Association Rule Mining Algorithm and Analysis of Missing Values

Jae-Wan Lee, *Member, KIMICS*, **Bobby D. Gerardo, Gui-Tae Kim**, *Nonmembers* and **Jin-Seob Jeong**, *Member, KIMICS*

Abstract—This paper explored the use of an algorithm for the data mining and method in handling missing data which had generated enhanced association patterns observed using the data illustrated here. The evaluations showed that more association patterns are generated in the second analysis which suggests more meaningful rules than in the first situation. It showed that the model offer more precise and important association rules that is more valuable when applied for business decision making. With the discovery of accurate association rules or business patterns, strategies could be efficiently planned out and implemented to improve marketing schemes. This investigation gives rise to a number of interesting issues that could be explored further like the effect of outliers and missing data for detecting fraud and devious database entries.

Index Terms–Association Rule Algorithm, Data Mining, Distributed Systems, Missing Data

I. INTRODUCTION

This paper on data mining will investigate the use of Apriori algorithm for data summarization and pattern extraction where missing data will be taken into account. The three tiers of data warehousing architecture on multidimensional data are considered in this study. The principle of rolling-up and drill down operations were considered to generate data cubes from large databases in order to optimize the process of data mining.

Data cubes have been well-adopted by users as a tool to analyze data collected in multi-dimensional way. However, often users are overwhelmed by the massive amount of data presented in a data cube; especially it is not uncommon for the number of cells to reach the magnitude of thousand or even million. This poses nontrivial challenges to address this using some measures within a data cube. Pre-computed aggregate calculations in a data cube can provide efficient query processing for OLAP applications.

Data warehousing has become a standard practice for most large companies worldwide. The data stored in the data warehouse captures many different aspects of the business process such as manufacturing, distribution, sales, and marketing. This data reflects explicitly and implicitly customer patterns and trends, business practices, strategies, know-how and other characteristics. Therefore, this data is of vital importance to the success of the business whose state it captures, which is why companies choose to engage in the relatively expensive undertaking of creating and maintaining the data warehouse.

While some information and facts can be gleaned from the data warehouse directly, much more remains hidden as implicit patterns and trends. The discovery of such information often yields important insights into the business and its customers and may lead to unlocking hidden potentials by devising innovative strategies. The discoveries go beyond the standard on-line analytical processing (OLAP) which mostly serves reporting purposes. One of the most important and successful methods for finding new patterns is association-rule mining. Typically, if an organization wants to employ association rule mining on their data warehouse data, it has to acquire a separate data mining tool. Before the analysis is to be performed, the data must be retrieved from the database repository that stores the data warehouse, which is often a bulky and time-consuming process. The vendors of data management software are becoming aware of the need for integration of data mining capabilities into database engines, and some companies are already allowing for tighter integration of their database and data mining software.

This paper will use data mining techniques as tool to process and analyze data as one of the output of the data mining result. The model that will be investigated will determine the functionality and efficiency for summarization and pattern extraction with missing data. In this paper we describe an approach to association-rule data mining within data warehouses that utilizes the model for missing data and then application of the algorithm for association rule mining without using a separate data mining tool.

II. RELATED WORKS

Data mining uses a various data analysis tools such as from simple to complex and advanced mathematical algorithms, to discover patterns and relationships in data

Manuscript received August 13, 2003.

J.W. Lee is with the School of Electronic and Information Engineering, Kunsan National University, San 68 Miryong-dong, Kunsan, Chonbuk 573-701, Korea (phone: +82-63-469-4696, fax: +82-63-469-4699, e-mail: jwlee@kunsan.ac.kr)

B.D. Gerardo is with the School of Electronic and Information Engineering, Kunsan National University, San 68 Miryong-dong, Kunsan, Chonbuk 573-701, Korea (phone: +82-63-469-4696, fax: +82-63-469-4699, e-mail: bgerardo@kunsan.ac.kr)

