Marsland Press
Journal of American Science 2009:5(3) 52-62

Applying Ordinal Association Rules for Cleansing Data With Missing Values

[1]Azzam Sleit, [2]Mousa Al-Akhras, [3]Inas Juma, [4]Marwah Alian

[1]Computer Science Department, King Abdullah II School for Information Technology, Jordan University, P.O. Box 13898, Amman, 11942, Jordan, azzam.sleit@ju.edu.jo

[2]Computer Information System, King Abdullah II School for Information Technology, Jordan University, P.O. Box 13835, Amman, 11942, Jordan, mousa.akhras@ju.edu.jo

[3]Jordan University, Amman, Jordan, i.jumaa@ju.edu.jo

[4]Isra Private University, Amman, Jordan, marwah2001@yahoo.com

Abstract:

Cleansing data of errors is an important processing step particularly when integrating heterogeneous data sources. Dirty data files are prevalent in data warehouses because of incorrect or missing data values, inconsistent attribute naming conventions or incomplete information. This paper improves the data cleansing ordinal association rules technique by proposing a solution for the missing values problem. The approximated values for missing data items can be incorporated in the ordinal association rules. Experimental results confirm the effectiveness of the proposed enhancement.[Journal of American Science 2009:5(3) 52-62] (ISSN: 1545-1003)

Key word: Ordinal Association rules; Data cleansing; missing values; data mining.

1. Introduction

One important step in any data processing task is to verify the correctness of data values. Data cleaning also called data cleansing or scrubbing, detects and removes errors and inconsistencies in data in order to improve the quality of data. Causes of data quality problems include misspellings during data entry, missing data, invalid or incomplete information or other reasons such as inconsistent attribute naming conventions.

Data cleaning is especially required and should be addressed when integrating heterogeneous data sources in data warehouses. Data warehouses receive huge amounts of data from a variety of sources which may contain “dirty data” and is used in decision making.

One way of detecting data errors is by utilizing association rules which specify relationships between record attributes [Marcus et al., 2001]. An extension to this approach is to apply relational association rules to discover various relationships between attributes. The relational association rules can express various kinds of relationships between record attributes not only partial ordering relations [Campan et al., 2006; Hipp et al., 2000].

The term association rule was first introduced in the Apriori algorithm developed by Agrawal et al. in the context of market basket analysis [Agrawal et al., 1993]. Another work of using association rule induction utilized Apriori method to find duplicate relations in a representative biological dataset as introduced by Koh et al. [Koh et al., 2004]. Another research in the area of using association rules was done by Srikant et al. who considered the problem of discovering association rules in the presence of constraints that are Boolean expressions over the presence or absence of items [Srikant et al., 1996].

Marcus et al. introduced ordinal association rules to uncover relationships, numerical ordering or equality between attributes that commonly occur in the dataset which help in identifying attributes that do not conform to the discovered ordering [Marcus et al., 2001]. Later, Campan introduced relational association rules which are an extension of the ordinal association rules to be able to capture various kinds of relationships between record attributes [Campan et al., 2006]. Nayak and Cook described an association rule mining algorithm called approximate association rules (~AR), which is an enhancement of the Apriori algorithm [Nayak and Cook, 2001]. It allows data that approximately matches the pattern to contribute toward the overall support of the pattern which is useful in processing missing values.

The work by Hipp et al. compares between several algorithms dealing with association rules [Hipp et al., 2000]. Several researchers worked on the issue of resolving missing values in datasets [Ragel and Crémilleux, 1999; Calders et al., 2007; Wu et al., 2004; Othman and Yahia, 2008]. Rangel and Crémilleux proposed a method called Missing Values Completion (MVC) which is concerned with solving missing values in decision trees [Ragel and Crémilleux, 1999].

Calders et al. introduced new definitions for the terms support and confidence based on the absence of missing values in database for the attributes of the itemsets [Calders et al., 2007]. In addition, a new notation called representative was introduced to restrict the influence of itemsets that are not observed thoroughly by confidence and support. The XMiner algorithm was proposed using the new measures. An evaluation function and a completion procedure for finding missing values were presented by Wu et al. [Wu et al., 2004]. The proposed evaluation function is calculated according to the support, confidence, and the antecedent of the association rules.

