Expert Systems With Applications 201 (2022) 117132

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

# Dynamic repair of categorical data with edit rules

Antoon Bronselaer *, Toon Boeckling, Filip Pattyn

*Ghent University, Department of Telecommunications and Information Processing, St.-Pietersnieuwstraat 41, B-9000, Ghent, Belgium*

ARTICLE INFO

*Keywords:*
Data quality
Data repair
Edit rules

ABSTRACT

In this paper, a dynamic setting for data quality improvement is studied. In such a setting, there is a repeated search for data quality rules and a fix of their violations until stability is reached. The constraints considered here are simple constant edit rules and searching is done via association analysis. Repair of violations relies on the set cover method. This paper contributes to the field of data quality in three ways. First, it is shown that with appropriate filtering, association analysis is an appealing tool to discover data quality rules with high precision. Second, when edit rules are limited to logical implications such as association rules, then under reasonable circumstances, time complexity of rule implication reduces from exponential to quadratic. This result is formalized as the strong generator theorem. Third, a detailed analysis of data repair in a dynamic setting is provided and the conditions for termination are shown. Empirical results indicate that if the initial precision of rules is high, then repeated search-and-repair offers a boost in recall with a mitigated drop in precision.

## 1. Introduction

Over the past decades, there has been an increasing interest in techniques and mechanisms to safeguard data quality. Although data quality has many facets like accuracy, completeness and currency, the most prominent aspect is *consistency* (Batini et al., 2009; Batini & Scannapieco, 2006; Bronselaer et al., 2017; Fan & Geerts, 2012). Many approaches towards verification of consistency have adopted some kind of constraint (Abiteboul et al., 1995). Notable examples include functional dependencies (FDs) (Bohannon et al., 2005), conditional functional dependencies (CFDs) (Bohannon et al., 2007; Cong et al., 2007; Fan et al., 2008), inclusion dependencies (INDs) (Bohannon et al., 2005) and denial constraints (DCs) (Chu et al., 2013). A recent survey shows that there are many other constraints, conjointly referred to as *relaxed functional dependencies* (RFDs), that are being investigated and for which theory is being developed (Caruccio et al., 2016). In general, research regarding data consistency seems to focus on finding more expressive constraints.

An important issue with this trend is that constraints that are more expressive also tend to be more complex with regard to the study of their fundamental properties. In particular, *repair* of violations and *discovery* of constraints tends to be more complex. To resolve the violations against a *static* set of constraints, the Chase algorithm has been studied to create a search tree that models all repairs (Geerts et al., 2019). One problem with high expressive constraints like CFDs or DCs is that the number of branches of a search tree increases so fast that heuristics are required to keep searching feasible. Moreover, in a more *dynamic* setting, discovery of CFDs and DCs has turned out to be computationally intensive as well (Chu et al., 2013; Fan et al., 2009).

One way to mitigate the complexity of discovery and repair of constraints, is to consider *less complicated* formalisms. In a recent paper, it was argued that a significant amount of inconsistencies in real-life datasets can be captured by constant tuple-level constraints (Rammelaere & Geerts, 2019) and that more complicated constraints like FDs can often be modeled by sets of these constant tuple-level constraints. For such tuple-level constraints, simple and efficient algorithms exist to find them (Boeckling et al., 2019; Rammelaere & Geerts, 2019; Rammelaere et al., 2017). With respect to repair, minimal repairs can be found by searching for minimal set covers of failing constraints if the set of constraints satisfies some closure properties (Boskovitz, 2008; De Waal et al., 2011; Fellegi & Holt, 1976). Using this set cover method provides a huge advantage when searching for repairs because it allows to search in particular parts of the search space. This methodology is sometimes criticized because it may lead to repairs that are unlikely with respect to the distribution of the data (Hang et al., 2015). Such criticism is somehow obsolete for several reasons. First, empirical studies show that, in general, the set cover method is more robust when compared to other methods (De Waal et al., 2011). Second, under particular assumptions, minimal solutions lead to approximate

* Corresponding author.
*E-mail addresses:* antoon.bronselaer@ugent.be (A. Bronselaer), toon.boeckling@ugent.be (T. Boeckling), filip.pattyn@ugent.be (F. Pattyn).

https://doi.org/10.1016/j.eswa.2022.117132
Received 19 May 2021; Received in revised form 3 September 2021; Accepted 29 March 2022
Available online 18 April 2022

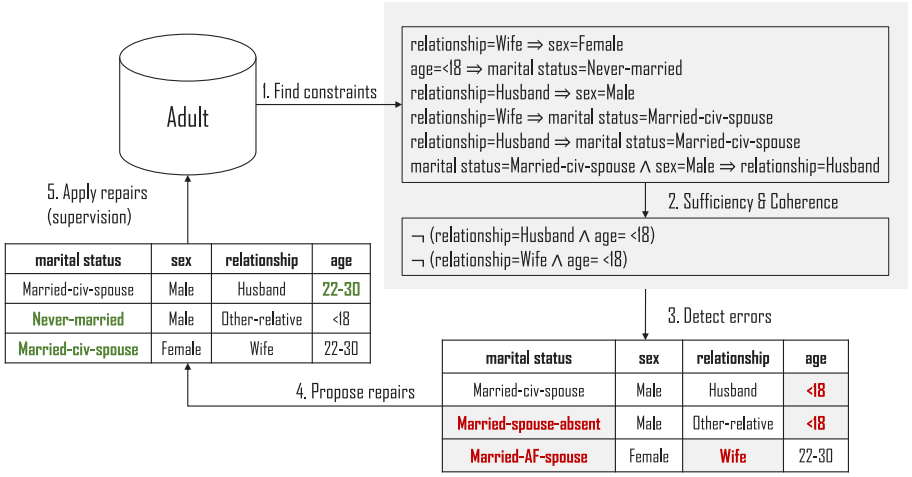



**Fig. 1.** Illustration of the dynamic repair methodology applied to the UCI Adult dataset, restricted to four attributes. Possible errors are indicated in bold red font and corrections are indicated in bold green font.

maximum likelihood estimates of consistent data, given some inconsistent data (Scholtus, 2014). Third, the general methodology can be extended to account for several very specific error mechanisms in the case of additive data (Scholtus, 2014). Examples of such mechanisms include typographical errors, rounding errors, value swaps and unit errors. Fourth, the methodology of minimal set covers can easily be extended to account for data distributions when making a selection among minimal repairs.

This paper investigates the use of tuple-level constraints to improve data quality in a dynamic manner, where constraints are constant edit rules. Fig. 1 illustrates the methodology on the Adult dataset[1] from the UCI repository, restricted to four attributes. In the first step of the approach, constraints are searched by means of association analysis. In a second step, the rules are inspected for a property called *sufficiency*. Informally, this step will search for any additional rules that are necessary for the set cover approach to work. In Fig. 1, two rules are added by this step. At the same time, this step also verifies the internal coherence of the rules to ensure they are not self-conflicting. In the third step, the data are inspected for possible errors by using the set cover method. With this method, minimal sets of attributes that cover all violated rules are composed. For example, the first tuple in Fig. 1 violates two rules, both of them involving attribute `age`, which is the only minimal set cover. Note here that one of the two failing rules was added during the verification of sufficiency. The second and third tuple violate a single rule that involves two attributes, so there are two possible locations for the error. After error detection is completed, a repair algorithm will propose repairs for the errors that were found and these corrections are then applied to the data. The final step is then to apply corrections on the original data. This is usually done in a supervised setting for several reasons. First of all, it might be the case that constraints found in the first step do not qualify as hard data quality rules. Violations of those rules are then outliers rather than errors. Second, a repair function is in general not deterministic and the choice it makes to correct an error, might be wrong. Although the solutions presented in this paper try to avoid these scenarios, they cannot be ruled out and supervision therefore remains a vital step in the entire dynamic procedure presented here. The methodology described here comes with three main problems that are studied throughout this paper:

1. If association analysis is used to search for edit rules, a high probability is required that an association rule is indeed a constraint. Choosing confidence high is a first step, but it is not always sufficient. In Fig. 1, This paper introduces two simple filters to ensure that the rules found are of sufficient quality in Section 3.
2. When applying the set cover method for repairing, the set of constraints needs to contain sufficiently many implicit rules (Fellegi & Holt, 1976). Construction of such a sufficient set is achieved by rule implication and in general, this is a problem with an exponential time complexity. It is shown in Section 4 that for implicative rules in general, time complexity reduces to *quadratic* under reasonable conditions. The construction of a sufficient set also ensures satisfiability of the set of constraints and this guarantees the existence of repairs.
3. Third, with a sufficient set of constraints at hand, violations of constraints can be repaired. However, doing so might introduce new association rules that are possibly violated. This cyclic process of search and repair is studied in Section 5 and the conditions for termination are shown. It can be readily seen that if precision of the initial rules is low, a propagation of errors can render the dynamic process useless. The filters introduced in this paper can help in increasing precision but they might not be sufficient and again, supervision is necessary to monitor this. Fortunately, empirical results indicate that the propagation of errors is low if the initial rules are of sufficiently high precision.

The remainder of this paper is organized as follows. In Section 2, some basic concepts and notations are recalled. In Sections 3, 4 and 5, the three main problems listed above are studied. In Section 6, the approach is positioned in the large body of literature on data quality. In Section 7, several empirical results are reported with respect to precision, computational complexity and logical coherence of the proposed approach. In addition, some results about the termination speed of repair steps are reported. Finally, in Section 8, the key findings of this paper are summarized.

## 2. Preliminaries

The relational model for databases considers a countable set of attributes $\mathcal{A}$. For each attribute $a \in \mathcal{A}$, the set $A$ denotes the *domain* of $a$, which is the set of permitted values an attribute $a$ can take. A *schema* $\mathcal{R} = \{a_1, \ldots, a_k\}$ is defined by a non-empty and finite subset of $\mathcal{A}$ where indices of attributes provide us with a suitable total order on the attributes. A *relation* $R$ with schema $\mathcal{R}$ is defined by a finite set $R \subseteq A_1 \times \cdots \times A_k$. Each element of a relation $R$ with schema $\mathcal{R}$ is called a *tuple* $r$ with schema $\mathcal{R}$. The combined universe $A_1 \times \cdots \times A_k$ will be denoted as $\mathrm{dom}(\mathcal{R})$ for short. For a relation $R$ with schema $\mathcal{R}$ and

[1] https://archive.ics.uci.edu/ml/index.php.

$X \subseteq \mathcal{R}$, the *projection* of $R$ over $X$ is defined by a relation $R[X]$ with schema $X$ that is obtained by retaining only the values of the attributes in $X$ for each $r \in R$ and by removing all duplicate tuples. The projection of a single tuple $r$ over $X$ is denoted by $r[X]$ and if $X = \{a\}$, the notation $r[a]$ is used. For a predicate $P$ defined on $\mathrm{dom}(\mathcal{R})$, the *selection* $P$ over a relation $R$ is defined by $R_P = \{r \mid r \in R \wedge P(r)\}$.

In their seminal paper on automated editing and imputation, Fellegi and Holt laid the foundations for the theory of statistical editing (Fellegi & Holt, 1976). For nominal scaled data where all domains $A_i$ are finite, an *edit rule* on a schema $\mathcal{R}$ is defined as a relation $E$ with schema $\mathcal{R}$ that can be written as the cross product of subsets of the attribute domains.

**Definition 1** (*Edit Rule*)**.** An edit rule $E$ on $\mathcal{R} = \{a_1, \ldots, a_k\}$ is an expression of the form $E_1 \times \cdots \times E_k$ where $E_i \subseteq A_i$. A tuple $r$ is said to *satisfy* $E$ if $r \notin E$ and to *fail* $E$ if $r \in E$. The set of all edit rules is denoted by $\mathbb{E}$.

If a tuple $r$ satisfies (resp. fails) a rule $E$, the notation $r \vDash E$ (resp. $r \nvDash E$) is used. For a set of rules $\mathcal{E}$, $r \vDash \mathcal{E}$ implies $r \vDash E$ for each $E \in \mathcal{E}$. Similarly, $r \nvDash \mathcal{E}$ means $r \nvDash E$ for at least one $E \in \mathcal{E}$.

**Example 1.** Consider the second rule shown in Fig. 1 which expresses that, if the value for attribute `age` equals '<18', then the value for attribute `marital status` must be equal to 'Never-married'. Logically, this is equivalent to the edit rule:

$$\{<18\} \times \overline{\{\text{Never-married}\}} \times A_{\text{relationship}} \times A_{\text{sex}}$$

where $\overline{\{\text{Never-married}\}}$ is the set of values *not* containing the value 'Never-married'. In the example of Fig. 1, this rule is failed by the first and second tuple.

