Critical Evaluation of Validation Rules Automated Extraction from Data

David Pejcoch
Department of Information and Knowledge Engineering
University of Economics, Prague
david@pejcoch.com

Abstract: The goal of this article is to critically evaluate a possibility of automatic extraction of such kind of rules which could be later used within a Data Quality Management process for validation of records newly incoming to Information System. For practical demonstration the 4FT-Miner procedure implemented in LISpMiner System was chosen. A motivation for this task is the potential simplification of projects focused on Data Quality Management. Initially, this article is going to critically evaluate a possibility of fully automated extraction with the aim to identify strengths and weaknesses of this approach in comparison to its alternative, when at least some a priori knowledge is available. As a result of practical implementation, this article provides design of recommended process which would be used as a guideline for future projects. Also the question of how to store and maintain extracted rules and how to integrate them with existing tools supporting Data Quality Management is discussed.

Key words: Data Quality, Data Validation, validation rules, Canonical Data Model, 4FT-Miner, GUHA, Drools, business rules

1. Introduction

An extraction of Association Rules could be considered as one of the most frequently used approaches within a process of Knowledge Discovery from Databases (KDD). It’s true especially in context of identifying the relations among miscellaneous items purchased by single customer. The aim of this article is to focus on application of these rules within Data Quality Management (DQM) area to verify newly incoming records. The main purpose is to simplify and accelerate this process. The minor goal of this article is to provide a summarization of current approaches to data validation (especially those which use rules) and to provide a classification of validation rules.

First, it’s necessary to provide brief introduction to the logic of Data Quality Management process and basic terms must be defined. Then the role of validation within this process will be accented and different supporting methods will be introduced. At the end of first section a brief introduction to theoretical background of rules theory and overview of different validation rules classifications will be described. In analytical section, initially a description of a methodology and used data sets will be provided. Then results of application of 4FT-Miner procedure on respective data sets will be commented. After summarization of results, recommended process of implementation and maintenance will be introduced. At the end of analytical section, the example of how to integrate extracted rules with current tools and decision engines will be mentioned.

From the perspective of data validation, the approach evaluated in this article represents a relatively unique method. Although a possibility of automatic extraction of validation rules has been introduced several times e.g., in (Pirkl, 2004), deep dive into this problematic and its real evaluation has never been published. A thesis of (Morisaki et al., 2007) may attract us with promising title mentioning “Defect Data Analysis Based on the Mining of Extended Association Rules”. However, at the end, their approach aims just on defects in software development and looks for associations among attributes of such defects and their correction effort. Although later their method could be theoretically used also in Data Quality Management to determine costs of observed defects, original publication is not directly connected to data validation topic.

(Paladi and Arts, 2011) in their work describe automatic extraction of constraints. However, these constraints are considered only from database model perspective and for their extraction they have been used just a method of reverse engineering. Although their approach would be really useful for data validation in the world of relation databases, my work is focused on more general validation rules.
The main contribution of this work is to provide a proof of concept for practical application of Association Rules method to extraction of such kind of rules which could be further used for validation of newly incoming tuples (in other terminology: records, observations, etc.).

1.1 Data Quality Management

Data Quality Management is usually considered as a set of activities to reach and maintain such level of predefined data characteristics which enable stakeholders to use data for their intended usage. This definition corresponds to the earlier mentioned by M. Juran (well-known guru of quality management) who considers quality data as those which are "appropriate for their intended use in operational activities, decision making and planning" (Juran and Godfrey, 2010). Direct relations between fitness for use and data characteristics could be found e.g., in (Redman, 2001). The classification of typical characteristics is introduced in Tab. 1 below as a result of compiling the approaches published in (Redman, 2001), (Kral and Zemlicka, 2006), (Battini and Scannapieco, 2006), (Pipino et al., 2002), (Lee et al., 2006), (McGilvray, 2008) and (Vorisek, 2008).

1.2 The Role of Validation within Data Quality Management Process

Data Validation is one of the typical tasks within a Data Quality Management Process (DQM). Validation methods are usually defined on tactical level of DQM and implemented and executed on its operational level. Supporting functionality for validation is at least somehow included in all DQM tools. Another topic “Verification” is usually considered as a synonym of Validation.

(Olson, 2003) defines valid values as being “in the collection of possible accurate values and represented in an unambiguous and consistent way”. The similar approach we can see in (English, 1999). Validation is mentioned in description of Process P2 (Assess Information Quality) of his TQdM (Total Quality data Management) methodology as a step 5: Identify Data Validation Sources for Accuracy Assessment. These validation sources consist from different internal and external reports, official documents, forms or demographic data. In another part of his book (English, 1999) mentions “validity or business conformance” as a “measure of the degree of conformance of data to its domain values and business rules”. (McGilvray, 2008) explicitly mentions Validation and Verification as synonyms. She defines Validation as a confirmation of validity, however in another part of her book she mentions the same term in context of functionality provided by Data Cleansing tools with the meaning of Data Augmentation and Data Enhancing (adding missing values, correct / update incorrect values).

