RULE-BASED MEASUREMENT OF DATA QUALITY IN NOMINAL DATA

(Research-in-Progress)
(IQ Concepts, Tools, Metrics, Measures, Models, and Methodologies)

Jochen Hipp
Group Research & Advanced Engineering
DaimlerChrysler AG, Ulm, Germany
jochen.hipp@daimlerchrysler.com

Markus Müller
Laboratory for Semantic Information Technology
University of Bamberg, Germany
markus.mueller@wiai.uni-bamberg.de

Johannes Hohendorff
Institute of Applied Information Processing
University of Ulm, Germany
johannes.hohendorff@uni-ulm.de

Felix Naumann
Hasso Plattner Institute
University of Potsdam, Germany
naumann@hpi.uni-potsdam.de

Abstract: Sufficiently high data quality is crucial for almost every application. Nonetheless, data quality issues are nearly omnipresent. The reasons for poor quality cannot simply be blamed on software issues or insufficiently implemented business processes. Based on our experiences the main reason is that data quality shows the strong tendency to converge down to a level that is inherent to the existing applications. As soon as applications and data are used for other than the established tasks they were originally designed for, problems arise.
In this paper we extend and evaluate an approach to measure the accuracy dimension of data quality based on association rules. The rules are used to build a model that is intended to capture normality. Then, this model is employed to divide the database records into three subsets: "potentially incorrect", "no decision", and "probably correct". We thoroughly evaluate the approach on data from our automotive domain. The results it achieves in identifying incorrect data entries are very promising. In the described setting, for the first time ever it was possible to highlight a significant number of incorrect data records that otherwise disappear in the millions of correct records. This ability enables domain experts to understand what is going wrong and how to improve data quality. Moreover, our approach is a first step towards automatically quantifying the overall accuracy of a yet unseen dataset.

Key Words: Data Quality, Information Quality, Outlier Detection, Data Mining

INTRODUCTION

High data quality is crucial for almost every application. But in contrast to its obvious importance for organizations, data quality is still an often underestimated issue. This valuation is especially astonishing in the face of the many examples for projects that have failed or were seriously detracted due to data qual-

ity deficiencies, c.f. [8], [10], and [12]. There might be several explanations that go beyond blaming data quality problems on implementation issues or insufficient business processes only. One of the main reasons we experienced is the following: Data quality has the strong tendency to converge down to a level that is postulated by existing applications. As soon as data is used in new applications that are put into production to perform other tasks than the data collection was intended for in the beginning then data quality typically becomes an issue.

For illustration consider the following example. In the early 90s bar-code scanners were introduced in retail business. Their first application was to automate pricing and price registration. The cashier was satisfied as long as the scanned price corresponded to the actual price. In the sense of this application data quality was perfect. But with collecting such data, the desire for new applications arose. Soon, managers began to believe that they know exactly what is sold on each day and as a result, for example, merchandise planning and control systems arose. Besides, data mining research tremendously gained momentum based on the expectations arising from this and comparable new data sources, c.f. [1]. What people painfully learned after a first phase of enthusiasm was that in the cashier's world only prices were collected and not which actual items were sold. A cashier did not care whether sweets or bolts went over the counter as long as $1.99 was put on the bill. In other words, the level of data quality necessary to run the current application was far below the demands of the new applications and in the first instance had to be tediously improved in order to gain the expected benefits.

Another point we would also like to mention is that in our experience many IT-people, who are usually responsible for handling data, focus on technology and not on the business. As a result they do not seem to be very interested in data quality issues. As long as their application stores, handles, and processes data correctly they typically do not see any need to care for data quality.

Data Quality

Since the beginning of academic research in the field of data quality, great progress has been made. Today the basic concepts behind data and information quality are well understood, c.f. [4], [8], [9], [10], and [12]. The common notion is that data quality should always be seen in an application context. In that sense, data quality is captured by the definition “fitness for use”. This general term has been broken down into many dimensions of nearly arbitrary detail. Some of the most common dimensions are accuracy, completeness, relevance, and interpretability. Other dimensions include reputation, accessibility, or even access security. Without going too much into detail, there is an obvious difference in how far these dimensions are core dimensions or cover information systems in a broader sense. In any case, in accord with Olson we regard accuracy as the most important data quality dimension [10]: “The accuracy dimension is the foundation measure of the quality of data. If data is just not right, the other dimensions are of little importance.” Thus, this paper tackles a core problem of data quality assessment, namely measuring accuracy.

Measuring Accuracy by Outlier Detection

In spite of recent advances in the field of data quality, our impression is that the actual measurement of accuracy is still far from being effectively supported by tools or techniques. Data Profiling, c.f. [10], as a univariate approach can be only a first step. In other words, measuring accuracy is still a duty carried out by domain experts manually inspecting database records. In the context of databases beyond several gigabytes of size, manual inspection is an infeasible task. In this paper we pick up and improve an approach to overcome this unsatisfying situation: We employ *rule based outlier detection* to measure accuracy. One of our basic assumptions is that outliers are highly suspect of accuracy deficiencies, i.e., records that contradict a model of normality captured in a rule set are likely to be incorrect.

Detecting outliers as records that violate certain rules is a straightforward approach. Think of univariate rules, such as

ZIP=72076 → CITY=Tübingen
PREC=snowfall → TEMPERATURES=cold

or multivariate rules, such as

CAR=Mercedes C-Class AND TYPE=Station Wagon → PRODUCTION PLANT=Bremen.

Obviously conformance or nonconformance to such rules allows conclusions about the correctness of records. The exemplary rules above are quite trivial. Even a non-expert can compile a set of such rules. For more complex domains, as is the case in our application, a domain expert is needed to formulate such rules. Typically experts are rare, expensive, and in practice not easily convinced to thoroughly formalize their entire knowledge. Moreover, even experts cannot be aware of all dependencies. In addition, rules that do not hold by 100% may often be neglected by the experts. In many domains the number of valid rules is in the thousands. Finally, it is indispensable to keep the rule set updated with regard to the data. If the characteristics of the underlying data are fast changing then the obstacles mentioned above are multiplied. For example, new products may be introduced or old products may disappear on a regular basis. In our domain we would need an expert to adjust the rule set at least once per month.

