks (although this is carried out by the automated learning process).

ARMADA has been designed so as to allow for both straightforward mining in the form of rule extraction and as a means of analyzing the knowledge extracted during a separate data mining session. ARMADA has been evaluated in detail. However, the results are produced by the Association Rule Miner were not as useful as had initially been hoped. This is both due to the fact that it is difficult to obtain useful association rules from continuous numerical data and the fact that despite the claims made regarding the tool, it performed very poorly in many cases. ARMADA is not able to mine fully numerical data sets, despite its claim to the contrary. ARMADA is able to extract some rules from this data by firstly rounding the value of the dependant variable (the simplest means of providing some categorization) and then using a very small portion of the data. Even this, however, does not produce very useful results [9].

2.4 FP-growth Algorithm

FP-growth algorithm is an efficient method of mining all frequent itemsets without candidate's generation. The algorithm mine the frequent itemsets by using a divide-and-conquer strategy as follows: FP-growth first compresses the database representing frequent itemset into a frequent-pattern tree, or FP-tree, which retains the itemset association information as well. The next step is to divide a compressed database into set of conditional databases (a special kind of projected database), each associated with one frequent item. Finally, mine each such database separately. Particularly, the construction of FP-tree and the mining of FP-tree are the main steps in FP-growth algorithm [7].

FP-Growth works in a divide and conquers way. It requires two scans on the database. FP-Growth first computes a list of frequent items sorted by frequency in descending order (F-List) during its first database scan. In its second scan, the database is compressed into a FP-tree. Then FP-Growth starts to mine the FP-tree for each item whose support is larger than minimum support by recursively building its conditional FP-tree. The algorithm performs mining recursively on FP-tree. The problem of finding frequent itemsets is converted to searching and constructing trees recursively.

Figure 2 shows a simple example. The example DB has five transactions composed of lower-case alphabets. The first step that FP-Growth performs is to sort items in transactions with infrequent items removed. In this example, we set minimum support = 3 and hence keep alphabets f; c; a; b; m; p. After this step, for example, T1 (the first row in the figure) is pruned from (f, a, c, d, g, i, m, p) to (f, c, a, m, p). FP-Growth then compresses these "pruned" transactions into a prefix tree, which root is the most frequent item f. Each path on the tree represents a set of transactions that share the same prefix; each node corresponds to one item. Each level of the tree corresponds to one item, and an item list is formed to link all transactions that possess that item. The FP-tree is a compressed representation of the transactions, and it also allows quick access to all

transactions that share a given item. Once the tree has been constructed, the subsequent pattern mining can be performed. However, a compact representation does not reduce the potential combinatorial number of candidate patterns, which is the bottleneck of FP-Growth [10]. A lot of algorithms have been proposed to optimize the performance of the FP-growth algorithm [7].

Map inputs (transactions) key="": value	Sorted transactions (with infrequent items eliminated)	Map outputs (conditional transactions) key: value
f a c d g i m p	f c a m p	p: f c a m m: f c a a: fc c: f
a b c f l m o	f c a b m	m: f c a b b: f c a a: f c c: f
b f h j o	f b	b: f
b c k s p	c b p	p: c b b: c
a f c e l p m n	f c a m p	p: f c a m m: f c a a: f c c: f

Reduce inputs (conditional databases) key: value		Conditional FP−trees
p:	{ f c a m / f c a m / c b }	{(c:3)} | p
m:	{ f c a / f c a / f c a b }	{ (f:3, c:3, a:3) } | m
b:	{ f c a / f / c }	{} | b
a:	{ f c / f c / f c }	{ (f:3, c:3) } | a
c:	{ f / f / f }	{ (f:3) } | c

Figure2:A simple example of distributed FP-Growth

3. The Proposed Mining Course Maps

This study investigates the relationship among decision variables and results by illustrating the course map according to data mining results. Course map describes association relationship on student data, course data, and transaction data and purpose the student demands for SNC, Thus this paper presents several SNC knowledge patterns and rules of language courses to implement new courses and pricing designs. The system begins at the database of the school, which each branch data set is integrated into the database of the head office. Thus it is possible to get the data from every branch quickly and easily. After clustering and grouping the data variables, the data will represent the customer needs and wants. This phase uses the data mining technique association rule to group and cluster the data in the same section. The algorithm used for association rule is the FP-growth algorithm. After gathering the results from the association rule, the diagram is built to illustrate the course map.