My own definition of Data Validation considers it as: “a procedure of assurance whether a value of respective column meets the criteria defined for respective data characteristic”. In my approach I strictly distinguish between Validation as a controlling step and Data Cleansing as one particular way how to improve identified discrepancies. Common mistake is to consider Data Validation just as a part of data preparation step within a process of Knowledge Discovery from Databases. Nevertheless, this topic has significantly wider meaning in context of all data sources within a Data Universe (consider as a set of all data sources processed by respective subject) and all processes which create, transform, store and use these data.

Based on the analysis of functionality of selected DQM tools (e.g., SAS DataFlux, Talend Open Studio for MDM) and my own previous experience from projects in Insurance, Bank and e-Business verticals I identified below mentioned list of different categories of methods:

- Control for a range of values;
- Lookup for single value into some list of values or external registry (using exact or similar key);
- Merge with an external registry (using exact or similar key);
- Syntactical control using regular expressions;
- Checksum digit (e.g., modulo control);
- Validation of reference integrity (e.g., conformance of primary and foreign keys);
- Different kinds of validation rules (also called business or production rules);
- Validation by running the process (e.g., address could be considered as correct when correspondence is usually successfully delivered to its recipient);
- Security of information systems is usually tested using penetration tests that simulate hackers’ attacks;
- Cost aspects of data quality could be validated against statements of Finance controlling.
Tab. 1: Mapping Between Data Characteristics and Methods of Validation

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Data Characteristic</th>
<th>Description</th>
<th>Method of Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intrinsic Dimension</td>
<td>Reliability (Credibility)</td>
<td>Percentage of general acceptance of data by their users</td>
<td>Business rules</td>
</tr>
<tr>
<td></td>
<td>Uniqueness</td>
<td>1 - percentage of not-intended duplicities in data sources</td>
<td>Descriptive statistics, business rules</td>
</tr>
<tr>
<td></td>
<td>Semantic Accuracy</td>
<td>Percentage of values which correspond with examples from respective ontology / vocabulary</td>
<td>Range, lookup / merge, checksum</td>
</tr>
<tr>
<td></td>
<td>Syntactic Accuracy</td>
<td>Percentage of values with correct syntax</td>
<td>Regular expressions</td>
</tr>
<tr>
<td></td>
<td>Precision</td>
<td>Percentage of values which correspond with real entity</td>
<td>Confrontation with real entity or lookup / merge to reference source</td>
</tr>
<tr>
<td>Time Dimension</td>
<td>Timeliness</td>
<td>Percentage of actual values</td>
<td>Business rules</td>
</tr>
<tr>
<td></td>
<td>Volatility</td>
<td>Period for which data stays valid in real world</td>
<td>Business rules</td>
</tr>
<tr>
<td>Context Dimension</td>
<td>Internal Consistency</td>
<td>Percentage of consistent values within single record (e.g., row of data table)</td>
<td>Business rules</td>
</tr>
<tr>
<td></td>
<td>External Consistency</td>
<td>Percentage of consistent values across different data sources</td>
<td>Reference integrity, business rules</td>
</tr>
<tr>
<td></td>
<td>Completeness</td>
<td>Percentage of intended complete values</td>
<td>Descriptive statistics, business rules</td>
</tr>
<tr>
<td>Dimension of Usage</td>
<td>Availability</td>
<td>Percentage of time period when data have been available for their intended users</td>
<td>Validation by user (interview, qualitative research)</td>
</tr>
<tr>
<td></td>
<td>Comprehensibility</td>
<td>Percentage of data comprehensiveness to their users</td>
<td>Validation by user (interview, qualitative research)</td>
</tr>
<tr>
<td></td>
<td>Interoperability</td>
<td>Percentage in which data are documented and have assigned metadata</td>
<td>Validation by user (interview, qualitative research)</td>
</tr>
<tr>
<td></td>
<td>Security</td>
<td>Indexed (Y/Y, M/M) number of security incidents</td>
<td>Penetration tests</td>
</tr>
<tr>
<td>Economical Dimension</td>
<td>Costs (absolute or maintenance)</td>
<td>Y/Y index of data acquisition and maintenance</td>
<td>Financial controlling</td>
</tr>
<tr>
<td></td>
<td>Costs (absolute or availability)</td>
<td>Y/Y index of data storage, archiving and availability</td>
<td>Financial controlling</td>
</tr>
<tr>
<td></td>
<td>Costs (absolute or availability)</td>
<td>Y/Y index of data security</td>
<td>Financial controlling</td>
</tr>
</tbody>
</table>

Source: Derived from approaches to data characteristics published in (Redman, 2001), (Kral and Zemlicka, 2006), (Battini and Scannapieco, 2006), (Pipino et al., 2002), (Lee et al., 2006), (McGilvray, 2008) and (Vorishek, 2008)

Mapping between different validation techniques and key data characteristics is described in Tab. 1. This mapping represents also my own contribution to this problematic. From table above it is evident
that so called business rules represent relatively often method of validation. From perspective of practical realization they are mostly represented as IF THEN statements or SQL LIKE clauses. Origin of their popularity we can find e.g., in (Lošin, 2002) who mentions as their advantages:

- Speed of implementation;
- Reuse in multiple contexts;
- Ease of modification;
- Persistence of Business Knowledge;
- Ease of re-engineering (rules are often something like documentation of implemented functionality).