As a consequence we decided to employ an approach that automatically derives the rules from the data itself. We chose association rules for that purpose, first because there is a broad range of algorithms available to efficiently generate such rules, c.f. [2], [6], and second, in contrast to other approaches such as decision trees, the complete search space with respect to the minimum thresholds for support and confidence is enumerated. Moreover, association rules are symbolic as most attributes[1] in our application domain. Discretizing the remaining attributes into meaningful intervals turned out to be straightforward. Employing other approaches to capture the structure of the data also seems promising, for example, decision trees or neural networks. However, identifying outliers in nominal and sparse data is not straightforward with statistical outlier approaches.

Our basic idea described in [5] is to induce association rules from the data and hypothesize that these rules capture the structure of the considered data, i.e., represent a model of normality as long as the available data set is large enough. Then, we take the generated rule set and apply it to the data for which the degree of accuracy is to be measured. In our case the rules are always employed to the same data set from which they originate. Depending on the application, splitting the data into training and evaluation sets may also be a good choice. Each single record in the data may support rules, some or all rules may not be applicable to it, or the record may contradict to one or more of the rules. Depending on this, every record in the data is assigned a score to it. This score is computed as the number of violated rules where every rule is weighted by its confidence. Details are given in the third section, where we describe our extensions to the approach presented in [5].

Of course, if the accuracy of the data is too poor, deriving a valuable model of normality is infeasible, i.e., when incorrect values become normality our approach will no longer be able to identify outliers in the sense of incorrect records. Nevertheless we obtained very promising results in several evaluations.

[1] In this paper we will use the terms *attribute* and *variable* synonymously.

Outline of this Paper

First we characterize the application scenario to which we want to apply our approach. We start out the third section by formalizing the initial idea that we took from [5]. Then, we describe several enhancements, which go far beyond the existing approach. These improvements originate from the demands of our practical application. In Section 4 we thoroughly evaluate the usefulness of the basic approach and our extensions with the help of a domain expert. We want to point out that for the first time ever the basic approach is applied to real data. Finally, the paper concludes with a short summary and future work.

OUR APPLICATION SCENARIO

In our application an operative database system stores information on business transactions. Each transaction is described by a varying number of attribute values, sometimes just three values, sometimes up to several hundred values per transaction. The attributes are mainly symbolic; discretizing the few numerical values into meaningful buckets is straightforward. Moreover each business transaction gets classified by a human being during data collection. This classification is also stored in the database. We call the actual value assigned to each transaction its *label* and distinguish it from the *descriptive attribute values* that characterize each transaction (Table 1).

Collecting the labels of the transactions is already implemented in the operative system. Yet, up to now the classification labels have not been seriously exploited for downstream business processes. Accordingly, it is not sure whether the quality of the data stored in the label attribute is already sufficient for upcoming applications, such as decision support systems and business intelligence applications. Our task is to measure the quality of this classification field in historic data with regard to potential new applications. The huge number of transactions, at least several million per year, together with the low number of actually wrong data entries, makes manual inspection infeasible. In addition, the collected business transactions follow the same general structure, but finally describe events from a large number of different technical areas (*domains*). Therefore, even domain experts can evaluate the correctness only of those transactions that belong to their area of expertise. Note that the sets of labels for each area are disjoint (Table 1).

tid	descriptive attributes	label attribute	domain	correct label	correct domain
1	a1, b2, c1	lx1	X	lx1	X
2	a1, b3, d1	lx1	X	lx1	X
3	**a2, b4**	**lx1**	**X**	**ly1**	**Y**
4	a3, c1	lx2	X	lx2	X
5	a3, c2, d1	lx2	X	lx2	X
6	a2, b4, d2	ly1	Y	ly1	Y
7	a2, b4	ly1	Y	ly1	Y
8	**a3, c1, d1**	**ly2**	**Y**	**lx2**	**X**

Table 1: The example shows exemplary transactions consisting of descriptive attribute values and a label. Each transaction belongs to a specific domain derived from its label. The dotted line separates the two domains X and Y. Transactions 3 and 8 are marked as incorrect by domain experts.

EXTENDED APPROACH

Our application scenario differs in several ways from the general scenario on which the approach introduced in [5] is based on. First, in [5] the focus is on simply sorting the outliers by descending order according to the number of unsatisfied rules. However, in our real-world application we need to go beyond sorting, we need to identify and quantify outliers and correct values, i.e., we want to clearly separate records for which the rules indicate correctness from the records that are likely to be incorrect. Furthermore in [5] there is not a single variable for which the quality is to be measured, but the dependencies between all existing attributes contribute equally to the score. Moreover the problem of redundancy is not addressed in [5].

Inclusion of Positive Scores

The general goal is to assign a score to each record that captures in how far this record is conforming or non-conforming to the available rule set. For that purpose we assign a score $s \in \Re$ to each record that is computed on the basis of the generated rules. The score is a means to capture the consistency of a single record with the rule set as a whole. The basic idea in [5] is to assign high scores to records that are suspected of deficiencies. Here we extend this approach by distinguishing between negative and positive scores: We assign negative scores to outliers and positive scores to those records that are likely to be correct.

In the following we define the extended score as the number of fulfilled rules minus the number of violated rules while weighting each rule with its corresponding confidence. Let R be a set of association rules and let D be a database of transactions, both containing only items from a common universe I. Let $r = X \rightarrow Y$ be an association rule with $body(r) = X$ and $head(r) = Y$. Let the mapping violates, which determines whether a transaction $T \in D$ violates a rule $r \in R$ or not, be defined as:

$$violates : D \times R \rightarrow \{-1, 0, 1\} :$$

$$(T, r) \mapsto \begin{cases} -1 & \text{if } body(r) \subseteq T \wedge head(r) \not\subseteq T \\ +1 & \text{if } body(r) \subseteq T \wedge head(r) \subseteq T \\ 0 & \text{else} \end{cases}$$

Based on this mapping we assign a score to each transaction by summing the confidence values of the rules it violates. As only such rules should be taken into account that hold with a certain confidence, we restrict the rule set R to $R_\gamma = \{r \in R \mid confidence(r) \geq \gamma\}$.