The purpose of the work described in this paper is to improve the ordinal association rules algorithm by detecting and approximating missing values. By doing so, the approximated values for missing data items can be incorporated in the ordinal association rules. The rest of this paper is organized as follows: Section 2 presents a review of the definition of ordinal association rules. In section 3, the proposed algorithms for detecting errors and handling missing values are presented. Section 4 demonstrates experiments and results. Conclusions are drawn in section 5.

2. Ordinal Association Rules

Association rules are statements of the form $\{X_1, X_2, \ldots, X_n\} \Rightarrow X_{n+1}$ where X_i is a field in a dataset, meaning that if we know the values of $X_1, X_2, \ldots, X_n$ in a dataset, then we have a very a good chance of finding X_{n+1}. Association rules are required to satisfy a minimum support and a minimum confidence constraint at the same time [Hahsler et al., 2005]. To accept an association rule, a certain confidence of the rule is defined as the probability of finding X_{n+1}. Support and confidence are formally defined shortly in this section.

Definition:

A dataset R is a finite set of k records, $R = \{R_1, R_2, \ldots, R_k\}$ where each record R_i is a tuple of m attributes $\langle a_1, a_2, \ldots, a_m \rangle$. Each attribute a_j

has an associated domain (data type) such as integer domain. The domain of attribute a_j denoted by $Dom(a_j)$ consists of constants, and a special null or empty value (which is a member of every domain) and relational operators namely less than or equal ($\leq$), equal (=), and greater than or equal ($\geq$). A tuple $R_i \in Dom(a_1) \times Dom(a_2) \ldots \times Dom(a_m)$ over R is a sequence of data values $<v_1, v_2, \ldots, v_m>$ where the value v_i for attribute a_i is an element of $Dom(a_j)$.

Accordingly, set of attributes can be related in the following way $(a_1, a_2, \ldots, a_l) \Rightarrow (a_1\ op_1\ a_2\ op_2\ a_3\ op_3\ a_4 \ldots\ op_{l-1}\ a_l)$ where each op_i is either $\leq, =, or \geq$, is an ordinal association rule if:

1. $a_1, a_2, \ldots, a_l$ are not empty and occur together in at least S% of the K records, where S is called the support of the rule, and
2. In a subset of the records, $R' \subseteq R$ where $a_1, a_2, \ldots, a_l$ occur together and $v_1\ op_1\ v_2\ op_2\ v_3 \ldots op_{l-1}\ v_l$ is true for each $R_i \in R'$. Thus |R'| is the number of records that the rule holds for and the confidence C of the rule is the percentage of records that hold for the rule; Confidence=|R'|/|R| [Marcus and Maletic, 2000].

Error detection using ordinal association rules has two steps [Marcus et al., 2001; Rahm and Do, 2000]:

1. Discover ordinal rules with a minimum confidence C.
2. Determine data that can be considered as potential errors (outliers).

The proposed approach introduces other steps to deal with the problem of missing values as explained in the next section.

3. Handling Missing Values

A variety of approaches have been proposed to deal with the problem of missing data. The most common and simplest approach is to omit cases with missing values. Other approaches replace missing values with a special symbol using a standard learning technique [Nayak and cook, 2001, Shen et al., 2007; Lin and Tseng, 2006].

Our approach consists of two phases. The first phase enhances the algorithms of the ordinary ordinal association rules for comparing data and analyzing records. The purpose of this phase is to discover the ordinal association rules from a dataset that may include missing values. The second phase uses the discovered rules in the first phase and analyzes the records to detect potential data outliers. Then, it attempts to obtain the best approximate values for the missing items. The first phase may repeatedly be applied to confirm existing rules. Fig. 1 depicts the data flow of the proposed approach in its two phases for finding ordinal association rules and for detecting errors and approximating missing values. The proposed algorithm accepts a dataset with missing values and outputs the discovered ordinal association rules and cleaned dataset after replacing missing values with their approximations.

In the first step, data is normalized by converting it from its original form into numeric value. For instance, date values such as “23-12-1987” will be converted into the form 23121987. Then, for each record, each pair of fields in the normalized record is compared. An ordinal relationship of attributes will be constructed between each pair of different attributes. For example, if a_1 and a_2 are two different attributes and $a_1 > a_2$, then “>” is the ordinal relationship between a_1 and a_2. Such relationship between the two attributes is called a pattern and is saved in a comparison file.