1.3 General Classification of Rules

According to (Berka, 1999) we can distinguish between Association Rules looking for interesting relations among values of different attributes (and their combinations) and decision rules. While Association Rules have typically syntax \( \text{Ant} \Rightarrow \text{Suc} \) (where \( \text{Ant} \) is called Antecedent and \( \text{Suc} \) is called Succeedent), decision rules have generally form of IF \( \text{Ant} \) THEN \( \text{Class} \). From the perspective of practical data validation the rules having \( \text{Ant} \Rightarrow \text{Suc} \) syntax are most frequent. In case of using Decision Rules for validation, the \( \text{Class} \) would have the meaning of binary indicator whether the result of control is valid or not.

Alternatively, we can also classify rules as exact or probabilistic. Second concept is mentioned e.g., in (Goodman & Smyth, 1989). Introduced rules have syntax IF \( \text{Ant} \) THEN \( \text{Suc} \) \( (p) \), where \( p \) is conditioned probability \( P(\text{Suc}|\text{Ant}) \).

From the perspective of how exactly are \( \text{Ant} \) and \( \text{Suc} \) defined we can distinguish between crisp and fuzzy rules. Fuzzy rules use so called fuzzy sets represented by membership function instead of exact specifying the subsets of values, intervals, etc.

(Ras and Wieczorkowska, 2000) also define so called Action Rules describing relations between sets of attributes (stable attributes or potentially modified by users) and decisions. Action Rules topic is also sometimes mentioned in connection to Data Validation as a part of rules which is responsible for initialization of some workflow when condition of rule wasn’t valid. Practical application of this kind of validation rules is described e.g., in (Berestizhevsky and Kolosova, 2000). Their article is focused on interactive validation within “SAS-based date-entry” applications. Authors consider action rules which could be characterized as the Rule (1).

\[
\text{IF } A > a \text{ AND } B < b \Rightarrow \_\text{MSG}_\text{}=\text{"Alarm"}. \tag{1}
\]

In this example if attribute \( A \) exceeds some pre-defined threshold and attribute \( B \) stays under some limit, the process will send alarm message. These rules could be generally represented in syntax described by the Rule (2).

\[
\text{Ant} \wedge \sim\text{Suc} \Rightarrow \text{Action} \tag{2}
\]

In batch processing “Alarm” concept could be changed to general escalation process. (Shiffman, 1997) calls this kind of rules “condition-action rules”. However, we can also find some its domain specific synonyms. E.g., (Medlock et al., 2011) is talking about “clinical rules”.

1.4 Classification of Validation Rules

(Olson, 2003) considers two groups of Business Rules: (1) Data Rules and (2) Process Rules which are in his approach explained as Data Rules that “operate within an application”. Typical example of Process Rules is calculation of derived value (e.g., customer segment). Data Rules (Olson, 2003) describes as “specific statements that define conditions that should be true all of the time”.

Data Rules represent always a relation among multiple values. From this perspective the original definition of Data Rules could be modified to the form of “expression of a condition involving multiple values that must be true over a collection of data for that data to be considered accurate”.

The classification of Data Rules published in (Olson, 2003) has two dimensions. From one perspective he distinguishes among rules based on (1) single column, (2) multiple columns and (3) columns that cross over multiple tables. From the second perspective he considers rules based on single row (record) or multiple rows.
From the perspective of impact which has violation of these rules (Olson, 2003) defines hard rules (which cannot be violated) and soft rules for which could exist some exceptions. These undefined exceptions mostly represent weakness of data model. However, in the analytical part of this article, most of extracted rules will be classified as "soft".

(Lee et al., 2006) considers data integrity rules which follow the definition of Codd's integrity constraints. He distinguishes among: (1) entity integrity (primary key is not null), (2) referential integrity (primary key is equal to foreign key), (3) domain integrity (all columns in a database must be declared upon a defined domain) and (4) column integrity (predefined set of values).

From the perspective of Data Rules sources (Olson, 2003) considers: (1) extraction from source code of implemented application, (2) extraction from stored procedures within RDBMS (Relation Database Management Systems), (3) derivation of rules from business procedures and (4) rules retrieval from data and business expert.

2. Critical Evaluation of Automated Rules Extraction

2.1 Why to Extract Rules Automatically?

Traditional way how to design validation rules is their definition by domain expert. As mentioned in the introduction of this article, the concept of automated extraction of validation rules has been outlined in several publications e.g., in (Pirkl, 2004). However, it never evolved to some real evaluation. Automated Extraction of Rules for purpose of future validation of Neural Networks is mentioned in (Taylor and Darrah, 2005). However, it is just confrontation of KDD model with another mined knowledge (basically with a different model). There seems not to be any known article or conference proceeding explicitly describing the extraction of validation rules from data for purpose of global Data Quality Management.

Some kind of automated rules extraction seems to be trivial. For example the pulling subsets of expected values or constraints such as actual min and max values using SQL (Structured Query Language). The aim of this article is to evaluate a possibility of extracting the more complex Association Rules.