In general, our experience is that minimum support should be chosen as low as possible. It is mainly the algorithmic challenge of generating and applying huge rule sets that places a limit on this value. The rule sets are typically not inspected manually by humans, so their size is mainly a technical issue.

Based on the definition from above we compute the scores as follows:

$$score_{R\,\gamma}: D \rightarrow \Re :$$

$$T \mapsto \sum_{r \in R_\gamma} confidence(r)^{\tau} \cdot violates(T, r)$$

The tuning parameter $\tau \in \Re_0^+$ introduces a nonlinear weight to suppress rules with smaller confidence values.

Using the small data set in Table 1, we want to show the intuition behind the algorithm: Table 2 contains some of the rules generated from the data set when no minimum support threshold is set. Although we show rules with only one item in the rule premise, in practice, longer rules are preferred. After having generated and filtered the rules with low confidence values, the scoring function is applied to the transactions. Table 3 shows the scoring after having processed the first rules I to X with a minimum confidence above 50%. The incorrect transactions 3 and 8 already catch one's eye because of their low scores.

#	rule	confidence		confidence > 0.5
I	a1 → lx1	1	(100%)	✓
II	a2 → lx1	0.33	(33%)	×
III	**a2 → ly1**	**0.66**	**(66%)**	✓
IV	a3 → lx2	0.66	(66%)	✓
V	a3 → ly2	0.33	(33%)	×
VI	**lx1 → a1**	**0.66**	**(66%)**	✓
VII	lx1 → a2	0.33	(33%)	×
VIII	ly1 → a2	1	(100%)	✓
IX	lx2 → a3	1	(100%)	✓
X	ly2 → a3	1	(100%)	✓
.	.	.		.
.	.	.		.
.	.	.		.

Table 2: Some exemplary rules generated from the transactions shown in Table 1. In this example we show rules with only a single item in the premise to keep things simple. Although computational costs increase, in practice longer rules are preferred as they provide additional and more accurate information.

transactions			rules that contribute to score: confidence > 50%								current score (τ =1)	correct label
			I	III	IV	VI	VIII	IX	X	. . .		
			1	0.66	0.66	0.66	1	1	1			
1	a1, b2, c1	lx1	1	0	0	1	0	0	0	.	1.66	lx1
2	a1, b3, d1	lx1	1	0	0	1	0	0	0	.	1.66	lx1
3	**a2, b4**	**lx1**	**0**	**-1**	**0**	**-1**	**0**	**0**	**0**	.	**-1.32**	**ly1**
4	a3, c1	lx2	0	0	1	0	0	1	0	.	1.66	lx2
5	a3, c2, d1	lx2	0	0	1	0	0	1	0	.	1.66	lx2
6	a2, b4, d2	ly1	0	1	0	0	1	0	0	.	1.66	ly1
7	a2, b4	ly1	0	1	0	0	1	0	0	.	1.66	ly1
8	a3, c1, d1	ly2	0	0	-1	0	0	0	1	.	0.33	lx2

Table 3: Application of the scoring function on the exemplary transactions from Table 1. The table shows the current score for each transaction after having applied the rules I to X from Table 2 with confidence > 50%.

Introduction of Thresholds

We introduce thresholds as a means to distinguish (a) potentially incorrect records, (b) records for which there is no decision available and (c) records that are likely to be correct. Separating and quantifying these sets is a major requirement of our application. Of course, defining the thresholds from scratch is quite arbitrary. We can anticipate that both thresholds are probably not symmetric around zero. The reason is the high a priori probability that a transaction is correct. This implies that classifying a transaction as incorrect or outlier should be based on strong arguments, i.e., a comparably low negative score. On the other hand, even small positive scores strongly support the a priori proposition that a transaction is correct. At least in our application this meets the user's expectations best. At the same time one has to keep in mind that the number of records with "no decision", of course, should be kept as small as possible.

Introduction of a Target Variable

In the approach from [5] all attribute values, i.e., descriptive attribute values and labels, available for each record are treated equally. This implies that all violated and satisfied dependencies between attribute values contribute to the score of a transaction in the same way. Whereas this is reasonable for the purpose of finding outliers in general, our application is different: we want to measure the accuracy of the classification of transactions done by human beings, i.e., we want to identify outliers with regard to the label assigned to a transaction. Therefore, the approach should primarily focus on dependencies between descriptive attribute values and the classifying label of a transaction. Consider the example from above (Table 1): The dependency a2 → b4 (confidence = 100%), for example, does not directly contribute to uncover the fact that transaction 3 is mislabeled.

Our solution is to leave the scoring function introduced above as it is, but to filter the rule set instead. The probably most obvious approach would be to include only those rules into the scoring that contain a label in the consequence of the rule. However, in this case we observed that the information contained in the rules would not be fully exploited: We would miss the information of those rules that state that a certain label also requires certain descriptions in a transaction. Therefore, we filter the rule set in such a way that the result set consists of exactly those rules that contain a label, no matter in the rule premise or the rule consequence. The exemplary rule set shown in Table 2 is the result of such a filtering.

Dealing with Separate Domains

As mentioned in Section 2, the transactions of our data set describe events from several distinct technical areas respectively domains. The records follow the same structure, i.e., there exist the same descriptive attributes and there exists a label attribute that is set by a human being. However, descriptive attribute values and especially labels differ among the domains. Besides, users are not interested in several domains at a time. Moreover, an expert can evaluate only whether a label is correct or not within his specific domain. Consider the example from Table 1: An expert for domain X would be interested only in transactions containing a label of form lx. As a result we apply the scoring for each domain separately. So the question we want to answer is: From all transactions labeled with labels from a certain domain, how many of those labels are correct, how many of them are incorrect?