G.T. Kim is with the School of Electronic and Information Engineering, Kunsan National University, San 68 Miryong-dong, Kunsan, Chonbuk 573-701, Korea (phone: +82-63-469-4696, fax: +82-63-469-4699, e-mail: gtkimkorea@hanmail.net)

J.S. Jeong is with the School of Electronic and Information Engineering, Kunsan National University, San 68 Miryong-dong, Kunsan, Chonbuk 573-701, Korea (phone: +82-63-469-4696, fax: +82-63-469-4699, e-mail: jjsjss@hanmail.net)

that can be used to establish association rules and make effective predictions. The objective of data mining is prediction, or a generalization of data patterns. Data mining goes beyond the typical data queries, On-Line Analytical Processing (OLAP), visualization and other techniques for describing data.

Data mining is usually a group effort that requires expertise in algorithms and the data mining process. It requires a thorough knowledge of the problem domain in order to select variables and try a variety of data transformations that will lead to useful patterns. It also requires data management skills to assemble the database and make the identified transformations, and knowledge of the business along with application development skills to incorporate the data mining results into the organization's business processes [3].

A. Principle of Association Rule Mining

There are variety of data mining algorithms that have been recently developed that greatly facilitate the processing and interpreting of large database. One example is the association rule mining algorithm, which discovers correlations between items in transactional databases. The Apriori algorithm is an example association rule mining algorithm. Using this algorithm, candidate patterns which receive sufficient support from the database are considered for transformation into a rule. This type of algorithm works well for complete data with discrete values. One limitation of many association rule mining algorithms, such as the Apriori algorithm is that only database entries which exactly match the candidate patterns may contribute to the support of the candidate pattern [9]. This creates a problem for databases containing many small variations between otherwise similar patterns, and for databases containing missing values. Missing and noisy data is prevalent in data gathered today, particularly in business databases. There are reports that some data contain erroneous data entry. Important features may also be frequently missing from databases if collection was not designed with mining in mind. The goal of this research is to develop an association rule algorithm that accepts partial support from data. By generating rules, data can contribute to the discovery despite the presence of missing values [9].

Numerous data mining algorithms have been introduced that can perform summarization, classification, deviation detection, and other forms of data characterization and interpretation. One popular summarization and pattern extraction algorithm is the association rule algorithm, which identifies correlations between items in transactional databases.

For instance, given a set of transactions, an association rule $X \Rightarrow Y$ may be discovered in the data, where X and Y are combinations of items. The intuitive meaning of such rule is that transactions in the database which contain the items in X may also contain the items in Y. Association is established if it indicates that the support and confidence of the rule is beyond the minimum threshold. The support of the rule $X \Rightarrow Y$ represents the percentage of transactions from the original database that contain both X and Y. The confidence of rule $X \Rightarrow Y$ represents the percentage of transactions containing items in X that also contain items in Y. Various applications of association rule mining may include but not limited to cross marketing, attached mailing, record design and customer classifications.

The algorithm to discover association rule searches the possible patterns for rules that meet the minimum support and confidence thresholds. The Apriori algorithm which is an example of an association rule algorithm was developed by Srikant and Agrawal. The following are the two steps discovering association rules: (1) find all itemsets (sets of items appearing together in a transaction) whose support is greater than the specified threshold. Itemsets with minimum support are called frequent itemsets, and (2) generate association rules from the frequent itemsets. Confidence of a candidate rule $X \Rightarrow Y$ is calculated as *support* $(X \cup Y)$/*support*(X). All rules that meet the confidence threshold are reported as pattern or discoveries of the algorithm.