2.2 Method Used for Evaluation

The hypothesis behind expects that so called "business" or "production" rules represent knowledge in a similar way to those rules extracted from data in typical KDD (Knowledge Discovery in Databases) task. For the practical evaluation of this hypothesis I used so called 4FT-Miner ASSOC procedure, an implementation of GUHA method in LISp-Miner System, developed at the Department of Information and Knowledge Engineering at the University of Economics in Prague.

A concept of Association Rules has been popularized especially by (Agrawal et al., 1993). Generally it is based on expressing the relations (associations or correlations) among attributes within large data set. (Agrawal et al., 1993) provided the most used form of these rules in form of implications between Antecedents and Consequents. In Agrawal's concept they are represented by items from market basket. This form also corresponds with commonly used syntax of validation rules. The main reason why I decided not to use this original method is its assumption to have binarized values of source attributes as an input. This would bring more unnecessary effort to process of rules extraction. My goal was to find a method which is able to simplify this extraction as much as it is possible.

Theoretical concept of 4ft-Miner procedure has been introduced in (Rauch and Simunek, 2000). It is based on generation and verification of hypothesis with syntax

\[ \varphi \equiv \psi, \]  

where \( \varphi \) and \( \psi \) are Boolean attributes, \( \varphi \) is called Antecedent, \( \psi \) is called Succedent and \( \equiv \) represents 4ft-quantifier. Optionally, conditioned Succedent could be considered. LISp-Miner System provides different kinds of quantifiers. However, from the perspective of validation rules, the most suitable seems to be Founded Implication, defined by condition

\[ a/(a+b)\geq p \land a\geq B \]  

where 0<\( p \leq 1 \), B>0 and \( p \) (Confidence) is derived from so called Four-fold table (see Tab. 2). Condition above could be interpreted as that at least 100\( p \) of objects must satisfy both \( \varphi \) and \( \psi \) (a Confidence
must be at least 100p) and the number of these objects a must be at least equal to B (so called Support).

Table 2: Four-fold table

<table>
<thead>
<tr>
<th></th>
<th>ψ</th>
<th>¬ψ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Φ</td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>¬Φ</td>
<td>c</td>
<td>d</td>
</tr>
</tbody>
</table>

Source: (Rauch and Simunek, 2000)

In Four-fold table, a means the number of objects satisfying both φ and ψ. b means the number of objects satisfying φ but not ψ. c represents the number of objects satisfying ψ but not φ and finally d represents the number of objects which satisfy neither φ nor ψ.

2.3 Methodology of Evaluation

Apart from two already mentioned criteria of model quality (Confidence and Support), refined rules will be evaluated from perspective of their consistency (using confrontation with background knowledge) and their potential usability.

In experiments described below, generally only conjunction between different parts (so called literals) of Antecedent and Succeedent is considered. Applied procedure was set up to find only positive literals (not their negations) and also only single value (category, created interval) in literal is considered.

Described experiments have been realized on server with instance of OS MS Server 2008, with 30 GB RAM and 4 virtual CPUs.

2.4 Description of Data Sets

In next paragraphs I’m going to provide brief description of data sets used in experiments. Four different data sets have been used for automated extraction of validation rules. Three of them are publicly available. The last one has been derived from different levels of product information describing items listed during single week on ebay.com, ebay.it, ebay.fr, ebay.es sites, in combination with data from already implemented product catalogue as a part of eBay Int. information system. Although this kind of information is generally available on sites, according internal eBay policy it is not possible for me as the company employee to publish source data.

The first data set describes different attributes of vehicles. It is one of commonly available data sets within UCI Machine Learning Repository, known as MPG data (UCI, 1993). Originally it was taken from StatLib library maintained at Carnegie Mellon University. In the 1983 it was used in American Statistical Association Exposition and later in 1993 it was analysed by R. Quinlan in his proceeding on the Tenth International Conference of Machine Learning. The specific characteristic of this data set is the fact that each row represents a unique combination of values with Car Name as natural primary key. Car Name attribute contains from string concatenated from vehicle mark (brand) and type. Using pattern analysis,four different tokens (sub strings) delimited by space have been identified within this string. These tokens have been later parsed and inserted into separate attributes Car Name #1 – Car Name #4. Character of this data set predetermines it to be used for a demonstration of extracting the data validation rules from so called reference sources (data sources used as an etalon). Other attributes in data set are miles per gallon (Mpg), Number of Cylinders, Displacement, Horsepower, Weight, Acceleration, Model Year and Origin.

The second data set is publicly available on website of Czech Ministry of Internal Affairs and it represents statistics of registered motor vehicles (Car Brand, Car Model, Motor, Displacement, Fuel, and Year of Registration). Original data set contained summarized numbers of vehicles within respective years, split by different dimensions. Although this kind of data source should be also considered as a reference source, its poor quality doesn’t enable us to do it. For purpose of rules extraction I transformed original data to such kind of source which simulates real population of vehicles. Resulting data set then could be considered as a part of database of insurance company providing motor vehicle insurance services. Only records with Year of Registration >= 1997 have been considered in experiments below.
The third data set has been downloaded from web pages of Czech Postal Office and it represents a list of all postal codes with assigned regional units (counties, cities, villages and parts of cities). Using this data set I was going to demonstrate an extraction of rules which are able to bring some level of volatility to validation process. In real world application we can simply imagine that for some postal codes some their part is not important for derivation of the city name. This part of experiment focused on automated extraction of such kind of rules.