If for some rule $E$ it holds that $\exists a_i \in \mathcal{R} : E_i = \emptyset$, then $E$ is always satisfied and $E$ is called a *tautology*.[2] In what follows, edit rules are assumed not to be tautologies. An attribute $a_i \in \mathcal{R}$ is said to *enter* an edit rule $E$ if and only if $E_i \subset A_i$. Alternatively, it is said that $E$ *involves* $a_i$. If an attribute $a_i$ is not involved in $E$, it follows that $E_i = A_i$ and satisfaction of $E$ does not depend on the value a tuple takes for $a_i$. The set of attributes involved in $E$ is denoted by $\mathcal{I}(E)$. If a tuple $r$ fails some rules in $\mathcal{E}$, a *solution* for $r$ is any set of attributes $S \subseteq \mathcal{R}$ for which different values can be assigned such that the new tuple fails no rules. In other words, a solution satisfies:

$$\exists r^* \in \mathrm{dom}(\mathcal{R}) : r^* \vDash \mathcal{E} \wedge r\left[\overline{S}\right] = r^*\left[\overline{S}\right] \tag{1}$$

where $\overline{S} = \mathcal{R} \setminus S$. The new tuple obtained from a solution is called a *repair*. A solution is called *minimal* if there are no solutions that contain fewer attributes. Finding minimal solutions can be done with the *set cover* method. Fellegi and Holt showed that if a tuple $r$ fails some rules in $\mathcal{E}$, any minimal solution is a minimal set cover of the failing rules, provided that $\mathcal{E}$ is *closed* under rule implication. This closure property is satisfied if $\mathcal{E}$ contains all *implied* rules.

**Definition 2** (*Implied Rule*)**.** For a set of edit rules $\mathcal{E}$ on $\mathcal{R}$ and an attribute $a_g \in \mathcal{R}$, let $E_i^*$ be defined by:

$$E_i^* = \begin{cases} \bigcap_{E \in \mathcal{E}} E_i & \textbf{if} \quad g \neq i \\ \bigcup_{E \in \mathcal{E}} E_i & \textbf{if} \quad g = i \end{cases} \tag{2}$$

for any $i \in \{1, \ldots, k\}$, then $E^*(g, \mathcal{E}) = E_1^* \times \cdots \times E_k^*$ is an *implied* edit rule with $a_g$ as *generator* and $\mathcal{E}$ as *contributing rules* if it is not a tautology.

[2] The original definition in Fellegi and Holt (1976) explicitly rejects tautologies.

**Example 2.** In Fig. 1, the contributing rules

$$\text{age} =< 18 \Rightarrow \text{marital status} = \text{Never-married} \tag{3}$$

$$\text{relationship} = \text{Husband} \Rightarrow \text{marital status} = \text{Married-civ-spouse} \tag{4}$$

lead to an implied rule:

$$\{<18\} \times A_{\text{marital status}} \times \{\text{Husband}\} \times A_{\text{sex}}$$

by using `marital status` as generator.

A problem with this approach is that finding *all* implied rules, is an exponential problem. Fortunately, the requirement of closure can be relaxed. First of all, implied rules are necessary only when they are *new*. An implied rule $E^*(g, \mathcal{E})$ is called new if it does not involve $a_g$ and if each contributing rule involves $a_g$. For a given set of rules $\mathcal{E}$, the set $\mathcal{E}$ together with all new rules is denoted by $\Omega(\mathcal{E})$ and this set is called *complete* (Fellegi & Holt, 1976).

**Example 3.** The implied rule from Example 2 was generated from two contributing rules in which the generator (`marital status`) was involved. The generator is not involved in the implied rule and therefore the implied rule is new.

Besides being not new, a rule can also be omitted if it is *dominated* by some other rule. For two edit rules $E$ and $E'$, $E$ dominates $E'$ if $E' \subseteq E$. A rule that is dominated by some other rule is called *redundant*. This leads to the notion of an NNR rule.

**Definition 3** (*NNR Rule*)**.** For a set of edit rules $\mathcal{E}$ on a schema $\mathcal{R}$ and a generator $a_g \in \mathcal{R}$, a rule $E^*(g, \mathcal{E})$ is called new and non redundant (NNR) if it is new and not redundant to any rule that can be implied from $\mathcal{E}$ with $a_g$ as generator.

Further restrictions can be applied in this rule generation process and the best known algorithm for rule implication is the Field Code Forest (FCF) algorithm (Boskovitz, 2008). This algorithm produces a set of rules denoted $\underline{\Omega}(\mathcal{E})$ that is shown to contain sufficiently many new rules to ensure the property that set covers are solutions and vice versa. For that reason, $\underline{\Omega}(\mathcal{E})$ is called a *sufficient* set of rules.

**Example 4.** In Fig. 1, the initial association rules combined with the two additional rules, form a sufficient set. The first tuple from Fig. 1 fails the following two rules:

$$\begin{array}{lllllll} \{<18\} & \times & \overline{\{\text{Never-married}\}} & \times & A_{\text{relationship}} & \times & A_{\text{sex}} \\ \{<18\} & \times & A_{\text{marital status}} & \times & \{\text{Husband}\} & \times & A_{\text{sex}}. \end{array}$$

Because the set of rules is sufficient and because attribute `age` is involved in both rules, the set {age} is a minimal set cover of the failing rules and therefore a minimal solution for this tuple.

## 3. From association rules to edit rules

In this section, the conditions under which *association rules* can serve as data quality constraints are investigated. After a brief refresher on association analysis, an association rule is first shown to be a special kind of edit rule and thus fit for the set cover methodology. Second, next to the obvious requirement that $\beta \approx 1$, it is shown that filtering strong rules based on ratios of confidence and support aids in obtaining a better quality of the rules.

*A brief refresher.* Association analysis is a branch from the field of data mining that focuses on (i) finding associations between the occurrence of *items* and (ii) finding suspected causal relationships between items (Agrawal et al., 1993). For a schema $\mathcal{R}$ and a relation $R$, a *frequent itemset* is a tuple $x$ with schema $\mathcal{X} \subseteq \mathcal{R}$ such that $\Pr[R_{\mathcal{X}=x}] \geq \alpha$ where

$$\Pr[R_{\mathcal{X}=x}] = \frac{|R_{\mathcal{X}=x}|}{|R|} \tag{5}$$

In association analysis, the estimate of probability is often referred to as the *support* of an itemset. An *association rule* is an expression of the form $x \Rightarrow y$, where $x$ and $y$ are tuples with resp. schemata $\mathcal{X} \subset \mathcal{R}$ and $\mathcal{Y} \subset \mathcal{R}$ such that $\mathcal{X} \cap \mathcal{Y} = \emptyset$. A *strong* rule is an association rule $x \Rightarrow y$ such that $xy$ is frequent and such that:

$$\frac{\Pr\left[R_{\mathcal{X}\mathcal{Y}=xy}\right]}{\Pr\left[R_{\mathcal{X}=x}\right]} \geq \beta. \tag{6}$$

The left hand side of this inequality is called the *confidence* of the rule $x \Rightarrow y$ (denoted by $\text{Conf}(x \Rightarrow y)$) and is an estimate for the conditional probability $\Pr[\mathcal{Y} = y \mid \mathcal{X} = x]$. For given $\alpha$ and $\beta$, the set of *all* strong rules (with at least one condition) that can be found in a relation $R$ is denoted by $\mathcal{E}_{\alpha,\beta}(R)$ or simple $\mathcal{E}_{\alpha,\beta}$ if $R$ is understood.

*Conversion to edit rules.* It is now shown how a set of strong rules $\mathcal{E}_{\alpha,\beta}$ is converted into an equivalent set of edit rules. Without loss of generality, strong rules in $\mathcal{E}_{\alpha,\beta}$ are assumed to have precisely one attribute in their conclusion.[3] As such, assuming there is a suitable permutation of attributes, a strong rule is a logical expression of the form:

$$a_1 = v_1 \wedge \cdots \wedge a_{|\mathcal{X}|} = v_{|\mathcal{X}|} \Rightarrow a_{|\mathcal{X}|+1} = v_{|\mathcal{X}|+1}. \tag{7}$$

By applying the rules of logic, satisfaction of this statement is equivalent to:

$$r \notin \{v_1\} \times \cdots \times \{v_{|\mathcal{X}|}\} \times \overline{\{v_{|\mathcal{X}|+1}\}} \times \cdots \times A_k \tag{8}$$

This latter expression is an edit rule $E$ on $\mathcal{R}$ that involves $|\mathcal{X}| + 1$ attributes. Note that, strictly speaking, the equivalence only holds when $a_{|\mathcal{X}|+1}$ is not NULL. Indeed, if $r[a_{|\mathcal{X}|+1}]$ is missing, the association rule fails if the left hand side holds. However, the edit rule is satisfied if $r[a_{|\mathcal{X}|+1}]$ is missing.

From the above given conversion, it can be seen that if there are two strong rules $x_1 \Rightarrow y$ and $x_2 \Rightarrow y$ such that $\mathcal{X}_1 \subseteq \mathcal{X}_2$ and $x_2[\mathcal{X}_1] = x_1$, then after conversion to edit rules, the first rule dominates the second one. In other words, redundancy of association rules translates to redundancy of edit rules. As a consequence, when generating $\mathcal{E}_{\alpha,\beta}$, it suffices to consider only those rules for which the left hand side is *minimal* in the sense of $\subseteq$.

The following characterization can now be given to edit rules produced from a set of strong rules.

**Proposition 1.** *For a relation $R$ with schema $\mathcal{R}$, an edit rule $E \in \mathcal{E}_{\alpha,\beta}$ satisfies:*

$$\forall a_i \in \mathcal{I}(E) : |E_i| = 1 \vee |E_i| = |A_i| - 1 \tag{A1}$$

$$\exists a_i \in \mathcal{I}(E) : |E_i| = 1 \tag{A2}$$

$$\exists a_i \in \mathcal{I}(E) : |E_i| = |A_i| - 1 \tag{A3}$$

$$|\{a_i \mid a_i \in \mathcal{I}(E) \wedge |E_i| > 1\}| \leq 1 \tag{A4}$$

In this characterization, A1 states that if an attribute is involved in an edit rule, the corresponding set of values for this attribute is either a *singleton* or a *singleton complement*. Hereby, sets that are singletons correspond to conditions from the left hand side of the original association rule while sets equal to singleton complements correspond to the right hand side of the original association rule. The singletons will be called *condition* sets and the singleton complements will be called *result* sets. Conditions A2 and A3 state that there is always at least one condition set and one result set. Condition A4 states that there is at most one involved attribute for which the corresponding set of values is not a singleton. It will be shown in Section 4 that this characterization is important when studying the construction of a sufficient set.

[3] It is easy to show that the more general case with $\ell$ attributes in the conclusion is equivalent to $\ell$ rules with a single-attribute conclusion.

*Ensuring constraint quality.* Under which circumstances is it viable to accept strong rules as data quality constraints? The main idea of creating these circumstances is very simple. Consider some association rules found with a certain support and with a *very high* confidence, close to 1. Arguably, because of the very high confidence, the few cases where the rule does *not* apply, *might* be due to errors in the database. Our argument thus hinges strongly on the assumption that strong rules with (very) high confidence are acceptable as data quality rules that need to hold.

One obvious possibility to make this assumption more viable, is to use additional measures of interest such as *lift* or *conviction* to prune itemsets (Brin et al., 1997). Although such additional constraints can help, it has been reported in Chiang and Miller (2008) that even if lift or conviction are used to prune itemsets, one still obtains an unreasonable amount of rules. Of course, it must be mentioned that the findings from Chiang and Miller (2008) are based on experiments where $\beta = 0.5$ and thus not really qualify for 'very high confident' rules. Nevertheless, empirical tests show that even with much higher values of $\beta$, many rules will still be found that do not really qualify as hard constraints. In the following, two common cases are identified where strong rules are falsely identified as a data quality rule and a simple solution for each case is provided.

*Dealing with non-informative conditions.* The first problem is that conditions of a rule often have no significant contribution in explaining the right hand side of the rule. As an example of this problem, with $\alpha = 0.01$ and $\beta = 0.99$, the following rule is strong in the 'Adult' dataset: occupation = Adm-clerical $\wedge$ age = 22-30 $\wedge$ relationship = Not-in-family $\Rightarrow$ income = LessThan50K. The confidence of this rule is 0.9926. However, if the condition on `relationship` is removed, a rule with confidence of about 0.949 is obtained, indicating that this condition has little contribution in the statement that `income` needs to take a specific value. Additionally, removing the condition on `age`, produces a rule with confidence of about 0.86. Semantically, none of these rules are hard constraints and should therefore not be accepted as data quality rules. What happens here is that, because of the relatively low value for $\alpha$, conditions are added until sufficient confidence is reached. However, if $\alpha$ is increase, many correct rules would not be found.

To avoid this problem, it is proposed to accept an association rule $x \Rightarrow y$ only if each condition in $x$ contributes sufficiently to the confidence of the rule. When doing so, the fact that confidence is not necessarily monotone in the $\subseteq$-relation of the left hand side should be accounted for. To measure the contribution of conditions, the ratio of the confidence of $x \Rightarrow y$ and the confidence of the rule without those conditions is computed.