L_1 = {frequent 1-itemsets};
k:= 2; // k represents the pass number
While ($L_{k-1} \neq \varnothing$)
  C_k = New candidates of size k generated from L_{k-1}
  For all transactions t $\in$ D Increment count of all candidates in C_k that are contained in t
  L_k = All candidates in C_k with minimum support
  $k = k+1$
Report $\cup_k L_k$ as the discovered frequent itemsets

Where:		
k-itemset		An itemset containing k items
L_k		Set of frequent k-itemsets (Key-itemsets with minimum support)
C_k		Set of candidate k-itemsets
$\cup_k L_k$		Set of generated itemsets

Fig. 1 The Apriori algorithm

B. From Data to Knowledge

Learning the application domain will require determination of the relevant prior knowledge and goals of application. In this stage the data may be huge and complex which each geographically different database may have specific dimensions.

The data cleaning and preprocessing may need intensive attention because of time and efforts that is required. Few of the goals of data cleaning is creating a target data set by data selection and finding useful attributes, generate association, and discretization of values.

Deciding on the use of tools or algorithm is necessary to summarize, classify, regress, associate, and cluster data. The algorithm could produce patterns of interest which will then be considered as model.

The data could be then subject for interpretation and analysis which will become the results. The data could be presented by means of visualization, transformation and removing of redundant patterns. The discovered knowledge could be used for market, business or commercial purpose which will serve for decision making and other development plans.

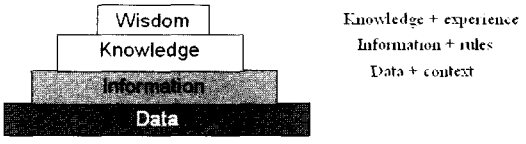

Fig. 2 The Data Pyramid

C. Mining from Aggregated Data

The association rule mining can be generated from transaction data that reflects items that are purchased. Often, in the case of many large corporations, such data is kept in the data warehouse only within a certain limited time horizon. Afterwards transaction-level data is stored off-line on a medium suited for bulk management of archival data, such as magnetic tape or optical disk. Such data is still available electronically but it is not on-line because it does not reside in the data warehouse [10].

The data warehouse continues to store summarizations of the transaction-level data, for example data aggregated by day. Aggregated data is created by rolling up the transaction-level data by one or more attributes. One of the most common cases is aggregating data by some measure of time. Other forms may include aggregating by using mining algorithm for cluster and association patterns.

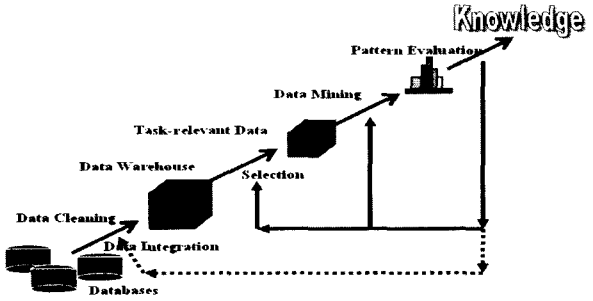

Fig. 3 Data mining as the core of knowledge discovery process

D. Problem in mining with missing data

Often, it is a challenge in data mining with missing data. There are varieties of approaches to deal with missing data. The most common method is to omit cases with missing values. Because deleting data with missing values may waste valuable data points, missing values are often filled. Missing values may be replaced with a special symbol that the mining algorithm ignores. The missing value may also be induced using standard learning techniques, though these approaches yield the most successful results when only one attribute is missing [9]. The popular approach is to assign missing values by globally replacing them with a single value such as the feature average before initiating the mining algorithm.

III. SYSTEM ARCHITECTURE

Typically, huge database in a distributed environment may contain missing data. With the presence of missing data, it will mean that there should be an effort to be done to handle the considerable amount of information that maybe lost. The purpose of data cleaning is eliminating entries and reducing the database by removing records with noisy data. This method, by simply deleting records, may not be effective because deleted data could still be valuable. Such data may significantly contribute to the outcome of the association rules which will be more sensible than eliminating it.

The proposed architecture describes that missing data are resolve first by using an algorithm mentioned in the preceding sections. Next, the data cubes generated is the result of the association rule algorithm using the Apriori algorithm. It could be noted that aggregated data in the form of data cubes is the result of data mining process.

