Technical Report CS-00-04
Utilizing Association Rules for the Identification of Errors in Data[1]

Andrian Marcus
Jonathan I. Maletic[2]

Division of Computer Science
The Department of Mathematical Sciences
The University of Memphis
Campus Box 526429
Memphis, TN 38152

amarcus@memphis.edu, jmaletic@memphis.edu

Abstract: The paper analyzes the application of association rules to the problem of data cleansing and automatically identifying potential errors in data sets. Association rules are a fundamental class of patterns that exist in data. These patterns have been widely utilized (e.g., market basket analysis) and extensive studies exist to find efficient association rule mining algorithms. Special attention is given in literature to the extension of binary association rules (e.g., ratio, quantitative, generalized, multiple-level, constrained-based, distance-based, composite association rules). A new extension of the boolean association rules, *ordinal association rules*, that incorporates ordinal relationships among data items, is introduced. These rules are used to identify outliers in data. An algorithm that finds these rules and identifies potential errors in data is proposed. A prototype tool is described and the results of applying it to a real-world data set are given. The tool is designed to use as little domain knowledge as possible and constitutes the first part in a proposed framework for automated data cleansing. Other approaches to data cleansing are described and compared.

[1] This research is supported in part by a grant from the Office of Naval Research.
[2] Contact Author

1. Introduction

The quality, correctness, consistency, completeness, and reliability of any large real world data set depend on a number of factors [Wang95, Wang96, English99]. But the source of the data is often times the crucial factor. Data entry and acquisition is inherently prone to errors both simple and complex. Much effort is typically given to this front-end process, with respect to reduction in entry error, but the fact often remains that errors in a large data set are common. Unless an organization takes extreme measures in an effort to avoid data errors the field errors rates are typically around 5% [Redman98, Orr98]. Where the error rate is equal to the number of error fields over the number of total fields. There is a wide range of impacts to the organization including higher operational costs, poorer decision-making, increased organizational mistrust, and diversion of management attention.

The logical solution to this problem is to attempt to cleanse the data in some way. That is, explore the data set for possible problems and endeavor to correct the errors. Of course, for any real world data set, doing this task "by hand" is completely out of the question given the amount of person hours involved. Some organizations spend millions of dollars per year to detect data errors [Redman96]. A manual process of data cleansing is also laborious, time consuming and, prone to errors. The automation of the data cleansing process for large sets of data may be the only practical and cost effective way to achieve a reasonable quality level in a data set. While this may seem to be an obvious solution, little research has been directly aimed at this problem. Related research addresses the issues of data quality [Ballou89, Redman98, Redman96, Wang95] and tools to assist in "by hand" data cleansing and/or relational integrity checking (e.g., [EDD99, ETI99, Pak93, QMSoft99, Trillium99, Vality99, Wang93]).

1.1. Data Cleansing

Data cleansing is a relatively new field. The process is computationally expensive and thus it is almost impossible to do with old technology. The new faster computers allow performing the data cleansing process in acceptable time on large amounts of data. There are many issues in the

data cleansing area that researchers are attempting to tackle. They consist of dealing with missing data, determining record usability, erroneous data etc. Different approaches address different issues. Of interest to this research is the search context for what is called in literature and the business world as "dirty data" [Hernandez97, Kimball96, Fox94, Flanagan98, English99].

There is no commonly agreed definition of the data cleansing. Various definitions depend on the particular area in which the process is applied. Three major areas include data cleansing as part of their defining processes: data warehousing, knowledge discovery in databases (KDD), and total data quality management (TDQM).

1.2. Related Work

Within the data warehousing field, data cleansing is applied especially when several databases are merged. Records referring to the same entity are represented in different formats in the different data sets or are represented erroneously. Thus, duplicate records will appear in the merged database. The issue is to identify and eliminate these duplicates. The problem is known as the *merge/purge problem* [Hernandez97, Galhardas99, Moss98]. Instances of this problem appearing in literature are called record linkage, semantic integration, instance identification, or object identity problem.

From this perspective data cleansing is defined in several (but similar) ways. In [Galhardas99] data cleansing is the process of eliminating the errors and the inconsistencies in data, and solving the object identity problem. The [Hernandez97] paper defines the data-cleansing problem as the merge/purge problem and proposes the basic sorted-neighborhood method to solve it. The proposed method is the basis for the DataBlade module of the DataCleanser tool [EDD99].