Last data set represents a domain of online business. Source attributes contain information about EAN (European Article Number), MPC (Manufacturer Product Code), UPC (Unique Product Code), Brand, Model and Colour. EAN represents something like birth code of items listed on e-shop. UPC is its equivalent in US. The goal was to extract such kind of rules which would be able to verify the internal consistency of these attributes and identify such records which have been incorrectly inserted by seller or which have been wrongly processed during their automated derivation from unstructured item title.

2.5 Extracted Rules

In the first experiment, “blind” extraction of rules from MPG data set, without any additional domain knowledge, has been used. Also no transformations have been considered as a part of data preparation, so Car Name attribute has been kept unparsed. All columns have been considered as potential part of Antecedent, Succedent and Condition as well. This had to simulate situation when extraction of validation rules is provided absolutely automatically. Parameter of Support $a$ has been fixed at the level of 0.1. The Confidence $p$ has been decreased step by step from 1 to 0.95. Results of this experiment are summarized in Tab. 3 in column marked by “With Condition Defined” label.

<table>
<thead>
<tr>
<th>$p$</th>
<th>$a$</th>
<th>Number of Hypotheses</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>With Condition Defined</td>
</tr>
<tr>
<td>0.999</td>
<td>0.1</td>
<td>15</td>
</tr>
<tr>
<td>0.995</td>
<td>0.1</td>
<td>15</td>
</tr>
<tr>
<td>0.990</td>
<td>0.1</td>
<td>15</td>
</tr>
<tr>
<td>0.950</td>
<td>0.1</td>
<td>42</td>
</tr>
</tbody>
</table>

Since a lot of extracted rules were just redundant (with the same values of respective attribute either in Condition or Succedent), in second experiment I decided to use only Antecedent and Succedent as a part of model. Results of this experiment are described in Tab. 3, in “Without Condition” column. Examples of extracted rules are (5) and (6) below. The string “$\div <$” represents Founded Implication operand.

\[\text{Cylinders} \ (8) \ & \ \text{Mpg} \ (9;19>) > \div < \text{Origin} \ (1) \]  
\[\text{Displacement} \ (88;98>) \ & \ \text{Horsepower} \ (66;86>) > \div < \text{Cylinders} \ (4) \]  

As it was already mentioned, the original Car Name attribute consists from four different tokens. Using first of them (with the meaning of vehicle mark / brand), the number of hypothesis with Confidence $p \geq 0.99$ increased to 7. Newly identified knowledge described dependency between some marks and country of origin.

\[\text{Car Name} \ #1(\text{CHEVROLET}) > \div < \text{Car Origin} \ (1) \]  
\[\text{Car Name} \ #1(\text{FORD}) > \div < \text{Car Origin} \ (1) \]  

Rules (7) and (8) above refer to US as a country of origin for brands Chevrolet and Ford, with Confidence $p=1$. 

---

38 JOURNAL OF SYSTEMS INTEGRATION 2014/4
Unfortunately none one of extracted hypotheses had Confidence $p = 1$. That means, in real application, for this kind of rules it should be necessary to specify which process must be followed in case of their FALSE result. In terminology of (Olson, 2003), these rules are considered as „soft rules‟. Examples of top three rules related to Displacement Interval are listed below.

\[ \text{Displacement Interval (1000;1350>) & Car Brand (ŠKODA) >÷< Fuel (BA)} \] (9)

\[ \text{Displacement Interval (1000;1350>) & Year of Registration (1997) >÷< Fuel (BA)} \] (10)

\[ \text{Displacement Interval (1350;1400>) & Car Brand (ŠKODA) >÷< Fuel (BA)} \] (11)

E.g., the rule (10) covers model “Felicia” and the beginning of model “Octavia A4”, both with lower displacement, which really corresponds to gasoline motors. Generally, higher number of the most confident rules when using Displacement Interval instead of Displacement proves the importance of domain knowledge for a process of rules extraction. Definition of Displacement Interval by the expert also significantly changed the character of extracted rules.

Examples of rules extracted from the list of Czech postal codes are shown in Tab. 5. The first one represents too general rule. It indicates that postal code “10?00” (where “?” represents whichever number) leads to “Prague” (Praha) as a city, with Confidence $p = 1$. Unfortunately, for Prague it is practically impossible to get more concrete rules. E.g., “10600” leads not just to “Prague 10” but also to “Prague 15” and “Prague 4”. Using RUIAN (Registry of Geo Identification, Addresses and Properties) as an additional reference source and considering the frequency of real address points for respective postal code as votes, we can get the rule which with the same Antecedent leads to “Prague 10” but only with Confidence $p = 0.77$. However, using this additional data source, also some additional expert made standardization of Part of City attribute would be necessary. The rule with Confidence $p = 0.998$ is possible to extract e.g., for Postal Code “13???”; since only a small number of addresses comes from another parts of the city.

The rule (13) indicates that for “Liberec” the last position of Postal Code is not important, as well as for “Krupka” in the rule (14). Rules (15) and (16), which are related to “Rokytnice nad Jizerou”, indicate the necessity of their future combination. It is obvious that the rule (15) is more general, so the rule (16) could be neglected. Especially in situation when its Confidence $p$ is lesser than 1.