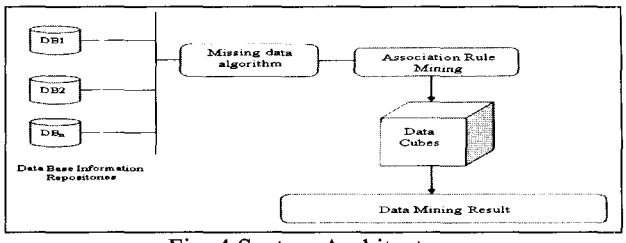

Fig. 4 System Architecture

The purpose of such cubes is to reduce the size of database by extracting dimensions that are relevant to the analysis. The process allows the data to be modeled and viewed in multiple dimensions [4]. The data cubes will reveal the frequent dimensions, thus, could generate pattern or rules from it. The final stage is the utilization of the result for decision making or strategic planning purposes. The rules are usually given in visual presentation. In this study, it is illustrated using the tables presented in the final section of this paper.

IV. APRIORI ALGORITHM AND MISSING DATA ALGORITHM

Instead of disregarding the missing value, we considered introducing solution for it because of the premise that cases or records with the presence of missing value might have contributions to the rule or pattern that could be obtained. The missing value may be corrected or replaced by using a missing value algorithm which will replace it with a value generated from the algorithm then apply the mining algorithm using Apriori for association rule mining.

A. Algorithm for Missing Data (AMD)

Ideally, there are two common ways to handle missing data, the pairwise deletion and the casewise deletion. When **pairwise** deletion of missing data is selected, then cases will be excluded from any calculations involving variables for which they have missing data. In the case of correlations, the correlations between each pair of variables are calculated from all cases having valid data for those two variables. When **casewise** deletion of missing data is selected, then only cases that do not contain any missing data for any of the variables selected for the analysis will be included in the analysis. In the case of correlations, all correlations are calculated by excluding cases that have missing data for any of the selected variables.

Statistically, two methods namely: (1) the mean substitution of missing data (replacing all missing data in a variable by the mean of that variable) and (2) pairwise deletion of missing data could either be used to avoid loosing data due to casewise deletion of missing data.

The mean substitution method is used to permanently remove missing data from dataset. Mean substitution offers some advantages and some disadvantages as compared to pairwise deletion. Mean substitution produces "internally consistent" sets of results or true correlation matrices. Its advantages are (a) artificially decreases the variation of scores and this decrease in individual variables is proportional to the number of missing data, and (b) because it substitutes missing data with artificially created "average" data points, mean substitution may considerably change the values of correlations.

There are several ways in handling database with missing data. The most common approach is to exclude cases with missing values, however, in this process the important contribution to the rule or pattern of the deleted tuples might be of significant value. The other common approach is to assign missing values by replacing them with a single value such as the average before initiating the mining process on the database. In this study, the procedure for missing data is illustrated in Figure 5.

B. Implementation of AMD and Apriori

Apriori algorithm is a level-wise search strategy used in Boolean association rule for mining frequent itemsets. This algorithm has an important property called Apriori property which is used to improve the efficiency of the level-wise generation of frequent itemsets. There are two steps in the implementation of Apriori property, namely the **join step** which will find L_k, a set of candidate k-itemsets by joining L_{k-1} with itself. The next step is the **prune step** in which C_k is generated as a superset of L_k, that is, its members may or may not be frequent, but all of the frequent k-itemsets are included in C_k. The Apriori property implies that any *(k-1)*-itemset that is not frequent cannot be a subset of a frequent k-itemset; hence, the candidate can be removed.