Data cleansing is much more than simply updating a record with good data. Serious data cleaning involves decomposing and reassembling the data. According to [Kimball96] one can break down the cleansing into six steps: elementizing, standardizing, verifying, matching, house

holding, and documenting. Although data cleansing can take many forms, the current marketplace and the current technology for data cleaning are heavily focused on customer lists [Kimball96]. In this area, three companies dominate the data-cleaning marketplace [Kimball96], and all three specialize in cleaning large customer address lists. The three companies are Harte-Hanks Data Technologies [Trillium99], Innovative Systems Inc. [Innovative99], and Vality Technology [Vality99]. Recently, companies have started to produce tools and offer data cleaning services that do not address specifically the customer address lists but instead rely on domain specific information provided by the customer: Centrus Merge/Purge Module [QMSoft99], DataCleanser [EDD99], etc. A very good description and design of a framework for assisted data cleansing within the merge/purge problem is available in [Galhardas99].

TDQM is an area of interest both within the research and business communities. The data quality issue and its integration in the entire information business process are tackled from various points of view in the literature (e.g., [Fox95, Fox94, Levitin95, Orr98, Pak93, Redman96, Redman98, Strong97, Svanks84, Wang96]). Other work refers to the same problem as the enterprise data quality management [Flanagan98]. The most comprehensive survey of the research in this area is available in [Wang95].

Unfortunately, none of the mentioned papers refer explicitly to the data cleansing problem. Some of the papers deal strictly with the process management issues from data quality perspective, others with definition of data quality. The later category is of interest to this research. In the proposed model of data life cycles with application to quality [Levitin95] the data acquisition and data usage cycles contain a series of activities: assessment, analysis, adjustment, and discarding of data. Although it is not specified addressed in the paper, if one integrated the data cleansing process with the data life cycles, this series of steps would define it in the proposed model from the data quality perspective. In the same framework of data quality, Fox [Fox94] proposes four quality dimensions of the data: accuracy, current-ness, completeness, and consistency. The correctness of data is defined in terms of these dimensions. Again, a simplistic attempt to define data cleansing process within the framework would be the process that assesses the correctness of data and improve its quality.

More recently, data cleansing is regarded as a first step, or a preprocessing step, in the KDD process [Fayyad96, Brachman96]. Though no precise definition and perspective over the data cleansing process is given. Various KDD and Data Mining systems perform data cleansing activities in a very domain specific fashion. In [Guyon96] discovering of informative patterns is used to perform one kind of data cleansing by discovering *garbage patterns* – meaningless or mislabeled patterns. Machine learning techniques are used to apply the data cleansing process in the written characters classification problem. In [Simoudis95] data cleansing is defined as the process that implements computerized methods of examining databases, detecting missing and incorrect data, and correcting errors. The Recon Data Mining system is used to assist the human expert to identify a series of errors types in financial data systems.

2. Problem Definition

This research is concerned with the following generic framework for automated data cleansing:

- Define and determine errors types
- Search and identify error instances
- Correct the uncovered errors

This paper is focused on the first two aspects of this generic framework for automated data cleansing. The data set currently under investigation is comprised of real world data supplied by the Naval Personnel Research, Studies, and Technology (NPRST). The data set is part of the officer personnel information system including midshipmen and officer candidates. Similar data sets are in use at personnel records division in companies all over the world. The investigated data and its use represent a large problem space.

This particular data set is updated constantly through a large number of different mechanisms (other data bases, data entry, and a variety of personnel, etc.). The entry (and origin) of the data stretches over a very long period of time and there have been few rigid processes in place (i.e., TDQM) for quality assurance. All these factors increase the possibility for errors in the data set. There is a general lack of quality metrics and measurements on the data set and therefore the

overall quality of the data set is unknown. Assessing the overall quality of such a data set, and correcting mistakes is of high priority.

The following are some of the characteristics of the data set and the problem at hand:

- Exploitation of the data set and the data cleansing are performed off-line. Therefore, computation time is not a primary concern, but the data cleansing process should be scalable to larger data sets.
- The size of the data set is medium (45,000 records with 311 fields). The average size of a field is 8 bytes. However, a similar data set is used for sailors and it has around 230,000 records. Although not extremely, large this data set cannot be analyzed in the memory of a regular computer.
- More than half of the elements in the data set are empty.
- Over 50% of the fields are date type, representing events. The rest of the fields represent domain specific attributes, which would vary among different problems.