**Tab. 4: Registry of Vehicles**

<table>
<thead>
<tr>
<th>$p$</th>
<th>$a$</th>
<th>Number of Hypotheses</th>
<th>Using Displacement</th>
<th>Using intervals of Displacement</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.999</td>
<td>0.2</td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0.995</td>
<td>0.2</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0.990</td>
<td>0.2</td>
<td>5</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0.950</td>
<td>0.2</td>
<td>8</td>
<td>9</td>
<td></td>
</tr>
</tbody>
</table>

**Tab. 5: Postal Codes**

<table>
<thead>
<tr>
<th>Rule No</th>
<th>Postal Code</th>
<th>City</th>
<th>$p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>#1 1 #2 0 #3 ? #4 0</td>
<td>Praha</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>#1 4 #2 6 #3 0 #4 1</td>
<td>Liberec</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>#1 4 #2 1 #3 7 #4 ?</td>
<td>Krupka</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>#1 5 #2 1 #3 ? #4 4</td>
<td>Rokytnice nad Jizerou</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>#1 ? #2 1 #3 2 #4 4</td>
<td>Rokytnice nad Jizerou</td>
<td>0.9996</td>
</tr>
</tbody>
</table>

Now let’s focus on completeness of the rule (15). The list of all parts of “Rokytnice nad Jizerou” city is shown in Tab. 6. As we can see, the rule (15) covers only some part of potential values of Postal Code. Using RUIAN again, we can observe that this coverage represents just 60% of all possible address points.
However, as it is shown in Tab. 7, it would be bad mistake to extend the original rule by “51745” constraint, since this Postal Code also covers “Chleny”. Using RUIAN we can find that resulting rule would have Confidence \( p \) just 0.95.

Applying 4FT-miner procedure to US attributes of products (UPC, Brand, Model, MPN) with pre-defined Confidence \( p = 0.999 \) and required Support \( a = 2\% \) three interesting validation rules have been extracted. Since all of them have Confidence \( p = 1 \) they could be considered as “hard rules”:

- Model (3GS) \( \gg \ll \) Brand (Apple) \( (17) \)
- Model (4s) \( \gg \ll \) Brand (Apple) \( (18) \)
- Model (5) \( \gg \ll \) Brand (Apple) \( (19) \)

Reducing required Support \( a \) to just 0.5\% a huge number of interesting rules based on UPC has been identified, e.g., rule number (20) which assigns “Nokia” brand to single UPC. This rule represents very interesting and potentially useful knowledge, since the relation between UPC and brands is usually more complex.

### Tab. 6: Deep dive into “Rokytnice nad Jizerou”

<table>
<thead>
<tr>
<th>Part of City / Village from LOV</th>
<th>Postal Code</th>
<th>Covered by Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#1</td>
<td>#2</td>
</tr>
<tr>
<td>Dolní Rokytnice</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Františkov</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Hleďsebe</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Horní Rokytnice (část)</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Horní Rokytnice (část)</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Hranice</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Rokytnice nad Jizerou-Dolní Rokytnice x)</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Rokytnice nad Jizerou-Františkov x)</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Rokytnice nad Jizerou-Hleďsebe x)</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Rokytnice nad Jizerou-Horní Rokytnice (část)</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Rokytnice nad Jizerou-Hranice x)</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Rokytnice nad Jizerou-Rokytno x)</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Rokytnice nad Jizerou-Studenov x)</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Rokytno</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Studenov</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>

### Tab. 7: Deep Dive into Postal Code “51745”

<table>
<thead>
<tr>
<th>Part of City / Village from LOV</th>
<th>Postal Code</th>
<th>Covered by Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#1</td>
<td>#2</td>
</tr>
<tr>
<td>Chleny</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Františkov</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Horní Rokytnice (část)</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Rokytnice nad Jizerou-Františkov x)</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Rokytnice nad Jizerou-Horní Rokytnice (část)</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Rokytnice nad Jizerou-Rokytno x)</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Rokytno</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>
Another kind of rules which have been recognized is related to structure of EAN. This unique product code could be parsed either to tokens representing country / numbering system (first 2-3 digits), manufacturer (next 4-5 digits) and product code (8-12 digit) or information about brand. Brand name could be captured e.g., from GEPIR (Global Electronic Party Information Registry) using first 7 digits of EAN (so called GCP) as a merge key. As an example of such kind of potentially useful rules I could mention the rule (21) below which links “Apple” brand to concrete GCP.

\[
\text{GCP (0885909) } \rightarrow \leftarrow \text{Brand (Apple)}
\]

In practice, these rules have been successfully implemented as a part of inventory linking process in eBay. The purpose of this initiative was an identification of inventory gaps between different eBay sites in Europe (FR, IT, ES, DE and UK). To get the right story it was absolutely necessary to have an assurance that information extracted from unstructured textual fields serving as an item description is valid and corresponds with already known reference facts. These facts were represented as a set of rules stored in knowledge base.