At this point, we will explore the algorithms stated by finding the association rules based on the transactions manifested in the table. The entry marked by "*x*" means that the record has missing data. For instance, we have the given **database Y** below:

Table 1 Database Y

TID	List of Items
100	Computer, Software, PCBooks, ComputerTable
200	Computer, Software, PCGames, ComputerTable, ColorPrinter, LaserPrinter
300	Computer, PCBooks, PCGames, "x", LaserPrinter
400	"x", PCBooks, PCGames, ColorPrinter, LaserPrinter
500	Computer, Software, PCBooks, PCGames
600	Computer, "x", "x", PCGames
700	PCBooks, PCGames, ComputerTable
800	ColorPrinter
900	PCBooks, PCGames, ComputerTable, "x", LaserPrinter
1000	Computer, Software, ComputerTable, ColorPrinter
1100	Computer, Software, PCGames, ComputerTable, ColorPrinter, LaserPrinter
1200	PCGames, ComputerTable, ColorPrinter, LaserPrinter

The support count corresponding to each candidate where transaction is removed because of the presence of missing data is given by the list denoted by L_r. Thus, we have L_r=[*Computer:5, Software:5, PCBooks:3, PCGames:4, ComputerTable:6, ColorPrinter:5, LasePrinter:*3].

On the other hand, the support count of each candidate where missing data is not disregarded is given by L_s. The set is L_s= [*Computer: 7, Software: 5, PCBooks: 6, PCGames: 7, ComputerTable: 7, ColorPrinter: 6, LasePrinter: 6*]

C. Steps:

(1) Find and replace missing values using the approximate value generated from the algorithm. Missing data algorithm handling may be (a) pairwise deletion, excluded cases from any calculations involving attributes for which they have missing data, (b) casewise deletion, include cases that do not contain any missing data for any of the attributes selected, and (c) replacement of value to missing data.

(2) In the first iteration of the algorithm, each item is a member of the set of candidate 1-itemsets, C_1. The algorithm simply scans all the transactions in order to count the number of occurrences of each item.

(3) Assume minimum support count from the given transaction. The set of 1-itemsets, L_1, can be determined. The L_1 consists of the candidates of 1-itemsets equal to or greater than the minimum support threshold.

(4) The next step is to find out the set of frequent 2-itemsets in L_2, the algorithm uses L_1X L_1, and this means that L_1 is joined with itself.

(5) The transactions in Y are scanned and the support count of each candidate itemset in C_2 is accumulated.

(6) The set of frequent 2-itemsets, L_2 is then determined, consisting of those candidate 2-itemsets in C_2 having minimum support.

(7) Next is to generate the set of candidate 3-itemsets, C_3. Again the algorithm uses L_2X L_2, and this means that L_2 is joined with itself.

(8) The transactions in Y is again scanned in order to determine L_3, consisting of candidate 3-itemsets in C_3 with minimum support. Non-frequent candidates are being removed in C_3 which will yield L_3.

(9) The process will continue until C_k found all the frequent itemsets, that is $C_k = \varnothing$.

Algorithms:

(a) Missing data algorithm. To find and replace missing values using the approximate value generated from the algorithm.

//missing data algorithm, handling may be any of the following

Pairwise Deletion
excluded cases from any calculations involving attributes for which they have missing data

exclude record where any dimensions have *k*-itemsets= "."
    generate L_k= *k*-itemsets(Y) // deleting cases where attribute has missing data

Casewise Deletion
include *cases that do not contain any missing data for any of the attributes selected*
    generate L_k= *k*-itemsets, where *any k-itemsets* = "."

Replacement of value to missing data
Assign *estimated value for missing data*

(b) Apriori algorithm. To find frequent itemsets based on the generation of candidates.

//Apriori algorithm

L_1 = find_frequent_1-itemsets(Y)
for (k=2; $L_{k-1} \neq \varnothing$, k++) {
    C_k= **apriori_gen** (L_{k-1}, *min_sup*);
    For each transaction $t \in$ Y {//scan Y for counts
        C_k= subset (C_k, t), // get the subsets of t that are candidates
        for each candidate $c \in C_t$
            c.count++;
    }
    L_k= {$c \in C_k | c.count \geq min\_sup$}
}
return $L=U_k L_k$ //as the discovered frequent itemsets