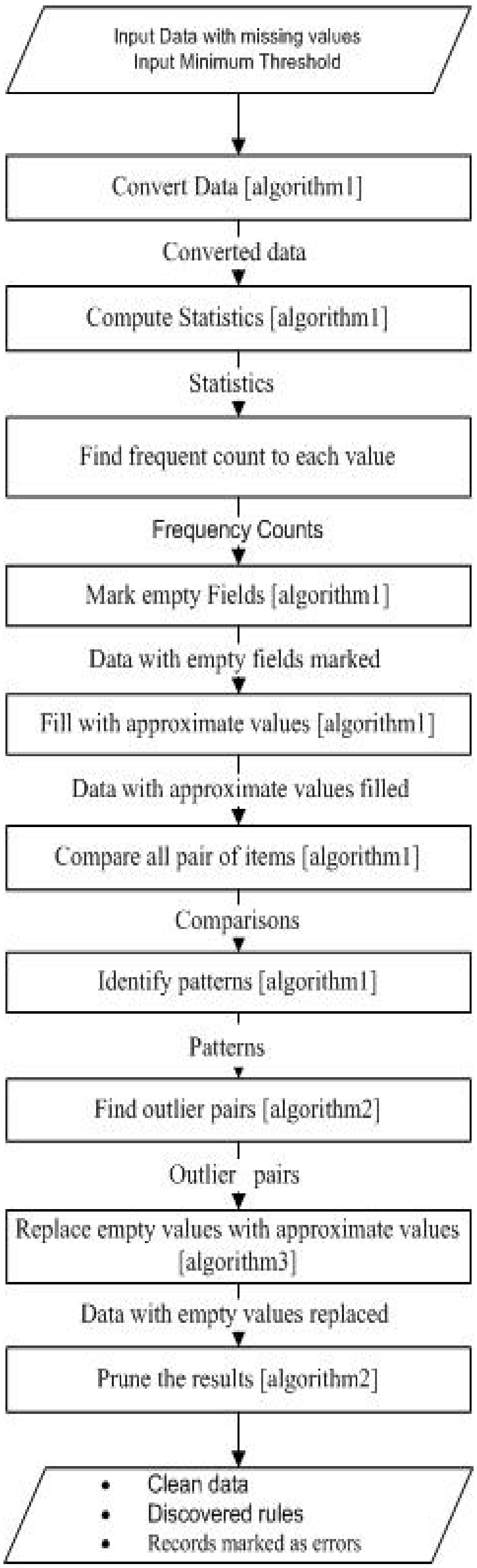

Fig. 1 Dataflow diagram for applying ordinal association rules for cleansing data with missing values.

After the comparison file has been constructed, results are passed to the next step where the patterns that hold true for at least the chosen minimum confidence are identified. Next, identified patterns are analyzed to find outlier pairs and empty values are replaced with their approximations. Finally, only those records that violate the rules are marked as possible errors.

During dataflow of the proposed approach, various algorithms are called. The details of the components, mainly the Compare Item, Best Approximate Value, and Analyze Records algorithms are outlined in Figures 2, 3, and 4, respectively. The main target of algorithm 1 is to find the patterns after running comparisons to save them in a file or database to be utilized by the remaining algorithms.

Before running the Compare Item algorithm a minimum threshold and a minimum confidence should be determined as input for the algorithm. Minimum threshold is typically determined by a field expert responsible for data accuracy while minimum confidence is chosen empirically by applying all aforementioned algorithms on different values then selecting the most appropriate confidence percentage. Compare Item algorithm turns all data in the original dataset into numeric. Afterwards, a new database called frequent transactions database is created that includes: attribute name, attribute value and frequency count of the field value. The intuition behind the creation of the frequent transactions database is to find the most frequent value of the attribute that has missing values. However, the most frequent value of an attribute may not the value to be used to replace the missing value. If the attribute does not have a frequent value, the missing value will not be replaced.

Algorithm 1: Compare Item (Minimum threshold percentage Min_{th} , Minimum Confidence)

Input: Sample of data from a dataset that includes missing values.

Minimum threshold percentage Min_{th} which is the percentage of frequency of items values.

Minimum Confidence which is the percentage of records that hold for the rule.