**Definition 4.** For a schema $\mathcal{R}$ and an association rule $x \Rightarrow y$ with $\mathcal{X}$ not empty, the confidence ratio of a set of attributes $\emptyset \subset X' \subseteq \mathcal{X}$ is defined by:

$$\frac{\text{Conf}(x \Rightarrow y)}{\text{Conf}(x[\mathcal{X} \setminus X'] \Rightarrow y)} \tag{9}$$

The rationale behind the confidence ratio is easy to explain. If the rule $x[\mathcal{X} \setminus X'] \Rightarrow y$ has a confidence that is close to the confidence of rule $x \Rightarrow y$, then the conditions in $X'$ have little or no value in drawing the conclusion $y$. The higher the confidence ratio gets, the bigger the leap in confidence by adding specific conditions to the left hand side. Good rules are rules that have a confidence ratio that is sufficiently high with respect to *all* its sub-rules. More formally, in addition to the requirements on support and confidence, an association rule $x \Rightarrow y$ should be accepted only if it satisfies:

$$\forall X' \subseteq \mathcal{X} : X' \neq \emptyset \wedge \frac{\text{Conf}(x \Rightarrow y)}{\text{Conf}(x[\mathcal{X} \setminus X'] \Rightarrow y)} \geq \sigma \tag{10}$$

where $\sigma \geq 1$ is a predefined threshold. Rules that satisfy the above test are called $\sigma$-strong rules and it is shown in the following that they

satisfy appealing properties. First, if $\sigma = 1$, then any strong rule is $\sigma$-strong. In other words, for $\sigma = 1$, the association rules with support and confidence above resp. $\alpha$ and $\beta$ are obtained. Second, $X' = \mathcal{X}$ is allowed in the test. In that case $\text{Conf}\left(x[\mathcal{X} \setminus X'] \Rightarrow y\right) = \Pr\left[R_{y=y}\right]$. Plugging this into the above inequality yields the requirement that the *lift* of rule $x \Rightarrow y$ must be greater than or equal to $\sigma$. Hence, the confidence ratio is a natural extension of the concept of lift. Third, because the test requires that the confidence ratio is at least $\sigma$ for all $X'$, a strong rule must satisfy

$$\forall a \in \mathcal{X} : \frac{\text{Conf}\,(x \Rightarrow y)}{\text{Conf}\,(x[a] \Rightarrow y)} \geq \sigma. \tag{11}$$

This gives rise to the following pruning strategy.

**Proposition 2.** *For a schema $\mathcal{R}$ and an association rule $x \Rightarrow y$ with $\mathcal{X}$ not empty, if there exists an $a \in \mathcal{X}$ such that* $\text{Conf}\,(x[a] \Rightarrow y) \cdot \sigma > 1$ *then $x \Rightarrow y$ is not a strong rule.*

The idea of this pruning strategy is that calculating the confidences for rules where both left hand side and right hand side contain only one attribute can be done very fast.

*Dealing with rare values.* The second problem addressed here, is the case where exceptions to a very high confident rule are not errors, but simply very rare cases. An example is the following rule from Fig. 1: relationship = Wife ⇒ marital status = Married-civ-spouse. At first glance, this rule seems correct, were it not that there are two values for `marital status` that are exceptional, but correct, given that the value of `relationship` is 'Wife'. These value are 'Married-AF-spouse' and 'Married-spouse-absent', which are shown in Fig. 1. What is observed here, is the undesirable phenomenon where a high confident rule pushes out some very infrequent values. To avoid such rules $x \Rightarrow y$ from being selected, consider the ratio:

$$\frac{\Pr\left[R_{\mathcal{X}\mathcal{Y}=xz}\right]}{\Pr\left[R_{\mathcal{Y}=z}\right]} \tag{12}$$

for each $z \neq y$ but with schema $\mathcal{Y}$. If there exists one $z$ for which this ratio is sufficiently high (i.e., above a threshold $\epsilon$), the rule $x \Rightarrow y$ is rejected. The rationale behind this decision is the following. Suppose $x \Rightarrow y$ is accepted as a rule, then each tuple that satisfies $\mathcal{X}\mathcal{Y} = xz$ is considered erroneous. Now suppose these errors are repaired by changing the value for $\mathcal{Y}$, then clearly the tuple must take a value different from $z$ after the repair. In that case, Eq. (12) is a lower bound to the error ratio for value $z$ in attribute $\mathcal{Y}$. If that lower bound is high, it is implicitly assumed that there is a strong bias in the occurrence of errors, because in a large proportion of the cases where $\mathcal{Y} = z$, a single rule indicates that this value is wrong. This conflicts with the usual assumption that errors are distributed randomly across data. It is worthwhile mentioning that Eq. (12) represents the confidence of the rule $z \Rightarrow x$, which is in fact conflicting with $x \Rightarrow y$. So the decision to reject $x \Rightarrow y$ can also be justified by the fact that there is a conflicting rule with a reasonably high confidence but low support.

## 4. Sufficiency and coherence

In the previous section, the conversion of strong rules into edit rules is discussed with the aim to repair violations by using the set cover method (Fellegi & Holt, 1976). However, to do so, it must be ensured that the converted set of edit rules, is sufficient. Despite the powerful FCF algorithm (Boskovitz, 2008), this can still be a computational intensive task. For that purpose, the properties of implication are investigated in the case where edit rules are derived from strong rules. In addition, it is investigated whether the rules are coherent. During our analysis of sufficiency and coherence, no assumptions on parameters $\sigma$ and $\epsilon$ are made for the sake of generality. As such, the set of rules produced for a relation $R$ is completely determined by $\alpha$ and $\beta$ and the notation $\mathcal{E}_{\alpha,\beta}$ will be used.

### 4.1. Sufficiency

In general, a set $\mathcal{E}_{\alpha,\beta}$ has no guarantee to be sufficient. In fact, in the general case, it cannot be ensured that any of the conditions A1-A4 are transferred to implied rules. However, under reasonable circumstances, implication of rules has some appealing properties. The key understanding in all this, is that there is a special type of generator that is of interest here. This type of generator is called a *strong* generator and is defined as follows.

**Definition 5.** For a set of edit rules $\mathcal{E}$ on $\mathcal{R}$, an attribute $a_g \in \mathcal{R}$ is called a *strong generator* for $\mathcal{E}$ if $E^*(g, \mathcal{E})$ is an NNR rule and if in addition:

$$\exists E \in \mathcal{E} : \exists v_g \in A_g : E_g = \overline{\{v_g\}}. \tag{13}$$

In the general case, the following proposition holds for strong generators.

**Proposition 3.** *For a set of edit rules $\mathcal{E}$ on schema $\mathcal{R}$, if $|\mathcal{E}| \geq 3$ then $a_g \in \mathcal{R}$ is not a strong generator for $\mathcal{E}$.*

The consequence of Proposition 3 is that if $\mathcal{E}$ is used to imply a new rule with a strong generator, then exactly two rules are needed as contributors. Using more than two rules necessarily produces a redundant rule and is therefore of no interest. With Proposition 3 established, strong generators can be investigated in case of association rules and it can be verified which conditions (A1-A4) are preserved. To begin with, each rule in $\mathcal{E}_{\alpha,\beta}$ involves at least one attribute that meets Eq. (13), so strong generators appear quite naturally in the setting of association rules. Now suppose some attribute $a_g$ produces an NNR rule but is not a strong generator. Then the following holds.

**Proposition 4.** *For a relation $R$ with schema $\mathcal{R}$ and a set of edit rules $\mathcal{E} \subseteq \mathcal{E}_{\alpha,\beta}$ on $\mathcal{R}$, if $E^*(g, \mathcal{E})$ is an NNR rule and $a_g$ is not a strong generator, then $a_g$ enters in each contributing rule with a singleton set.*

As such, *weak* generation on the rules that were initially found corresponds to the case where all contributing rules have a singleton for the generator. Note that if $a_g$ satisfies $|A_g| = 2$, $a_g$ is always a strong generator for implication of NNR rules. If some attribute $a_g$ can be used to generate NNR rules but $a_g$ is not a strong generator, the rule set is said to *feature weak generators*. It can readily be seen that examining the initial set $\mathcal{E}_{\alpha,\beta}$ for weak generators is a simple operation that has a worst-case time complexity of $\mathcal{O}\left(|\mathcal{R}| \cdot |\mathcal{E}_{\alpha,\beta}|\right)$. Moreover, choosing $\alpha$ sufficiently high guarantees that no weak generators are featured.

**Proposition 5.** *Let $R$ be a relation with schema $\mathcal{R}$. A set of rules $\mathcal{E}_{\alpha,\beta}$ features no weak generators if $\alpha > \max_i |A_i|^{-1}$*

Proposition 5 provides a strict lower bound on $\alpha$ in order to avoid weak generators in $\mathcal{E}_{\alpha,\beta}$. There is an intuitive reason why this lower bound is interesting. Suppose a dataset in which attribute values are uniformly distributed over tuples in a relation. In that case, each value for attribute $a$ has a probability of $|A|^{-1}$ and choosing $\alpha$ greater than this probability implies patterns in the data that are more frequent than in a pure random setting. Two important properties of strong generators are now shown. The first one deals with the preservation of conditions A1, A2 and A4.

**Proposition 6.** *For a relation $R$ with schema $\mathcal{R}$ and a set of two rules $\mathcal{E}$ on $\mathcal{R}$ that satisfy A1, A2 and A4, then if $a_g \in \mathcal{R}$ is a strong generator for $\mathcal{E}$, $E^*(g, \mathcal{E})$ satisfies A1, A2 and A4.*

Proposition 6 states that if a strong generator is used, then A1, A2 and A4 are transferred to NNR rules. This means that for each $\mathcal{E}_{\alpha,\beta}$, if NNR rules are implied that are in their turn used as contributing rules, then conditions A1, A2 and A4 remain valid for all implied NNR rules,

as long as strong generators are used. In general, A3 is not preserved. However, if a strong generator is used, then because of Proposition 3, there are basically two types of implication that are possible on some set $\mathcal{E}_{\alpha,\beta}$.

- **Transitive implication (TI)** If one contributing rule involves the generator with a singleton and the other involves the generator with a singleton complement, then the NNR rule satisfies A1-A4. Rule implication is then equivalent to *transitivity* in the sense of association rules.
- **Non contradiction implication (NCI)** If both contributing rules involve the generator with a singleton complement, A3 is not preserved. Rule implication is then an application of the law of non contradiction.

Note that, unless data are binary, an implied rule obtained via NCI can only be used as contributing rule in combination with either an original rule or an implied rule obtained via TI.

The second important property deals with the fact that strong generators will not introduce any singleton sets that were not already present in some initial edit rule.

**Proposition 7.** *For a relation $R$ with schema $\mathcal{R}$ and a set of two rules $\mathcal{E} \subseteq \mathcal{E}_{\alpha,\beta}$ on $\mathcal{R}$, if $a_g \in \mathcal{R}$ is a strong generator for $\mathcal{E}$, then* $\left(\forall a_i \in \mathcal{I}\left(E^*\left(g,\mathcal{E}\right)\right) : |E_i^*| = 1\right) \Rightarrow \left(\exists E \in \mathcal{E} : E_i^* = E_i\right)$

Although Proposition 7 only applies to subsets of $\mathcal{E}_{\alpha,\beta}$, the proof (Appendix) only requires rules to satisfy conditions A1 and A4. Hence, as long as new rules satisfy conditions A1 and A4 (and they do because of Proposition 6), the result of Proposition 7 can be extended to the entire process of rule implication. The following important result can now be shown.

**Theorem 1** (*Strong Generator Theorem*). *Let $R$ be a relation with schema $\mathcal{R}$. If the set of edit rules $\mathcal{E}_{\alpha,\beta}$ on $\mathcal{R}$ does not feature weak generators, then $\underline{\Omega}\left(\mathcal{E}_{\alpha,\beta}\right)$ is generated by using strong generators only.*

The crux of Theorem 1 is the following. If association rules are produced from a relation $R$ and converted into a set of edit rules $\mathcal{E}_{\alpha,\beta}$ that features no weak generators, then no weak generators can occur during the entire implication process. An important consequence of the Strong Generator theorem is captured in the following corollary.

**Corollary 1.** *Let $R$ be a relation with schema $\mathcal{R}$. If $\mathcal{E}_{\alpha,\beta}$ features no weak generators, generating all NNR rules with generator $a_g$ has a worst-case time complexity of $\mathcal{O}\left(|\mathcal{E}_{\alpha,\beta}|^2\right)$.*

In general, finding all NNR rules with generator $a_g$ and $\mathcal{E}$ as possible contributing rules is an exponential problem in the worst case because all subsets of $\mathcal{E}$ need to be examined (see Boskovitz (2008), Garfinkel et al. (1986)). However, if $a_g$ is a strong generator, it suffices to examine only couples of rules instead of subsets of rules. Because the initial lack of weak generators is propagated throughout the entire procedure, Theorem 1 provides us with a tremendous gain in computational complexity. To conclude this section, the rather strict condition where $\mathcal{E}_{\alpha,\beta}$ is sufficient by itself, is provided.

**Proposition 8.** *If $\alpha > 0.5$, then $\mathcal{E}_{\alpha,1}$ is sufficient.*

In this proposition, the initial set is sufficient because there are no implied rules to generate. In a more general case, finding conditions where implied rules are necessarily part of the initial set, is hard because it is possible that there are two contributing rules that imply a rule which has a confidence below the initial threshold $\beta$.

### 4.2. Coherence

In order to find repairs for violated edit rules, it must be certain that these rules can be satisfied. In the case of edit rules, there are several notions of coherence that can be used. The following notion of coherence is for example coined in Fellegi and Holt (1976).

**Definition 6** (*Coherence*). A set of edit rules $\mathcal{E}$ on $\mathcal{R}$ is coherent if $\forall E \in \underline{\Omega}(\mathcal{E}) : |\mathcal{I}(E)| > 1$.