For the experiments presented here, a part of the original data set, referred from now on simply as the *data set* is utilized. Also, only the date type fields are being examined by the methods presented. Since all these fields represent the same type of data, they can be analyzed without domain knowledge. All the date data elements have more or less the same format, and the range is rather data driven than domain driven. The only issue that needs some domain knowledge is the fact that the first field in the data set now should contain the earliest date for each record. This however does not constrain the problem too much, since most data sets of this type contain a field corresponding to the date of birth, which is the earliest date that appears in an individual personnel record. This field is referred further in the paper as the *reference date*. If there is no such a field in the data set, an extra field can be added to the data set, which would satisfy this criterion.

All the date type values are converted to integer numbers. Since dates have different representations in different application, a tool was designed to recognize and convert twenty-eight types of date representations, grouped in five classes. The only restriction is that the data elements in the same data set should be represented in the formats from the same class. An

integer type is utilized for memory concerns. Thus, the data cleansing tools developed operate on any data set that has date type or integer type elements, with the only restriction that the first field of each record contains the smallest value in that record. The current experiments were done only on the data set containing the dates. This kind of data allows the establishment of some hypothesis on data, as shall be seen further in the paper, that makes the definition and identification of possible errors, without using domain knowledge.

In previous work [Maletic00], several error types were identified and a prototype tool was implemented that automatically uncovered these errors. The implemented tool is designed to identify the following potential errors: non-numerical value in a field, a value that represents a date earlier than the reference date (the data element in the first field), missing value in the reference field, too many empty fields, outlier values in a record, records that do not follow existing (and identified) patterns in the data, outlier records according to clustering. These results attempted to look for potential errors beyond the integrity.

Three major approaches were considered:

1. Identifying outlier fields using statistical values (averages, standard deviation, range), based on Chebyshev's theorem [Barnett94], [Bock98], considering the confidence intervals for each field [Johnson98];
2. Identify outlier records that do not conform to existing pattern in the data. Combined techniques (partitioning and classification) are used to identify patterns that apply to most records.
3. Identify outlier records using clustering based on Euclidian distance. Existing clustering algorithms provide little support for identifying outliers [Knorr97, Murtagh83, Zhang97]. A combined clustering method is utilized along with the group-average clustering algorithm [Yang99] considering the Euclidean distance between records.

The results showed that statistical outlier detection methods could be successfully used to identify errors in a data set. The methods also showed that real-life data is highly uncorrelated and it is hard (often impossible) to identify patterns that apply to most of the records. Finally,

clustering the entire data set using the Euclidian distance proved to be of little help to identify the errors and computationally very expensive.

Among the 5000 records of the experimental data set, 164 were identified to contain outlier values that proved to correspond to erroneous data items. The work presented in this paper extends these results by identify new types of errors using association rules. Under this framework, the goal is to find rules like "the event in field A always happens before (or after, or at the same time as) the event in field B". Once these rules are identified and a certain confidence determined, the data records are identified where the rule does not hold. The particular values in the fields A and B are analyzed and based on the confidence of the rule, these are declared possible errors or not.

3. Association Rules and Data Cleansing

The term *association rule* was first introduced by Agrawal et. al. [Agrawal93] in the context of market basket analysis. In this kind of analysis, the data set is defined as the *basket data* $B = \{b_1, b_2, \ldots, b_n\}$, where each *basket* $b_i \subseteq I$ is a collection of *items*, and where $I = \{i_1, i_2, \ldots, i_k\}$ is a set of k elements. An *association rule* in the database B is defined as follows.

$i_1 \Rightarrow i_2$ is a an *association rule* if:

1. i_1 and i_2 occur together in at least *s%* of the n baskets, where *s* is the *support* of the rule;
2. and, of all baskets containing i_1, at least *c%* contain i_2, where *c* is the *confidence* of the rule.

In fact, Agrawal et al. define association rules between item sets $A \Rightarrow B$, where *A* and *B* are disjoint sets of items instead of single items. In general *A* is referred to as the *antecedent* (or *left-hand side*) of the rule, and *B* as the *consequence* (or *right-hand side*) of the rule. In real-life cases and spoken language terms, an association rule is phrased as: "50% of people who buy diapers, also buy beer, and 20% of all buyers buy diapers." In this case, *diapers* and *beer* are the items, the confidence of the rule is 50%, and the support of the rule is 20%. Association rule of

this type are also referred to in the literature as *classical* or *boolean* association rules. For the purposes of this paper boolean association rules between single items are considered as basis for subsequent definitions and the rules between item sets are considered generalizations.