An obvious approach would be to pick up the filtering of the rules from above and extend it to restrict the rule set to contain just those rules that contain labels from the domain under consideration. But actually this would not always yield to the desired results. The problem is the existence of incorrect values that span two different domains. In fact, to discover mislabeling that goes beyond domain boundaries we should employ the complete set of rules. If a transaction is incorrectly labeled and therefore appears in the wrong domain, then it is not sure whether there exists a rule in this domain that would uncover this mislabeling. In fact, it is even more likely that rules from the actually correct domain exist to indicate the

incorrect label. In order to address this we always need to generate rules from the entire set of transactions. Then, we can filter the transactions that belong to the domain under consideration. To shed some light on this, have a look at our example again (Table 1, Table 2, and Table 3): The misclassification of Transaction 3 in domain X would be less obvious if we did not consider rule a2 → ly1 that belongs to domain Y.

Tackling Redundancy

Initially, we follow the approach introduced in [11] and [13] to address redundancy: Instead of deriving rules from all frequent itemsets by considering all subsets, c.f. [2], we derive rules only from the so called "closed itemsets". In the following, when we refer to the set of association rules, we always have these "closed" association rules in mind.

In addition to this basic filtering we employ other distance measures to restrict the rule set. In particular we use measures that are based on similarity of the attribute values in the rule premises and on the similarity of the transactions covered by the rules. In the first case, a rule is rated as redundant if it shares most of its attribute values with another rule. In the second case, a rule is redundant if another rule exists that covers the same or nearly the same database transactions. Besides, we implemented hybrid measures that incorporate both approaches.

EVALUATION

We evaluate the proposed approach on a real-world dataset containing about 100,000 transactions from the application scenario described above using an implementation in Perl and SQL. In addition, we employ an existing implementation of the Apriori algorithm for frequent itemset mining [3]. Rule generation takes only a few minutes and has never been a serious problem in our experiments. In contrast exporting the data from the database system takes several hours and is therefore the true bottleneck. Instead, we evaluate the *quality of the ranking* provided by the score, i.e., we test whether a transaction is more likely to be an outlier if its score increases and vice versa.

Test Data Characteristics

As we do not add artificial impurities to the data, we can test our algorithm only if an expert validates the labels of all, or at least a random sample of all transactions. It is obvious that no expert will ever check the labels of 100,000 transactions. So why not take a representative sample? As in our application scenario the fraction of incorrect labels is very small, we can motivate the experts to reassess transactions only if they can actually identify a large number of incorrect ones during their evaluation. Therefore, we focus on approximately 16,000 transactions from a single domain and a specific time period for which a data accuracy problem has already been identified and for which the experts expected a high percentage of data entries with incorrect labels. Nevertheless, the fraction of incorrect labels is still too small in this set to pick a random sample that would be presented to the experts. To pre-select conspicuous transactions, we run the algorithm with a rough parameterization and rely on explanatory narratives, which help to understand the business transactions. After several iterations of presenting transactions with low scores to the experts, we end up with 1,336 validated transactions. 375 of these transactions are identified to be labeled correct, 551 are labeled incorrect and 410 are probably wrong, but the experts were not capable of making a final decision. The explanation for the latter is: The events that initially caused these business transactions took place in the distant past and can no longer be reproduced for sure from the data and additional narratives. In the following, we distinguish three base classes of transactions: "correct", "incorrect", and "undecided". Furthermore, based on interviews with the experts, we introduce the class "probably incor-

rect", which covers all "incorrect" transactions and 50% of the "undecided" transactions in order to reflect the strong bias in the class "undecided" towards "incorrect".

Experiments

Based on the original data and the expert information we compare the basic approach from [5] with the extensions proposed in this paper. In general, the experiments are set up as follows: First, closed itemsets are generated from the entire dataset (the approximately 100,000 transactions) under a minimum support constraint. In the next step, closed rules are generated from these itemsets. After this, all resulting rules are applied to the about 16,000 domain specific transactions to calculate their scores. As mentioned, for 1,336 of these 16,000 transactions there exists a validated label that can be used to evaluate the quality of the score.

Parameter values have to be chosen carefully with regard to the application domain. A first critical value is the right minimum support: If minsupp is chosen too small, we run the risk of treating outliers as "normal" transactions. On the other hand, we have to keep in mind that in our domain, target values can be quite infrequent, but one would still consider these values "outliers". In our context a minsupp between 0.1‰ and 0.3‰ best accounts for this tradeoff. Furthermore we set minconf to 75% and 85% as rules with lower confidence values proved to be of limited use for outlier detection in tests on synthetic data. For a comparison of the impact of different minsupp and minconf values on our real-world data see Figure 1. The calibration parameter τ is set to 3 and the maximum rule length to 4. Different τ-values and a rule length of 5 do not have a significant impact on the results.

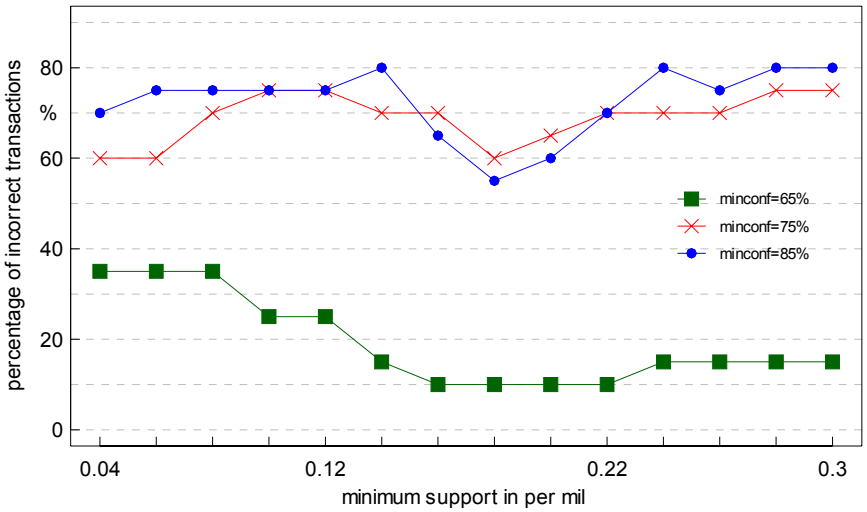

Figure 1: Percentage of actually incorrect labels among the 20 transactions with the lowest scores for various combinations of minimum support and minimum confidence.