Simply put, a set of rules is coherent if the sufficient set contains only rules that involve at least two attributes. A second notion of coherence that is often considered is *satisfiability*.

**Definition 7** (*Satisfiability*). A set of edit rules $\mathcal{E}$ on $\mathcal{R}$ is satisfiable if $\exists r \in \mathrm{dom}(\mathcal{R}) : r \vDash \mathcal{E}$.

It is not difficult to show that a set of edit rules $\mathcal{E}$ on $\mathcal{R}$ is satisfiable if and only if $\forall E \in \underline{\Omega}(\mathcal{E}) : |\mathcal{I}(E)| > 0$. A first consequence of this is that any set of rules that is coherent, is also satisfiable. A second and more important consequence of satisfiability is actually tested during the construction of $\underline{\Omega}(\mathcal{E})$. Besides the above two notions of coherence, a third kind of coherence is introduced here.

**Definition 8** (*Weak Coherence*). A set of edit rules $\mathcal{E}$ is weakly coherent if and only if $\forall \mathcal{E}^* \subseteq \mathcal{E} : \forall a_g \in \mathcal{R} : |\mathcal{I}(E^*(g,\mathcal{E}^*))| > 1$.

Weak coherence implies that, in the first iteration of rule implication, there should be no incoherent rules. The notion of weak coherence is mostly useful to formalize some intuitive choices for parameters $\alpha$ and $\beta$. Thereby, it aims to avoid some 'obvious' mistakes in the set of rules. The main result in this regard is summarized in the following proposition.

**Proposition 9.** *A set of rules $\mathcal{E}_{\alpha,\beta}$ is weakly coherent if $\alpha > \frac{1}{3}$ and $\beta > \frac{1-\alpha}{2\alpha}$.*

This proposition provides a monotonically decreasing upper bound in terms of increasing $\alpha$ where $\alpha = 0.5$ is a special boundary case in the sense that the upper bound equals 0.5. Rules found under these conditions are always ensured to have confidence above 0.5 and are ensured to have strong generators. Despite this result, there is no general guarantee for coherence or satisfiability. This calls for strategies to deal with incoherence. One simple strategy is to systematically increase $\alpha$ and $\beta$ to remove rules and test whether the newly obtained set is still incoherent. Association rules need not be recomputed: the initial (and incoherent) set $\underline{\Omega}\left(\mathcal{E}_{\alpha,\beta}\right)$ can be used as a set of candidates. Any incoherence can be traced back to a proper subset of $\mathcal{E}_{\alpha,\beta}$ and it is easy to compute the minimal increase in $\alpha$ and/or $\beta$ to remove the cause of the incoherence. Another strategy would be to supervise the procedure and let human agents decide which of the contributing rules are invalid. If possible, this second strategy is preferable as the importance of human supervision during discovery of edit rules was already established in Rammelaere et al. (2017) and Rammelaere and Geerts (2019).

## 5. Repair

*Repair functions.* In this section, the last of three problems laid out in the introduction of this paper, is studied: repairing violations. Hereby, the sufficiency of the rules can be exploited to use the set cover method, which allows to find solutions as set covers. The actual modification of the inconsistent tuple is done by a repair function.

**Definition 9** (*Repair Function*). For a schema $\mathcal{R}$ and a set of edit rules $\mathcal{E}$ on $\mathcal{R}$, a repair function is defined by a function $\mathrm{Rep}_{\mathcal{E}} : \mathrm{dom}(\mathcal{R}) \to \mathrm{dom}(\mathcal{R})$ such that $\forall r \in \mathrm{dom}(\mathcal{R}) : \mathrm{Rep}_{\mathcal{E}}(r) \vDash \mathcal{E}$.

If $\mathcal{E}$ is clear from the context, $\text{Rep}_{\mathcal{E}}$ is denoted by Rep. A repair function is said to be *minimal* if it always utilizes minimal solutions. As a consequence, a minimal repair function corresponds to identity (i.e., $\text{Rep}(r) = r$) whenever $r \models \mathcal{E}$. Therefore, for a relation $R$, a (minimal) repair corresponds to application of a (minimal) repair function on those tuples that fail one or more rules in $\mathcal{E}$. The repair of a relation is denoted by $\text{Rep}(R)$.

In general, any repair function that keeps values for $\overline{S}$ must select values for attributes in $S$ from the set of permitted tuples $\mathcal{P}_S$. To define this set, only those rules from $\mathcal{E}$ must be considered that are not a priori satisfied after changing the values $S$. These are the rules for which the value of any attribute in $\overline{S}$ lies in the corresponding $E$-sets of the rule. Any other rule will never be failed when changing only the values of the attributes in $S$. Therefore, for any $S = \{a_1, \ldots, a_s\}$, the set of triggered rules is defined by $\mathcal{E}_{S,r} = \left\{ E \mid E \in \mathcal{E} \wedge \forall a_i \in \overline{S} : r[a_i] \in E_i \right\}$. With this notation at hand, the following result can be provided.

**Theorem 2.** *For a schema $\mathcal{R}$ and a set of edit rules $\mathcal{E}$ on $\mathcal{R}$, let $r$ be a tuple that fails $\mathcal{E}$ and let $S = \{a_1, \ldots, a_s\}$ be a minimal solution. Any minimal repair function* Rep *that satisfies* $\text{Rep}(r)\left[\overline{S}\right] = r\left[\overline{S}\right]$ *must also satisfy:*

$$\text{Rep}(r)[S] \in \overline{r[a_1]} \times \cdots \times \overline{r[a_s]} \setminus \bigcup_{E \in \mathcal{E}_{S,r}} E_1 \times \cdots \times E_s \tag{14}$$

*where* $\overline{r[a_i]} = A_i \setminus \{r[a_i]\}$.

Theorem 2 is important for several reasons. First, it provides us with a simple formalism to describe all permitted combinations of values for $S$ that $r$ is allowed to take, given $\mathcal{E}$ and a minimal solution $S$. This set of permitted tuples will be denoted by $\mathcal{P}_S$ or simply by $\mathcal{P}$ if $S$ is understood from the context. With additional information such as observed frequencies for each of the permitted value combinations, be it in the entire population or in a stratum, a motivated choice can be made from $\mathcal{P}$ to guide repairing. Second, because $\mathcal{P}$ models all possible repairs, it generalizes the two repair strategies provided by Fellegi and Holt (1976). In fact, it generalizes any minimal repair strategy as soon as the choice for $S$ has been set. Hence, Theorem 2 allows us to talk about repairing in very general terms. With the notions of repair functions and permitted tuples set, two important results of repairing in the case of association rules can be shown. The first one deals with *deductive repairing* and the second one with the dynamic behavior of repairing.

*Deductive repairs.* A tuple $r$ can be repaired *deductively* if there exists a minimal solution $S$ such that $\mathcal{P}_S$ is a singleton (De Waal et al., 2011). This means that there is only one possible way of repairing it. Deductive repairs are of interest because they are often considered as preferential over other repairs. In general, they are not easy to guarantee (De Waal et al., 2011) but when using strong rules, some interesting properties hold.

**Proposition 10.** *For a schema $\mathcal{R}$ and a set of rules $\underline{\Omega}(\mathcal{E})$ on $\mathcal{R}$, a tuple $r$ that fails all rules $\mathcal{E}^F \subseteq \underline{\Omega}(\mathcal{E})$ has a deductive repair if there exists a minimal solution $S$ such that $\forall E \in \mathcal{E}^F : \exists a_i \in S : |\overline{E_i}| = 1$.*

In words, the proposition states that there exists a deductive repair for $r$ if there is a minimal solution, such that for any rule that is failed, there is an attribute in the solution that enters this rule with a singleton complement. The rationale of the proposition is that, if the requirement of the proposition is true, there is no reason for attributes that enter failing rules with a set other than a singleton complement, to be part of the solution. There are some obvious situations in which this proposition applies. First and foremost, if $\mathcal{E}^F$ contains only one rule, then the proposition applies and the attribute from the right hand side can be imputed. Second, if for all rules in $\mathcal{E}^F$, the sets of involved attributes do not mutually overlap, the proposition again applies. Moreover, the restriction on overlap can be loosened in the following way. If for any pair of failing rules $(E, E')$ it holds that $\exists a_i \in \mathcal{I}(E) \cap \mathcal{I}(E') : |\overline{E_i}| = 1$, then the proposition applies and there exists a solution that leads to a deductive repair. These conditions are often (but not always) reasonable and achievable.

**Example 5.** In Fig. 1, the second tuple fails a single rule and there are two minimal covers for this rule. A deductive repair emerges by enforcing the consequent of the rule and change the value of `marital status` to 'Never-married'.

*Dynamic repairing.* The process of (i) finding and filtering strong rules, (ii) constructing a sufficient set and (iii) applying a repair function on all tuples that contain violations, is called a *repair step* and is denoted by $\rho$. The output of this process is a relation $\rho(R)$ in which all violations of rules found at the beginning, are repaired. After a repair step, however, new strong rules may occur that are possibly again violated. At this point, a second repair step can be applied and this continues until stability is reached. It is shown next that the characterization of this dynamic process depends solely on which rules are found after a repair step has been terminated. The following definitions are first provided.

**Definition 10.** A repair step $\rho$ is *upper preservative* if $\forall R : \mathcal{E}_{\alpha,\beta}(\rho(R)) \subseteq \mathcal{E}_{\alpha,\beta}(R)$.

**Definition 11.** A repair step $\rho$ is *lower preservative* if $\forall R : \mathcal{E}_{\alpha,\beta}(R) \subseteq \mathcal{E}_{\alpha,\beta}(\rho(R))$.

In words, $\rho$ is upper preservative if no new rules are found in consecutive repair steps and lower preservative if rules found in a previous step are found again in a next step.

**Definition 12.** A repair step $\rho$ is *finite* if, for any $R$ with schema $\mathcal{R}$ there exists an $n \in \mathbb{N}$ such that either $\rho^n(R) \models \mathcal{E}_{\alpha,\beta}(\rho^n(R))$ or $\mathcal{E}_{\alpha,\beta}(\rho^n(R))$ is not satisfiable. Hereby, $\rho^n = \underbrace{\rho \circ \ldots \circ \rho}_{n}$.

The following result can be shown.

**Proposition 11.** *A repair step $\rho$ that is lower or upper preservative, is finite.*

So, in order to ensure termination of dynamic repairing, it suffices to show either lower or upper preservation. At this point, strong rules pose an interesting problem. First of all, if a repair function is applied, new positive correlations can occur and thus new association rules are often found after repairing a relation. Upper preservation is thus often not ensured. Lower preservation is also not guaranteed, but failure of it is bound by quite strong conditions. Indeed, failing lower preservation means that the support and/or confidence of some rules is decreased by the repair function Rep. Such a decrease can however not occur for an association rule that fails some tuple $r$ after repairing $r$.

**Proposition 12.** *Consider a relation $R$, rules $\mathcal{E}_{\alpha,\beta}$ and a tuple $r \in R$ that fails all rules $\mathcal{E}^F \subseteq \underline{\Omega}(\mathcal{E}_{\alpha,\beta})$. For any $E \in \mathcal{E}^F \cap \mathcal{E}_{\alpha,\beta}$, its confidence must increase and its support will not decrease after repairing $r$.*

Because of Proposition 12, when repairing a violation of a rule, it still has sufficient support and confidence. In other words, repairing tuples can only negatively affect support or confidence of rules if rules are *satisfied* by tuples under repair. At the same time, it is known that repairing a tuple with a minimal repair function will affect attributes that occur in failing rules. That means, in order for support and confidence of a rule to be negatively affected, that rule must share some attributes with rules that fail. This leads to the definition of the concept of *rule competition*.

**Definition 13** (*Competing Rules*)**.** For a schema $\mathcal{R}$, a rule $E$ is *competing* with $E'$ if repairing a violation of $E'$ can decrease the support or confidence of $E$.

If $E$ is competing with $E'$, then after a certain number of repairs, it is possible that support (resp. confidence) of $E$ drops below $\alpha$ (resp. $\beta$). Whenever this happens, $E$ is said to be *invalidated*. The question is now under which conditions competition can occur. These conditions are characterized in the following theorem.

**Theorem 3.** *For a schema $\mathcal{R}$, a rule $E = x \Rightarrow y$ is competing with $E' = x' \Rightarrow y'$ if and only if (i) $\mathcal{X} \cap \mathcal{I}(E') \neq \emptyset$ and (ii) $\forall a \in \mathcal{I}(E)$, it holds that $a \in \mathcal{X}' \Rightarrow xy[a] = x'[a]$ and $a \in \mathcal{Y}' \Rightarrow xy[a] \neq y'[a]$.*

The characterization of rule competition from Theorem 3 comes with a few insights. First and almost trivially, $E$ never competes with itself, which makes competition anti-reflexive. Second, competition is not necessarily symmetric. More precisely, if $\mathcal{Y} \subseteq \mathcal{X}'$ or $\mathcal{Y}' \subseteq \mathcal{X}$, then competition is *asymmetric*, meaning that if $E$ competes with $E'$ then $E'$ does not compete with $E$. Third, if $E$ competes with $E'$ then it is *possible but not certain* that support and confidence of $E$ decrease after repair. An interesting case in this regard is the following. Suppose $E$ equals $b = 1 \Rightarrow a = 0$ and $E'$ equals $b = 1 \Rightarrow a = 1$. These rules are in competition with each other. Moreover, an NNR rule can be implied that only involves $b$ which means that these rules are in fact incoherent. Now suppose this rule would be allowed in $\underline{\Omega}(\mathcal{E}_{\alpha,\beta})$, then repair will systematically remove the value $x[b]$ from $R$. In other words, it is not only possible that support and confidence of a competing rule will decrease, it is now *certain* and as a result, both rules will be invalidated. In practice, even when a lot of rules are competing with some other rule, invalidation can often be avoided (Section 7). In fact, Theorem 3 can help in detecting decreases in support and confidence and thus also in avoiding invalidation. Moreover, if invalidation of some rules cannot be avoided, invalidated rules can be passed on to the next iteration of $\rho \circ \ldots \circ \rho$. In that latter case, lower preservation can simply be enforced.