2.6 Results of Evaluation

During evaluation of “blind” extraction, several potential issues have been identified. However, at least some essential knowledge about respective domain is needed on the researcher’s side. Association Rules require category data. Continuous variables should be transformed to intervals rather to separated values. There are several possible strategies how to realize this transformation:

- To split values into intervals with the same length (equidistant intervals)
- To split values into intervals with the same frequency of values
- To discretize values using criteria of minimal entropy
- To use discretization criteria defined by user (e.g., pre-defined subsets, using regular expressions, using blocking rules, using restriction to not missing values, etc.)

It’s on researcher to decide which strategy to use and which exact frequency / length / rules to choose. Unfortunately in a lot of real applications these intervals could have exact meaning. They can have different lengths and different frequencies of values. As an example I can mention tariff categories in Motor Vehicle Insurance. Boundaries of intervals are derived from risk which is related to respective category. The frequency of vehicles in categories differs as a structure of portfolio is not homogenous. However, even using this “blind” extraction, at least some interesting and potentially useful hypotheses have been identified.

The second main issue is the necessity to evaluate resulting rules from the point of their validity. There is some significant difference between traditional extraction of Association Rules as frequent and accurate patterns and the extraction for a purpose of validation. Whereas in traditional approach the best rules are those with high Confidence and at least defined level of Support, in Data Validation area a lot of rules can have relatively low Support, since sometimes they describe rare patterns. However, even those patterns could be used for validation, but after necessary evaluation by domain expert. An idea to extract all potentially useful hypotheses could lead to overflow of using system and validating expert as well.

The problem is also the general validity of source data. Within a process of KDD, data are usually validated before an extraction of knowledge is started. However, in extraction of validation rules, we can either use already validated data source with a goal to identify all correct rules which could be directly used for future validation, or review all “raw” rules and use this technique also for identification of hidden discrepancies in data (to use a process of knowledge discovery as a process identifying frequent or hidden errors).

During my experiments I was also dealing with some limitations of particular software I used. For some variables with a huge amount of categories it wasn’t possible to load all of them. This led to some additional work when knowledge captured from reduced data had to be confronted with original data. On the other side as a significant advantage of this software I see a possibility to set up a maximum length of Antecedent and Succeedent which I haven’t found in other tools.
2.7 Typical Scenarios of Application

A typical situation when described approach could be used is when a large amount of data should be evaluated in relatively short time period. With increasing volume, velocity and variety of processed data sources this scenario becomes more than realistic. It is no more possible to realize assessments of data quality on basis of manual inspection and provided by several voters as described e.g., in (English, 1999).

A typical application scenario is also a situation when large amount of attributes for each entity is available. However, even a small number of attributes could be potential good source of rules. Necessary condition is of course an existence of dependencies among these attributes.

Typical application areas which meet both conditions above (large amount of data and dependency between attributes) could be:

- Attributes describing entities like objects (motor vehicles), services, products (product catalogue), individuals, commercial parties, their contacts and addresses. Generally so called Master Data.
- Attributes describing transactions and additional observations (e.g., using different sensors) related to Master Data, e.g., diagnostics, bank transactions, insurance claims, purchases, etc.

2.8 Recommended Process of Implementation

The process described below I consider as my own contribution for discussed problematic. However, we can find there some links to different methodologies from KDD area such as CRISP-DM. I also found some corresponding steps in not just domain specific approach of (Medlock et al., 2011) who consider 7 steps of transforming clinical rules for their use in decision support system:

- Determine whether rule can be proactively operationalized;
- Reformulate rules to logical statements;
- Assess for conflicts between rules;
- Exclude unnecessary and redundant concepts;
- Classify concepts as crisp or fuzzy;
- Determine the relatedness of rules;
- Determine the availability of data in local systems.

In approach recommended by me (see Fig. 1) I consider these important steps of rules extraction process:

- Definition of a priori knowledge by domain expert;
- Data Extraction from original data sources. In this phase, a priori knowledge about definition of Data Universe is used;
- Data Preparation as a common step of Knowledge Discovery process;
- Rules Extraction using some algorithm for mining Association Rules. In this phase, additional knowledge about used algorithm is required;
- Next step represents a selection of those rules which are potentially useful and their validation with domain experts or by confrontation with current rules;
- After that, similar rules are combined and generalized;
- Once the rules are prepared for intended use, it is necessary to store them and make them accessible for different applications and users. The recommended storage is a knowledge base focused on support of Data Quality Management processes, so called QKB (Quality Knowledge Base). The model of this QKB should be derived from industry Common Data Model (see the explanation below);
- Last two steps in validation rules lifecycle represent their application and maintenance by Data Steward, the person responsible for data quality within respective domain.

The notion of Canonical Data Model (also Common Data Model, CDM) comes from data integration area. Its meaning can be understood as a derivative relating to the canon (a set of principles, a concept of ideal proportions). In mathematics the canonical form is the form in which an object can be clearly presented. In the field of data integration (Stumpf and Dzmuran, 2008) mention it as the model independent of the particular application. (Howard, 2008) speaks about "a data model that spans enterprise applications and different data sources". As in practical implementations of Business Rules...
within tools supporting Data Quality Management (e.g., SAS DataFlux) these rules are usually defined using some common attributes, it seems to be reasonable to align these common attributes with elements of Canonical Data Model. In my approach I consider Canonical Data Model as a basis of Quality Knowledge Base.