Since its introduction, the problem of mining association rules from large databases has been subject of numerous studies. These studies cover a broad spectrum of topics including: (a) fast algorithms, partitioning, and sampling; (b) incremental updating and parallel algorithms; (c) mining of generalized and multi-level rules; (d) mining quantitative and ratio-rules; (e) mining of multidimensional rules; (f) mining rules with item constraints; (g) distance-based association rules; (h) composite association rules; (i) association rule-based query languages. The work of Ng et. al. [Ng98] contains a comprehensive list of references related to the above mentioned studies. Of interest to this research are a select few that are referenced and presented in the following sections. The results of these studies typically utilize boolean or categorical data.

In practice, the information in many, if not most, databases is not limited to categorical attributes, but also contains much quantitative data. The problem defined in this paper addresses numerical data. Unfortunately, the definition of categorical association rules does not translate directly to the case of quantitative attributes. It is therefore necessary to provide a definition of association rules for the case of a database containing quantitative attributes. Srikant et. al. [Srikant96] extended the categorical definition to include quantitative data. The basis for their definition is to build categorical events from the quantitative data by considering intervals of the numeric values. Thus, each basic event is either a categorical item or a range of numerical values. Such rules are called *quantitative association rules* [Srikant96]. A more formal definition is given there, and confidence and support of the rule are slightly redefined. An example of such rule is “People who spent on bread between $3-$5, and on milk at the same time between $1-$2, usually spend between $1.5-$2 on butter in the same transaction”.

A stronger set of rules is defined in [Korn98] as *ratio-rules*. A rule under this framework is expressed in the form: “Customers typically spend 1: 2: 5 dollars on bread: milk: butter”. This time the strength of the rule allows multiple applications, including data cleansing and outlier detection. However, the paper does not exploit this idea. It is only mentioned that the power of

ratio-rules to reconstruct data could support the data cleansing process. Eigen system analysis is used to find these rules and induces the strength of the rules as well as a computational overhead.

A series of generalizations of quantitative association rules are defined in [Aumann99]. From the perspective of this paper, it is useful to utilize the formulation of a general association rule as

$$population\text{-}subset \Rightarrow interesting\text{-}behavior$$

A further generalization is made in [Padmanabhan98] where a general form of rules are considered: *body* $\Rightarrow$ *head*, where *body* is a conjunction of atomic conditions of the form *attribute op value*, and *head* is a single atomic condition of the form *attribute op value*, where $op \in \{\leq, =, \geq\}$.

3.1. Ordinal Association Rules

The presented extensions and generalizations of the association rule concept can be used for the identification of possible erroneous data items with certain modifications. These considerations lead to a new extension of the association rule – *ordinal association rules*.

Definition. Let $B = \{b_1, b_2, \ldots, b_n\}$ a *data set*, where each *record* $b_i \subseteq I$ is a collection of *items*, and $I = \{i_1, i_2, \ldots, i_k\}$ is a set of k items. Each item i_i has the same numerical *domain* D ($i_i \in D$) and the following relationships are defined in D: $\leq$ - less or equal, = - equal, $\geq$ - greater or equal (having the standard meaning).

Then $i_1, i_2 \Rightarrow i_1\ op\ i_2$, where $op \in \{\leq, =, \geq\}$, is a an *ordinal association rule* if:

1. i_1 and i_2 occur together in at least *s%* of the n records, where *s* is the *support* of the rule;
2. and, in *c%* of the records where i_1 and i_2 occur together $i_1\ op\ i_2$ is true, and where *c* is the *confidence* of the rule.

This definition extends easily to $J \Rightarrow j_1\ op\ j_2\ op \ldots op\ j_m$, $op \in \{\leq, =, \geq\}$, $m \in \{1, \ldots, k\}$, where $J = \{j_1, j_2, \ldots, j_m\}$ is a set of *m* items, $J \subseteq I$.

Ordinal association rules bear some similarity with the above-mentioned extensions of boolean association rules. However, they are better suited to the problem of identifying possible errors in the type of data sets being analyzed for the following reasons:

- They are easier and faster to compute than quantitative association rules or ratio-rules.
- Although they are weaker than quantitative association rules or ratio-rules, they give very good results in the case of date type data. This kind of data lend naturally to ordinal relationships, rather than to precise ratio-relationship, especially at lower granularity (days in this case).
- Distance-based association rules (over interval data) [Miller97] could be also used in this framework, but it is inherently hard to find the right intervals in the absence of specific domain knowledge, and the method is rather expensive.

The process to identify potential errors in data sets using ordinal association rules is composed of the following steps:

1. Prepare the data
2. Find ordinal rules with a minimum confidence *c*.
3. Identify data items that broke the rules and can be considered outliers (potential errors).