The main purpose of the evaluation is to compare the effect of different extensions of the approach on the quality of the scoring. We introduce two types of charts to visualize the scoring of the transactions: Both chart types show relabeled transactions ranked by their assigned score in ascending order. Charts of the first type (Figure 2, Figure 4, and Figure 6) show a fine-grained view of the first 150 transactions with the lowest scores, while charts of the second type (Figure 3, Figure 5, and Figure 7) show a coarse view of all relabeled transactions. In the first, high-resolution view, each data point (m, $\text{fraction}_k(m)$) represents the fraction of transactions of class $k \in$ {correct, incorrect, probably incorrect, undecided} within the first m

transactions. In contrast, the second chart type shows a moving average, i.e., each data point represents the fraction of class k transactions within the sliding window [m-150, m]. The trend is highlighted by smoothing splines in charts of the second type.

Figure 2 and Figure 3 show the results of the original algorithm presented in [5] applied to the data with minsupp = 0.25‰ and minconf = 85%. For the first time, the approach presented in [5] is evaluated not only on synthetic data, but on real-world data. 37,006 relevant rules are generated from the 100,000 transactions and 15,069 of the 16,416 transactions are covered by the rules. There are only about 30% outliers among the 150 transactions with the lowest score. Moreover, plenty of outliers can be found among the transactions with the highest score. All in all the initial results are disappointing.

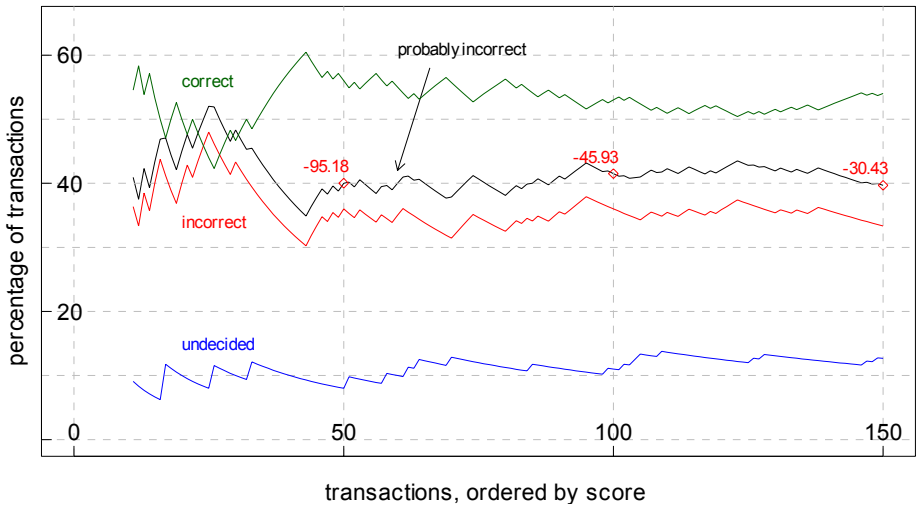

Figure 2: Resulting scores for the validated transactions when applying the algorithm as presented in [5]. The chart shows the 150 transactions with the lowest scores sorted in ascending order (minsupp = 0.25‰ and minconf = 85%).

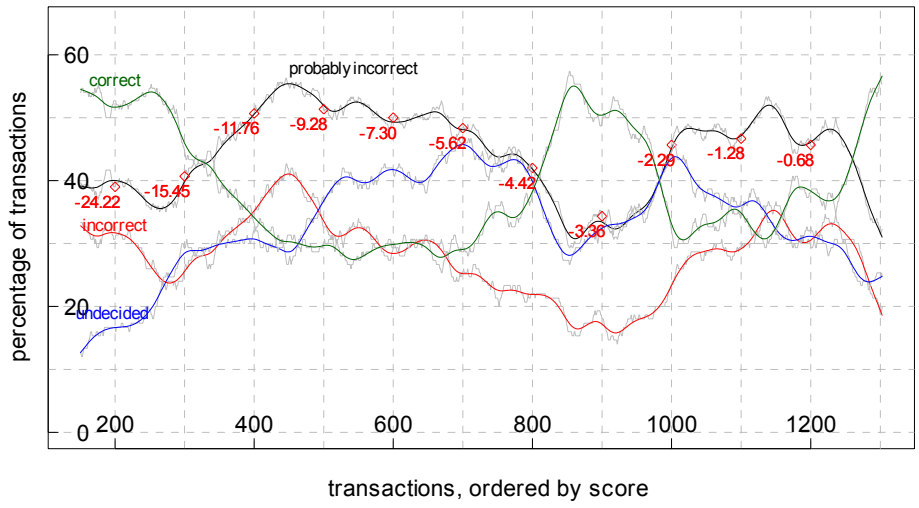

Figure 3: Same version of the algorithm and same parameterization as in Figure 2. The chart shows the distribution of all reassessed transactions as a moving average (size of sliding window = 150).

In the next step the algorithm is adapted to the application scenario by introducing a target variable and by including positive scores. First, this means that itemsets are filtered and only itemsets that contain a target label are kept: correlations among explaining attributes do not seem to help in identifying possibly wrong labels. Second, we apply the extended function violates, which includes positive scores. Finally, support and confidence thresholds are lowered to minsupp = 0.1‰ and minconf = 75% in order to create more rules. The results can be seen in Figure 4 and Figure 5. The fraction of mislabeled transactions among the first 150 records increases to over 40% and is almost 80% among the 20 transactions with the most negative score. What is more, the fraction of actually correct labels is very small among the first records and very large among the last ones (Figure 5).