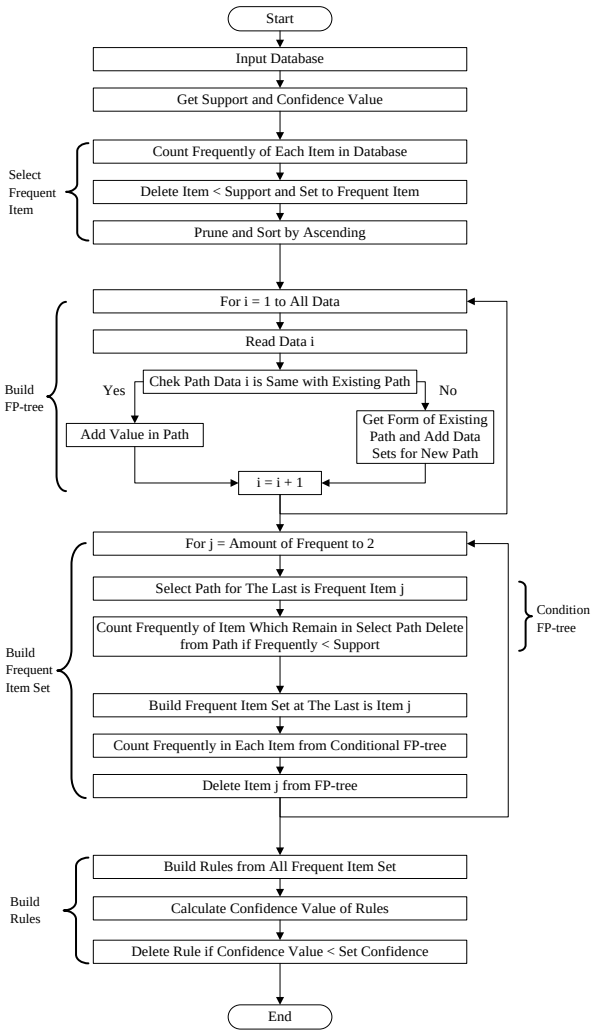

Figure 3: The proposed mining course maps methodology

The proposed mining course maps algorithm is shown in figure 3. According to FP-growth Algorithm, the MCM is separated in four steps as follows:

Step1: Select frequent items
After input database in mining process the first step is define support and confidence value for calculate

then passing through select frequent item stem step. This step starts at count frequently of each item in database. After that check items if item less than support, items are deleted. Set remain items in frequent item, sort and prune frequent item by ascending.

Step2: Build FP-tree
In build tree step, read and check data i from path if it same with existing path add the value in path if not get the form from existing path and add it for set the new path. Recursive until end at the last of data set.

Step3: Build frequent item sets
Start at j = 2, we have to set the condition first (condition FP-tree). In condition FP-tree, select path for the last at frequent item j then count frequently of items which remain in select path. Delete item from path if frequently less than support, end process for set condition of FP-tree. At the last item j, count frequently in each item from condition FP-tree, delete item j from FP-tree.

Step4: Build rules
In the last step, build rules from all frequent items set and calculate the confidence value of rules. Check rules and delete if confidence values from calculated less than defining confidence value.

4. Experimental Results

The proposed mining course maps algorithm in section 3 was experimented with a language course dataset. Each course is separated as follows: English course, Business Computer, Cambridge English, etc. The performance measures the execution time (seconds) of the methodology datasets with the following minimum support settings 2%, 4%, 6%, 8% and 10%, the minimum confidence was always set to 50%. We measured the execution time for both creating the frequent itemsets and creating the association rules whenever possible.

The computer used to run the experiments had a dual 400MHz Pentium M processors 725 and 1.6 GB of memory, the operating system used was Windows XP. To make the time measurements more reliable, no other application was running on the machine while the experiments were running. Although none of the methodology supported parallel processing, the second processor helped to stabilize the measured results since the system processes could run on the other processor.

Table 1: Execution time between ARMADA and MCM at Minimum Confidence 50% with 10,000 datasets.

Minimum Support (%)	Execution Time (Sec.)	
	ARMADA	MCM
2	170.4735	4.2229
4	146.9319	1.0337
6	104.3551	0.7301
8	86.0777	0.6071
10	56.295	0.4935

Table 2: Execution time between ARMADA and MCM at Minimum Confidence 50% with 20,000 datasets.

Minimum Support (%)	Execution Time (Sec.)	
	ARMADA	MCM
2	289.5806	2.4447
4	189.818	0.6545
6	141.8737	0.5755
8	111.879	0.537
10	65.0268	0.4792

Table 3: Execution time between ARMADA and MCM at Minimum Confidence 50% with 30,000 datasets.