The support of the rules is not relevant and therefore only a support of zero is considered. Future work will investigate user-specified minimum support. However, this will only change the number of initially identified patterns. Since only pairs of items are considered, there can be at most C(M,2) patterns. Where M is the number of attributes (fields) of the data set. Let N be the number of records in the data set.

A prototype tool is implemented that deals with all these steps automatically. The tool has been tested on the data described in the section 1. Following, the architecture of the tool and the results are presented.

3.2. Implementation

C++ was chosen as implementation language. The core of the system is implemented such to assure certain portability between platforms (e.g., Windows, Unix).

A reusable data structure was created and implemented as a C++ class. The purpose of the data structure is to handle date type data in a standardized manner. It accepts as default date format year-month-day. It can work also with the other four date types and various date ranges. Altogether, the structure deals with twenty-eight date formats. The main feature of the data structure is that it converts dates to integers representing number of days passed between a reference date and the given date. This type of conversion is used for each date in the data set, considering the first field as the reference date. Thus, each field is converted to an integer representing the number of days passed between the date in the field and the reference date, which is the first field. For those dates that did not have a specified the month or the day, it is assigned 1 for each (January for the month and 1st for the day). The empty fields get a specific value, indicated by a constant (EMPTY). The fields with non-numeric values get a value specified by another constant (WRONG). Each field that contained the number 0 was considered empty. However, since this represents an inconsistency in the representation of the data it is mentioned as a possible error.

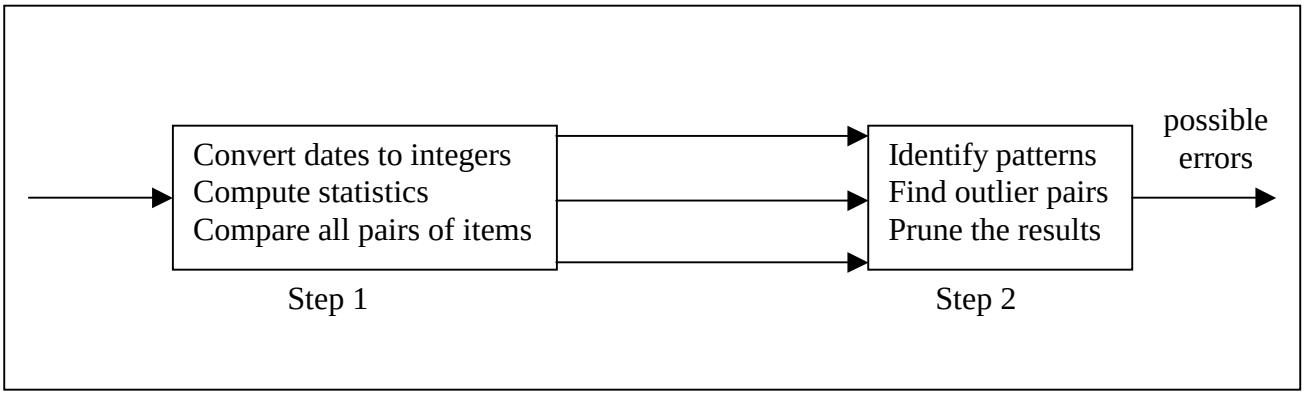

Figure 1: The two main components of the prototype tools. The boxes correspond to each component and show the performed tasks. The arrows indicate the data flow in the system.

The tool has two main components presented in Figure 1. The first component converts the dates from the original data set into integers, and saves them. From that point on all the

computation is done on the new data set that contains the converted dates. A single pass over the data set is necessary. Within this step a series of statistical data is computed for each field (minimum, maximum, mean, standard deviation, number of empty items) and saved. At the same time, each pair of fields is compared for each record and the results are saved in a *comparisons* file. The following data is saved in the file:

- left-hand field number,
- right-hand field number,
- how many comparisons were performed,
- how many times the left-hand field contained a large value than the right-hand field,
- how many times the values were equal, how many times the left-hand field was empty,
- how many times the right-hand field was empty.

An array with the results of the comparisons is maintained in the memory. The array will only have C(M,2) elements, as will have the comparisons file. Figure 2 contains the algorithm for this step. The complexity of this step is only $O(N*M^2)$ where N is the number of records in the data set, and M is the number of fields. Usually M is much smaller than N.

```
Algorithm compare fields.
for each record in the data set (1...N)
        extract and convert the record
        update statistics
        for each l-h field no. (1 .. M-1)
                for each r-h field no. (l-h field no+1 ... M-1)
                        compare the values in l-h field and r-h field
                        update the comparisons array
                end for.
        end for.
        output the record with the converted values
end for.
output the comparisons array in the comparisons file
end algorithm.
```

Figure 2: The algorithm for the first step. Complexity is *O*(N). l-h field and r-h field means left-hand and right-hand field respectively.