Output: Comparison file that only includes patterns with minimum confidence

1. For each record in the dataset
   - 1.a. Normalize or Convert Data
   - 1.b. Insert or (Update if exist) into frequent transactions database each attribute name, its value and the frequent count of the corresponding value of the attribute.
   - 1.c. Update Statistics.
2. For each record in the dataset
   - 2.a. Get all attributes that have missing values.
   - 2.b. Get the highest frequent value greater than or equal to the minimum threshold Min_{th} from the frequent transactions database of the corresponding attribute that have missing value.
   - 2.c. Update the database by replacing the missing value with the value with the highest frequent value of the corresponding attribute.
   - 2.d. Compare each attribute in the record with all other left hand attributes.
     - 2.d.1. Update the comparison file.

Fig. 2 Algorithm Compare Item

Algorithm 2: Best Approximate Value (Record Identifier, Attribute)

Input: Comparison File which includes patterns, Record Identifier

Attribute Name, Datasets which includes Missing Values

Output: Best Approximation for the missing value

1. Find the patterns with equal type and the input attribute is one of the pair.
   - 1.a. If the pattern of equal operation exists then
     - 1.a.1. Apply the pattern of equal operation on the given record.
     - 1.a.2. Update the dataset by replacing the missing value in the input attribute with the corresponding equal value that matches the same pattern.
2. If the patterns of equal operation does NOT exist then
   - 2.a. Find all patterns that include the given attribute.
   - 2.b. Apply the patterns on the input record.
   - 2.c. Get the largest attribute's value that is smaller than the input attribute and call it MIN_v
   - 2.d. Get the smallest attribute's value that is larger than the input attribute and call it MAX_v.
   - 2.e. Get the value of the input attribute that satisfy the maximum frequent count from frequent transaction database and its value is between MIN_v and MAX_v.
   - 2.f. Update the dataset by replacing the missing value in the input attribute with the max frequent count value.
   - 2.g. If there is NO value with maximum frequent between MIN_v and MAX_v then
     - 2.g.1. Update the database by replacing the missing value in the input attribute with $(MIN_v + MAX_v)/2$.

Fig. 3 Algorithm Best Approximate Value.

Algorithm 3: Analyze Records

Input: Comparison file which includes patterns with minimum confidence.

Output: Attributes marked as high possible error attributes.

1. For each record in the dataset
   - 1.a. For each rule in the pattern file
     - 1.a.1. Determine rule type and pairs
     - 1.a.2. If there are empty values in the pairs Then
       - 1.a.2.1. Apply Approximate Data Value (Record Identifier., Attribute Name) algorithm
     - 1.a.3. Compare item pairs
     - 1.a.4. If pattern does NOT holds Then
       - 1.a.4.1. mark each attribute as possible error
   - 1.b. Compute average number of marks
   - 1.c. Select the high probability marked errors

Fig. 4 Algorithm Analyze Records.

The target of the replaced value is to find an initial replacement for the missing values to discover patterns which is the purpose of the first phase. In the second phase, the best approximation for the missing value that can successfully satisfy the pattern is found taking into account that the discovered missing value in the first phase will still be regarded as missing value during the course of running the algorithms in the second phase and they may be changed in order to get a better approximation value.

During the second phase Analyze Records (Algorithm3) and the Best Approximate Value (Algorithm2) are called. The second phase starts by executing Algorithm3 where all records are manipulated and each pattern is scanned. The record identifier and each attribute in the pairs of the pattern are used as input parameters to algorithm2 during which input patterns are searched with equality operation. Pairs marked as attributes with missing values are replaced by the same attribute value in the same record using the matching pattern. If there is more than one equality rule that includes the attribute, a transitive closure operation is applied; that is, A=B and B = K ➔ A= K. If no such rule exists, search using other rules proceeds. Other steps such as finding all patterns that have the input attribute are used to determine the range for the missing value. In other words, the maximum and minimum possible values are determined. The maximum frequent value is also considered to get appropriate recurring value. If none of the previous conditions are met, the missing value is replaced by (min + max)/2 since the target is to reduce the number of errors as much as possible.