Minimum Support (%)	Execution Time (Sec.)	
	ARMADA	MCM
2	434.5725	2.9228
4	262.3745	0.7668
6	261.902	0.6788
8	168.9704	0.5934
10	113.006	0.5465

Table 1-3 characterizes the tree datasets in terms of the number of transactions at 10,000/20,000/30,000. Comparison of execution time between ARMADA and MCM are showed in the table.

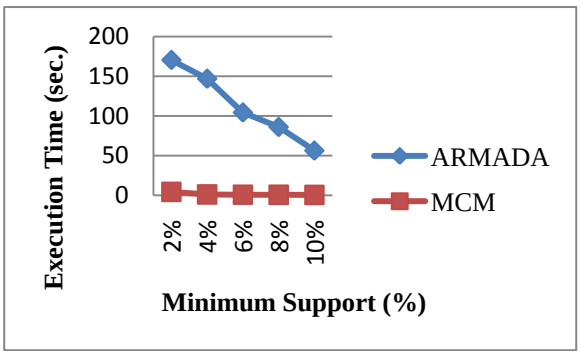

Figure4: Comparison graph of performance between ARMADA and MCM with 10,000 datasets.

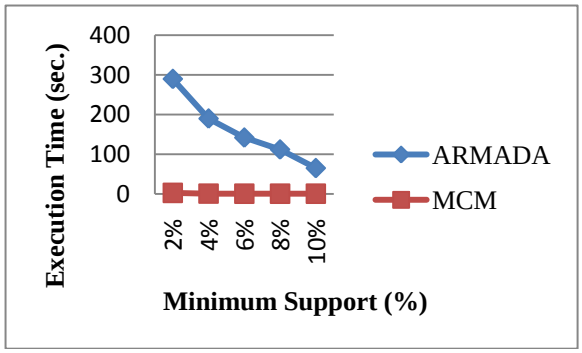

Figure 5: Comparison graph of performance between ARMADA and MCM with 20,000 datasets.

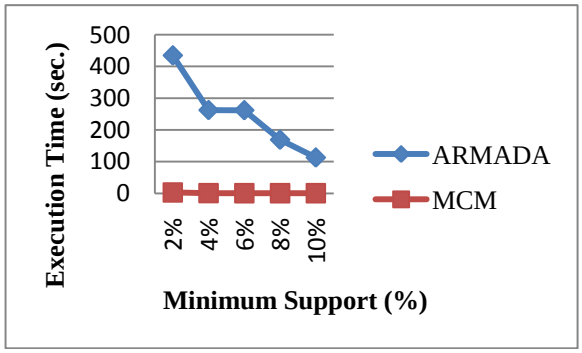

Figure 6: Comparison graph of performance between ARMADA and MCM with 30,000 datasets.

Figures 4 to 6 present comparison graphs of performance between ARMADA and MCM with three datasets. The first thing noticed about these results was that when the minimum support is large, the execution time is decreased. We focused only on the results where the minimum supports ranges from 2% to 10%.

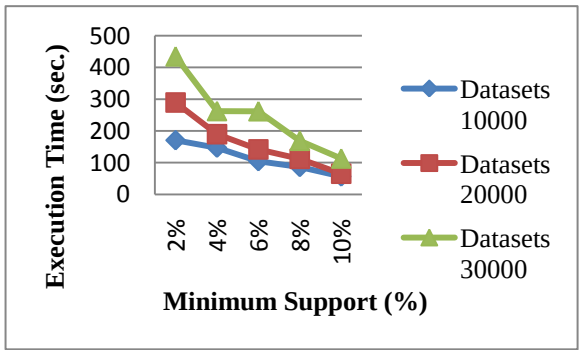

Figure 7: Comparison graph of performance between datasets by ARMADA.

Figure 7 shows the performance of ARMADA between datasets. If datasets are increase, the execution time must also increase. But ARMADA takes more time to run the process, usually more than minute extra.

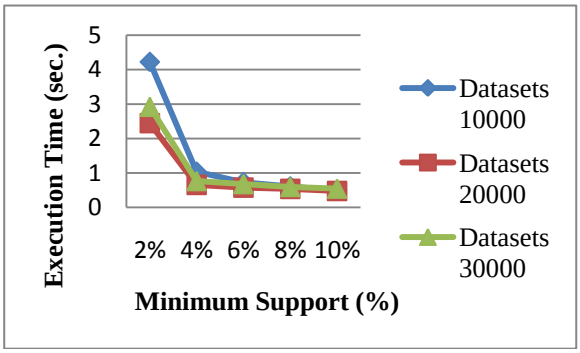

Figure 8: Comparison graph of performance between datasets by MCM.