In the second step the patterns are identified based on the chosen minimum confidence. There are several researched methods to determine the strength including interestingness and statistical

significance of a rule (minimum support and minimum confidence, chi-square test, etc.). Using confidence intervals [Johnson98] to determine the minimum confidence is currently under investigation. However previous work on the data set [Maletic00] showed that the distribution of the data is not normal. Therefore the minimum confidence was chosen empirically, several values were considered and the algorithm was executed. The results are shown in Table 1.

Confidence	Patterns	Records with errors	Possible errors	High probability errors
95	1581	2747	17842	4130
96	1490	2295	13006	2831
97	1381	1982	9660	1915
98	1236	1210	4975	971
99	1050	522	1820	305
99.3	988	363	1194	216
99.5	942	271	803	127
99.7	876	156	438	57
99.9	783	25	62	6
99.95	760	1	2	0
99.99	757	0	0	0

Table 1: Obtained number of patterns, records with errors, possible errors, and high probability errors, according to the selected confidence

The second component extracts from the comparison file and stores in memory the data associated with the patterns. There are three types of patterns that are identified:

1) value in the left-hand field = value in the right-hand field with confidence *c*;
2) value in the left-hand field ≥ value in the right-hand field with confidence *c*; and
3) value in the left-hand field ≤ value in the right-hand field with confidence *c*.

This is done with a single pass over the comparisons file (complexity $O(C(M,2))$. Then for each record in the data set, each pair of fields that correspond to a pattern it is check to see if the values in those fields within the relationship indicated by the pattern. If they are not, each field is marked as possible error. Of course, in most cases only one of the two values will actually be an error. Once every pair of fields that correspond to a pattern is analyzed, the average number of possible error marks for each marked field is computed. Only those fields that are marked as possible errors more times than the average are finally marked as containing high probability errors. Again, the average value was empirically chosen as threshold to prune the possible errors set. Other methods to find such a threshold, without using domain knowledge, are under investigation. The time complexity of this step is $O(N*C(M,2))$, and the analyzes of each record

is done entirely in the main memory. Figure 3 shows the algorithm used in the implementation of the second component.

```
Algorithm analyze records.
for each record in the data set (1…N)
        for each pattern in the pattern array (1…C(M,2) maximum)
                determine pattern type and pairs
                compare item pairs
                if pattern NOT holds
                        then mark each item as possible error
        end for.
        compute average number of marks
        select and output only the high probability marked errors
end for.
end algorithm.
```

Figure 3: Algorithm for the second step. Time complexity *O*(N).

The results are stored in a tabular text file (record no. x field no.), so that for each record and field where high probability errors were identified, the number of marks is shown.

4. Experiments

Using a 98% confidence, 971 high probability errors were identified out of 5000 records. These were compared with those outliers obtained in previous mentioned work [Maletic00]. These possible errors not only matched many of the previously discovered ones, but 173 were errors unidentified by the previous methods. The distribution of the data influenced dramatically the error identification of the data process in the previous utilized methods. This new method is not influenced as much by the distribution of the data and is proving to be more robust.

Table 2 shows an error identified by ordinal association rules and missed with the previous methods. Here two patterns were identified with confidence higher than 98%: values in field 4 ≤ values in field 14, and values in field 4 ≤ values in field 15. In the record no. 199, both fields 14 and 15 were marked as high probability errors. Both values are in fact minimum values for their respective fields. The one in field 15 was identified previously as outlier, but the one in field 14

was not, because of the high value of the standard deviation for that field. It is obvious, even without consulting a domain expert, that both values are in fact wrong. The correct values (identified later) are 800704. Other values that did not lie at the edge of the distributions were identified as errors as well.

Record no.	Field 1	…	Field 4	…	Field 14	Field 15	…
				. .			
199	600603	…	780709	…	700804	700804	…
				. .			

Table 2: A part of the data set. An error was identified in record 199, field 14, which was not identified previously.

5. Conclusions and Future Work

Association rule mining proves to be useful in identifying not only interesting patterns for fields such as market basket analysis or census data, but also patterns that uncover errors in other kind data sets. The classical notion of association rules has been extended to include ordinal relationships between pairs of numerical attributes, thus defining ordinal association rules. This extension allows the uncovering of stronger rules that yielded potential errors in the data set, while keeping the computation simple and efficient. These results are currently under detailed investigation by domain experts for the data set. This research address two important steps in defining and building a framework for automated data cleansing namely defining error types and automatically identify possible errors.