Fig. 5 Pseudo code using the missing data and Apriori for discovering frequent itemsets for mining association rules

D. Generating Association Rules from Frequent Itemsets

From the frequent itemsets obtained in the database, an association rule could be generated based on the two important criteria (1) the rules satisfy the minimum support threshold assumed and (2) the rule has greater confidence limit compared to the minimum confidence threshold assumed. The conditional probability illustrated by the equation a.1 was used to calculate for the confidence based on itemset support count.

$$confidence(X \Rightarrow Y) = \frac{Support\_count(X \cup Y)}{Support\_count(X)} \quad (a.1)$$

Where $Support\_count(X \cup Y)$ is the number of transactions containing the itemset $(X \cup Y)$, and $Support\_count(X)$ is the numbers of transactions containing the itemset X.

The association rules were generated by means of the following procedures: (a) generating all nonempty subsets of l, for every frequent itemset; and (b) for every nonempty subsets of l, the output rule is given by:

$$\text{“}s \Rightarrow (l\text{-}s)\text{” if } \frac{Support\_count(l)}{Support\_count(s)} \geq Min\_confidence\_threshold \quad (a.2)$$

The given rule in a.2 implies that it satisfies the minimum support threshold because the rules are generated from frequent itemsets as shown by equation.

V. EXPERIMENTAL EVALUATIONS

In the dataset given, about 33.3 % of the records were removed due to missing data. It could be noted that the proportion of deleted cases has a significant effect to the association rules generated. Other studies suggest that pairwise and casewise deletion in handling missing data is suitable if there are fewer than 5% of missing values present in proportion to the given dataset. The assumed support count is 4 or (33%) and the confidence threshold is 75%. Rule a.3 shows the frequent 3-itemsets generated using the Apriori property, pairwise and casewise deletion algorithm for missing data:

$$L_3 = \{\{Computer, Software, ComputerTable\}\} \geq min\_sup \quad (a.3)$$

It is interesting to observe that the frequent 3-itemsets generated is *{Computer, Software, ComputerTable}*. The subsets of L_3 are *{Computer, Software}*, *{Computer, ComputerTable}*, *{Software, ComputerTable}*, *{Computer}*, *{Software}*, and *{ComputerTable}*. With this frequent itemset, the association rules are shown in the table below with its corresponding confidence. Since the minimum support threshold is 75 %, the fourth association rule is ignored. The strong correlation is shown by the second and third rules, this implies that those who purchased *Computer* and *ComputerTable* will more likely to buy *Software* with confidence of 100 %.

Table 2 Association rules generated using pairwise and casewise deletion algorithms

	Association Rules	Confidence
1.	Computer^Software ⊃ ComputerTable	80.0 %
2.	Computer^ComputerTable ⊃ Software	100 %
3.	Software^ComputerTable ⊃ Computer	100 %
4.	ComputerTable ⊃ Computer^Software	67.0 %
5.	Software ⊃ Computer^ComputerTable	80.0 %
6.	Computer ⊃ Software^ComputerTable	80.0 %

If the missing values are not replaced, it is ideally ignored as presented in the previous table. Replacing the missing data would not mean that it could accurately predict closer estimate of data but to ensure that the entire dataset could not be fuzzy and thus could generate more precise correlation rule. Rules a.4 and a.5 reveal the frequent 3-itemsets, L_3 and the candidate 4-itemsets, L_4, respectively.

$$L_3 = \{\{Computer, Software, ComputerTable\}, \{PCGames, ComputerTable, LaserPrinter\}, \{PCGames, ColorPrinter, LaserPrinter\}\} \geq min\_sup \quad (a.4)$$

$$L_4 = \{\{Computer, Software, ComputerTable, LaserPrinter\}, \{Computer, Software, ComputerTable, LaserPrinter, ColorPrinter\}\} < min\_sup \quad (a.5)$$

There are three subsets indicating frequent itemsets in C_3. When generating candidates for frequent 4-itemsets, two subsets were produced as shown above but is less than the minimum support threshold so the two candidates were pruned. However, it can be noted that it expressed interesting 4-itemsets candidates which include *Computer, Software, ComputerTable, LaserPrinter or ColorPrinter*.