The Analyze Records algorithm, shown in Fig. 4 scans all patterns and applies them on the records of the dataset. Each pair of attributes that corresponds to a pattern is checked to see if values of the attributes match the patterns. If they do not, each attribute is marked as possible error. After this step, the average number of possible error marks for each considered attribute is computed. Those attributes that are marked as possible error with values greater than the average will be considered as high possible error attributes. Table 1 displays a sample database which contains six records each has five attributes of numeric data type. If attribute A is larger than attribute B over a large percentage of records or in other words greater than the minimum confidence then we can discover (establish) the ordinal rule A>B. However, as shown in Table 1, the value of attribute B at record 5 is 11 larger than the value of attribute A which is 10.5. Therefore, this record does not conform to the above ordinal rule. By scanning the whole table, the obvious relation between A and B is A>B concluding that the value at record 5 may be considered as error.

Table 1: A sample of data used in phase1

Record #	A	B	C	D	E
1	5	4	9	5	2
2	7	3		7	1.23
3	8	3	10	8	3
4	16	7.9	9	16	19
5	10.5	11	13	10.5	9
6	7	3	9	7	5.5

The missing value for attribute C of the second record will be replaced by the approximate value. Approximate value is derived from the probability of distribution which represents the likelihood of possible values calculated using frequency counts of data for the corresponding attribute. In our case P(C=9) = 3/6 or 1/2, P(C=10) = 1/6, and P(C=13) = 1/6. Considering the minimum threshold to be 0.5, the most probable approximate value for C is 9. To emphasize the correctness of the value, we compare it with the min, max, mean and standard deviation. This rule and all other derived rules will be saved in a database.

The second phase of the proposed approach analyzes records in an attempt to find the high probability of errors in data in order to obtain the best approximation for the missing values after applying the derived patterns. Thus the percentage of errors will decrease. Table 2 shows a sample dataset, with the same structure as Table 1 which will be used in this part.

Table 2: A sample of data used in phase 2

Record #	A	B	C	D	E
1	15	2.6	9	15	3
2	70	3	9	70	9
3	6	2	10	6	3
4		8.1	9	16	9
5	10	7	13	10	
6	29	3	9	29	9

The set of discovered rules from this table are listed in Fig. 5. The first step of phase 2 is to scan each pattern (discovered rule) and determine the rule type and the involved pair of attributes. If one or two of the paired attributes have empty values, then Approximate Data Value algorithm described in Fig. 3 is called.

A>B
A<C
A=D
A>E
B<C
B<D
B<E
C>D
C>E
D>E

Fig. 5: The discovered ordinal rules for Table 2

In our example, the missing value in attribute A in the 4th record of Table 2 (Table 2 includes data from the same dataset which was used to find patterns in phase1) is replaced by the value 16 since one of the discovered patterns is A=D (Fig. 5) and the corresponding value for attribute D is 16. On the other hand, the missing value of attribute E in the 5th record of Table 2 needs more computations to conclude since there is no pattern that can be deducted using E and another attribute. Again by scanning all patterns we can find all attributes less than or equal to attribute E and all attributes larger than or equal to attribute E. The next step is to compare the larger attributes with E attribute and compare the smaller attributes with E. In our case, the rules of interest are A>E, C >E, D>E and B<E. By combining the above discovered rules, we can conclude that B<E<min(A,C,E).

By applying the patterns at the same record (10>E, 13>E, 10>E, and 7<E), the value of attribute E will be constrained between 7 and 10. This is because only attribute B is less than or equal to E. The value of B at the same record (which is equal to 7) is the minimum value for E and as attributes A and D which have the same value 10 are greater than E. In other words, the missing value for E at record number 5 is between 7 and 10. To find the approximate value, we need to get a high frequent value that should be in the range of the attribute, in this case between 7 and 10. Also confidence intervals as a statistic measure must be taken into consideration. After these computations, the best approximate value is 9.

4. Experiments and Results

Experiments to demonstrate the effectiveness of the proposed approach were conducted using PL/SQL under Windows platform. Data was generated randomly for 31 attributes and 5000 records. Statistical functions such as min, max, standard deviation, mean and confidence intervals were calculated for each attribute and saved in the database. The minimum confidence was chosen empirically since the data may have different distribution other than normal distribution. Several experiments were performed with various minimum confidence values starting with 0.95 and the programs were executed to find the possible number of discovered patterns, number of records with attributes which do not satisfy the discovered patterns, number of possible errors in all attributes of all record, and high probability errors as shown in Table 3.