Future steps in the research will address the merger of the two prototype implementations (i.e., the one presented here and one described in [Maletic00]) and incorporating other new and/or existing methods to identify other error types. A generalization of the proposed association rules on sets of data items, rather than on pairs of items is currently under investigation. Quantitative

association rules will be also considered. Depending on the data, the appropriate type of association rules will be mined. The method needs to be tested on other types of data.

Lastly, but not least, the next step of the data cleansing process will be tackled, that is, correcting the errors. The goal here is to provide a strong assistant to the human expert and minimize the work of the expert to clean the data. Data driven (e.g., ratio-rules) and knowledge-based methods are to be taken in consideration as underlying methods to address these issues.

6. References

[Agrawal93] Agrawal, Rakesh; Imielinski, Tomasz; Swami, Arun: "Mining Association rules between Sets of Items in Large Databases", *Proceedings of ACM SIGMOD International Conference on Management of Data*, Washington, May 1993, pp. 207-216

[Aumann99] Aumann, Y.; Lindell, Y.: "A Statistical Theory for Quantitative Association Rules", *Proceedings KDD99*, San Diego, CA, 1999, pp. 261 - 270

[Ballou89] Ballou, D.; Tayi, K.: "Methodology for Allocating Resources for Data Quality Enhancement", *Communications of the ACM*, Vol. 32, No. 3, 1989, pp. 320-329

[Barnett94] Barnett, V.; Lewis, T.: *Outliers in Statistical Data*, John Wiley and Sons, 1994

[Bock98] Bock, R.K.; Krischer, W.: "The Data Analysis Briefbook", Springer, 1998

[Brachman96] Brachman, R. J.; Anand, T.: "The Process of Knowledge Discovery in Databases: A Human-Centered Approach", in *Advances in Knowledge Discovery and Data Mining, MIT Press/AAAI Press*, 1996, pp. 97-58

[EDD99] EDD. Home page of DataCleanser tool: http://www.npsa.com/edd/, last accessed 01/15/2000

[English99] English, J.: Column "Plain English on Data Quality", *DM Review*: http://www.dmreview.com, last accessed 02/10/99

[ETI99] E. T. International. Home page of ETI-Data Cleanse tool. http://www.evtech.com/products/dc2.html, last accessed 01/15/2000

[Fayyad96] Fayyad, U.; Piatetsky-Shapiro, G.; Smyth, P.: "From Data Mining to Knowledge Discovery: An Overview", in *Advances in Knowledge Discovery and Data Mining, MIT Press/AAAI Press* 1996, pp. 1-36

[Hernandez97] Hernandez, M. A.; Stolfo, J. S.: "Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem", *Journal of Data Mining and Knowledge Discovery*, No. 2, 1998, pp. 9-37

[Flanagan98] Flanagan, T.; Safdie, E. (Eds.): "A Practical Guide to Achieving Enterprise Data Quality", http://www.techguide.com/ last accessed, 12/01/99

[Fox95] Fox, C.; Levitin, A.; Redman, T., "Data and Data Quality", in *Encyclopedia of Library and Information Science*, 1995

[Fox94] Fox, C.; Levitin, A.; Redman, T.: "The notion of Data and Its Quality Dimensions", *Information Processing and Management*, Vol. 30, No. 1, 1994, pp. 9-19

[Galhardas99] Galhardas, H.; Florescu, D.; Shasha, D.; Simon, E.: "An Extensible Framework for Data Cleaning", *Technical Report, Institute National de Recherche en Informatique ét en Automatique*, 1999

[Guyon96] Guyon, I.; Matic, N.; Vapnik, V.: "Discovering Information Patterns and Data Cleaning", in *Advances in Knowledge Discovery and Data Mining, MIT Press/AAAI Press* 1996, pp. 181-203

[Johnson98] Johnson, R. A.; Wichern, D. W: "Applied Multivariate Statistical Analysis – Fourth Edition", Prentice Hall, 1998

[Kimball96] Kimball, R.: "Dealing with Dirty Data", *DBMS*, Vol. 9, No. 10, September 1996, p. 55

[Korn98] Korn, Flip; Labrinidis, Alexandros; Yannis, Kotidis; Faloustsos, Chrostos: "Ratio Rules: A New Paradigm for Fast, Quantifiable Data Mining", *Proceedings of the 24th VLDB Conference*, New York, 1998, pp. 582--593

[Knorr97] Knorr, E. M.; Ng, R. T.: "A Unified Notion of Outliers: Properties and Computation", *Proceedings of KDD 97*, pp. 219-222