Fig. 1: Recommended Validation Rules Lifecycle

2.9 Possibilities of Rules Integration

Common way how to share models created within Knowledge Discovery from Databases (KDD) process across different tools is PMML (Predictive Model Markup Language) standard. This kind of output is provided by majority of KDD tools (e.g., Lisp Miner). However, direct PMML support in tools for Data Quality Management is not implemented yet. So it is not possible to use found validation rules directly. On the other side, it’s not difficult task to parse PMML elements and transform them to any other syntax. See example of structure below.

(Sottara et al., 2011) provide description how to use PMML to enhance JBoss Drools production rule engine. Drools represents open source platform for business logic integration, rules, workflows and events processing. Decision engines like Drools are standard component of DQM tools and enable batch execution of predefined business rules (with the functionality of validation rules) against production data.

Code below demonstrates the example of how it is possible to represent the rule (22) using selected elements of PMML. First part represents description of rule itself.

\[
\text{Displacement} \ (1000;1350>) \ \& \ \text{Year of Registration} \ (1997) \ \leftrightarrow \ \text{Fuel} \ (BA)
\]

(22)

```
<AssociationRule id="844" antecedent="DBA_Antecedent_844"
consequent="DBA_Succedent_844">
  <Text>Displacement (1000;1350) &\ YearofRegistration (1997)
  
  &\ Fuel (BA)</Text>
  <IMValue name="Conf">0.9997945979</IMValue>
</AssociationRule>
```

Only few elements important for construction of validation rule have been selected from relatively long original code. In the role of attributes of respective rules we can see rule identification and relations to Antecedent and Succeedent. As a part of rule metadata there is also mentioned its plaintext form
placed inside of <Text> element. Following the reference to respective Antecedent and Succedent we can find corresponding elements.

<DBA id="DBA_Antecedent_844" connective="Conjunction">
  <Text>Objem_int(I2) &amp; Rokvyr(1997)</Text>
  <BARef>DBA_FTCedentI_1687</BARef>
</DBA>

<DBA id="DBA_Succedent_844" connective="Conjunction">
  <Text>Palivo(BA)</Text>
  <BARef>DBA_FTCedentI_1688</BARef>
</DBA>

Following the reference to partial cedents:

<DBA id="DBA_FTCedentI_1687" connective="Conjunction">
  <Text>Objem_int(I2) &amp; Rokvyr(1997)</Text>
  <BARef>DBA_FTLiteralI_Sign_2860</BARef>
  <BARef>DBA_FTLiteralI_Sign_2861</BARef>
</DBA>

Partial cedents refer to respective definition of literals connection:

<DBA id="DBA_FTLiteralI_Sign_2860" connective="Conjunction" literal="true">
  <Text>Objem_int(I2)</Text>
  <BARef>BBA_FTLiteralI_2860</BARef>
</DBA>

And finally at the level of literals we can find the reference to respective field.

<BBA id="BBA_FTLiteralI_2860" literal="false">
  <Text>Objem_int(I2)</Text>
  <FieldRef>Objem_int</FieldRef>
  <CatRef>I2</CatRef>
</BBA>

Using this composite information we are able to reconstruct validation rule and transpose it to any other form of representation.

3. Conclusion

Practical experiments confirmed that it would be possible to use 4FT-miner procedure for automatic extraction of at least some hypothesis which could serve as validation rules. However, using this approach, a lot of important rules remained uncovered since this kind of rules has usually small Support a. From this perspective the extraction of validation rules a little bit differs from a traditional KDD task of Associative Rules mining. Reducing the parameter of required Support could lead to expansion of identified hypotheses and increase allocated capacity of expert to filtrate / generalize / combine rules and exclude those which represent current issues in data.

At least some basic hypotheses have been extracted without any additional expert knowledge applied in phase of data preparation. However, later inclusion of this knowledge helped to extract higher number of more useful hypothesis.

Extracted rules could be possibly integrated with some kind of decision engine using e.g., PMML standard. Drools tool represents an example of such kind of engines. These engines are then responsible for execution of predefined rules on production data.

Important question is why to use rules instead of original lists of values and reference sources from which are rules introduced in this article derived. Added value of this concept is especially in a situation when (1) data used for validation had to be integrated from a lot of different sources, (2) when some level of intended volatility (not just trivial application of some similarity metric) is necessary to use and (3) in a situation when some conditioned combinations of attributes should be used for validation.
However, it needs to be mentioned, that Data Validation is not the only single possible way how to use Association Rules within Data Quality Management area. Extracted rules could be used also for:

- Imputation of missing values in situation when attribute with missing values is in the role of Succedent;
- Standardization of formats and synonyms when these rules are extracted from so called standardization schemes (pairs of original and standardized values) or manually defined by experts;
- Data merging and deduplication (with the meaning of duplicity removal) when rules are combined with similarity metrics.

Bibliography


Berka, P., 2003: Dobyvání znalostí z databázi. Academia


McGilvray, D., 2008: Executing Data Quality Projects: Ten Steps to Quality Data and Trusted Information. Morgan Kaufmann


Vorisek, J. et al., 2008: Principy a modely řízení podnikové informatiky. Oeconomica

JEL Classification: C80, Y80