This section is concluded with a note on human supervision. It is advocated in Rammelaere and Geerts (2019), Rammelaere et al. (2017) that human supervision is an essential part to differentiate between errors and very rare events. There are at least two other aspects in the entire process where human supervision can be useful. First, it has been mentioned that the existence of incoherence indicates that some rules must be discarded. In that case, it is best to rely on human supervision to identify which rules are bogus. Second, it should be clear after presentation of the results about competing rules that in a system of repeated repair steps, a single association rule that is falsely seen as an edit rule can rapidly propagate through consecutive repair steps. It is therefore of great importance that repair steps are verified with sufficient supervision. This is why in the general methodology (Fig. 1), supervision is included in the final step.

## 6. Related work and discussion

In the past decades, there has been a tremendous interest in the development of formalisms for data quality rules and in discovery algorithms for these formalisms. Most of them adopt a static context, where either a static set of rules is used, or rules are discovered only once. In such a setting, repairing can be done with frameworks like HoloClean (Rekatsinas et al., 2017) or Llunatic (Geerts et al., 2019). To find constraints algorithms for FDs (Huhtala et al., 1999; Papenbrock et al., 2015), CFDs (Chiang & Miller, 2008; Fan et al., 2009; Rammelaere & Geerts, 2018) and DCs (Chu et al., 2013) have been studied. In more recent approaches, authors abandon the idea of an *a priori* constraint model and instead propose models to learn errors, without an explicit constraint model. The most notable example in this category is Raha: a tool that learns to recognize errors from the comparison of dirty and clean data (Mahdavi et al., 2019). Errors found this way can then be fixed by using a complementary tool called Baran (Mahdavi & Abedjan, 2020). In this setting of learning errors, the role of active learning has been investigated as well (Neutatz et al., 2019). Other approaches use learning approaches to *combine* different error mechanisms like numerical outliers, dependencies and typographical errors (Wang & He, 2019).

In the nominal case, any FD corresponds to a set of edit rules (Rammelaere & Geerts, 2019) and each association rule with maximal confidence is in fact a constant CFD (Fan et al., 2009). Moreover, minimality of constant CFDs has a strong affinity with closed frequent itemsets (Fan et al., 2009). To see how the contributions of this paper fit in the scope of CFDs and FDs, several arguments can be presented. First, this paper has considered the case where confidence does not need to be maximal but *close* to maximal. Doing so creates two main problems for which a solution has been proposed. If confidence is allowed to be lower than 1, then many rules may be found of which most are not data quality rules (i.e., precision is low) (see also (Chiang & Miller, 2008)). In that regard, redundancy of association rules collides with redundancy of edit rules and a threshold on the confidence ratios has be defined. Moreover, if confidence is 1, then for sufficiently high $\alpha$, the rules are sufficient. It has also been shown that for confidence thresholds lower than 1, this is not the case and even worse, some implied rules cannot be written as association rules. Because of this, a study of rule implication in the case of high confident association rules has been presented. A second argument is that the framework of edit rules comes with the benefit that it is guaranteed to find minimal adjustments to tuples. Third, as advocated in Rammelaere and Geerts (2019), a joint treatment of discovery and repair has been presented and it is characterized how these two procedures interact in the case of association rules.

Perhaps most affiliated with the work presented here, is the theory on forbidden itemset mining with the FBIMiner (Rammelaere & Geerts, 2019; Rammelaere et al., 2017) and its generalization based on triangular norms (Boeckling et al., 2019). Our work can be differentiated from that theory in different ways. First, the FBIMiner framework does not adopt the set cover method and has no inherent verification of satisfiability, coherence or sufficiency. Second, FBIs are edit rules with singletons only and are therefore characteristically different from strong rules. The empirical results presented below show that in real-life datasets, different rules are found and this supports the claim that, although both techniques search for some kind of edit rule, there are some differences in the underlying error mechanisms they assume. Third, FBIs are edit rules that are only detected when violated, whereas strong rules need not be violated to be accounted for.

Association analysis has also been used in data quality applications more generally. For example, in He et al. (2005), frequent itemsets are used to search for outliers. Hereby, outliers are defined as tuples that contain few frequent itemsets. In the same spirit, also the approach in Maervoet et al. (2012) considers association analysis as a fit tool for outlier detection. In other applications, association rules have also been used as a measurement tool (Alpar & Winkelsträter, 2014). In these approaches, association rules are used in a quite straightforward counting procedure, where there is for example little attention to concerns about coherence and redundancy.

## 7. Experimental evaluation

In this section, some empirical results are reported to gain more insight in the quality and coherence of strong rules, as well as their ability to repair violations. The results were obtained with a custom implementation of the FP-tree algorithm (Grahne & Zhu, 2005; Han et al., 2000) in Java (Version 8), as well as an implementation of algorithm to search and filter for association rules[4]. The rules found are compared with the FBIMiner (Rammelaere & Geerts, 2019) for finding edit rules, for which an implementation in Java based on FP-trees is used (Boeckling et al., 2019). During experiments with repairing, FP-trees built from clean data are used to find donors. Experiments were executed on a machine running Windows 10 as operating system, with an Intel 64-bit CPU (2.6 GHz) with six cores.

[4] All code is available at https://gitlab.com/ledc/ledc-dino

**Table 1**
Summary of datasets used in the experiments.

| Dataset | $\lvert R\rvert$ | $\lvert\mathcal{R}\rvert$ |
|---|---|---|
| Adult | 48842 | 11 |
| Census | 199523 | 10 |
| Trials-1 | 86670 | 13 |
| Trials-2 | 1512 | 25 |
| Trials-3 | 86670 | 27 |

**Table 2**
Impact of $\sigma$ on the number of rules and on the average number of conditions. Results are obtained with $\beta = 0.98$ and values for $\alpha$ as low as possible without violating coherence. Weak generator cases are marked with $*$.

| | $\sigma = 1$ | | $\sigma = 1.1$ | |
|---|---|---|---|---|
| | $\lvert\mathcal{E}\rvert$ | Avg. $\lvert\mathcal{X}\rvert$ | $\lvert\mathcal{E}\rvert$ | Avg. $\lvert\mathcal{X}\rvert$ |
| Adult ($\alpha = 0.01$) | 80* | 2.84 | 9 | 1.11 |
| Census ($\alpha = 0.01$) | 68 | 1.72 | 40 | 1.23 |
| Trials-1 ($\alpha = 0.01$) | 588 | 3.68 | 16 | 1.38 |
| Trials-2 ($\alpha = 0.35$) | 135 | 1.71 | 64 | 1.00 |
| Trials-3 ($\alpha = 0.2$) | 1468 | 5.91 | 23 | 1.34 |

*Description of the datasets.* In the experiments reported in the following, five datasets are used of which the main characteristics are summarized in Table 1. All used datasets can be downloaded from a public OSF repository.[5] Two datasets (Adult, Census) are taken from the UCI Machine Learning Repository[6] and three datasets (Trials-1, Trials-2 and Trials-3) are self-composed by collecting public data concerning clinical trials. The Adult dataset contains data from the 1994 US Census and relates several demographic attributes to the income of US citizens. The Census dataset contains similar data but comprises more tuples. The Trials-1 dataset contains data about the design of studies that are reported in the EudraCT database.[7] In composing this dataset, a tuple is considered for each country that executed a study so consistency can be verified in the design of clinical trials across countries in Europe. The Trials-1 dataset contains many attributes of which the domains are mostly binary-valued. The Trials-2 dataset also contains data about the design of clinical trials, but it differs from Trials-1 in the sense that it contains data of two different repositories: the German trials register.[8] and the US trials register[9] The reasons to compose Trials-2 from these two registers are twofold. First, the German register has a relatively high number of references to the US register via specification of secondary identifiers. Second, both registers report the design in a similar way and are quite well aligned in their reporting. Finally, Trials-3 contains data about the population of studies reported in the EudraCT database.

### 7.1. Analysis of parameters $\alpha$ and $\sigma$

As mentioned in Chiang and Miller (2008), searching for association rules without any restrictions would yield a very high number of rules with relative poor quality. So, in the first place, the number of rules must be reduced to a reasonable amount. This could be done by choosing $\alpha$ high, but then only high-frequent information is found. To that extent, it is verified how confidence ratios can aid in filtering rules. Recall that the usage of confidence ratios (tested with parameter $\sigma$) has the main goal of avoiding rules with an overly amount of non-relevant conditions. If $\sigma = 1$, no filtering is done. For higher values of $\sigma$, more contribution of each condition to the confidence of the rule is required (Section 3).

Table 2 shows the impact of $\sigma$ on the number of rules as well as on the average number of conditions in each rule. These results are produced for $\beta = 0.98$ and $\alpha$ as low as possible such that rules are still coherent in the sense of Definition 6, but greater than 0.01. The choice for $\beta = 0.98$ is motivated by empirical observation that confidence below 0.98 almost never leads to correct edit rules. For values of $\beta$ below 0.98, $\alpha$ needs to be relatively high in order to avoid violations of coherence. In the case of Trials-1, for example, $\beta = 0.97$ requires $\alpha \geq 0.3$ in order to obtain a coherent set. For $\beta = 0.98$, $\alpha$ can be lowered significantly without producing any incoherences. A few things can be observed.

First, values for $\alpha$ are quite different amongst the datasets. Datasets with more attributes (Trials-2 and Trials-3) require higher values for $\alpha$. Even for those high values, they produce a relative high number of rules. Nevertheless, for three datasets (Adult, Census and Trials-1), low values for $\alpha$ can be used without producing a set of rules that is incoherent. Second, rule sets that feature weak generators are marked with a '*'. It can be seen that only in one scenario, the rule set features weak generators, which is good news for the generation of implied rules because of Theorem 1. This means, for example, that verification of coherence can be done relatively fast. Third, increasing the value of $\sigma$ to 1.1 effectively reduces the number of rules. Moreover, the rules obtained for $\sigma = 1.1$ have less conditions on average. The reported averages are close to 1, but not always equal to 1, indicating that the filter produced by $\sigma = 1.1$ does not simply reject all rules which have more than 1 condition. Especially for Trials-1, which is a dataset with a relative high number of NULL values, there is a significant number of rules with two or more conditions. Although it cannot be concluded from these results that using the filter on confidence ratio helps on producing rules with high precision, it can be observed that the number of rules is reduced significantly and is in line with results reported in Chiang and Miller (2008) for datasets Adult and Census.