The resulting association rules for the three frequent subsets of L_3 are shown in the subsequent three tables with its corresponding confidence:

Table 3 Association rules for {Computer, Software, Computer Table}

	Association Rules	Confidence
1.	Computer^Software ⊃ ComputerTable	80.0 %
2.	Computer^ComputerTable ⊃ Software	100 %
3.	Software^ComputerTable ⊃ Computer	100 %
4.	ComputerTable ⊃ Computer^Software	67.0 %
5.	Software ⊃ Computer^ComputerTable	80.0 %
6.	Computer ⊃ Software^ComputerTable	80.0 %

In Table 3, the fourth rule is ignored because its confidence threshold is below the minimum confidence, while the other fives are retained. In table 4, the first, second, third and fourth rules are considered with minimum threshold greater than 75 %, respectively. While in Table 5, all rules except the sixth showed strong confidence, thus, retained.

Table 4 Association rules for {PCGame, Computer Table, laser Printer}

	Association Rules	Confidence
1.	PCGames^ComputerTable ⇒ LaserPrinter	100 %
2.	PCGames^LaserPrinter ⇒ ComputerTable	80.0 %
3.	ComputerTable^LaserPrinter ⇒ PCGames	100 %
4.	LaserPrinter ⇒ PCGames^ComputerTable	80.0 %
5.	ComputerTable ⇒ PCGames^LaserPrinter	57.0 %
6.	PCGames ⇒ ComputerTable^LaserPrinter	50.0 %

Table 5 Association rules for {PCGame, ColorPrinter, Laser Printer}

	Association Rules	Confidence
1.	PCGames^ColorPrinter ⇒ LaserPrinter	100 %
2.	PCGames^LaserPrinter ⇒ ColorPrinter	80.0 %
3.	ColorPrinter^LaserPrinter ⇒ PCGames	100 %
4.	LaserPrinter ⇒ PCGames^ColorPrinter	80.0 %
5.	ColorPrinter ⇒ PCGames^LaserPrinter	80.0 %
6.	PCGames ⇒ ColorPrinter^LaserPrinter	44.0 %

Efficiency using the mentioned algorithms is obtained by addressing two issues: (1) how to test (qualify) that an itemset has a pattern; and (2) how to exploit level-wise search [6]. In terms of efficiency, the model explored by this study showed that it generated more interesting patterns than simply ignoring the missing data.

In the first test, the algorithm generated one frequent 3-itemset and its corresponding association rules as shown in Table 2. In the second test, it generated three frequent 3-itemsets and its corresponding association rules as shown in Table 3, 4 and 5, respectively. This implies that more association patterns are generated in the second analysis which may suggest more meaningful rules obtained than in the first situation. The level-wise search runtime may be longer but further investigation to this effect may discover more efficient algorithm and association rules that could address the model proposed here.

VI. CONCLUSIONS

This paper explored the use of an algorithm for the data mining which had generated the association rules observed using the data illustrated here. We have provided examples and generated rules out of it but more rigorous treatment maybe needed if dealing with more complex database.

Ignoring missing data and deleting records may end up to an association rule which may express weaker data mining results. It is evident that replacing the missing data would not mean that it could accurately predict closer estimate of data, but to ensure that the dataset could provide a non-fuzzy correlation rule, and thus, will give more meaningful information than disregarding it.

The evaluations showed that more association patterns are generated in the second analysis which suggests more meaningful rules than in the first situation. By principle, the runtime versus the support threshold comparison may suggest that integrating algorithm for noisy or missing data may take longer, however, it will give us more precise and important association rules that could be more valuable for business decision making. With the discovery of accurate association rules or business patterns, strategies could be efficiently planned out and implemented to improve marketing ventures.