The comparative graphs of performance between datasets by MCM are shown in figure 8. The execution time when datasets are 10,000 takes more time than other datasets. Second is 30,000 datasets and last are 20,000 datasets. However MCM still takes less time than ARMADA.

4.1 Extracted rules

All data are collected by customer service officer in each branch of language schools. The customer service officer in each branch will fill the data when students enroll the language course. The data include the general information and target of customer. After that the data are sent and collected in the database of head office. Data collection was then conducted between January 2007 and 2008 on branch locations in Thailand. Data are collected in 4 attributes as follow: Students course preference data, time of students' data, course duration data, date of study data.

Table 4: Example transaction of language courses database.

Course Type	Start Time	Course Duration	Date
11	1100	400	6
66	1700	400	1
66	1400	400	4
66	1400	400	5
77	1400	400	7
66	1300	400	5
66	1300	400	1
66	1300	400	5
77	1300	400	5
11	1100	400	1
11	1100	400	5
66	1100	400	5
55	1000	400	5
66	1000	400	6
66	1000	400	6

Table 4 show example transaction of language course database after preprocessing, first column show course type, second column show starting time, third column show course duration time and last column show date of study. The detail in each attribute of language course database is given in the table below.

Table 5: Detail of language course database.

Course Type	Start Time	Course Duration	Date
11:Junior	800:8 a.m.	100:1 hrs	1:Mon
22:Starter	900:9 a.m.	200:2 hrs	2:Tue
33:Advance	1100:11 a.m.	300:3 hrs	3:Wed
44:Intermediate	1300:13 p.m.	400:4 hrs	4:Thur
55:Elementary	1700:17 p.m.		5:Fri
66:Cambridge English	1800:18 p.m.		6:Sat
77:Business Com	1900:19 p.m.		7:Sun

The broad criteria used 10,000 transactions in mining the language course database were set minimum support as 10% and minimum confidence as 50%. Extracted rules of MCM can display by detail of language course database. Rules are showed in figure 9.

```
junior ->300 Sup=3234 Conf=80.8702
300 ->junior Sup=3234 Conf=71.4538
900 ->300 Sup=2298 Conf=88.6916
300 ->900 Sup=2298 Conf=50.7733
Sun ->junior Sup=2243 Conf=58.748
junior ->Sun Sup=2243 Conf=56.089
1300 ->300 Sup=2064 Conf=70.8791
Sun ->300 Sup=2027 Conf=53.0906
1300 ->junior Sup=1904 Conf=65.3846
Business Com ->200 Sup=1578 Conf=69.9468
900 ->junior Sup=1467 Conf=56.6191
Wed ->300 Sup=1396 Conf=65.4171
Cambride English ->200 Sup=1335 Conf=60.3526
Wed ->junior Sup=1256 Conf=58.8566
junior 1300 ->300 Sup=1814 Conf=95.2731
300 1300 ->junior Sup=1814 Conf=87.8876
junior 300 ->1300 Sup=1814 Conf=56.0915
Sun 300 ->junior Sup=1803 Conf=88.9492
Sun junior ->300 Sup=1803 Conf=80.3834
junior 300 ->Sun Sup=1803 Conf=55.7514
junior 900 ->300 Sup=1417 Conf=96.5917
300 900 ->junior Sup=1417 Conf=61.6623
Wed junior ->300 Sup=1103 Conf=87.8185
Wed 300 ->junior Sup=1103 Conf=79.0115
Sun 1300 ->junior Sup=1087 Conf=82.4109
junior 1300 ->Sun Sup=1087 Conf=57.0903
Sun 1300 ->300 Sup=1076 Conf=81.577
Sun 300 ->1300 Sup=1076 Conf=53.0834
300 1300 ->Sun Sup=1076 Conf=52.1318
```

Figure 9: Extracted rules from MCM methodology.

The first rule in this rule set (junior →300 sup=3234 conf=80.8702), can be interpreted as follows. The junior course will imply that school should open junior course for 3 hours. This rule is supported by 3,234 instances and 80.87% confidential in the accuracy of this rule.