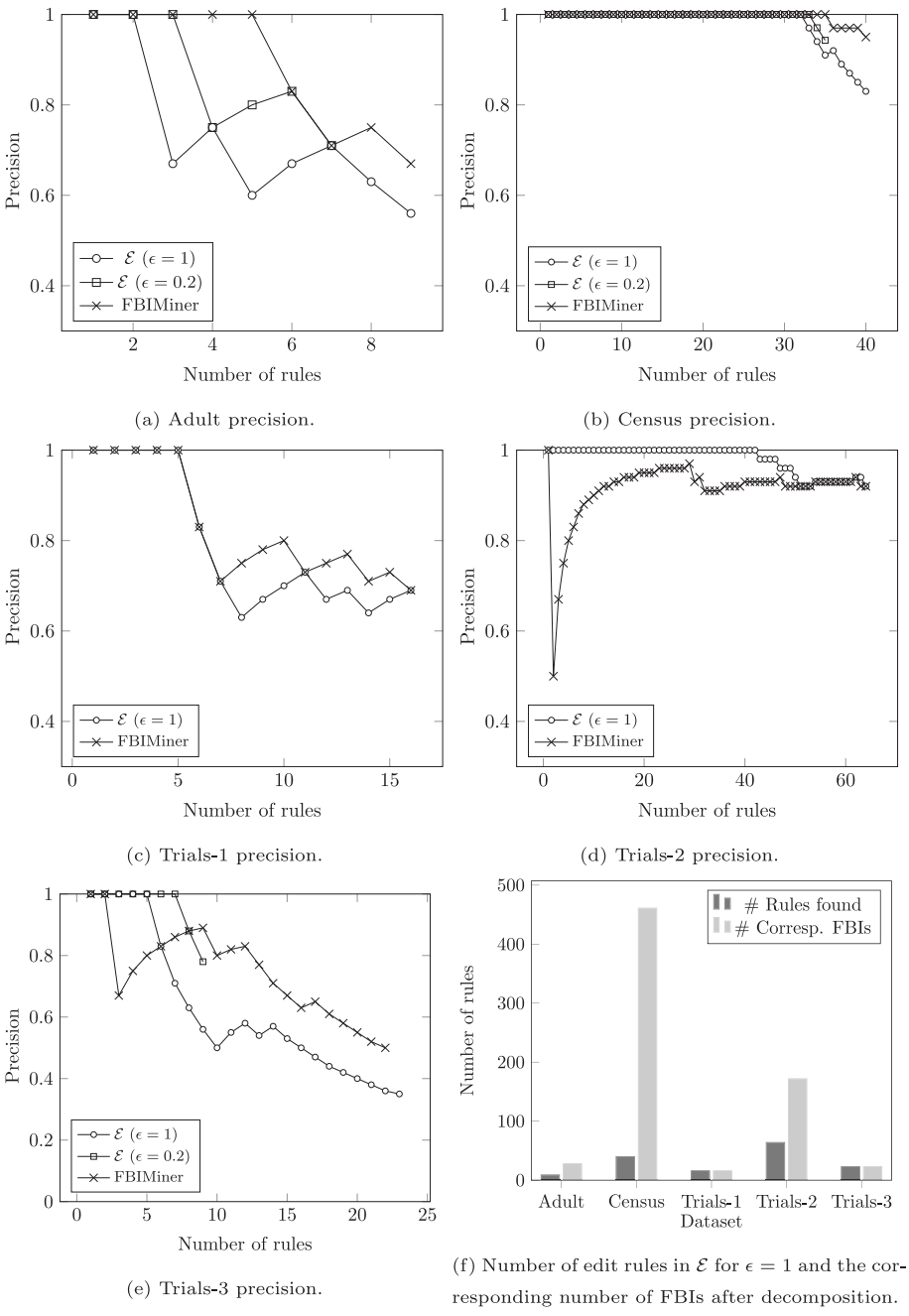
### 7.2. Precision and recall

In this section, an analysis of *precision* and *recall* of the rules is presented. The main focus is on precision as gold standards are cumbersome, but at the end of this section, some results on recall are presented for one dataset. Precision is measured as the ratio of the number of correctly identified rules over the number of rules that are found. Recall is measured as the ratio of the number of correctly identified rules over the total number of rules that should be found by a perfect algorithm.

It is assumed that the values for $\alpha$ are those listed in Table 2 and that $\sigma = 1.1$. In order to report precision, the following procedure is used. The value for $\beta$ is 0.98 and the initial value for $\epsilon$ is 1, which means that the corresponding test (Eq. (12)) is always satisfied and no rules are rejected. Rules are then sorted by their confidence (largest to smallest) and this allows to calculate a precision curve for the rules that are found. Our approach is compared with FBIMiner (Rammelaere & Geerts, 2019) by calculating a precision curve for the top-$k$ FBIs, where $k$ is the number of rules that are found. A third precision curve is obtained by filtering out rules with an $\epsilon$-threshold of 0.2. This third curve allows to verify the impact of using the $\epsilon$-filter on the precision.

Figs. 2(a) to 2(e) show the precision curves for two variants of the association-based mining and the FBIMiner, each on a different dataset. A first observation is that most precision curves display turning points. These are caused by rules that are discovered with high certainty (i.e., very high confidence or very low lift), but that do not qualify as hard data quality rules. A violation of these rules is very rare, but not an error. Because of the high certainty, these rules are high in the ranking of rules and therefore cause an early drop in precision, after which precision increases again as more correct rules are found. This phenomenon again shows the importance of supervision to identify these rules. With respect to the influence of parameter $\epsilon$, it can be seen that if precision is influenced, it is influenced positively. Indeed, for three datasets (Adult, Census and Trials-3), the precision curve for

[5] https://osf.io/p53wu/.
[6] https://archive.ics.uci.edu/ml/index.php.
[7] https://www.clinicaltrialsregister.eu/ctr-search/search.
[8] https://www.drks.de.
[9] https://clinicaltrials.gov/.




(a) Adult precision.

(b) Census precision.

(c) Trials-1 precision.

(d) Trials-2 precision.

(e) Trials-3 precision.

(f) Number of edit rules in $\mathcal{E}$ for $\epsilon = 1$ and the corresponding number of FBIs after decomposition.

**Fig. 2.** Accuracy results for association rules for different datasets compared to FBIMiner.

$\epsilon = 0.2$ lies above the precision curve for $\epsilon = 1$. For those datasets, the additional filter removes mostly erroneous rules from the end result. This effect is highly significant for Trials-3, where the precision curve for $\epsilon = 1$ drops quite fast. For the two remaining datasets (Trials-1 and Trials-2), no rules are filtered. Interestingly enough, for those datasets, precision was already relatively high. These results indicate that the test on $\epsilon$ can increase precision without affecting relative recall too much. As mentioned before, a study in Chiang and Miller (2008) compared association analysis with discovery of CFDs. In that study, the maximal reported precision of CFD discovery was 0.75 for Adult and 0.92 for Census, with a comparable number of rules. Figs. 2(a) and 2(b) show that, especially when $\epsilon = 0.2$, comparable results are obtained in terms of precision.

When comparing precision of edit rules derived from association analysis with FBIs, results for precision are *comparable*. The overall conclusion is that, in most cases, the precision curve of FBIMiner drops slightly slower than the curve for association-based methods. In order to correctly read and interpret the presented precision plots, some remarks are necessary.

First, FBIs are characterized by singleton sets only, whereas this is not the case for edit rules found via association analysis. As such, the latter may correspond to multiple FBIs. This has an effect on the precision curves in two ways. On the one hand, an edit rule found via association analysis may be represented by multiple FBIs. On the other hand, a stronger rule has a higher risk of being erroneous. To gain more insight in this difference between association rules and FBIs, association rules are decomposed for $\epsilon = 1$ by splitting their singleton complement into singletons and counted how many FBIs found by FBIMiner correspond to decomposed association rules. These results are shown in Fig. 2(f). It can be seen that for datasets Census and Trials-2 (and to a lesser extent also for Adult), decomposing the association rules yields a large number of FBIs, suggesting that for these datasets,

**Table 3**
Rules recall, tuple recall, tuple precision and tuple $f$-value on Trials-2 for association rules ($\mathcal{E}(\alpha = 0.35, \beta = 0.98, \sigma = 1.1, \epsilon = 1)$), FBIs ($\mathcal{F}_{\tau=0.0384}$), the union of both and their corresponding sufficient sets.

| Rules | Rules recall | Tuple recall | Tuple precision | Tuple $f$-value |
|---|---|---|---|---|
| $\mathcal{E}$ | 0.23 | 0.24 | 0.84 | 0.37 |
| $\underline{\Omega}(\mathcal{E})$ | 0.23 | 0.24 | 0.84 | 0.37 |
| $\mathcal{F}$ | 0.08 | 0.19 | 0.88 | 0.31 |
| $\underline{\Omega}(\mathcal{F})$ | 0.13 | 0.19 | 0.88 | 0.31 |
| $\mathcal{F} \cup \mathcal{E}$ | 0.31 | 0.33 | 0.87 | 0.48 |
| $\underline{\Omega}(\mathcal{F} \cup \mathcal{E})$ | 0.38 | 0.33 | 0.87 | 0.48 |

the rules found via association analysis are indeed stronger than the FBIs found via FBI mining. For Trials-1 and Trials-3, there is no such effect and the opposite is observed. This is due to the fact that for these datasets, some association rules tend to include some unnecessary conditions, which makes them redundant to some FBIs.

Second, edit rules found via association analysis can be non-violated in the dataset. For example, for Trials-2, there are 24 association rules that have no violations in the dataset and turn out to be correct edit rules. Opposed to that, FBIs always correspond to suspected errors. Accounting for non-violated rules is actually important as knowing more rules allows for better repairing and thus aids in obtaining higher quality after repairing.

Third, both methods might find *different* edit rules. In this regard, the notion of redundancy allows to easily compare rules from both sets and verify whether there is overlap. When further investigating inspecting this, it is found that at most half of the rules are redundant, indicating that both sets of rules capture some information not captured by the other set. For datasets Trials-1 and Trials-3, the different rules that are found, are mostly the erroneous ones. For the other three datasets, both techniques (association rules and FBIs) produce correct rules that are found by the other. As such, there is evidence that in datasets Adult, Census and Trials-2, association rules are *complementary* to edit rules found with FBIMiner.

In order to further investigate this synergy, a golden standard of a rule set for dataset Trials-2 was constructed, to our best effort. The reason to choose this dataset is twofold. On the one hand, data in Trials-2 originated from two clinical trial repositories that provide good documentation about their data.[10] Because of this documentation and the fact that both registers align quite well in their representation of design data, construction of a (near-) golden standard is a reasonable task. On the other hand, the number of rules that can be identified in Trials-2 is sufficiently high to make meaningful statements about recall. After the golden standard was constructed, it was processed with a sufficient set generation algorithm to ensure that all NNRs are present. In order to report recall in a fair manner, sets of rules are decomposed in the same way as during the analysis of precision. That is to say, all sets of rules are converted into an FBI-like representation where each involved attribute corresponds to a singleton set. In this representation, recall of a set of rules can be computed as the intersection of a decomposed rule set with the decomposed golden standard, divided by the number of rules in the decomposed golden standard. This number is called the *rule recall*. Besides rule recall, precision and recall on a *tuple level* is reported. To clarify this, when searching for tuples that fail at least one rule, the set of tuples covered by some rule set can be compared with the set of tuples that is covered by the rules in the golden standard. This way, the tuple precision (resp. tuple recall) can be computed as the size of intersection of both tuple sets divided by the number of tuples found (resp. the number of tuples that are in error according to the golden standard). In addition, we report the harmonic mean of tuple precision and recall as the tuple $f$-value.

[10] https://prsinfo.clinicaltrials.gov/definitions.html.

Table 3 reports rules recall, tuple recall and tuple precision for association rules and FBIs (denoted by $\mathcal{F}$) that correspond to the rules for which precision was reported (Fig. 2(d)), as well as a number of derived rule sets. It comes as no surprise that strong rules achieve a higher recall than FBIs because of reasons already stated above: association analysis allows to find rules that are not in violation and tend to be stronger in terms of edit rule redundancy. Interestingly, the recall of FBIs increases significantly when processed with a sufficient set generator, whereas the number of edit rules found via association analysis turns out to be a sufficient set already. An important finding is that, if edit rules from the FBI miner are combined with edit rules found via association analysis, recall again increases significantly. This confirms the hypothesis that different strategies for mining different types of edit rules are synergistic. These observations are confirmed by tuple recall. Overall recall of association analysis is higher, but the synergy of both is noticeable. Note that for tuple precision and recall, sufficient sets make no difference as they do not cover additional tuples. Tuple precision also confirms earlier findings: precision of the top-$k$ FBIs is higher than precision of the top-$k$ association rules. As a final note, it can be seen that recall is overall relatively low. This is partly due to the fact that quite stringent parameter settings are chosen as the main goal is to keep precision high. When modifying parameters in favor of recall (lower $\alpha$ for association analysis and higher lift for FBIs), recall increases at the cost of precision. However, it is also true that many constraints remain undetected because Trials-2 simply has a lot of constraints (the sufficient set generated from the golden standard contains more than 300 edit rules). This clearly leaves an opportunity for further research in which the goal should be to further increase recall, potentially by investigating other types of edit rules.

### 7.3. Analysis of repairing

The experimental evaluation is continued with studying the behavior of repairing. The evaluation is restricted to repairs where attributes in a minimal solution are assigned with different values such that a dirty tuple becomes clean. In that spirit, two strategies are described in Fellegi and Holt (1976): a *joint* repair that copies the values for attributes in the solution from a donor and a sequential *repair* that fills in values for attributes in the solution one by one. The method used here first tries to apply joint repair, where donors are selected randomly with a probability proportionate to their frequency in the dataset. If no donor is found, sequential repair is applied (see Fellegi and Holt (1976) for more details).

Two variants are considered for selection of the minimal solution. The first variant selects solutions at random and is in fact the strategy from Fellegi and Holt (1976). The second variant sorts solutions according to the number of attributes that can be repaired deductively. In other words, it counts how many attributes from a solution have only one possible value and prefers solutions with more of these attributes. If all attributes in a solution can be repaired deductively, then the repair can be done deductively.

Tables 4 and 5 summarize the results for repairing errors found by association analysis if repair steps $\rho$ are consecutively applied until no more failing rules are found. For each step, the number of edit rules found via association analysis, the size of the sufficient set, the number of rules competing with some other rule and the number of dirty tuples (i.e., the number of tuples with at least one error) are shown. In case of the "Deductive First" strategy ( Table 5), the number of dirty tuples with a deductive repair is also shown. Results were generated by using the same values for $\alpha$ and $\beta$ as in previous experiments and with $\sigma = 1.1$ and $\epsilon = 1.0$. A few interesting observations can be made.