This investigation gives rise to a number of interesting issues that could be explored further. Another direction that could be studied is the effect of outliers and missing data. It is evident that outliers could significantly distort the normality of the dataset hence it is further proposed to investigate this matter. Treatment on this issue is anticipated to present more interesting pattern than merely ignoring them. Such investigation could lead the way to the application in business transaction and e-commerce to detect devious and fraud data records.

REFERENCES

[1] Agrawal and Srikant. Fast Algorithms for Mining Association Rules. Proceeding of International Conference on Very Large Databases VLDB, 1994, 487-499.

[2] Coenen, F. The Apriori Algorithm. http://www.csc.liv.ac.uk/~frans/KDD/aprioriTdemo.html#algorithm (2001).

[3] Edelstein, Herb. Data Mining: Can you dig it? http://www.teradatamagazine.com/articles/2003/vol3_no2/enterpriseviews/default.htm

[4] Han J. and Kamber M. Data mining concepts and techniques. USA: Morgan Kaufmann (2001).

[5] Handling missing or incomplete data. http://www.utexas.edu/cc/faqs/stat/general/gen25.html

[6] Hellerstein, J.L., Ma, S. and Perng, C. S. Discovering actionable patterns in event data. IBM Systems Journal, Vol. 41, No. 3, 2002.

[7] Knowledge Discovery in Databases. http://www.cs.ualberta.ca/~joerg/courses/cmput690/slides/Overview-s4.pdf

[8] Multi-Dimensional Constrained Gradient Mining. ftp://fas.sfu.ca/pub/cs/theses/2001/JoyceManWingLamMSc.pdf

[9] Nayak, Jyothsna R. and Cook, Diane J. Approximate Association Rule Mining. Proceedings of the Florida Artificial Intelligence Research Symposium, 2001.

[10] Nestorov, Svetlozar and Jukic, Nenad. Ad-Hoc Association-Rule Mining within the Data Warehouse. Proceedings of 36th Annual Hawaii International Conference on System Sciences, page 232a, January 2003.

[11] Pairwise Deletion of Missing Data vs. Mean Substitution. http://www.statsoftinc.com/textbook/glosp.html

[12] Text Mining. http://www.cs.waikato.ac.nz/~nzdl/textmining

Jae-Wan Lee
Received his B.S., M.S., and Ph.D. degrees in Computer Engineering from Chungang University in 1984, 1987, and 1992, respectively. Currently, he is a professor at the School of Electronic and Information Engineering in Kunsan National University, Kunsan City, Korea. His research interests include distributed systems, database systems, data mining, and computer networks. He is a member of KIMICS.

Bobby D. Gerardo
was born in Iloilo City, Philippines on March 18, 1972. He received his B.S. in Electrical Engineering from Western Institute of Technology in Iloilo City Philippines in 1994, and M.A. Ed. Mathematics from the University of the Philippines, Diliman Philippines in 2000. Currently, he is completing his Ph.D. degree in Electronic and Information Engineering at the Kunsan National University, Kunsan City, Korea. His research interests lie on the area of data mining, distributed systems, ubiquitous computing, and mobile communications.

Gui-Tae Kim
was born in Kangwon province, Korea on August 1, 1963. He received his M.A in Computer Science from the Ge-Myeong University in 1995. Currently, he is completing his Ph.D. degree in Electronic and Information Engineering at the Kunsan National University, Kunsan, Korea. His research interests lie on the area of data mining, distributed systems and mobile communications.

Jin-Seob Jeong
was born in Kunsan City, Korea on November 7, 1954. He received his B.S. in International Business from Wonkwang University, Iksan City Korea in 1984 and Master of Science in Information Technology from Kunsan National University, Kunsan City Korea in 2002. Currently, he is completing his Ph.D. degree in Electronic and Information Engineering at the Kunsan National University, Kunsan City, Korea. His research interests lie on the area of data mining, distributed systems and Load balancing in distributed environment. He is a member of KIMICS.