[Levitin95] Levitin, A.; Redman, T.: "A Model of the data (life) cycles with application to quality", in *Information and Software Technology*, Vol. 35, No 4, 1995, pp. 217-223

[Maletic00] Maletic, J.; Marcus, A.: “Automated Identification of Errors in Data Sets“, *TR-013-2000*, University of Memphis

[Miller97] Miller, R. J.; Yang, Y.: “Association Rules over Interval Data”, *ACM SIGMOD*, 26(2), May, 1997, pp. 452-461

[Moss98] Moss, L.: "Data Cleansing: A Dichotomy of Data Warehousing", *DM Review*, February 1998

[Murtagh83] Murtagh F., “A Survey of Recent Advances in Hierarchical Clustering Algorithms”, *The Computer Journal*, Vol. 26, No. 4, 1983, pp. 354-359

[Ng98] Ng, Raymond T.; Lakshmanan, Laks V. S.; Han, Jiawei; Pang, Alex: "Exploratory Mining and Pruning Optimizations of Constrained Association Rules", *Proceedings ACM SIGMOD*, Seattle, Washington, June 1998, pp. 13-24

[Orr98] Orr, K.: "Data Quality and Systems Theory" *Communications of the ACM*, Vol. 41, No. 2, February 1998, pp. 66-71

[Padmanabhan98] Padmanabhan, Balaji; Tuzhilin, Alexander: "A Belief-Driven Method for Discovering Unexpected Rules", *Proceedings KDD 98*, pp. 94-100

[Pak93] Pak, S.; Pando, A.: "Data Quality Analyzer: A Software Tool for Analyzing Data Quality in Data Manufacturing Systems", on-line: http://web.mit.edu/tdqm/papers/93/pass1/93-10.html, 1993, last accessed 02/10/99

[QMSoft99] Qualitative Marketing Software. Home page of Centrus Merge/Purge Module: http://www.qmsoft.com/solutions/Merge.htm last accessed 01/15/2000

[Redman98] Redman, T.: "The Impact of Poor Data Quality on the Typical Enterprise", *Communications of the ACM*, Vol.41, No.2, February 1998, pp. 79-82.

[Redman96] Redman, T.: *Data Quality for the Information Age*, Artech House, 1996

[Simoudis95] Simoudis, E.; Livezey, B.; Kerber, R.: "Using Recon for Data Cleaning", in *Proceedings KDD 1995*, pp. 282-287

[Srikant96] Srikant, Ramakrishnan; Vu, Quoc; Agrawal, Rakesh: "Mining Association Rules with Item Constraints", *Proceedings ACM SIGMOD International Conference on Management of Data*, Montreal, Canada, June 1996, pp. 1-12

[Strong97] Strong, D.; Yang, L.; Wang, R.: “Data Quality in Context”, *Communications of the ACM*, May 1997, Vol. 40, No. 5, pp. 103-110

[Svanks84] Svanks, M.: "Integrity Analysis: Methods for Automating Data Quality Assurance", *EDP Auditors Foundation*, Vol. 30, No. 10, 1984, pp. 595-605

[Trillium99] Trillium Software System for data warehousing and ERP: http://www.trilliumsoft.com/products.htm last accessed 01/15/2000

[Vality99] Vality Technology. Home page of the INTEGRITY Data Re-engineering Environment: http://www.vality.com/ last accessed 01/15/2000

[Wang93] Wang, R.; Reddy, M., P.; Gupta, A.: "An Object-Oriented Implementation of Quality Data Products", in Proceedings WITS, 1993

[Wang95] Wang, R.; Storey, V.; Firth, C.: "A Framework for Analysis of Data Quality Research", *IEEE Trans On Knowledge and Data Engineering*, Vol. 7, No. 4, August 1995, pp. 623-639

[Wang96] Wang, R.; Strong, D.; Guarascio, L.: "Beyond Accuracy: What Data Quality Means to Data Consumers", in *Journal of Management Information Systems*, Vol. 12, No 4, Spring 1996, pp. 5-34

[Yang99] Yang, Y.; Carbonell, J.; Brown, R. and Pierce, T. and Archibald, B. T. and Liu, X.: “Learning approaches for Detecting and Tracking News Events”, *IEEE Intelligent Systems*, Vol. 14, No. 4, July/August 1999

[Zhang97] Zhang, T.; Ramakrishnan, R.; Livny, M.: “BIRCH: A New Data Clustering Algorithm and Its Applications”, *Data Mining and Knowledge Discovery*, no. 1, 1997, pp. 141-182.