First, the sequence of repair steps converges quite fast overall. At most four steps (Trials-2) are executed before stability is reached. Moreover, after the first step, the number of dirty tuples drops fast and the number of rules does not increase much, which indicates that there is no strong 'propagation' effect. This claim is supported by the fact

**Table 4**
Consecutive repair steps with random selection of the minimal solution.

| Dataset | $\rho$ | $\lvert\mathcal{E}\rvert$ ($\lvert\underline{\Omega}(\mathcal{E})\rvert$) | Comp. rules | #Dirty tuples |
|---|---|---|---|---|
| Adult | 1 | 9 (11) | 8 | 342 |
| | 2 | 10 (14) | 9 | 63 |
| Census | 1 | 40 (47) | 37 | 1387 |
| | 2 | 41 (48) | 38 | 49 |
| | 3 | 42 (49) | 39 | 41 |
| Trials-1 | 1 | 16 (16) | 12 | 3021 |
| | 2 | 24 (24) | 18 | 1158 |
| Trials-2 | 1 | 64 (64) | 64 | 75 |
| | 2 | 67 (67) | 66 | 54 |
| Trials-3 | 1 | 23 (39) | 23 | 2921 |

**Table 5**
Consecutive repair steps for selection of the minimal solution with the "Deductive First" strategy.

| Dataset | $\rho$ | $\lvert\mathcal{E}\rvert$($\lvert\underline{\Omega}(\mathcal{E})\rvert$) | Comp. rules | #Dirty tuples | Deductive repairs |
|---|---|---|---|---|---|
| Adult | 1 | 9 (11) | 8 | 342 | 341 |
| Census | 1 | 40 (47) | 37 | 1387 | 1361 |
| | 2 | 41 (48) | 38 | 42 | 42 |
| | 3 | 42 (49) | 39 | 26 | 26 |
| Trials-1 | 1 | 16 (16) | 12 | 3021 | 3021 |
| | 2 | 25 (25) | 20 | 1363 | 1363 |
| Trials-2 | 1 | 64 (64) | 64 | 75 | 56 |
| | 2 | 68 (68) | 67 | 73 | 73 |
| | 3 | 69 (69) | 68 | 12 | 12 |
| | 4 | 70 (70) | 69 | 12 | 12 |
| Trials-3 | 1 | 23 (39) | 23 | 2921 | 2921 |

that, in none of the cases, an incoherence was observed. In order to analyze the impact of consecutive steps on the quality of edit rules, precision and recall are measured after the four steps of the "Deductive First" strategy for dataset Trials-2. It is found that the recall in terms of rules only slightly increases to 0.24 as compared to Table 3. However, tuple recall almost doubles to 0.47 while tuple precision drops to 0.76. This provides evidence that consecutive application of repair steps allows to detect more errors. It should be noted that the decrease in precision is not small, which means that the process of consecutive repairs is best done with supervision in order to prevent propagation of errors.

Second, there is a remarkably high number of dirty tuples that can be repaired deductively. This is partly due to the fact that, often, a tuple fails only a single rule. In that case, random selection of a minimal solution has a fairly high probability of making the same decisions as the "Deductive First" strategy, especially if the left hand side of the rule involves only one attribute. As a result, there is no big difference in the number of steps that are executed.

Third, it can be seen that the number of rules that compete with at least one other rule, is very high. Nonetheless it is verified that not a single rule was invalidated by repair, even though the used repair function has no explicit strategy for avoiding rule competition.

### 7.4. Analysis of scalability

As a last part of the experimental evaluation, scalability of the approach is studied. For discovery, the method is basically a variant of traditional association analysis, but with some additional filters. Because there is no use in choosing $\beta$ below 0.98, performance is mostly influenced by the value for $\alpha$.

Scalability is first investigated in terms of the number of rows (i.e., $|R|$). For different number of tuples, 20 samples are drawn from the dataset with most rows (Census). The mean runtime in ms is then measured for (a) generating the edit rules via association analysis (b) generating the forbidden itemsets and (c) repairing the dirty tuples. During these experiments, the same parameter values as before are used, but for very small sample sizes $\alpha$ is adapted to avoid that all value combinations are frequent. Fig. 3 (left panel) shows the results of these experiments. It can be seen that both FBIMiner and finding high confident association rules scale linear in terms of increasing $|R|$ and their runtimes are comparable. It can be seen that also repairing scales linearly in terms of increasing $|R|$. Although runtimes for construction of $\underline{\Omega}(\mathcal{E})$ are not reported here, they were measured and found to be negligible. This is partly due to the strong generator theorem, but also to the relative small sizes of $\mathcal{E}$ observed in the experiments.

Next, scalability is investigated in terms of the number of attributes. For different numbers of attributes, 20 samples are drawn from the Trials-3 dataset. This dataset has a high number of attributes and the value for $\alpha$ used, leads to a high number of frequent itemsets. The mean runtime in ms is measured for (a) generating the edit rules via association analysis (b) generating the forbidden itemsets and (c) repairing the dirty tuples. The parameter values are chosen as before. Fig. 3 (right panel) shows the results of these experiments. As expected, repairing scales linearly in terms of an increasing amount of attributes because the number of rules derived increases linearly. However, both algorithms for finding edit rules show exponential behavior in this trend as the runtime approximately doubles for each additional attribute. Interesting to see is that association analysis behaves slightly better than the FBIMiner which is attributed to the pruning strategy based on $\sigma$ based on Proposition 2.

## 8. Conclusion

This paper investigates dynamical repair of data quality where constraints are association rules and where the repair method relies on set covers. In such a setting, three problems have been identified. First, the quality of rules is crucial and in that regard, two filter strategies have been presented. Second, to ensure the repair method works, rules must be verified for sufficiency and coherence. It is shown that, under reasonable conditions, this problem can be solved far more efficiently for association rules than in the general case. Third, after repairing violations, new rules can occur that are possibly again violated. This dynamic process has been studied and the conditions for termination are shown. Empirical results indicate that repeated repair of violations helps to increase recall at the cost of a mitigated drop in precision.

## CRediT authorship contribution statement

**Antoon Bronselaer:** Methodology, Software, Writing – original draft. **Toon Boeckling:** Methodology, Software, Writing – review & editing. **Filip Pattyn:** Writing – review & editing, Validation.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This research received funding from the Flemish Government, Belgium under the "Onderzoeksprogramma 338 Artificiële Intelligentie (AI) Vlaanderen" programme.

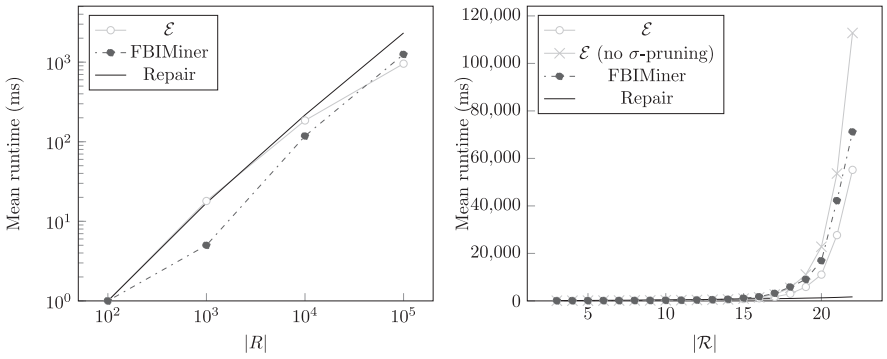



**Fig. 3.** Mean runtime (ms) for samples drawn from Census dataset with variation in $|R|$ (left panel) and samples drawn from Trials-3 dataset with variation in $|\mathcal{R}|$ (right panel).

## Appendix A. Proofs

**Proof of Proposition 3.** For $a_g$ to be a strong generator, the implied rule must be NNR and Eq. (13) needs to hold. The proof follows from showing that if Eq. (13) holds and $|\mathcal{E}| \geq 3$, then the implied rule is not NNR. It is known that, if there exists some $\mathcal{E}' \subset \mathcal{E}$ such that $E^*(g, \mathcal{E}')$ is a new rule, then this latter rule will dominate $E^*(g, \mathcal{E})$ (see (Fellegi & Holt, 1976), p.29 and (Garfinkel et al., 1986), p. 746). Now suppose $|\mathcal{E}| \geq 3$ and suppose $E^*(g, \mathcal{E})$ is a new rule. There must be a rule in $\mathcal{E}$, say $E^\bullet$, that satisfies $|E^\bullet_g| = |A_g| - 1$. For any other rule $E \in \mathcal{E}$ such that $E \neq E^\bullet$ then $E_g \neq A_g$ because $E^*(g, \mathcal{E})$ is a new rule. There are now two options. First, if $E_g \subseteq E^\bullet_g$, then $E$ can be removed from $\mathcal{E}$ and a new rule will still be obtained. Second, if $E_g \nsubseteq E^\bullet_g$, then $E_g$ contains at least one element from $A_g$ not in $E^\bullet_g$. In this case, $E_g \cup E^\bullet_g = A_g$, which means that $E^*(g, \{E, E^\bullet\})$ is a new rule that dominates $E^*(g, \mathcal{E})$. □

**Proof of Proposition 4.** Any rule initially found as an association rule in $\mathcal{E}_{\alpha,\beta}$ must satisfy conditions A1-A4. As such, for a set of contributing rules $\mathcal{E}$ composed of such initial rules and a generator $a_g$, either at least one rule involves $a_g$ with a singleton complement $E_g = \overline{\{v_g\}}$ ($a_g$ is then a strong generator) or all rules involve $a_g$ with singleton sets. □

**Proof of Proposition 5.** Follows from the fact the sum of $\Pr[R_{a=v}]$ over all $v \in A$ equals 1. □

**Proof of Proposition 6.** Assume that $\mathcal{E} = \{E, E'\}$ and that $a_g$ is a strong generator with $E'_g$ a singleton complement. For any $a_i \neq a_g$, $E^*_i$ is computed as the intersection of two sets. Because $E$ satisfies A1 and A4, there is at most one attribute $a_j$ for which $E_j$ is not a singleton and not $A_j$. Moreover, $E'_j$ is either a singleton or $A_j$. As a result, at most one attribute is involved in $E^*(g, \mathcal{E})$ with a set that is not a singleton (A4). For other attributes $a_\ell$ different from $a_j$ and $a_g$, one of $E_\ell$ and $E'_\ell$ is a singleton, or they are both $A_\ell$. Because $E^*(g, \mathcal{E})$ is a new rule, the intersection is not empty for each $i \neq g$. The intersection is thus either a singleton or an entire domain (A1). Finally, because both $E$ and $E'$ satisfy A2, there is a least one attribute that will lead to a singleton (A2). □

**Proof of Proposition 7.** By construction of implied rules, any attribute $a_i$ involved in a new rule $E^*(g, \mathcal{E})$ satisfies $E^*_i = E^1_i \cap E^2_i$. If $a_i$ is not involved in one of both rules, then $E^*_i$ must be equal to either $E^1_i$ or $E^2_i$. If $a_i$ is involved in both rules, then either $E^1_i$ or $E^2_i$ must be a singleton (A1 and A4). Clearly, because $E^*_i$ is not empty, either $E^1_i = E^2_i$ or one is a subset of the other. In either case, the proposition holds. □

**Proof of Theorem 1.** If no weak generators are initially featured, then for each attribute $a$, there is some value $v$ that does not occur as a singleton set in any $E \in \mathcal{E}_{\alpha,\beta}$. Because rules in $\mathcal{E}_{\alpha,\beta}$ satisfy condition A1, any NNR rule that is initially implied with contributing rules $\mathcal{E} \subseteq \mathcal{E}_{\alpha,\beta}$ is implied by use of a strong generator (Proposition 3). A rule implied by a strong generator satisfies conditions A1, A2 and A4 (Proposition 6) and it does not involve attributes with new singleton sets (Proposition 7). The latter ensures that, for each attribute $a$, the 'missing' singleton $\{v\}$ is never introduced in the rules. In other words, making the new rule available as a contributing rule will not cause weak generators to be featured. □

**Proof of Proposition 8.** Because $\alpha > 0.5$, there is for each $a_i \in \mathcal{R}$ only one value $v_i \in A_i$ that determines $E_i$ for any rule $E$. More specifically, if $a_i \in \mathcal{I}(E)$, then either $E_i = \{v_i\}$ or $E_i = \overline{\{v_i\}}$. Moreover, if $\alpha > 0.5$, then $\mathcal{E}_{\alpha,\beta}$ features no weak generators (Proposition 5). The combination of these observations implies that any NNR rule must be generated by transitive implication. However, for two rules $x \Rightarrow y$ and $y \Rightarrow z$ in $\mathcal{E}_{\alpha,1}$ it holds that $x \Rightarrow z$ is also in $\mathcal{E}_{\alpha,1}$. It follows that $\underline{\Omega}(\mathcal{E}_{\alpha,1}) = \mathcal{E}_{\alpha,1}$ □

**Proof of Proposition 9.** Consider a set of rules $\mathcal{E}_{\alpha,\beta}$. Because of Proposition 5 and $\alpha > \frac{1}{3}$, weak generators cannot occur for this set. As such, to generate an incoherent rule in the sense of Fellegi and Holt, two rules must be considered that involve two attributes: the generator and another one. There are then two cases that lead to an incoherence.