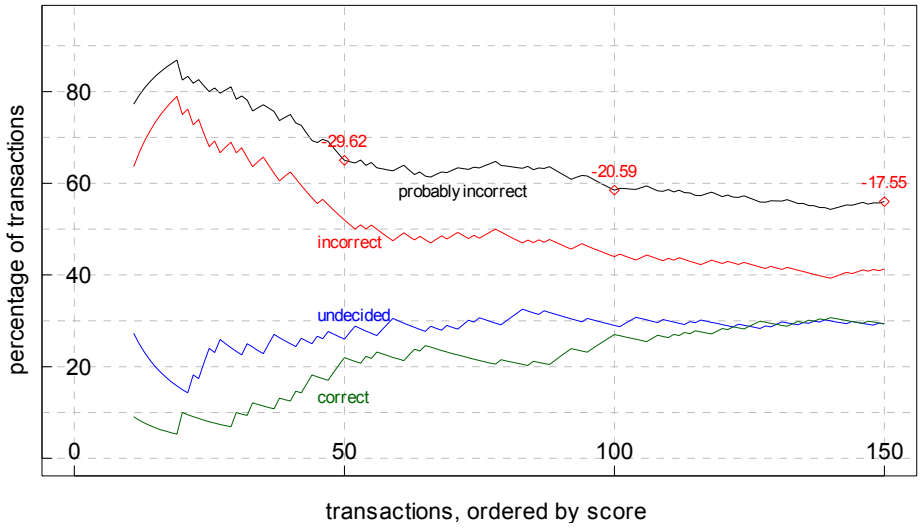

Figure 4: In this experiment the algorithm is adapted to include a target variable and positive scores. To create more rules, minsupp is lowered to 0.1‰ and minconf is set to 75%. The fraction of mislabeled transactions is almost 80% among the 20 transactions with the most negative score increases and does not fall below 40% within the first 150.

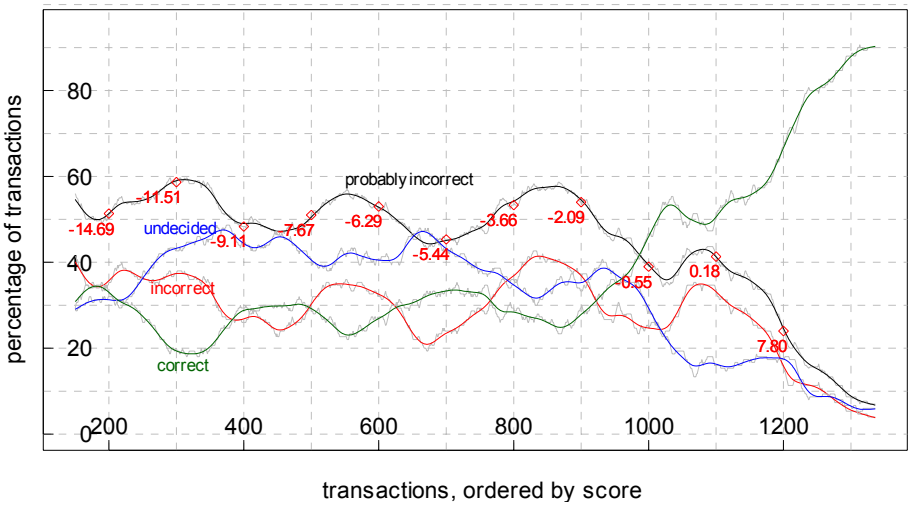

Figure 5: The chart is based on the same experiment as Figure 4. One can recognize the small fraction of correct transactions among these with the lowest scores and the strong increase of this percentage by the end of the list.

The extensions to reduce redundancy do not lead to the expected improvements. The best results are achieved applying an asymmetric distance measure that takes into account the number of transactions covered by each rule condition. One can observe that eliminating redundant rules leads to smoother curves. Nevertheless, further research will be needed to investigate the impact of redundancy on the scoring function.

Further Improvement

To improve our results, we extended the underlying data by adding descriptive attributes from various additional data sources. Finally, we were able to identify a promising new data source at the end of the underlying business process: Collecting and labeling the data as described in Section 2 is only the first part of the process. One of the downstream tasks requires that each transaction is priced semi-automatically. To exploit this additional expert information, we discretize these new values and insert them as descriptive attribute values. The results are shown in Figure 6 and Figure 7: The fraction of probably mislabeled transactions among the first 150 records increases up to 70% and is almost 90% among the 20 transactions with the most negative score. Besides, one can observe that the fraction of correct labels continuously increases among transactions with high scores.

Whereas the results show that our approach is promising and performs even better with more suitable data, it is important to note that in our application this especially valuable data is available only near the end of the business process chain. Measuring data quality that late might be *too* late and thus outweigh the gain in accuracy of the results. In our application, correcting the values would not be possible any more in many cases, which is reflected by the large percentage of transactions that are classified as "undecided" by our experts. Nevertheless, if the goal is to improve the data collection process instead of correcting single data values, more accurate results are still very useful.

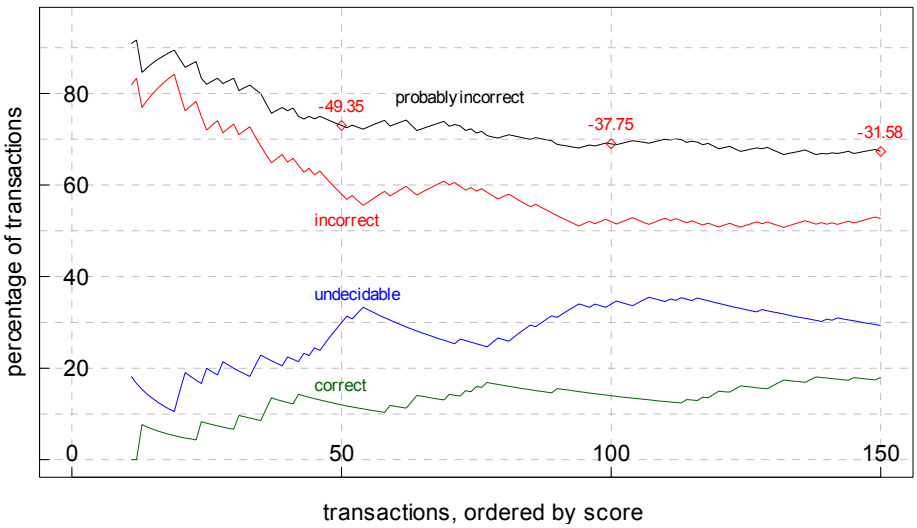

Figure 6: This chart shows the scoring of the 150 transactions with the lowest score after a pricing attribute that is assigned semi-automatically to each transaction downstream in the business process has been added as descriptive attribute.