Outlier pairs were identified statistically and each attribute value was compared with its confidence interval taking into consideration the range of the field values. The second part of our approach decreases the average number of possible errors generated by the algorithm introduced by Marcus et al. [Marcus et al., 2001]. The original algorithm treats records with empty values as errors while our proposed enhancement handles missing values before the error detection phase. Comparisons were made between the set of rules and results obtained using the proposed approach and those obtained by Marcus et al. The minimum threshold controls the level of accuracy so that by increasing the minimum threshold the approximate values in the first part become more accurate. Fig 6 shows the relation between the rules and their support before and after handling missing values with a minimum threshold value of 0.1.

For the enhanced approach, it can be seen from Fig. 6 that the support for many rules increases after approximating the missing values, whereas missing values in the Marcus et al. algorithm are discarded and play no role in the support of the rule. Also, the second phase of the enhanced approach decreases the average possible errors generated by the Marcus et al. algorithm. Fig. 7 shows the relationship between the fields and the number of possible errors. It can be seen from the figure that after handling missing values, the number of possible errors decreases.

Table 3: Experiment runs for randomly generated data (5000 records with 31 fields)

Number of high probability errors	Number of possible error	Number of records with error	Number of patterns	Minimum confidence
18	59069	517	114	0.95
18	47512	417	97	0.96
18	40145	415	94	0.97
17	40060	54	54	0.98
16	35016	54	52	0.99
16	44943	54	52	0.991
16	44781	54	52	0.992
16	35142	54	52	0.993
16	35016	54	52	0.994
16	40057	49	46	0.995
13	44637	35	41	0.996
13	39919	39	35	0.997
10	28888	12	10	0.998
9	28000	12	10	0.999
0	0	0	0	1