Case 1 If both contributing rules have a singleton complement for the generator, there are two rules in $\mathcal{E}_{\alpha,\beta}$ of the following form:

$$\{v_1\} \times \overline{\{v_2\}} \times A_3 \times \cdots \times A_k$$
$$\{v_1\} \times \overline{\{v'_2\}} \times A_3 \times \cdots \times A_k$$

which allow implication of a new rule in which only $a_1$ is involved (if $a_2$ is used as generator). Now, if $\alpha \leq 0.5$ then $\beta > 0.5$. Also, if $\alpha > 0.5$, then because confidence is lower bounded by $\alpha$, confidences of rules are greater than 0.5. So, for the two rules to occur jointly in $\mathcal{E}_{\alpha,\beta}$, both their confidences must be strictly greater than 0.5. In order for this to happen, it must hold that:

$$\left|R_{(a_1=v_1,a_2=v_2)}\right| + \left|R_{(a_1=v_1,a_2=v'_2)}\right| > \left|R_{a_1=v_1}\right|$$

which is not possible.

Case 2 If one contributing rule has a singleton for the generator, there are two rules in $\mathcal{E}_{\alpha,\beta}$ of the following form:

$$\{v_1\} \times \overline{\{v_2\}} \times A_3 \times \cdots \times A_k$$
$$\overline{\{v'_1\}} \times \{v_2\} \times A_3 \times \cdots \times A_k$$

which again leads to a new rule in which only $a_1$ is involved (if $a_2$ is used as generator). In their association rule form, the contributing rules have the form $(a_1 = v_1) \Rightarrow (a_2 = v_2)$ and $(a_2 = v_2) \Rightarrow (a_1 = v'_1)$. Suppose $(a_1 = v_1) \Rightarrow (a_2 = v_2)$ is in $\mathcal{E}_{\alpha,\beta}$, then the confidence of the second rule equals:

$$\frac{\left|R_{(a_1=v'_1,a_2=v_2)}\right|}{\left|R_{(a_2=v_2)}\right|}. \tag{A.1}$$

Clearly, the numerator must be smaller than $(1-\alpha)\cdot|R|$ because otherwise, rule $(a_1 = v_1) \Rightarrow (a_2 = v_2)$ could not have been in $\mathcal{E}_{\alpha,\beta}$. At the same time, because $Pr[R_{a_2=v_2}] \geq Pr[R_{a_1a_2=v_1v_2}] + Pr[R_{a_1a_2=v_1'v_2}]$, the denominator should be at least $2\alpha\cdot|R|$ because otherwise, the tuple $(a_1 = v_1', a_2 = v_2)$ is not a frequent itemset and the second rule cannot be found. Under these conditions, if $(a_1 = v_1', a_2 = v_2)$ is a frequent itemset, then the confidence of the rule $(a_2 = v_2) \Rightarrow (a_1 = v_1')$ is upper bounded by $\frac{1-\alpha}{2\alpha}$. Choosing $\beta$ strictly greater than this upper bound ensures that the two contributing rules that lead to the conflict can never jointly occur in the set $\mathcal{E}_{\alpha,\beta}$. □

**Proof of Theorem 2.** The proof follows from three observations. First, by definition, for any attribute in $S$, its value must be changed. If not, $S$ would not be minimal. As such, any valid combination of values for $S$ must be in $\overline{r[a_1]}\times\cdots\times\overline{r[a_s]}$. Second, any $E \notin \mathcal{E}_{S,r}$ is satisfied regardless of $\text{Rep}(r)[S]$ and thus poses no constraints on the value that $S$ can take. Third, for any $E \in \mathcal{E}_{S,r}$, each combination of values from the set $E_1\times\cdots\times E_s$ necessarily leads to an invalid tuple and can thus not be regarded as repair. Conversely, each combination of values not in the set $E_1\times\cdots\times E_s$, would lead to a tuple that satisfies $E$. Combining these three observations, all possible values that $S$ can take are considered, after which all (but only those) values that are invalid, are removed. □

**Proof of Proposition 11.** *Upper* If no new rules are found and all rules from a previous step are satisfied at the beginning of a new step, $\rho$ is finite.

*Lower* In step $i$, all rules found at the start of $i$ are satisfied at the start of iteration $i+1$. It thus follows from lower preservation that violated rules in $i+1$ are not rules found in $i$. Because all domains are finite, only a finite number of edit rules exist, which implies that $\rho\circ\ldots\circ\rho$ must be a finite sequence. □

**Proof of Proposition 12.** If $r$ is repaired, then $R$ is replaced by $R\setminus\{r\}\cup\text{Rep}(r)$. Because $E$ is in $\mathcal{E}_{\alpha,\beta}$, it can be written as an association rule $x \Rightarrow y$. Moreover, because $r$ fails $E$, it holds that $r[\mathcal{X}] = x$ and $r[\mathcal{Y}] \neq y$. There are two possible scenarios for repairing $r$. On the one hand, if an attribute from $\mathcal{X}$ changes, then $|R_{\mathcal{X}=x}|$ decreases while $|R_{\mathcal{X}\mathcal{Y}=xy}|$ remains constant. In that case, the support of $x \Rightarrow y$ is the same, while the confidence increases. On the other hand, if $\mathcal{Y}$ changes, $r[\mathcal{Y}]$ must be set to $y$ such that $|R_{\mathcal{X}\mathcal{Y}=xy}|$ increases while $|R_{\mathcal{X}=x}|$ remains constant. In that case, both support and confidence of $x \Rightarrow y$ increase. □

**Proof of Theorem 3.** Clearly, $E$ competes with $E'$ only if $r \vDash E$, $r[\mathcal{X}] = x$, $r \nvDash E'$ and $\text{Rep}(r)[\mathcal{X}] \neq x$. Now suppose (i) is not satisfied, then $\text{Rep}(r)[\mathcal{X}] = x$. Moreover, if (ii) is not satisfied, then $r \nvDash E'$ always implies either $r \nvDash E$ or $r[\mathcal{X}] \neq x$. Hence, conditions (i) and (ii) are needed for $E$ to compete with $E'$. Conversely, if conditions (i) and (ii) are satisfied, then whenever $r \vDash E$, $r[\mathcal{X}] = x$ and $r \nvDash E'$ are true, it is possible that $\text{Rep}(r)[\mathcal{X}] \neq x$ and support/confidence of $E$ decreases. Hence, (i) and (ii) are sufficient conditions. □

## References


Abiteboul, S., Hull, R., & Vianu, V. (Eds.), (1995). Foundations of databases: The logical level, (1st ed.). Addison-Wesley Longman Publishing Co., Inc.

Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *SIGMOD Record*, *22*(2), 207–216.

Alpar, P., & Winkelsträter, S. (2014). Assessment of data quality in accounting data with association rules. *Expert Systems with Applications*, *41*(5), 2259–2268.

Batini, C., Cappiello, C., Francalanci, C., & Maurino, A. (2009). Methodologies for data quality assessment and improvement. *ACM Computing Surveys*, *41*(3), 16:1–16:52.

Batini, C., & Scannapieco, M. (2006). *Data Quality: Concepts, Methodologies and Techniques (Data-Centric Systems and Applications)*. Berlin, Heidelberg: Springer-Verlag.

Boeckling, T., Bronselaer, A., & De Tré, G. (2019). Mining data quality rules based on T-dependence. *vol. 1*, In *Proceedings of the 11th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT 2019)* (pp. 184–191).

Bohannon, P., Fan, W., Geerts, F., Jia, X., & Kementsietsidis, A. (2007). Conditional functional dependencies for data cleaning. In *Proceedings of the IEEE International Conference on Data Engineering* (pp. 746–755).

Bohannon, P., Flaster, M., Fan, W., & Rastogi, R. (2005). A cost-based model and effective heuristic for repairing constraints by value modification. In *SIGMOD Conference* (pp. 143–154).

Boskovitz, A. (2008). Data editing and logic: The covering set method from the perspective of logic. (Ph.D. thesis), The Australian National University.

Brin, S., Motwani, R., Ullman, J., & Tsur, S. (1997). Dynamic itemset counting and implication rules for market basket data. In *Proceedings of the SIGMOD Conference* (pp. 255–264).

Bronselaer, A., De Mol, R., & De Tré, G. (2017). A measure-theoretic foundation for data quality. *IEEE Transactions on Fuzzy Systems*, *26*(2), 627–639. http://dx.doi.org/10.1109/TFUZZ.2017.2686807.

Caruccio, L., Deufemia, V., & Polese, G. (2016). Relaxed functional dependencies - a survey of approaches. *IEEE Transactions on Knowledge and Data Engineering*, *28*(1), 147–165.

Chiang, F., & Miller, R. J. (2008). Discovering data quality rules. In *Proceedings of the VLDB Endowment* (pp. 1166–1177).

Chu, X., Ilyas, I., & Papotti, P. (2013). Discovering denial constraints. In *Proceedings of the VLDB Endowment* (pp. 1498–1509).

Cong, G., Fan, W., Geerts, F., Jia, X., & Ma, S. (2007). Improving data quality: Consistency and accuracy. In *VLDB 2007* (pp. 315–326).

De Waal, T., Pannekoek, J., & Scholtus, S. (2011). *Handbook of Statistical Data Editing and Imputation*. Wiley.

Fan, W., & Geerts, F. (2012). *Foundations of Data Quality Management*. Morgan & Claypool Publishers.

Fan, W., Geerts, F., Jia, X., & Kementsietsidis, A. (2008). Conditional functional dependencies for capturing data inconsistencies. *ACM Transactions on Database Systems*, *33*(2), 1–48.

Fan, W., Geerts, F., Lakshmanan, L., & Xiong, M. (2009). Discovering conditional functional dependencies. In *Proceedings of the IEEE International Conference on Data Engineering* (pp. 1231–1234).

Fellegi, I., & Holt, D. (1976). A systematic approach to automatic edit and imputation. *Journal of the American Statistical Association*, *71*(353), 17–35.

Garfinkel, R., Kunnathur, A., & Liepins, G. (1986). Optimal imputation of erroneous data: categorical data, general edits. *Operations Research*, *34*(5), 744–751.

Geerts, F., Mecca, G., Papotti, P., & Santoro, D. (2019). Cleaning data with llunatic. *The VLDB Journal*, http://dx.doi.org/10.1007/s00778-019-00586-5.

Grahne, G., & Zhu, J. (2005). Fast algorithms for frequent itemset mining using FP-trees. *IEEE Transactions on Knowledge and Data Engineering*, *17*(10), 1347–1362.

Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. *SIGMOD Records*, *29*(2), 1–12.

Hang, K., Cox, L., Karr, A., Reiter, J., & Wang, Q. (2015). Simultaneous edit-imputation for continuous microdata. *Journal of the American Statistical Association*, *110*(511), 987–999.

He, Z., Xu, X., Huang, J. Z., & Deng, S. (2005). FP-Outlier: frequent pattern based outlier detection. *Computer Science Information Systems*, *2*(1), 103–118.

Huhtala, Y., Kärkkäinen, J., Porkka, P., & Toivonen, H. (1999). TANE: AN efficient algorithm for discovering functional and approximate dependencies. *The Computer Journal*, *42*(2), 100–111.

Maervoet, J., Vens, C., Vanden Berghe, G., Blockeel, H., & De Causmaecker, P. (2012). Outlier detection in relational data: A case study in geographical information systems. *Expert Systems with Applications*, *39*(5), 4718–4728.

Mahdavi, M., & Abedjan, Z. (2020). Baran: Effective error correction via a unified context representation and transfer learning. *vol. 13*, In *Proc. VLDB Endowment* (12), (pp. 1948–1961). VLDB Endowment.

Mahdavi, M., Abedjan, Z., Castro Fernandez, R., Madden, S., Ouzzani, M., Stonebraker, M., & Tang, N. (2019). Raha: A configuration-free error detection system. In *Proceedings of the 2019 International Conference on Management of Data. SIGMOD '19* (pp. 865–882). Association for Computing Machinery.

Neutatz, F., Mahdavi, M., & Abedjan, Z. (2019). ED2: A Case for active learning in error detection. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (pp. 2249–2252).

Papenbrock, T., Ehrlich, J., Marten, J., Neubert, T., Rudolph, J. P., Schönberg, M., Zwiener, J., & Naumann, F. (2015). Functional dependency discovery: an experimental evaluation of seven algorithms. *vol. 8*, In *Proceedings of the VLDB Endowment* (pp. 1082–1093).

Rammelaere, J., & Geerts, F. (2018). Revisiting conditional functional dependency discovery: Splitting the "c" from the "fd". In *Proceedings of the ECML Conference* (pp. 552–568).

Rammelaere, J., & Geerts, F. (2019). Cleaning data with forbidden itemsets. *IEEE Transactions on Knowledge and Data Engineering*, 1–1.

Rammelaere, J., Geerts, F., & Goethals, B. (2017). Cleaning data with forbidden itemsets. In *33rd IEEE International Conference on Data Engineering, ICDE 2017, San Diego, CA, USA, April 19-22, 2017* (pp. 897–908). IEEE Computer Society, http://dx.doi.org/10.1109/ICDE.2017.138.

Rekatsinas, T., Chu, X., Ilyas, I., & Ré, C. (2017). Holoclean: Holistic data repairs with probabilistic inference. In *Proceedings of the VDLB Endowment* (pp. 1190–1201). VLDB.

Scholtus, S. (2014). A generalised fellegi-holt paradigm for automatic editing. In *UN/ECE Work Session on Statistical Data Editing 2014*. http://dx.doi.org/10.13140/2.1.2211.7446.

Wang, P., & He, Y. (2019). Uni-detect: A unified approach to automated error detection in tables. In *Proceedings of the 2019 International Conference on Management of Data* (pp. 811–828).