4.2 Mapping graph

Holmlund and Strandvik [4] explain for marketing map to represent the best practice in marketing and also used the process map to understand how IT can be deployed in order to support a marketing information system. This research investigates the following research issues in the specific new course of language course

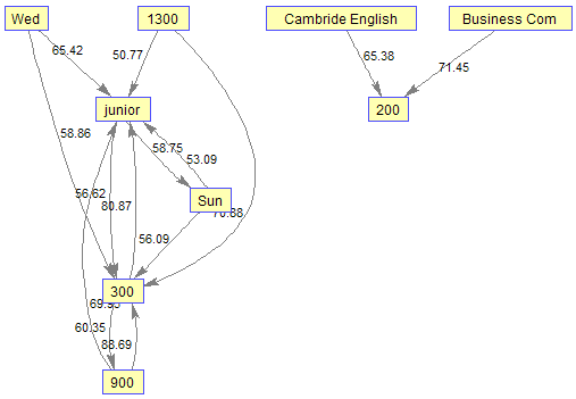

Figure 10: Language course mapping.

Mapping approach is helped to specific new course in this research. The language course mapping is displayed in figure 10. Course maps from MCM are easy to understand and make the decision, example if we open Cambridge English should be open for 3 hours. The confidential is 65.38%.

5. Conclusion

Student‘s demands are complex and sensitive. Knowing and understanding them will make efforts to fulfill student demands easier and also increase the profit and loyalty to the school. This paper proposes MCM as a methodology of association rules for data mining based on FP-growth.

The proposed MCM was evaluated and tested in the experiments compared with ARMADA. The execution time of MCM is much quicker than ARMADA, which means MCM is more efficient than the ARMADA.

The MCM can be implemented to make decisions and specify new courses for the customer. This will reflect the needs of customer, and help specifying future actions of the school. The course maps can show customer’s trends, which season or month, level of English course and future course advisements. The proposed method will help not only the school class planning, but also beneficial to the students.

References

[1] Keim, D, A., Pansea, C., Sipsa, M. and Northb, S. C, "Pixel based visual data mining of geo-spatial data", *Computer & Graphics*, Vol. 28, 2004. pp. 327-344.

[2] Hui, S. C. and Jha, G., "Data mining for customer service support," *Information & Management*, Vol. 38, Issue 1, 2000. pp. 1-13.

[3] Mehta, K. and Bhattacharyya, S., "Adquacy of training data for evolutionary mining of trading rules," *Decision Support Systems*, Vol. 37, Issue 4 2004. pp 461-474.

[4] Holmlund, M. and Strandvik, T., "Perception configurations in business relationships," *Management Decision*, Vol. 37, Issue 9, 1999. pp. 686-696.

[5] Liao, S. H., Hsieh, C. L. and Huang S. P., "Mining product maps for new product development," *Expert Systems with Applications*, Vol. 34, Issue 1, 2008. pp. 20-62.

[6] Manlone, J., *Association Rule Miner And Deduction Analysis (ARMADA)*, User Manaual, 2003. pp. 1-20.

[7] Said, A. M., Dominic, P.D.D. and Abdullah, A. B., "A Comparative Study of FP-growth Variations", *International Journal of Computer Science and Network Security*, VOL.9 No.5, May 2009, pp. 266-272.

[8] Jeffrey W. Seifert, *Data mining: And overview*, Congressional Research Service, Order Code RL31798, 2004.

[9] Trewartha, D., *Investigating Data Mining in MATLAB*, Bachelor (Honours) of Science Thesis of Rhodes University, 2006.

[10] Li, H., Wang, Y., Zhang, D., Zhang, M. and Chang, E., "PFP: Parallel FP-Growth for Query Recommendation", *ACM Recommendation Systems*, October 2008.

[11] Frawely, W. J., Shapiro, P. G., and Matheus, C. J. "Knowledge discovery in databases: An overview", *AAAI/MIT Press*, 1991, pp. 1-27.

[12] Erwin, A., Gopalan, R. P., and Achuthan, N. R. "CTU-Mine: An Efficient High Utility Itemset Mining Algorithm Using the Pattern Growth Approach", *IEEE 7th International Conferences on Computer and Information Technology*, 2007, pp. 71-76.

[13] Cunningham, S. J. and Holmes, G. "Developing innovative applications in agriculture using data mining," *the Proceedings of the Southeast Asia Regional Computer Confederation Conference*, Singapore, 1999.

[14] Wojciechowski, M., Galecki, K., and Gawronek, K. "Concurrent Processing of Frequent Itemset Queries Using FP-Growth Algorithm", *Proc. of the 1st ADBIS Workshop on Data Mining and Knowledge Discovery (ADMKD'05)*, Tallinn, Estonia, 2005.