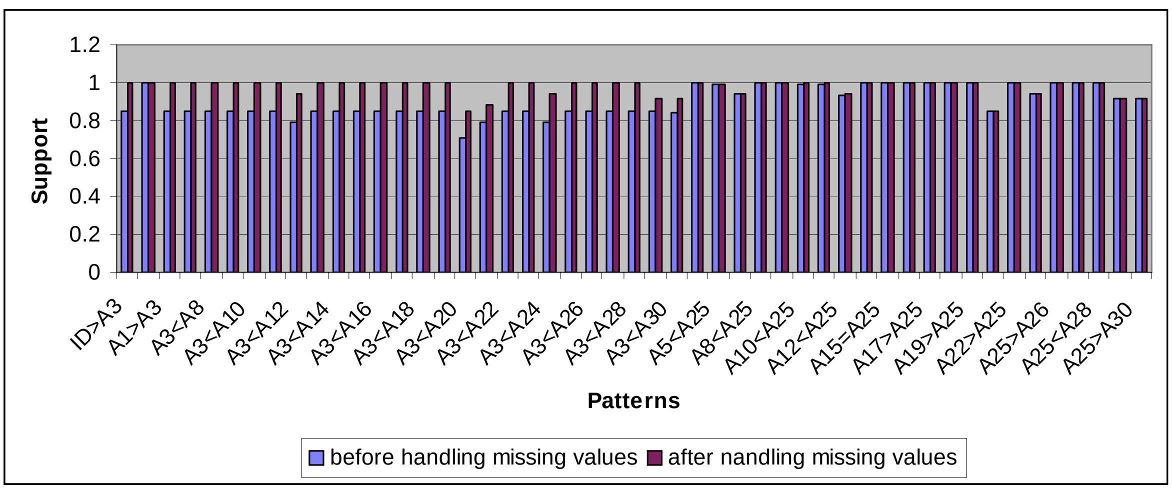

Fig. 6: Relationship between patterns and support

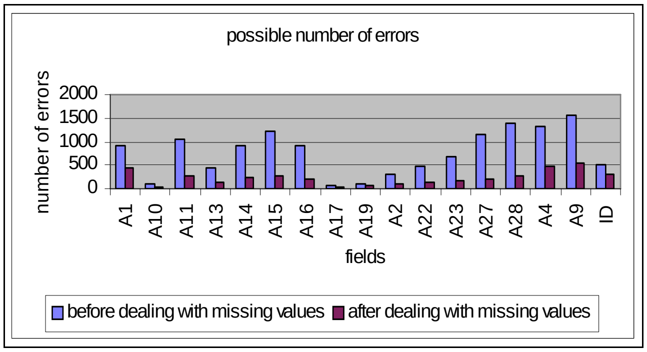

Fig 7: Relation between fields and number of errors

5. Conclusion

Association rule mining is useful in identifying not only interesting patterns for fields such as market basket analysis, but also patterns that uncover errors in datasets [Calders et al., 2007]. This research proposes an enhanced algorithm which handles missing values that are considered as errors when applying ordinal association rules described by Marcus et al. [Marcus et al., 2001]. The number of errors discovered after applying the proposed approach is far less than that discovered by the original algorithm. The first phase of the enhanced approach derives approximate values using the ordinal association rules from a dataset that may include missing values. The second phase identifies data with high probability errors to obtain the best approximations for the missing values after applying the derived patterns. The proposed approach must be seen as a complement not a substitute for Marcus et al. ordinal association rules algorithm.

This work is convenient for those dealing with budgetary datasets. It provides approximate values instead of missing values, thus eliminating the necessity to discard such records. This helps in making estimations more realistic and accurate by including more records.

References

Agrawal, R., Imielinski, T., Swami, A., 1993. Mining Association Rules Between Sets of Items in Large Databases. In ACM SIGMOD International Conference on Management of Data. Washington D.C. 207-216.

Calders, T., Goethals, B., and Mampaey, M., 2007. Mining Itemsets in the Presence of Missing Values. In Proceedings of the ACM Symposium on Applied computing. 404-408.

Campan, A., Serban, G.R., Marcus, R., 2006. Relational Association Rules and Error Detection. Studia University Babes–Bolyai, Informatica. L1(1): pp. 50-58.

Hahsler, M., Grün, B., Hornik, K., 2005. A Rules - A Computational Environment for Mining Association Rules and Frequent Item Sets. Journal of Statistical Software. 14(15), 1-25.

Hipp, J., Guntzer, U., Nakhaeizadeh, G., 2000. Algorithms for Association Rule Mining - A General Survey and Comparison. SIGKDD Explorations Journal. 2(1): 58-64.

Koh, J. L.Y., Lee, M. L., Khan, A. M., Tanand, P., Brusic, V., 2004. Duplicate Detection in Biological Data using Association Rule Mining. In Proceedings of the Second European Workshop on Data Mining and Text Mining in Bioinformatics. Pisa, Italy, 122-129.

Lin, W.Y., Tseng, M.C., 2006. Automated Support Specification for Efficient Mining of Interesting Association Rules. Journal of Information Science. 32(1): 238 - 250.

Marcus, A., Maletic, J.I., 2000. Utilizing Association Rules for Identification of Possible Errors in Data Sets. Technical Report CS-00-03. Kent State University.

Marcus, A., Maletic, J. I., Lin, K, 2001. Ordinal

Association Rules for Error Identification in Data Sets. In Proceedings of the ACM CIKM International Conference on Information and Knowledge Management. Atlanta, Georgia, USA. 589 – 591.

Nayak, J. R., Cook, D. J., 2001. Approximate Association Rule Mining. In Proceedings of the Fourteenth International Florida Artificial Intelligence Research Society Conference. 259 – 263.

Othman, L.B., Yahia, S.B., 2008. Yet Another Approach for Completing
Missing Values. LNCS 4923. 155–169.

Ragel, A., Crémilleux, B., 1999. MVC-A Preprocessing Method to Deal with Missing Values. Knowledge-Based Systems. 285–291.

Rahm, E., Do, H., 2000. Data Cleaning: Problems and Current Approaches. IEEE Technical Bulletin on Data Engineering. 34-42.

Shen, J.J., Chang, C.C., Li, Y.C., 2007. Combined Association Rules for Dealing with Missing Values, Journal of Information Science, 33(2): 468-480.

Srikant, R., Vu Q., Agrawal, R., 1996. Mining Association Rules with Item Constraints. In SIGMOD International Conference on Management of Data. Montreal, Canada, 1-12.

Wu, C., Wun, C., Chou, H., 2004. Using Association Rules for Completing Missing Data. Fourth International Conference on Hybrid Intelligent Systems. 5(8), 236-241.