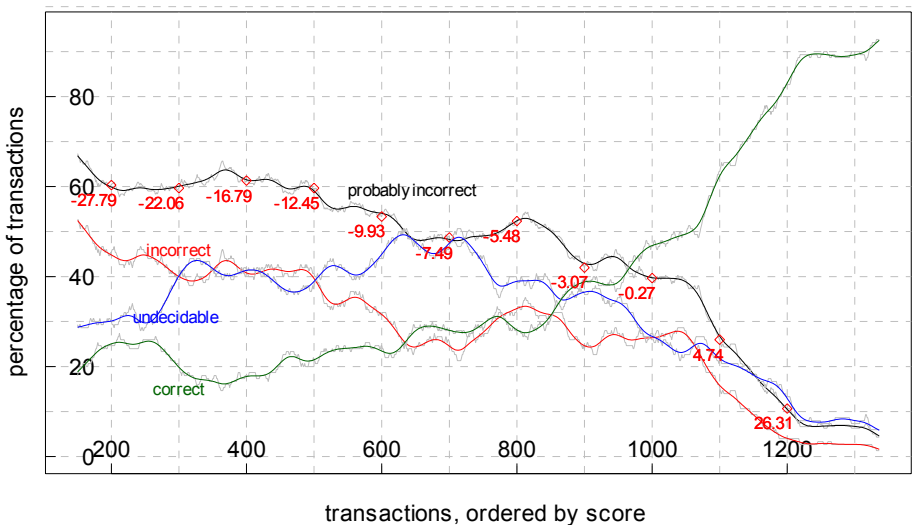

Figure 7: Resulting from the same experiment as the chart in Figure 6, this chart also shows the better quality of the score when adding variables that are available later in the business process and are strongly correlated to the label attribute.

Measuring Accuracy

The experiments and evaluations show that the sequence obtained from sorting the transactions based on the assigned scores actually orders the transactions by their likelihood of being incorrect or correct. Nevertheless, our initial goal was to go beyond pure sorting: We want to give the users percentages for "incorrect", respectively "correct" transactions in their data. For that purpose, we introduced a maximum threshold below zero indicating incorrectness and a minimum threshold above zero indicating correctness.

In Figure 8(a), the percentage of transactions classified as incorrect based on the score is shown for different threshold values. For example, if all transactions up to a maximum score of -1 are treated as incorrect, slightly above 6% of all transactions are covered, while applying a maximum score of -2 results in about 4.5% of the records being classified as "incorrect". Accordingly, Figure 8(b) shows the corresponding relation between the minimum score threshold and the percentage of transactions treated as correct when the threshold is applied. From the evaluations we learned that with increasing negative scores, the probability of a transaction being correct, is increasing. Although, not yet properly proven by the experiments, for decreasing positive scores one can expect an increasing probability for transactions being incorrect. So the problem is to deal with trade-off between high accuracy for a classification into "incorrect" and "correct" versus a large number of "undecided" transactions.

Figure 3 shows that a minimum threshold of -1 leads to about 50% of probably incorrect transactions in the result set. Such a high percentage of incorrect values is considered sufficient to treat this set of transactions as conspicuous and to present it to our domain experts for further investigation. For the positive threshold we do not have comparable results from experiments yet, but based on interviews with the experts we chose +0.5 as a rough estimation for the minimum threshold. The resulting measurement of data accuracy is shown in Figure 9(a). About 6.2% of the transactions are considered incorrect, 71.7% are classified as correct and the remaining 22.1% remain undecided. Relaxing the thresholds would reduce the number of undecided transactions, but at the same time, accuracy for both, incorrect and correct, would decrease. Figure 9(b) shows the absolute numbers of transactions that are classified "incorrect", "undecided", and "correct" for the 10 most frequent labels in the domain under consideration.

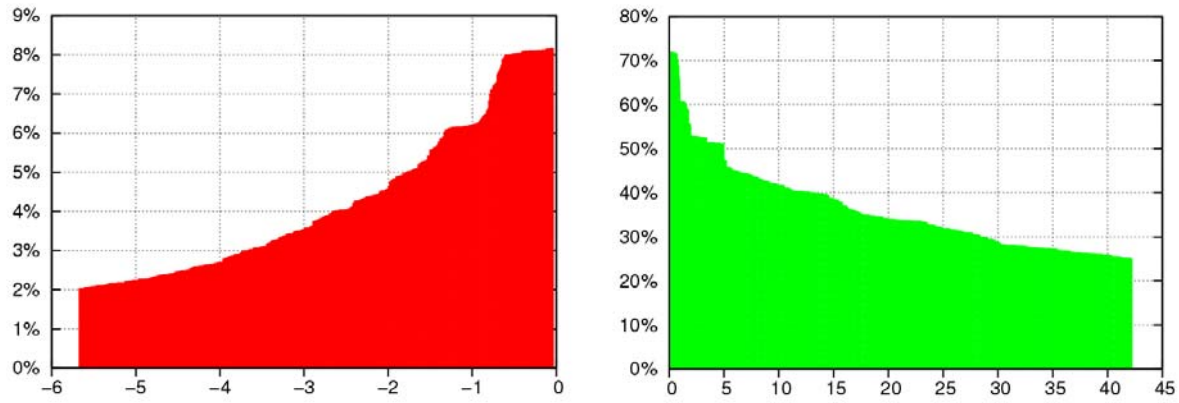

Figure 8: (a) Percentage of transactions classified as incorrect based on different negative thresholds. (b) Percentage of transactions classified as correct based on different positive thresholds.

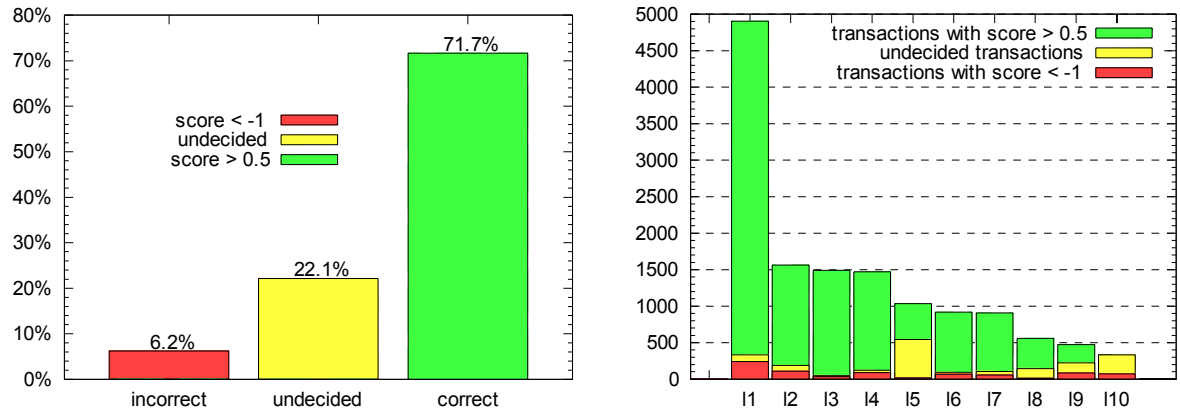

Figure 9: (a) Fraction of transactions that are considered incorrect, undecided, and correct when setting score thresholds to -1 and 0.5. (b) Classification results for the ten most frequent labels in the considered domain for the defined score thresholds -1 and 0.5.

We are aware that this can only be a first step towards an automated measurement of accuracy. Further research for calibrating the thresholds is necessary. For example, we learned that thresholds are not straightforward to be generalized across the domains inside our application, not to think of completely different application scenarios. Nevertheless, taken as a rough estimation and monitored over time, charts like the ones in Figure 9 proved to be very valuable.

CONCLUSION AND FUTURE WORK

In this paper we presented an approach for measuring the accuracy dimension of data quality based on association rules. The basic idea is to generate association rules from the data to be measured and take the resulting rule set as a model of normality. In a second step, the data is matched against this rule set and a score value is assigned to each record, respectively business transaction, in the data set. This score captures in how far the record is an outlier or is conform to the model of normality. Furthermore, we introduced thresholds that partition the data set based on the score into the three subsets "potentially incorrect", “undecided" and "probably correct".

A thorough evaluation of the approach showed that the ranking of transactions by their score indeed reflects the likelihood of these transactions being incorrect or correct. For the first time the approach from [5], which had yet been evaluated only on synthetic data, was extended to be applied and tested on real data. A key to the evaluation was that a domain expert helped us in identifying actual misclassification. For low scores our evaluation showed rates between 50% and up to 90% of incorrect values. With regard to the very low a priori probability of incorrectness, this is a very promising result. However, we do not have sufficient results to prove a low rate of incorrect transactions for high scores. Although we were not able to identify a single incorrect transaction for very high scores in first random samples, further research will be needed to evaluate the quality of high scores. There is no straightforward way to accomplish this due to the low rate of incorrect values among these transactions.

Apart from that, there are several other open issues that require future research and further evaluations: Our first results in applying redundancy filters that go beyond the application of closed itemsets were not very promising. However, we must evaluate whether eliminating redundancy is not useful in general, or just in our special case where very good results where already achieved by adding meaningful attributes to the dataset. Second, calibrating the score thresholds that determine correct and incorrect transactions together with the experts was very time-consuming. Finding a way to generalize thresholds between different applications would help a lot. In addition to improvements of the underlying algorithm, we are going to fundamentally extend our approach. Violated rules express that a certain value is expected, but is not found in the data. In other words, our rule based approach is able to predict correct values for incorrect transactions. We have not fully exploited this information yet, but based on first explorations we expect a high potential for automated or at least semi-automated correction of incorrect data.

REFERENCES

[1] Agrawal, R., Imielinski, T., and Swami, A.: Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (ACM SIGMOD '93)*, pages 207-216, Washington, USA, May 1993.

[2] Agrawal, R., and Srikant, R.: Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Databases (VLDB '94)*, pages 487-499, Santiago, Chile, June 1994.

[3] Borgelt, C., and Kruse, R.: Induction of association rules: Apriori implementation, In *Proceedings of the 15th Conference on Computational Statistics (Compstat 2002)*, Physica, Heidelberg, Germany 2002.

[4] English, L.P.: *Improving Data Warehouse and Business Information Quality: Methods for Reducing Costs and Increasing Profits. John Wiley & Sons, New York, USA, 1999.*

[5] Hipp, J., Güntzer, U., and Grimmer, U.: Data quality mining – making a virtue of necessity. In *Proceedings of the 6th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD 2001)*, pages 52-57, Santa Barbara, California, USA, May 2001.

[6] Hipp, J., Güntzer, U., and Nakhaeizadeh, G.: Algorithms for association rule mining – a general survey and comparison. SIGKDD Explorations, 2(1):58-64, July 2000.

[7] Hrycej, T., and Hipp, J.: Outlier detection by rareness assumption. In *GI Jahrestagung (1)*, pages 244-248, Ulm Germany, 2004.

[8] Huang, K., Lee, Y. W., Wang, R. Y.: *Quality Information and Knowledge*. Prentice Hall, 1999.

[9] Lee, Y. W., Strong, D. M., Kahn, B. K., and Wang, R. Y.: AIMQ: A Methodology for Information Quality Assessment. In: *Information & Management*. Vol. 40, Issue 2, pages 133-146, Dec. 2002.

[10] Olson, J.: *Data Quality – The Accuracy Dimension*. Morgan Kaufmann, 2002.

[11] Pasquier, N., Bastide, Y., Taouil, R., and Lakhal, L.: Pruning closed itemset lattices for association rules. In *Proceedings of the BDA French Conference on Advanced Databases*. Oct. 1998.

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

[13] Zaki, M. J., and Hsiao, C.: Charm: An efficient algorithm for closed itemset mining. In *Proceedings of the 2nd Siam International Conference on Data Mining*. Arlington, Virginia, USA, April 2002.